



**UNIVERSIDAD NACIONAL
“PEDRO RUIZ GALLO”
FACULTAD DE CIENCIAS FÍSICAS Y
MATEMÁTICAS**



**ESCUELA PROFESIONAL DE INGENIERIA EN
COMPUTACION E INFORMATICA**

TESIS

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL
CLIENTE EN RESTAURANTES”**

**PRESENTADO PARA OBTENER EL TÍTULO PROFESIONAL
DE INGENIERO EN COMPUTACIÓN E
INFORMÁTICA**

AUTORES

BACH. HUAMAN VALQUI JOSE FELICIANO

BACH. VENTURA CHERO JULIO ALEX

ASESOR:

ING. VALDIVIA SALAZAR CARLOS ALBERTO

LAMBAYEQUE - PERU

2016

**"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA
MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES"**

**TESIS PARA OPTAR EL TÍTULO DE INGENIERO EN COMPUTACIÓN E
INFORMÁTICA, QUE PRESENTAN**

LOS BACHILLERES:



Bach. Jose Feliciano Huaman Valqui


Tesista



Bach. Julio Alex Ventura Chero

Tesista


ASESOR



Ing. Carlos Alberto Valdivia Salazar

Asesor

APROBADO POR:



M.Sc. Segundo Pedro Fiestas Rodriguez
Presidente



M.Sc. Consuelo I. Del Castillo Castro
Secretaria



Ing. Franklin Edinson Teran Santa Cruz
Vocal

DEDICATORIA

Dedico este proyecto en primer lugar a Dios por darme el apoyo y fuerzas para culminar con éxito este proyecto. A mis padres por su apoyo, dedicación, y comprensión ya que fueron los encargados de inculcarme los valores que tengo el día de hoy.

Julio Alex Ventura Chero

A Dios por mostrarme, día a día, que, con esfuerzo, humildad, y paciencia todo es posible.

A mis padres por su apoyo incondicional a lo largo de mi vida.

José Huamán Valqui

AGRADECIMIENTO

Agradecer a Dios, a mi familia y a todas las personas que me han apoyado en el desarrollo de la tesis; por la paciencia y el apoyo incondicional para la realización con éxito del proyecto. También expresar mi reconocimiento, de manera muy especial, a nuestro asesor: Ing. Carlos Valdivia Salazar.

Julio Alex Ventura Chero

Agradezco a Dios, mi familia por la paciencia y el apoyo incondicional.

Un agradecimiento muy especial al Ing. Carlos Valdivia Salazar por su paciencia y apoyo constante durante el desarrollo del presente proyecto.

José Huamán Valqui

RESUMEN

El presente proyecto de investigación tiene como objetivo principal, el desarrollo de una solución de escritorio y móvil que permita automatizar los procesos de restaurantes: almacén, venta, facturación, caja, atención y despachos de pedidos.

Automatizar los procesos permite aumentar la productividad y eficiencia de los restaurantes, además que la información sea de calidad, precisa y confiable.

Para realizar el presente proyecto se ha optado por la metodología de desarrollo de programación extrema (XP - SCRUM), que es una alternativa ideal para el desarrollo de software de pequeña y mediana complejidad, ya que omite diagramas y calendarios, pocas veces reales, que se deben realizar con otras metodologías de desarrollo de software.

El proyecto está conformado por varios módulos (almacén, ventas, facturación, caja y restaurante), los cuales han sido desarrollados de acuerdo a las historias de usuarios.

Las herramientas utilizadas para el desarrollo son: Visual Studio 2015, que es una herramienta muy robusta, y el sistema de manejo de bases de datos SQL Express Edition 2014.

ABSTRACT

This research project has as its main objective, the development of a mobile and desktop solution that automates the processes of restaurants: warehouse, sales, billing, cash, care and taking and despatching of orders.

Automating the processes can increase the productivity and efficiency of restaurants, besides that the information be of quality, accurate and reliable.

To carry out this project, we have opted for the Extreme Programming development methodology (XP - SCRUM), which is ideal for the development of small and medium-complexity software because it omits charts and calendars, which are rarely realistic, that must be done with other software development methodologies.

The project consists of several modules (warehouse, sales, billing, cash and restaurant), which have been developed according to the user stories.

The tools used for the development process are: Visual Studio 2015, which is a very robust tool, and the database management system SQL 2014 Express Edition.

INDICE

INTRODUCCION	12
CAPITULO I: Problemática de la Investigación	22
1.1. Realidad problemática	22
1.1.1. Planteamiento del Problema	22
1.2. Formulación del Problema	25
1.3. Justificación e Importancia de la Investigación.....	25
1.3.1. Justificación Metodológica	25
1.3.2. Justificación Práctica.....	25
1.4. Objetivos de la Investigación	26
1.4.1. Objetivo General	26
1.4.2. Objetivos Específicos.....	26
CAPITULO II: Marco Metodológico	28
2.1. Tipo de Investigación	28
2.2. Hipótesis.....	28
2.3. Variables.....	28
2.3.1. Variable Independiente	28
2.3.2. Variable Dependiente.....	28
CAPITULO III: Marco Teórico	30
3.1. Antecedentes de la Investigación.....	30
3.1.1. Antecedentes en el contexto internacional	30
3.1.2. Antecedentes en el contexto nacional.....	30
3.1.3. Antecedentes en el contexto local.....	31
3.2. Desarrollo de la Temática	32
3.2.1. Arquitectura de Desarrollo.....	32
3.2.2. Sistema Gestor de Base de Datos	35
3.2.3. Lenguaje de Programación	39
3.2.4. Servicio Web.....	45
3.2.5. Herramientas adicionales para el desarrollo	52
3.3. Selección de la Metodología a utilizar para el desarrollo de la investigación.....	75
CAPITULO IV: DESARROLLO DE LA PROPUESTA	84
4.1. Estándares para la Implementación del Análisis	85
4.2. Estándares para la Implementación del Diseño	86
4.3. Estándares para la Implementación de la Construcción	89
4.4. Pila de Productos.....	89

4.5. Plan de Entrega de los Sprints.....	110
4.6. Definición De Sprints	114
4.6.1. Sprint 1	114
4.6.2. Sprint 2	159
4.6.3. Sprint 3	204
4.6.4. Sprint 4	233
4.6.5. Sprint 5	247
4.6.6. Sprint 6	291
4.6.7. Sprint 7	347
4.7. Reuniones con el cliente	114
4.8. Plan de integracion	114
4.8.1. Consideraciones de Seguridad de Base de Datos	383
4.8.2. Consideraciones de Seguridad de la Aplicación.....	383
4.8.3. Consideraciones de Seguridad a Nivel de Opciones de Menú	383
4.8.4. Consideraciones de Seguridad de la Aplicación Móvil	384
4.9. Puesta En Marcha	385
4.9.1. Establecimiento del Plan de Implantación	385
4.9.2. Especificaciones Técnicas	386
CAPITULO V: Costos y Beneficios	387
5.1. Análisis de Costos	388
5.2. Beneficios	391
CAPITULO VI: Conclusiones.....	395
CAPITULO VII Recomendaciones	397
CAPITULO VIII Referencias Bibliograficas.....	399
ANEXOS.....	402
Anexo 1. Acta N° 01 de Reunión de Coordinación del Proyecto	403
Anexo 2. Acta N° 02 de Reunión de Coordinación del Proyecto	403
Anexo 3. Acta N° 03 de Reunión de Coordinación del Proyecto	403
Anexo 4. Acta N° 04 de Reunión de Coordinación del Proyecto	404
Anexo 5. Acta N° 05 de Reunión de Coordinación del Proyecto	404
Anexo 6. Acta N° 06 de Reunión de Coordinación del Proyecto	405
Anexo 7. Acta N° 07 de Reunión de Coordinación del Proyecto	405
Anexo 8. Acta N° 08 de Reunión de Coordinación del Proyecto	405
Anexo 9. Acta N° 09 de Reunión de Coordinación del Proyecto	406
Anexo 10. Acta N° 10 de Reunión de Coordinación del Proyecto	406
Anexo 11. Acta N° 11 de Reunión de Coordinación del Proyecto	407

Anexo 12. Acta N° 12 de Reunión de Coordinación del Proyecto	407
Anexo 13. Acta N° 13 de Reunión de Coordinación del Proyecto	407
Anexo 14. Acta N° 14 de Reunión de Coordinación del Proyecto	408
Anexo 15. Acta N° 15 de Reunión de Coordinación del Proyecto	408
Anexo 16. Acta N° 16 de Reunión de Coordinación del Proyecto	409
Anexo 17. Acta N° 17 de Reunión de Coordinación del Proyecto	409
Anexo 18. Acta N° 18 de Reunión de Coordinación del Proyecto	410
Anexo 19. Acta N° 19 de Reunión de Coordinación del Proyecto	410

INDICE DE TABLAS

Tabla 01. Diferencia entre aplicaciones nativas y web (desarrollo)	70
Tabla 02. Diferencia entre aplicaciones nativas y web (capacidades)	70
Tabla 03. Diferencia entre aplicaciones nativas y web (monetización)	71
Tabla 04. Diferencia entre aplicaciones nativas y web (entrega)	71
Tabla 05. Diferencia entre aplicaciones nativas y web (control de versiones)	72
Tabla 06. Diferencia entre aplicaciones nativas y web (fortalezas).....	73
Tabla 07. Diferencia entre aplicaciones nativas y web (debilidades)	74
Tabla 08. Pila de productos	109
Tabla 09. Calculo Días Hombre Primer Sprint.....	115
Tabla 10. Productos en el Primer Sprint.....	116
Tabla 11. 1° Historia de Usuario - Primer Sprint.....	117
Tabla 12. 2° Historia de Usuario - Primer Sprint.....	118
Tabla 13. 3° Historia de Usuario - Primer Sprint.....	120
Tabla 14. 4° Historia de Usuario - Primer Sprint.....	121
Tabla 15. 5° Historia de Usuario - Primer Sprint.....	122
Tabla 16. 6° Historia de Usuario - Primer Sprint.....	123
Tabla 17. 7° Historia de Usuario - Primer Sprint.....	125
Tabla 18. 8° Historia de Usuario - Primer Sprint.....	126
Tabla 19. 9° Historia de Usuario - Primer Sprint.....	127
Tabla 20. 10° Historia de Usuario - Primer Sprint.....	128
Tabla 21. 11° Historia de Usuario - Primer Sprint.....	129
Tabla 22. 12° Historia de Usuario - Primer Sprint.....	130
Tabla 23. Calculo Días Hombre Segundo Sprint.....	159
Tabla 24. Productos en el Segundo Sprint.....	160
Tabla 25. 1° Historia de Usuario - Segundo Sprint.....	162
Tabla 26. 2° Historia de Usuario - Segundo Sprint.....	163
Tabla 27. 3° Historia de Usuario - Segundo Sprint.....	164
Tabla 28. 4° Historia de Usuario - Segundo Sprint.....	165
Tabla 29. 5° Historia de Usuario - Segundo Sprint.....	167
Tabla 30. 6° Historia de Usuario - Segundo Sprint.....	168
Tabla 31. 7° Historia de Usuario - Segundo Sprint.....	169
Tabla 32. 8° Historia de Usuario - Segundo Sprint.....	170
Tabla 33. 9° Historia de Usuario - Segundo Sprint.....	171
Tabla 34. 10° Historia de Usuario - Segundo Sprint	172
Tabla 35. 11° Historia de Usuario - Segundo Sprint	174
Tabla 36. 12° Historia de Usuario - Segundo Sprint	175
Tabla 37. Calculo Días Hombre -Tercer Sprint.....	204
Tabla 38. Productos en el Tercer Sprint.....	205
Tabla 39. 1° Historia de Usuario - Tercer Sprint.....	207
Tabla 40. 2° Historia de Usuario - Tercer Sprint.....	208
Tabla 41. 3° Historia de Usuario - Tercer Sprint.....	209
Tabla 42. 4° Historia de Usuario – Tercer Sprint.....	210
Tabla 43. 5° Historia de Usuario - Tercer Sprint.....	212

Tabla 44. 6° Historia de Usuario - Tercer Sprint.....	213
Tabla 45. 7° Historia de Usuario - Tercer Sprint.....	214
Tabla 46. 8° Historia de Usuario – Tercer Sprint.....	215
Tabla 47. 9° Historia de Usuario - Tercer Sprint.....	216
Tabla 48. Calculo Días Hombre – Cuarto Sprint	233
Tabla 49. Productos en el cuarto Sprint	234
Tabla 50. 1° Historia de Usuario - Cuarto Sprint	235
Tabla 51. 3° Historia de Usuario - Cuarto Sprint	236
Tabla 52. 3° Historia de Usuario - Cuarto Sprint	237
Tabla 53. Calculo Días Hombre – Quinto Sprint.....	247
Tabla 54. Productos en el Quinto Sprint.....	248
Tabla 55. 1° Historia de Usuario - Quinto Sprint.....	250
Tabla 56. 2° Historia de Usuario - Quinto Sprint.....	251
Tabla 57. 3° Historia de Usuario - Quinto Sprint.....	253
Tabla 58. 4° Historia de Usuario - Quinto Sprint.....	254
Tabla 59. 5° Historia de Usuario - Quinto Sprint.....	256
Tabla 60. Calculo Días Hombre - Sexto Sprint.....	291
Tabla 61. Productos en el Sexto Sprint.....	292
Tabla 62. 1° Historia de Usuario - Sexto Sprint	294
Tabla 63. 2° Historia de Usuario - Sexto Sprint.....	296
Tabla 64. 3° Historia de Usuario - Sexto Sprint.....	297
Tabla 65. 4° Historia de Usuario - Sexto Sprint.....	298
Tabla 66. 5° Historia de Usuario - Sexto Sprint.....	300
Tabla 67. 6° Historia de Usuario - Sexto Sprint.....	302
Tabla 68. 7° Historia de Usuario - Sexto Sprint.....	303
Tabla 69. Calculo Días Hombre – Séptimo Sprint	347
Tabla 70. Productos en el Séptimo Sprint	349
Tabla 71. 1° Historia de Usuario – Séptimo Sprint	351
Tabla 72. 2° Historia de Usuario – Séptimo Sprint	352
Tabla 73. 3° Historia de Usuario – Séptimo Sprint	353
Tabla 74. 4° Historia de Usuario – Séptimo Sprint	355
Tabla 75. 5° Historia de Usuario – Séptimo Sprint	356
Tabla 76. 6° Historia de Usuario – Séptimo Sprint	357
Tabla 77. 7° Historia de Usuario – Séptimo Sprint	358
Tabla 78. Reuniones con el cliente	382
Tabla 79. Inversión en Software	388
Tabla 80. Inversión en Software	388
Tabla 81. Recursos Humanos para el Desarrollo.....	389
Tabla 82. Recursos de Materiales para el Desarrollo.....	389
Tabla 83. Costo Total por los Servicios.....	390
Tabla 84. Costos de Hardware	390
Tabla 85. Costos de Implementación.....	390
Tabla 86. Resumen de Costos.....	391
Tabla 87. Costo con el proceso actual	392
Tabla 88. Costo con el sistema.....	392
Tabla 89. Beneficio del sistema	392
Tabla 90. Comparativo: Inversión, Desarrollo, Operación, Beneficio.....	393

INDICE DE TABLAS

Tabla 01. Diferencia entre aplicaciones nativas y web (desarrollo)	70
Tabla 02. Diferencia entre aplicaciones nativas y web (capacidades)	70
Tabla 03. Diferencia entre aplicaciones nativas y web (monetización)	71
Tabla 04. Diferencia entre aplicaciones nativas y web (entrega)	71
Tabla 05. Diferencia entre aplicaciones nativas y web (control de versiones)	72
Tabla 06. Diferencia entre aplicaciones nativas y web (fortalezas)	73
Tabla 07. Diferencia entre aplicaciones nativas y web (debilidades)	74
Tabla 08. Pila de productos	109
Tabla 09. Calculo Días Hombre Primer Sprint	115
Tabla 10. Productos en el Primer Sprint	116
Tabla 11. 1° Historia de Usuario - Primer Sprint	117
Tabla 12. 2° Historia de Usuario - Primer Sprint	118
Tabla 13. 3° Historia de Usuario - Primer Sprint	120
Tabla 14. 4° Historia de Usuario - Primer Sprint	121
Tabla 15. 5° Historia de Usuario - Primer Sprint	122
Tabla 16. 6° Historia de Usuario - Primer Sprint	123
Tabla 17. 7° Historia de Usuario - Primer Sprint	125
Tabla 18. 8° Historia de Usuario - Primer Sprint	126
Tabla 19. 9° Historia de Usuario - Primer Sprint	127
Tabla 20. 10° Historia de Usuario - Primer Sprint	128
Tabla 21. 11° Historia de Usuario - Primer Sprint	129
Tabla 22. 12° Historia de Usuario - Primer Sprint	130
Tabla 23. Calculo Días Hombre Segundo Sprint	159
Tabla 24. Productos en el Segundo Sprint	160
Tabla 25. 1° Historia de Usuario - Segundo Sprint	162
Tabla 26. 2° Historia de Usuario - Segundo Sprint	163
Tabla 27. 3° Historia de Usuario - Segundo Sprint	164
Tabla 28. 4° Historia de Usuario - Segundo Sprint	165
Tabla 29. 5° Historia de Usuario - Segundo Sprint	167
Tabla 30. 6° Historia de Usuario - Segundo Sprint	168
Tabla 31. 7° Historia de Usuario - Segundo Sprint	169
Tabla 32. 8° Historia de Usuario - Segundo Sprint	170
Tabla 33. 9° Historia de Usuario - Segundo Sprint	171
Tabla 34. 10° Historia de Usuario - Segundo Sprint	172
Tabla 35. 11° Historia de Usuario - Segundo Sprint	174
Tabla 36. 12° Historia de Usuario - Segundo Sprint	175
Tabla 37. Calculo Días Hombre -Tercer Sprint	204
Tabla 38. Productos en el Tercer Sprint	205
Tabla 39. 1° Historia de Usuario - Tercer Sprint	207
Tabla 40. 2° Historia de Usuario - Tercer Sprint	208
Tabla 41. 3° Historia de Usuario - Tercer Sprint	209
Tabla 42. 4° Historia de Usuario – Tercer Sprint	210
Tabla 43. 5° Historia de Usuario - Tercer Sprint	212
Tabla 44. 6° Historia de Usuario - Tercer Sprint	213
Tabla 45. 7° Historia de Usuario - Tercer Sprint	214

Tabla 46. 8° Historia de Usuario – Tercer Sprint.....	215
Tabla 47. 9° Historia de Usuario - Tercer Sprint.....	216
Tabla 48. Calculo Días Hombre – Cuarto Sprint	233
Tabla 49. Productos en el cuarto Sprint	234
Tabla 50. 1° Historia de Usuario - Cuarto Sprint	235
Tabla 51. 3° Historia de Usuario - Cuarto Sprint	236
Tabla 52. 3° Historia de Usuario - Cuarto Sprint	237
Tabla 53. Calculo Días Hombre – Quinto Sprint.....	247
Tabla 54. Productos en el Quinto Sprint.....	248
Tabla 55. 1° Historia de Usuario - Quinto Sprint.....	250
Tabla 56. 2° Historia de Usuario - Quinto Sprint.....	251
Tabla 57. 3° Historia de Usuario - Quinto Sprint.....	253
Tabla 58. 4° Historia de Usuario - Quinto Sprint.....	254
Tabla 59. 5° Historia de Usuario - Quinto Sprint.....	256
Tabla 60. Calculo Días Hombre - Sexto Sprint.....	291
Tabla 61. Productos en el Sexto Sprint.....	292
Tabla 62. 1° Historia de Usuario - Sexto Sprint.....	294
Tabla 63. 2° Historia de Usuario - Sexto Sprint.....	296
Tabla 64. 3° Historia de Usuario - Sexto Sprint.....	297
Tabla 65. 4° Historia de Usuario - Sexto Sprint.....	298
Tabla 66. 5° Historia de Usuario - Sexto Sprint.....	300
Tabla 67. 6° Historia de Usuario - Sexto Sprint.....	302
Tabla 68. 7° Historia de Usuario - Sexto Sprint.....	303
Tabla 69. Calculo Días Hombre – Séptimo Sprint	347
Tabla 70. Productos en el Séptimo Sprint	349
Tabla 71. 1° Historia de Usuario – Séptimo Sprint	351
Tabla 72. 2° Historia de Usuario – Séptimo Sprint	352
Tabla 73. 3° Historia de Usuario – Séptimo Sprint	353
Tabla 74. 4° Historia de Usuario – Séptimo Sprint	355
Tabla 75. 5° Historia de Usuario – Séptimo Sprint	356
Tabla 76. 6° Historia de Usuario – Séptimo Sprint	357
Tabla 77. 7° Historia de Usuario – Séptimo Sprint	358
Tabla 78. Reuniones con el cliente	382
Tabla 79. Inversión en Software	388
Tabla 80. Inversión en Software	388
Tabla 81. Recursos Humanos para el Desarrollo.....	389
Tabla 82. Recursos de Materiales para el Desarrollo.....	389
Tabla 83. Costo Total por los Servicios.....	390
Tabla 84. Costos de Hardware	390
Tabla 85. Costos de Implementación.....	390
Tabla 86. Resumen de Costos.....	391
Tabla 87. Costo con el proceso actual	392
Tabla 88. Costo con el sistema.....	392
Tabla 89. Beneficio del sistema	392
Tabla 90. Comparativo: Inversión, Desarrollo, Operación, Beneficio.....	393

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 DIAGRAMA LÓGICO – PRIMER SPRINT.....	132
ILUSTRACIÓN 2 DIAGRAMA FÍSICO – PRIMER SPRINT	133
ILUSTRACIÓN 3 DIAGRAMA DE COMPONENTES – PRIMER SPRINT	134
ILUSTRACIÓN 4 INTERFAZ REGISTRAR USUARIO – PRIMER SPRINT	135
ILUSTRACIÓN 5 INTERFAZ MODIFICAR USUARIO – PRIMER SPRINT.....	135
ILUSTRACIÓN 6 INTERFAZ ELIMINAR USUARIO – PRIMER SPRINT	136
ILUSTRACIÓN 7 INTERFAZ REGISTRAR ROL – PRIMER SPRINT.....	136
ILUSTRACIÓN 8 INTERFAZ MODIFICAR ROL – PRIMER SPRINT	137
ILUSTRACIÓN 9 INTERFAZ ELIMINAR ROL – PRIMER SPRINT.....	137
ILUSTRACIÓN 10 INTERFAZ LOGUEAR USUARIO – PRIMER SPRINT	137
ILUSTRACIÓN 11 INTERFAZ ASIGNAR ACCESO – PRIMER SPRINT.....	138
ILUSTRACIÓN 12 INTERFAZ ASIGNAR ROL – PRIMER SPRINT	138
ILUSTRACIÓN 13 INTERFAZ REGISTRAR AMBIENTE – PRIMER SPRINT	139
ILUSTRACIÓN 14 INTERFAZ MODIFICAR AMBIENTE – PRIMER SPRINT.....	139
ILUSTRACIÓN 15 INTERFAZ ELIMINAR AMBIENTE – PRIMER SPRINT.....	139
ILUSTRACIÓN 16 INTERFAZ REGISTRAR MESA – PRIMER SPRINT.....	140
ILUSTRACIÓN 17 INTERFAZ MODIFICAR MESA – PRIMER SPRINT	140
ILUSTRACIÓN 18 INTERFAZ ELIMINAR MESA – PRIMER SPRINT.....	141
ILUSTRACIÓN 19 INTERFAZ REGISTRAR DIRECTORIO – PRIMER SPRINT	141
ILUSTRACIÓN 20 INTERFAZ MODIFICAR DIRECTORIO – PRIMER SPRINT.....	144
ILUSTRACIÓN 21 INTERFAZ ELIMINAR DIRECTORIO – PRIMER SPRINT.....	145
ILUSTRACIÓN 22 HISTORIAS DE USUARIO – PRIMER SPRINT.....	145
ILUSTRACIÓN 23 H. DE USUARIO REGISTRAR USUARIO.....	146
ILUSTRACIÓN 24 H. DE USUARIO MODIFICAR Y ELIMINAR USUARIO.....	146
ILUSTRACIÓN 25 H. DE USUARIO LOGUEAR UN USUARIO	146
ILUSTRACIÓN 26 H. DE USUARIO REGISTRAR ASIGNAR ACCESOS / ROLES.....	147
ILUSTRACIÓN 27 H. DE USUARIO REGISTRAR ROL	147
ILUSTRACIÓN 28 H. DE USUARIO MODIFICAR Y ELIMINAR ROL	147
ILUSTRACIÓN 29 H. DE USUARIO REGISTRAR AMBIENTE	148
ILUSTRACIÓN 30 H. DE USUARIO MODIFICAR Y ELIMINAR AMBIENTE	148
ILUSTRACIÓN 31 H. DE USUARIO REGISTRAR MESA.....	148
ILUSTRACIÓN 32 H. DE USUARIO MODIFICAR Y ELIMINAR MESA.....	149
ILUSTRACIÓN 33 H. DE USUARIO REGISTRAR DIRECTORIO.....	149
ILUSTRACIÓN 34 H. DE USUARIO MODIFICAR Y ELIMINAR DIRECTORIO.....	149
ILUSTRACIÓN 35 PROCESO DE TAREAS HISTORIA 1 – PRIMER SPRINT	150
ILUSTRACIÓN 36 PROCESO DE TAREAS HISTORIA 2 – PRIMER SPRINT	151
ILUSTRACIÓN 37 PROCESO DE TAREAS HISTORIA 3 – PRIMER SPRINT	151
ILUSTRACIÓN 38 PROCESO DE TAREAS HISTORIA 4 – PRIMER SPRINT	152
ILUSTRACIÓN 39 PROCESO DE TAREAS HISTORIA 5 – PRIMER SPRINT	152
ILUSTRACIÓN 40 PROCESO DE TAREAS HISTORIA 7 – PRIMER SPRINT	153
ILUSTRACIÓN 41 PROCESO DE TAREAS HISTORIA 9 – PRIMER SPRINT	154
ILUSTRACIÓN 42 PROCESO DE TAREAS HISTORIA 11 – PRIMER SPRINT.....	154
ILUSTRACIÓN 43 HISTORIAS DE USUARIO CONCLUIDAS – PRIMER SPRINT.....	155
ILUSTRACIÓN 44 DIAGRAMA DE BURN DOWN – PRIMER SPRINT	155
ILUSTRACIÓN 45 DIAGRAMA LÓGICO – SEGUNDO SPRINT	176

ILUSTRACIÓN 46 DIAGRAMA FÍSICO – SEGUNDO SPRINT	177
ILUSTRACIÓN 47 INTERFAZ REGISTRAR MOZO – SEGUNDO SPRINT	179
ILUSTRACIÓN 48 INTERFAZ MODIFICAR MOZO – SEGUNDO SPRINT	179
ILUSTRACIÓN 49 INTERFAZ ELIMINAR MOZO – SEGUNDO SPRINT	180
ILUSTRACIÓN 50 INTERFAZ REGISTRAR CLASE – SEGUNDO SPRINT	180
ILUSTRACIÓN 51 INTERFAZ REGISTRAR SUBCLASE – SEGUNDO SPRINT	181
ILUSTRACIÓN 52 INTERFAZ MODIFICAR CLASE – SEGUNDO SPRINT	181
ILUSTRACIÓN 53 INTERFAZ MODIFICAR SUBCLASE – SEGUNDO SPRINT	182
ILUSTRACIÓN 54 INTERFAZ ELIMINAR CLASE – SEGUNDO SPRINT	182
ILUSTRACIÓN 55 INTERFAZ ELIMINAR SUBCLASE – SEGUNDO SPRINT	182
ILUSTRACIÓN 56 INTERFAZ REGISTRAR ALMACÉN – SEGUNDO SPRINT	183
ILUSTRACIÓN 57 INTERFAZ MODIFICAR ALMACÉN – SEGUNDO SPRINT	184
ILUSTRACIÓN 58 INTERFAZ ELIMINAR ALMACÉN – SEGUNDO SPRINT	184
ILUSTRACIÓN 59 INTERFAZ REGISTRAR ARTICULO – SEGUNDO SPRINT	185
ILUSTRACIÓN 60 INTERFAZ MODIFICAR ARTICULO – SEGUNDO SPRINT	186
ILUSTRACIÓN 61 INTERFAZ ELIMINAR ARTICULO – SEGUNDO SPRINT	186
ILUSTRACIÓN 62 INTERFAZ REGISTRAR CARTA – SEGUNDO SPRINT	187
ILUSTRACIÓN 63 INTERFAZ MODIFICAR CARTA – SEGUNDO SPRINT	187
ILUSTRACIÓN 64 INTERFAZ ELIMINAR CARTA – SEGUNDO SPRINT	188
ILUSTRACIÓN 65 INTERFAZ REGISTRAR RESERVACIÓN MESA – SEGUNDO SPRINT	188
ILUSTRACIÓN 66 INTERFAZ MODIFICAR RESERVACIÓN MESA – SEGUNDO SPRINT	189
ILUSTRACIÓN 67 INTERFAZ ELIMINAR RESERVACIÓN MESA – SEGUNDO SPRINT	189
ILUSTRACIÓN 68 HISTORIAS DE USUARIO –SEGUNDO SPRINT	190
ILUSTRACIÓN 69 H. DE USUARIO REGISTRAR MOZO	190
ILUSTRACIÓN 70 H. DE USUARIO MODIFICAR Y ELIMINAR MOZO	191
ILUSTRACIÓN 71 H. DE USUARIO REGISTRAR CLASE – SUBCLASE ARTICULO	191
ILUSTRACIÓN 72 H. DE USUARIO MODIFICAR Y ELIMINAR CLASE – SUBCLASE ARTICULO	192
ILUSTRACIÓN 73 H. DE USUARIO REGISTRAR ALMACÉN	192
ILUSTRACIÓN 74 H. DE USUARIO MODIFICAR Y ELIMINAR ALMACÉN	193
ILUSTRACIÓN 75 H. DE USUARIO REGISTRAR ARTICULO	193
ILUSTRACIÓN 76 H. DE USUARIO MODIFICAR Y ELIMINAR ARTICULO	193
ILUSTRACIÓN 77 H. DE USUARIO REGISTRAR CARTA	194
ILUSTRACIÓN 78 H. DE USUARIO MODIFICAR Y ELIMINAR CARTA	194
ILUSTRACIÓN 79 H. DE USUARIO REGISTRAR RESERVACIÓN MESA	194
ILUSTRACIÓN 80 H. DE USUARIO MODIFICAR Y ELIMINAR RESERVACIÓN MESA	195
ILUSTRACIÓN 81 PROCESO DE TAREAS HISTORIA 1 –SEGUNDO SPRINT	195
ILUSTRACIÓN 82 PROCESO DE TAREAS HISTORIA 3 –SEGUNDO SPRINT	196
ILUSTRACIÓN 83 PROCESO DE TAREAS HISTORIA 5 –SEGUNDO SPRINT	197
ILUSTRACIÓN 84 PROCESO DE TAREAS HISTORIA 7 –SEGUNDO SPRINT	197
ILUSTRACIÓN 85 PROCESO DE TAREAS HISTORIA 9 –SEGUNDO SPRINT	198
ILUSTRACIÓN 86 PROCESO DE TAREAS HISTORIA 11 –SEGUNDO SPRINT	199
ILUSTRACIÓN 87 HISTORIAS DE USUARIO CONCLUIDAS –SEGUNDO SPRINT	200
ILUSTRACIÓN 88 DIAGRAMA DE BURN DOWN –SEGUNDO SPRINT	200
ILUSTRACIÓN 89 DIAGRAMA LÓGICO – TERCER SPRINT	217
ILUSTRACIÓN 90 DIAGRAMA FÍSICO – TERCER SPRINT	218
ILUSTRACIÓN 91 INTERFAZ REGISTRAR CLIENTE – TERCER SPRINT	220

ILUSTRACIÓN 92 INTERFAZ MODIFICAR CLIENTE – TERCER SPRINT	220
ILUSTRACIÓN 93 INTERFAZ ELIMINAR CLIENTE – TERCER SPRINT	221
ILUSTRACIÓN 94 INTERFAZ REGISTRAR PEDIDO – TERCER SPRINT	221
ILUSTRACIÓN 95 INTERFAZ MODIFICAR PEDIDO – TERCER SPRINT	222
ILUSTRACIÓN 96 INTERFAZ ELIMINAR PEDIDO – TERCER SPRINT	222
ILUSTRACIÓN 97 INTERFAZ ANULAR PEDIDO – TERCER SPRINT	223
ILUSTRACIÓN 98 INTERFAZ REGISTRAR COMPROBANTE DE VENTA – TERCER SPRINT	224
ILUSTRACIÓN 99 INTERFAZ MODIFICAR COMPROBANTE VENTA – TERCER SPRINT	224
ILUSTRACIÓN 100 INTERFAZ ELIMINAR COMPROBANTE VENTA – TERCER SPRINT	225
ILUSTRACIÓN 101 HISTORIAS DE USUARIO – TERCER SPRINT	225
ILUSTRACIÓN 102 H. DE USUARIO REGISTRAR CLIENTES	226
ILUSTRACIÓN 103 H. DE USUARIO MODIFICAR Y ELIMINAR CLIENTES	226
ILUSTRACIÓN 104 H. DE USUARIO REGISTRAR PEDIDO	226
ILUSTRACIÓN 105 H. DE USUARIO MODIFICAR PEDIDO	227
ILUSTRACIÓN 106 H. DE USUARIO ELIMINAR PEDIDO	227
ILUSTRACIÓN 107 H. DE USUARIO ANULAR PEDIDO	227
ILUSTRACIÓN 108 H. DE USUARIO REGISTRAR COMPROBANTE VENTA	228
ILUSTRACIÓN 109 H. DE USUARIO MODIFICAR COMPROBANTE VENTA	228
ILUSTRACIÓN 110 H. DE USUARIO ELIMINAR COMPROBANTE VENTA	228
ILUSTRACIÓN 111 PROCESO DE TAREAS HISTORIA 1 – TERCER SPRINT	229
ILUSTRACIÓN 112 PROCESO DE TAREAS HISTORIA 3 – TERCER SPRINT	230
ILUSTRACIÓN 113 PROCESO DE TAREAS HISTORIA 5 – TERCER SPRINT	230
ILUSTRACIÓN 114 PROCESO DE TAREAS HISTORIA 7 – TERCER SPRINT	231
ILUSTRACIÓN 115 PROCESO DE TAREAS HISTORIA 9 – TERCER SPRINT	232
ILUSTRACIÓN 116 HISTORIAS DE USUARIO CONCLUIDAS – TERCER SPRINT	232
ILUSTRACIÓN 117 DIAGRAMA DE BURN DOWN – TERCER SPRINT	232
ILUSTRACIÓN 118 DIAGRAMA LÓGICO – CUARTO SPRINT	238
ILUSTRACIÓN 119 DIAGRAMA FÍSICO – CUARTO SPRINT	239
ILUSTRACIÓN 120 INTERFAZ ANULAR COMPROBANTE VENTA – CUARTO SPRINT	241
ILUSTRACIÓN 121 INTERFAZ LOGUEO DE USUARIO DESPACHO COCINA – CUARTO	241
ILUSTRACIÓN 122 INTERFAZ GESTIÓN DESPACHO COCINA – CUARTO SPRINT	242
ILUSTRACIÓN 123 INTERFAZ PEDIDOS EN MESA – CUARTO SPRINT	242
ILUSTRACIÓN 124 HISTORIAS DE USUARIO – CUARTO SPRINT	243
ILUSTRACIÓN 125 H. DE USUARIO ANULAR COMPROBANTE VENTA	243
ILUSTRACIÓN 126 H. DE USUARIO DESPACHAR COCINA	243
ILUSTRACIÓN 127 H. DE USUARIO SEGUIMIENTO PEDIDOS	244
ILUSTRACIÓN 128 PROCESO DE TAREAS HISTORIA 1 – CUARTO SPRINT	244
ILUSTRACIÓN 129 PROCESO DE TAREAS HISTORIA 2 – CUARTO SPRINT	245
ILUSTRACIÓN 130 PROCESO DE TAREAS HISTORIA 3 – CUARTO SPRINT	245
ILUSTRACIÓN 131 HISTORIAS DE USUARIO CONCLUIDAS – CUARTO SPRINT	246
ILUSTRACIÓN 132 DIAGRAMA DE BURN DOWN – CUARTO SPRINT	246
ILUSTRACIÓN 133 DIAGRAMA DE CLASES – QUINTO SPRINT	257
ILUSTRACIÓN 134 DIAGRAMA LÓGICO – QUINTO SPRINT	258
ILUSTRACIÓN 135 DIAGRAMA FÍSICO – QUINTO SPRINT	259
ILUSTRACIÓN 136 DIAGRAMA DE COMPONENTES – QUINTO SPRINT	260
ILUSTRACIÓN 137 INTERFAZ CONFIGURAR CONEXIÓN – QUINTO SPRINT	261
ILUSTRACIÓN 138 INTERFAZ AUTENTICAR USUARIO – QUINTO SPRINT	262

ILUSTRACIÓN 139 SELECCIONAR EMPRESA Y REGISTRAR CREDENCIALES DE USUARIO – QUINTO SPRINT.....	263
ILUSTRACIÓN 140 REGISTRAR INICIO DE PERIODO EN EL SERVIDOR Y EN EL DISPOSITIVO MÓVIL – QUINTO SPRINT.....	264
ILUSTRACIÓN 141 REGISTRO LOCAL DE DATOS PARA TOMA DE PEDIDOS – QUINTO SPRINT.....	265
ILUSTRACIÓN 142 HISTORIAS DE USUARIO–QUINTO SPRINT	265
ILUSTRACIÓN 143 H. DE USUARIO CONFIGURAR CONEXIÓN.....	266
ILUSTRACIÓN 144 H. DE USUARIO AUTENTICAR USUARIO.....	266
ILUSTRACIÓN 145 H. DE USUARIO SELECCIONAR EMPRESA Y REGISTRAR CREDENCIALES DE USUARIO	267
ILUSTRACIÓN 146 H. DE USUARIO REGISTRAR INICIO DE PERIODO EN EL SERVIDOR Y EN EL DISPOSITIVO MÓVIL.....	267
ILUSTRACIÓN 147 H. DE USUARIO REGISTRO LOCAL DE DATOS PARA TOMA DE PEDIDOS	268
ILUSTRACIÓN 148 PROCESO DE TAREAS HISTORIA 1 –QUINTO SPRINT.....	269
ILUSTRACIÓN 149 PROCESO DE TAREAS HISTORIA 2 –QUINTO SPRINT.....	269
ILUSTRACIÓN 150 PROCESO DE TAREAS HISTORIA 3 –QUINTO SPRINT.....	270
ILUSTRACIÓN 151 PROCESO DE TAREAS HISTORIA 4 –QUINTO SPRINT.....	270
ILUSTRACIÓN 152 PROCESO DE TAREAS HISTORIA 5 –QUINTO SPRINT.....	271
ILUSTRACIÓN 153 HISTORIAS DE USUARIO CONCLUIDAS–QUINTO SPRINT	271
ILUSTRACIÓN 154 DIAGRAMA DE BURN DOWN–QUINTO SPRINT	272
ILUSTRACIÓN 155 DIAGRAMA DE CLASES –SEXTO SPRINT	304
ILUSTRACIÓN 156 DIAGRAMA FÍSICO -SEXTO SPRINT	305
ILUSTRACIÓN 157 DIAGRAMA LÓGICO -SEXTO SPRINT	306
ILUSTRACIÓN 158 DIAGRAMA DE COMPONENTES –SEXTO SPRINT	307
ILUSTRACIÓN 159 LISTADO GENERAL ACTUALIZADO DE MESAS. –SEXTO SPRINT.....	308
ILUSTRACIÓN 160 BÚSQUEDA DE RESERVAS Y LISTADO ACTUAL DE MESAS RESERVADAS – SEXTO SPRINT	309
ILUSTRACIÓN 161 INTERFAZ SELECCIÓN DE MESA Y ACTUALIZACIÓN DE SU ESTADO EN EL DISPOSITIVO Y EN EL SERVIDOR– SEXTO SPRINT.....	310
ILUSTRACIÓN 162 LISTADO DE CATEGORÍAS– SEXTO SPRINT.....	311
ILUSTRACIÓN 163 LISTADO DE ARTÍCULOS DE LA CATEGORÍA SELECCIONADA– SEXTO SPRINT.....	311
ILUSTRACIÓN 164 MANTENIMIENTO DE PEDIDO EN EL DISPOSITIVO– SEXTO SPRINT	312
ILUSTRACIÓN 165 REGISTRAR CLIENTE ENVÍO DEL PEDIDO AL SERVIDOR PARA SU PROCESAMIENTO – SEXTO SPRINT	313
ILUSTRACIÓN 166 HISTORIAS DE USUARIO–SEXTO SPRINT.....	314
ILUSTRACIÓN 167 H. DE USUARIO LISTADO GENERAL ACTUALIZADO DE MESAS.....	314
ILUSTRACIÓN 168 H. DE USUARIO BÚSQUEDA DE RESERVAS Y LISTADO ACTUAL DE MESAS RESERVADAS	315
ILUSTRACIÓN 169 H. DE USUARIO SELECCIÓN DE MESA Y ACTUALIZACIÓN DE SU ESTADO EN EL DISPOSITIVO Y EN EL SERVIDOR.....	315
ILUSTRACIÓN 170 H. DE USUARIO LISTADO DE CATEGORÍAS	315
ILUSTRACIÓN 171 H. DE USUARIO LISTADO DE ARTÍCULOS DE LA CATEGORÍA SELECCIONADA	316
ILUSTRACIÓN 172 H. DE USUARIO MANTENIMIENTO DE PEDIDO EN EL DISPOSITIVO.....	316

ILUSTRACIÓN 173 H. DE USUARIO ENVÍO DEL PEDIDO AL SERVIDOR PARA SU PROCESAMIENTO.....	316
ILUSTRACIÓN 174 PROCESO DE TAREAS HISTORIA 1 –SEXTO SPRINT.....	317
ILUSTRACIÓN 175 PROCESO DE TAREAS HISTORIA 2 –SEXTO SPRINT.....	318
ILUSTRACIÓN 176 PROCESO DE TAREAS HISTORIA 3 –SEXTO SPRINT.....	319
ILUSTRACIÓN 177 PROCESO DE TAREAS HISTORIA 4 –SEXTO SPRINT.....	319
ILUSTRACIÓN 178 PROCESO DE TAREAS HISTORIA 5 –SEXTO SPRINT.....	320
ILUSTRACIÓN 179 PROCESO DE TAREAS HISTORIA 6 –SEXTO SPRINT.....	321
ILUSTRACIÓN 180 PROCESO DE TAREAS HISTORIA 7 –SEXTO SPRINT.....	321
ILUSTRACIÓN 181 HISTORIAS DE USUARIO CONCLUIDAS – SEXTO SPRINT	322
ILUSTRACIÓN 182 DIAGRAMA DE BURN DOWN–SEXTO SPRINT.....	322
ILUSTRACIÓN 183 DIAGRAMA DE CLASES – SÉPTIMO SPRINT	359
ILUSTRACIÓN 184 DIAGRAMA LÓGICO - SÉPTIMO SPRINT	360
ILUSTRACIÓN 185 DIAGRAMA FÍSICO – SÉPTIMO SPRINT	361
ILUSTRACIÓN 186 DIAGRAMA DE COMPONENTES – SÉPTIMO SPRINT	362
ILUSTRACIÓN 187 ACTUALIZACIÓN DE PEDIDOS A RECOGER. – SÉPTIMO SPRINT.....	363
ILUSTRACIÓN 188 SELECCIÓN DE PEDIDO Y DE LOS ÍTEMS ESPECÍFICOS RECOGIDOS. – SÉPTIMO SPRINT.....	364
ILUSTRACIÓN 189 ACTUALIZACIÓN DE ESTADO DE PEDIDO E ÍTEMS RECOGIDOS TANTO EN EL SERVIDOR COMO EN EL DISPOSITIVO – SÉPTIMO SPRINT.....	365
ILUSTRACIÓN 190 LISTADO DE PEDIDOS A FACTURAR– SÉPTIMO SPRINT.....	366
ILUSTRACIÓN 191 SELECCIÓN DE PEDIDO A FACTURAR Y BÚSQUEDA DE CLIENTE – SÉPTIMO SPRIN.....	366
ILUSTRACIÓN 192 ENVÍO DE DATOS DEL PEDIDO FACTURADO Y ACTUALIZACIÓN DEL ESTADO DEL PEDIDO EN EL DISPOSITIVO– SÉPTIMO SPRINT.....	367
ILUSTRACIÓN 193 REGISTRO DE CIERRE DE PERIODO Y ELIMINACIÓN DE DATA LOCAL– SÉPTIMO SPRINT.....	368
ILUSTRACIÓN 194 HISTORIAS DE USUARIO–SÉPTIMO SPRINT	369
ILUSTRACIÓN 195 H. DE USUARIO ACTUALIZACIÓN DE PEDIDOSA RECOGER.....	369
ILUSTRACIÓN 196 H. DE USUARIO SELECCIÓN DE PEDIDO Y DE LOS ÍTEMS ESPECÍFICOS RECOGIDOS.....	370
ILUSTRACIÓN 197 H. DE USUARIO ACTUALIZACIÓN DE ESTADO DE PEDIDO E ÍTEMS RECOGIDOS TANTO EN EL SERVIDOR COMO EN EL DISPOSITIVO	370
ILUSTRACIÓN 198 H. DE USUARIO LISTADO DE PEDIDOSA FACTURAR	370
ILUSTRACIÓN 199 H. DE USUARIO SELECCIÓN DE PEDIDO A FACTURAR Y BÚSQUEDA DE CLIENTE	371
ILUSTRACIÓN 200 H. DE USUARIO ENVÍO DE DATOS DEL PEDIDO FACTURADO Y ACTUALIZACIÓN DEL ESTADO DEL PEDIDO EN EL DISPOSITIVO.....	371
ILUSTRACIÓN 201 . H. DE USUARIO REGISTRO DE CIERRE DE PERIODO Y ELIMINACIÓN DE DATA LOCAL.....	371
ILUSTRACIÓN 202 PROCESO DE TAREAS HISTORIA 1 –SÉPTIMO SPRINT.....	372
ILUSTRACIÓN 203 PROCESO DE TAREAS HISTORIA 2 –SÉPTIMO SPRINT.....	373
ILUSTRACIÓN 204 PROCESO DE TAREAS HISTORIA 3 –SÉPTIMO SPRINT.....	374
ILUSTRACIÓN 205 PROCESO DE TAREAS HISTORIA 4 –SÉPTIMO SPRINT.....	374
ILUSTRACIÓN 206 PROCESO DE TAREAS HISTORIA 5 – SÉPTIMO SPRINT.....	375
ILUSTRACIÓN 207 PROCESO DE TAREAS HISTORIA 6 –SÉPTIMO SPRINT.....	376
ILUSTRACIÓN 208 PROCESO DE TAREAS HISTORIA 7 –SÉPTIMO SPRINT.....	377

ILUSTRACIÓN 209 HISTORIAS DE USUARIO CONCLUIDAS –SÉPTIMO SPRINT	377
ILUSTRACIÓN 210 DIAGRAMA DE BURN DOWN – SÉPTIMO SPRINT	378
ILUSTRACIÓN 211 DIAGRAMA DE COMPONENTES ESCRITORIO	379
ILUSTRACIÓN 212 DIAGRAMA DE COMPONENTES MOVIL.....	380
ILUSTRACIÓN 213 DIAGRAMA DE DESPLIEGUE.....	381
ILUSTRACIÓN 214 ESTABLECIMIENTO DEL PLAN DE IMPLANTACIÓN.....	385

INTRODUCCION

El proyecto lleva por título: “Diseño e Implementación de un Sistema Informático Móvil para Mejorar la Atención del Cliente en Restaurantes” teniendo como finalidad mejorar la atención de clientes de restaurantes.

El proyecto consta de 8 capítulos los cuales son: aspecto informativo, planificación del proyecto, desarrollo de la metodología, plan de integración, definición de términos, conclusiones y recomendaciones, referencias bibliográficas.

En aspecto informativo se describe las deficiencias que se presentan en un restaurante y la solución a éstas, la definición de los objetivos, justificación e importancia y definición de conceptos teóricos utilizados en el desarrollo del proyecto.

En planificación del proyecto se describe el alcance del sistema con un cronograma de actividades de proyecto.

El desarrollo de la metodología se establece las historias de usuario priorizadas a desarrollar, se establecen los diagramas de clases y base de datos por cada Sprint, diagrama de componentes y arquitectura del sistema.

El plan de integración se describe la preparación del entorno de integración, las consideraciones de seguridad y la integración de las entregas.

CAPITULO I

PROBLEMÁTICA DE LA INVESTIGACIÓN

CAPITULO I: Problemática de la Investigación

1.1. Realidad problemática

1.1.1. Planteamiento del Problema

El desarrollo de aplicaciones para plataformas móviles se ha visto influenciado actualmente por la necesidad en la administración de negocios; en los restaurantes, por ejemplo, es más evidente la deficiencia en los servicios ofrecidos.

La principal idea es pasar de la era “de escritorio” a la era de la “portabilidad”, es decir en nuestra actualidad surge el pensamiento de que la tecnología es más valiosa si la podemos transportar con nosotros, claro está que el núcleo de ese pensamiento es estar “siempre conectado” es decir con acceso a Internet que para muchos se ha convertido en su principal medio de comunicación.

Los dispositivos móviles (Smartphones, Tablets, etc.) han evolucionado mucho en los últimos años, en especial los teléfonos celulares que han dejado de usarse solo para realizar llamadas telefónicas o enviar mensajes de texto, ahora se utilizan como medios de acceso a internet, como plataforma de juegos, muchos graban y editan video en alta definición. Al comparar los dispositivos móviles de última generación con los Computadores del 2001-2002-2003 obtenemos un dato curioso pues el hardware de los móviles supera al de dichos computadores.

Gracias al uso de teléfonos móviles inteligentes con sus variadas aplicaciones y diminutos tamaños, nuestros trabajos y vidas han sido cambiadas y con ellas se han creado nuevas formas de negocios y trabajos como son el caso de monitoreo de salud, tráfico, educación, compras, ventas, etc.

Esta revolución tecnológica también ha abierto un nuevo negocio: el del software aplicativo para dispositivos móviles. Son muchas las empresas y desarrolladores independientes se han dedicado al desarrollo de aplicaciones para móviles; es más el éxito de los móviles actualmente se determina por el número de aplicaciones de calidad que tienen en el mercado. Los principales fabricantes de móviles y quienes tienen sistemas operativos de móviles en el mercado han facilitado las herramientas de desarrollo para dichas plataformas, permitiendo crear aplicaciones interconectadas por medio del Internet siendo el límite la imaginación del desarrollador.

En países desarrollados tecnológicamente ya se ha implementa soluciones de software apoyadas en el uso de dispositivos móviles, existen hospitales que llevan la ficha de sus pacientes automatizadas con base de datos y acceden a ellas a través de Tablets, de esta manera por medio del dispositivo miran los resultados de sus exámenes, radiografías, prescripciones, etc. En tiendas se está reemplazando los típicos puntos de venta basados en sistemas de escritorio por otros basados en Tablets, que nos dan la facilidad de que los periféricos (teclado, ratón) son reemplazados por una única pantalla táctil que puede usar varias interfaces programadas, por ejemplo, simular un teclado virtual, etc.

El desarrollo del presente proyecto busca presentarse más que como una solución completa de software, como un ejemplo aplicativo de móviles en proyectos de la vida real. Cuando desarrollamos proyectos de software el principal indicador para la evaluación de la calidad del software es que éste cumpla con el propósito para el que fue creado; en la actualidad se considera también la experiencia de usuario como un indicador de peso para determinar la calidad de software. La experiencia de usuario no

se limita únicamente a la interface de usuario que programamos, sino que va mucho más allá, hasta considerar nuevos medios de interactuar con los usuarios finales. Es así que las tecnologías como: pantallas táctiles, dispositivos móviles, etc., se usan en escenarios en los que son aplicables, como medio alternativo de interacción con el usuario.

Un restaurante es el escenario ideal para la aplicación de dispositivos móviles con pantallas táctiles como medio de interacción con el usuario. Generalmente el mayor problema en un restaurante es el flujo de comunicación entre los meseros y la cocina.

Las órdenes de los clientes son anotadas en papeles que al ser llenados a mano son susceptibles a tener errores en detalles que los pedidos pueden tener, por ejemplo, en una pizzería un cliente puede sugerir que no se le ponga aceitunas en la pizza pero el mesero puede no tomar nota de este detalle y de esta manera generar insatisfacción en el cliente.

Otro inconveniente a considerar es el gasto innecesario en el que incurren los restaurantes que utilizan formatos impresos como lo son las llamadas “comandas”, formatos que con frecuencia y facilidad pueden ser extraviados perdiéndose con ellos los pedidos de los clientes. La pérdida de las comandas, y con ellas los pedidos, no solo puede generar el malestar de los clientes, en la mayoría de los casos puede ocasionar que los clientes que recibieron un mal servicio no quieran regresar al establecimiento. Es necesario también mencionar que al manejar la carta manualmente no se puede dar dinamismo a los platos con facilidad, por ejemplo, un restaurante que se especializa en asados mediante un sistema que gestione su carta puede hacer sus platos dinámicos con facilidad.

Las ventajas que una aplicación así tendría son muchas en un restaurante, mejorar la comunicación entre los meseros y la cocina; mejoraría considerablemente los tiempos de espera de los clientes, reduciría los errores en los pedidos y evitaría la pérdida de los mismos, ayudando de esta manera a incrementar la satisfacción de los clientes facilitando su fidelización.

1.2. Formulación del Problema

¿El diseño e implementación de un sistema informático móvil permitirá mejorar la atención del cliente en restaurantes en términos de reducción de tiempos de espera y propensión a errores?

1.3. Justificación e Importancia de la Investigación

1.3.1. Justificación Metodológica

Este desarrollo, como todo proyecto de software se basará principalmente en la observación y la consulta de manuales que expliquen el funcionamiento básico estandarizado para los restaurantes en general.

1.3.2. Justificación Práctica

La implementación de una aplicación que se apoye en el uso de móviles en un Restaurante supondría una mejora en la satisfacción de los clientes que recibirán un mejor servicio debido a la mejor comunicación entre los meseros que tienen contacto directo con los clientes y la cocina, de esta manera se reducirán los tiempos de espera, se reducirán los errores en los pedidos que se dan en horas pico del local, se evitará incurrir en gastos innecesarios por impresión de comandas. De esta manera la eficiencia y eficacia de los meseros se verá beneficiada. También

evitaremos el uso excesivo de papel, colaborando un poco con el medioambiente.

1.4. Objetivos de la Investigación

1.4.1. Objetivo General

Desarrollar un software de aplicación apoyado en el uso de dispositivos móviles como medio de interacción con el usuario, que permita mejorar la atención del cliente en restaurantes.

1.4.2. Objetivos Específicos

1. Recopilar información para lograr capturar los requerimientos del proceso de registro del pedido integrado con la facturación y gestión de cocina.
2. Presentar el diseño y arquitectura de la aplicación apoyada en el uso de dispositivos móviles (Tablet) como medio de interacción con el usuario.
3. Diseñar una base de datos transaccional de acuerdo a los requerimientos, así como también definir las políticas de seguridad.
4. Desarrollar la aplicación apoyada en el uso de dispositivos móviles (Tablet) con una interface amigable y fácil de usar cumpliendo los requerimientos identificados y solicitados por los usuarios.
5. Realizar las pruebas necesarias del sistema propuesto que garanticen su confiabilidad y calidad.
6. Diseñar y presentar los manuales de usuario respectivos.
7. Realizar la Evaluación Económica del Modelo Informático propuesto.

CAPITULO II

MARCO METODOLÓGICO

CAPITULO II: Marco Metodológico

21. Tipo de Investigación

Tecnológica – Formal

22. Hipótesis

El diseño e implementación de un sistema informático móvil para mejorar la atención del cliente en restaurantes permitirá incrementar la eficiencia del proceso de toma de pedidos en términos de reducción de tiempos de espera y propensión a errores.

23. Variables

2.3.1. Variable Independiente

Sistema informático para la gestión de pedidos de los clientes de restaurantes apoyada en el uso de dispositivos móviles.

2.3.2. Variable Dependiente

Atención de clientes.

CAPITULO III

MARCO TEORICO

CAPITULO III: Marco Teórico

3.1. Antecedentes de la Investigación

3.1.1. Antecedentes en el contexto internacional

- ✓ Rolando Reyes (2009) en su trabajo de investigación “Sistema de distribución, gestión de venta, auto venta y solución de problemas en ruta con comunicación hacia dispositivos móviles”, concluye lo siguiente:
 - El propósito de este trabajo es elaborar un sistema de distribución, gestión de venta, auto venta y solución de problemas en ruta con comunicación hacia dispositivos móviles, mediante el uso de la metodología RUP.

- ✓ Oscar Gabriel Bedon Teran (2006) en su trabajo de investigación “Desarrollo de un sistema de control de inventario mediante el uso de tecnología de asistente personal digital (PDA) para el almacén Arte Colonial”, concluye lo siguiente:
 - El propósito de este trabajo es automatizar y mejorar el Control de Inventario evitando errores y perdida de información, además podrán acceder en tiempo real y de forma inalámbrica a la base de datos de la aplicación, registrando de forma inmediata la información necesaria para llevar correctamente el inventario de la empresa.

3.1.2. Antecedentes en el contexto nacional

- ✓ Brallan Balarezo Paredes (2012) en su trabajo de investigación “Desarrollo de un sistema de información de registro de pedidos para ventas usando dispositivos móviles”, concluye lo siguiente:
 - El sistema permite el registro de pedidos en línea, obteniendo información de clientes y productos de manera más rápida y que provea reportes que exploten la información registrada y a su vez ayuden en la toma de decisiones.

- Un punto importante de este proyecto es la posibilidad del trabajo tanto en línea (registrando la información en una base de datos centralizada) como el trabajo fuera de línea, que se podría utilizar en caso el dispositivo móvil pierda la conexión a la base de datos.
- ✓ Lisset Marcela Chavez Vargas (2011) en su trabajo de investigación “Sistema Informático Web-Movil para el proceso de Contratación de Medidores de la Empresa Fagel Contratistas S.R.L utilizando RUP y XML”, concluye lo siguiente:
- El sistema permite verificar errores de medición y control de los medidores mediante su comparación con sistemas de patrón, consolidando reportes de contrastación vía Móvil y Web para el análisis y toma de decisión de los directivos.
 - Un punto importante de este proyecto es la posibilidad de guardar en memoria el registro de las mediciones cuando no se encuentra conectado a internet y sincronizar después cuando ya se tenga internet para la actualización en la base de datos.

3.1.3. Antecedentes en el contexto local

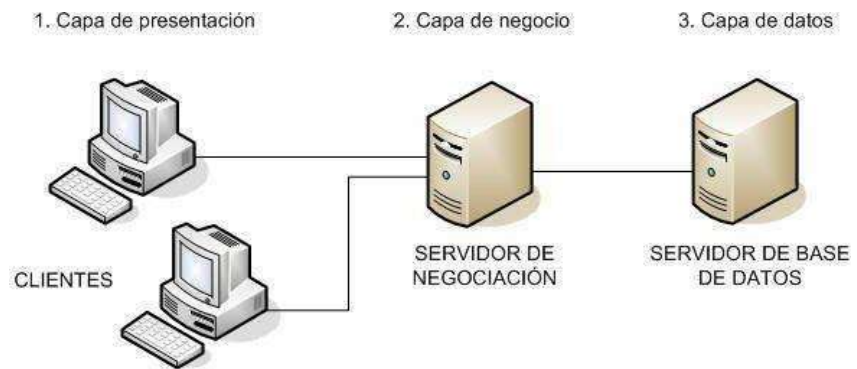
- ✓ SANCHEZ GUEVARA, OMAR ANTONIO (2011) en su trabajo de investigación “implementación de un sistema bajo tecnología wap para mejorar el proceso de ventas para los clientes en el supermercado el Súper de la ciudad de Chiclayo”, concluye lo siguiente:
- El propósito de este trabajo es maximizar el número de colocaciones de pedidos la cual implica concretar una venta, es por ello que se define una herramienta que las ha permitido registrar pedidos en tiempo real para satisfacer rápidamente los requerimientos del cliente.

3.2. Desarrollo de la Temática

3.2.1. Arquitectura de Desarrollo

3.2.1.1. Arquitectura de tres capas

El modelo de 3 o más capas que es ampliamente utilizado hoy en día en todo tipo de aplicaciones distribuidas. Esta especialmente indicado para proporcionar un adecuado soporte a los cambiantes requerimientos que demandan los procesos de negocios actuales. La separación por capas permite descomponer la aplicación de manera que cada parte sea gestionada por uno varios servidores especializados. El cliente-usuario hace una solicitud al servidor de la capa inmediata que está preparado para recibir y gestionar solicitudes de clientes. El servidor de esta capa a su vez, actúa como cliente y hace una solicitud a otro servidor, por ejemplo, para conseguir información de una base de datos. Finalmente, el servidor de la última capa devuelve un resultado que en sentido inverso a las solicitudes cruza y es tratado en cada capa hasta llegar de vuelta al usuario. **[LIB03]** La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.



Capas o Niveles:

Capa de presentación: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al sistema administrador de base de datos para almacenar o recuperar datos.

Capa de datos: es donde residen los datos. Está formada por uno o más sistemas administradores de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en una única computadora (no sería lo normal), si bien lo más usual es que haya una multitud de computadoras donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en la misma computadora, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más computadoras. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varias computadoras las cuales recibirán las peticiones de la computadora en que resida la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en una o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de computadoras sobre las cuales corre la capa de datos, y otra serie de computadoras sobre las cuales corre la base de datos. En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

- ✓ Una solución de tres capas (presentación, lógica, datos) que residen en una sola computadora (presentación + lógica + datos). Se dice, que la arquitectura de la solución es de tres capas y un nivel.
- ✓ Una solución de tres capas (presentación, lógica, datos) que residen en dos computadoras (presentación + lógica, lógica + datos). Se dice, que la arquitectura de la solución es de tres capas y dos niveles.

- ✓ Una solución de tres capas (presentación, lógica, datos) que residen en tres computadoras (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y tres niveles.

3.2.2. Sistema Gestor de Base de datos

3.2.2.1. SQL

SQL (Structured Query Language) es un lenguaje de programación diseñado específicamente para el acceso a sistemas de gestión de base de datos relacionales (SGBDR). Como la mayor parte de los sistemas actuales son de este tipo, y como el lenguaje SQL es el más ampliamente usado en estos, se puede decir sin ningún género de dudas que este lenguaje es empleado mayoritariamente en los sistemas existentes hoy en día.

Entre los grandes avances que presenta SQL Server 2014 podemos citar por ejemplo el soporte nativo de datos XML, una mejora de seguridad, pero también una mejora en necesidades de los desarrolladores, se puede trabajar en Visual Basic. Net y C# como complementos a transact SQL

Existen cuatro herramientas de trabajo principales:

- ✓ SQL Server Management Studio;
- ✓ BI Management Studio.
- ✓ SQL Profiler.
- ✓ SQL Configuration Manager

SQL Server 2014 ofrece una utilidad común para el desarrollo y la administración del servidor. Esta herramienta única redunda en el hecho de que ya no es posible desarrollar sin tener en cuenta las tareas administrativas. De la misma forma,

la administración debe realizarse teniendo en cuenta los problemas y las restricciones específicas de la programación. Solo una buena comunicación entre el desarrollador y el administrador de la base de datos puede dar lugar a una solución de calidad. Además de que las herramientas promuevan implícitamente el dialogo entre administrador y desarrollador, la convergencia entre ambos trabajos viene dictada por cuestiones técnicas y de rendimiento. Así los administradores deben asegurarse de que el método de acceso a la base de datos por programación es válido. Los desarrolladores por su parte, deben diseñar los nuevos objetos de la base de datos en asociación con el administrador para que la coherencia a nivel de la base está garantizada. **[LIB05]**

Las características de SQL Server Compact Edition incluyen lo siguiente:

- ✓ Un motor de base de datos compacto y un sólido optimizador de consultas.
- ✓ Compatibilidad con la réplica de mezcla y el acceso a datos remotos (RDA).
- ✓ Integración con Microsoft SQL Server 2014.
- ✓ Las herramientas de administración son Microsoft SQL Server Management Studio y SQL Server Management Studio Express.
- ✓ Integración con Microsoft Visual Studio 2015.
- ✓ Acceso a datos remotos y réplica de mezcla para sincronizar datos.
- ✓ Microsoft Proveedor de datos .NET Framework y .NET Compact Framework para SQL Server Compact Edition (System.Data.SqlServerCe).
- ✓ Compatibilidad con Microsoft ADO.NET y el proveedor de OLE DB para SQL Server Compact Edition.

- ✓ Un subconjunto de sintaxis SQL.
- ✓ Se implementa como una base de datos incrustada en equipos de escritorio, dispositivos móviles y Tablet PC.
- ✓ Compatibilidad con la tecnología de implementación ClicOnce.

El Modelo Relacional

Permite representar la información del mundo real de una manera intuitiva, introduciendo conceptos cotidianos y fáciles de entender por cualquier inexperto. Asimismo, mantiene información sobre las propias características de la base de datos (metadatos), que facilitan las modificaciones, disminuyendo los problemas ocasionados en las aplicaciones ya desarrolladas. Por otro lado, incorpora mecanismos de consulta muy potentes, totalmente independientes del S.G.B.D., e incluso de la organización física de los datos; el propio S.G.B.D. Es el encargado de optimizar estas preguntas en formato estándar, a sus características propias de almacenamiento.

Objetivos del Modelo Relacional

- ✓ **Independencia física:** el modo en el que se almacenan los datos no influye en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- ✓ **Independencia lógica:** el añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).

- ✓ **Flexibilidad:** en el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera. Uniformidad: las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- ✓ **Sencillez:** las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.

3.2.2.2. SQLLITE

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente – servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

Sus ventajas son:

- ✓ Su tamaño, pues tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- ✓ Realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- ✓ Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- ✓ Implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo subconsultas, generación de usuarios, vistas y triggers.
- ✓ Cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, groovy, etc.
- ✓ Es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

[WEB06]

3.2.3. Lenguaje de Programación

3.2.3.1. Visual C#

C# es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

Características:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
 - ✓ El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales a la propia fuente tales como ficheros de cabecera o ficheros IDL.
 - ✓ El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
 - ✓ No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::).
- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico string para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos.
- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una

característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

En lo referente a la encapsulación es importante señalar que aparte de los típicos modificadores **public**, **private** y **protected**, C# añade un cuarto modificador llamado **internal**, que puede combinarse con **protected** e indica que al elemento a cuya definición precede sólo puede accederse desde su mismo ensamblado.

Respecto a la herencia la diferencia de C++ y al igual que Java **C# sólo admite herencia simple de clases** ya que la múltiple provoca más quebraderos de cabeza que facilidades y en la mayoría de los casos su utilidad puede ser simulada con facilidad mediante herencia múltiple de interfaces. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.

Por otro lado, y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan de marcarse con el modificador **virtual** (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los

métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos

3.2.3.2. Visual Basic

Visual Basic es un lenguaje orientado hacia líneas que usa sintaxis clara y similar al inglés para mejorar la legibilidad. Esto, sin embargo, con frecuencia resulta en código que se extiende más allá del límite de 80 caracteres por línea, lo que obliga a los desarrolladores a desplazarse mucho. Se puede usar el carácter de subrayado para indicar al compilador que debe continuar procesando la siguiente línea como parte de la línea actual (es decir, tratar varias líneas físicas como una única línea lógica). Sin embargo, tener que escribir caracteres de subrayado constantemente siempre ha sido molesto y, de hecho, durante años la principal solicitud ha sido que el compilador “lo deduzca solo”.

Bueno, en Visual Basic 2013 el compilador puede hacerlo. Ahora, éste sabe qué símbolos (tokens), tales como comas, paréntesis y operadores, tienden a ocurrir justo antes del carácter de continuación de línea e inserta el carácter para que los desarrolladores ya no tengan que hacerlo. Por ejemplo, terminar una instrucción de Visual Basic con una coma nunca es legal; el compilador lo sabe, de manera que cuando ve una secuencia de símbolos que se ve como {coma, entrar}, infiere la presencia del carácter de continuación de línea.

Lo más interesante acerca de todas las características de Visual Basic es que incluso puede usarlas en proyectos compatibles con .NET Framework 2.0 hasta .NET Framework

3.5. Esto significa que las características como continuación de línea implícita, literales de matriz, inicializadores recolección, lambdas de instrucción o propiedades de implementación automática siempre funcionarán en proyectos existentes sin tener que hacerlas compatibles con .NET Framework 4. **[LIB04]**

Windows Forms

Los formularios Windows Forms se utilizan para desarrollar aplicaciones en las que se espera que el cliente maneje una parte significativa de la carga de trabajo de la aplicación. Entre ellas, se incluyen las aplicaciones clásicas de escritorio Win32 que solían desarrollarse en versiones anteriores de Visual Basic y Visual C++.

Todas estas aplicaciones tienen en común el hecho de que dependen de la potencia del equipo de escritorio para el procesamiento y la presentación de contenidos de alto rendimiento. Algunas aplicaciones de formularios Windows Forms pueden estar completamente autocontenidas y ejecutar todo el proceso de la aplicación en el equipo del usuario. A menudo, los juegos se escriben de este modo. Otras pueden ser parte de un sistema mayor y, fundamentalmente, utilizan el equipo de escritorio para procesar los datos proporcionados por el usuario. Por ejemplo, un sistema de punto de venta normalmente necesita una interfaz de usuario rápido y sofisticado que se crea en el equipo de escritorio, pero está enlazado a otros componentes que ejecutan el procesamiento en segundo plano.

3.2.3.3. Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un

lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo, Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

Google, por su parte, utilizó Java a su favor para crear el sistema operativo especializado para teléfonos móviles y tabletas, Android, el cual controla aplicaciones y gestiona el hardware y dispositivos. Además, al ser libre y de código abierto, permite desarrollar una gran cantidad de aplicaciones. [WEB07].

3.2.4. Servicio Web

3.2.4.1. Framework 4.5

El Microsoft .NET Framework, es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este framework.

Microsoft desea que todas las aplicaciones creadas para la plataforma Windows, sean basadas en el .NET Framework. Su objetivo es crear un marco de desarrollo de software sencillo, reduciendo las vulnerabilidades y aumentando la seguridad de los programas desarrollados.

Las soluciones pre-codificadas que forman la biblioteca .NET, cubren un gran rango de necesidades de la programación de programas. Los programadores las emplean y combinan con sus propios códigos en sus programas. El framework incluye soluciones en áreas como: la interfaz de usuario, acceso a datos, conectividad a bases de datos, criptografía, desarrollo de aplicaciones web, algoritmos numéricos y comunicación de redes.

Con esta plataforma Microsoft incursiona de lleno en el campo de los servicios web y establece al XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.

Características:

- ✓ Capacidad para reducir los reinicios del sistema mediante la detección y cierre de las aplicaciones de .NET Framework 4 durante la implementación.
- ✓ Compatibilidad con matrices mayores de 2 gigabytes (GB) en plataformas de 64 bits. Esta característica se puede habilitar en el archivo de configuración de la aplicación.
- ✓ Mayor rendimiento a través de la recolección de elementos no utilizados en segundo plano en el caso de los servidores. Cuando se usa la recolección de elementos no utilizados de los servidores en .NET Framework 4.5, se habilita automáticamente la recolección de elementos no utilizados en segundo plano.
- ✓ Compilación Just-in-time (JIT) en segundo plano, que se encuentra disponible opcionalmente en los procesadores de varios núcleos para mejorar el rendimiento de la aplicación.
- ✓ Capacidad para limitar el tiempo durante el cual el motor de expresiones regulares intentará resolver una expresión regular antes de agotar el tiempo de espera.
- ✓ Capacidad para definir la referencia cultural predeterminada de un dominio de aplicación.
- ✓ Compatibilidad de la consola con la codificación Unicode (UTF-16).
- ✓ Compatibilidad con el control de versiones de ordenación cultural de cadenas y datos de comparación.
- ✓ Mayor rendimiento al recuperar recursos.
- ✓ Mejoras en la compresión Zip para reducir el tamaño de un archivo comprimido.

- ✓ Compatibilidad con la versión 2008 del estándar de internacionalización de nombres de dominio de las aplicaciones (IDNA)
- ✓ Delegación de comparación de cadenas en el sistema operativo, que implementa Unicode 6.0, cuando se usa .NET Framework en Windows 8. Al ejecutarse en otras plataformas, .NET Framework incluye sus propios datos de comparación de cadenas, que implementan Unicode 5.x.
- ✓ Capacidad para calcular los códigos hash de cadenas en cada dominio de aplicación.

3.2.4.2. WEB API

Es un término usado para referirse al conjunto de APIs compatibles y de acceso a los dispositivos que permite a las Web apps y contenido acceder al hardware del dispositivo (como el estado de la batería o la vibración de hardware), al igual que acceso a información almacenada en el dispositivo (como el calendario o la lista de contactos). Agregando estas APIs, esperamos expandir lo que la Web puede hacer hoy y solo plataformas propietarias fueron capaces de hacer en el pasado.

ASP.NET Web API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles. ASP.NET Web API es la plataforma perfecta para crear aplicaciones RESTful en .NET Framework. Está pensada para el trabajo con HTTP y aunque su funcionamiento es muy similar al de cualquier controlador, es cierto que hay también bastantes diferencias.

Construyendo la capa de servicios a través de la Web API, podemos dejar que nuestros controladores se centren solamente en la función de la que deberían encargarse y que

no es otra que tratar con las vistas. Serán los controladores de la Web API los que ofrecerán los métodos de la capa de servicios que devuelvan a las aplicaciones clientes los datos Json, XML o cualquier otra cosa dependiendo de la negociación de contenidos efectuada a través de formatters. [WEB01]

3.2.4.3. Windows Communication Foundation (WCF)

Windows Communication Foundation (WCF) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Con WCF, es posible enviar datos como mensajes asíncronos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra enviados como XML, o tan complejos como un flujo de datos binarios. A continuación, se indican unos cuantos escenarios de ejemplo:

- ✓ Un servicio seguro para procesar transacciones comerciales.
- ✓ Un servicio que proporciona datos actualizados a otras personas, como un informe sobre tráfico u otro servicio de supervisión.
- ✓ Un servicio de chat que permite a dos personas comunicarse o intercambiar datos en tiempo real.
- ✓ Una aplicación de panel que sondea los datos de uno o varios servicios y los muestra en una presentación lógica.
- ✓ Exponer un flujo de trabajo implementado utilizando Windows Workflow Foundation como un servicio WCF.
- ✓ Una aplicación de Silverlight para sondear un servicio en busca de las fuentes de datos más recientes.

Si bien era posible crear tales aplicaciones antes de que existiera WCF, con WCF el desarrollo de extremos resulta más sencillo que nunca. En resumen, WCF se ha diseñado para ofrecer un enfoque manejable para la creación de servicios web y clientes de servicios web.

Características de WCF

WCF incluye el siguiente conjunto de características:

- Orientación a servicios

Como consecuencia del uso de los estándares de WS, WCF permite crear aplicaciones orientadas a servicios. SOA, la arquitectura orientada a servicios es el uso de servicios web para enviar y recibir datos. Los servicios tienen la ventaja general de estar débilmente acoplados entre una aplicación y otra en lugar de incluidos en el código. Una relación de acoplamiento débil implica que cualquier cliente creado en cualquier plataforma puede conectar con cualquier servicio siempre y cuando se cumplan los contratos esenciales.

- Interoperabilidad

WCF implementa los estándares del sector modernos para la interoperabilidad de servicios web.

- Varios patrones de mensajes

Los mensajes se intercambian mediante uno de los distintos patrones. El más común es el de solicitud/respuesta, en que un extremo solicita datos de otro extremo y el otro extremo responde. Existen otros patrones, como un mensaje unidireccional, en que un único extremo envía un mensaje sin esperar ninguna respuesta. Un patrón más complejo es el patrón de intercambio dúplex donde dos extremos establecen una conexión y envían datos hacia delante y hacia atrás, similar a un programa de mensajería instantánea.

- Metadatos de servicios

WCF admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares de la industria, como WSDL, Esquemas XML y WS-Policy. Estos metadatos pueden utilizarse para generar y configurar automáticamente clientes para el acceso a los servicios de WCF. Los metadatos se pueden publicar sobre HTTP y HTTPS, o utilizando el estándar Intercambio de metadatos de servicios web.

- Contratos de datos

Dado que WCF se basa en .NET Framework, también incluye métodos con código sencillo para proporcionar los contratos que desea aplicar. Uno de los tipos de contrato universales es el contrato de datos. Básicamente, mientras se escribe el código del servicio usando Visual C# o Visual Basic, la forma más sencilla de controlar los datos consiste en crear clases que representan una entidad de datos con propiedades que pertenecen a la misma. WCF incluye un completo sistema para trabajar con datos de esta manera fácil. Cuando se han creado las clases que representan los datos, el servicio genera automáticamente los metadatos que permiten a los clientes ajustarse a los tipos de datos que se han diseñado.

- Seguridad

Es posible cifrar los mensajes para proteger la privacidad, así como obligar a los usuarios a que se autenticuen antes de permitirles recibir mensajes. La seguridad puede implementarse utilizando estándares conocidos como SSL o WS-SecureConversation.

- Varios transportes y codificaciones

Los mensajes pueden enviarse con cualquiera de los protocolos y codificaciones integrados. La combinación

más frecuente de protocolo y codificación consiste en enviar mensajes SOAP codificados de texto utilizando el Protocolo de transferencia de hipertexto (HTTP) usado en World Wide Web. WCF también le permite enviar mensajes sobre TCP, canalizaciones con nombre o MSMQ. Estos mensajes pueden codificarse como texto o utilizando un formato binario optimizado. Los datos binarios pueden enviarse de manera eficaz utilizando el estándar MTOM. Si ninguno de los transportes o codificaciones proporcionados satisface sus necesidades, puede crear uno personalizado. Para obtener más información sobre los transportes y codificaciones admitidos por WCF.

- Mensajes confiables y en cola

WCF admite intercambio de mensajes confiable usando sesiones confiables implementadas sobre mensajería WS-Reliable y mediante MSMQ.

- Mensajes duraderos

Un mensaje duradero es aquel que nunca se pierde debido a una interrupción de la comunicación. Los mensajes que forman parte de un patrón de mensajes duraderos siempre se guardan en una base de datos. Si se produce una interrupción, la base de datos le permite reanudar el intercambio de mensajes cuando se restablezca la conexión. También puede crear un mensaje duradero utilizando Windows Workflow Foundation (WF).

- Transacciones

WCF también admite las transacciones que usan uno de los tres modelos de transacción: las transacciones WS-Atomic, las API del espacio de nombres System.Transactions y Coordinador de transacciones distribuidas de Microsoft.

- Compatibilidad con AJAX y REST

REST es un ejemplo de una tecnología de la Web 2.0 en evolución. WCF se puede configurar para procesar datos XML "sin formato" que no se ajustan en un sobre SOAP. WCF también se puede extender para admitir formatos XML concretos, como ATOM (un estándar popular de RSS), e incluso formatos no XML, como notación de objetos JavaScript (JSON).

- **Extensibilidad**

La arquitectura de WCF tiene varios puntos de extensibilidad. Si se necesita una función adicional, existen una serie de puntos de entrada que le permiten personalizar el comportamiento de un servicio. [WEB03]

3.2.5. Herramientas Adicionales para el Desarrollo

3.2.5.1. Bizagi Process Modeler

Es un Modelador de Procesos, un Freeware que cuenta con miles de usuarios a nivel mundial y es utilizado para diagramar y documentar procesos de manera gráfica usando la notación estándar BPMN (Business Process Modeling Notation).

Características

Su "comportamiento inteligente" permite a los usuarios no-técnicos diagramar procesos arrastrando las figuras (drag-and-drop) hacia el sitio adecuado en el diagrama. Una vez se han creado los diagramas de proceso y la documentación correspondiente, éstos pueden exportarse a Word, PDF, Visio, la web o SharePoint[5] para compartirlos con otras personas que no tengan el modelador de procesos instalado.

Aplicaciones y uso

Bizagi sigue un modelo de "no código" que permite crear aplicaciones basadas en procesos, coordinando gente y

sistemas sin la necesidad de programación. Bizagi puede ser utilizado para diagramar y automatizar cualquier tipo de proceso, desde el más simple hasta el más complejo.

Bizagi ha puesto a disposición de la comunidad un conjunto de plantillas de procesos ejecutables y listas para utilizar que se pueden descargar del sitio web. Las plantillas incluyen Mesa de ayuda, Seis Sigma, Solicitud de créditos personales, Solicitud de pólizas de automóvil, Proceso transaccional, entre otras. [WEB05]

3.2.5.2. Allfusion Erwin Data Modeler V7

AllFusion ERwin Data Modeler es una herramienta de diseño de base de datos que ayuda a los usuarios a diseñar, generar y mantener alta calidad de las aplicaciones de base de datos de altas prestaciones.

AllFusion ERwin Data Modeler permite al usuario visualizar la estructura correcta, elementos claves y el diseño optimizado de su base de datos, desde los requerimientos de un modelo lógico de información y reglas de negocio que definen la base de datos, a un modelo físico optimizado para las características específicas de la base de datos seleccionada. Su tecnología de “comparación completa” permite el desarrollo iterativo, de forma tal que los modelos están siempre sincronizados con la base de datos del usuario.

Al integrarse con entornos de desarrollo líderes, AllFusion ERwin Data Modeler también acelera la creación de aplicaciones centralizadas en datos.

AllFusion ERwin Data Modeler establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión, Erwin genera automáticamente tablas, vistas, índices, reglas de integridad

referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.

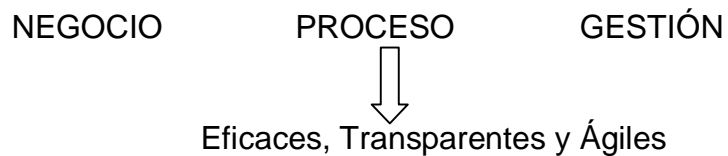
AIIFusion ERwin Data Modeler soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

Ventajas

- Erwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad - relación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y llaves importantes.
- Más que una herramienta de dibujo, Erwin automatiza el proceso de diseño de una manera inteligente.
- Se mantienen las vistas de la base de datos como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas. La migración automática garantiza la integridad referencial de la base de datos.
- Erwin soporta principalmente bases de datos relacionales que incluyen Oracle, Microsoft SQL Server. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

3.2.5.3. Business Process Management o BPM

Es un conjunto de herramientas tecnológicas que junto con una nueva filosofía de negocio permite diseñar la arquitectura empresarial modelando los procesos de negocio mediante workflows, automatizando su funcionamiento de principio a fin y permitiendo su monitorización y control. En BPM se aborda el amplio mundo de la empresa a través de sus tres (3) dimensiones básicas:



✓ Negocio

La dimensión de negocio es la dimensión de valor y de la creación de valor tanto para los clientes como para los “stakeholders” (personas interesadas en la buena marcha de la empresa como empleados, accionistas, proveedores, etcétera).

BPM facilita directamente los fines y objetivos de negocio de la compañía: crecimiento sostenido de los ingresos brutos y mejora del rendimiento mínimo; aumento de la innovación; mejora de la productividad; incremento de la fidelidad y satisfacción del cliente y niveles elevados de eficiencia del personal.

✓ Proceso

La dimensión de proceso crea valor a través de actividades estructuradas llamadas procesos. Los procesos operacionales transforman los recursos y materiales en productos o servicios para clientes y consumidores finales. Esta “transformación” es el modo en que funciona un negocio; el elixir mágico de la empresa.

Mientras más efectiva sea esta transformación, con mayor éxito se crea valor. Mediante BPM, los procesos de

negocio son más efectivos, más transparentes y más ágiles. Los problemas se resuelven antes de que se conviertan en asuntos más delicados. Los procesos producen menos errores y estos se detectan más rápido y se resuelven antes.

✓ Gestión

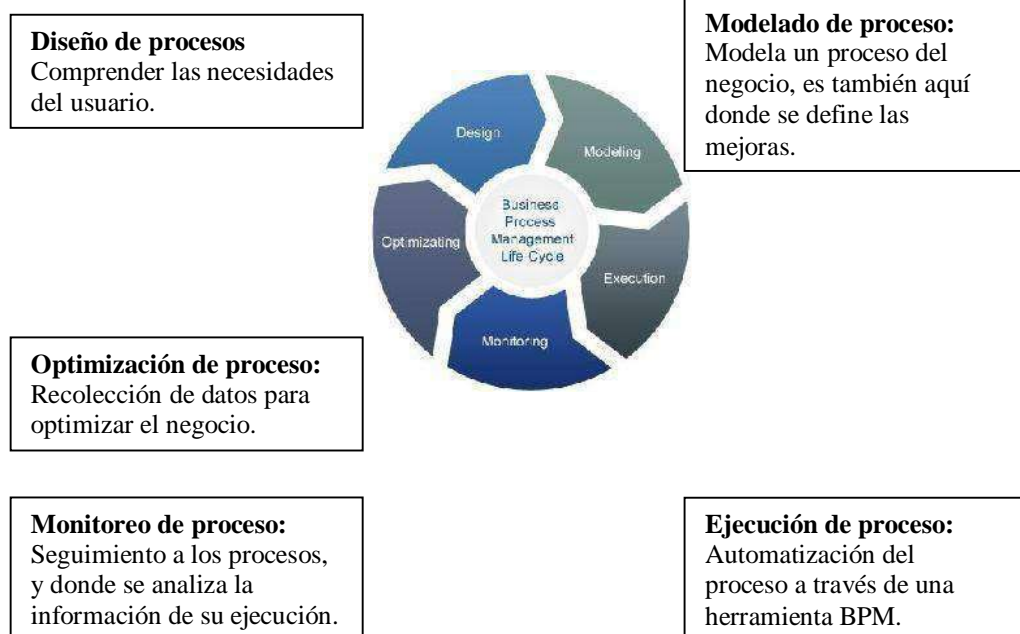
La gestión es la dimensión de capacitación. La gestión pone a las personas y a los sistemas en movimiento y empuja a los procesos a la acción en pos de los fines y objetivos del negocio. Para la gestión, los procesos son las herramientas con las que se forja el éxito empresarial. Antes de BPM, construir y aplicar estas herramientas engendraba una mezcla poco manejable de automatización de clase empresarial, muchas herramientas de escritorio aisladas, métodos y técnicas manuales y fuerza bruta.

Con BPM, puede aunar todos los sistemas, métodos, herramientas y técnicas de desarrollo de procesos y la gestión de procesos en un sistema estructurado, completo, con la visibilidad y los controles necesarios para dirigirlo y afinarlo.

Beneficios BPM

- ✓ Se convierte en un instrumento fundamental para el logro de la Formulación Estratégica de la Organización.
- ✓ Presenta una visión sistémica de la organización y sus procesos, lo que facilita y mejora su dirección y gobernabilidad.
- ✓ Mejora la interacción con los clientes, satisface sus requerimientos y facilita el camino hacia la superación de sus expectativas.

- ✓ Proporciona agilidad para adaptarse a los cambios del mercado y el entorno.
- ✓ Facilita y propicia la medición, evaluación y control de los procesos que permite identificar puntos críticos y soluciones que se traducen en mejoramiento continuo.
- ✓ Permite gestionar adecuadamente los recursos, acorde con los requerimientos de los procesos.
- ✓ Crea procesos independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y continuidad.
- ✓ Cimenta y propicia el camino hacia la automatización.



Satisfacción del Cliente

Es el nivel del estado de ánimo de una persona que resulta de comparar el rendimiento percibido de un producto o servicio con sus expectativas.

Beneficios de Lograr la Satisfacción del Cliente:

Si bien, existen diversos beneficios que toda empresa u organización puede obtener al lograr la satisfacción de

sus clientes, éstos pueden ser resumidos en tres grandes beneficios que brindan una idea clara acerca de la importancia de lograr la satisfacción del cliente:

1. El cliente satisfecho, por lo general, vuelve a comprar. Por tanto, la empresa obtiene como beneficio su lealtad y por ende, la posibilidad de venderle el mismo u otros productos adicionales en el futuro.
2. El cliente satisfecho comunica a otros sus experiencias positivas con un producto o servicio. Por tanto, la empresa obtiene como beneficio una difusión gratuita que el cliente satisfecho realiza a sus familiares, amistades y conocidos.
3. El cliente satisfecho deja de lado a la competencia. Por tanto, la empresa obtiene como beneficio un determinado lugar (participación) en el mercado.

En síntesis, toda empresa que logre la satisfacción del cliente obtendrá como beneficios: 1) La lealtad del cliente (que se traduce en futuras ventas), 2) difusión gratuita (que se traduce en nuevos clientes) y 3) una determinada participación en el mercado.

Fórmula para Determinar el Nivel de Satisfacción del Cliente:

Para darle una aplicación práctica se puede utilizar la siguiente fórmula:

$$\text{Rendimiento Percibido} - \text{Expectativas} = \text{Nivel de Satisfacción.}$$

Para aplicarla, se necesita primero obtener mediante una investigación de mercado: 1) el rendimiento percibido y 2) las expectativas que tenía el cliente antes de la compra. Luego, se les asigna un valor a los resultados obtenidos,

por ejemplo, para el rendimiento percibido se puede utilizar los siguientes parámetros:

Excelente = 10

Bueno = 7

Regular = 5

Malo = 3

En el caso de las expectativas se pueden utilizar los siguientes valores:

Expectativas Elevadas = 3

Expectativas Moderadas = 2

Expectativas Bajas = 1

Para el nivel de satisfacción se puede utilizar la siguiente escala:

Complacido: De 8 a 10

Satisfecho: de 5 a 7

Insatisfecho: Igual o Menor a 4

Finalmente, se aplica la fórmula. Por ejemplo: Si la investigación de mercado ha dado como resultado que el rendimiento percibido ha sido "Excelente" (valor: 10), pero que las expectativas que tenían los clientes eran muy "elevadas" (Valor: 3), se realiza la siguiente operación:

$$10 - 3 = 4$$

Lo que significa que el cliente está: SATISFECHO

3.2.5.4. MICROSOFT AZURE

Microsoft Azure es una colección de servicios integrados en la nube (análisis, proceso, bases de datos, móviles, redes, almacenamiento y Web) para moverse con más rapidez, llegar más lejos y ahorrar dinero.

Servicio de Windows Azure

Dentro de la plataforma, el servicio de Windows Azure es el encargado de proporcionar el alojamiento de las aplicaciones y el almacenamiento no relacional. Dichas aplicaciones deben funcionar sobre Windows Server 2008 R2. Pueden estar desarrolladas en .NET, PHP, C++, Ruby, Java. Además del servicio de ejecución, dispone de diferentes mecanismos de almacenamiento de datos: tablas NoSQL, blobs, blobs para streaming, colas de mensajes o 'drives' NTFS para operaciones de lectura / escritura a disco. [WEB02]

Características de Windows Azure

- ✓ Proceso: el servicio de proceso de Windows Azure ejecuta aplicaciones basadas en Windows Server. Estas aplicaciones se pueden crear mediante .NET Framework en lenguajes como C# y Visual Basic, o implementar sin .NET en C++, Java y otros lenguajes.
- ✓ Almacenamiento: objetos binarios grandes (blobs) proporcionan colas para la comunicación entre los componentes de las aplicaciones de Windows Azure y ofrece un tipo de tablas con un lenguaje de consulta simple.
- ✓ Servicios de infraestructura: posibilidad de desplegar de una forma sencilla máquinas virtuales con Windows Server o con distribuciones de Linux.
- ✓ Controlador de tejido: Windows Azure se ejecuta en un gran número de máquinas. El trabajo del controlador de tejido es combinar las máquinas en un solo centro de datos de Windows Azure formando un conjunto armónico. Los servicios de proceso y almacenamiento de Windows Azure se implementan encima de toda esta eficacia de procesamiento.

- ✓ Red de entrega de contenido (CDN): el almacenamiento en caché de los datos a los que se accede frecuentemente cerca de sus usuarios agiliza el acceso a esos datos.
- ✓ Connect: organizaciones interactúan con aplicaciones en la nube como si estuvieran dentro del propio firewall de la organización.¹
- ✓ Administración de identidad y acceso: La solución Active Directory permite gestionar de forma centralizada y sencilla el control de acceso y la identidad. Esta solución es perfecta para la administración de cuentas y la sincronización con directorios locales.

Azure Platform Componentes

- ✓ Windows Azure Compute, es una plataforma para hospedar y administrar aplicaciones en los centros de datos de Microsoft. Una aplicación de Windows Azure consta de uno o varios componentes denominados 'roles.' Los roles pueden ser de tres tipos: rol web, rol de trabajo y rol de máquina virtual (VM).
- ✓ Windows Azure Storage, tiene servicios básicos como parte de la cuenta de almacenamiento de Windows Azure. Los blobs, tablas y colas están accesibles a aplicaciones o instancias de aplicaciones simultáneamente.
- ✓ Microsoft SQL Azure, es un servicio de base de datos en la nube basado en las tecnologías de SQL Server. Los servicios de SQL Azure incluyen: Base de datos SQL Azure, SQL Azure Reporting y SQL Azure Data Sync Aspectos destacados de la base de datos de SQL Azure.
- ✓ Content Delivery Network (CDN) de Windows Azure, coloca copias de los datos cerca de donde estos se encuentran. La CDN de Windows Azure entrega actualmente muchos productos de Microsoft, como

Windows Update, vídeos de Zune y Bing Maps, que los clientes conocen y usan todos los días. Gracias a la incorporación de la CDN a los servicios de Windows Azure, ahora esta red a gran escala está disponible a todos los usuarios de Windows Azure.⁶

- ✓ Azure AppFabric, El servicio de Appfabric (en fase beta se llamaba .NET Services) ofrece diferentes servicios para aplicaciones. Los servicios de autenticación, autorización y mensajería permiten la comunicación segura entre aplicaciones y servicios desplegados tanto en la nube y en local. Los diferentes servicios que ofrece el servicio de AppFabric se pueden dividir en dos grandes bloques: AppFabric Service Bus y AppFabric Access Control.
- ✓ Azure Market Place, es un mercado en línea global compartir, comprar y vender aplicaciones SaaS completas y conjuntos de datos. La sección de datos de Windows Azure Marketplace incluye datos, imágenes y servicios Web en tiempo real de proveedores de datos comerciales, líderes en el sector y orígenes de datos públicos acreditados.⁸
- ✓ Azure Virtual Network, es una serie de funciones de red. Windows Azure Connect es la primera característica de Azure Virtual Network que configura la conectividad de red basada en IP entre recursos locales y de Windows Azure. Windows Azure Traffic Manager equilibra la carga del tráfico en servicios hospedados.

3.2.5.5. ANDROID

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles,

inicialmente desarrollado por Android Inc., que Google respaldó económicamente y más tarde compró en el año 2005.

Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.

El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

La versión básica de Android es conocida como Android AOSP. [WEB08]

Características:

- ✓ Diseño de dispositivo: La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
- ✓ Almacenamiento: SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
- ✓ Conectividad: Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX.GPRS, UMTS y HSDPA+.
- ✓ Mensajería: SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.
- ✓ Navegador web: El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google

Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.

- ✓ Soporte de Java: Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
- ✓ Soporte multimedia: Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
- ✓ Soporte para streaming: Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
- ✓ Soporte para hardware adicional: Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
- ✓ Entorno de desarrollo: Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de

desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.

- ✓ Google Play: Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
- ✓ Multi-táctil: Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
- ✓ Bluetooth: El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.
- ✓ Videollamada: Android soporta videollamada a través de Google Talk desde su versión HoneyComb.
- ✓ Multitarea: Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
- ✓ Características basadas en voz: La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
- ✓ Tethering: Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo,

PdaNet). Para permitir a un PC usar la conexión de datos del móvil android se podría requerir la instalación de software adicional.

Sistema Informático

Un sistema informático es un conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso. Sus partes son: hardware, software y las personas que lo usan. Por ejemplo, una computadora, sus dispositivos periféricos y la persona que la maneja, pueden constituir un sistema informático. Un sistema informático puede formar parte de un sistema de información; en este último la información, uso y acceso a la misma, no necesariamente está informatizada. Por ejemplo, el sistema de archivo de libros de una biblioteca y su actividad en general es un sistema de información. Si dentro del sistema de información hay computadoras que ayudan en la tarea de organizar la biblioteca, entonces ese es un sistema informático.

Diferencia entre sistema informático y sistema de información:

- ✓ En un sistema informático se utilizan computadoras para almacenar, procesar y/o acceder a información.
- ✓ En un sistema de información se pueden utilizar computadoras, pero no es necesario. El acceso a la información puede ser físico (por ejemplo, una persona se encarga de buscar en un archivador).
- ✓ Tanto el sistema informático como el sistema de información, incluyen a las personas que acceden o producen información dentro del sistema. Las personas tienen que capacitarse para entender el funcionamiento y procedimientos que soporta sistema.
- ✓ Ambos sistemas tienen un propósito. Por ejemplo, gestionar el acceso y distribución de libros una

biblioteca, administrar la entrada/salida de mercadería, personal y otros recursos de un comercio, etc.

Componentes de un Sistema Informático

Elementos o componentes de un sistema informático:

- ✓ Hardware: Computadoras.
- ✓ Software: Medios de programación, sistema operativo, programas de comunicaciones, aplicaciones.
- ✓ Manual de instrucciones para el operador, usuarios y proveedores de la información del sistema.
- ✓ Base de datos e información en general.
- ✓ Trabajadores relacionados al sistema, encargados de emitir o utilizar información que resulta de la actividad del sistema informático.
- ✓ Otros dispositivos: Sensores, captadores de información, equipos auxiliares, etc.
- ✓ Redes, líneas y equipos de comunicación entre computadoras.
- ✓ Dispositivos de almacenamiento de información para ser utilizados por el sistema.

Dispositivo Móvil

Los dispositivos móviles son una forma de interconectarse con otros dispositivos de manera sencilla facilitando la forma de comunicarse con las de más persona. Permiten la fácil conexión a redes móviles, y la permanente conexión a Internet

Los dispositivos móviles (también conocidos como computadora de mano, Palmtop o simplemente handheld) son aparatos de pequeño tamaño. Con algunas capacidades de procesamiento, móviles o no, con conexión permanente o intermitente a una red, con memoria limitada, diseñados

específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

Tipos de dispositivos:

- ✓ Dispositivo Móvil de Datos Limitados (Limited Data Mobile Device): teléfonos móviles clásicos. Se caracterizan por tener una pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS y acceso WAP.
- ✓ Dispositivo Móvil de Datos Básicos (Basic Data Mobile Device): se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a emails, lista de direcciones, SMS, y, en algunos casos, un navegador web básico. Un típico ejemplo de este tipo de dispositivos son los teléfonos inteligentes (smartphones).
- ✓ Dispositivo Móvil de Datos Mejorados (Enhanced Data Mobile Device): se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 pixels), navegación de tipo stylus, y que ofrecen las mismas características que el “Dispositivo Móvil de Datos Básicos” (Basic Data Mobile Devices) más aplicaciones nativas como aplicaciones de Microsoft Office Mobile (Word, Excel, PowerPoint) y aplicaciones corporativas usuales, en versión móvil, como Sap, portales intranet, etc. Este tipo de dispositivos incluyen los S.O. como Windows Mobile.

Aplicación Nativa

Una aplicación nativa es una aplicación para cierto tipo de dispositivos (smartphone, tablet, etc.) Se instalan directamente en el dispositivo. Los usuarios típicamente las

adquieren a través de tiendas online (marketplaces) tales como “The App Store” o “Google Play”.

Ejemplos de aplicaciones nativas son Camera+ para dispositivos iOS y KeePassDroid para dispositivos Android.

[WEB04]

Aplicación Web

Cuando hablamos de aplicaciones web, nos referimos a aplicaciones habilitadas para Internet que tienen funcionalidades específicas para dispositivos móviles. A estas aplicaciones se accede a través de los navegadores web de los dispositivos móviles (ejem. En iPhone, el navegador por defecto es Safari) y no necesitan ser descargadas e instaladas en los dispositivos.

Diferencias Entre Aplicaciones Nativas Y Aplicaciones Web

Desarrollo

Aplicaciones Nativas	Aplicaciones Web
Cada plataforma de desarrollo de aplicaciones móviles (por ejemplo, iOS, Android) requiere su propio proceso de desarrollo	Se ejecuta en el navegador web del dispositivo móvil y cada uno puede tener sus propias características y peculiaridades
Cada plataforma de desarrollo de aplicaciones móviles tiene su propio lenguaje de programación nativo: Java (Android), Objective-C (iOS) y Visual C # (Windows Mobile), etc.	Las aplicaciones web móviles están escritas en HTML5, CSS3, JavaScript y lenguajes del lado del servidor o entornos de aplicaciones Web de la elección del desarrollador (por ejemplo, PHP, Rails, Python)
Kits de desarrollo de software estandarizados (SDK), herramientas de	No hay kits de desarrollo de software estándar (SDK) que se requiera que los desarrolladores

desarrollo y los elementos de la interfaz de usuario común (botones, campos de introducción de texto, etc.) a menudo son proporcionados por el fabricante de la plataforma	utilicen para crear una aplicación web para móviles
Existen herramientas y frameworks para ayudar en el desarrollo de aplicaciones que se implementarán en múltiples plataformas de sistemas operativos móviles y navegadores web (por ejemplo, PhoneGap, Sencha Touch 2, Appcelerator Titanium, etc.)	

Tabla 01. Diferencia entre aplicaciones nativas y web (desarrollo)

Capacidades

Aplicaciones Nativas	Aplicaciones Web
Puede interactuar con las características nativas del dispositivo, la información y el hardware (cámara, acelerómetro, etc.)	Las aplicaciones web móviles pueden acceder a una cantidad limitada de características e información nativas del dispositivo (orientación, geolocalización, medios de comunicación, etc.)
Puede ejecutar operaciones complejas y prolongadas en un hilo diferente al hilo de interfaz gráfica. Para esto dispone de APIs y herramientas especializadas, propias del sistema operativo.	No dispone de herramientas para ejecutar procesos en segundo plano (background) por lo que todo el trabajo se termina ejecutando en el mismo hilo de interfaz gráfica.

Tabla 02. Diferencia entre aplicaciones nativas y web (capacidades)

Monetización

Aplicaciones Nativas	Aplicaciones Web
Plataformas de anuncios específicas para móviles como AdMob (aunque puede haber restricciones establecidas por el fabricante del dispositivo móvil)	Las aplicaciones web móviles pueden obtener beneficios económicos a través de publicidad y tarifas de suscripción
Los desarrolladores tienen la capacidad de cobrar un precio por descarga y las tiendas de aplicaciones suelen manejar el proceso de pago (a cambio de un porcentaje de las ventas)	Cobrar a los usuarios para que utilicen la aplicación web móvil requiere configurar un muro de pago o un sistema basado en suscripción

Tabla 03. Diferencia entre aplicaciones nativas y web (monetización)

Método de entrega

Aplicaciones Nativas	Aplicaciones Web
Se descarga en un dispositivo móvil	Se accede a través del navegador web de un dispositivo móvil
Una vez instalado se ejecuta como una aplicación independiente (no se necesita navegador web)	No hay necesidad de instalar nuevo software
Los usuarios deben descargar e instalar actualizaciones de aplicaciones de forma manual	Las actualizaciones se realizan en el servidor web sin intervención del usuario

Hay tiendas virtuales para ayudar a los usuarios a encontrar su aplicación	Dado que no existe tiendas de aplicaciones para las web móviles, puede ser más difícil para los usuarios encontrar la aplicación
--	--

Tabla 04. Diferencia entre aplicaciones nativas y web (entrega)

Control de versiones de la aplicación

Aplicaciones Nativas	Aplicaciones Web
Algunos usuarios pueden optar por ignorar una actualización, dando lugar a que diferentes usuarios ejecutan diferentes versiones de la aplicación	Todos los usuarios acceden a la misma versión

Tabla 05. Diferencia entre aplicaciones nativas y web (control de versiones)

Fortalezas

Aplicaciones Nativas	Aplicaciones Web
Normalmente se ejecutan más rápido que las aplicaciones web.	Utilizan el mismo código base para todas las plataformas.
Las tiendas de aplicaciones ayudan a los usuarios a encontrar las aplicaciones nativas	Los usuarios no tienen que ir a una tienda de aplicaciones, descargar la aplicación o instalarla
Los procesos de aprobación de las tiendas de aplicaciones pueden	Pueden ser lanzadas en cualquier forma y en cualquier momento, ya que ninguna

brindar confianza a los usuarios sobre la calidad y la seguridad de las aplicaciones	tienda de aplicaciones tiene que aprobar su publicación.
Los fabricantes de dispositivos pueden ayudar a acelerar el desarrollo brindando herramientas, soporte y buenas prácticas de desarrollo	Si ya se dispone de una aplicación web, se puede adaptar para que utilice diseño adaptativo.

Tabla 06. Diferencia entre aplicaciones nativas y web (fortalezas)

Debilidades

Aplicaciones Nativas	Aplicaciones Web
Suelen ser más costosas de desarrollar, especialmente si necesitan implementar en diferentes plataformas.	Las aplicaciones web aún no pueden acceder a todas las funciones que tienen los dispositivos.
Tener versiones para múltiples plataformas requiere mantener múltiples códigos fuente y puede resultar en mayores costos de desarrollo,	Brindar soporte a múltiples navegadores web para móviles puede resultar en mayores costos en el desarrollo y mantenimiento, etc.

mantenimiento, publicación de actualizaciones, etc.	
Los usuarios pueden tener diferentes versiones lo que puede hacer que sea más difícil proporcionarles apoyo.	Los usuarios pueden utilizar diferentes navegadores móviles, lo que puede hacer que sea más difícil dar mantenimiento a la aplicación y proporcionar apoyo a los usuarios.
Los procesos de aprobación de las tiendas de aplicaciones pueden retrasar el lanzamiento de la aplicación o evitar su publicación	Para los usuarios, puede resultar más difícil encontrar una aplicación web móvil, debido a la falta de una tienda de aplicaciones centralizada

Tabla 07. Diferencia entre aplicaciones nativas y web (debilidades)

3.2.6. DEFINICION DE TERMINOS

- ✓ Common Type System (CTS): Tipo Común es el conjunto de reglas que han de seguir las definiciones de tipos de datos para que el CLR las acepte. Es decir, aunque cada lenguaje gestionado disponga de sus propias sintaxis para definir tipos de datos, en el MSIL resultante de la compilación de sus códigos fuente se ha de cumplir las reglas del CTS.
- ✓ Framework: Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve

como referencia, para enfrentar y resolver nuevos problemas de índole similar.

- ✓ IDE: Entorno de desarrollo integrado s un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).
- ✓ PRODUCT OWNER: El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.
- ✓ SCRUM: Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

3.3. Selección de la Metodología a utilizar para el desarrollo de la investigación

3.3.1. Programación Extrema (XP)

Extreme Programming (XP) surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Las metodologías de desarrollo de software tradicionales (ciclo de vida en cascada, evolutivo, en espiral, iterativo, etc.) aparecen, comparados con los nuevos métodos propuestos en XP, como pesados y poco eficientes. La crítica más frecuente a estas metodologías “clásicas” es que son demasiado burocráticas. Hay tanto que hacer para seguir la metodología que, a veces, el ritmo entero del desarrollo se retarda. Como respuesta a esto, se ha visto en

los últimos tiempos el surgimiento de “Metodologías Ágiles”.

[LIB01]

Estos nuevos métodos buscan un punto medio entre la ausencia de procesos y el abuso de los mismos, proponiendo un proceso cuyo esfuerzo valga la pena.

Los métodos ágiles cambian significativamente algunos de los énfasis de las metodologías “clásicas”:

- Los métodos ágiles son adaptables en lugar de predictivos. Los métodos “clásicos” tienden a intentar planear una gran parte del proceso del software en gran detalle para un plazo largo de tiempo. Esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio.
- Los métodos ágiles son orientados a la gente y no orientados al proceso. El objetivo de los métodos “clásicos” es definir un proceso que funcionará bien independientemente de quien lo utilice. Los métodos ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo.

XP es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo.

La metodología también enfatiza el trabajo en equipo. Tanto gerentes como clientes y desarrolladores son partes del mismo equipo dedicado a entregar software de calidad.

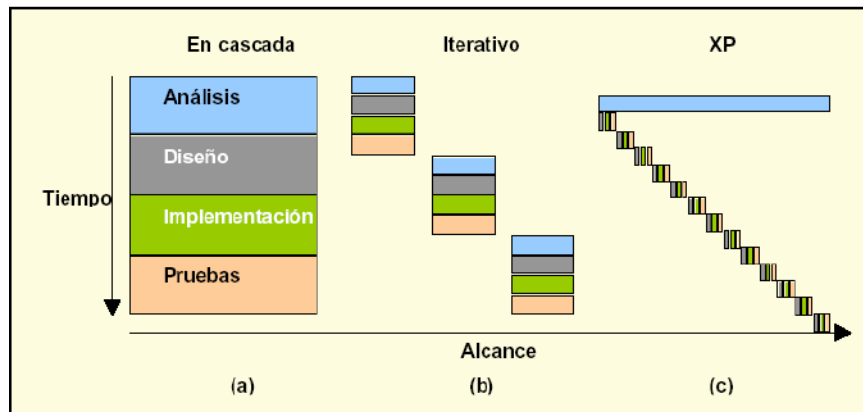
La metodología XP define cuatro variables para cualquier proyecto de software: **costo, tiempo, calidad y alcance**.

Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes y jefes de proyecto). El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres. Este mecanismo indica que, por ejemplo, si el cliente establece el alcance y la calidad, y el jefe de proyecto el precio, el grupo de desarrollo tendrá libertad para determinar el tiempo que durará el proyecto.

El ciclo de vida de un proyecto XP incluye, al igual que las otras metodologías, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto.

Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan a XP (y que serán detalladas más adelante).

Típicamente un proyecto con uml lleva 10 a 15 ciclos o iteraciones tradicionales (por ejemplo, incremental o espiral), comparados con el de XP.



Visión general del modelo

Si bien el ciclo de vida de un proyecto XP es muy dinámico, se puede separar en fases:

Fase de exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración.

Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado.

Fase de planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o “Release Plan”.

Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

Fase de puesta en producción

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”).

3.3.2. SCRUM

Scrum es una metodología ágil de desarrollo de proyectos, es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums. **[LIB02]**

Scrum es una metodología ágil, y como tal:

- ✓ Es un modo de desarrollo de carácter adaptable más que predictivo.
- ✓ Orientado a las personas más que a los procesos.
- ✓ Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Características de Scrum

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el Product Owner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos autoorganizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

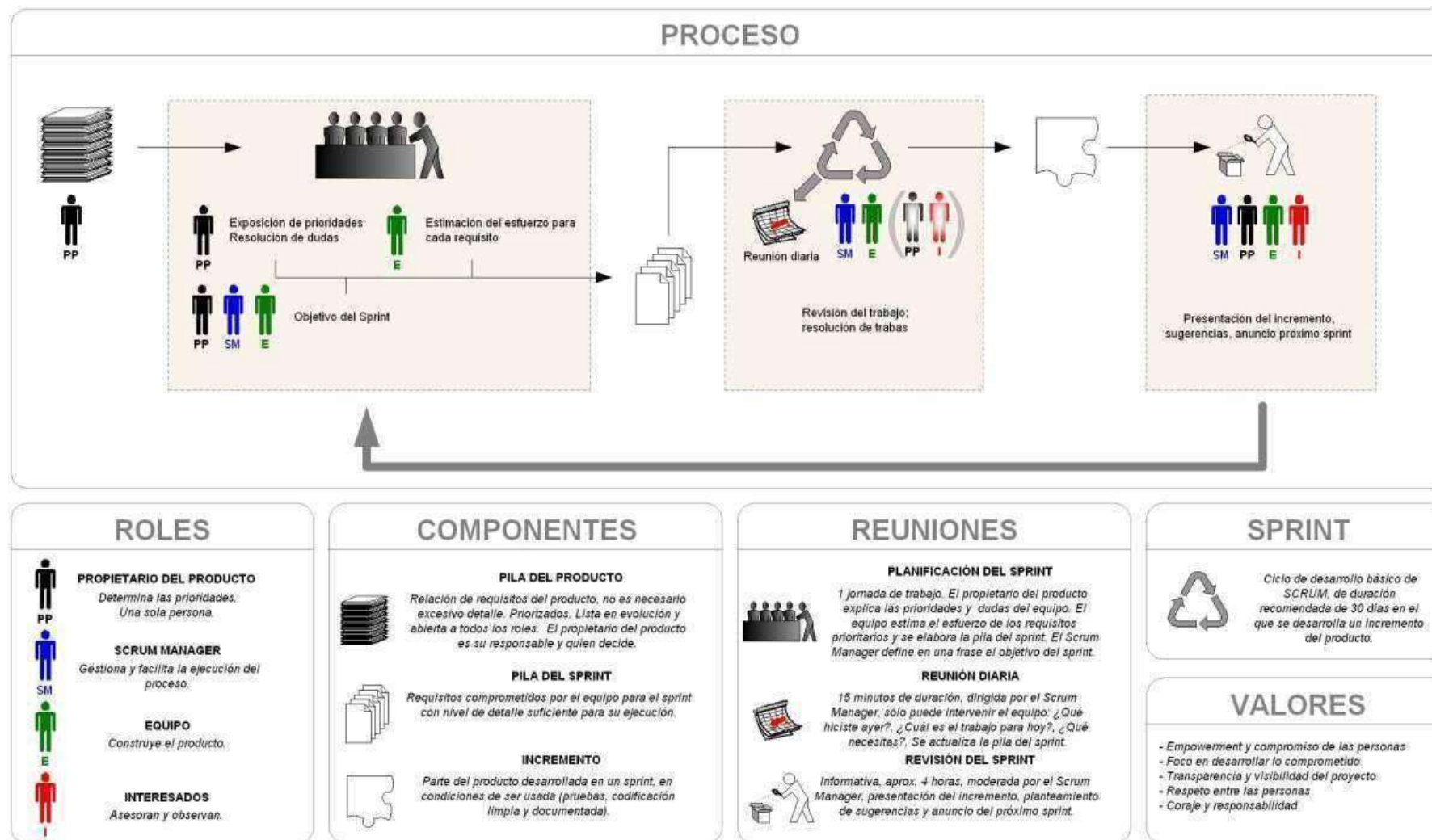
Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

Scrum aplicado al desarrollo de software

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). Más tarde, en 2001 serían dos de los promulgadores del Manifiesto ágil. En el desarrollo de software scrum está considerado como modelo ágil por la Agile Alliance.



Visión general del modelo

La ilustración anterior (Ilustración N° 2) incluye una descripción sinóptica del proceso y sus elementos que son:

- Roles: Propietario del producto, Gestor o Manager del Scrum, Equipo e Interesados.
- Componentes del proceso: Pila del producto (Product Backlog), Pila del sprint (Sprint Backlog), Incremento.
- Reuniones: Planificación del sprint, Revisión diaria, Revisión del sprint.

CAPITULO IV

DESARROLLO DE LA PROPUESTA

4. DESARROLLO DE LA PROPUESTA

4.1. Estándares para la Implementación del Análisis

- Estándares de historias de usuario
 - ✓ Se establece la fecha de inicio y fecha de término para la iteración.
 - ✓ Los nombres de las historias de usuario se escriben todo con mayúsculas.
 - ✓ Los nombres de las historias de usuario deben ser claros y entendibles.
 - ✓ A cada historia de usuario se le elabora su respectivo prototipo (especie de modelo de cómo se verá la historia de usuario).
 - ✓ Se describirá cada historia de usuario usando un determinado formato que contiene los siguientes puntos:
 - Número: Indica el número de historia de usuario.
 - Usuario: Persona que solicita la historia de usuario
 - Nombre: Nombre de la historia de usuario.
 - Prioridad en negocio: Importancia de la historia de usuario, categorizada en: Alta, media y baja.
 - Puntos estimados: Estimación en número de la historia de usuario, se utiliza el siguiente intervalo: 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 7/7.
 - Iteración asignada: Número de la iteración que pertenece dicha historia de usuario.
 - Programador responsable: Persona encargada del desarrollo de la historia de usuario.
 - Descripción: Indicar la funcionalidad de la historia de usuario que va hacer desarrollada por el programador.
 - Observaciones: Comentario adicional acerca de la historia de usuario.
 - Tareas: Relación de las actividades a llevar a cabo para el desarrollo de la historia de usuario.

- Estándares de tareas de historias de usuario
 - ✓ Los nombres de las tareas se escriben en mayúscula la primera letra de cada palabra y las demás en minúsculas.
 - ✓ Los nombres de las tareas deben ser claros y entendibles.
 - ✓ A cada tarea se le elabora su respectivo prototipo (especie de modelo de cómo se verá la tarea).
 - ✓ Se describirá cada tarea usando un determinado formato que contiene los siguientes puntos:
 - Número: Indica el número de tarea.
 - Número de Historia: Indica el número de Historia a la que pertenece.
 - Nombre tarea: Nombre de tarea.
 - Tipo de tarea: Identificar la tarea, categorizada en: Desarrollo/Corrección/Mejora/Otra (especificar).
 - Puntos estimados: Los puntos estimados se pondrán dependiendo de lo que se crea conveniente para realizar esta tarea.
 - Fecha Inicio y Fin: Fecha de inicio y fin de la tarea.
 - Programador responsable: Persona encargada del desarrollo de la tarea.
 - Descripción: Indicar la funcionalidad de la tarea que va hacer desarrollada por el programador.

4.2. Estándares para la Implementación del Diseño

- Estándares de base de datos

Para el diseño de la base de datos contamos con nueve archivos: Crear Tablas en donde creamos las tablas con sus respectivas relaciones, Llenado donde insertamos algunos valores que ya están definidos y siete Paquetes el que contendrá los procedimientos almacenados.

- En el dispositivo móvil los registros se eliminarán de la Base de Datos (SQLite) luego de cerrar el día, debido a que dichos datos se guardan en la base de datos de SQL Server que se encuentra en la nube.
- En la aplicación de escritorio y touchscreen los registros no se eliminarán de la Base de Datos, sino se cambiará el estado de marca de borrado para poder llevar un historial.

✓ Tablas:

Los nombres de las tablas deben reflejar de manera precisa su contenido y función utilizando la notación Camel Case. No se deben utilizar espacios en el nombre. Según el módulo de que se encuentre se antepondrá las 2 iniciales del modulo

Módulos:

- ✓ RE = Restaurante.
- ✓ AL = Almacén
- ✓ MG = Modulo General
- ✓ VE = Ventas

Ejemplos:

- ✓ REMESA: Tabla de Mesas.
- ✓ ALART: Tabla de Artículos.
- ✓ VECOMPROBANTEC: Tabla Cabecera Comprobantes de Venta.

✓ Columnas:

Cada nombre de campo debe ser único dentro de su tabla correspondiente.

No se deben utilizar palabras reservadas como nombres de campos.

Los nombres de las columnas de una tabla deben estar expresados en singular.

Los nombres deben ser simples, representativos e intuitivos.
Ejemplo: NroMesa, CodReserva.

✓ Procedimientos Almacenados y Funciones:

El nombre de los procedimientos almacenados y funciones empiezan con el prefijo “PA” y “FN”, seguido de un subguión “_”.

El nombre del procedimiento almacenado/Función está formado por la primera letra en minúscula.

No se utilizarán palabras reservadas como nombres de campos.

Tipo	Prefijo	Ejemplo
Procedimiento Almacenado	PA	PA_ListarClientes
Función	FN	FN_ObtenerPrecioArticulo

✓ Restricciones:

Para los nombres de las claves primarias la estructura es la siguiente:

```
ALTER TABLE Nombre de la tabla
ADD PRIMARY KEY(nombre de columna)
Ejemplo: ALTER TABLE REMesa
ADD PRIMARY KEY(codMesa).
```

Para nombrar las claves foráneas la estructura es la siguiente:

```
ALTER TABLE nombre de la tabla alterada
ADD ( FOREIGN KEY (nombre de columna) REFERENCES
nombre de la tabla referente(nombre de columna)).
Ejemplo: ALTER TABLE MGDistrito
ADD ( FOREIGN KEY (codigo_Pro) REFERENCES Provincia
referente(codigo_Pro)).
```


4.3. Estándares para la Implementación de la Construcción

- En la creación del proyecto en Visual Studio los componentes deben estar organizados en carpetas la primera letra de cada palabra en mayúsculas y el resto en minúsculas: Datos, Lógica, Presentación.
- Las clases que contenga cada capa, deberán llevar un nombre claro y preciso para su rápida identificación: Reserva, Pedido, Cliente, Mozo
- En interfaces, los títulos indicarán la acción que se realiza, todas estarán en mayúscula y los botones empezarán en mayúscula la primera letra y el resto en minúsculas.
- Todos los mensajes que se muestren en las interfaces deben empezar la primera letra en mayúscula y las demás en minúsculas.

4.4. Pila de Productos:

N°	NOMBRE	COMO PROBARLO	PUNTOS
1	LOGUEAR A UN USUARIO	Al dar clic en el ícono de la aplicación e ingresar nuestros datos correctamente, cargará el menú principal de la aplicación.	4
2	REGISTRAR USUARIO	Abrir la aplicación, ir a la opción de menú Administración/Usuarios, luego dar clic en nuevo e ingresar los datos necesarios para registrar	4

		un usuario y guardarlo en la base de datos.	
3	MODIFICAR USUARIO	Abrir la aplicación, ir a la opción de menú Administración/Usuario, luego dar clic en Modificar e ingresar los datos necesarios para modificar un usuario y guardarlo en la base de datos.	4
4	ELIMINAR USUARIO A	Abrir la aplicación, ir a la opción de menú Administración/Usuario, luego dar clic en Eliminar para eliminar el usuario y darle de baja.	3
5	ASIGNAR DE ACCESOS / ROLES	Abrir la aplicación, ir a la opción de menú Administración/Usuario, seleccionar el usuario y darles accesos a las opciones de menú o asignarle un rol.	4
6	REGISTRAR ROL	Abrir la aplicación, ir a la opción de menú Administración/Roles, luego dar clic en nuevo e ingresar los datos	3

		necesarios para registrar un rol y guardarlo en la base de datos.	
7	MODIFICAR ROL	Abrir la aplicación, ir a la opción de menú Administración/Roles, luego dar clic en Modificar e ingresar los datos necesarios para modificar un rol y guardarlo en la base de datos.	3
8	ELIMINAR ROL	Abrir la aplicación, ir a la opción de menú Administración/Roles, luego dar clic en Eliminar para eliminar el rol y darle de baja.	3
9	REGISTRAR AMBIENTE	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Ambientes, luego dar clic en nuevo e ingresar los datos necesarios para registrar un ambiente y guardarlo en la base de datos.	3
10	MODIFICAR AMBIENTE	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Ambientes, luego dar clic	3

		en Modificar e ingresar los datos necesarios para modificar un ambiente y guardarlo en la base de datos.	
11	ELIMINAR AMBIENTE	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Ambientes, luego dar clic en Eliminar para eliminar el ambiente y darle de baja.	3
12	REGISTRAR MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mesas, luego dar clic en nuevo e ingresar los datos necesarios para registrar una mesa y guardarlo en la base de datos.	4
13	MODIFICAR MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mesas, luego dar clic en Modificar e ingresar los datos necesarios para modificar una mesa y guardarlo en la base de datos.	4

14	ELIMINAR MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mesas, luego dar clic en Eliminar para eliminar la mesa y darle de baja.	4
15	REGISTRAR DIRECTORIO	Abrir la aplicación, ir a la opción de menú Administración/Directorio, luego dar clic en nuevo e ingresar los datos necesarios para registrar un directorio y guardarlo en la base de datos.	9
16	MODIFICAR DIRECTORIO	Abrir la aplicación, ir a la opción de menú Administración/Directorio, luego dar clic en Modificar e ingresar los datos necesarios para modificar un directorio y guardarlo en la base de datos.	8
17	ELIMINAR DIRECTORIO	Abrir la aplicación, ir a la opción de menú Administración/Directorio luego dar clic en Eliminar para eliminar el directorio y darle de baja.	7

18	REGISTRAR MOZO	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mozos, luego dar clic en nuevo e ingresar los datos necesarios para registrar un mozo y guardarlo en la base de datos.	4
19	MODIFICAR MOZO	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mozos, luego dar clic en Modificar e ingresar los datos necesarios para modificar un mozo y guardarlo en la base de datos.	4
20	ELIMINAR MOZO	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Mozos, luego dar clic en Eliminar para eliminar el mozo y darle de baja.	3
21	REGISTRAR CLASE – SUBCLASE ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Clases-Subclases Artículos, luego dar clic en nuevo e ingresar los datos necesarios para registrar una Clase-	7

		SubClase de Artículo y guardarlo en la base de datos.	
22	MODIFICAR CLASE – SUBCLASE ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Clases-Subclases Artículos, luego dar clic en Modificar e ingresar los datos necesarios para modificar una Clase-SubClase y guardarlo en la base de datos.	7
23	ELIMINAR CLASE – SUBCLASE ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Clases-Subclases Artículos, luego dar clic en Eliminar para eliminar la Clase-SubClase y darle de baja.	5
24	REGISTRAR ALMACEN	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Almacén, luego dar clic en nuevo e ingresar los datos necesarios para registrar un almacén y guardarlo en la base de datos.	4

25	MODIFICAR ALMACEN	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Almacén, luego dar clic en Modificar e ingresar los datos necesarios para modificar un almacén y guardarlo en la base de datos.	4
26	ELIMINAR ALMACEN	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Almacén, luego dar clic en Eliminar para eliminar el almacén y darle de baja.	4
27	REGISTRAR ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Artículos, luego dar clic en nuevo e ingresar los datos necesarios para registrar un Artículo y guardarlo en la base de datos.	6
28	MODIFICAR ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Artículos, luego dar clic en Modificar e ingresar los datos necesarios para modificar un artículo y guardarlo en la base de datos.	5

29	ELIMINAR ARTICULO	Abrir la aplicación, ir a la opción de menú Almacén/Gestión Artículos, luego dar clic en Eliminar para eliminar el artículo y darle de baja.	5
30	REGISTRAR CARTA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Carta, luego dar clic en nuevo e ingresar los datos necesarios para registrar una carta y guardarlo en la base de datos.	5
31	MODIFICAR CARTA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Carta, luego dar clic en Modificar e ingresar los datos necesarios para modificar una carta y guardarlo en la base de datos.	4
32	ELIMINAR CARTA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión de Carta, luego dar clic en Eliminar para eliminar la carta y darle de baja.	3

33	REGISTRAR RESERVACION MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión Reservas de Mesa, luego dar clic en nuevo e ingresar los datos necesarios para registrar una reservación y guardarlo en la base de datos.	4
34	MODIFICAR RESERVACION MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión Reservas de Mesa, luego dar clic en Modificar e ingresar los datos necesarios para modificar una reservación y guardarlo en la base de datos.	4
35	ELIMINAR RESERVACION MESA	Abrir la aplicación, ir a la opción de menú Restaurante/Gestión Reservas de Mesa, luego dar clic en Eliminar para eliminar la reservación y darle de baja.	3
36	REGISTRAR PEDIDO	Abrir la aplicación, ir a la opción de menú Ventas/Pedido, luego dar	10

		clic en nuevo e ingresar los datos necesarios para registrar un pedido y guardarlo en la base de datos.	
37	MODIFICAR PEDIDO	Abrir la aplicación, ir a la opción de menú Ventas/Pedido, luego dar clic en Modificar e ingresar los datos necesarios para modificar un pedido y guardarlo en la base de datos.	10
38	ELIMINAR PEDIDO	Abrir la aplicación, ir a la opción de menú Ventas/Pedido luego dar clic en Eliminar para eliminar un pedido y darle de baja.	6
39	ANULAR PEDIDO	Abrir la aplicación, ir a la opción de menú Ventas/Pedido, luego dar clic en Anular para anular un pedido.	6
40	REGISTRAR COMPROBANTE VENTA	Abrir la aplicación, ir a la opción de menú Ventas/Comprobante Venta, luego dar clic en nuevo e ingresar los datos necesarios para registrar	10

		un comprobante de venta y guardarlo en la base de datos.	
41	MODIFICAR COMPROBANTE VENTA	Abrir la aplicación, ir a la opción de menú Ventas/Comprobante de Venta, luego dar clic en Modificar e ingresar los datos necesarios para modificar un comprobante de venta y guardarlo en la base de datos.	10
42	ELIMINAR COMPROBANTE VENTA	Abrir la aplicación, ir a la opción de menú Ventas/Comprobante de Venta luego dar clic en Eliminar para eliminar un Comprobante de Venta y darle de baja.	9
43	ANULAR COMPROBANTE VENTA	Abrir la aplicación, ir a la opción de menú Ventas/Comprobante de venta, luego dar clic en Anular para anular un comprobante de venta.	9
44	CONSULTAR SEGUIMIENTO DE PEDIDOS	Abrir la aplicación, ir a la opción de menú Restaurante/Seguimiento de Pedidos, luego dar clic	8

		en modificar para modificar el estado del pedido.	
45	REGISTRAR CLIENTES A	Abrir la aplicación y registrar cliente con los datos indicados.	4
46	MODIFICAR CLIENTES.	Abrir la aplicación, ir a la opción de menú indicada y seleccionar al cliente que se desea modificar sus datos.	4
47	ELIMINAR CLIENTES	Abrir la aplicación, ir a la opción de menú indicada y seleccionar al cliente que se desea eliminar sus datos.	4
48	DESPACHAR COCINA	Abrir la aplicación ir a la opción de menú respectiva y proceder con los cambios según se requiera.	10
49	CONFIGURAR CONEXIÓN	En la ventana “Configurar Conexión”, al ingresar los datos necesarios y presionar el botón “Guardar” se mostrará un mensaje de éxito si no hubo ningún problema.	2
50	AUTENTICAR USUARIO	Habiendo indicado el nombre de usuario y contraseña, presionar el	4

		botón “Conectar”. A continuación, se verificará los datos del cliente y de haber éxito se mostrará una lista de empresas disponibles.	
51	SELECCIONAR EMPRESA	En la pantalla de inicio de sesión, seleccionar una empresa y presionar el botón, si todo es correcto se mostrará la pantalla “Iniciar Periodo”	2
52	REGISTRAR CREDENCIALES DE USUARIO	Cerrar la aplicación y volver a iniciarla para comprobar que no se vuelve a solicitar las credenciales de usuario	2
53	REGISTRAR INICIO DE PERIODO EN EL SERVIDOR	Consultar la tabla “RECONTROLENIOMOVIL”, filtrando los resultados por el usuario del mozo y comprobar que el estado del registro es “020”	3
54	REGISTRAR INICIO DE PERIODO EN EL DISPOSITIVO	En la opción “Inicio de Periodo”, verificar que el botón “Iniciar Periodo” aparecerá deshabilitado.	3
55	REGISTRO LOCAL DE ARTÍCULOS	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2

56	REGISTRO LOCAL DE CARTA	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2
57	REGISTRO LOCAL DE CATEGORÍAS	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2
58	REGISTRO LOCAL DE CLIENTES	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2
59	REGISTRO LOCAL DE MESAS	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2
60	REGISTRO LOCAL DE INFORMACIÓN DE MESAS	En la opción “Sincronizar”, verificar que el botón “Sincronizar” aparecerá deshabilitado.	2
61	LECTURA DE ESTADO ACTUAL DE LAS MESAS	Abrir la aplicación, ir a la opción “Realizar Pedido” y de no haber problema se mostrará el listado completo de mesas. En caso contrario se mostrará un mensaje de error y las mesas no podrán ser seleccionadas.	3

62	ACTUALIZACIÓN DE ESTADO ACTUAL DE LAS MESAS EN EL DISPOSITIVO	Abrir la aplicación, ir a la opción “Realizar Pedido” y de no haber problema se mostrará el listado completo de mesas. En caso contrario se mostrará un mensaje de error y las mesas no podrán ser seleccionadas.	2
63	LISTADO ACTUALIZADO DE MESAS	Abrir la aplicación, ir a la opción “Realizar Pedido” y de no haber problema se mostrará el listado completo de mesas. En caso contrario se mostrará un mensaje de error y las mesas no podrán ser seleccionadas.	3
64	SELECCIÓN DE MESA Y ACTUALIZACIÓN DE SU ESTADO EN EL DISPOSITIVO Y EN EL SERVIDOR	En la opción “Realizar Pedido”, seleccionar una mesa. Aceptar el mensaje de confirmación. De no haber problema se mostrará la pantalla de registro de pedido.	6
65	BÚSQUEDA DE RESERVAS Y LISTADO ACTUAL DE MESAS RESERVADAS	Abrir la aplicación, ir a la opción “Buscar Reservas”, ingresar los criterios de búsqueda y verificar la	7

		relación de mesas resultantes.	
66	LISTADO DE CATEGORÍAS	En la pantalla de registro de pedido, verificar la relación de categorías.	4
67	LISTADO DE ARTÍCULOS DE LA CATEGORÍA SELECCIONADA	En la pantalla de registro de pedido, seleccionar una categoría para listar los artículos (platos) disponibles en esa categoría.	4
68	MANTENIMIENTO DE PEDIDO EN EL DISPOSITIVO	<p>En la pantalla de registro de pedido:</p> <ul style="list-style-type: none"> • Seleccionar un artículo para agregarlo al pedido. • Mantener presionado un ítem del pedido para mostrar las opciones disponibles: aumenta, disminuir la cantidad del ítem, o quitarlo del pedido. 	10
69	ENVÍO DEL PEDIDO AL SERVIDOR PARA SU PROCESAMIENTO	En la pantalla de registro de pedido, seleccionar la acción "Enviar" de la barra de acción. Si el envío fue correcto se mostrará un mensaje y el usuario	10

		<p>volverá a la pantalla anterior.</p> <p>En el aplicativo de escritorio ingresar a la opción “Restaurante/Seguimiento de Pedidos”.</p>	
70	ACTUALIZACIÓN DE PEDIDOS A RECOGER	<p>Abrir la aplicación, ir a la opción “Pedidos a recoger”, de no haber problemas se mostrará la relación de pedidos a recoger.</p>	10
71	SELECCIÓN DE PEDIDO RECOGIDO	<p>Abrir la aplicación, ir a la opción “Pedidos a recoger” y seleccionar un ítem de la lista ubicada en el centro de la pantalla.</p>	2
72	SELECCIÓN DE LOS ÍTEMS ESPECÍFICOS RECOGIDOS	<p>Abrir la aplicación, ir a la opción “Pedidos a recoger” y habiendo previamente seleccionado un pedido, seleccionar los ítems a establecer como recogidos.</p>	10
73	ACTUALIZACIÓN DE ESTADO DE PEDIDO E ÍTEMS RECOGIDOS TANTO EN EL SERVIDOR COMO	<p>Abrir la aplicación, ir a la opción “Pedidos a recoger” y habiendo previamente elegido los ítems de un determinado pedido, seleccionar la acción</p>	10

	EN DISPOSITIVO.	EL	<p>“Enviar” de la barra de acción.</p> <p>Verificar que los ítems/producto enviados desaparecerán de las respectivas listas.</p> <p>En el aplicativo de escritorio, ingresar a la opción “Restaurante/Seguimiento de Pedidos”.</p>	
74	LISTADO PEDIDOS FACTURAR	DE A	<p>Abrir la aplicación, ir a la opción “Pedidos a facturar” de no haber problemas se mostrará la relación de pedidos listos para facturar.</p>	6
75	SELECCIÓN PEDIDO FACTURAR	DE A	<p>Abrir la aplicación, ir a la opción “Pedidos a facturar” y seleccionar un ítem de la lista ubicada en el centro de la pantalla. En el panel del extremo derecho se mostrará un el monto total de pedido seleccionado.</p>	2
76	BÚSQUEDA CLIENTE	DE	<p>Abrir la aplicación, ir a la opción “Pedidos a facturar”.</p> <p>En el panel del extremo derecho seleccionar el tipo de venta, “Factura”. En la caja de texto “RUC”</p>	2

		<p>ingresar el ruc del cliente. Si el cliente existe se mostrará la razón social/nombre del cliente en el campo “Cliente”.</p>	
77	ENVÍO DE DATOS DEL PEDIDO FACTURADO	<p>Abrir la aplicación, ir a la opción “Pedidos a facturar”. Habiendo completado los datos del panel del extremo derecho, presionar el botón “Aceptar”.</p> <p>En el aplicativo de escritorio, ingresar a la opción “Restaurante/Seguimiento de Pedidos”.</p>	10
78	ACTUALIZACIÓN DEL ESTADO DEL PEDIDO EN EL DISPOSITIVO	<p>Abrir la aplicación, ir a la opción “Pedidos a facturar”. De haber sido exitoso el envío del pedido a facturar, los campos del panel derecho se restablecerán a sus valores por defecto y el pedido enviado desaparecerá de la lista de pedidos.</p>	3
79	REGISTRO DE CIERRE DE	<p>Abrir la aplicación, ir a la opción “Cerrar Día” y confirmar la acción. Si el</p>	3

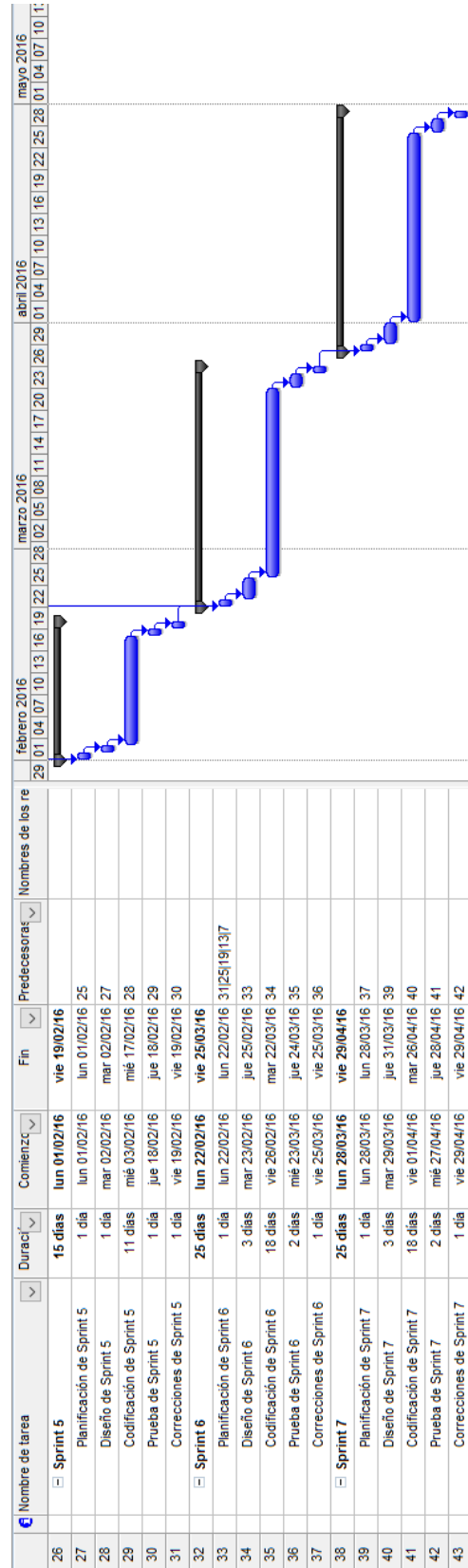
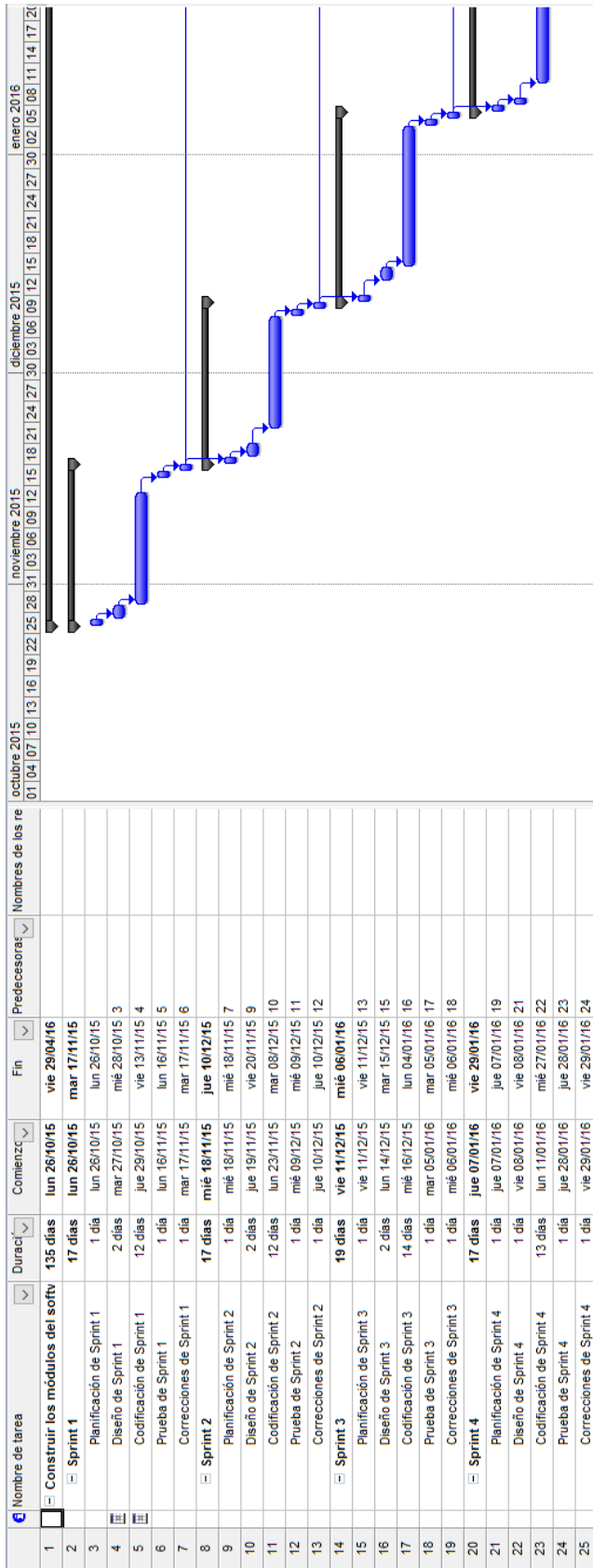
	PERIODO EN EL DISPOSITIVO	proceso tiene existo, se mostrará un mensaje de confirmando la acción y el botón será deshabilitado.	
80	REGISTRO DE CIERRE DE PERIODO EN EL SERVIDOR	Consultar la tabla "RECONTROLENIOMOVIL", filtrando los resultados por el usuario del mozo y comprobar que el estado del registro es "030"	3
81	ELIMINACIÓN DE DATA LOCAL	Abrir la aplicación y comprobar que esta vuelve a solicitar las credenciales de usuario.	4

Tabla 08. Pila de productos

4.5. PLAN DE ENTREGA DE LOS SPRINTS

Cuando se han obtenido las historias de usuario, y además de determinar la necesidad en el orden de realización, se procedió a ubicar los tiempos de las entregas de las mismas, con el fin de estructurar el plan de iteraciones.

El tiempo que tomara desarrollar estas historias se definió en 120 días laborables, como se muestra en la figura:



45.1. SPRINT 1

Consta de 12 historias de usuario:

- ✓ Loguear usuario
- ✓ Registrar usuario
- ✓ Modificar y eliminar usuario
- ✓ Asignar de accesos / roles
- ✓ Registrar rol
- ✓ Modificar y eliminar rol
- ✓ Registrar ambiente
- ✓ Modificar y eliminar ambiente
- ✓ Registrar mesa
- ✓ Modificar y eliminar mesa
- ✓ Registrar directorio.
- ✓ Modificar y eliminar directorio.

45.2 SPRINT 2

Consta de 12 historias de usuario:

- ✓ Registrar mozo
- ✓ Modificar y eliminar mozo
- ✓ Registrar clase – subclase artículos
- ✓ Modificar y eliminar clase – subclase artículos
- ✓ Registrar almacén
- ✓ Modificar y eliminar almacén.
- ✓ Registrar artículo.
- ✓ Modificar y eliminar artículo.
- ✓ Registrar carta
- ✓ Modificar y eliminar carta
- ✓ Registrar reservación mesa
- ✓ Modificar y eliminar reservación mesa

453. SPRINT 3

Consta de 8 historias de usuario:

- ✓ Registrar a clientes
- ✓ Modificar y Eliminar clientes.
- ✓ Registrar pedido
- ✓ Modificar pedido
- ✓ Eliminar pedido
- ✓ Anular pedido.
- ✓ Registrar comprobante venta.
- ✓ Modificar comprobante venta.
- ✓ Eliminar comprobante venta.

454. SPRINT 4

Consta de 3 historias de usuario:

- ✓ Anular comprobante venta.
- ✓ Consultar seguimiento de pedidos
- ✓ Despachar cocina

455. SPRINT 5

Consta de 5 historias de usuario:

- ✓ Configurar conexión
- ✓ Autenticar usuario
- ✓ Seleccionar empresa y registrar credenciales de usuario
- ✓ Registrar inicio de periodo en el servidor y en el dispositivo móvil
- ✓ Registro local de datos para toma de pedidos

456. SPRINT 6

Consta de 7 historias de usuario:

- ✓ Listado general actualizado de mesas
- ✓ Búsqueda de reservas y listado actual de mesas reservadas

- ✓ Selección de mesa y actualización de su estado en el dispositivo y en el servidor
- ✓ Listado de categorías
- ✓ Listado de artículos de la categoría seleccionada
- ✓ Mantenimiento de pedido en el dispositivo
- ✓ Envío del pedido al servidor para su procesamiento

457. SPRINT 7

Consta de 7 historias de usuario:

- ✓ Actualización de pedidos a recoger
- ✓ Selección de pedido y de los ítems específicos recogidos
- ✓ Actualización de estado de pedido e ítems recogidos tanto en el servidor como en el dispositivo.
- ✓ Listado de pedidos a facturar
- ✓ Selección de pedido a facturar y búsqueda de cliente
- ✓ Envío de datos del pedido facturado y actualización del estado del pedido en el dispositivo
- ✓ Registro de cierre de periodo y eliminación de data local

4.6. DEFINICIÓN DE SPRINTS

4.6.1. SPRINT 1

4.6.1.1. CALCULO DE LA VELOCIDAD ESTIMADA PARA EL PRIMER SPRINT

- ✓ Calculando duración del sprint:
1 semana y 4 días= 9 días laborables.
- ✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	8
Programador 2	9
Total	17 días hombres disponible

Tabla 09. Calculo Días Hombre Primer Sprint

- ✓ Calculando factor de dedicación (FD):

FD = Puntos de historia (último sprint) / días hombres disponible

$$FD = 73/17 = 4.29$$

- ✓ Velocidad estimada: FD X Días –Hombre- Disponible

$$17 * 4.29 = 72.93$$

4.6.12. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL PRIMER SPRINT:

Nº	SPRINT 1	PUNTO S	ESTADO
1	REGISTRAR USUARIO	4	Terminado
2	MODIFICAR Y ELIMINAR USUARIO	7	Terminado
3	LOGUEAR A UN USUARIO	4	Terminado
4	ASIGNAR DE ACCESOS / ROLES	4	Terminado
5	REGISTRAR ROL	3	Terminado
6	MODIFICAR Y ELIMINAR ROL	6	Terminado
7	REGISTRAR AMBIENTE	3	Terminado
8	MODIFICAR Y ELIMINAR AMBIENTE	3	Terminado
9	REGISTRAR MESA	4	Terminado

10	MODIFICAR Y ELIMINAR MESA	8	Terminado
11	REGISTRAR DIRECTORIO	9	Terminado
12	MODIFICAR Y ELIMINAR DIRECTORIO	15	Terminado
*	TOTAL DE PUNTOS DE HISTORIA	73	73
Objetivo	<ul style="list-style-type: none"> • Ingresar al sistema, así como también brindarle accesos por roles. • Registrar ambiente/Pisos, Mesas y directorio. 		

Tabla 10. Productos en el Primer Sprint

4.6.1.3. DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre de Historia: Registrar usuarios	
Prioridad en Negocio: media	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	

<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo usuario el cual más adelante podrá loguearse al mismo.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Administración /Usuarios. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del usuario. ✓ Dar clic en el botón Aceptar para confirmar la creación del usuario. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 11. 1° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador
Nombre de Historia: Modificar y eliminar usuario	

Prioridad en Negocio: media	
Puntos estimados: 7	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente a los usuarios, esta opción te permitirá modificar los datos del usuario que sean conveniente, así como también eliminar al usuario de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un usuario y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del usuario y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 12. 2° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador
Nombre de Historia: Registrar Rol	
Prioridad en Negocio: media	
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo rol el cual más adelante un usuario podrá ser asignado.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Administración / Rol. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del rol. ✓ Dar clic en el botón Aceptar para confirmar la creación del rol. 	
Tareas	Responsable
Hacer el diseño de la base de datos	José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Human Valqui

Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 13. 3° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Rol	
Prioridad en Negocio: media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente los roles, esta opción te permitirá modificar los datos del rol, así como también eliminar al rol de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un rol y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del rol y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui

Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 14. 4° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 5	Usuario: Todos
Nombre de Historia: Loguear a usuario	
Prioridad en Negocio: baja	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Una vez que haya sido creada la cuenta del usuario, este mismo podrá acceder al sistema, proceso también conocido como Logue, con su propio usuario y contraseña previamente creada.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Una vez ingresado el nombre de la empresa en donde se logueará, indicar el nombre de usuario y la contraseña respectiva. ✓ Presionar la tecla ENTER e inmediatamente de existir el usuario, se abrirá la ventana principal del sistema; en el caso que el usuario no exista, o haya digitado mal su nombre de usuario o contraseña aparecerá un mensaje indicando dicho error. 	
<p>Observaciones: Deben estar registrados previamente</p> <p style="text-align: center;">* Usuario * Empresa</p>	

* Almacén	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	José Human Valqui
Diseñar clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 15. 5° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador
Nombre de Historia: Asignar de accesos / roles	
Prioridad en Negocio: baja	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción te permitirá asignar a los usuarios a un rol, así como también accesos a las opciones de menú del sistema.	

Pruebas: <ul style="list-style-type: none"> ✓ Verificar que al dar clic en el botón Roles aparezca una ventana para la asignación del rol a un usuario seleccionado. ✓ Verificar que al dar clic en el botón Acceso aparezca una ventana para la asignación de las opciones del menú a un usuario seleccionado. ✓ Verificar que si el usuario está asignado a un rol este no puede ser acceso de forma independiente. 	
Observaciones: Deben estar registrados previamente <ul style="list-style-type: none"> • Usuario • Roles 	
Tareas	Responsable
Identificar y construir procedimientos almacenados necesarios para el proceso	Julio Alex Ventura Chero
Adicionar cambios necesarios a las clases correspondientes.	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	Jose Huaman Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 16. 6° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador
Nombre de Historia: Registrar Ambiente	
Prioridad en Negocio: media	
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo ambiente/piso el cual más adelante una mesa podrá ser asignada.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Restaurante / Gestión de Ambientes. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del ambiente. ✓ Dar clic en el botón Aceptar para confirmar la creación del ambiente. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui

Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 17. 7° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Ambiente	
Prioridad en Negocio: media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente los ambientes/pisos, esta opción te permitirá modificar los datos del ambiente/piso, así como también eliminar al ambiente de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un ambiente y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del ambiente y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero

Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 18. 8° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador
Nombre de Historia: Registrar Mesa	
Prioridad en Negocio: media	
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de una nueva mesa el cual más adelante se seleccionará cuando se realice un pedido.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Restaurante /Gestión de Mesas. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto de la mesa. ✓ Dar clic en el botón Aceptar para confirmar la creación de la mesa. 	

Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 19. 9° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Mesa	
Prioridad en Negocio: media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Habiendo registrado previamente las mesas, esta opción te permitirá modificar los datos de la mesa, así como también eliminar la mesa de ser necesario.	
Pruebas: ✓ Al seleccionar un rol y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos de la mesa	

y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar.	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 20.10° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 11	Usuario: Administrador/Caja
Nombre de Historia: Registrar Directorio	
Prioridad en Negocio: media	
Puntos estimados: 9	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción del sistema te permitirá la creación de persona Natural / Jurídica, direcciones, teléfonos el cual más adelante podrá utilizarse al asignar nuevo mozo, cliente, etc.	

Pruebas:	
<ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Administración / Directorio. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del directorio. ✓ Dar clic en el botón Aceptar para confirmar la creación del directorio. 	
Tareas	Responsable
Hacer el diseño de la base de datos	José Huamán Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Huamán Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Huamán Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

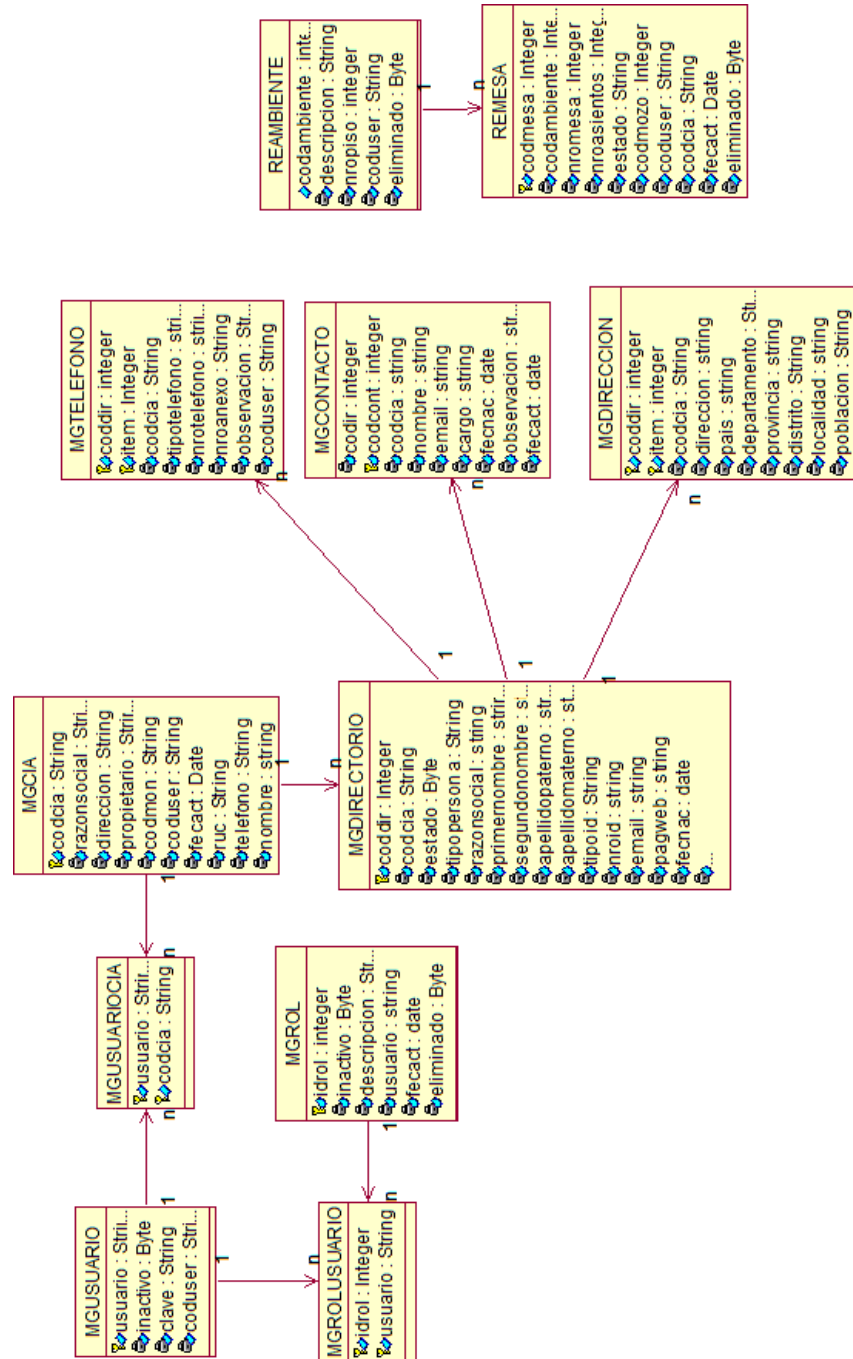
Tabla 21. 11° Historia de Usuario - Primer Sprint

HISTORIA DE USUARIO	
Número: 12	Usuario: Administrador/Caja
Nombre de Historia: Modificar y Eliminar Directorio	
Prioridad en Negocio: media	
Puntos estimados: 15	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	

<p>Descripción:</p> <p>Habiendo registrado previamente un directorio, esta opción te permitirá modificar los datos del directorio que sean conveniente, así como también eliminar de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un directorio y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del directorio y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	José Huamán Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Huamán Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 22. 12° Historia de Usuario - Primer Sprint

4.6.1.4. DIAGRAMA DE CLASES SPRINT 1



4.6.1.5. DIAGRAMA LOGICO Y FISICO SPRINT 1

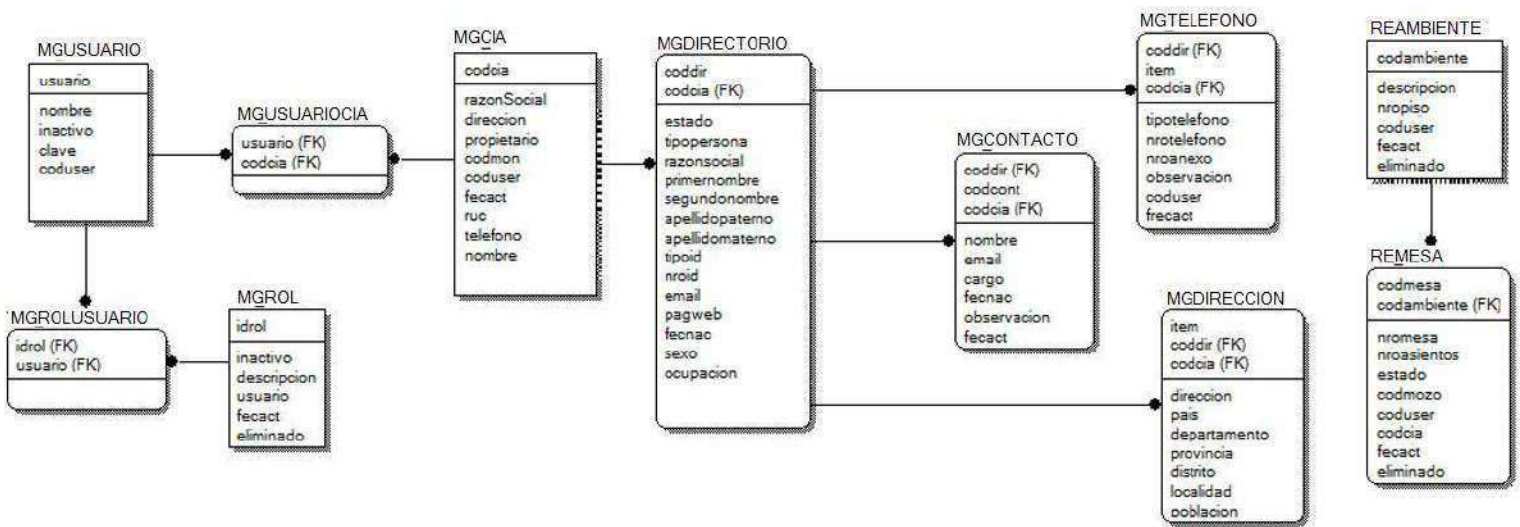


Ilustración 1 Diagrama lógico – Primer Sprint

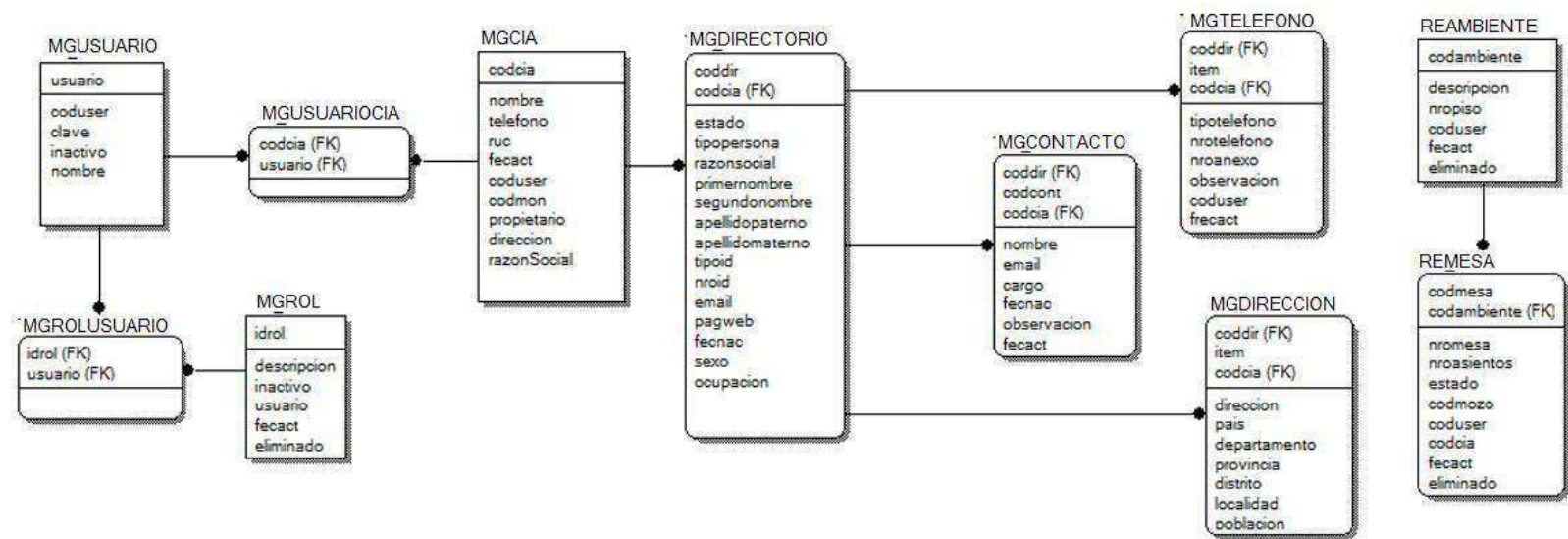


Ilustración 2 Diagrama físico – Primer Sprint

4.6.1.6. DIAGRAMA DE COMPONENTES SPRINT 1

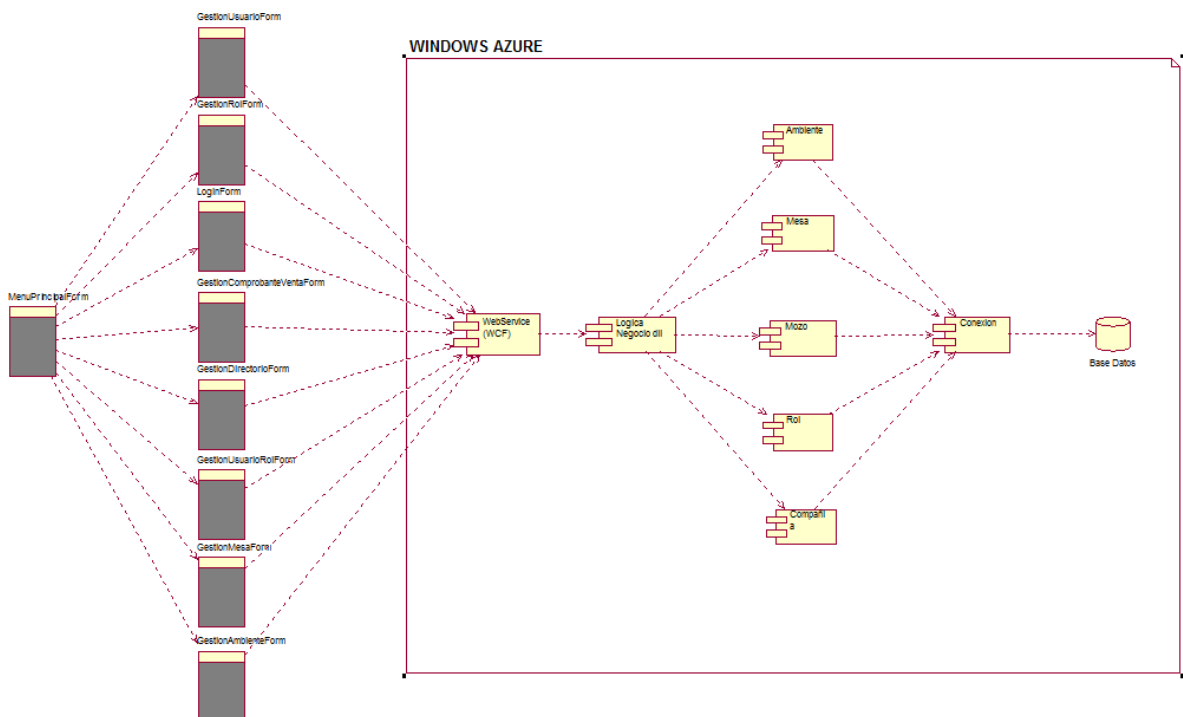
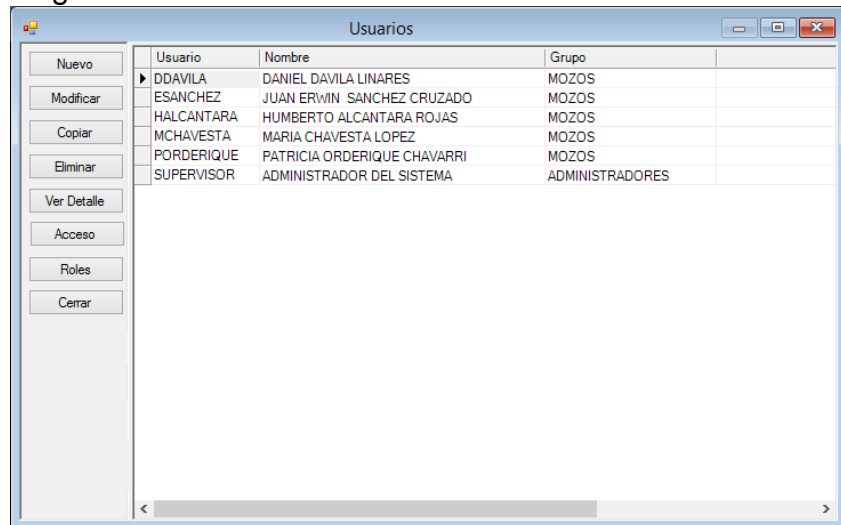


Ilustración 3 Diagrama de componentes – Primer Sprint

4.6.1.7. INTERFAZ DE LAS HISTORIAS DE USUARIO

✓ Registrar usuario:



The 'Nuevo Usuario' form includes the following fields and controls:

- Usuario:
- Nombre:
- Clave:
- ☐ Inactivo
- Aceptar button
- Cancelar button

Ilustración 4 Interfaz Registrar Usuario – Primer Sprint

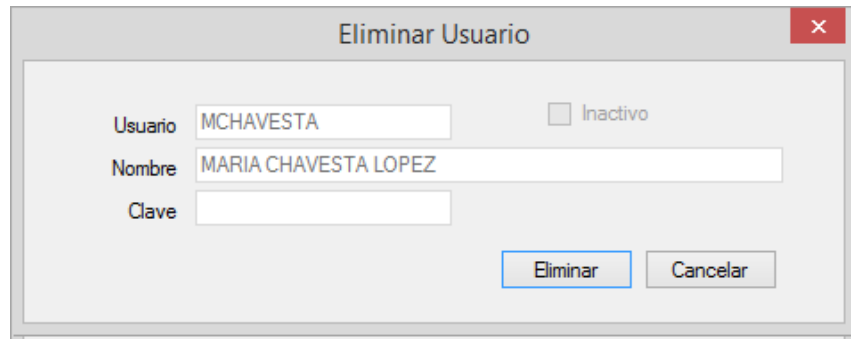
✓ Modificar usuario:

The 'Modificar Usuario' form includes the following fields and controls:

- Usuario:
- Nombre:
- Clave:
- ☐ Inactivo
- Aceptar button
- Cancelar button

Ilustración 5 Interfaz Modificar Usuario – Primer Sprint

✓ Eliminar usuario:



Eliminar Usuario

Usuario: MCHAVESTA ☐ Inactivo

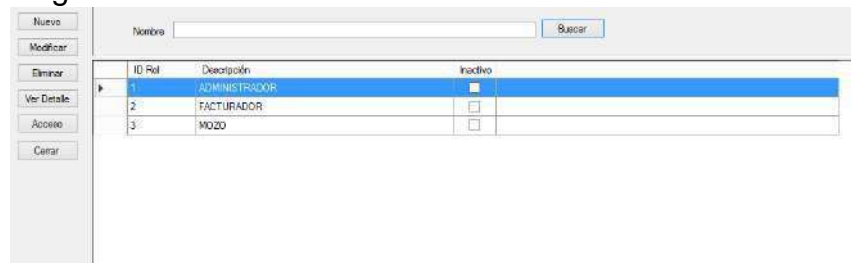
Nombre: MARIA CHAVESTA LOPEZ

Clave:

Eliminar Cancelar

Ilustración 6 Interfaz Eliminar Usuario – Primer Sprint

✓ Registrar Rol:



Nuevo

Modificar

Eliminar

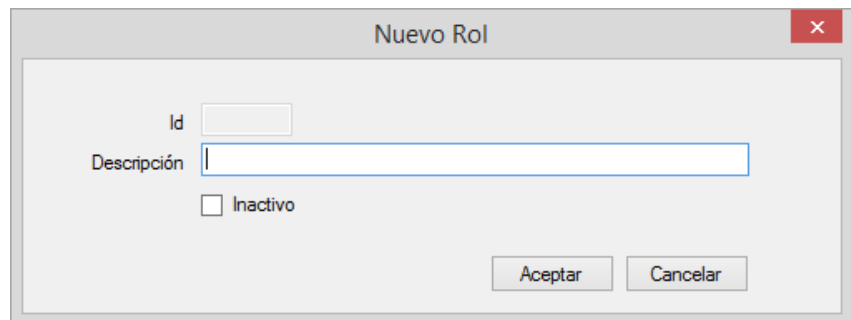
Ver Detalle

Aceptar

Cerrar

Nombre: Buscar

ID Rol	Descripción	Inactivo
1	ADMINISTRADOR	<input checked="" type="checkbox"/>
2	FACTURADOR	<input type="checkbox"/>
3	MOZO	<input type="checkbox"/>



Nuevo Rol

Id:

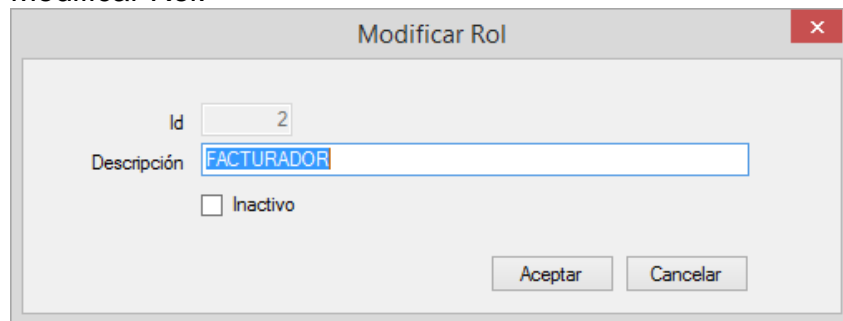
Descripción:

☐ Inactivo

Aceptar Cancelar

Ilustración 7 Interfaz Registrar Rol – Primer Sprint

✓ Modificar Rol:



Modificar Rol

Id 2

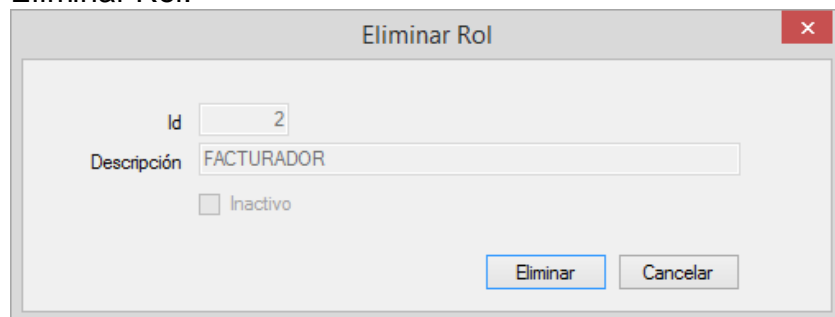
Descripción FACTURADOR

☐ Inactivo

Aceptar Cancelar

Ilustración 8 Interfaz Modificar Rol – Primer Sprint

✓ Eliminar Rol:



Eliminar Rol

Id 2

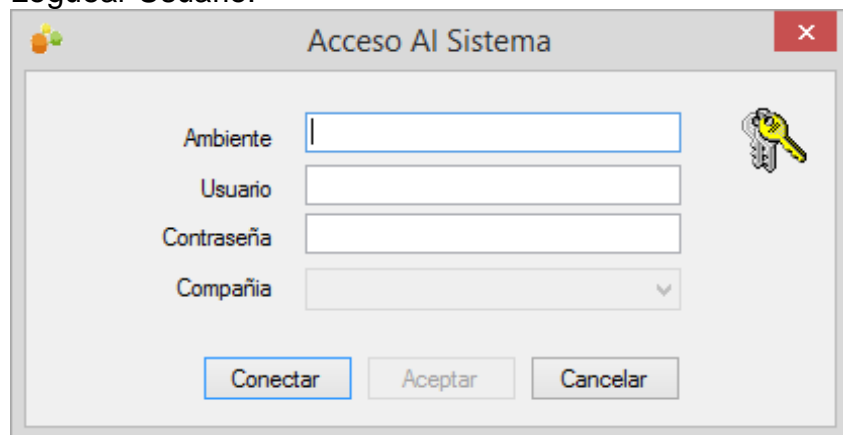
Descripción FACTURADOR

☐ Inactivo

Eliminar Cancelar

Ilustración 9 Interfaz Eliminar Rol – Primer Sprint

✓ Loguear Usuario:



Acceso Al Sistema

Ambiente

Usuario

Contraseña

Compañía

Conectar Aceptar Cancelar

Ilustración 10 Interfaz Loguear Usuario – Primer Sprint

✓ Asignar Roles / Accesos:

- Asignar Acceso

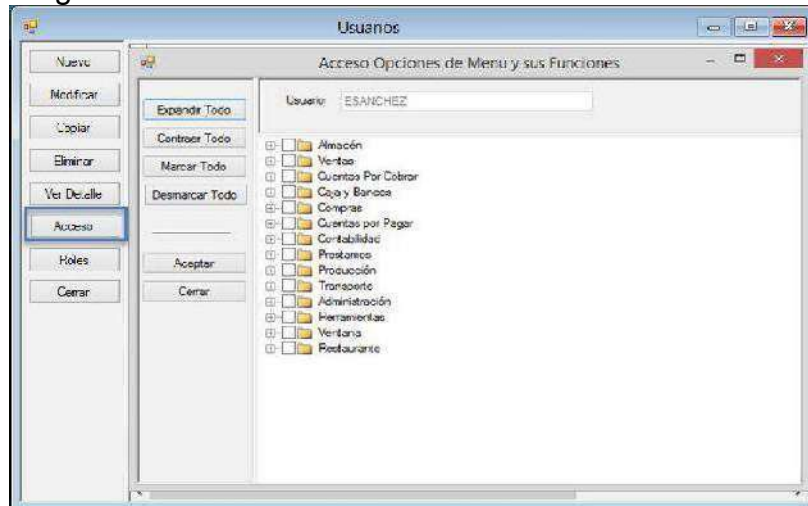


Ilustración 11 Interfaz Asignar Acceso – Primer Sprint

- Asignar Rol

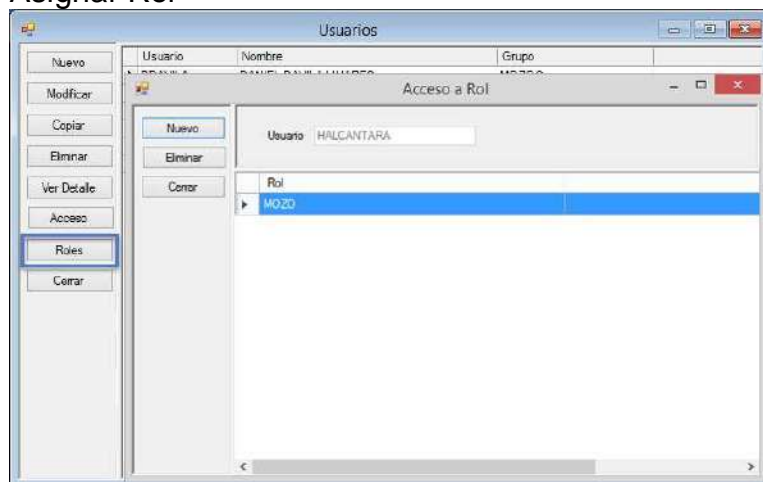


Ilustración 12 Interfaz Asignar Rol – Primer Sprint

✓ Registrar Ambiente:



Nuevo Ambiente

ID Ambiente

Nombre

Piso N°

Aceptar Cancelar

Ilustración 13 Interfaz Registrar Ambiente – Primer Sprint

✓ Modificar Ambiente:

Modificar Ambiente

ID Ambiente

Nombre

Piso N°

Aceptar Cancelar

Ilustración 14 Interfaz Modificar Ambiente – Primer Sprint

✓ Eliminar Ambiente:

Eliminar Ambiente

ID Ambiente

Nombre

Piso N°

Eliminar Cancelar

Ilustración 15 Interfaz Eliminar Ambiente – Primer Sprint

✓

7

Ilustración 16 Interfaz Registrar Mesa – Primer Sprint

✓

9

Ilustración 17 Interfaz Modificar Mesa – Primer Sprint

✓ Eliminar Mesa:

Eliminar Mesa

Codigo: 1

Ambiente: SALA COMUN

Mesa N°: 1

N° Asientos: 4

Estado: LIBRE

Eliminar Cancelar

Ilustración 18 Interfaz Eliminar Mesa – Primer Sprint

✓ Registrar Directorio

Directorio Maestro

Nuevo Modificar Eliminar Ver Detalle

Nombre: [] Dpto: (Todos) Provincia: (Todos) Poblado: []

Nro. Identidad: [] Dirección: []

Código	Nombre/Razón Social	Empleado	RUC/DNI	Dirección
80	LA CHURRASQUERIA SAN LUIS S.A.C.	<input type="checkbox"/>	20488014715	CAL. LOS ALUMNOS 157 URB. SAN LUIS LAMBAYEQUE CHICLAYO CHICLA...
67	LA ESPERANZA SOCIEDAD ANONIMA CERRA...	<input type="checkbox"/>	20480871082	CAL. PROLONG. INDEPENDENCIA 762 LA LIBERTAD CHEPEN PUEBLO NIJ...
212	LA MAGUINA CAFE Y CULTURA S.A.C.	<input type="checkbox"/>	20515062603	CAL. ALCANFOR 323 (CRUCE CON DIEZ CANSECO-A 1 CRA LARCO) LIM...
403	LA NOVENA RESTAURANTES PERUANOS E.I.	<input type="checkbox"/>	20561325546	AV. LA LIBERTAD 231 URB. SANTA VICTORIA LAMBAYEQUE CHICLAYO C...
404	LA NOVENA RESTOLOUNGE E.I.R.L.	<input type="checkbox"/>	20538936601	AV. LA LIBERTAD 231 URB. SANTA VICTORIA LAMBAYEQUE CHICLAYO C...
71	LA POSITIVA SEGUROS Y REASEGUROS	<input type="checkbox"/>	20100210909	CAL. FRANCISCO MASIAS 370 (CRUCE CON AV. JAVIER PRADO ESTE) LIM...
118	LEYVA ESPINAL GONZALO HERNANDO	<input type="checkbox"/>	10164782510	JR. SAN MARTIN 346 SECTOR PUEBLO NUEVO AMAZONAS UTCUBAMBA ...
241	LIBRERIAS CRISOL S.A.C.	<input type="checkbox"/>	20501457869	AV. LOS FAISANES 144 URB. LA CAMPANA#209-A (ESPALDA DEL MERCADO...
405	LIDSTER VALLES, RANDY CHRISTIAN	<input type="checkbox"/>	10064886130	JR. LAMAS 218 SAN MARTIN SAN MARTIN TARAPOTO
406	LILUSAGUI S.A.C.	<input type="checkbox"/>	20561131423	AV. ANDRES A. CACERES 449 URB. SANTA VICTORIA LAMBAYEQUE CHI...
10	LOPEZ DELGADO, JHONATAN SMITH	<input type="checkbox"/>	10419256621	CAL. 7 DE ENERO 945 (CRUCE VICENTE DE LA VEGA Y 7 DE ENERO) LAMB...
407	LOPEZ OTTOYA MARIA CECILIA	<input type="checkbox"/>	10164614412	CAL. FRANCISCO CABRERA 050 INT. 406 (GRAN PLAZA-4TO PISO) LAMBA...
464	LOS PORTALES HOTELES S.A.C.	<input type="checkbox"/>	20518845072	JR. MARISCAL LA MAR 991 (ESQUINA AV. DEL EJERCITO 1205-1215-1225) L...
410	LOZANO ZAVALA MARIO JOSUE	<input type="checkbox"/>	10466916825	MZA IDPTO. 202 LOTE 12 URB VILLA DEL NORTE II ETAPA IZQUIERDA DE...
309	LUBRICANTES Y SERVICIOS ALEX S.R.L.	<input type="checkbox"/>	20487777243	MZA D LOTE 1 A.H. INDOAMERICA COSTADO FABRICA DE DULCES B...
66	MACEDO VILLANUEVA HENRY	<input type="checkbox"/>	10167279584	-

Cerrar

Nuevo Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

Código Directorio: [] ☐ Inactivo

Tipo Persona: PERSONA JURIDICA

Razón Social: []

Primer Nombre: [] Segundo Nombre: []

Apellido Paterno: [] Apellido Materno: []

Dirección: []

Tipo Doc. Identidad: DNI País: PERÚ Provincia: [] Poblado: []

Cia.: THE BEER COMPANY S.A.C. Nro. Identidad: [] Dpto.: [] Distrito: []

Aceptar Cancelar

Ilustración 19 Interfaz Registrar Directorio – Primer Sprint

Nuevo Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

E-mail

Página Web

Fecha Nacimiento

Sexo

Ocupación

Aceptar Cancelar

Nuevo Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

Nuevo

Modificar

Eliminar

Ver Detalle

Item	Nombre	Area

Nuevo Contacto

Item

Saludo Sr.

Nombre

Email

Area

Cargo

Fecha Nacim.

Observación

Aceptar Cancelar

Aceptar Cancelar

Nuevo Item Directorio

Información General Información Adicional Contactos **Teléfonos** Direcciones Notas

Nuevo Modificar Eliminar Ver Detalle

Item	Tipo	Nro. Teléfono	Contacto
Nuevo Item Teléfono			
Item			
Tipo Teléfono	OFICINA		
Nro. Teléfono			
Nro. Anexo			
Contacto			
Observación			
		Aceptar	Cancelar

Aceptar Cancelar

Nuevo Item Directorio

Información General Información Adicional Contactos **Teléfonos** **Direcciones** Notas

Nuevo

Item	Dirección
Nueva Dirección	
Item	
Dirección	
País	PERÚ
Dpto.	
Provincia	
Distrito	
Poblado	
Observación	
Aceptar Cancelar	

Aceptar Cancelar

Nuevo Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

Aceptar Cancelar

✓ Modificar Directorio:

Modificar Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

Código Directorio 121 ☐ Inactivo

Tipo Persona PERSONA JURÍDICA

Razón Social A Y P DISTRIBUCIONES S.A.C.

Primer Nombre Segundo Nombre

Apellido Paterno Apellido Materno

Dirección CAL 6 180 URB. CORPAC LIMA LIMA SAN ISIDRO

Tipo Doc. Identidad RUC Cia. THE BEER COMPANY S.A.C.

País PERÚ Nro. Identidad 20487527646 Sunat

Provincia Dpto. LAMBAYEQUE

Poblado Distrito

Aceptar Cancelar

Ilustración 20 Interfaz Modificar Directorio – Primer Sprint

✓ Eliminar Directorio:

Eliminar Item Directorio

Información General Información Adicional Contactos Teléfonos Direcciones Notas

Código Directorio 121 ☐ Inactivo

Tipo Persona PERSONA JURÍDICA

Razón Social A Y P DISTRIBUCIONES S.A.C.

Primer Nombre Segundo Nombre

Apellido Paterno Apellido Materno

Dirección CAL. 6 180 URB. CORPAC LIMA LIMA SAN ISIDRO

Tipo Doc. Identidad RUC Cia. THE BEER COMPANY S.A.C.

País PERÚ Nro. Identidad 20487527646 Sunat

Provincia Dpto. LAMBAYEQUE

Poblado Distrito

Aceptar Cancelar

Ilustración 21 Interfaz Eliminar Directorio – Primer Sprint

4.6.1.8. GRAFICANDO EL TABLÓN DE TAREAS

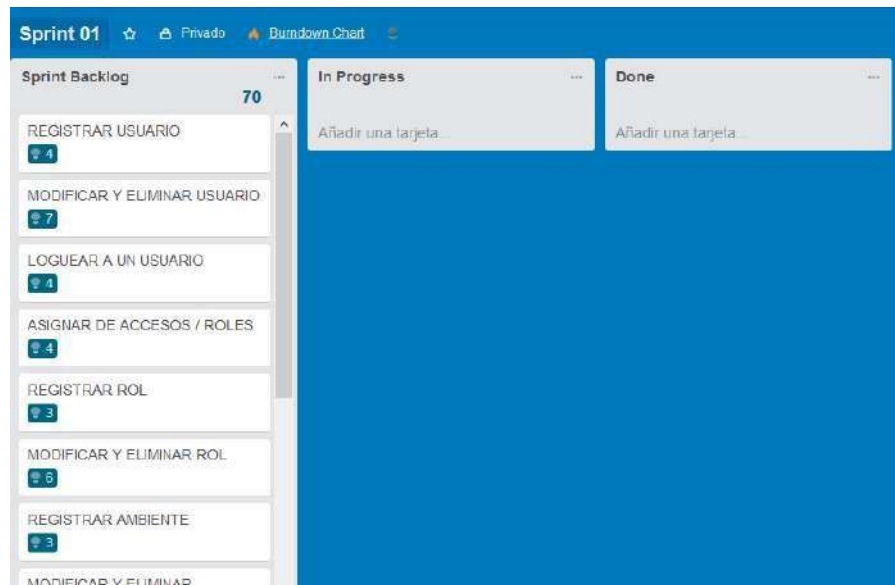


Ilustración 22 Historias de Usuario –Primer Sprint

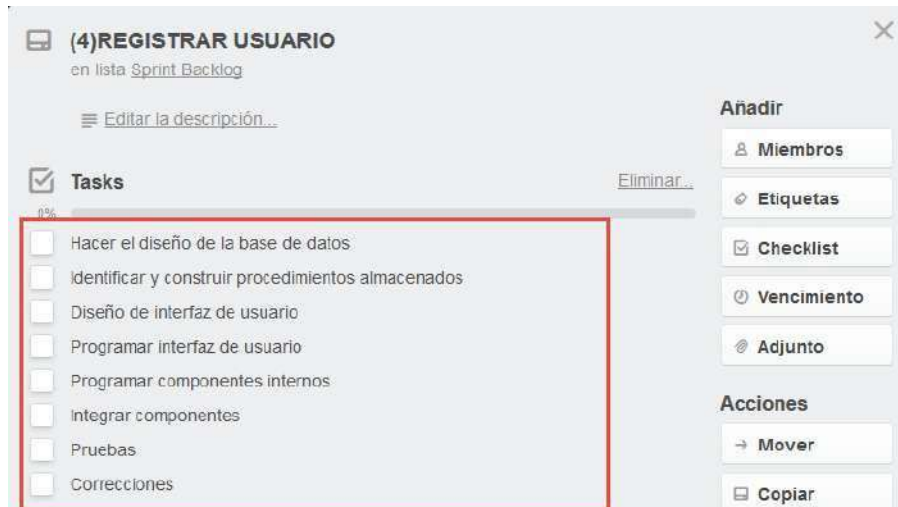


Ilustración 23 H. de Usuario Registrar Usuario

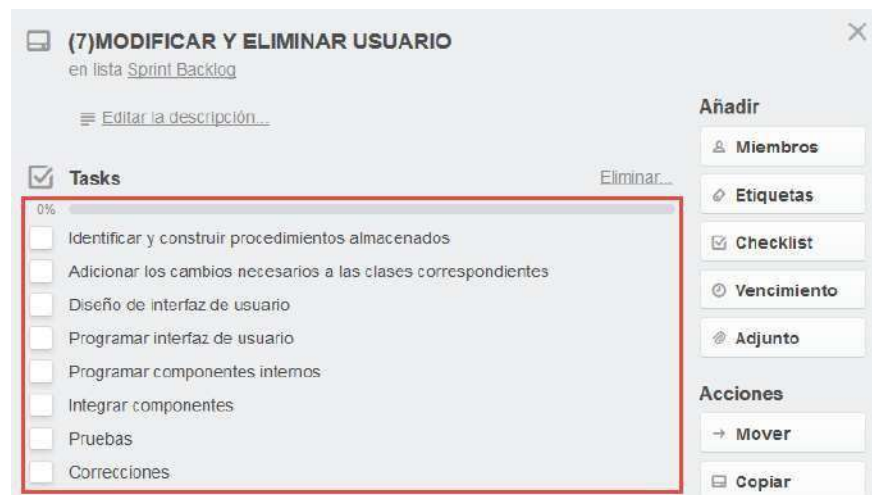


Ilustración 24 H. de Usuario Modificar y Eliminar Usuario

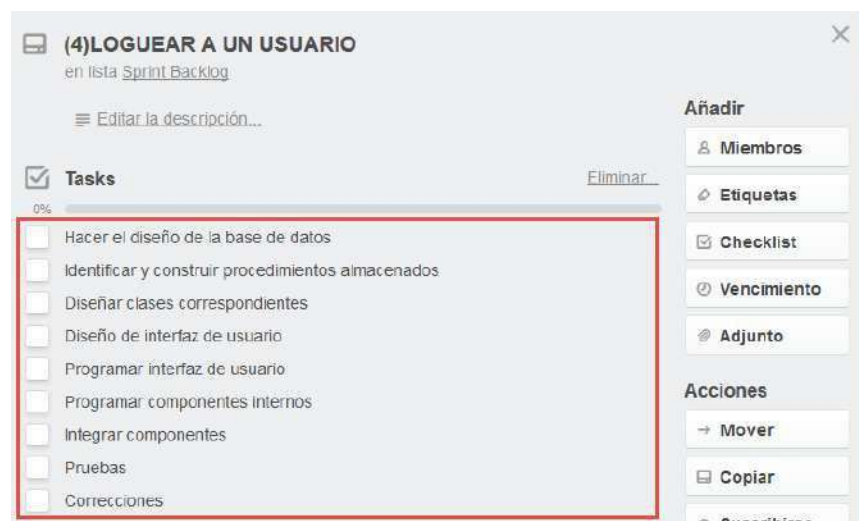


Ilustración 25 H. de Usuario Loguear un Usuario

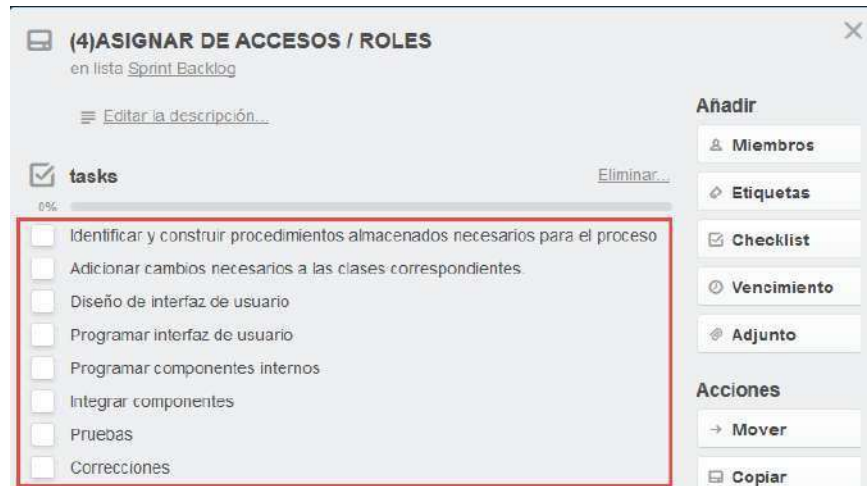


Ilustración 26 H. de Usuario Registrar Asignar Accesos / Roles

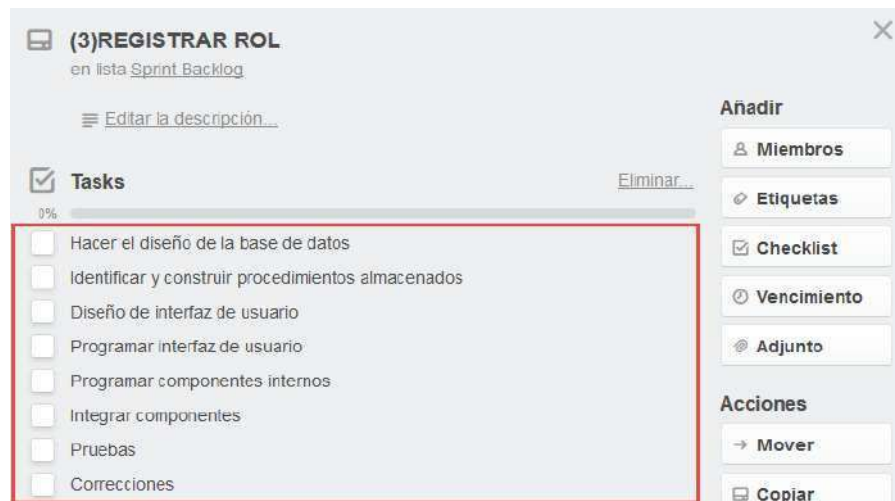


Ilustración 27 H. de Usuario Registrar Rol

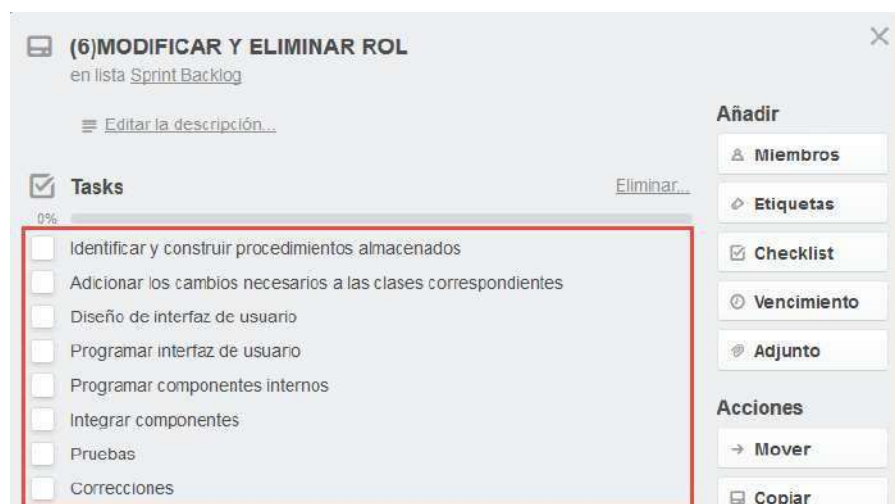


Ilustración 28 H. de Usuario Modificar y Eliminar Rol

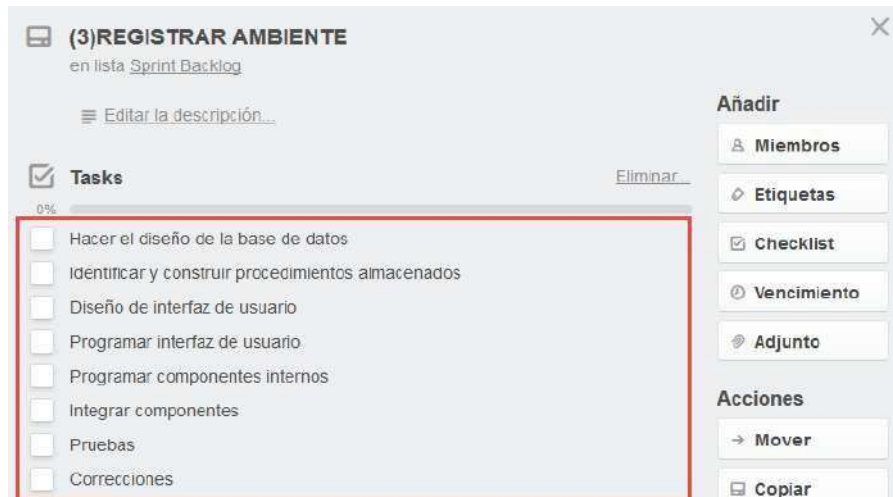


Ilustración 29 H. de Usuario Registrar Ambiente

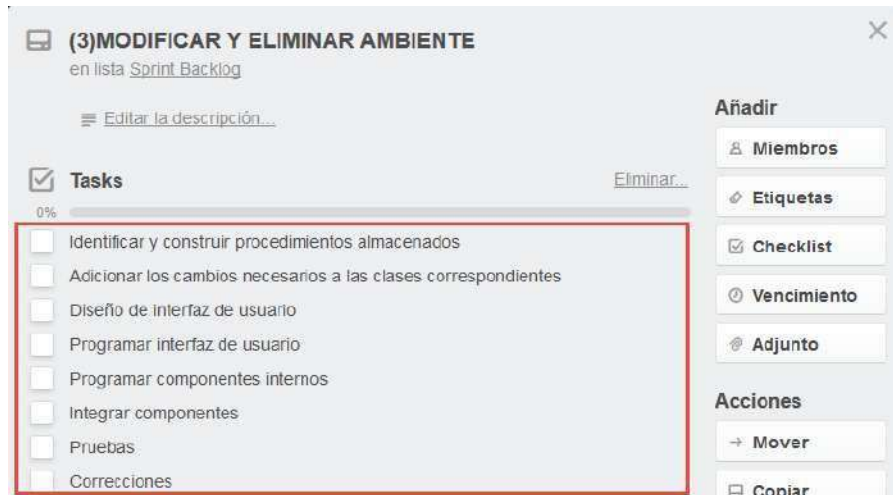


Ilustración 30 H. de Usuario Modificar y Eliminar Ambiente

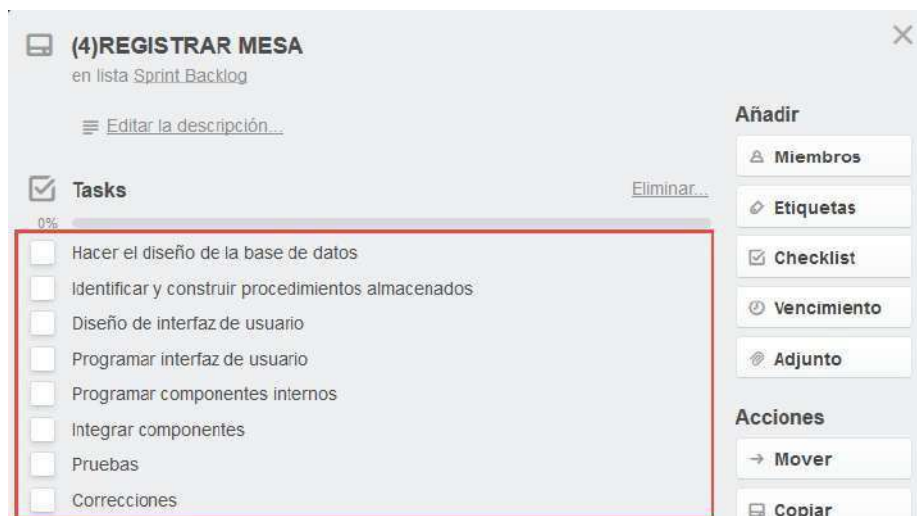


Ilustración 31 H. de Usuario Registrar Mesa

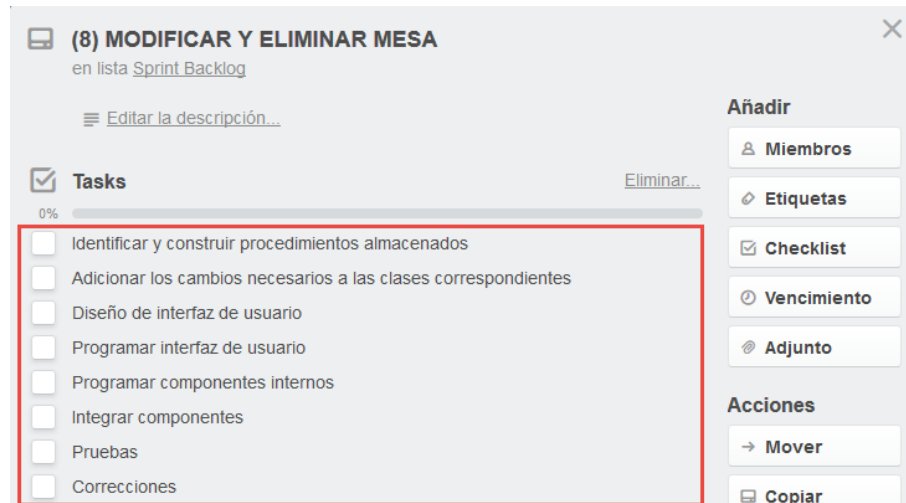


Ilustración 32 H. de Usuario Modificar y Eliminar Mesa

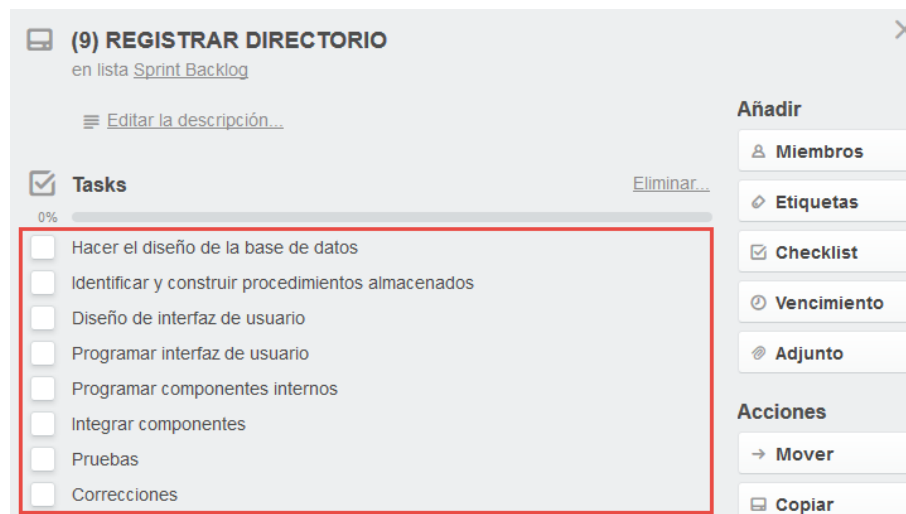


Ilustración 33 H. de Usuario Registrar Directorio

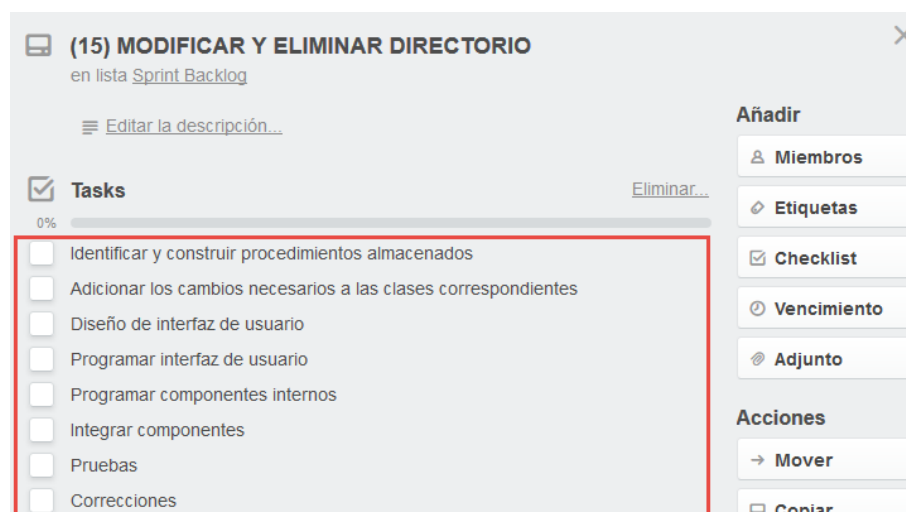


Ilustración 34 H. de Usuario Modificar y Eliminar Directorio

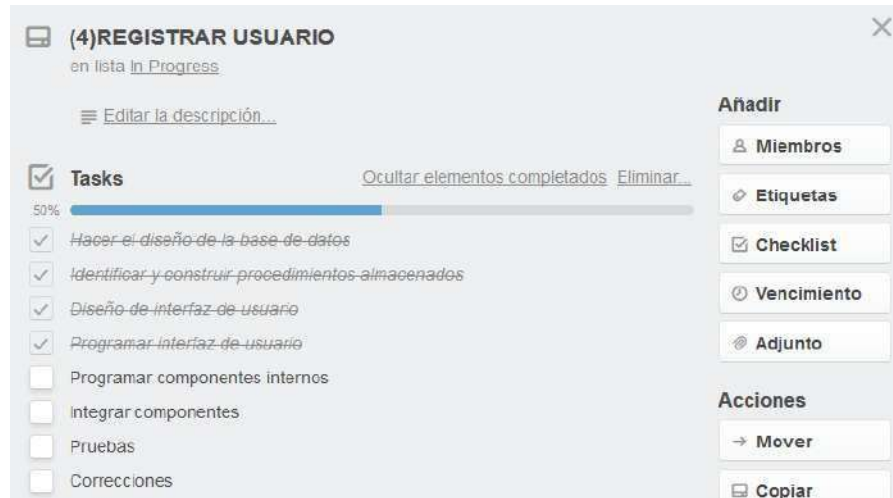
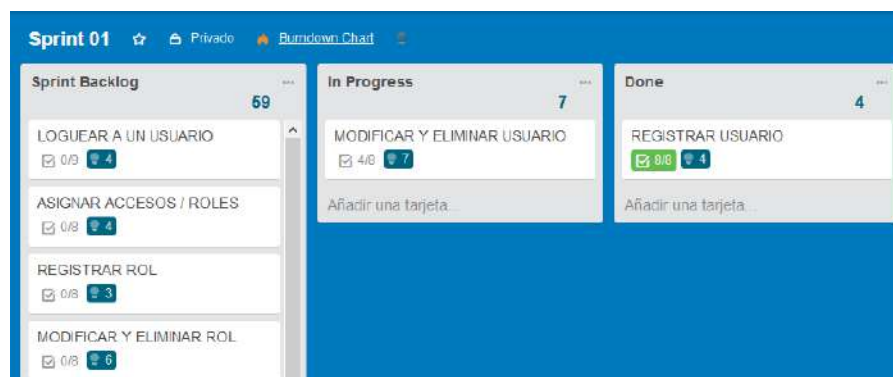


Ilustración 35 Proceso de Tareas Historia 1 – Primer Sprint



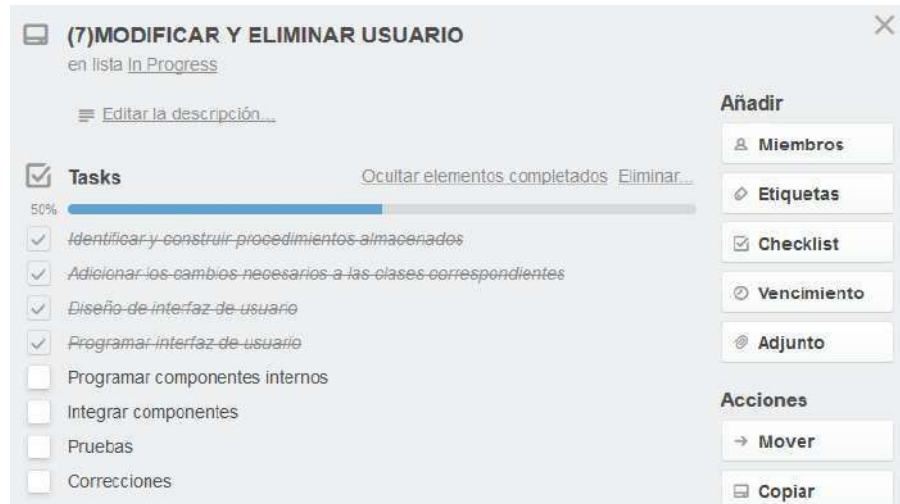


Ilustración 36 Proceso de Tareas Historia 2 – Primer Sprint

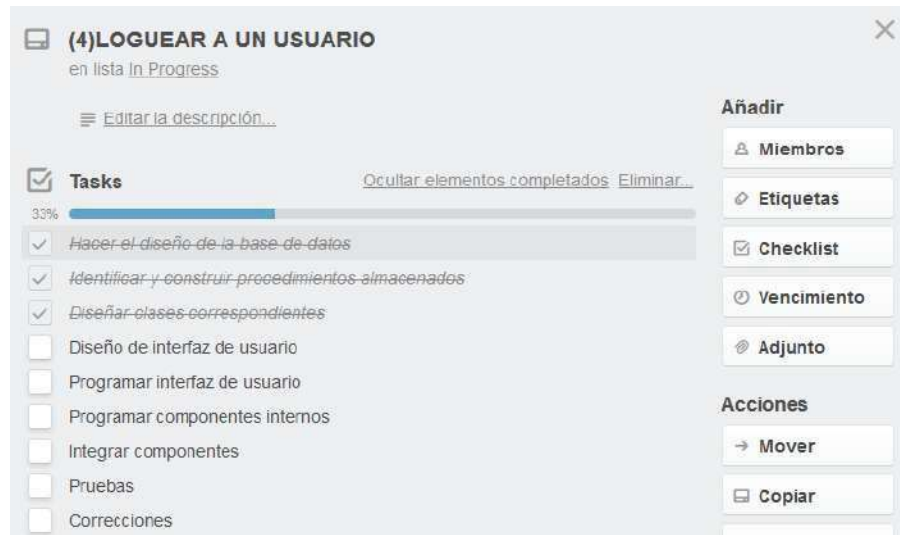
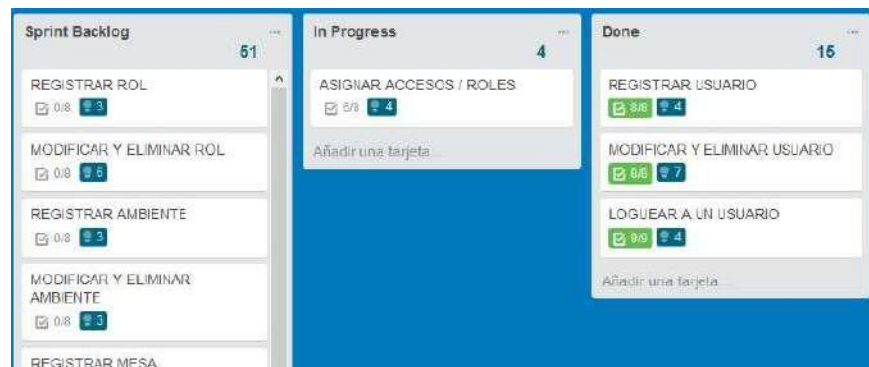


Ilustración 37 Proceso de Tareas Historia 3 – Primer Sprint



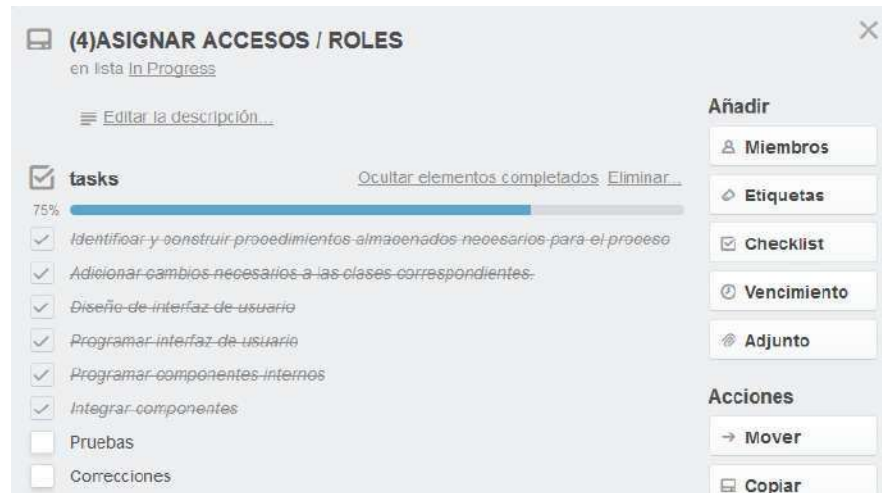


Ilustración 38 Proceso de Tareas Historia 4 – Primer Sprint

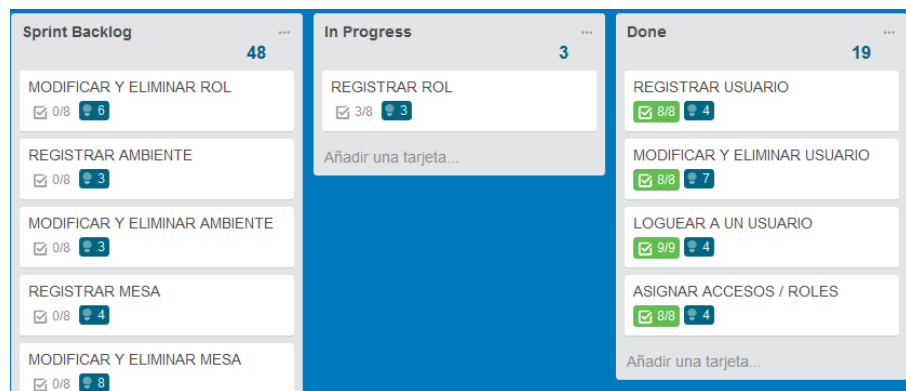


Ilustración 39 Proceso de Tareas Historia 5 – Primer Sprint

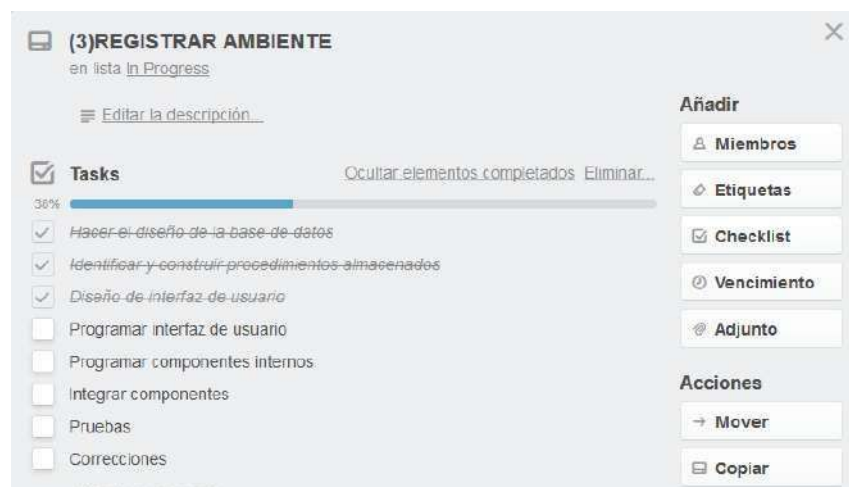
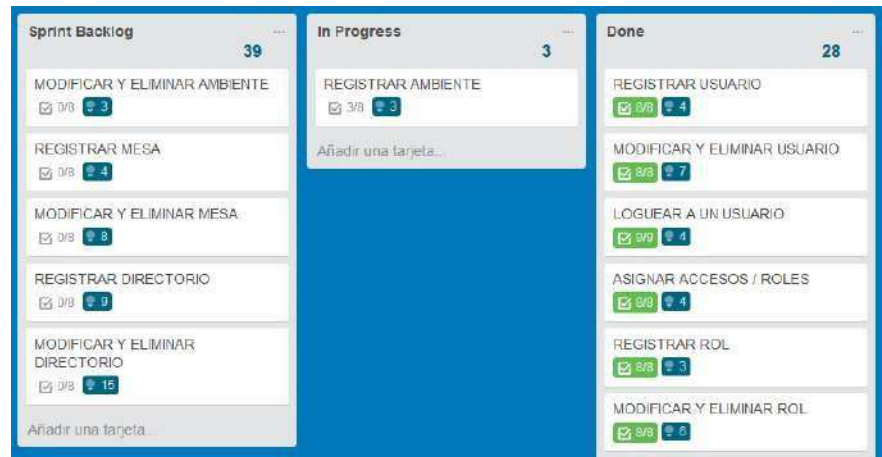
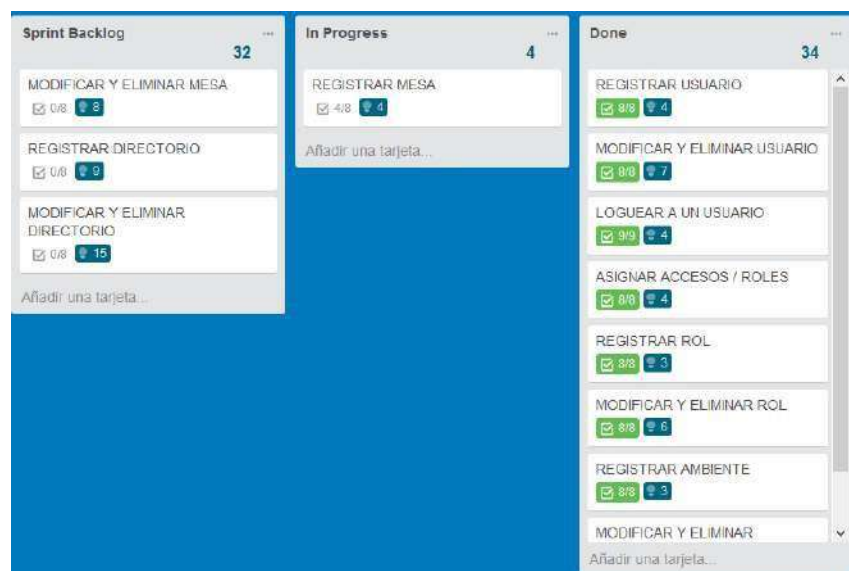


Ilustración 40 Proceso de Tareas Historia 7 – Primer Sprint



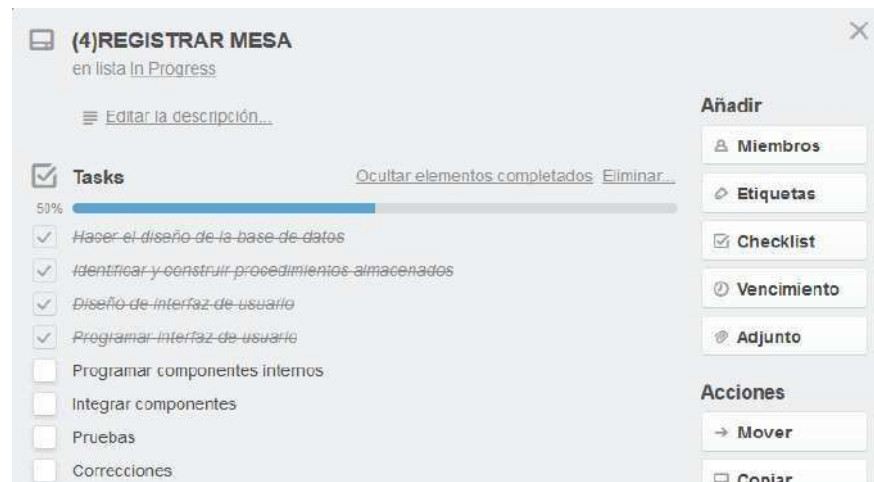


Ilustración 41 Proceso de Tareas Historia 9 – Primer Sprint

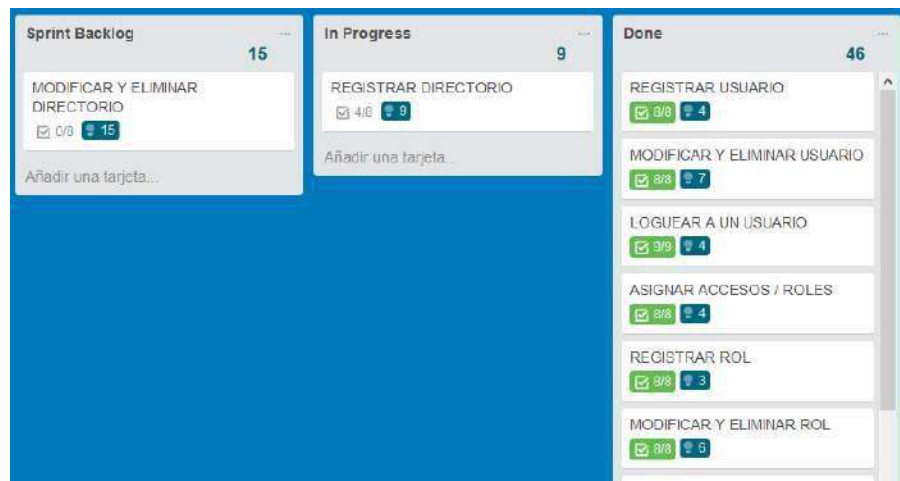


Ilustración 42 Proceso de Tareas Historia 11 – Primer Sprint

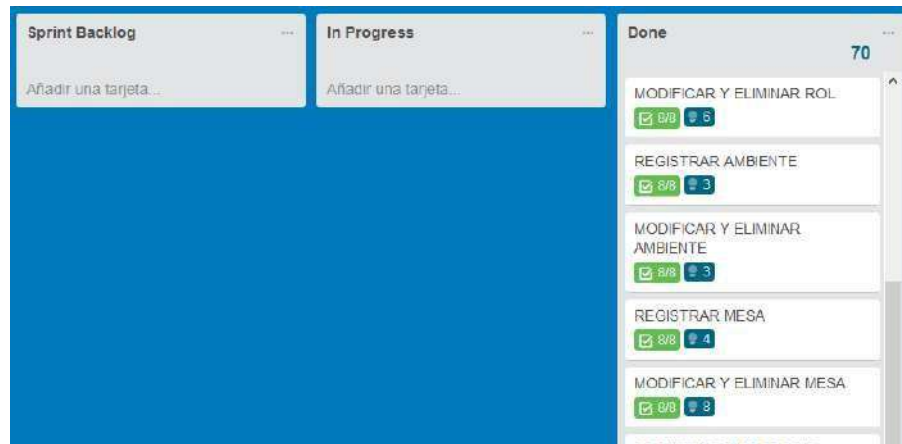


Ilustración 43 Historias de Usuario Concluidas –Primer Sprint

4.6.1.9. DIAGRAMA DE BURN DOWN

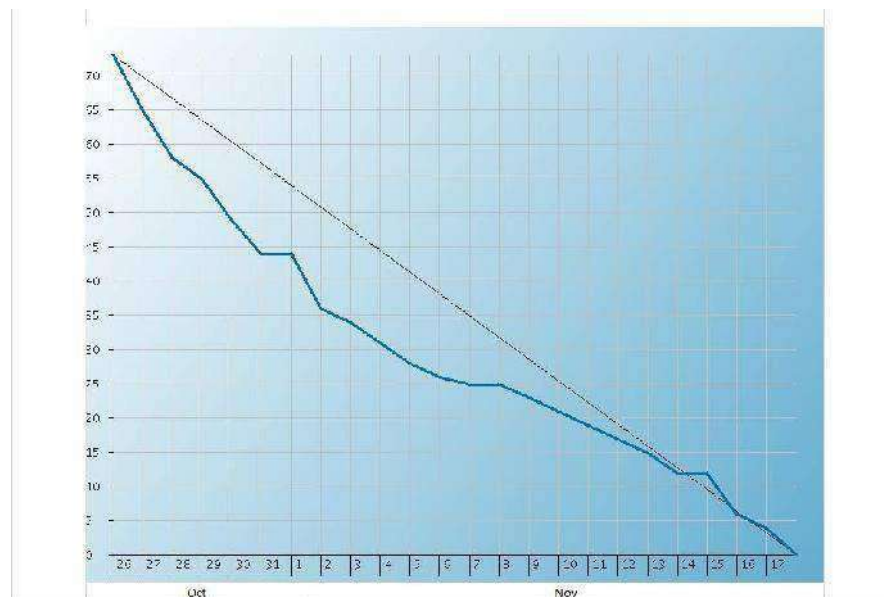
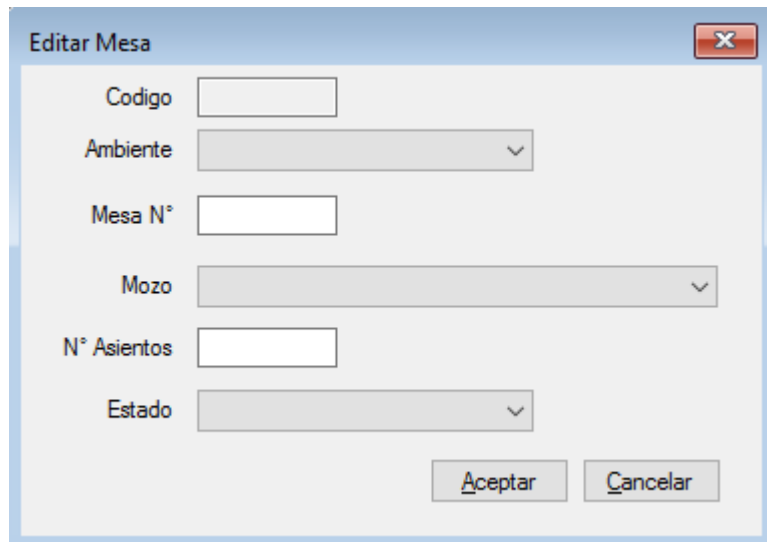


Ilustración 44 Diagrama de Burn Down –Primer Sprint

4.6.1.10. DESARROLLO DE HISTORIAS

Registrar Mesa



Presentacion

```
Private Sub AceptarNuevo()  
    Dim drNuevoAmbiente As DataRow  
    Dim strError As String  
    Try  
        strError = Me.ValidarEntradaDatosCab  
        If strError.Trim <> "" Then  
            MessageBox.Show("Se ha encontrado errores,  
verifique...", "Validar", MessageBoxButtons.OK,  
MessageBoxIcon.Exclamation)  
        Else  
            drNuevoAmbiente = dtMesa.NewRow  
            drMesa = Me.PrepararRegMesa(drNuevoAmbiente)  
            Me.LlamarAdicionar(drMesa, Configuracion.Usuario)  
            Me.LimpiarPantalla()  
            Me.grabacionOK = True  
            Me.Close()  
        End If  
    Catch ex As Exception  
        GestorError.MensajeError()  
    End Try  
End Sub  
  
Private Function PrepararRegMesa(ByVal RegAmbiente As DataRow) As  
    DataRow  
    RegAmbiente.Item("codmesa") = 0  
    RegAmbiente.Item("codambiente") = Me.cboAmbiente.SelectedValue  
    RegAmbiente.Item("nromesa") = Me.txtNroMesa.Text  
    RegAmbiente.Item("nroasientos") = Me.txtNroAsientos.Text  
    RegAmbiente.Item("estado") = Me.cboEstado.SelectedValue  
    RegAmbiente.Item("codmozo") = Me.cboMozo.SelectedValue  
    RegAmbiente.Item("coduser") = Configuracion.Usuario
```

```

        RegAmbiente.Item("coduserori") = Configuracion.Usuario
        RegAmbiente.Item("codcia") = Configuracion.CodCia
        RegAmbiente.Item("eliminado") = 0
        Return RegAmbiente
    End Function

    Private Sub LlamarAdicionar(ByVal drMesa As DataRow, ByVal Usuario
                                As String)

        Dim dsEnvio As New DataSet
        Dim dtTabla As New DataTable
        Dim clienteRemotoObj As ServicioRemotoRestauranteClient =
                                Nothing

        If Configuracion.UsarServidorAplicaciones Then
            Try
                dtTabla = drMesa.Table.Clone
                dtTabla.LoadDataRow(drMesa.ItemArray, True)
                dtTabla.TableName = "Mesa"
                dsEnvio.Tables.Add(dtTabla)
                clienteRemotoObj =
                    Proxy.Instanciar(TipoServicio.Restaurante)
                clienteRemotoObj.AdicionarMesa(dsEnvio,
                    Configuracion.CadenaConexion)
            Catch ex As Exception
                Throw ex
            Finally
                clienteRemotoObj.Close()
            End Try
        End If
    End Sub

```

(WFC) Servicio Remoto Restaurante

```

Public Sub AdicionarMesa(ByVal dsEnvio As DataSet, cadenaConexion
    As String) Implements IServicioRemotoRestaurante.AdicionarMesa
    Dim mesaObj As Mesa
    Try
        cadenaConexion =
            Configuracion.CompletarCredenciales(cadenaConexion)
        mesaObj = New Mesa(cadenaConexion)
        mesaObj.Adicionar(dsEnvio.Tables(0).Rows(0))
    Catch ex As Exception
        Throw ex
    End Try
End Sub

```

Logico de Negocio

```

Public Sub Adicionar(ByVal drMesa As DataRow)
    Dim correlativo As New CorrelativoAD(mCadenaConexion)
    Try
        Me.IniciarTransaccion()
        correlativo.RecibirTransaccion(Me.Transaccion, Me.Conexion)
        MesaADObj.RecibirTransaccion(Me.Transaccion, Me.Conexion)
        drMesa.Item("codmesa") =
            correlativo.ObtenerCorrelativo("RE", "002")
    End Try
End Sub

```

```

        ProcesarAdicionar(drMesa, MesaADObj)
        Me.AceptarTransaccion()
    Catch e As Exception
        CancelarTransaccion()
        Throw (e)
    End Try
End Sub

```

Acceso a Datos

```

Public Sub ProcesarAdicionar(ByVal drMesa As DataRow, ByVal
                                MesaADObj As MesaAD)
    MesaADObj.AdicionarMesa(drMesa)
End Sub

Public Function AdicionarMesa(ByVal filaMesa As DataRow) As Integer
    Me.LimpiarParametros()
    Me.AgregarParametro("@codmesa", filaMesa.Item("codmesa"),
        Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@codambiente",
        filaMesa.Item("codambiente"), Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@nromesa", filaMesa.Item("nromesa"),
        Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@nroasientos",
        filaMesa.Item("nroasientos"), Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@estado", filaMesa.Item("estado"),
        Enumeracion.TipoDatoBD.CharE, 10)
    Me.AgregarParametro("@codmozo", filaMesa.Item("codmozo"),
        Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@coduserori", filaMesa.Item("coduserori"),
        Enumeracion.TipoDatoBD.CharE, 10)
    Me.AgregarParametro("@coduser", filaMesa.Item("coduser"),
        Enumeracion.TipoDatoBD.CharE, 10)
    Me.AgregarParametro("@fecori", Date.Now,
        Enumeracion.TipoDatoBD.DateTime)
    Me.AgregarParametro("@fecact", Date.Now,
        Enumeracion.TipoDatoBD.DateTime)
    Me.AgregarParametro("@codcia", filaMesa.Item("codcia"),
        Enumeracion.TipoDatoBD.CharE, 3)
    Me.AgregarParametro("@idsucursal", filaMesa.Item("idsucursal"),
        Enumeracion.TipoDatoBD.Int)
    Me.AgregarParametro("@eliminado", filaMesa.Item("eliminado"),
        Enumeracion.TipoDatoBD.Bit)
    Return Me.MantenimientoBD(" INSERT INTO REMESA (codmesa,
        codambiente, nromesa, nroasientos, estado, codmozo,
        coduserori, coduser, fecori, fecact, codcia, idsucursal,
        eliminado) " +
        "VALUES (@codmesa, @codambiente,
        @nromesa, @posicionx, @estado, @codmozo, @coduserori,
        @coduser, @fecori, @fecact, @codcia, @idsucursal,
        @eliminado)", True)
End Function

```

4.6.2. SPRINT 2

4621. CALCULO DE LA VELOCIDAD ESTIMADA PARA EL SEGUNDO SPRINT

✓ Calculando duración del sprint:

1 semana y 4 días= 9 días laborables.

Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	8
Programador 2	9
Total	17 días hombres disponible

Tabla 23. Calculo Días Hombre Segundo Sprint

✓ Calculando factor de dedicación (FD):

FD = Puntos de historia (último sprint) / días hombres disponible

FD = 70/17 = 4.18

✓ Velocidad estimada: FD X Días –Hombre- Disponible

17 * 4.18 = 71.06

4622. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL SEGUNDO SPRINT:

Nº	SPRINT 1	PUNTOS	ESTADO
1	REGISTRAR MOZO	4	Terminado
2	MODIFICAR Y ELIMINAR MOZO	7	Terminado
3	REGISTRAR CLASE – SUBCLASE ARTICULO	7	Terminado
4	MODIFICAR Y ELIMINAR CLASE – SUBCLASE ARTICULO	12	Terminado
5	REGISTRAR ALMACEN	4	Terminado

6	MODIFICAR Y ELIMINAR ALMACEN	8	Terminado
7	REGISTRAR ARTICULO	6	Terminado
8	MODIFICAR Y ELIMINAR ARTICULO	10	Terminado
9	REGISTRAR CARTA	5	Terminado
10	MODIFICAR Y ELIMINAR CARTA	7	Terminado
11	REGISTRAR RESERVACIÓN MESA	4	Terminado
12	MODIFICAR Y ELIMINAR RESERVACIÓN MESA	7	Terminado
*	TOTAL DE PUNTOS DE HISTORIA	81	81
Objetivo	<ul style="list-style-type: none"> • Registrar los mozos de la empresa. • Registrar artículos. • Registrar la carta de los productos que tendrá un pedido. • Registrar la reservación de una mesa. 		

Tabla 24. Productos en el Segundo Sprint

4.6.2.3. DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre de Historia: Registrar Mozo	
Prioridad en Negocio: baja	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo mozo el cual más adelante podrá registrar un pedido mediante el dispositivo móvil.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Restaurante / Gestión de Mozos. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del mozo. ✓ Dar clic en el botón Aceptar para confirmar la creación del mozo. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero

	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 25. 1° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador
Nombre de Historia: Modificar y eliminar mozo	
Prioridad en Negocio: media	
Puntos estimados: 7	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente a los mozos, esta opción te permitirá modificar los datos del mozo que sean conveniente, así como también eliminar al mozo de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un usuario y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del mozo y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui

Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	Julio Alex Ventura Chero José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	Julio Alex Ventura Chero
Correcciones	José Human Valqui
Firma:	

Tabla 26 2° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador
Nombre de Historia: Registrar clase – subclase articulo	
Prioridad en Negocio: Media	
Puntos estimados: 7	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción del sistema te permitirá la creación de una nueva clase y subclase que será asignado a un artículo posteriormente.	
Pruebas: <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Almacén / Gestión Clase-Subclase Artículos. ✓ Dar clic en el botón nuevo. 	

<ul style="list-style-type: none"> ✓ Llenar los datos solicitados para el registro correcto de la clase-subclase. ✓ Dar clic en el botón Aceptar para confirmar la creación de la clase-subclase. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	Julio Alex Ventura Chero
Correcciones	José Human Valqui
Firma:	

Tabla 27. 3° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Clase SubClase	
Prioridad en Negocio: Media	
Puntos estimados: 12	Iteración asignada: 12
Programador responsable: Julio Alex Ventura Chero	

<p>Descripción:</p> <p>Habiendo registrado previamente una clase-subclase de artículo, esta opción te permitirá modificar los datos del registro seleccionado, así como también eliminar de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar una clase-subclase de articulo y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos de la carta y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 28. 4° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador
Nombre de Historia: Registrar Almacén	
Prioridad en Negocio: Baja	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo almacén el cual más adelante se le asignará a un artículo.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Almacén / Gestión Almacenes. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del almacén. ✓ Dar clic en el botón Aceptar para confirmar la creación del almacén. 	
Tareas	Responsable
Hacer el diseño de la base de datos	José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui

Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 29. 5° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Almacén	
Prioridad en Negocio: Media	
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente el almacén, esta opción te permitirá modificar los datos del almacén, así como también eliminar el registro de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un almacén y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos de la carta y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui

Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 30. 6° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador
Nombre de Historia: Registrar Artículo	
Prioridad en Negocio: Media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción del sistema te permitirá la creación de un nuevo artículo.	
Pruebas: <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Almacén / Gestión de Artículos. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del artículo. ✓ Dar clic en el botón Aceptar para confirmar la creación del artículo. 	

Tareas	Responsable
Hacer el diseño de la base de datos	José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 31. 7° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Artículo	
Prioridad en Negocio: Media	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Habiendo registrado previamente el artículo, esta opción te permitirá modificar los datos del artículo, así como también eliminar el artículo de ser necesario.	
Pruebas: ✓ Al seleccionar un artículo y dar clic en el botón modificar o eliminar, cargará otra ventana con los datos del	

articulo y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar.	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 32. 8° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador
Nombre de Historia: Registrar Carta	
Prioridad en Negocio: Media	
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción del sistema te permitirá la creación de una nueva carta el cual más adelante un usuario podrá ser asignado.	

Pruebas: <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Restaurante / Gestión de Carta. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto de la carta. ✓ Dar clic en el botón Aceptar para confirmar la creación de la carta. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	Julio Alex Ventura Chero
Correcciones	José Human Valqui
Firma:	

Tabla 33. 9° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar Carta	
Prioridad en Negocio: Media	
Puntos estimados: 7	Iteración asignada: 1

Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente la carta, esta opción te permitirá modificar los datos de la carta así como también eliminar la carta de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar una carta y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos de la carta y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	José Human Valqui
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 34. 10° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 11	Usuario: Caja
Nombre de Historia: Registrar reservación de mesa	
Prioridad en Negocio: Media	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá registrar una reservación de mesa.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Restaurante / Reservaciones de Mesa. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto de la reservación. ✓ Dar clic en el botón Aceptar para confirmar del registro de la reservación. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero

Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 35. 11° Historia de Usuario - Segundo Sprint

HISTORIA DE USUARIO	
Número: 12	Usuario: Caja
Nombre de Historia: Modificar y Eliminar Reservación de Mesa	
Prioridad en Negocio: Media	
Puntos estimados: 7	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente la reservación de una mesa, esta opción te permitirá modificar los datos de la reservación, así como también eliminar el registro de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar una reservación y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos de la reservación y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	

Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 36. 12° Historia de Usuario - Segundo Sprint

4.6.24. DIAGRAMA LOGICO Y FISICO SPRINT 2

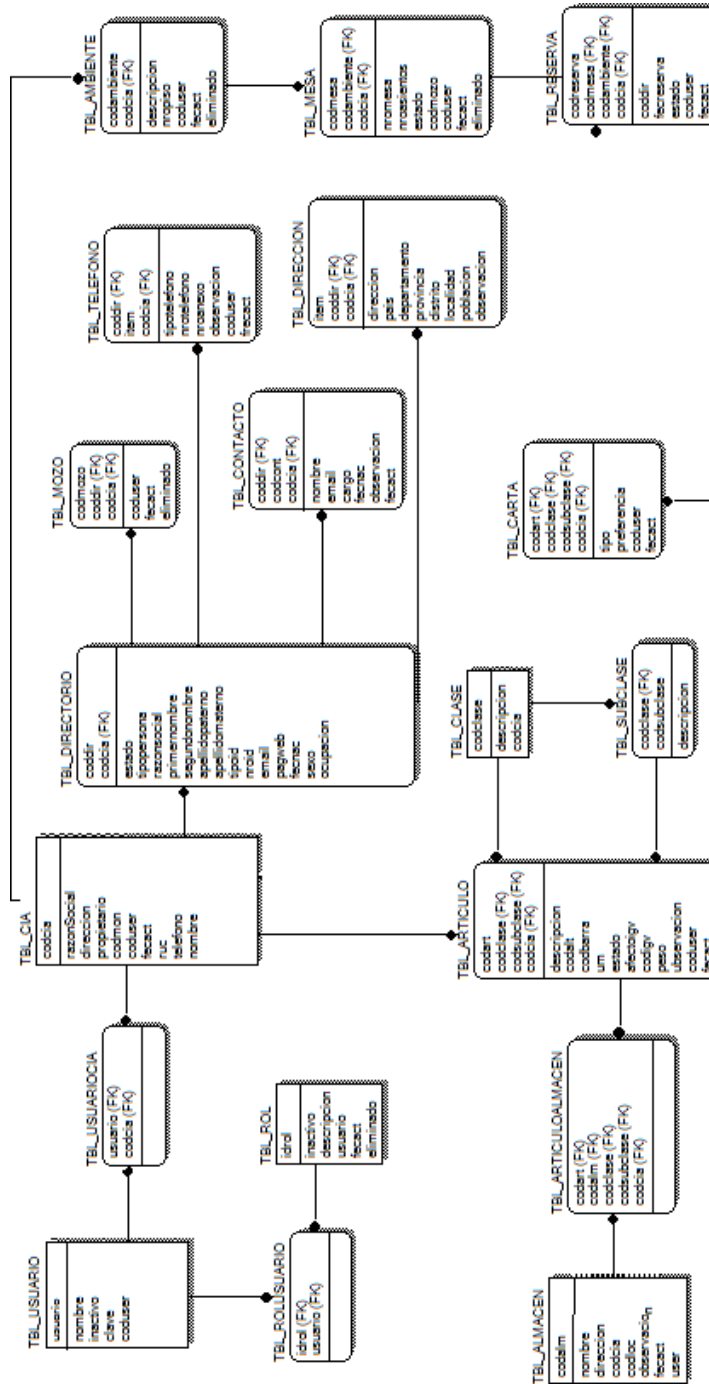
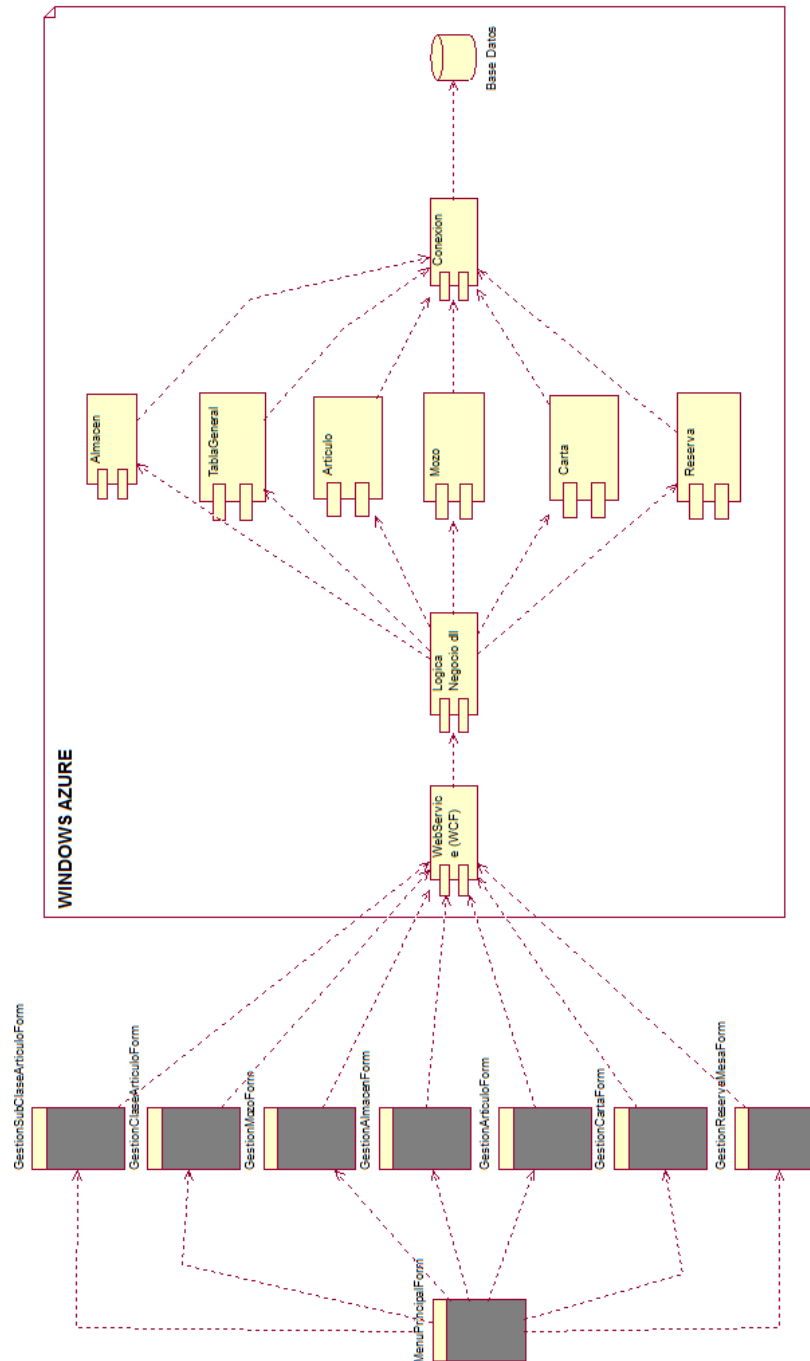


Ilustración 45 Diagrama lógico – Segundo Sprint

4.6.2.5. DIAGRAMA DE COMPONENTES SPRINT 2



4.6.26. INTERFAZ DE LAS HISTORIAS DE USUARIO

✓ Registrar Mozo:

The image shows two screenshots of a software interface. The top screenshot is a window titled 'Mozos'. On the left is a sidebar with buttons: 'Nuevo', 'Modificar', 'Eliminar', 'Ver Detalle', and 'Cerrar'. The main area has a search bar labeled 'Nombre' with a 'Buscar' button. Below is a table with columns: 'Codigo', 'Nombre', 'Direccion', 'RUC', 'Telefono', and 'Email'. One row is highlighted in blue: '121 | A Y P DISTRIBUCIONES S.A. | CAL 6 180 URB. CORPAC | 20487527646'. At the bottom are pagination controls: 'Primera', 'Anterior', 'Página 1 de 1', 'Siguiente', and 'Ultima'. The bottom screenshot is a dialog box titled 'Nuevo Mozo' with a red close button. It contains two input fields: 'Código' (with a dropdown arrow) and 'Nombre'. At the bottom are 'Aceptar' and 'Cancelar' buttons.

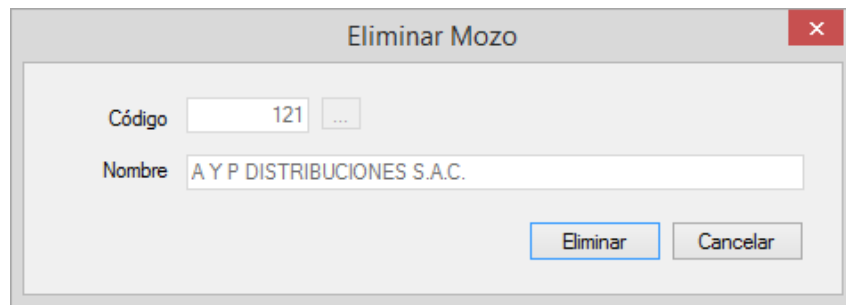
Ilustración 47 Interfaz Registrar Mozo – Segundo Sprint

✓ Modificar Mozo:

The image shows a dialog box titled 'Modificar Mozo' with a red close button. It contains two input fields: 'Código' (with a dropdown arrow) and 'Nombre'. The 'Código' field has the value '121' selected. The 'Nombre' field has the value 'A Y P DISTRIBUCIONES S.A.C.'. At the bottom are 'Aceptar' and 'Cancelar' buttons.

Ilustración 48 Interfaz Modificar Mozo – Segundo Sprint

✓ Eliminar Mozo:



Eliminar Mozo

Código 121 ...

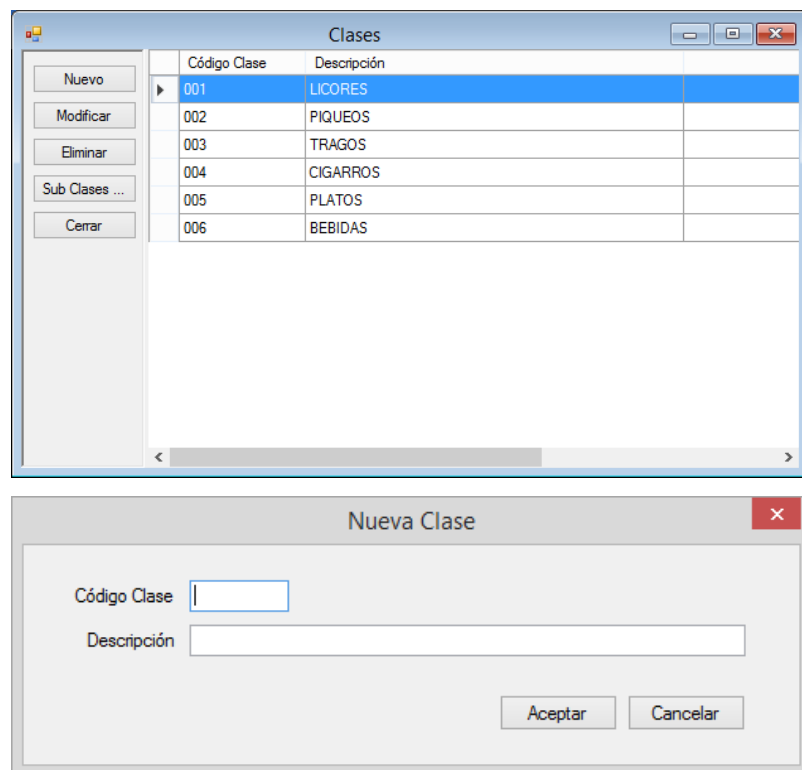
Nombre A Y P DISTRIBUCIONES S.A.C.

Eliminar Cancelar

Ilustración 49 Interfaz Eliminar Mozo – Segundo Sprint

✓ Registrar Clase-SubClase:

- Registrar Clase



Clases

Código Clase	Descripción
001	LICORES
002	PIQUEOS
003	TRAGOS
004	CIGARROS
005	PLATOS
006	BEBIDAS

Nueva Clase

Código Clase

Descripción

Aceptar Cancelar

Ilustración 50 Interfaz Registrar Clase – Segundo Sprint

- Registrar SubClase

The 'Sub Clases' window displays a table of sub-classes under the 'LICORES' class. The table has two columns: 'Sub Clase' and 'Descripción'.

Sub Clase	Descripción
001	TEQUILA
002	RON
003	WISKY
004	PISCO
005	VINOS
006	CREMAS
007	VODKA
008	HIERVAS
009	CERVEZAS

On the left side of the window, there are buttons: 'Nuevo', 'Modificar', 'Eliminar', and 'Cerrar'. At the top, the 'Clase' is set to '001' and 'LICORES'.

The 'Nueva SubClase' window contains two input fields: 'Código SubClase' and 'Descripción'. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Ilustración 51 Interfaz Registrar SubClase – Segundo Sprint

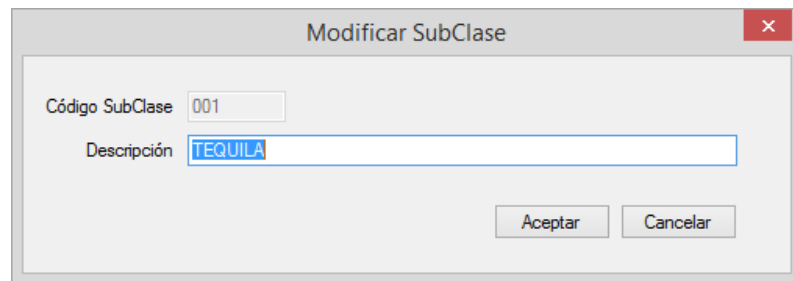
✓ Modificar Clase-SubClase:

- Modificar Clase:

The 'Modificar Clase' window shows the 'Código Clase' as '001' and the 'Descripción' as 'LICORES'. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Ilustración 52 Interfaz Modificar Clase – Segundo Sprint

- Modificar SubClase:



Modificar SubClase

Código SubClase 001

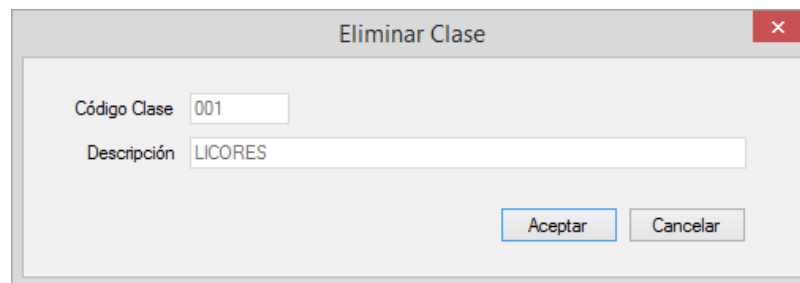
Descripción TEQUILA

Aceptar Cancelar

Ilustración 53 Interfaz Modificar SubClase – Segundo Sprint

- ✓ Eliminar Clase-SubClase:

- Eliminar Clase:



Eliminar Clase

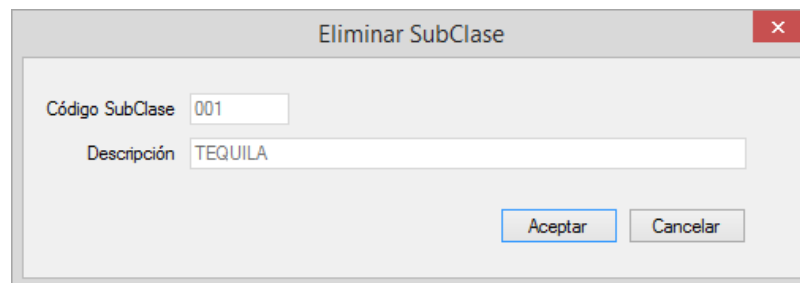
Código Clase 001

Descripción LICORES

Aceptar Cancelar

Ilustración 54 Interfaz Eliminar Clase – Segundo Sprint

- Eliminar SubClase:



Eliminar SubClase

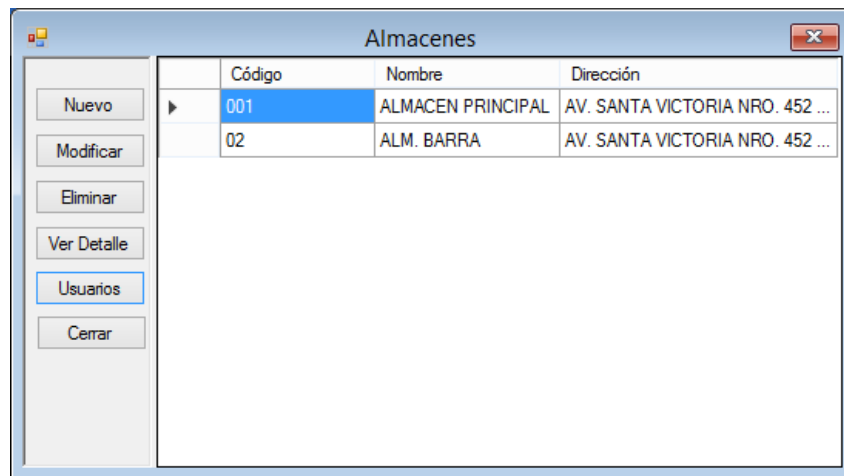
Código SubClase 001

Descripción TEQUILA

Aceptar Cancelar

Ilustración 55 Interfaz Eliminar SubClase – Segundo Sprint

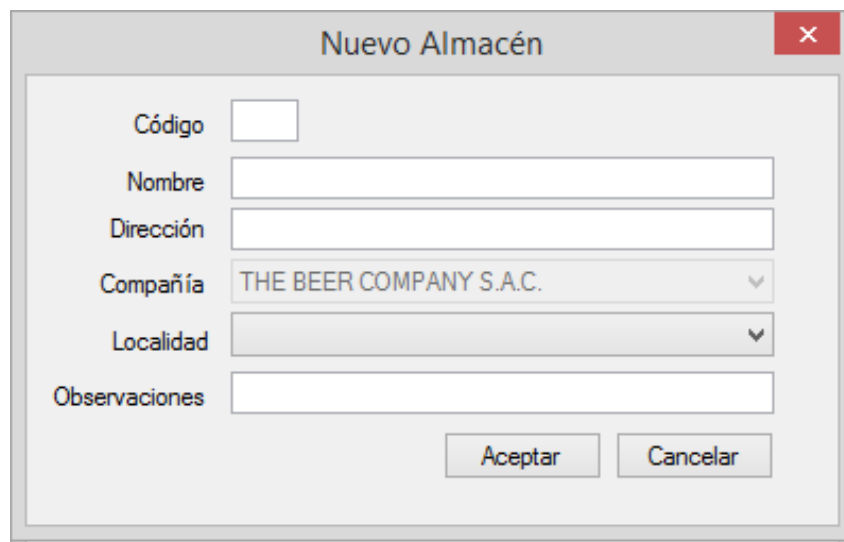
✓ Registrar Almacén:



The 'Almacenes' window displays a table with three columns: Código, Nombre, and Dirección. The first row is highlighted in blue.

Código	Nombre	Dirección
001	ALMACEN PRINCIPAL	AV. SANTA VICTORIA NRO. 452 ...
02	ALM. BARRA	AV. SANTA VICTORIA NRO. 452 ...

On the left side of the window, there is a vertical menu with the following buttons: Nuevo, Modificar, Eliminar, Ver Detalle, Usuarios (highlighted), and Cerrar.

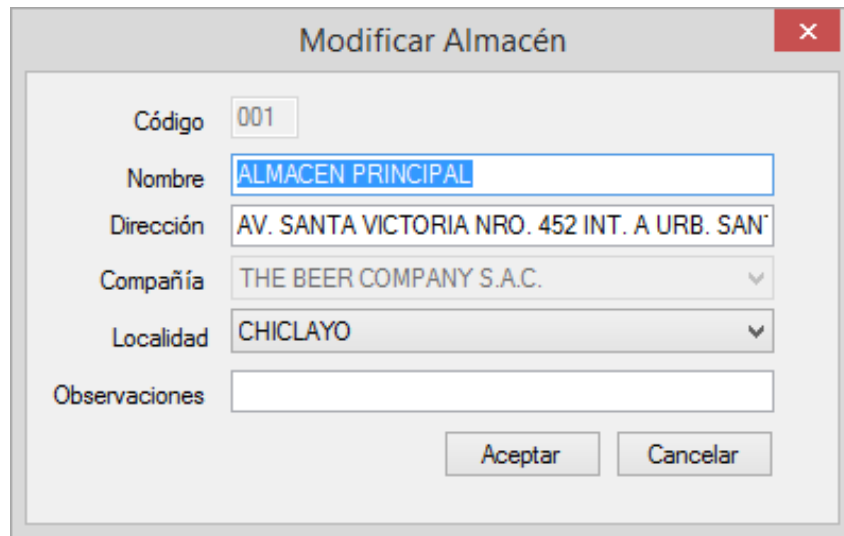


The 'Nuevo Almacén' window contains the following fields and controls:

- Código:
- Nombre:
- Dirección:
- Compañía: (dropdown arrow)
- Localidad: (dropdown arrow)
- Observaciones:
- Buttons: Aceptar, Cancelar

Ilustración 56 Interfaz Registrar Almacén – Segundo Sprint

✓ Modificar Almacén:

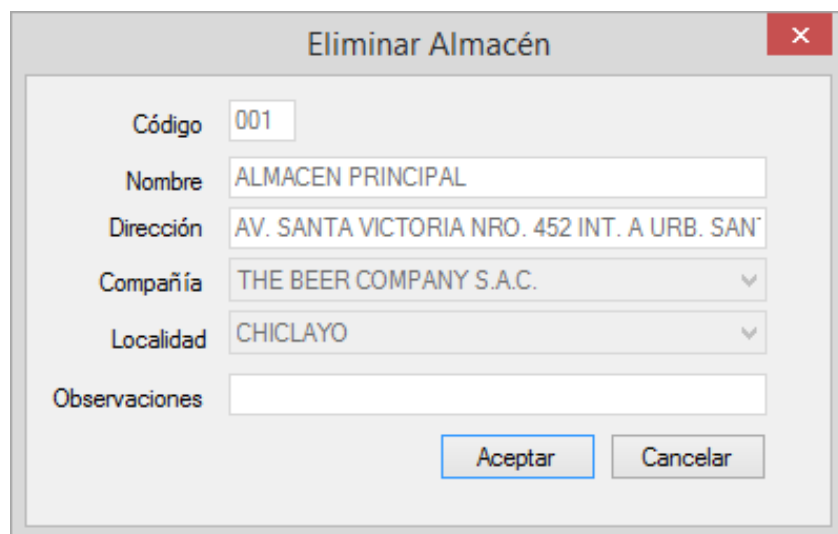


The screenshot shows a dialog box titled "Modificar Almacén" with a red close button in the top right corner. The dialog contains the following fields and controls:

- Código:** A text box containing the value "001".
- Nombre:** A text box containing the value "ALMACEN PRINCIPAL".
- Dirección:** A text box containing the value "AV. SANTA VICTORIA NRO. 452 INT. A URB. SAN".
- Compañía:** A dropdown menu with the selected value "THE BEER COMPANY S.A.C." and a downward arrow.
- Localidad:** A dropdown menu with the selected value "CHICLAYO" and a downward arrow.
- Observaciones:** An empty text box.
- Buttons:** Two buttons at the bottom right: "Aceptar" and "Cancelar".

Ilustración 57 Interfaz Modificar Almacén – Segundo Sprint

✓ Eliminar Almacén:



The screenshot shows a dialog box titled "Eliminar Almacén" with a red close button in the top right corner. The dialog contains the following fields and controls:

- Código:** A text box containing the value "001".
- Nombre:** A text box containing the value "ALMACEN PRINCIPAL".
- Dirección:** A text box containing the value "AV. SANTA VICTORIA NRO. 452 INT. A URB. SAN".
- Compañía:** A dropdown menu with the selected value "THE BEER COMPANY S.A.C." and a downward arrow.
- Localidad:** A dropdown menu with the selected value "CHICLAYO" and a downward arrow.
- Observaciones:** An empty text box.
- Buttons:** Two buttons at the bottom right: "Aceptar" and "Cancelar".

Ilustración 58 Interfaz Eliminar Almacén – Segundo Sprint

✓ Registrar Artículo:

The 'Artículos' window features a search interface on the right and a list table below it. The search filters include:

- Descripción: Contiene []
- Clase: (Todos) [v] Sub Clase: (Todos) [v]
- Cód. Artículo: Igual []
- Cód. Alterno: Contiene []

A 'Buscar' button is located to the right of the filters. The table below has columns: Código, Descripción, U. Medida, and Estado. The table is currently empty.

Navigation buttons at the bottom include: Primera, Anterior, Página [] de (0), Siguiente, and Última.

The 'Nuevo Artículo' window has tabs for 'Datos Generales', 'UM Implícitas', 'Categorías', 'Características Motos', and 'Producción'. The 'Datos Generales' tab is active, showing the following fields:

- Código: []
- Descripción: []
- Código Alterno: []
- Unid. Medida: [] [v]
- Clase: [] [v]
- Peso (Kg.): []
- Marca: [] [v]
- Almacenamiento: [] [v]
- Proveedor: [] [v]
- Observaciones: []

Additional fields on the right include:

- ☐ Servicio
- Código Barras: []
- Estado: [] [v]
- SubClase: [] [v]
- ☒ Afecto IGV
- Código IGV: [] [v]
- Tipo: COMPRADO [v]

'Aceptar' and 'Cancelar' buttons are at the bottom right.

Ilustración 59 Interfaz Registrar Artículo – Segundo Sprint

✓ Modificar Artículo:

Modificar Artículo

Datos Generales UM Implícitas Categorías Características Motos Producción

Código 804 ☐ Servicio

Descripción ARROZ CHAUFA

Código Alterno

Unid. Medida PTO PLATO Estado A ACTIVO

Clase 005 PLATOS SubClase 001 SUBCLASEA

Peso (Kg.) .00000 ☒ Afecto IGV

Marca

Almacenamiento STOCK Código IGV IG1 IGV 18%

Proveedor Tipo COMPRADO

Observaciones

Aceptar Cancelar

Ilustración 60 Interfaz Modificar Artículo – Segundo Sprint

✓ Eliminar Artículo:

Eliminar Artículo

Datos Generales UM Implícitas Categorías Características Motos Producción

Código 804 ☐ Servicio

Descripción ARROZ CHAUFA

Código Alterno

Unid. Medida PTO PLATO Estado A ACTIVO

Clase 005 PLATOS SubClase 001 SUBCLASEA

Peso (Kg.) .00000 ☒ Afecto IGV

Marca

Almacenamiento STOCK Código IGV IG1 IGV 18%

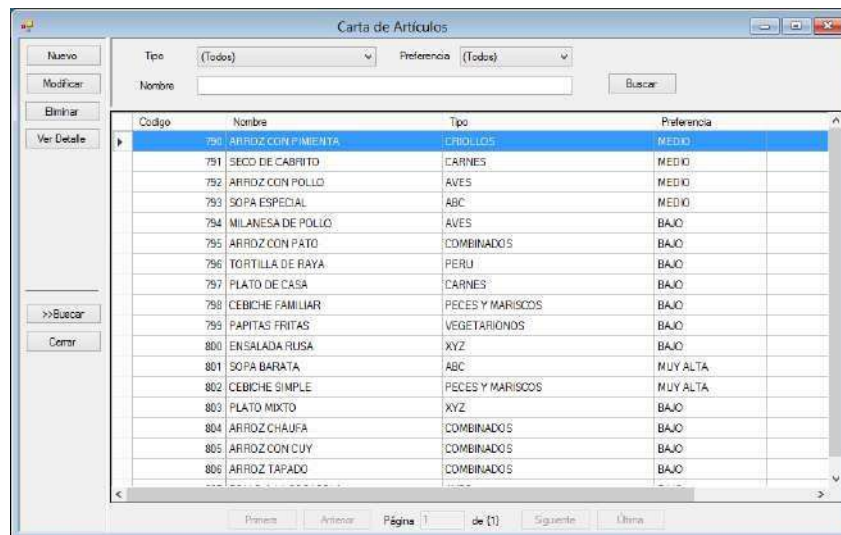
Proveedor Tipo COMPRADO

Observaciones

Eliminar Cancelar

Ilustración 61 Interfaz Eliminar Artículo – Segundo Sprint

✓ Registrar Carta:



The 'Nuevo Artículo en Carta' form has a title bar with a close button. It contains three input fields: 'Código' (a text box with a dropdown arrow), 'Tipo' (a dropdown menu), and 'Preferencia' (a dropdown menu). At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Ilustración 62 Interfaz Registrar Carta – Segundo Sprint

✓ Modificar Carta:

The 'Modificar Artículo de Carta' form has a title bar with a close button. It contains three input fields: 'Código' (a text box with the value '790' and a dropdown arrow), 'Tipo' (a dropdown menu with the value 'CRIOLLOS'), and 'Preferencia' (a dropdown menu with the value 'MEDIO'). To the right of the 'Código' field, the text 'ARROZ CON PIMIENTA' is displayed. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Ilustración 63 Interfaz Modificar Carta – Segundo Sprint

✓ Eliminar Carta:

The dialog box titled "Eliminar Artículo de Carta" contains the following fields and controls:

- Código:** A text box with the value "790" and a dropdown arrow.
- Nombre:** A text box with the value "ARROZ CON PIMIENTA".
- Tipo:** A dropdown menu with the selected value "CRIOLLOS".
- Preferencia:** A dropdown menu with the selected value "MEDIO".
- Buttons:** "Eliminar" and "Cancelar" buttons at the bottom right.

Ilustración 64 Interfaz Eliminar Carta – Segundo Sprint

✓ Registrar Reservación Mesa:

The "Reservas de Mesa" window displays a list of reservations with the following details:

- Left Sidebar:** Buttons for "Nuevo", "Modificar", "Eliminar", "Ver Detalle", ">>Buscar", and "Cerrar".
- Search Fields:** "Cliente" (text box), "Desde Fecha" (07/04/2016), "Hasta Fecha" (28/04/2016), and "Estado" (Todos).
- Table:** A table with columns "Codigo", "Ambiente", "Mesa N°", and "Cliente". One row is visible: Codigo 4, Ambiente SALA COMUN, Mesa N° 9, Cliente MALCA ZENA NELLY.
- Footer:** Navigation buttons: "Primera", "Anterior", "Página 1 de (1)", "Siguiente", and "Última".

The "Nueva Reservación" dialog box contains the following fields and controls:

- Código:** An empty text box.
- Cliente:** A text box with the value "326" and a dropdown arrow.
- Ambiente:** A dropdown menu with the selected value "SALA COMUN".
- Mesa:** A dropdown menu with the selected value "MESA N° 15 (6 ASIENTOS / LIBRE)".
- Fecha y Hora:** A text box with the value "jueves 07/04/2016 08:57" and a calendar icon.
- Estado:** A dropdown menu with the selected value "PENDIENTE".
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

Ilustración 65 Interfaz Registrar Reservación Mesa – Segundo Sprint

✓ Modificar Reservación Mesa:



The screenshot shows a dialog box titled "Modificar Reservación" with a close button (X) in the top right corner. The dialog contains several input fields and dropdown menus:

- Codigo:** A text box containing the number "4".
- Cliente:** A text box containing "87", followed by an ellipsis button and the text "MALCA ZEÑA, NELLY .".
- Ambiente:** A dropdown menu with "SALA COMUN" selected.
- Mesa:** A dropdown menu with "MESA N° 9 (6 ASIENTOS / LIBRE)" selected.
- Fecha y Hora:** A text box containing "sábado 23/04/2016 09:00" and a calendar icon.
- Estado:** A dropdown menu with "PENDIENTE" selected.

At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Ilustración 66 Interfaz Modificar Reservación Mesa – Segundo Sprint

✓ Eliminar Reservación Mesa:



The screenshot shows a dialog box titled "Eliminar Reservación" with a close button (X) in the top right corner. The dialog contains the same input fields and dropdown menus as the previous one:

- Codigo:** A text box containing the number "4".
- Cliente:** A text box containing "87", followed by an ellipsis button and the text "MALCA ZEÑA, NELLY .".
- Ambiente:** A dropdown menu with "SALA COMUN" selected.
- Mesa:** A dropdown menu with "MESA N° 9 (6 ASIENTOS / LIBRE)" selected.
- Fecha y Hora:** A text box containing "sábado 23/04/2016 09:00" and a calendar icon.
- Estado:** A dropdown menu with "PENDIENTE" selected.

At the bottom right, there are two buttons: "Eliminar" and "Cancelar".

Ilustración 67 Interfaz Eliminar Reservación Mesa – Segundo Sprint

4.6.2.7. GRAFICANDO EL TABLÓN DE TAREAS

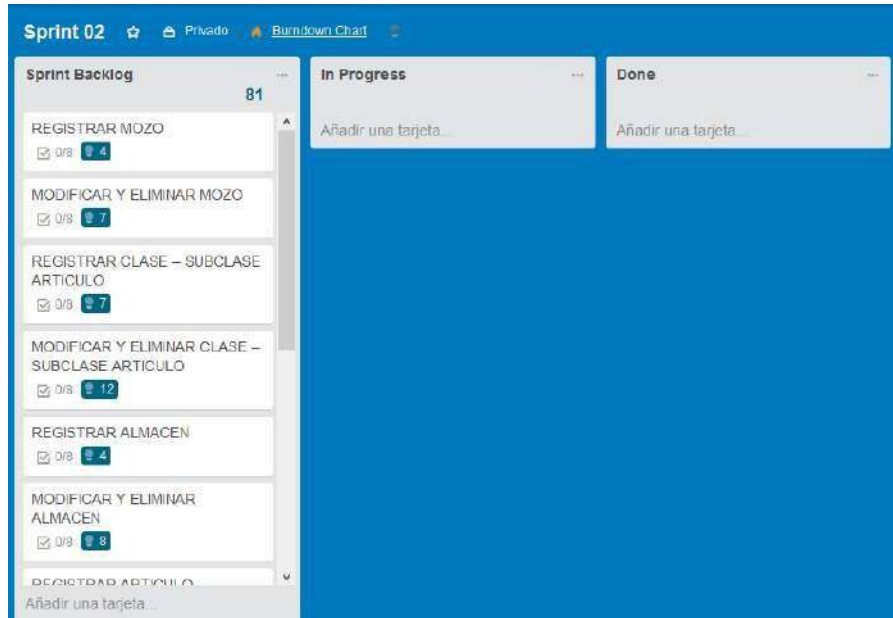


Ilustración 68 Historias de Usuario –Segundo Sprint

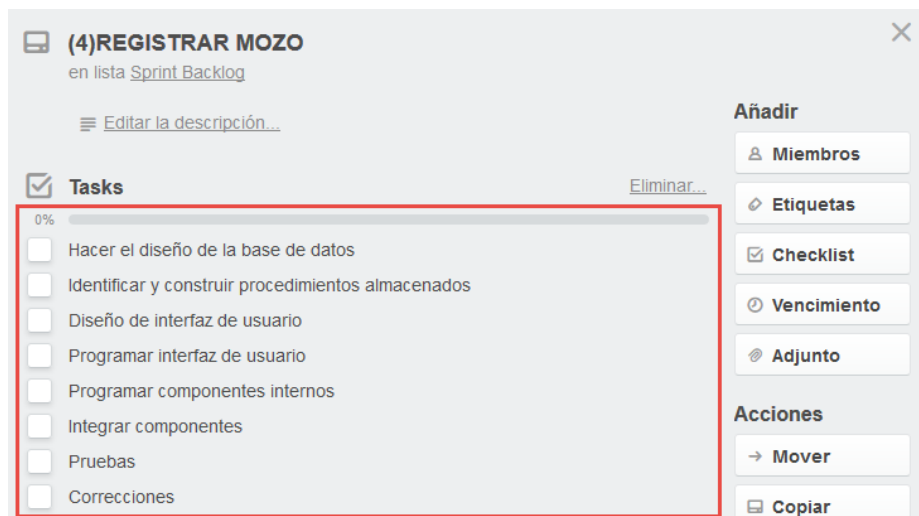


Ilustración 69 H. de Usuario Registrar Mozo

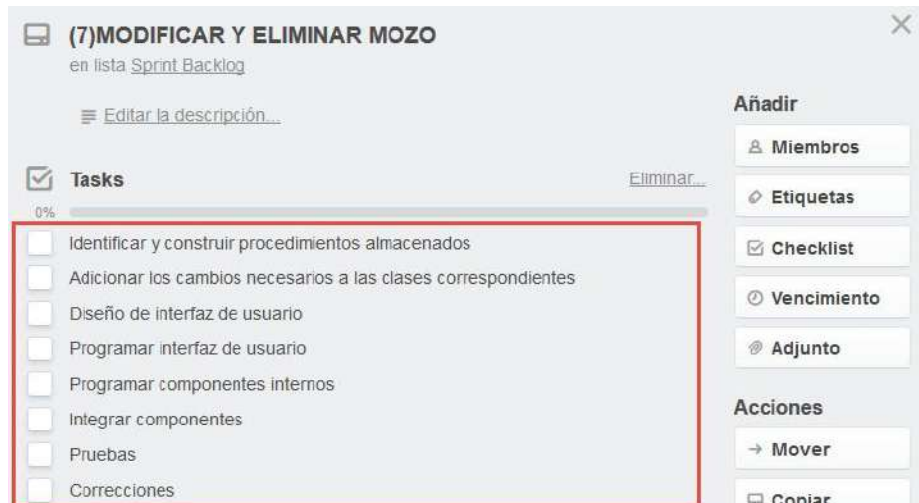


Ilustración 70 H. de Usuario Modificar y Eliminar Mozo

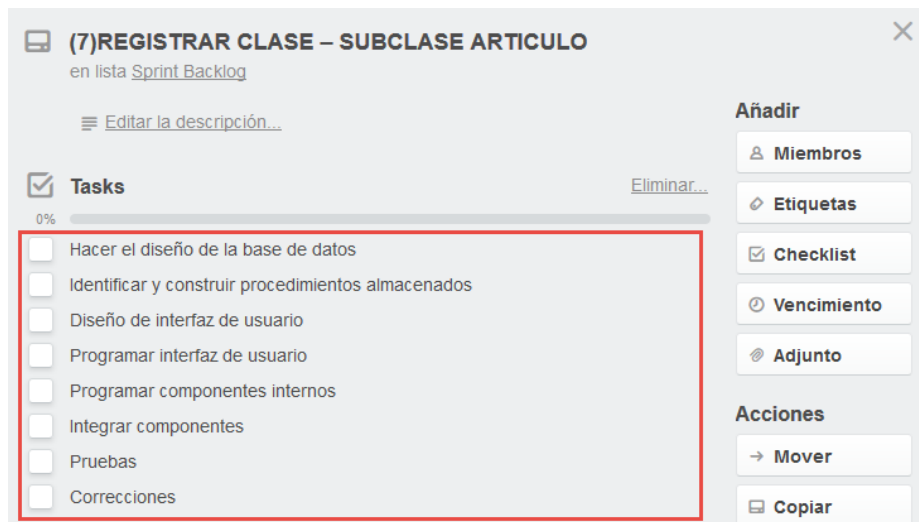


Ilustración 71 H. de Usuario Registrar Clase – SubClase Articulo

(12)MODIFICAR Y ELIMINAR CLASE – SUBCLASE ARTICULO

en lista [Sprint Backlog](#)

Editar la descripción...

✓

Tasks

Eliminar...

0%

☐ Identificar y construir procedimientos almacenados
☐ Adicionar los cambios necesarios a las clases correspondientes
☐ Diseño de interfaz de usuario
☐ Programar interfaz de usuario
☐ Programar componentes internos
☐ Integrar componentes
☐ Pruebas
☐ Correcciones

Añadir

Miembros

Etiquetas

✓

Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 72 H. de Usuario Modificar y Eliminar Clase – Subclase Artículo

(4)REGISTRAR ALMACEN

en lista [Sprint Backlog](#)

Editar la descripción...

✓

Tasks

Eliminar...

0%

☐ Hacer el diseño de la base de datos
☐ Identificar y construir procedimientos almacenados
☐ Diseño de interfaz de usuario
☐ Programar interfaz de usuario
☐ Programar componentes internos
☐ Integrar componentes
☐ Pruebas
☐ Correcciones

Añadir

Miembros

Etiquetas

✓

Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 73 H. de Usuario Registrar Almacén

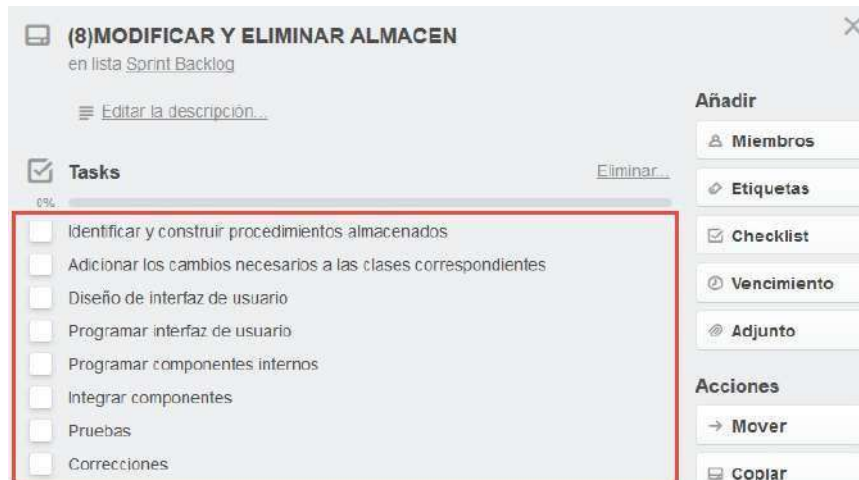


Ilustración 74 H. de Usuario Modificar y Eliminar Almacén

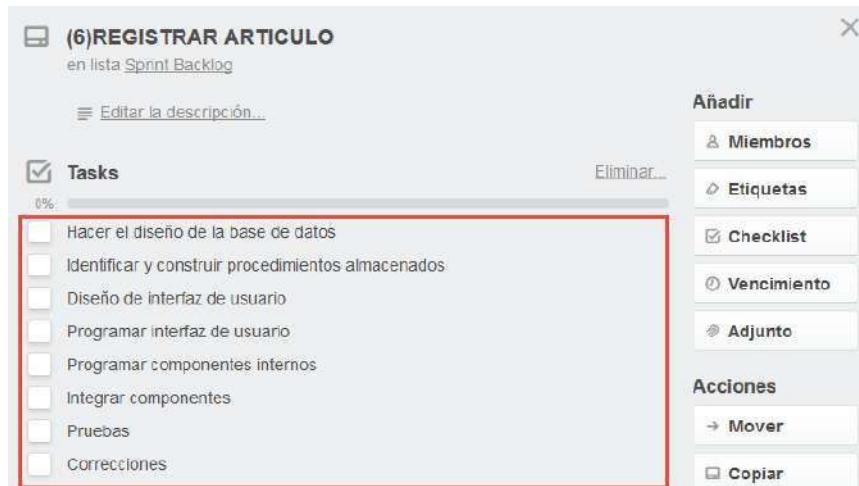


Ilustración 75 H. de Usuario Registrar Artículo

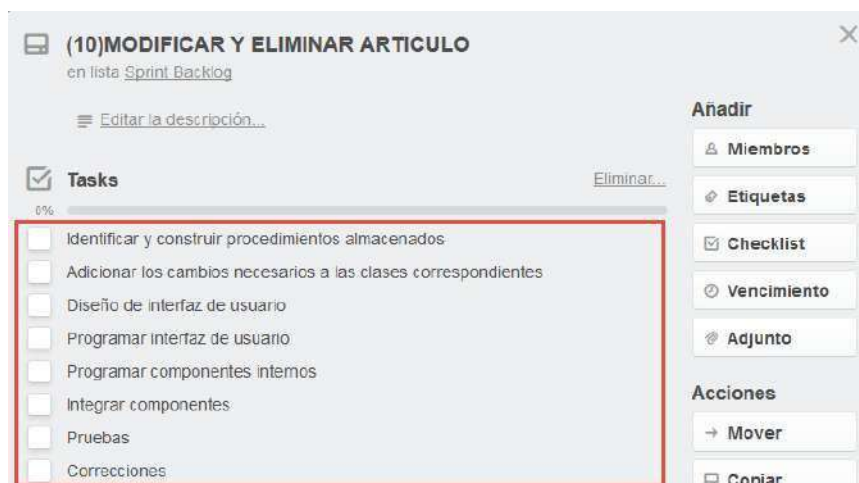


Ilustración 76 H. de Usuario Modificar y Eliminar Artículo

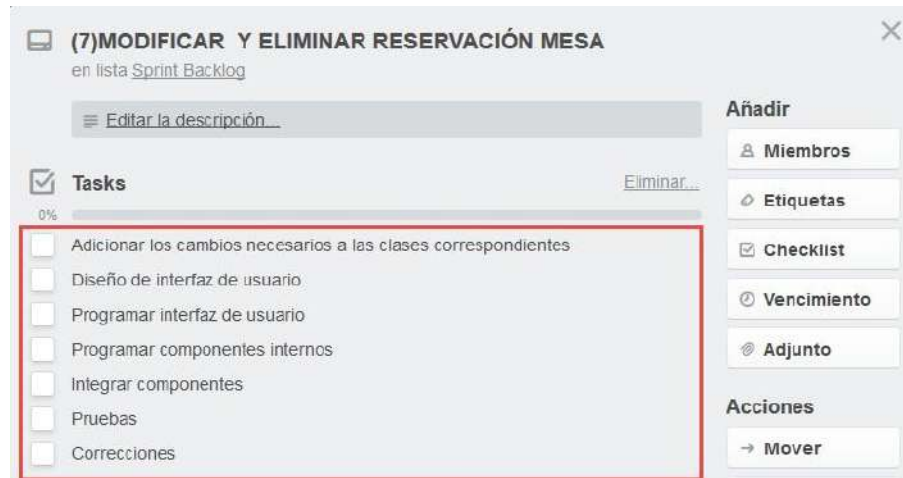


Ilustración 80 H. de Usuario Modificar y Eliminar Reservación Mesa

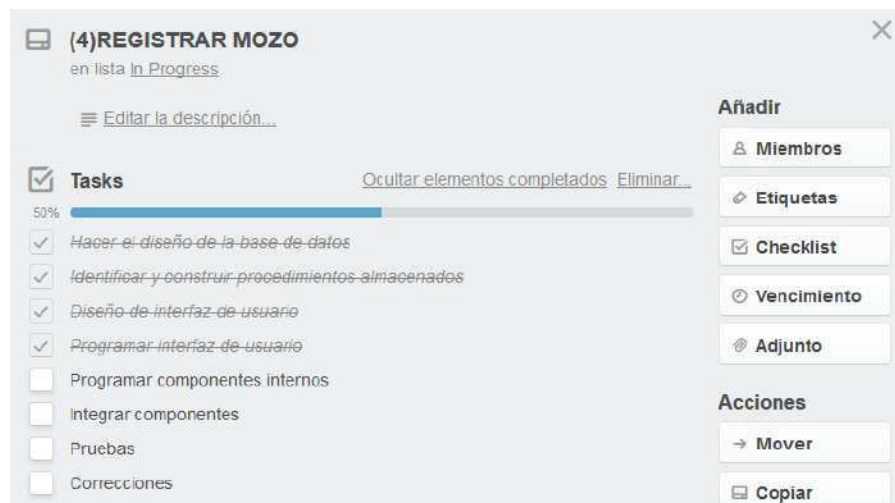
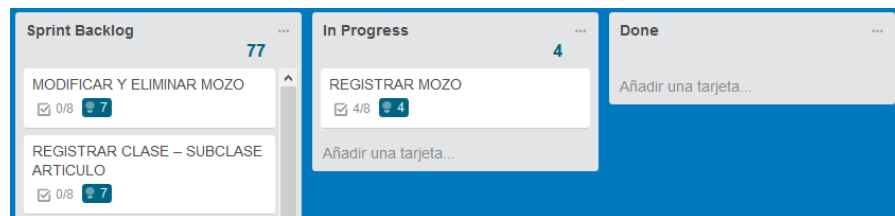


Ilustración 81 Proceso de Tareas Historia 1 – Segundo Sprint

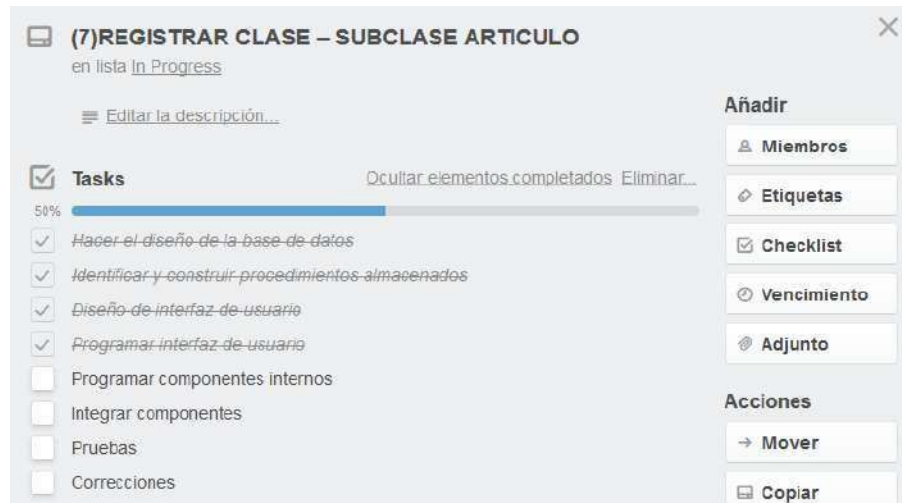
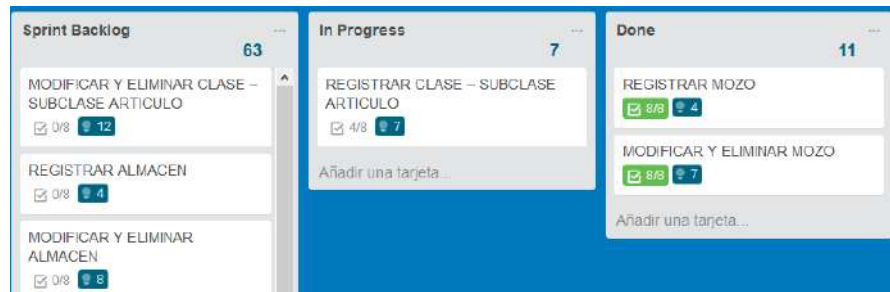
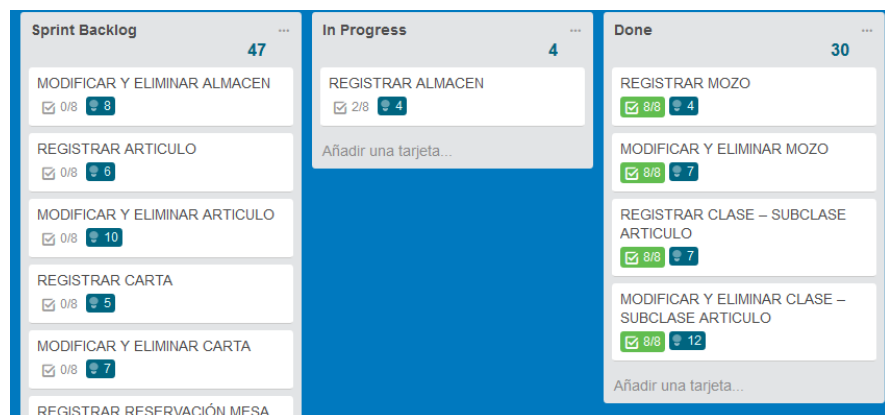


Ilustración 82 Proceso de Tareas Historia 3 – Segundo Sprint



(4)REGISTRAR ALMACEN

en lista

In Progress

Editar la descripción...

✓

Tasks

38%

✓

Hacer el diseño de la base de datos

✓

Identificar y construir procedimientos almacenados

✓

Diseño de interfaz de usuario

Programar interfaz de usuario

Programar componentes internos

Integrar componentes

Pruebas

Correcciones

Ocultar elementos completados

Eliminar...

Añadir

Miembros

Etiquetas

Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 83 Proceso de Tareas Historia 5 – Segundo Sprint

Sprint Backlog

33

In Progress

6

Done

42

MODIFICAR Y ELIMINAR ARTICULO

0/8 10

REGISTRAR CARTA

0/8 5

MODIFICAR Y ELIMINAR CARTA

0/8 7

REGISTRAR RESERVACIÓN MESA

0/8 4

MODIFICAR Y ELIMINAR RESERVACIÓN MESA

0/7 7

Añadir una tarjeta...

REGISTRAR ARTICULO

5/8 6

Añadir una tarjeta...

REGISTRAR MOZO

8/8 4

MODIFICAR Y ELIMINAR MOZO

8/8 7

REGISTRAR CLASE – SUBCLASE ARTICULO

8/8 7

MODIFICAR Y ELIMINAR CLASE – SUBCLASE ARTICULO

8/8 12

REGISTRAR ALMACEN

8/8 4

MODIFICAR Y ELIMINAR ALMACEN

8/8 8

Añadir una tarjeta...

(6)REGISTRAR ARTICULO

en lista

In Progress

Editar la descripción...

✓

Tasks

63%

✓

Hacer el diseño de la base de datos

✓

Identificar y construir procedimientos almacenados

✓

Diseño de interfaz de usuario

✓

Programar interfaz de usuario

✓

Programar componentes internos

Integrar componentes

Pruebas

Correcciones

Ocultar elementos completados

Eliminar...

Añadir

Miembros

Etiquetas

Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 84 Proceso de Tareas Historia 7 – Segundo Sprint

197

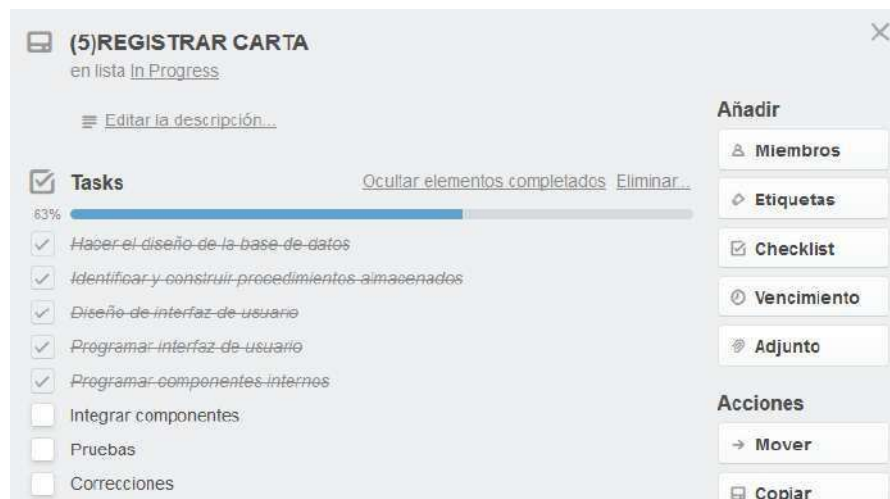
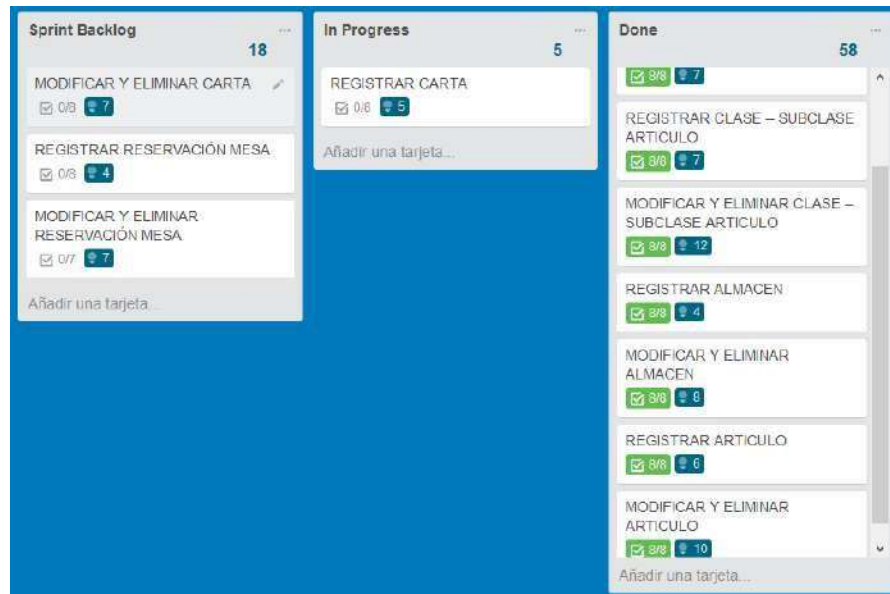


Ilustración 85 Proceso de Tareas Historia 9 – Segundo Sprint

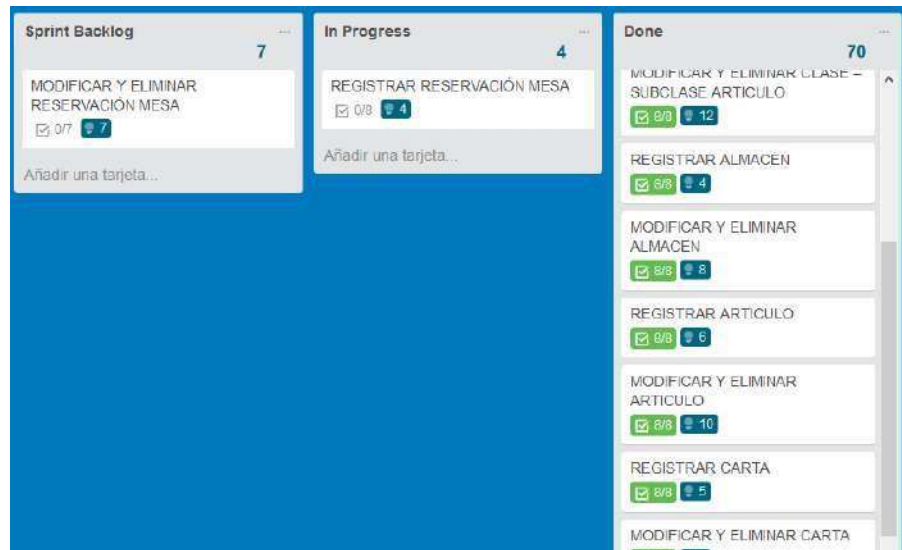


Ilustración 86 Proceso de Tareas Historia 11 – Segundo Sprint

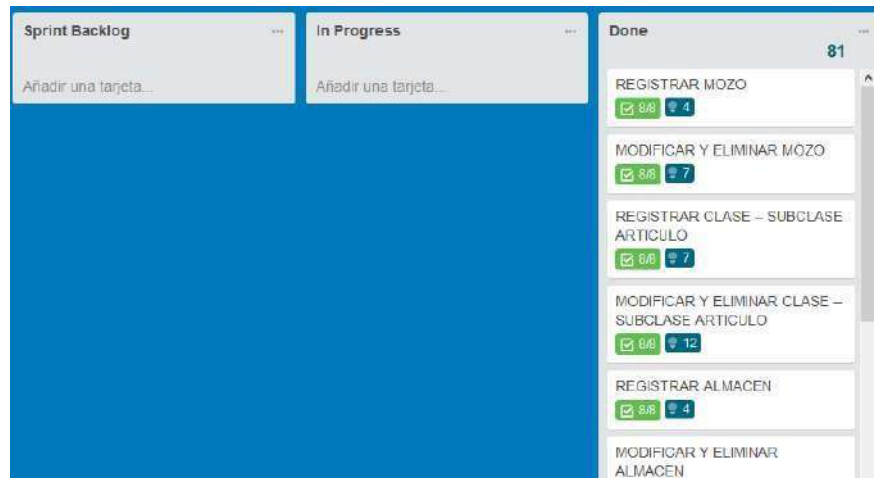


Ilustración 87 Historias de Usuario Concluidas –Segundo Sprint

4.6.28. DIAGRAMA DE BURN DOWN

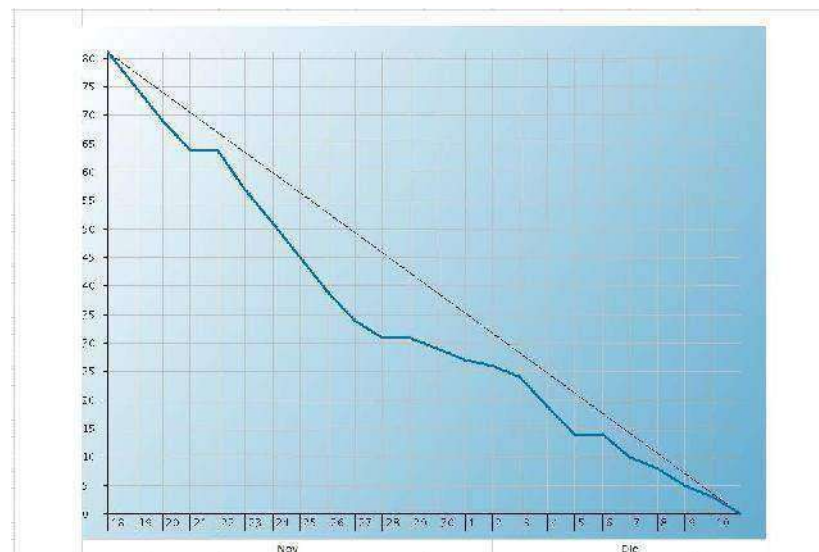


Ilustración 88 Diagrama de Burn Down –Segundo Sprint

4.6.2.9. DESARROLLO DE HISTORIAS

Registrar Artículo

Presentacion

```

Private Sub AceptarNuevo()
    Dim strerror As String
    Try
        strerror = Me.ValidarDatosEntrada
        If strerror <> "" Then
            MessageBox.Show("Verifique Errores en el Ingreso de
                              Datos", "Validación", MessageBoxButtons.OK,
                              MessageBoxIcon.Exclamation)
        Else
            PrepararRegistro()
            Me.LlamarAdicionar(dtArticulo.Rows(0))
            LimpiarPantalla()
            Me.mGrabadoOK = True
            txtDescripcion.Focus()
            Me.Close()
        End If
    Catch ex As Exception
        GestorError.MensajeError()
    End Try
End Sub

Private Sub PrepararRegistro()
    Dim drFila As DataRow
    dtArticulo.Clear()
    drFila = dtArticulo.NewRow
    With drFila
        .Item("codart") = Val(txtCodigo.Text)
        .Item("Descripcion") = txtDescripcion.Text
        .Item("um") = txtUM.Text
    End With
End Sub

```

```

        .Item("estado") = txtEstado.Text
        .Item("perecible") = chkPerecible.Checked
        .Item("codclase") = txtClase.Text
        .Item("codsubclase") = txtSubClase.Text
        .Item("afectoigv") = chkAfectoIgv.Checked
        .Item("codigv") = txtCodigoIgv.Text
        .Item("peso") = Val(txtPeso.Text)
        .Item("observacion") = txtObservacion.Text
        .Item("frapri") = Me.chkFraccionPrimaria.Checked
        .Item("codpro") = Val(Me.txtCodPro.Text)
        .Item("marca") = Me.cmbMarca.SelectedValue
        .Item("coduser") = Configuracion.Usuario
        .Item("servicio") = Me.chkServicio.Checked
        .Item("tipoalmacenamiento") =
            Me.cmbAlmacenamiento.SelectedValue

    End With
    dtArticulo.Rows.Add(drFila)
End Sub

Private Sub LlamarAdicionar(ByRef drFila As DataRow)
    Dim dsEnvio As New DataSet
    Dim dtTabla As New DataTable
    Dim clienteRemotoObj As ServicioRemotoAlmacenClient = Nothing
    dtTabla = drFila.Table.Clone
    dtTabla.LoadDataRow(drFila.ItemArray, True)
    dsEnvio.Tables.Add(dtTabla)
    If Configuracion.UsarServidorAplicaciones Then
        Try
            clienteRemotoObj = Proxy.Instanciar(TipoServicio.Almacen)
            drFila = clienteRemotoObj.AdicionarArticulo(dsEnvio,
                Configuracion.CodCia,
                Configuracion.CadenaConexion).Tables(0).Rows(0)
        Catch ex As Exception
            Throw ex
        Finally
            clienteRemotoObj.Close()
        End Try
    End If
End Sub

```

(WFC) Servicio Remoto Almacen

```

Public Function AdicionarArticulo(ByVal dsArticulo As DataSet,
    ByVal codCia As String, ByVal cadenaConexion As String) As
    System.Data.DataSet Implements
    ContratoSistRestaurante.IServicioRemotoAlmacen.AdicionarArticulo

    Dim articuloObj As Articulo
    Dim drEnvio As DataRow
    Dim dtTabla As New DataTable
    Try
        cadenaConexion =
            Configuracion.CompletarCredenciales(cadenaConexion)
        articuloObj = New Articulo(cadenaConexion)
    End Try

```

```

        drEnvio = dsArticulo.Tables(0).Rows(0)
        articuloObj.Adicionar(drEnvio, codCia)
    Catch ex As Exception
        Throw ex
    End Try
    Return dsArticulo
End Function

```

Logico de Negocio

```

Public Sub Adicionar(ByRef drFila As DataRow, ByVal codCia As
    String)
    Dim artObj As New ArticuloAD(mCadenaConexion)
    Dim corrObj As New CorrelativoAD(mCadenaConexion)
    Try
        Me.IniciarTransaccion()
        artObj.RecibirTransaccion(Me.Transaccion, Me.Conexion)
        corrObj.RecibirTransaccion(Me.Transaccion, Me.Conexion)
        drFila.Item("codart") = corrObj.ObtenerCorrelativo("AL",
            "001")
        artObj.Adicionar(drFila, codCia)
        Me.AceptarTransaccion()
    Catch ex As Exception
        Me.CancelarTransaccion()
        Throw (ex)
    End Try
End Sub

```

Acceso a Datos

```

Public Function Adicionar(ByVal drFila As DataRow, ByVal codCia As
    String) As Integer
    Me.LimpiarParametros()
    LlenarParametros(drFila, codCia)

    Return Me.MantenimientoBD("INSERT INTO ALART VALUES (@codart,
        @descripcion, @um, @estado, @perecible, @codclase,
        @codsubclase, @afectoigv, @codigv, @peso," _
        & "@observacion, @frapri, @codpro, @marca, @servicio,
        @coduser, @fecact,@codcia,@tipoalmacenamiento,@url)", True)
End Function

```

4.6.3. SPRINT 3

4.6.3.1. CALCULO DE LA VELOCIDAD ESTIMADA PARA EL TERCER SPRINT

- ✓ Calculando duración del sprint:
2 semanas = 10 días laborables.
- ✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	10
Programador 2	9
Total	19 días hombres disponible

Tabla 1. Calculo Días Hombre -Tercer Sprint

- ✓ Calculando factor de dedicación (FD):
 $FD = \text{Puntos de historia (último sprint)} / \text{días hombres disponible}$
 $FD = 82/19 = 4.3$
- ✓ Velocidad estimada: $FD \times \text{Días –Hombre- Disponible}$
 $19 * 4.3 = 81.7$

4.6.3.2. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL TERCER SPRINT:

Nº	SPRINT 1	PUNTO S	ESTADO
1	REGISTRAR CLIENTES A	4	Terminado
2	MODIFICAR Y ELIMINAR CLIENTES	8	Terminado
3	REGISTRAR PEDIDO	10	Terminado
4	MODIFICAR PEDIDO	10	Terminado
5	ELIMINAR PEDIDO	6	Terminado

6	ANULAR PEDIDO	6	Terminado
7	REGISTRAR COMPROBANTE VENTA	10	Terminado
8	MODIFICAR COMPROBANTE VENTA	10	Terminado
9	ELIMINAR COMPROBANTE VENTA	9	Terminado
*	TOTAL DE PUNTOS DE HISTORIA	73	73
Objetivo	<ul style="list-style-type: none"> • Registrar Clientes • Registrar un pedido de forma manual o uno realizado por el mozo de forma automática desde el dispositivo móvil. • Generar y registrar la facturación ya sea Factura o Boleta. • Realizar la impresión una vez generado la facturación. 		

Tabla 38. Productos en el Tercer Sprint

4.6.3.3. DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre de Historia: Registrar Clientes	
Prioridad en Negocio: Baja	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá la creación de un nuevo cliente.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Ventas/ Gestión de Clientes. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del cliente. ✓ Dar clic en el botón Aceptar para confirmar la creación del cliente. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero

Integrar componentes	Julio Alex Ventura Chero José Human Valqui
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 39. 1° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador
Nombre de Historia: Modificar y Eliminar cliente	
Prioridad en Negocio: Media	
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente al cliente, esta opción te permitirá modificar los datos del cliente que sea conveniente, así como también eliminar al cliente de ser necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un cliente y dar clic en el botón modificar u eliminar, cargará otra ventana con los datos del cliente y permitirá la actualización o eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	José Human Valqui

Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 40. 2° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Caja / Mozo
Nombre de Historia: Registrar Pedido	
Prioridad en Negocio: Alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema te permitirá registrar un pedido de forma manual por caja, así como también de forma automática realizada por el mozo desde el dispositivo móvil.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Ventas /Pedido. <p>Para el registro del pedido manual</p> <ul style="list-style-type: none"> ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del pedido. 	

✓ Dar clic en el botón Aceptar para confirmar la creación del pedido. Para el registro automático ✓ El mozo al realizar el registro del pedido desde su dispositivo móvil, se mostrará en la gestión de pedido, pero no se podrá modificar a excepción que el usuario tenga acceso.	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 41. 3° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Caja
Nombre de Historia: Modificar Pedido	
Prioridad en Negocio: Alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	

<p>Descripción:</p> <p>Habiendo registrado previamente un pedido, esta opción te permitirá modificar los datos del pedido que se ha realizado de forma manual.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un pedido y dar clic en el botón modificar, cargará otra ventana con los datos del pedido y permitirá la actualización, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 42. 4° Historia de Usuario – Tercer Sprint

HISTORIA DE USUARIO	
Número: 5	Usuario: Caja
Nombre de Historia: Eliminar Pedido	
Prioridad en Negocio: Media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente un pedido, esta opción te permitirá eliminar el pedido seleccionado que se ha realizado de forma manual.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un pedido y dar clic en el botón eliminar, cargará otra ventana con los datos del pedido y permitirá la eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero José Human Valqui

Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 43. 5° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 6	Usuario: Caja
Nombre de Historia: Anular Pedido	
Prioridad en Negocio: Media	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Habiendo registrado previamente un pedido, esta opción te permitirá anular el registro seleccionado si es necesario.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un pedido y dar clic en el botón anular, cargará otra ventana con los datos del pedido y permitirá la anulación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero

Programar interfaz de usuario	José Human Valqui
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 44. 6° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 7	Usuario: Caja
Nombre de Historia: Registrar Comprobante de Venta	
Prioridad en Negocio: Alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
<p>Descripción:</p> <p>Esta opción del sistema permitirá el registro del comprobante de venta ya sea Boleta o Factura, en donde una vez generada se imprimirá de forma automática.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Ingresar a la opción de menú Ventas / Comprobante de Venta. ✓ Dar clic en el botón nuevo. ✓ Llenar los datos solicitados para el registro correcto del comprobante. ✓ Dar clic en el botón Aceptar para confirmar la creación del comprobante de venta. 	

Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 45. 7° Historia de Usuario - Tercer Sprint

HISTORIA DE USUARIO	
Número: 8	Usuario: Caja
Nombre de Historia: Modificar Comprobante de Venta	
Prioridad en Negocio: Alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Habiendo registrado previamente un comprobante de venta, esta opción te permitirá modificar los datos del comprobante que sean conveniente.	

Pruebas:	
✓ Al seleccionar un comprobante y dar clic en el botón modificar, cargará otra ventana con los datos del comprobante y permitirá la actualización, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar.	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	Julio Alex Ventura Chero José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Human Valqui
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 46. 8° Historia de Usuario – Tercer Sprint

HISTORIA DE USUARIO	
Número: 9	Usuario: Caja
Nombre de Historia: Eliminar Comprobante de Venta	
Prioridad en Negocio: Alta	
Puntos estimados: 9	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	

<p>Descripción:</p> <p>Habiendo registrado previamente un comprobante de venta, esta opción te permitirá eliminar el comprobante que sean conveniente.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Al seleccionar un comprobante y dar clic en el botón eliminar, cargará otra ventana con los datos del comprobante y permitirá la eliminación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 47. 9° Historia de Usuario - Tercer Sprint

4.6.3.4. DIAGRAMA LOGICO Y FISICO SPRINT 3

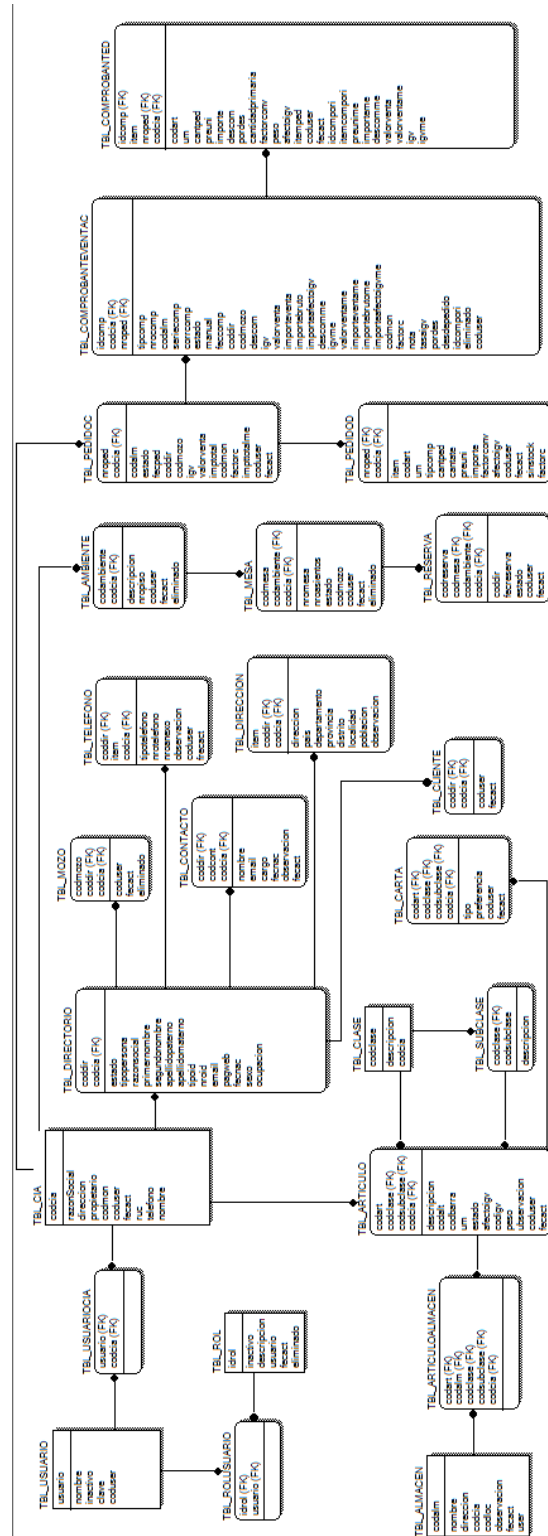
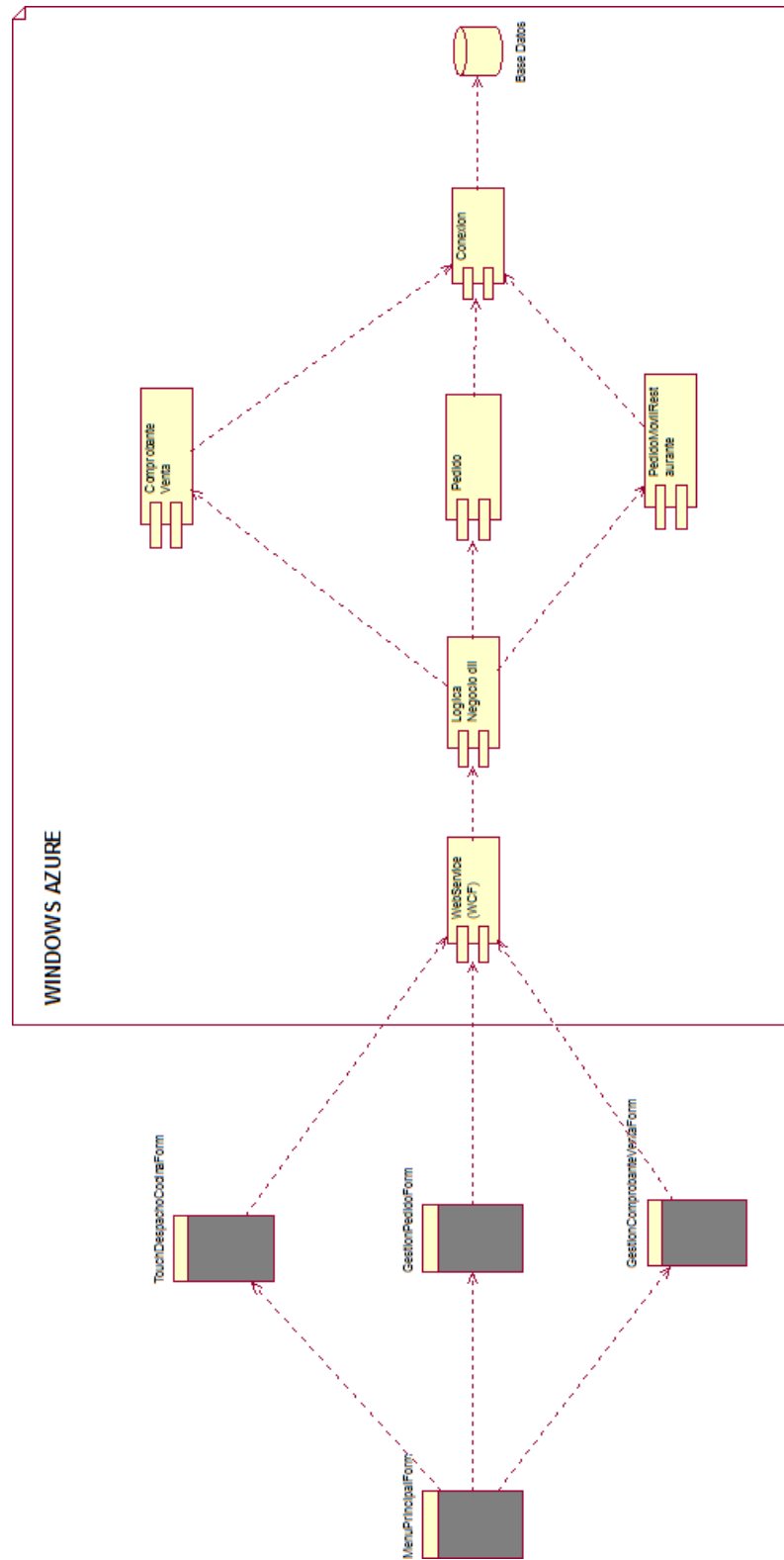


Ilustración 89 Diagrama lógico – Tercer Sprint

4.6.3.5. DIAGRAMA DE COMPONENTES SPRINT 3



4.6.3.6. INTERFAZ DE LAS HISTORIAS DE USUARIO

✓ Registrar Cliente:

The 'Clientes' window displays a list of clients with the following data:

Código	Nombre/Razón Social	Tipo Persona	RUC/DNI	Telefono
316	MANAYAY SANCHEZ CARLOS MARTIN	N	10484255054	
480	MAPFRE PERU COMPAÑIA DE SEGUROS Y R...	J	20202380621	
479	MAPFRE PERU VIDA COMPAÑIA DE SEGURO...	J	20418896915	
478	MARQUEZ AMENERO, RICARDO	N	10164511745	
411	MASIAS SAMAME MIRKA MORAIMA	N	10102751740	
412	MATUTE GROUP EIRL	J	20539065832	
413	MB CONFECCIONES Y SERVICIOS GENERALE...	J	20480557701	
414	MC ENTRETENIMIENTOS S.A.C.	J	20555396008	
415	MEAD JOHNSON NUTRITION (PERU) S.R.L.	J	20520485750	
416	MENDOZA GONZALEZ BENJAMIN FRANKLIN	N	10165721821	

The 'Nuevo Usuario' window contains the following fields and controls:

- Usuario:
- ☐ Inactivo
- Nombre:
- Clave:
- Aceptar
- Cancelar

Ilustración 91 Interfaz Registrar Cliente – Tercer Sprint

✓ Modificar Cliente:

The 'Modificar Usuario' window contains the following fields and controls:

- Usuario:
- ☐ Inactivo
- Nombre:
- Clave:
- Aceptar
- Cancelar

Ilustración 92 Interfaz Modificar Cliente – Tercer Sprint

✓ Eliminar Cliente:

Eliminar Usuario

Usuario: MCHAVESTA ☐ Inactivo

Nombre: MARIA CHAVESTA LOPEZ

Clave:

Eliminar Cancelar

Ilustración 93 Interfaz Eliminar Cliente – Tercer Sprint

✓ Registrar Pedido:

Gestion de Pedidos

Desde Fecha: 07/04/2016 Hasta Fecha: 07/04/2016 Buscar

Numero Pedido: Estado: (Todos)

Cliente: Mozo: (Todos)

Pedido	Tipo	Estado	Cliente	Fecha	Almacen	Moneda
--------	------	--------	---------	-------	---------	--------

Primera Anterior Página 0 de 0 Siguiente Ultima

Nuevo Pedido

Datos Pedido

Numero Pedido:

Factura/Boleta: FACTURA Fecha Pedido: 07/04/2016

Cliente:

Mozo:

Detalle

	Codigo	Descripcion	UM	Cantidad	Precio	%Descu...	Importe
* 1							

< >

<<Buscar Articulos... Nuevo Modificar Eliminar Importar...

Importe Bruto: Descuento: Total:

<<Otras Opciones... Limpier Aceptar Cancelar

Ilustración 94 Interfaz Registrar Pedido – Tercer Sprint

✓ **Modificar Pedido:**

Datos Pedido

Numero Pedido: 8 **PENDIENTE**

Factura/Boleta: FACTURA Fecha Pedido: 07/04/2016

Cliente: 328 AGENCIA DE PUBLICIDAD Y SERVICIOS AVEMORAM E.I.R.L.

Mozo: 001 CHICLAYO, TIENDA

Detalle

	Codigo	Descripcion	UM	Cantidad	Precio	%Descuento	Impo
▶ 1	804	ARROZ CHAUFA	PTO	2.000	17.9200	0.00	
2	805	ARROZ CON CUY	PTO	1.000	41.0400	0.00	
* 3							

<<Buscar Articulos... Nuevo Modificar Eliminar

Importe Bruto: 76.88 Descuento: 0.00 Total: 76.88

<<Otras Opciones... Limpiar Aceptar Cancelar

Ilustración 95 Interfaz Modificar Pedido – Tercer Sprint

✓ **Eliminar Pedido:**

Datos Pedido

Numero Pedido: 8 **PENDIENTE**

Factura/Boleta: FACTURA Fecha Pedido: 07/04/2016

Cliente: 328 AGENCIA DE PUBLICIDAD Y SERVICIOS AVEMORAM E.I.R.L.

Mozo: 001 CHICLAYO, TIENDA

Detalle

	Codigo	Descripcion	UM	Cantidad	Precio	%Descuento	Impo
1	804	ARROZ CHAUFA	PTO	2.000	17.9200	0.00	
▶ 2	805	ARROZ CON CUY	PTO	1.000	41.0400	0.00	

<<Buscar Articulos... Nuevo Modificar Eliminar

Importe Bruto: 76.88 Descuento: 0.00 Total: 76.88

Inicial: 0.00

<<Otras Opciones... Limpiar Eliminar Cancelar

Ilustración 96 Interfaz Eliminar Pedido – Tercer Sprint

✓ Anular Pedido:

Anular Pedido

Datos Pedido | **Datos Adicionales**

Numero Pedido: 8 PENDIENTE

Factura/Boleta: FACTURA Fecha Pedido: 07/04/2016

Cliente: 328 AGENCIA DE PUBLICIDAD Y SERVICIOS AVMORAM E.I.R.L.

Mozo: 001 CHICLAYO, TIENDA

	Codigo	Descripcion	UM	Cantidad	Precio	%Descuento	Impo
1	804	ARROZ CHAUFA	PTO	2.000	17.9200	0.00	35.8400
2	805	ARROZ CON CUY	PTO	1.000	41.0400	0.00	41.0400

Importe Bruto: 76.88 Descuento: 0.00 Total: 76.88 Inicial: 0.00

<<Buscar Articulos... Nuevo Modificar Eliminar <<Otras Opciones... Limpiar Eliminar Cancelar

Ilustración 97 Interfaz Anular Pedido – Tercer Sprint

✓ Registrar Comprobante Venta:

Gestion Comprobante por Punto de Venta

Nuevo Manual Modificar Eliminar Ver Detalle Anular Imprimir Cerrar

Desde Fecha: 07/04/2016 Hasta Fecha: 07/04/2016 Buscar

Cliente: Estado: (Todos) Mozo: (Todos)

Tipo Comprob.: (Todos)

Comprobante	Tipo	Estado	Cliente	Fecha	Alma
-------------	------	--------	---------	-------	------

Primera Anterior Página de (0) Siguiete Última

Comprobante Punto de Venta

Datos

Comprobante:

Tipo:

Fecha Comprobante: 07/04/2016

Ciente: 1 Cliente... PORTADOR

Enviar a: SIN DOMICILIO

Pagar ...

Detalle

Codigo	Descripcion	UM	Cantidad	Precio	%Descuento	Impo
1					0.00	

<<Buscar Articulos... Nuevo Modificar Eliminar

Valor Venta

Importar Descuento IGV Importe Venta

<<Otras Opciones... Observaciones ... Vista Previa

Aceptar Cancelar

Ilustración 98 Interfaz Registrar Comprobante de Venta – Tercer Sprint

✓ Modificar Comprobante de Venta:

Modificar

Datos

Comprobante: 0003 1376 EMITIDO

Tipo: FACTURA POR PUNTO DE VENTA

Fecha Comprobante: 03/12/2014

Ciente: 418 Cliente... NEGOCIOS HISPANO EIRL

Enviar a: AV. SANTA VICTORIA 444 URB. SANTA VICTORIA (FRENTE A MISTER PAN) LAMBAYEQUE CHICLAYO

Detalle

Codigo	Descripcion	UM	Cantidad	Precio	%Descuento	Impo
1	606 SAN LUIS SIN GAS 625ML	UND	30.000	1.2500		
2	14 AMARGO LA BARRA AMARGO 75ML	UND	1.000	9.0000		

<<Buscar Articulos... Nuevo Modificar Eliminar

Valor Venta 39.41

Descuento 0.00 IGV 7.09 Importe Venta 46.50

<<Otras Opciones... Observaciones ... Vista Previa

Aceptar Cancelar

Ilustración 99 Interfaz Modificar Comprobante Venta – Tercer Sprint

✓ Eliminar Comprobante Venta:

Eliminar Comprobante Venta

Datos

Comprobante: 0003 1376 **EMITIDO**

Tipo: FACTURA POR PUNTO DE VENTA Fecha Comprobante: 03/12/2014

Cliente: 418 ... Cliente... NEGOCIOS HISPANO EIRL

Enviar a: AV. SANTA VICTORIA 444 URB. SANTA VICTORIA (FRENTE A MISTER PAN) LAMBAYEQUE CHICLAYO

Detalle

	Codigo	Cod. Alterno	Descripcion	UM	Cantidad	Precio
▶ 1	606		SAN LUIS SIN GAS 625ML	UND	30.000	1.2500
2	14		AMARGO LA BARRA AMARGO 75ML	UND	1.000	9.0000

<<Buscar Articulos... Nuevo Modificar Eliminar

Valor Venta 39.41

Descuento 0.00 **IGV** 7.09 **Importe Venta** 46.50

<<Otras Opciones... Observaciones... Vista Previa Eliminar Cancelar

Ilustración 100 Interfaz Eliminar Comprobante Venta – Tercer Sprint

4.6.3.7. GRAFICANDO EL TABLÓN DE TAREAS

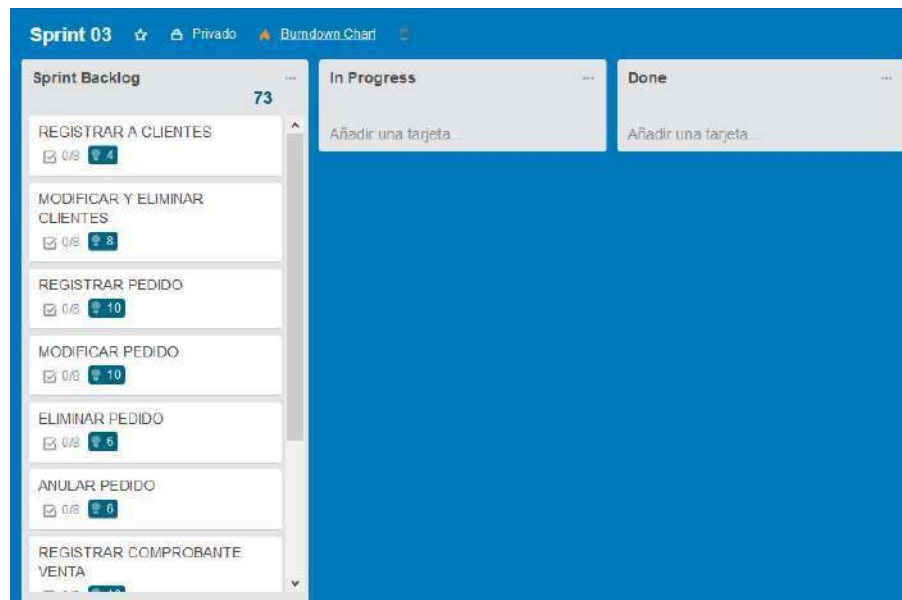


Ilustración 101 Historias de Usuario –Tercer Sprint

(10)MODIFICAR PEDIDO

en lista [Sprint Backlog](#)

Editar la descripción...

Tasks

Eliminar...

0%

☐ Identificar y construir procedimientos almacenados

☐ Adicionar los cambios necesarios a las clases correspondientes

☐ Diseño de interfaz de usuario

☐ Programar interfaz de usuario

☐ Programar componentes internos

☐ Integrar componentes

☐ Pruebas

☐ Correcciones

Añadir

Miembros

Etiquetas

☒ Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 105 H. de Usuario Modificar Pedido

(6)ELIMINAR PEDIDO

en lista [Sprint Backlog](#)

Editar la descripción...

Tasks

Eliminar...

0%

☐ Identificar y construir procedimientos almacenados

☐ Adicionar los cambios necesarios a las clases correspondientes

☐ Diseño de interfaz de usuario

☐ Programar interfaz de usuario

☐ Programar componentes internos

☐ Integrar componentes

☐ Pruebas

☐ Correcciones

Añadir

Miembros

Etiquetas

☒ Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 106 H. de Usuario Eliminar Pedido

(6)ANULAR PEDIDO

en lista [Sprint Backlog](#)

Editar la descripción...

Tasks

Eliminar...

0%

☐ Identificar y construir procedimientos almacenados

☐ Adicionar los cambios necesarios a las clases correspondientes

☐ Diseño de interfaz de usuario

☐ Programar interfaz de usuario

☐ Programar componentes internos

☐ Integrar componentes

☐ Pruebas

☐ Correcciones

Añadir

Miembros

Etiquetas

☒ Checklist

Vencimiento

Adjunto

Acciones

Mover

Copiar

Ilustración 107 H. de Usuario Anular Pedido

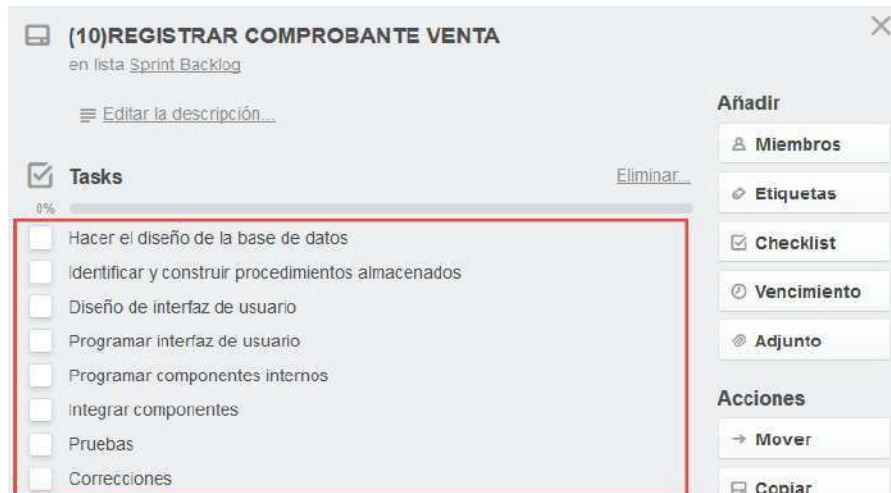


Ilustración 108 H. de Usuario Registrar Comprobante Venta

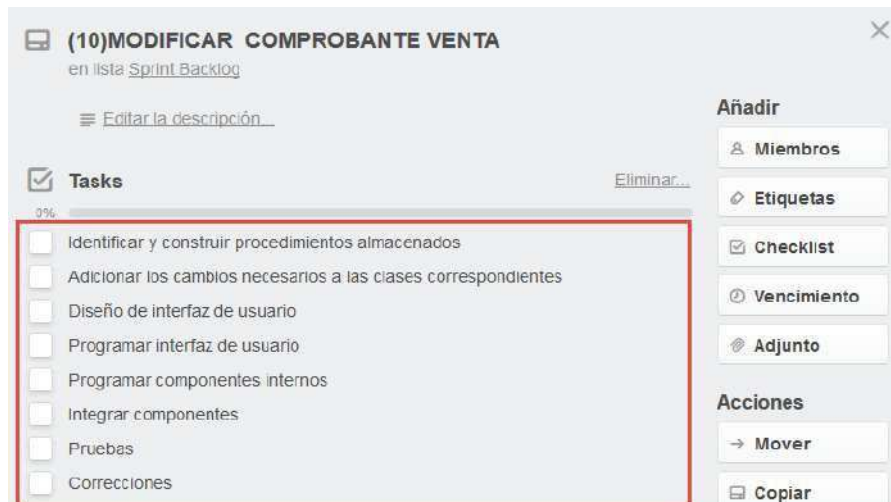


Ilustración 109 H. de Usuario Modificar Comprobante Venta

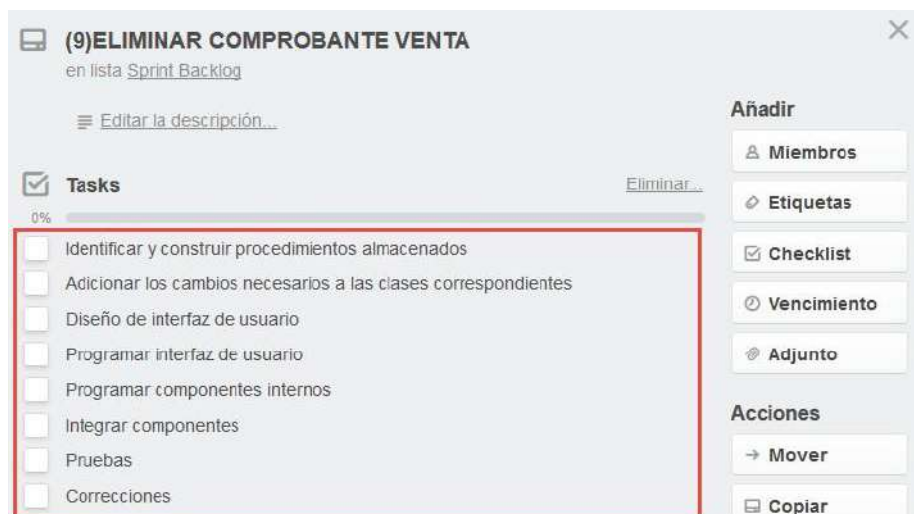


Ilustración 110 H. de Usuario Eliminar Comprobante Venta

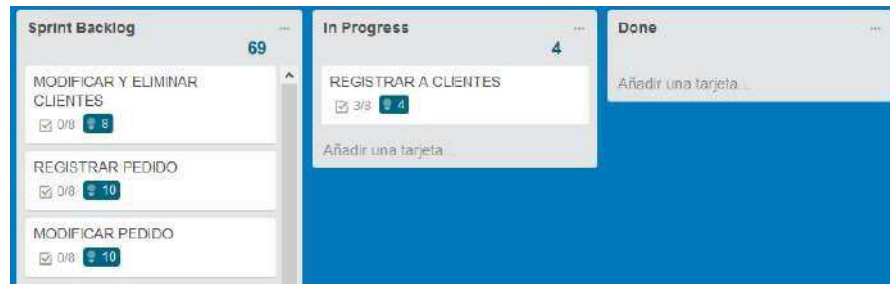
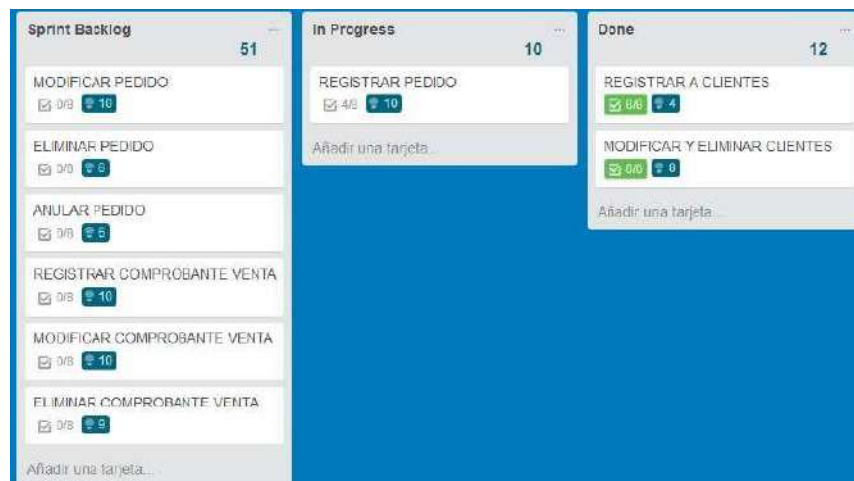


Ilustración 111 Proceso de Tareas Historia 1 – Tercer Sprint



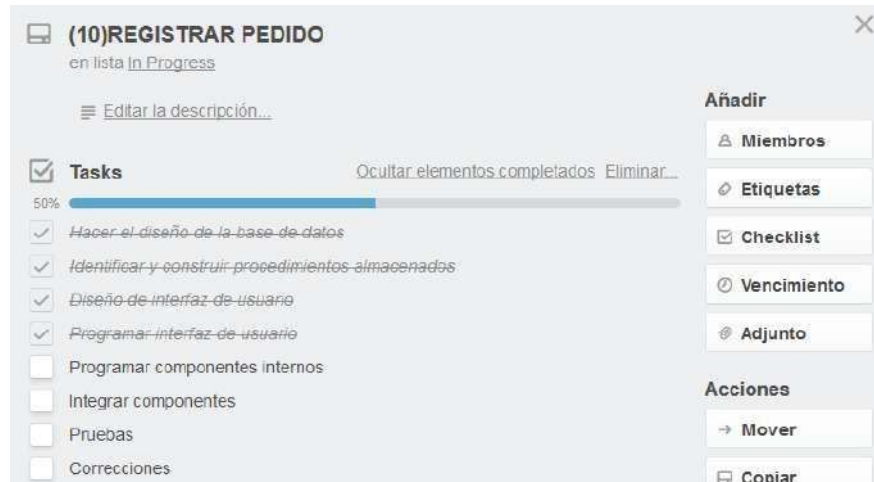


Ilustración 112 Proceso de Tareas Historia 3 – Tercer Sprint

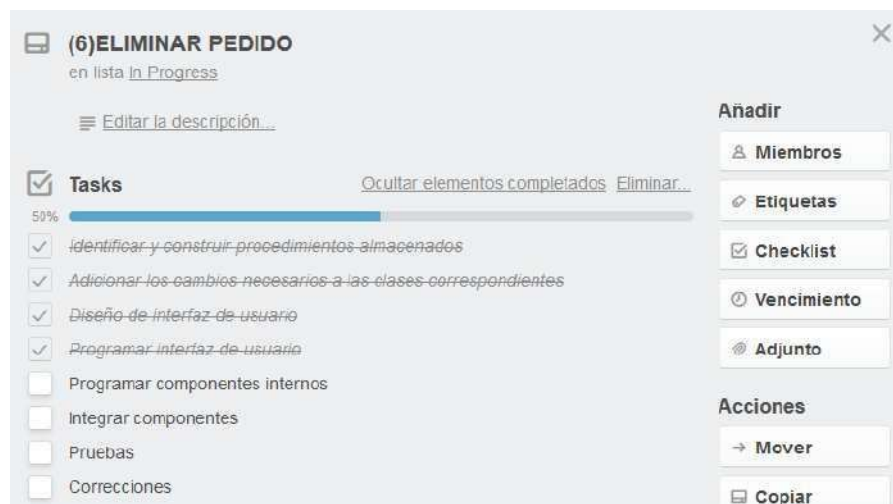
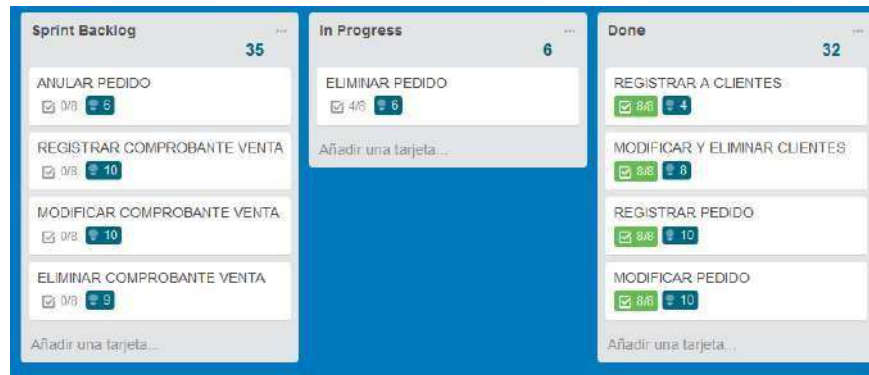


Ilustración 113 Proceso de Tareas Historia 5 – Tercer Sprint

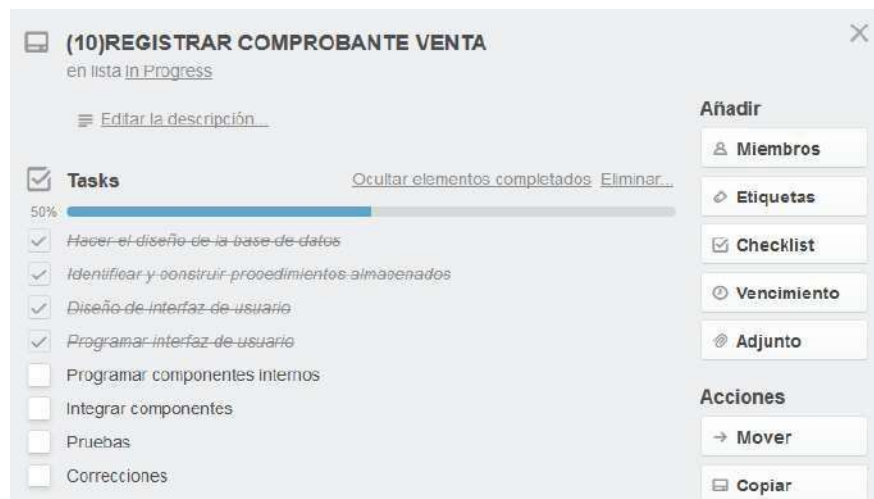
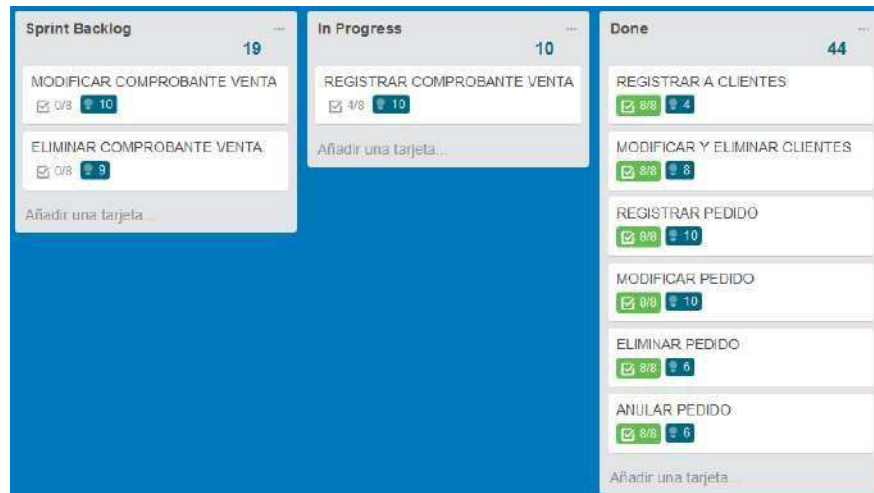
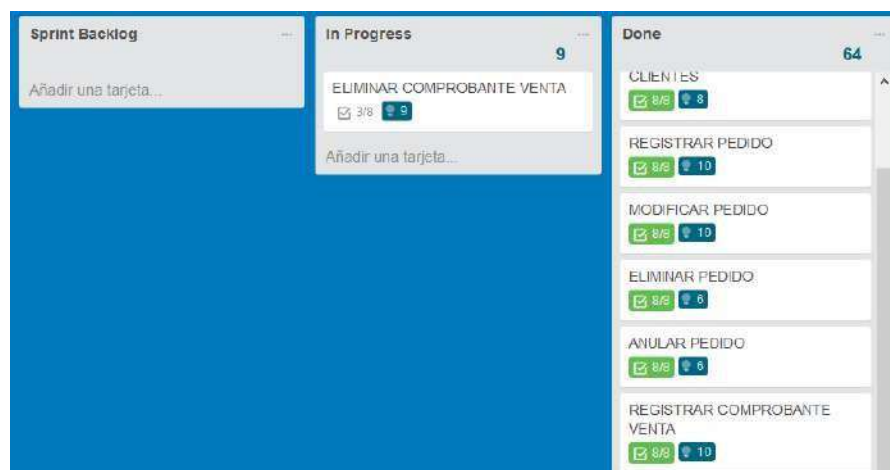


Ilustración 114 Proceso de Tareas Historia 7 – Tercer Sprint



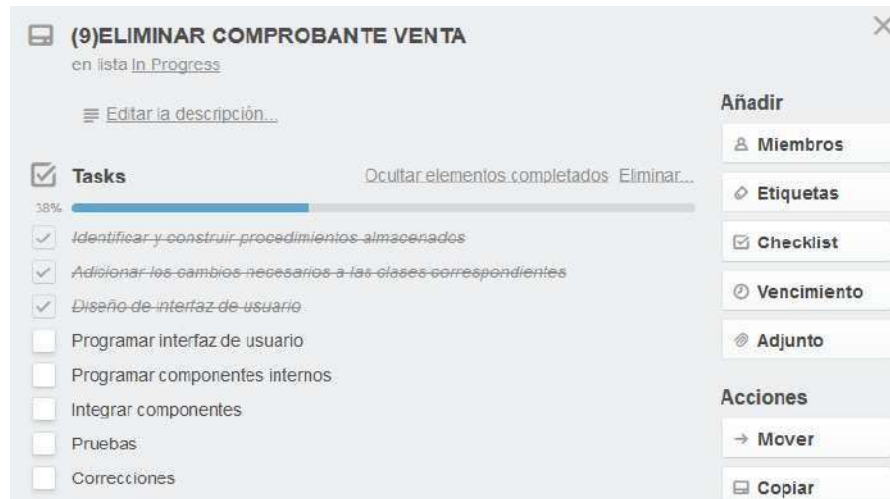


Ilustración 115 Proceso de Tareas Historia 9 – Tercer Sprint

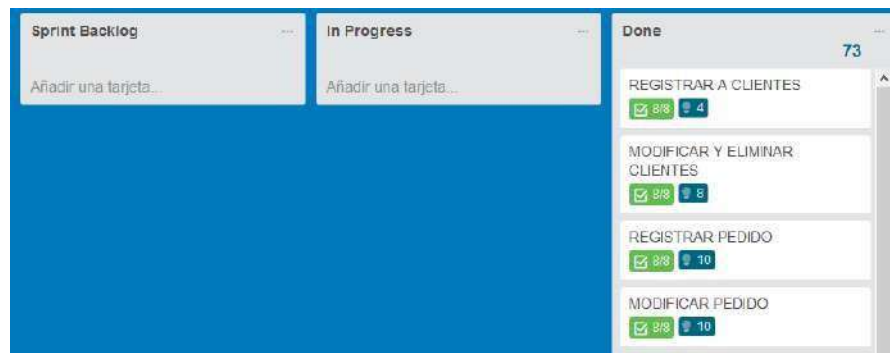


Ilustración 116 Historias de Usuario Concluidas –Tercer Sprint

4.6.3.8. DIAGRAMA DE BURN DOWN

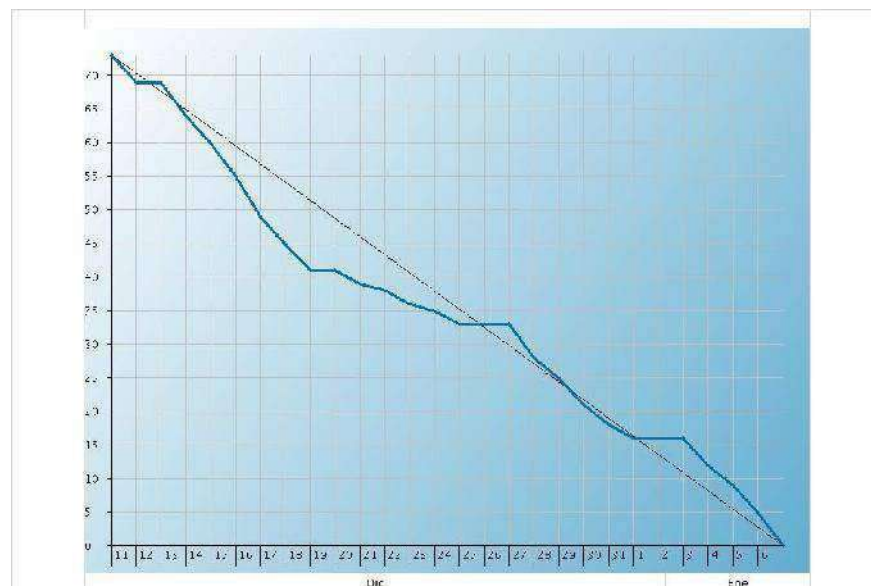


Ilustración 117 Diagrama de Burn Down –Tercer Sprint

4.6.4. SPRINT 4

4.6.4.1. CALCULO DE LA VELOCIDAD ESTIMADA PARA EL CUARTO SPRINT

✓ Calculando duración del sprint:

1 semana y 4 días= 9 días laborables.

✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	9
Programador 2	8
Total	17 días hombres disponible

Tabla 48. Calculo Días Hombre – Cuarto Sprint

✓ Calculando factor de dedicación (FD):

FD = Puntos de historia (último sprint) / días hombres disponible

$$FD = 73/17 = 4.29$$

✓ Velocidad estimada: FD X Días –Hombre- Disponible

$$17 * 4.29 = 72.93$$

4.6.4.2. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL CUARTO SPRINT:

Nº	SPRINT 1	PUNTO S	ESTADO
1	ANULAR COMPROBANTE VENTA	9	Terminado
2	DESPACHAR COCINA	10	Terminado
3	SEGUIMIENTO PEDIDOS	8	En desarrollo

*	TOTAL DE PUNTOS DE HISTORIA	27	27
Objetivo	<ul style="list-style-type: none"> • Realizar anulación de comprobante de venta • Realizar despacho de cocina. 		

Tabla 49. Productos en el cuarto Sprint

4.6.4.3. DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Caja
Nombre de Historia: Anular Comprobante de Venta	
Prioridad en Negocio: Alta	
Puntos estimados: 9	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Habiendo registrado previamente un comprobante de venta, esta opción te permitirá anular el registro si es necesario.	
Pruebas: <ul style="list-style-type: none"> ✓ Al seleccionar un comprobante de venta y dar clic en el botón anular, cargará otra ventana con los datos del comprobante y permitirá la anulación, luego aparecerá un mensaje de confirmación sobre si estamos seguros de la acción a realizar. 	

Tareas	Responsable
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Adicionar los cambios necesarios a las clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	José Human Valqui
Integrar componentes	José Human Valqui
Pruebas	José Huamán Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 50. 1° Historia de Usuario - Cuarto Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Cocina
Nombre de Historia: Despachar Cocina	
Prioridad en Negocio: Alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Esta opción permitirá realizar el despacho de cocina una vez que el producto ya se encuentre culminada su preparación	

Pruebas:	
<ul style="list-style-type: none"> ✓ Una vez que el pedido se haya realizado se mostrara en la pantalla de despacho de cocina. ✓ Seleccionar el producto(s) y dar clic en el botón Preparación cuando el producto va a iniciar su preparación, el cual cambiara de color verde ✓ Una vez que el producto haya culminado su preparación, seleccionarlo y dar clic en el botón despacho el cual se notificara mediante una alerta al mozo en su dispositivo móvil. ✓ Si por error se ha cambiado el estado de un producto dar clic en el botón Pedidos en Mesa para realizar la actualización, siempre y cuando el mozo no haya recogido el producto. 	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero José Human Valqui
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero José Human Valqui
Diseñar clases correspondientes	Julio Alex Ventura Chero
Diseño de interfaz de usuario	Julio Alex Ventura Chero
Programar interfaz de usuario	José Human Valqui
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	Julio Alex Ventura Chero
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 51. 3° Historia de Usuario - Cuarto Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Todos
Nombre de Historia: Seguimiento de Pedidos	
Prioridad en Negocio: Alta	
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Julio Alex Ventura Chero	
Descripción: Una vez que haya sido registrados los pedidos, se podrá realizar el seguimiento de los mismos mostrando su estado.	
Pruebas: ✓ Ingresar los filtros para la búsqueda de los pedidos realizados, luego dar clic en el botón buscar en donde mostrará los pedidos según los filtros seleccionados.	
Tareas	Responsable
Hacer el diseño de la base de datos	Julio Alex Ventura Chero
Identificar y construir procedimientos almacenados	Julio Alex Ventura Chero
Diseñar clases correspondientes	José Human Valqui
Diseño de interfaz de usuario	José Human Valqui
Programar interfaz de usuario	Julio Alex Ventura Chero
Programar componentes internos	Julio Alex Ventura Chero
Integrar componentes	José Human Valqui
Pruebas	José Human Valqui
Correcciones	Julio Alex Ventura Chero
Firma:	

Tabla 52. 3° Historia de Usuario - Cuarto Sprint

DIAGRAMA LOGICO Y FISICO SPRINT 4

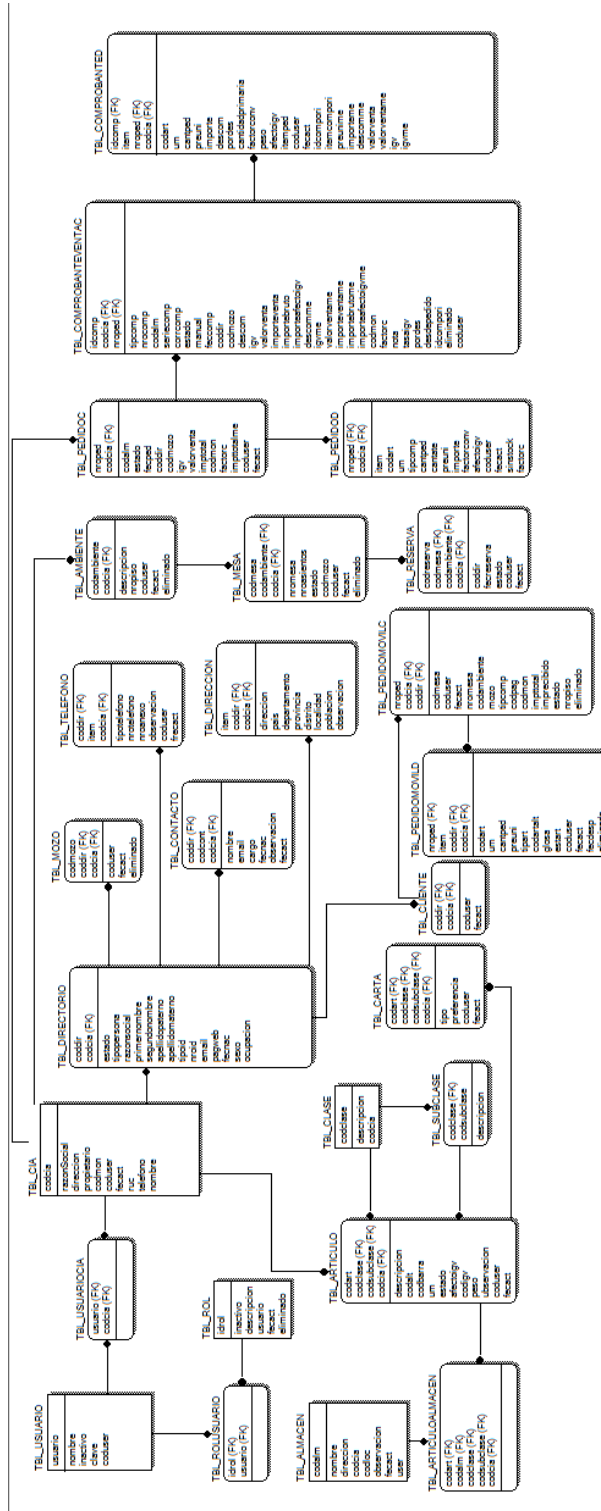
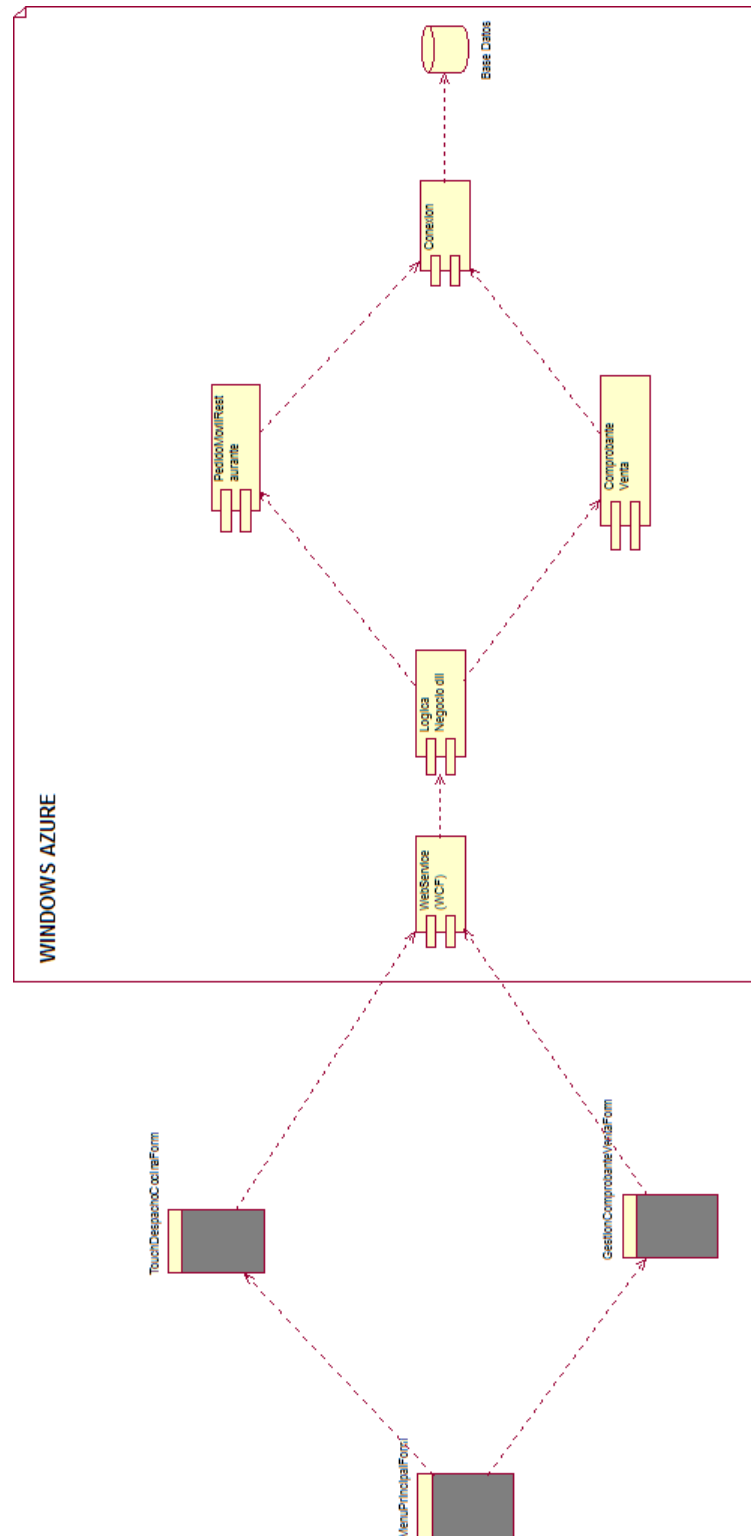


Ilustración 118 Diagrama lógico – Cuarto Sprint

4.6.4.5. DIAGRAMA DE COMPONENTES SPRINT 4



4.6.4.6. INTERFAZ DE LAS HISTORIAS DE USUARIO

✓ Anular Comprobante de Venta:

Datos

Comprobante: 0003 1376 EMITIDO

Tipo: FACTURA POR PUNTO DE VENTA Fecha Comprobante: 03/12/2014

Cliente: 418 ... Cliente... NEGOCIOS HISPANO EIRL

Enviar a: AV. SANTA VICTORIA 444 URB. SANTA VICTORIA (FRENTE A MISTER PAN) LAMBAYEQUE CHICLAYO

Motivo:

Detalle

	Codigo	Cod. Alterno	Descripcion	UM	Cantidad	Precio
1	606		SAN LUIS SIN GAS 625ML	UND	30.000	1.2500
2	14		AMARGO LA BARRA AMARGO 75ML	UND	1.000	9.0000

<<Buscar Articulos... Nuevo Modificar Eliminar

Valor Venta: 39.41

Descuento: 0.00 IGV: 7.09 Importe Venta: 46.50

<<Otras Opciones... Observaciones ... Vista Previa

Anular Cancelar

Ilustración 120 Interfaz Anular Comprobante Venta – Cuarto Sprint

✓ Despacho Cocina:

Ambiente:

Usuario:

Contraseña:

Compañía:

Conectar Aceptar Cancelar

Q W E R T Y U I O P << Borrar 7 8 9 OK

A S D F G H J K L Ñ OK 4 5 6

Z X C V B N M . , Limpiar Todo 1 2 3 Limpiar Todo

Bloq. Mayus ESPACIO * <— —> 0 <<

Ilustración 121 Interfaz Logeo de Usuario Despacho Cocina – Cuarto



Ilustración 122 Interfaz Gestión Despacho Cocina – Cuarto Sprint

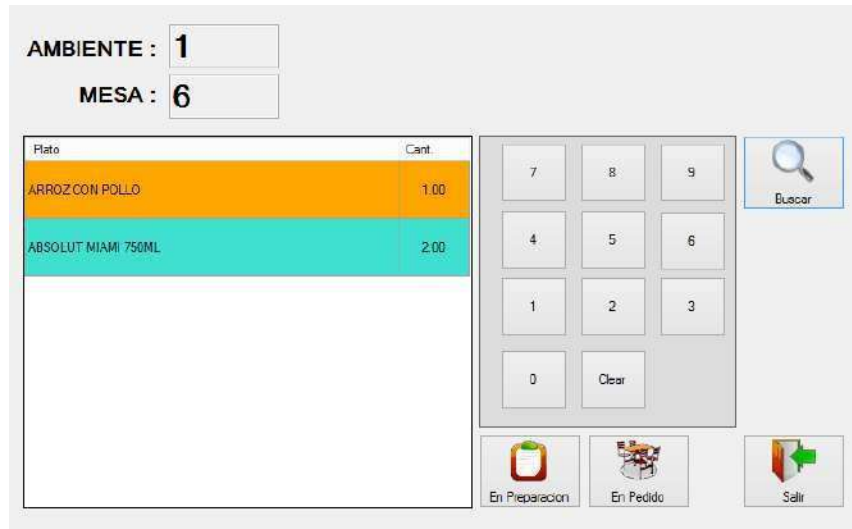


Ilustración 123 Interfaz Pedidos en Mesa – Cuarto Sprint

4.6.4.7. GRAFICANDO EL TABLÓN DE TAREAS

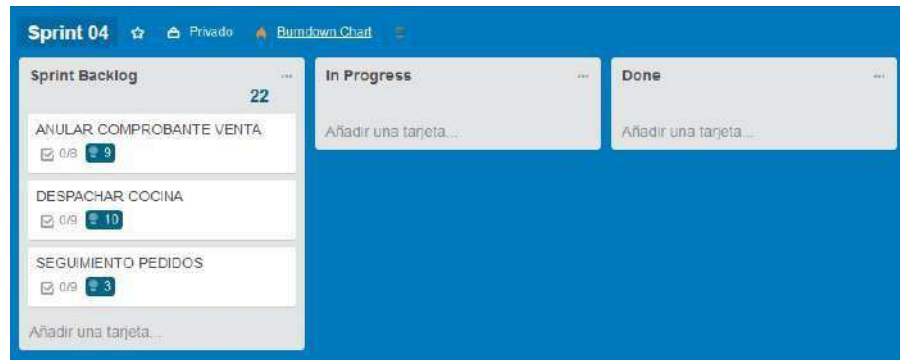


Ilustración 124 Historias de Usuario –Cuarto Sprint

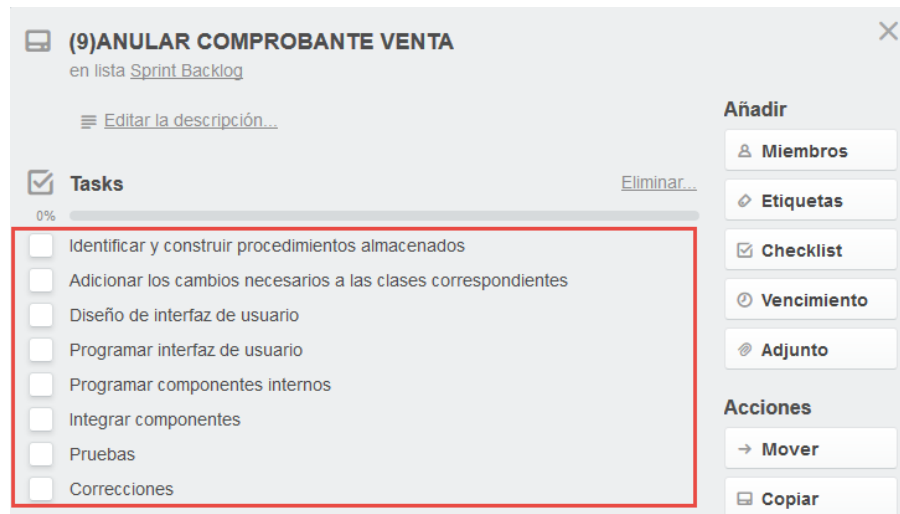


Ilustración 125 H. de Usuario Anular Comprobante Venta

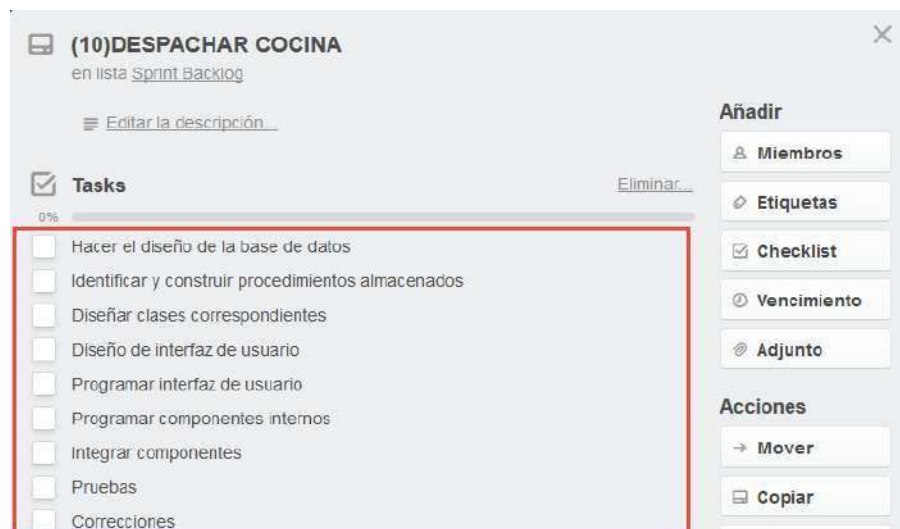


Ilustración 126 H. de Usuario Despachar Cocina

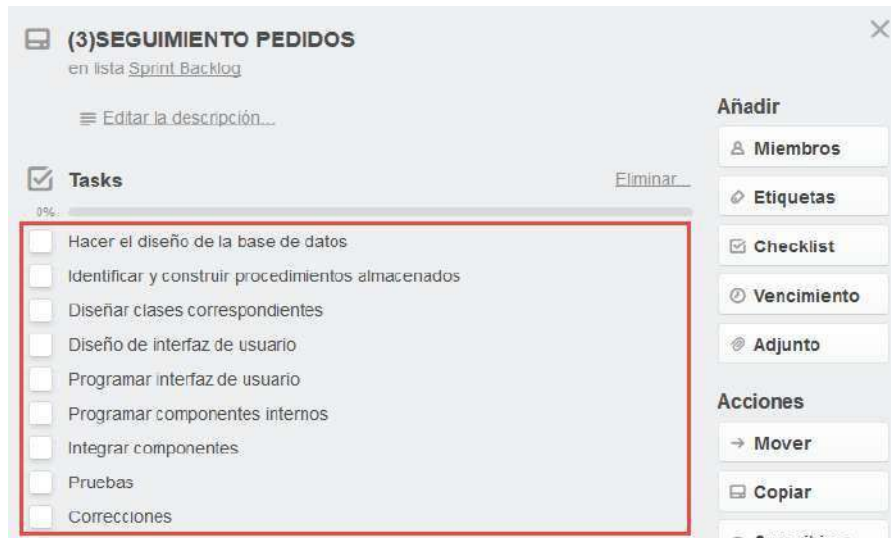


Ilustración 127 H. de Usuario Seguimiento Pedidos

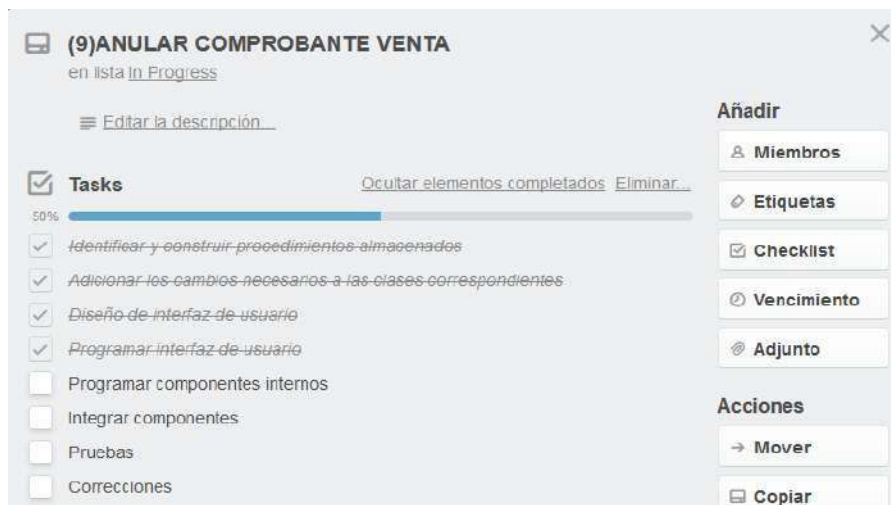


Ilustración 128 Proceso de Tareas Historia 1 – Cuarto Sprint

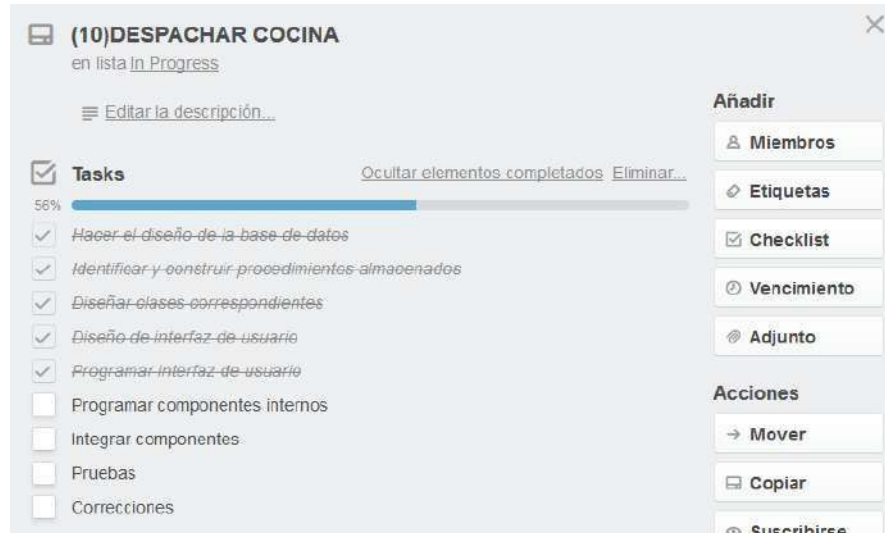
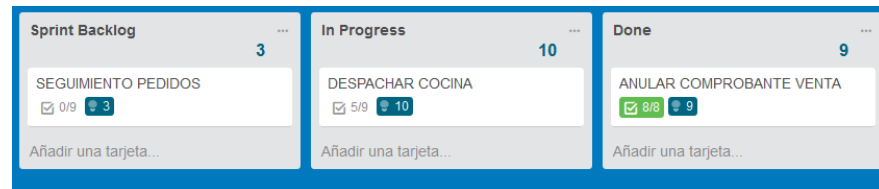


Ilustración 129 Proceso de Tareas Historia 2 – Cuarto Sprint

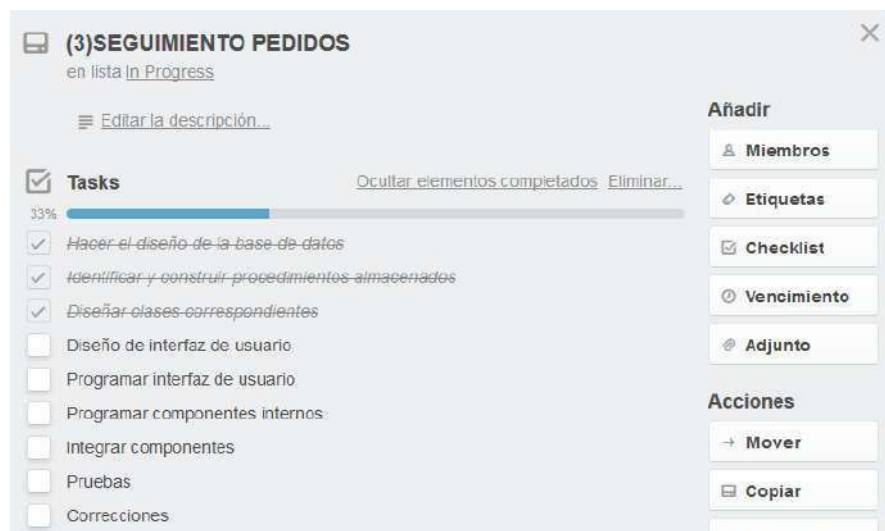


Ilustración 130 Proceso de Tareas Historia 3 – Cuarto Sprint

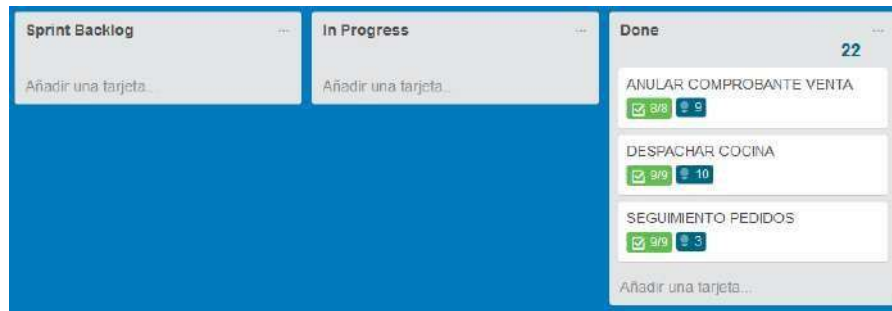


Ilustración 131 Historias de Usuario Concluidas –Cuarto Sprint

4.6.4.8. DIAGRAMA DE BURN DOWN

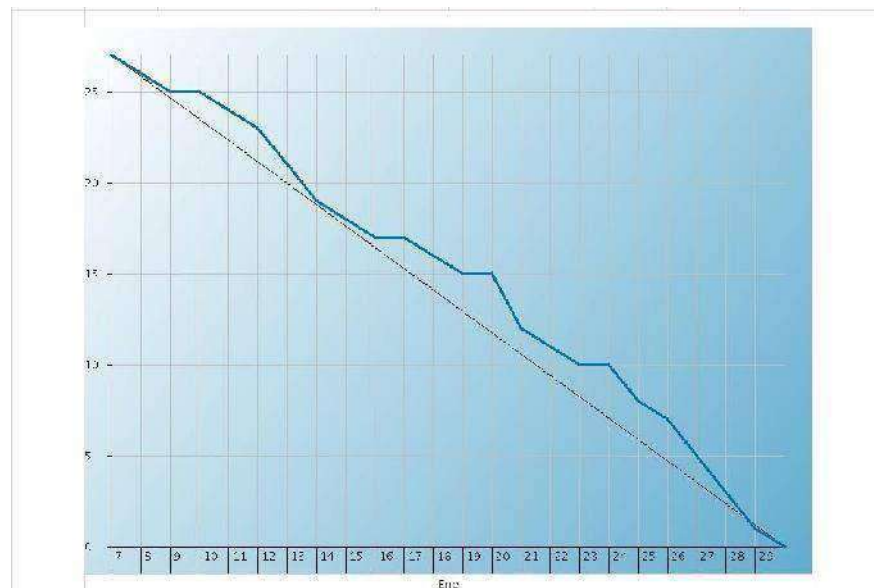


Ilustración 132 Diagrama de Burn Down –Cuarto Sprint

4.6.5. SPRINT 5

4.6.5.1. CALCULO DE LA VELOCIDAD ESTIMADA PARA EL QUINTO SPRINT

- ✓ Calculando duración del sprint:
3 semanas= 15 días laborables.
- ✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	15
Programador 2	15
Total	30 días hombres disponible

Tabla 53. Calculo Días Hombre – Quinto Sprint

- ✓ Calculando factor de dedicación (FD):
 $FD = \text{Puntos de historia (último sprint)} / \text{días hombres disponible}$
 $FD = 27/30 = 0.9$
- ✓ Velocidad estimada: $FD \times \text{Días –Hombre- Disponible}$
 $0.9 \times 30 = 27$

4.6.5.2. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL QUINTO SPRINT:

N°	SPRINT 5	PUNTO S	ESTADO
1	CONFIGURAR CONEXIÓN	2	Terminado
2	AUTENTICAR USUARIO	4	Terminado

3	SELECCIONAR EMPRESA Y REGISTRAR CREDENCIALES DE USUARIO	4	Terminado
4	REGISTRAR INICIO DE PERIODO EN EL SERVIDOR Y EN EL DISPOSITIVO MÓVIL	6	Terminado
5	REGISTRO LOCAL DE DATOS PARA TOMA DE PEDIDOS	12	Terminado
*	TOTAL DE PUNTOS DE HISTORIA	28	28
FECHA			
META	<ul style="list-style-type: none"> • Configurar los parámetros de conexión del dispositivo con el servidor web. • Iniciar sesión tanto en el servidor como en el dispositivo. • Iniciar el periodo de atención. • Descargar los datos necesarios para la toma de pedidos. 		

Tabla 54. Productos en el Quinto Sprint

4.6.5.3 DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: MOZO
Nombre de Historia: Configurar conexión	
Prioridad en Negocio: alta	
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá establecer los parámetros (servidor, aplicación, y base de datos) que se utilizarán para establecer conexión entre el dispositivo móvil y el servidor web.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al ingresar los campos requeridos y presionar el botón aceptar, se comprobará si la conexión fue exitosa, en cuyo caso se mostrará un mensaje de confirmación. ✓ Verificar que al volver a abrir la ventana “Configurar Conexión”, se mostrarán los parámetros previamente guardados, brindando esto la posibilidad de modificarlos. 	
Observaciones:	
Tareas	Responsable
Crear WebMethod que evalúe los parámetros de conexión	José Huamán Valqui
Construir procedimiento almacenado	Alex Ventura Chero

Diseñar clases de conexión al WebService	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 55. 1° Historia de Usuario - Quinto Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: MOZO
Nombre de Historia: Autenticar usuario	
Prioridad en Negocio: alta	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá contrastar las credenciales de usuario indicadas con la información registrada en el servidor. Si la información es correcta se mostrará la relación de empresas disponibles para el usuario autenticado.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al hacer presionar el botón “Aceptar”, se invocará al webmethod “Conectar” encargado de verificar las credenciales indicadas. 	

Verificar que si el proceso de autenticación fue correcto, se mostrará la relación de empresas disponibles para el usuario autenticado	
Observaciones: Previamente se debe haber realizado los siguiente: <ul style="list-style-type: none"> • Haber registrado las empresas disponibles para los usuarios (mozos) • Haber configurado correctamente los parámetros de conexión en el dispositivo móvil 	
Tareas	Responsable
Crear webmethod	José Huamán Valqui
Construir procedimientos almacenados	Alex Ventura Chero
Diseñar de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Elaborar las pruebas correspondientes al proceso.	Alex Ventura Chero
Hacer las correcciones que el proceso requiera	José Huamán Valqui
Firma:	

Tabla 56. 2° Historia de Usuario - Quinto Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: ADMINISTRADOR
Nombre de Historia: Seleccionar empresa y registrar credenciales de usuario	
Prioridad en Negocio: alta	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá seleccionar la empresa asociada al usuario que inició sesión. Además, permitirá registrar en el dispositivo el correcto inicio de sesión y las configuraciones establecidas para ese usuario específico.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar una empresa y presionar el botón “Aceptar” se mostrará la opción “Iniciar Periodo”. ✓ Verificar que al cerrar la aplicación no se volverá a solicitar las credenciales. 	
Observaciones:	
Tareas	Responsable
Crear webmethod	José Huamán Valqui
Construir procedimientos almacenados	Alex Ventura Chero
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui

Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Hacer las correcciones que el proceso requiera	José Huamán Valqui
Firma:	

Tabla 57. 3° Historia de Usuario - Quinto Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Mozo
Nombre de Historia: Registrar inicio de periodo en el servidor y en el dispositivo móvil.	
Prioridad en Negocio: alta	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá marcar el inicio de operaciones tanto en el servidor como en el dispositivo, permitiendo que sea posible proceder a la descarga de los datos necesarios para el proceso de toma de pedidos.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Iniciar Periodo”, se mostrará la interfaz necesaria para dicho propósito. ✓ Verificar que sin haber iniciado el periodo no se podrá realizar ninguna operación debido a que los datos necesarios son obtenidos en el proceso de sincronización que requiere que previamente se haya iniciado el periodo. 	

- ✓ Verificar que al presionar el botón “Iniciar Periodo”, en el servidor, en tabla “RECONTROLENIOMOVIL” se insertará un nuevo registro con el código del mozo, la fecha actual y el estado “020”.
- ✓ Verificar que una vez culminado el proceso se mostrará un mensaje de confirmación y el botón quedará deshabilitado.

Observaciones: Previamente debe haberse realizado lo siguiente:

- El mozo que pretende iniciar su periodo, debe haber iniciado sesión correctamente.

Tarea	Responsable
Construir los procedimientos almacenados necesarios	Alex Ventura Chero
Crear webmethod	José Huamán Valqui
Diseñar las clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Hacer las correcciones que el proceso requiera	José Huamán Valqui
Firma:	

Tabla 58. 4° Historia de Usuario - Quinto Sprint

HISTORIA DE USUARIO	
Número:5	Usuario: Mozo
Nombre de Historia: Registro local de datos para toma de pedidos	
Prioridad en Negocio: alta	
Puntos estimados: 12	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá descargar información actualizada, relacionada a la toma de pedido, desde el servidor y registrarla de manera local en el dispositivo para su posterior manejo y consulta.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Sincronizar datos”, se mostrará la interfaz necesaria para dicho propósito. ✓ Verificar que al presionar el botón “Sincronizar”, se iniciará el proceso de sincronización, al término del cual se mostrará un mensaje confirmando éxito del mismo. ✓ Verificar que después de culminado el proceso de sincronización, el botón quedará deshabilitado. ✓ Verificar que recién después de culminado el proceso de sincronización, se podrá iniciar el proceso de toma de pedido. 	

Observaciones: Previamente debe haberse iniciado el periodo tanto en el dispositivo como en el servidor.	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethod	José Huamán Valqui
Diseñar clases para el manejo de la base de datos local	Alex Ventura Chero
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Hacer las correcciones que el proceso requiera	José Huamán Valqui
Firma:	

Tabla 59. 5° Historia de Usuario - Quinto Sprint

4.6.5.4. DIAGRAMA DE CLASES SPRINT 5

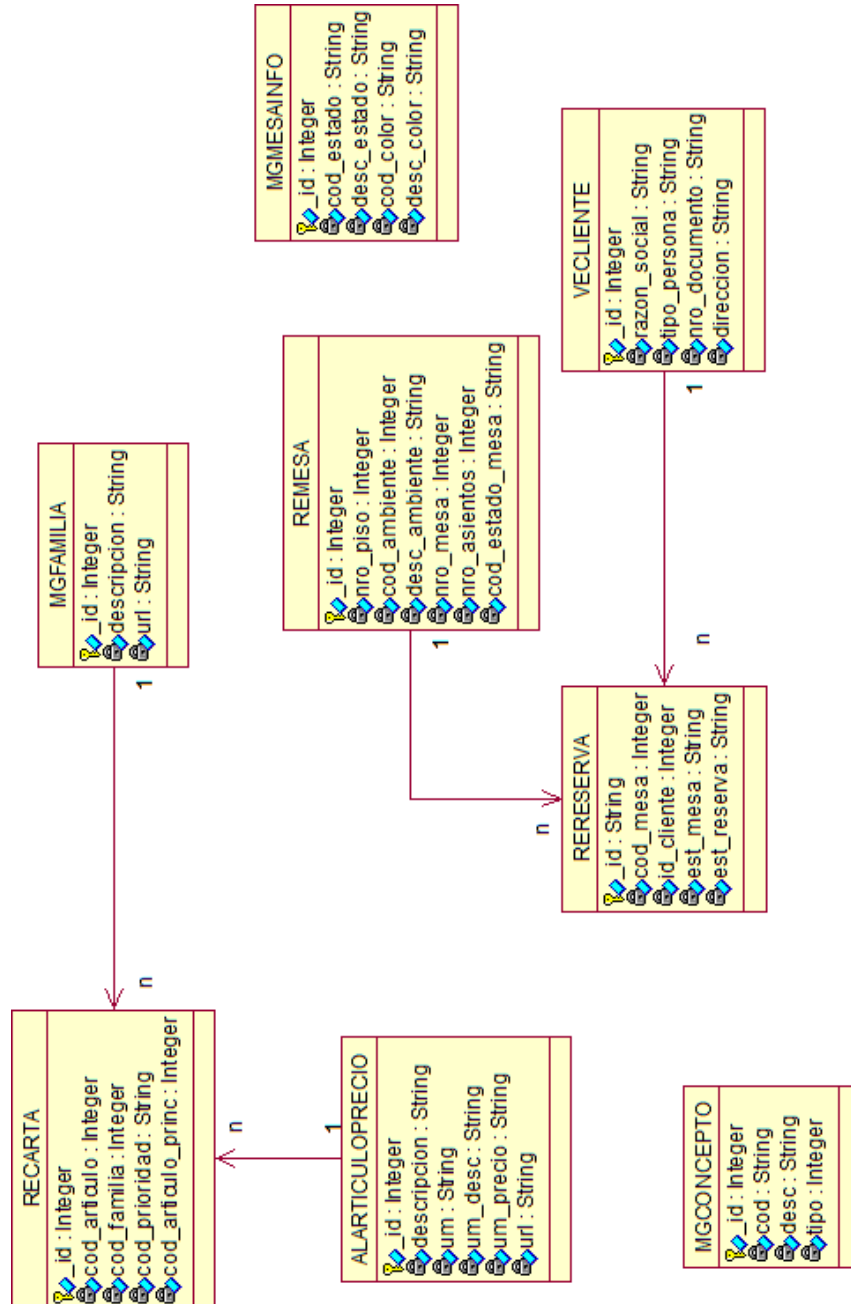


Ilustración 133 Diagrama de Clases – Quinto Sprint

4.6.5.5.

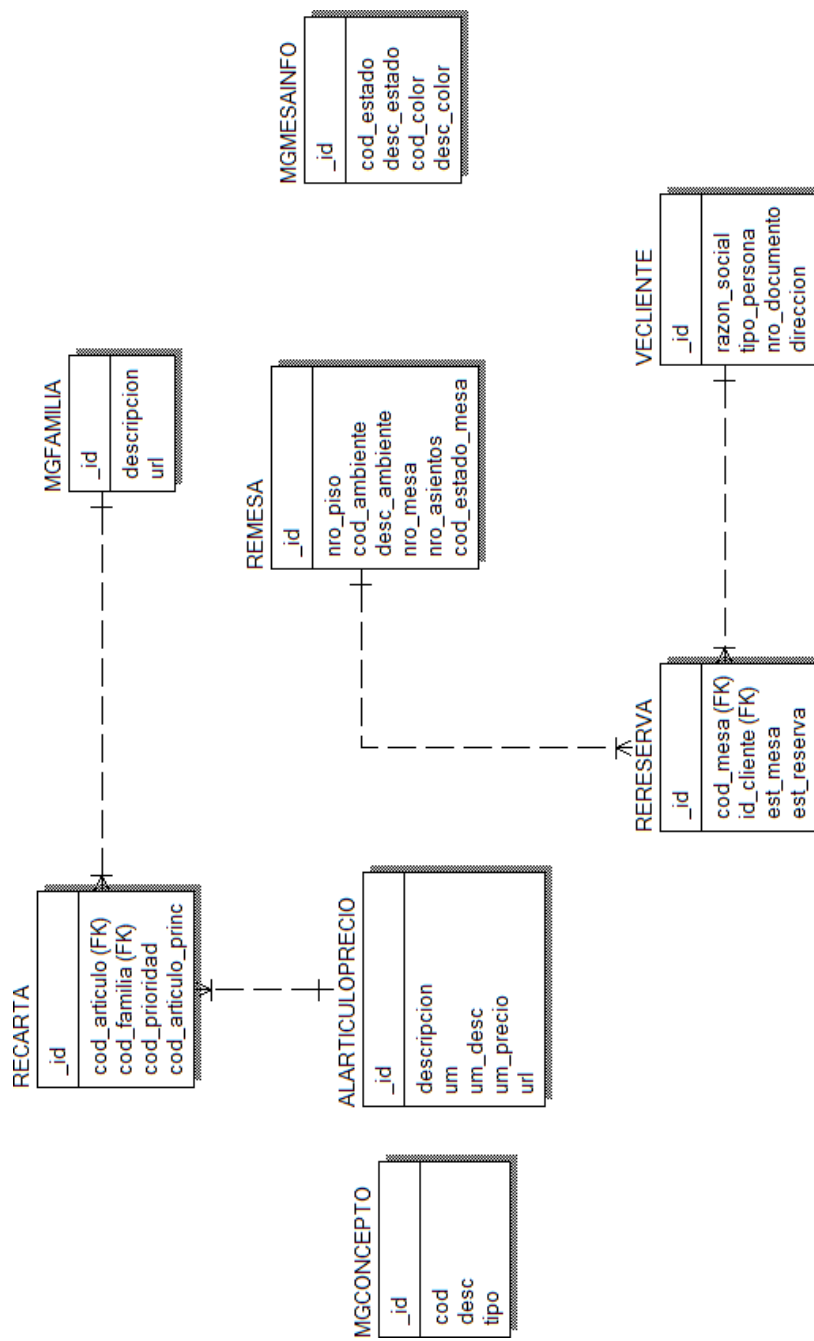


Ilustración 134 Diagrama Lógico – Quinto Sprint

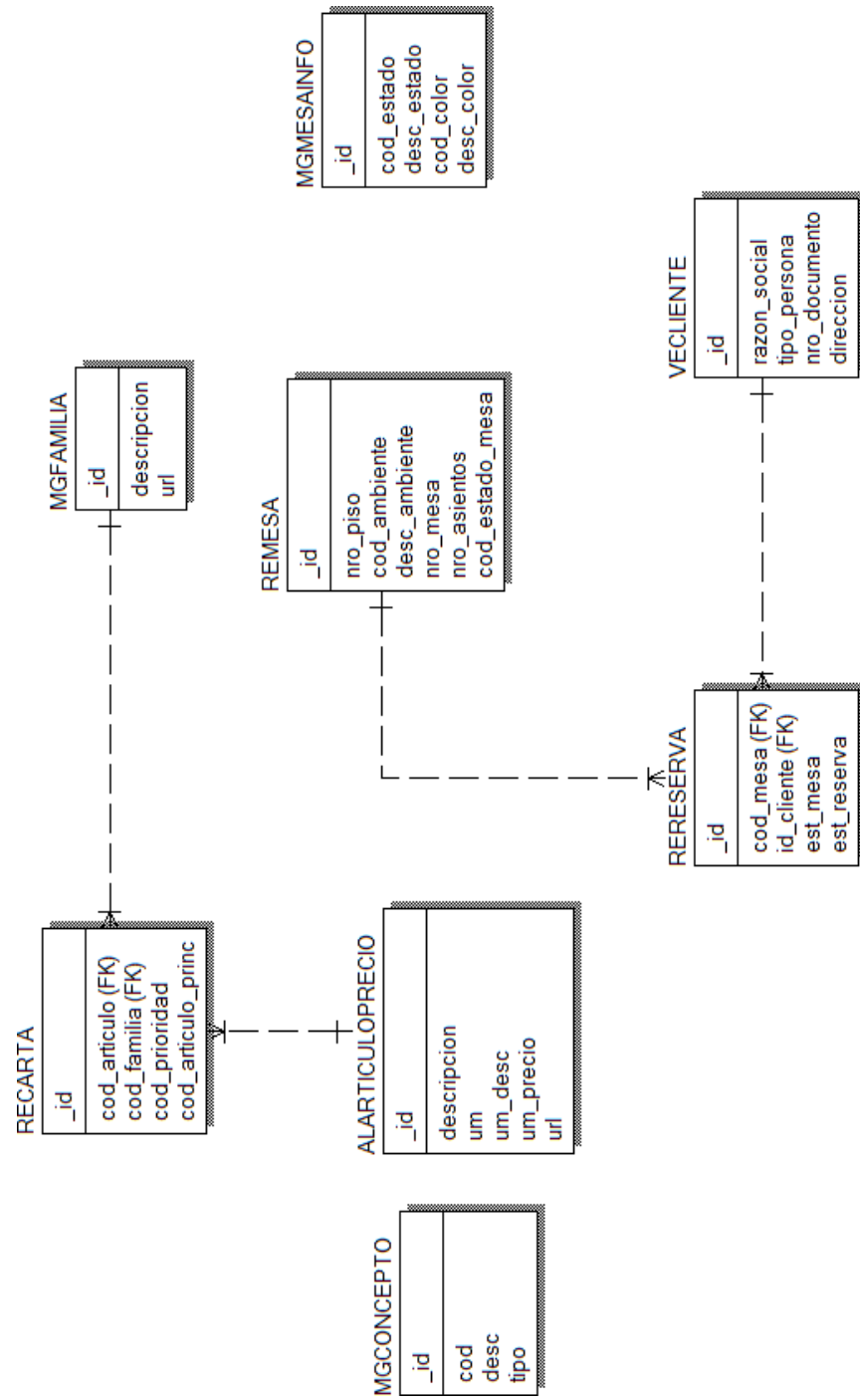


Ilustración 135 Diagrama Físico – Quinto Sprint

4.6.5.6. DIAGRAMA DE COMPONENTES SPRINT 5

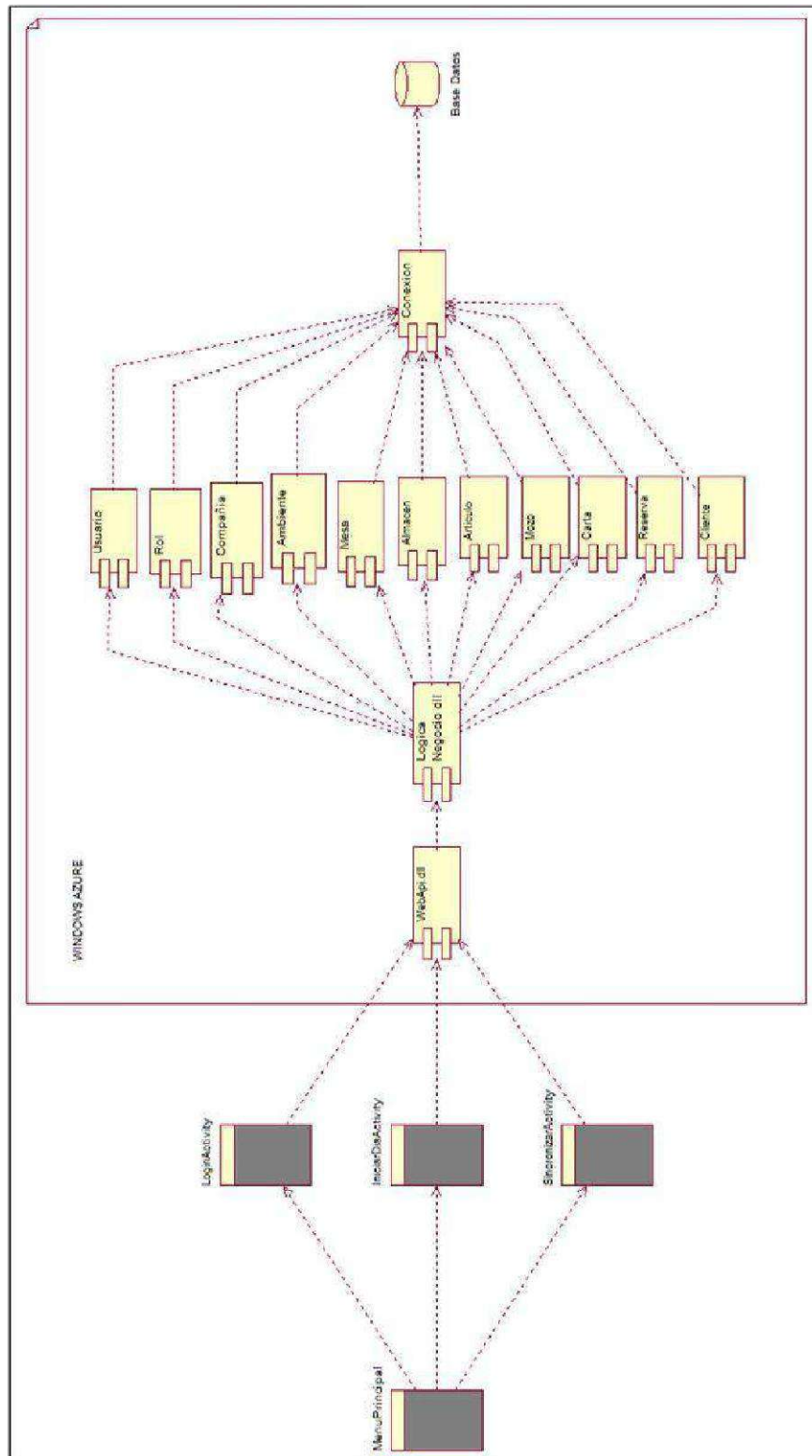


Ilustración 136 Diagrama de Componentes – Quinto Sprint

4.6.5.7. INTERFAZ DE LAS HISTORIAS DE USUARIO

- ✓ Configurar conexión:



SmartWaiter

Configurar Conexión

Usuario:

Ingrese nombre de usuario

Contraseña:

Ingrese su contraseña

Compañía:

Conectar



SmartWaiter

CONFIGURAR CONEXIÓN

Servidor:

siemprestiga.cloudapp.net

Aplicación:

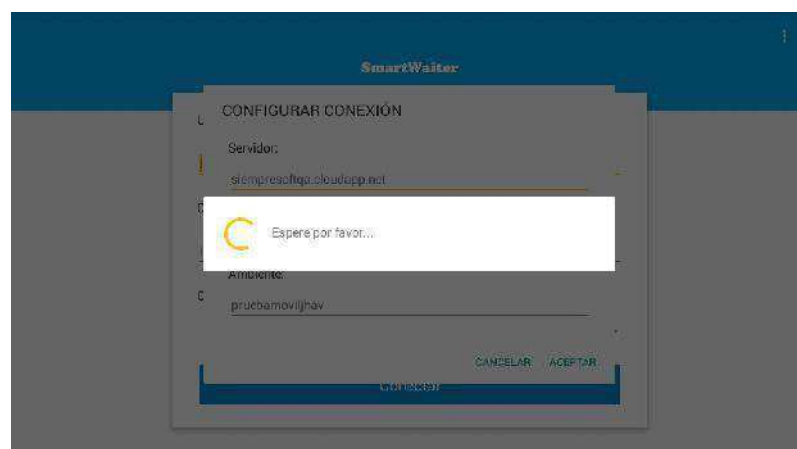
pruebamovil1.ec

Ambiente:

pruebamovil1.hav

CANCELAR ACEPTAR

Conectar



SmartWaiter

CONFIGURAR CONEXIÓN

Servidor:

siemprestiga.cloudapp.net

Ambiente:

pruebamovil1.hav

CANCELAR ACEPTAR

Conectar

Espera por favor...

Ilustración 137 Interfaz Configurar Conexión – Quinto Sprint

✓ Autenticar usuario:

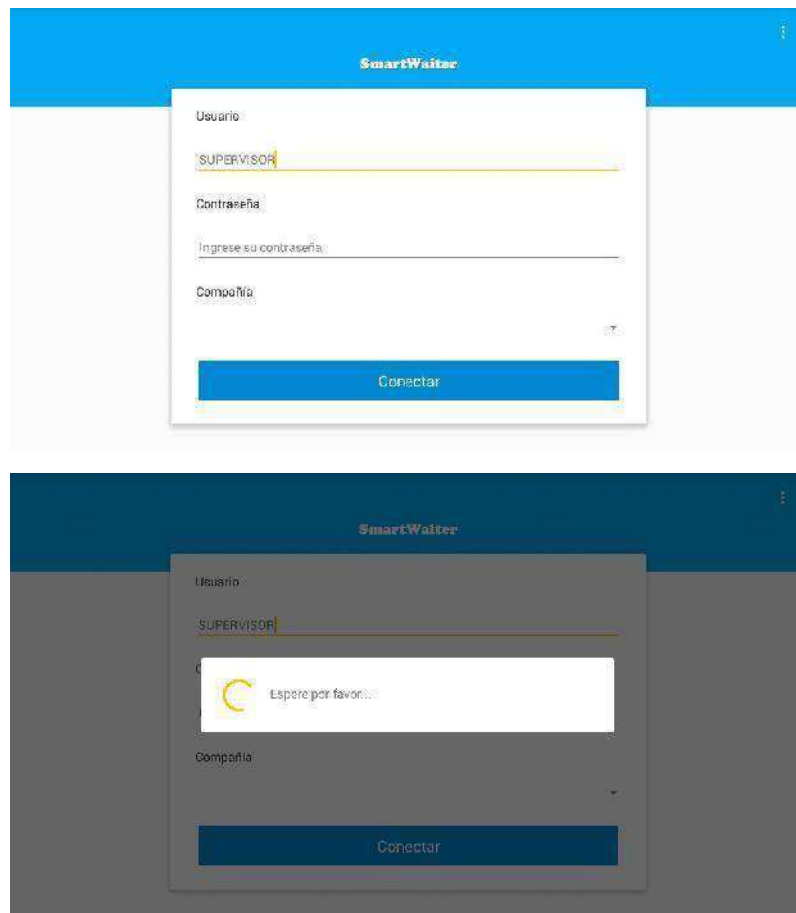
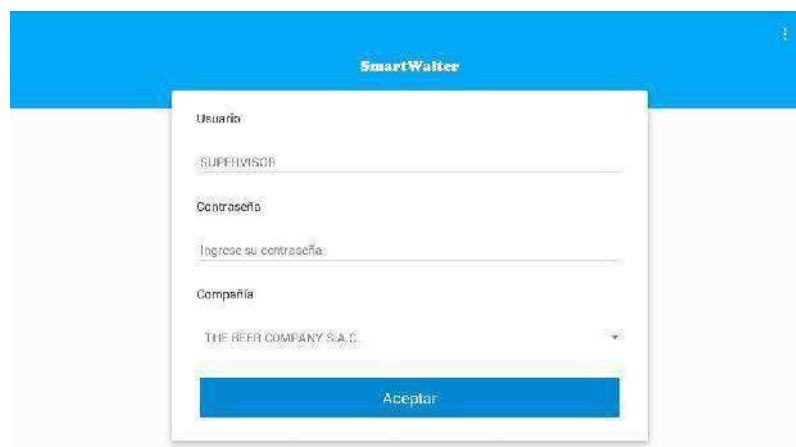


Ilustración 138 Interfaz Autenticar usuario – Quinto Sprint

✓ Seleccionar empresa y registrar credenciales de usuario:



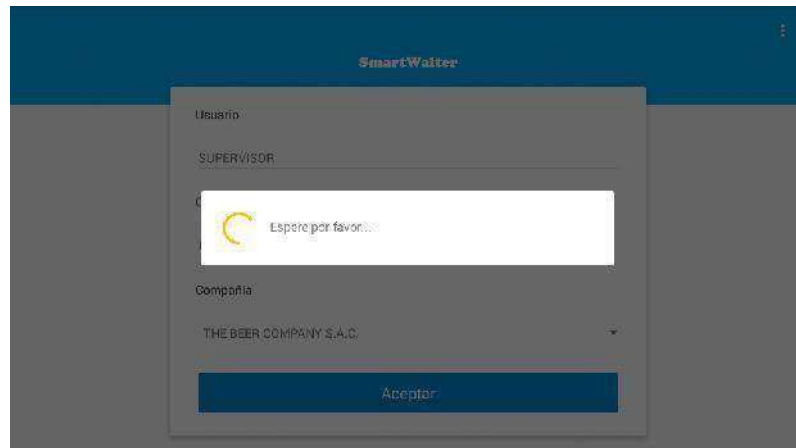
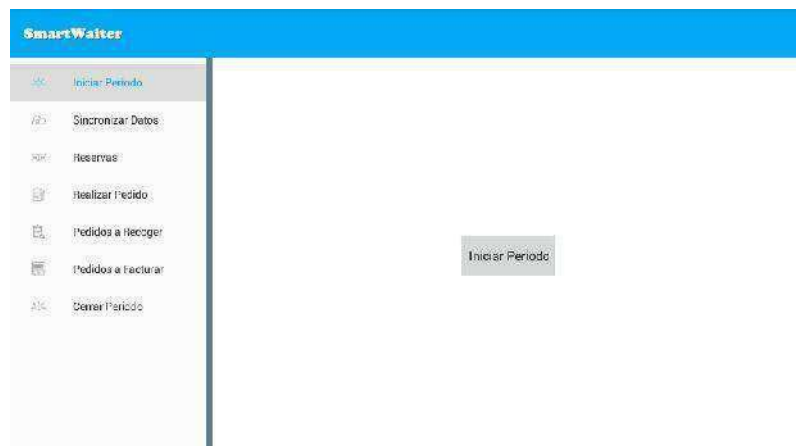


Ilustración 139 Seleccionar empresa y registrar credenciales de usuario – Quinto Sprint

- ✓ Registrar inicio de periodo en el servidor y en el dispositivo móvil:



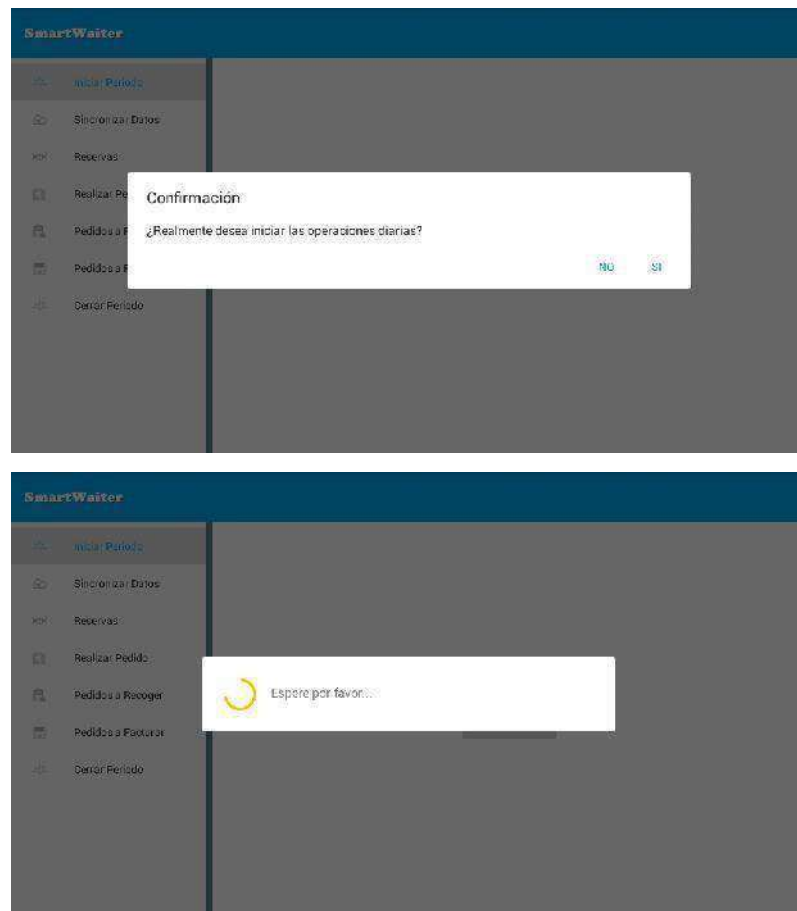


Ilustración 140 Registrar inicio de periodo en el servidor y en el dispositivo móvil – Quinto Sprint

✓ Registro local de datos para toma de pedidos:



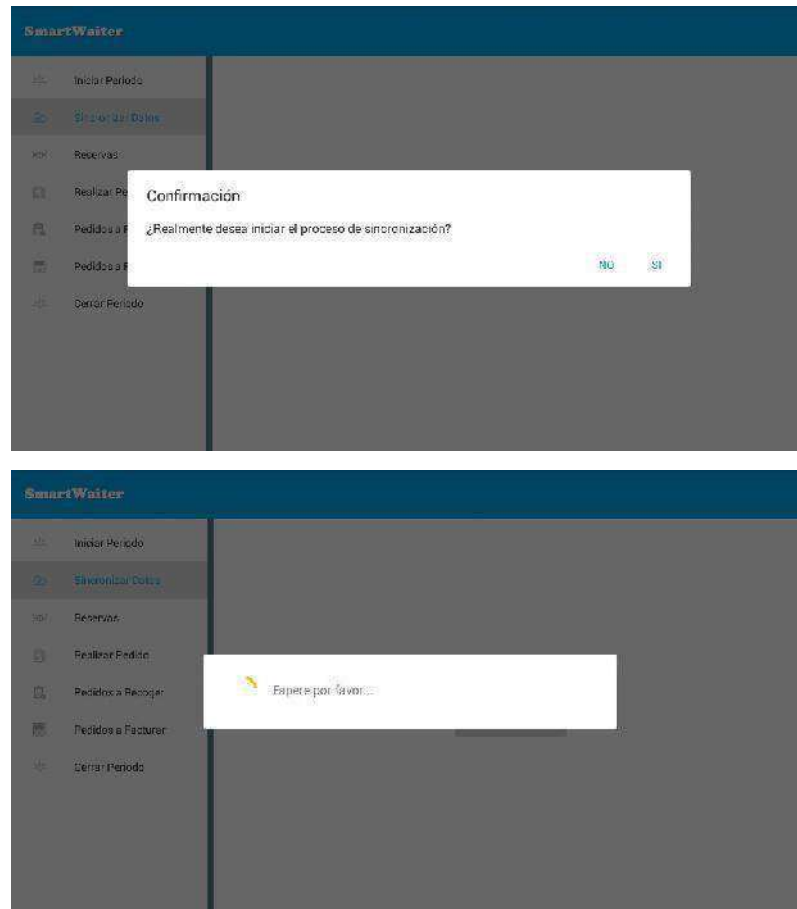


Ilustración 141 Registro local de datos para toma de pedidos – Quinto Sprint

4.6.5.8. GRAFICANDO EL TABLÓN DE TAREAS

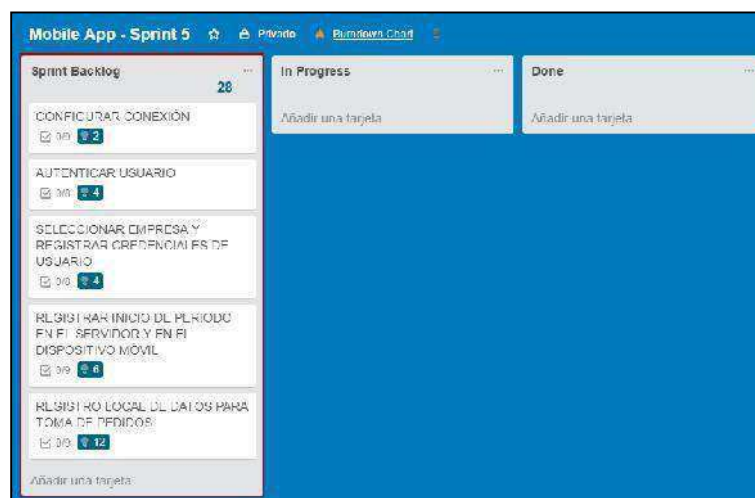


Ilustración 142 Historias de Usuario –Quinto Sprint

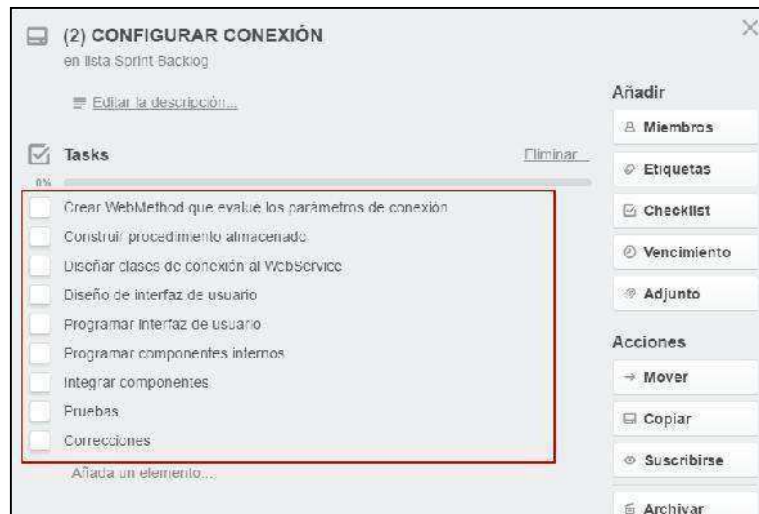


Ilustración 143 H. de Usuario Configurar Conexión

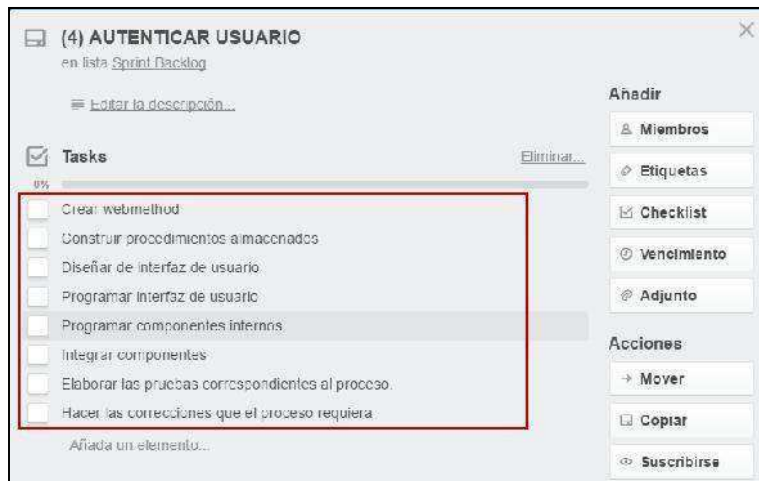


Ilustración 144 H. de Usuario Autenticar Usuario

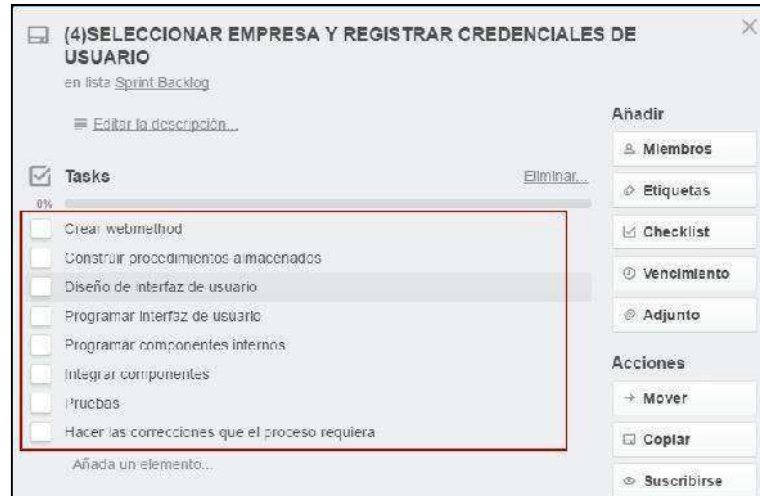


Ilustración 145 H. de Usuario Seleccionar Empresa y Registrar Credenciales De Usuario

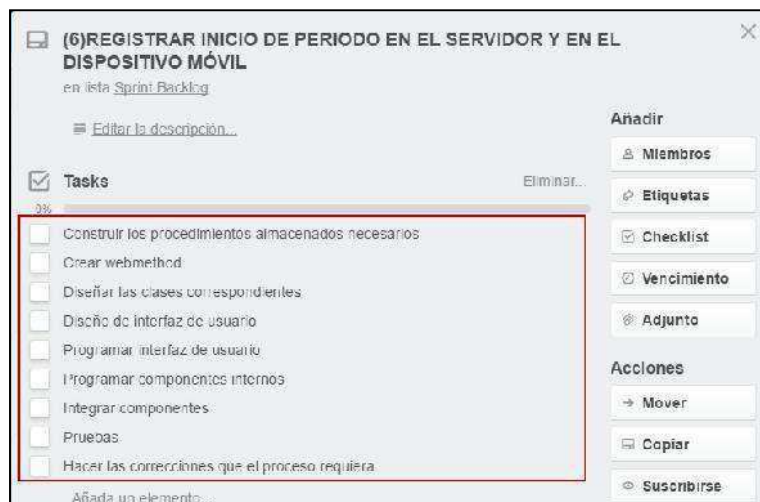


Ilustración 146 H. de Usuario Registrar Inicio de Periodo en el Servidor y en el Dispositivo Móvil

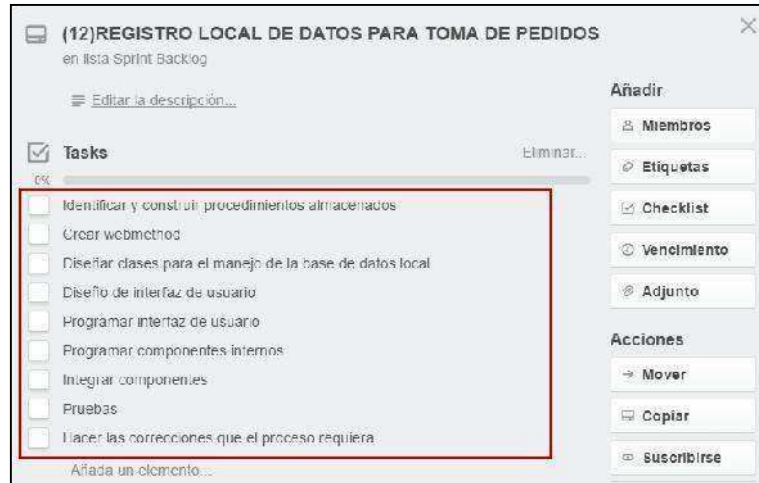
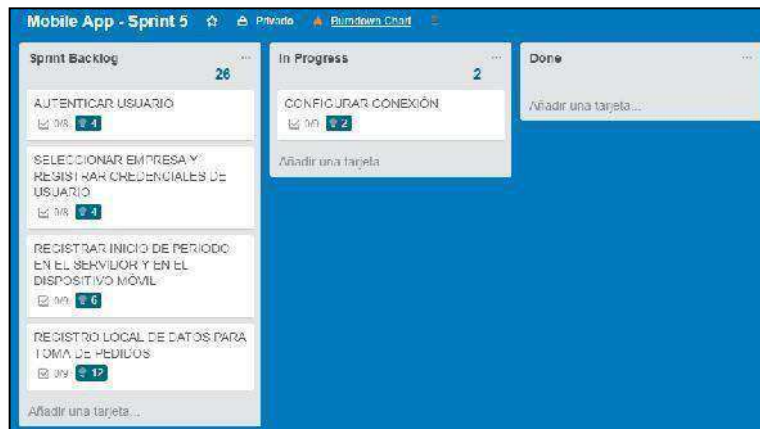


Ilustración 147 H. de Usuario Registro Local de Datos para Toma de Pedidos



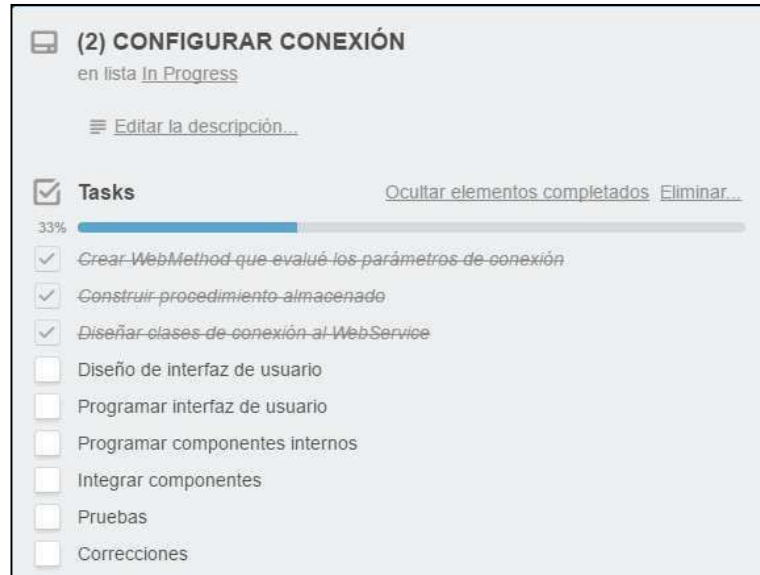


Ilustración 148 Proceso de Tareas Historia 1 – Quinto Sprint

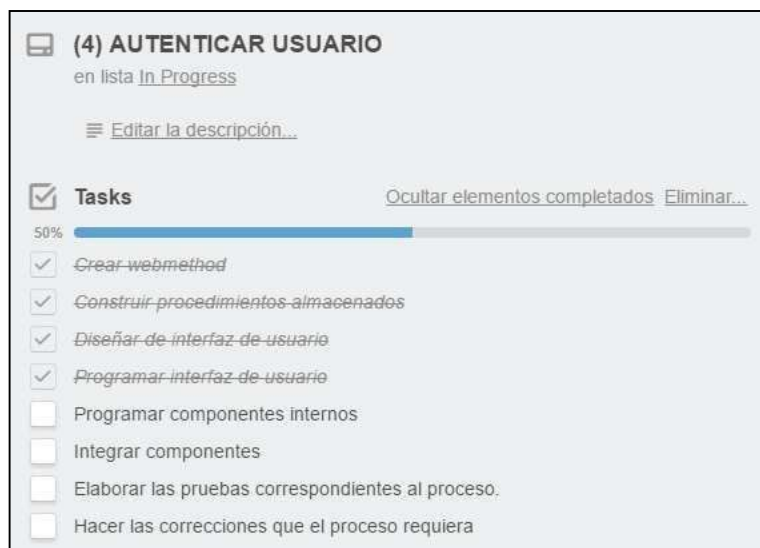
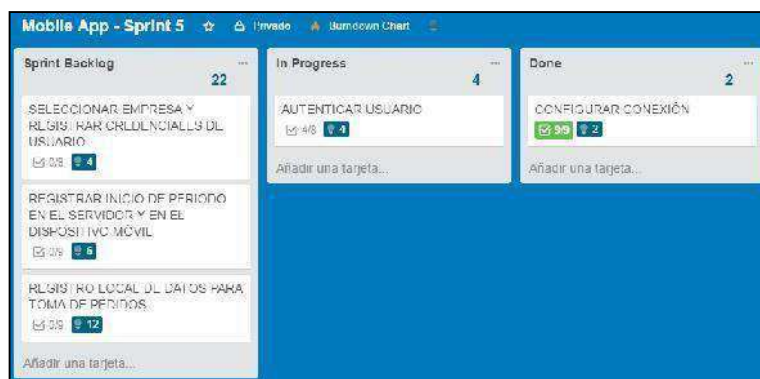


Ilustración 149 Proceso de Tareas Historia 2 – Quinto Sprint



Ilustración 150 Proceso de Tareas Historia 3 – Quinto Sprint



Ilustración 151 Proceso de Tareas Historia 4 – Quinto Sprint

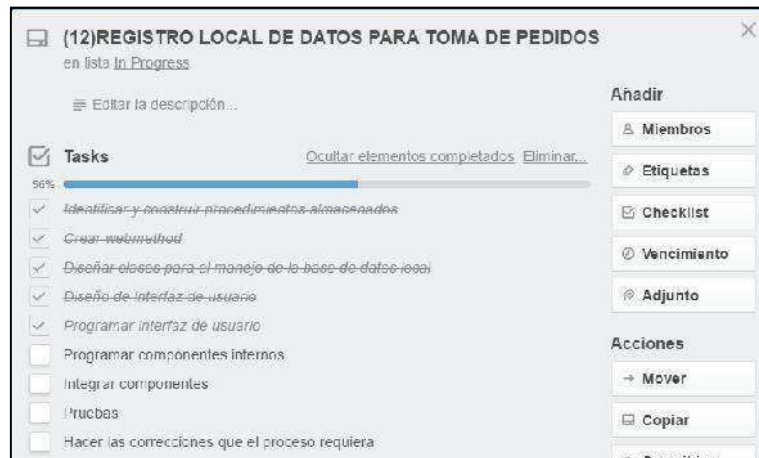
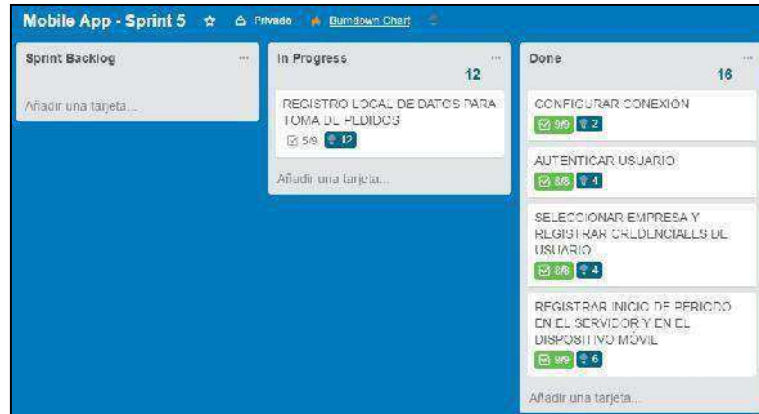


Ilustración 152 Proceso de Tareas Historia 5 – Quinto Sprint

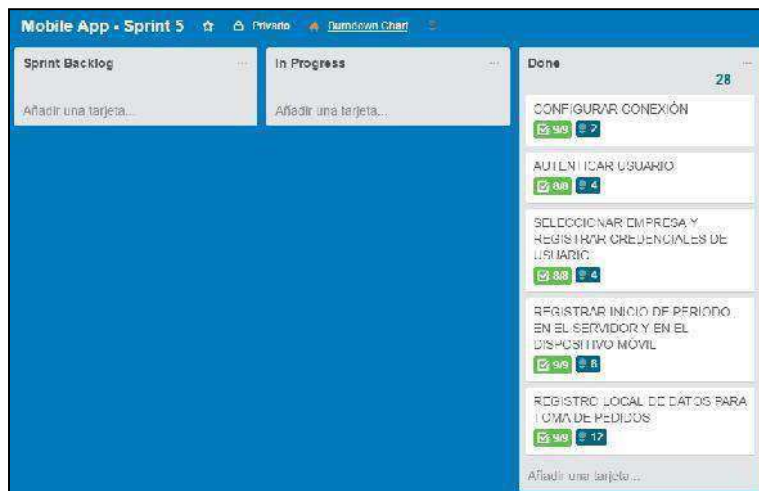


Ilustración 153 Historias de Usuario Concluidas –Quinto Sprint

4.6.5.9. DIAGRAMA DE BURN DOWN

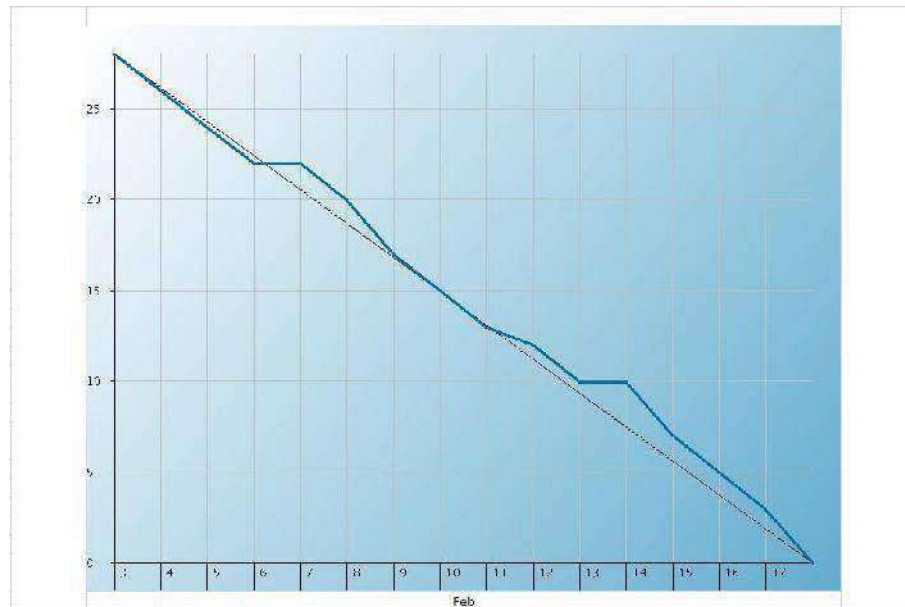


Ilustración 154 Diagrama de Burn Down –Quinto Sprint

4.6.5.10. DESARROLLO DE HISTORIAS

Configurar Conexión y Autenticar Usuario

➤ activity_login.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="?attr/listPreferredItemHeightSmall"
    android:layout_alignParentTop="true"
    android:elevation="@dimen/appbar_elevation"
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
    app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>
```

```
<ImageView
    android:id="@+id/app_logo_imageview"
    android:layout_width="match_parent"
    android:layout_height="?attr/listPreferredItemHeightSmall"
    android:layout_below="@+id/toolbar"
    android:background="@color/primary"
    android:contentDescription="@string/app_name">
```



```

        android:elevation="@dimen/appbar_elevation"
        android:scaleType="center"
        android:src="@drawable/ic_logo"/>

<View
    android:layout_width="match_parent"
    android:layout_height="@dimen/form_app_bar_overlap_large"
    android:layout_below="@id/app_logo_imageview"
    android:background="@color/primary"
    android:elevation="@dimen/appbar_elevation"/>

<android.support.v7.widget.CardView
    android:id="@+id/form_container"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/app_logo_imageview"
    android:layout_marginLeft="@dimen/form_horizontal_margin_large"
    android:layout_marginRight="@dimen/form_horizontal_margin_large"
    app:cardElevation="@dimen/form_card_elevation"
    app:cardPreventCornerOverlap="false">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"

        android:paddingBottom="@dimen/abc_list_item_padding_horizontal_material"
        android:paddingEnd="@dimen/waiter_detail_horizontal_padding"
        android:paddingLeft="@dimen/waiter_detail_horizontal_padding"
        android:paddingRight="@dimen/waiter_detail_horizontal_padding"
        android:paddingStart="@dimen/waiter_detail_horizontal_padding"

        android:paddingTop="@dimen/abc_list_item_padding_horizontal_material"
        tools:context="com.neversoft.smartwaiter.ui.LoginActivity">

        <TextView
            android:id="@+id/usuarioLabel"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/usuario_label"

            android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

        <EditText
            android:id="@+id/usuarioEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="@color/secondary_text"

            android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
            android:hint="@string/usuario_hint"
            android:imeOptions="actionNext"
            android:inputType="textNoSuggestions"

```

```

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <TextView
        android:id="@+id/contrasenaLabel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"

        android:text="@string/contrasena_label"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <EditText
        android:id="@+id/contrasenaEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
        android:hint="@string/contrasena_hint"
        android:textColor="@color/secondary_text"
        android:imeOptions="actionDone"
        android:inputType="textPassword"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <TextView
        android:id="@+id/companiaLabel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
        android:text="@string/compania_label"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <Spinner
        android:id="@+id/companiaSpinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/secondary_text"

android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
        android:spinnerMode="dropdown"

android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <Button
        android:id="@+id/iniciarSessionButton"
        android:layout_width="match_parent"
        android:layout_height="?attr/listPreferredItemHeight"

android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
        android:text="@string/conectar"
        android:textAppearance="@style/TextAppearance.AppCompat.Title"

```

```

        android:theme="@style/BrandButtonStyle"/>

        <Button
            android:id="@+id/aceptarButton"
            android:layout_width="match_parent"
            android:layout_height="?attr/listPreferredItemHeight"

            android:layout_marginTop="@dimen/abc_list_item_padding_horizontal_material"
            android:text="@string/aceptar"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"
            android:theme="@style/BrandButtonStyle"/>

    </LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

```

➤ LoginActivity.java

```

public class LoginActivity extends AppCompatActivity implements
OnClickListener {

```

```

    public static final String PREF_CONFIG = "prefConfig";
    private EditText mUsuarioEditText;
    private EditText mContrasenaEditText;
    private Spinner mCompaniaSpinner;
    private Button mIniciarSessionButton;
    private Button mAceptarButton;
    private MaterialDialog mProgress;
    private SharedPreferences mPrefLoginValues;
    private SharedPreferences mPrefControl;
    private SharedPreferences mPrefConfig;
    private SharedPreferences mPrefConexion;

```

```

    private String mAmbiente = "";
    private String mUsuario = "";
    private String mContrasena = "";
    private SpinnerEE mSelectedItem;

```

```

    private String mURLServer;

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(false);
        getSupportActionBar().setDisplayShowTitleEnabled(false);
    }

```

```

mUsuarioEditText = (EditText) findViewById(R.id.usuarioEditText);
mContrasenaEditText = (EditText) findViewById(R.id.contrasenaEditText);
mCompaniaSpinner = (Spinner) findViewById(R.id.companiaSpinner);
mIniciarSessionButton = (Button) findViewById(R.id.iniciarSessionButton);
mAceptarButton = (Button) findViewById(R.id.aceptarButton);

mUsuarioEditText.setFilters(new InputFilter[]{new InputFilter.AllCaps()});

mAceptarButton.setVisibility(View.GONE);
mUsuarioEditText.setEnabled(true);
mContrasenaEditText.setEnabled(true);

mIniciarSessionButton.setOnClickListener(this);
mAceptarButton.setOnClickListener(this);

SmartWaiter app = (SmartWaiter) getApplication();
if (app.getFirstRun()) {
    mPrefControl = getSharedPreferences(ControlSharedPref.NAME,
MODE_PRIVATE);
    ControlSharedPref.save(mPrefControl, false, "", false, false, false, "",
false);

    app.setRunned();
}
mPrefLoginValues = getSharedPreferences(LoginSharedPref.NAME,
MODE_PRIVATE);
mPrefConfig = getSharedPreferences(PREF_CONFIG, MODE_PRIVATE);
mPrefConexion = getSharedPreferences(ConexionSharedPref.NAME,
MODE_PRIVATE);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_login, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.menuConfigurarConexion:
            new
ConfigurarConexionDialogFragment().show(getSupportFragmentManager(),
"dlgConfigurarConex");
            return true;
    }
    return super.onOptionsItemSelected(item);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {

```

```

        case R.id.iniciarSessionButton:
            logInUsuario();
            break;
        case R.id.aceptarButton:
            aceptarIngresar();
            break;
    }
}

public void logInUsuario() {
    if (RestUtil.datosConexionCompleto(getApplicationContext())) {
        if (!Funciones.isEditTextEmpty(mUsuarioEditText)) {

            mURLServer = RestUtil.obtainURLServer(getApplicationContext());
            String url = mURLServer
                +
                "Compartido/Conectar/?ambiente=%s&usuario=%s&clave=%s&cadenaConexion
                =%s";
            mAmbiente =
            mPrefConexion.getString(ConexionSharedPref.AMBIENTE, "");
            mUsuario = mUsuarioEditText.getText().toString().trim()
                .toUpperCase(Locale.getDefault());
            mContraseña = mContraseñaEditText.getText().toString()
                .trim();

            try {
                String encodedAmbiente = URLEncoder.encode(mAmbiente,
                    "utf-8");
                String urlWithParams = String.format(url,
                    "", mUsuario, mContraseña,
                    encodedAmbiente);
                new DoLoginUsuario().execute(urlWithParams);
            } catch (Exception e) {
                Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG)
                    .show();
            }

        } else {
            Toast.makeText(this, "Debe ingresar un nombre de usuario.",
                Toast.LENGTH_LONG).show();
        }
    } else {
        Toast.makeText(this, "Datos de conexión incompletos. Por favor revise los
        datos y vuelva a intentarlo.",
            Toast.LENGTH_LONG).show();
    }
}

private void processResponse(String response) {

    Context context = LoginActivity.this;
    try {

        JSONObject jsonObjectResponse = (new
        JsonParser()).parse(response).getAsJsonObject();
    }
}

```

```

        JSONArray jsonTablaOtros = jsonObjectResponse
            .getAsJSONArray("TablaOtros");
        JsonObject jsonDatosAutenticacion =
            jsonTablaOtros.get(0).getAsJsonObject();
        boolean usuarioExite =
            jsonDatosAutenticacion.get("usuario").getAsBoolean();
        if (usuarioExite) {
            JSONArray jsonCompanyArray = jsonObjectResponse
                .getAsJSONArray("TablaCompañías");
            if (jsonCompanyArray.size() > 0) {
                showCompañías(jsonCompanyArray);

            } else {
                limpiarLogin();
                Toast.makeText(context, "El usuario no tiene compañías.",
                    Toast.LENGTH_LONG).show();
            }
        } else {
            limpiarLogin();
            Toast.makeText(context, "Usuario y/o contraseña no válidos.",
                Toast.LENGTH_LONG).show();
        }
    } catch (Exception e) {
        limpiarLogin();
        Log.d(DBHelper.TAG, e.toString());
        Toast.makeText(context, e.getMessage(), Toast.LENGTH_LONG).show();
    }
}

private void showCompañías(JSONArray jsonCompanyArray) throws Exception
{
    Context context = LoginActivity.this;

    ArrayList<SpinnerEE> listaCompanias =
        parseCompañías(jsonCompanyArray);
    ArrayAdapter<SpinnerEE> adapter = new ArrayAdapter<SpinnerEE>(
        context, android.R.layout.simple_spinner_item, listaCompanias);
    mCompaniaSpinner.setAdapter(adapter);
    mIniciarSessionButton.setVisibility(View.GONE);
    mUsuarioEditText.setEnabled(false);
    mContrasenaEditText.setEnabled(false);
    mAceptarButton.setVisibility(View.VISIBLE);

    LoginSharedPref.save(mPrefLoginValues, mUsuario, mContrasena, null, null,
        true);
}

private ArrayList<SpinnerEE> parseCompañías(JSONArray jsonCompanyArray)
    throws Exception {
    ArrayList<SpinnerEE> listaCompanias = new ArrayList<SpinnerEE>();
    for (int i = 0; i < jsonCompanyArray.size(); i++) {
        JsonObject comp = jsonCompanyArray.get(i).getAsJsonObject();
    }
}

```

```

        SpinnerEE item = new SpinnerEE();
        item.setCodigo(comp.get("codcia").getString());
        item.setDescripcion(comp.get("razonsocial").getString());
        listaCompanias.add(item);
    }
    return listaCompanias;
}

private void limpiarLogin() {
    mContrasenaEditText.setText("");
}

public void aceptarIngresar() {

    mSelectedItem = (SpinnerEE) mCompaniaSpinner.getSelectedItem();

    String url = mURLServer
        +
        "compartido/aceptar/?usuario=%s&codcia=%s&cadenaConexion=%s";
    if (mUsuario != "" && mSelectedItem.getCodigo().trim() != "") {
        try {

            String encodedAmbiente = URLEncoder.encode(mAmbiente, "utf-8");
            String urlWithParams = String.format(url, mUsuario,
                mSelectedItem.getCodigo().trim(), encodedAmbiente);
            new DoAceptar().execute(urlWithParams);
        } catch (Exception e) {
            Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
}

private void insertSettings(String response) {

    JSONArray jsArraySettings;
    try {
        Editor editor = mPrefConfig.edit();
        editor.clear();

        jsArraySettings = (new JsonParser()).parse(response).getAsJSONArray();
        for (int i = 0; i < jsArraySettings.size(); i++) {
            JSONObject jsObj = jsArraySettings.get(i).getAsJsonObject();
            String value = jsObj.get("Value").getString();
            String key = jsObj.get("Key").getString();
            editor.putString(key.trim(), value.trim());
        }
        editor.commit();
        LoginSharedPref.save(mPrefLoginValues, null, null,
            mSelectedItem.getDescripcion().trim(), true, false);
        Log.d(DBHelper.TAG, "Configuraciones Insertadas");

        startActivity(new Intent(LoginActivity.this, IniciarDiaActivity.class));
        finish();
    } catch (Exception e) {
        Toast.makeText(this, "Se produjo la excepción: " + e.getMessage(),

```

```

        Toast.LENGTH_LONG).show();
    }
}

private void showProgressIndicator(boolean showValue) {
    if (showValue) {
        mProgress = new MaterialDialog.Builder(LoginActivity.this)
            .content("Espere por favor...")
            .cancelable(false)
            .progress(true, 0)
            .show();
    } else {
        if (mProgress != null) {
            mProgress.dismiss();
        }
    }
}

class DoLoginUsuario extends AsyncTask<String, Void, Object> {

    @Override
    protected void onPreExecute() {

        showProgressIndicator(true);
    }

    @Override
    protected Object doInBackground(String... params) {
        Object requestObject = null;
        String url = params[0];
        Log.d(DBHelper.TAG, url);
        RestConnector restConnector;
        try {
            if (Funciones.hasActiveInternetConnection(getApplicationContext())) {
                restConnector = RestUtil.obtainGetConnection(url);
                requestObject = restConnector.doRequest(url);
            }

        } catch (Exception e) {
            requestObject = e;
        }
        return requestObject;
    }

    @Override
    protected void onPostExecute(Object result) {
        String response;
        showProgressIndicator(false);
        if (result instanceof String) {
            response = (String) result;
            processResponse(response);
        } else if (result instanceof Exception) {
            if (result instanceof HttpResponseException) {
                response = "Error en Conectar Ambiente";
            } else {

```



```

        response = ((Exception) result).getMessage();
    }
    Log.d(DBHelper.TAG, "Error en Iniciar sesion: " + response);
    Toast.makeText(LoginActivity.this, response, Toast.LENGTH_LONG)
        .show();
    }
}

}

class DoAceptar extends AsyncTask<String, Void, Object> {
    @Override
    protected void onPreExecute() {
        showProgressIndicator(true);
    }

    @Override
    protected Object doInBackground(String... params) {
        Object requestObject = null;
        String url = params[0];
        Log.d(DBHelper.TAG, url);
        RestConnector restConnector;
        try {
            if (Funciones.hasActiveInternetConnection(getApplicationContext())) {
                restConnector = RestUtil.obtainGetConnection(url);
                requestObject = restConnector.doRequest(url);
            }
        } catch (Exception e) {
            requestObject = e;
        }
        return requestObject;
    }

    @Override
    protected void onPostExecute(Object result) {
        String response;
        showProgressIndicator(false);
        if (result instanceof String) {
            response = (String) result;
            insertSettings(response);
        } else if (result instanceof Exception) {
            response = ((Exception) result).getMessage();
            Log.d(DBHelper.TAG, "Se produjo la excepción: " + response);
            Toast.makeText(LoginActivity.this, response, Toast.LENGTH_LONG)
                .show();
        }
    }
}
}

```

➤ **fragment_configurar_conexion_dialog.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"

    android:paddingBottom="@dimen/abc_list_item_padding_horizontal_material"
    android:paddingEnd="@dimen/waiter_detail_horizontal_padding"
    android:paddingLeft="@dimen/waiter_detail_horizontal_padding"
    android:paddingRight="@dimen/waiter_detail_horizontal_padding"
    android:paddingStart="@dimen/waiter_detail_horizontal_padding"

    android:paddingTop="@dimen/abc_list_item_padding_horizontal_material">

    <TextView
        android:id="@+id/servidorLabel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/dialog_servidor_label"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <EditText
        android:id="@+id/servidorEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/form_horizontal_padding_small"
        android:ems="8"
        android:textColor="@color/secondary_text"
        android:hint="@string/servidor_hint"
        android:imeOptions="actionNext"
        android:inputType="textUri"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>
    </EditText>

    <TextView
        android:id="@+id/aplicacionLabel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/form_horizontal_padding_small"
        android:text="@string/dialog_aplicacion_label"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

    <EditText
        android:id="@+id/aplicacionEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/form_horizontal_padding_small"
        android:textColor="@color/secondary_text"
        android:ems="8"
        android:hint="@string/aplicacion_hint"
        android:imeOptions="actionNext"
        android:inputType="textNoSuggestions"
        android:textAppearance="@style/TextAppearance.AppCompat.Subhead">
```

```

</EditText>

<TextView
    android:id="@+id/ambienteLabel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/form_horizontal_padding_small"
    android:text="@string/dialog_ambiente_label"
    android:textAppearance="@style/TextAppearance.AppCompat.Subhead"/>

<EditText
    android:id="@+id/ambienteEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/secondary_text"
    android:layout_marginTop="@dimen/form_horizontal_padding_small"
    android:ems="8"
    android:hint="@string/ambiente_hint"
    android:imeOptions="actionDone"
    android:inputType="textNoSuggestions"
    android:textAppearance="@style/TextAppearance.AppCompat.Subhead">
</EditText>

</LinearLayout>

```

➤ ConfigurarConexionDialogFragment.java

```

public class ConfigurarConexionDialogFragment extends
AppCompatActivity {

    private SharedPreferences mPrefConexion;
    private EditText mServidorEditText;
    private EditText mAplicacionEditText;
    private EditText mAmbienteEditText;
    private MaterialDialog mProgress;
    private View mForm = null;

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        mForm = getActivity().getLayoutInflater().inflate(
            R.layout.fragment_configurar_conexion_dialog, null);
        mServidorEditText = (EditText)
mForm.findViewById(R.id.servidorEditText);
        mAplicacionEditText = (EditText)
mForm.findViewById(R.id.aplicacionEditText);
        mAmbienteEditText = (EditText)
mForm.findViewById(R.id.ambienteEditText);

        mPrefConexion =
getActivity().getSharedPreferences(ConexionSharedPref.NAME, 0);

        mostrarDatosConexion();
    }
}

```

```

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
builder.setTitle(R.string.dialog_conf_conexion_title);
builder.setView(mForm);
builder.setPositiveButton(R.string.aceptar,
    new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {

        }

    });
builder.setNegativeButton(R.string.cancelar, null);
return builder.create();
}

private void mostrarDatosConexion() {

mServidorEditText.setText(mPrefConexion.getString(ConexionSharedPref.SERVIDOR, ""));

mAplicacionEditText.setText(mPrefConexion.getString(ConexionSharedPref.APLICACION, ""));

mAmbienteEditText.setText(mPrefConexion.getString(ConexionSharedPref.NOMBRE_AMBIENTE, ""));
}

@Override
public void onStart() {
    super.onStart();
    AlertDialog d = (AlertDialog) getDialog();
    if (d != null) {
        Button positiveButton = d.getButton(Dialog.BUTTON_POSITIVE);
        positiveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Boolean wantToCloseDialog = false;
                String msg = "";
                if (!Funciones.isEditTextEmpty(mServidorEditText)) {
                    if (!Funciones.isEditTextEmpty(mAplicacionEditText)) {
                        if (!Funciones.isEditTextEmpty(mAmbienteEditText)) {
                            ConexionSharedPref.save(mPrefConexion,
                                mServidorEditText.getText().toString().trim(),
                                mAplicacionEditText.getText().toString().trim(),
                                mAmbienteEditText.getText().toString().trim(),
                                true);
                            probarConexion(mServidorEditText.getText().toString().trim(),
                                mAplicacionEditText.getText().toString().trim(),
                                mAmbienteEditText.getText().toString().trim()
                            );
                            wantToCloseDialog = true;
                        } else {
                            msg = "Debe ingresar un ambiente.";
                        }
                    }
                }
            }
        });
    }
}

```

```

        } else {
            msg = "Debe ingresar una aplicación.";
        }
    } else {
        msg = "Debe ingresar la URL del servidor.";
    }

    if (!wantToCloseDialog) {
        Toast.makeText(getActivity(), msg,
Toast.LENGTH_SHORT).show();
    }

    }
    });
}

}

public void conexionProbada(String msg) {
    Toast.makeText(getActivity(), msg, Toast.LENGTH_LONG).show();
    getDialog().dismiss();
}

@Override
public void onCancel(DialogInterface dialog) {
    super.onCancel(dialog);
}

private void probarConexion(String server, String app, String ambiente) {
    String url =
"http://%s/%s/api/restaurante/VerificarParamsConexion/?cadenaConexion=%s";
    try {
        String encodedAmbiente = URLEncoder.encode("Initial Catalog=" +
ambiente, "utf-8");
        String urlWithParams = String.format(url, server, app, encodedAmbiente);
        new ProbarConfigConexion().execute(urlWithParams);
    } catch (Exception e) {
        Toast.makeText(getActivity(), e.getMessage(),
Toast.LENGTH_LONG).show();
    }
}

class ProbarConfigConexion extends AsyncTask<String, Void, Object> {
    Context ctxt = ConfigurarConexionDialogFragment.this.getActivity();

    @Override
    protected void onPreExecute() {
        mProgress = new MaterialDialog.Builder(ctxt)
            .content("Espere por favor...")
            .cancelable(false)
            .progress(true, 0)
            .show();
    }

    @Override
    protected Object doInBackground(String... params) {

```

```

        Object requestObject = null;
        String url = params[0];
        Log.d(DBHelper.TAG, url);
        RestConnector restConnector;
        try {
            if
(Funciones.hasActiveInternetConnection(ctxt.getApplicationContext())) {
                restConnector = RestUtil.obtainGetConnection(url);
                requestObject = restConnector.doRequest(url);
            }

        } catch (Exception e) {
            requestObject = e;
        }
        return requestObject;
    }

    @Override
    protected void onPostExecute(Object result) {
        String response;
        if (mProgress != null) {
            mProgress.dismiss();
        }
        if (result instanceof String) {

            ConfigurarConexionDialogFragment.this.conexionProbada("Configuración de
conexión probada y guardada correctamente.");

        } else if (result instanceof Exception) {
            response = ((Exception) result).getMessage();
            Log.d(DBHelper.TAG, "Error al configurar la conexión: " + response);
            Toast.makeText(ctxt, response, Toast.LENGTH_LONG).show();
        }
    }
}
}

```

➤ menu_login.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.neversoft.smartwaiter.ui.LoginActivity">
    <item
        android:id="@+id/menuConfigurarConexion"
        android:icon="@drawable/ic_action_settings"
        android:orderInCategory="1"
        android:title="@string/menu_configurar_conex"
        app:showAsAction="never"/>
</menu>

```

➤ Web Service

```
[HttpGet]
public DataSet Conectar(string ambiente, string usuario,
    string cadenaConexion, string clave = "")
{
    Configuracion.PrepararAppConfig();
    Compañia ciaObj;
    Usuario usuarioObj;
    DataSet dsRetorno = new DataSet();
    DataTable dtOtros = new DataTable();
    DataTable dtCompañias = new DataTable();
    DataColumn dcCol;
    DataRow drFilaOtros;
    string nuevaCadena = null;
    ConfiguracionAD configuracionADobj;

    Encripta encrip = new Encripta();
    string encriptaCadena = null;

    try
    {
        Configuracion.CadenaServidor =
            ConfigurationManager.ConnectionStrings["ConexionServidor"]
                .ConnectionString;
        encriptaCadena =
            System.Web.Configuration.WebConfigurationManager.AppSettings
                .Get("Encriptado");
        if (encriptaCadena.ToUpper().Trim() == "S")
        {
            Configuracion.CadenaServidor =
                encrip.DesencriptarCadena(Configuracion.CadenaServidor);
        }

        dcCol = dtOtros.Columns.Add("ambiente",
            typeof(System.Boolean));
        dcCol.DefaultValue = false;
        dcCol = dtOtros.Columns.Add("usuario",
            typeof(System.Boolean));
        dcCol.DefaultValue = false;

        drFilaOtros = dtOtros.NewRow();

        nuevaCadena =
            Configuracion.CompletarCredenciales(cadenaConexion);

        configuracionADobj = new ConfiguracionAD(nuevaCadena);
        drFilaOtros["ambiente"] =
            configuracionADobj.ObtenerBase(nuevaCadena);
        usuarioObj = new Usuario(nuevaCadena);
        drFilaOtros["usuario"] =
            usuarioObj.VerificarLogin(usuario, clave);

        dtOtros.TableName = "TablaOtros";
        if (Convert.ToBoolean(drFilaOtros["usuario"]) == true)
```

```

        {
            ciaObj = new Compañia(nuevaCadena);
            dtCompañias = ciaObj.ObtenerCompañias();
            dtCompañias.TableName = "TablaCompañias";
            dsRetorno.Tables.Add(dtCompañias.Copy());
        }
        dtOtros.Rows.Add(drFilaOtros);
        dsRetorno.Tables.Add(dtOtros);
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return dsRetorno;
}

```

```

[HttpGet]
public dynamic Aceptar([FromUri] string usuario,
[FromUri] string codcia, [FromUri] string cadenaConexion)
{
    Configuracion.PrepararAppConfig();
    Mozo mozoObj;
    DataTable dtConfiguracion = new DataTable();
    Dictionary<string, string> dictionaryConfig = new
    Dictionary<string, string>();
    DataTable dtCia;
    DataTable dtImplicito;
    DataSet dsRetorno;
    dtConfiguracion.Columns.Add("Usuario");
    dtConfiguracion.Columns.Add("CodCia");
    dtConfiguracion.Columns.Add("IdRol");
    dtConfiguracion.Columns.Add("NombreCia");
    dtConfiguracion.Columns.Add("RucCia");
    dtConfiguracion.Columns.Add("CodAlm");
    dtConfiguracion.Columns.Add("MonedaCia");
    dtConfiguracion.Columns.Add("CondPago");
    dtConfiguracion.Columns.Add("CodCli");
    dtConfiguracion.Columns.Add("CodMozo");

    DataRow drConfiguracion = dtConfiguracion.NewRow();
    try
    {
        Configuracion.CadenaServidor =
        ConfigurationManager.ConnectionStrings["ConexionServidor"]
        .ConnectionString;
        cadenaConexion =
        Configuracion.CompletarCredenciales(cadenaConexion);
        mozoObj = new Mozo(cadenaConexion);
        if (LlamarObtenerUsuariosCompañia(usuario, codcia,
        cadenaConexion) > 0)
        {

```



```

drConfiguracion["Usuario"] = usuario;
dsRetorno = LlamarAceptar(usuario, codcia,
cadenaConexion);
if (usuario != "SUPERVISOR")
{
    if (dsRetorno.Tables.Contains("dtRoles"))
    {
        drConfiguracion["IdRol"] =
dsRetorno.Tables["dtRoles"].Rows[0]["idrol"];
    }
    else
    {
        drConfiguracion["IdRol"] = 0;
    }
}
dtImplicito = dsRetorno.Tables["dtImplicito"];
if (dtImplicito.Rows.Count == 0)
{
    throw new Exception("No existe valores
    implicitos para usuario " + usuario);
}
dtCia = dsRetorno.Tables["dtCia"];

drConfiguracion["CodCia"] =
dtCia.Rows[0]["codcia"];
drConfiguracion["NombreCia"] =
dtCia.Rows[0]["razonsocial"];
drConfiguracion["RucCia"] = dtCia.Rows[0]["ruc"];
drConfiguracion["CodAlm"] =
dtImplicito.Rows[0]["codalm"];
drConfiguracion["MonedaCia"] =
dtCia.Rows[0]["codmon"];
drConfiguracion["CondPago"] =
dtImplicito.Rows[0]["conpag"];
drConfiguracion["CodCli"] =
dtImplicito.Rows[0]["codcli"];
drConfiguracion["CodMozo"] =
mozoObj.ObtenerCodigoMozo(usuario);
dtConfiguracion.Rows.Add(drConfiguracion);

foreach (DataRow dr in dtConfiguracion.Rows)
{
    foreach (DataColumn dc in dr.Table.Columns)
    {
        dictionaryConfig.Add(dc.ColumnName.ToString
        (), dr[dc].ToString());
    }
}
dictionaryConfig.ToArray();
}
else
{
    throw new Exception("Usuario no tiene acceso a
    esta compañía");
}
}

```

```
        catch (Exception ex)
        {
            throw ex;
        }
    return dictionaryConfig.ToArray();
}
```

4.6.6. SPRINT 6

4.6.6.1. CALCULO LA VELOCIDAD ESTIMADA PARA EL SEXTO SPRINT

- ✓ Calculando duración del sprint:
5 semanas= 25 días laborables.
- ✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	25
Programador 2	25
Total	50 días hombres disponible

Tabla 60. Calculo Días Hombre - Sexto Sprint

- ✓ Calculando factor de dedicación (FD):
 $FD = \text{Puntos de historia (último sprint)} / \text{días hombres disponible}$
 $FD = 28/50 = 0.56$
- ✓ Velocidad estimada: $FD \times \text{Días –Hombre- Disponible}$
 $0.56 * 50 = 28$

4.6.6.2. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL SEXTO SPRINT

N°	SPRINT 6	PUNTO S	ESTADO
1	LISTADO GENERAL ACTUALIZADO DE MESAS	8	TERMINADO
2	BÚSQUEDA DE RESERVAS Y LISTADO	7	TERMINADO

	ACTUAL DE MESAS RESERVADAS		
3	SELECCIÓN DE MESA Y ACTUALIZACIÓN DE SU ESTADO EN EL DISPOSITIVO Y EN EL SERVIDOR	6	TERMINADO
4	LISTADO DE CATEGORÍAS	4	TERMINADO
5	LISTADO DE ARTÍCULOS DE LA CATEGORÍA SELECCIONADA	4	TERMINADO
6	MANTENIMIENTO DE PEDIDO EN EL DISPOSITIVO	10	TERMINADO
7	ENVÍO DEL PEDIDO AL SERVIDOR PARA SU PROCESAMIENTO	10	TERMINADO
	TOTAL DE PUNTOS DE HISTORIA	49	49
Objetivo	<ul style="list-style-type: none"> • Buscar mesas disponibles. • Buscan mesas reservadas. • Registrar pedido. • Enviar pedido registrado para su procesamiento. 		

Tabla 61. Productos en el Sexto Sprint

4.6.6.3. DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Mozo
Nombre de Historia: Listado general actualizado de mesas	
Prioridad en Negocio: alta	
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá que cada vez que se acceda a la opción “Realizar Pedido”, se consulte el estado actual de las mesas en el servidor y se actualiza la base local de acuerdo a dicha información, de manera tal que, al seleccionar una mesa, esta refleje su verdadero estado.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none">✓ Verificar que al seleccionar la opción “Realizar Pedido”, se mostrará la pantalla que permitirá seleccionar la mesa que se asignará a los comensales.✓ Verificar que la pantalla, además de mostrar un listado general de las mesas, permitirá filtrar los resultados por piso y ambiente.✓ Verificar que ya sea al seleccionar la opción “Realizar Pedido” o al cambiar los filtros de búsqueda, la lista de mesas volverá a ser cargada para reflejar los estados actuales de las mesas.	
<p>Observaciones: Previamente se debe haber realizado el proceso de sincronización que comprende el registro de:</p>	

<ul style="list-style-type: none"> • Artículos • Carta • Categorías • Mesas • Clientes 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 62. 1° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Mozo
Nombre de Historia: Búsqueda de reservas y listado actual de mesas reservadas	
Prioridad en Negocio: alta	
Puntos estimados: 7	Iteración asignada: 1
Programador responsable: José Huamán Valqui	

<p>Descripción:</p> <p>Esta opción permitirá realizar búsquedas de reservas de mesas, tanto por código de reserva como por número de documento del cliente que la efectuó.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Reservas”, se mostrará pantalla que permitirá realizar búsquedas de reservas. ✓ Verificar que se puede realizar busque de reservas tanto por código de reserva como por número de documento del cliente que la efectuó. 	
<p>Observaciones: Previamente se debe haber realizado el proceso de sincronización que comprende el registro de :</p> <ul style="list-style-type: none"> • Artículos • Carta • Categorías • Mesas • Clientes 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui

Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 63. 2° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Mozo
Nombre de Historia: Selección de mesa y actualización de su estado en el dispositivo y en el servidor	
Prioridad en Negocio: alta	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá que al seleccionar una mesa, libre o reservada, el estado de esta se actualice en el servidor, permitiendo que todos los dispositivos que consulten su estado tengan acceso a su valor real.</p> <p>Además de actualizar el estado de la mesa seleccionada, esta acción permitirá dirigir al usuario hacia la pantalla de toma de pedido propiamente dicha.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar una mesa (libre o reservada), y tras confirmación la acción, el estado de la mesa será actualizado al estado “ocupado”. ✓ Verificar que después de haber seleccionado una mesa, se mostrará la actividad que permitirá realizar la toma de pedido. 	

Observaciones:	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 64. 3° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Mozo
Nombre de Historia: Listado de categorías	
Prioridad en Negocio: alta	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Dentro de la pantalla de toma de pedido, este proceso permitirá mostrar la relación de categorías de productos disponibles. Además, permitirá descargar y mostrar de manera paralela las imágenes de las categorías listadas.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que, al mostrarse la pantalla de toma de pedido, se mostrará la relación de categorías existente. 	

✓ Verificar que las imágenes que identifican a las categorías se irán mostrando paulatinamente, conforme son descargadas.	
<p>Observaciones: Previamente se debe haber realizado el proceso de sincronización que comprende el registro de:</p> <ul style="list-style-type: none"> • Artículos • Carta • Categorías • Mesas • Clientes 	
Tareas	Responsable
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 65. 4° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 5	Usuario: Mozo
Nombre de Historia: Listado de artículos de la categoría seleccionada	
Prioridad en Negocio: alta	
Puntos estimados: 4 Iteración asignada: 1	
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Dentro de la pantalla de toma de pedido, este proceso permitirá mostrar la relación de artículos disponibles dentro de la categoría seleccionada.</p> <p>Además permitirá descargar y mostrar de manera paralela las imágenes de los artículos listados.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que seleccionar una categoría en la pantalla de toma de pedido, se mostrará la relación de artículos disponibles para esa categoría. ✓ Verificar que las imágenes que identifican a los artículos se irán mostrando paulatinamente, conforme son descargadas. 	
<p>Observaciones: Previamente se debe haber realizado el proceso de sincronización que comprende el registro de :</p> <ul style="list-style-type: none"> • Artículos • Carta • Categorías • Mesas 	

<ul style="list-style-type: none"> Clientes 	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 66. 5° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 6	Usuario: Mozo
Nombre de Historia: Mantenimiento de pedido en el dispositivo	
Prioridad en Negocio: alta	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
Descripción: Dentro de la pantalla de toma de pedido, este proceso permitirá agregar nuevos artículos, aumentar o disminuir la cantidad de los mismos, quitar artículos de un pedido, o cancelarlo.	
Pruebas:	

- ✓ Verificar que al seleccionar un artículo este será agregado al pedido si es que no ha sido agregado previamente.
- ✓ Verificar que, al seleccionar un artículo, si esta ya existe en el pedido, solo se aumentará su cantidad en una unidad.
- ✓ Verificar que al mantener seleccionado un artículo dentro del pedido, se mostrara la barra de acción.
- ✓ Dentro de la barra de acción, verificar que al presionar el botón con el icono de suma (+) se aumentará en una unidad la cantidad del articulo seleccionado.
- ✓ Dentro de la barra de acción, verificar que al presionar el botón con el icono de resta (-) se disminuirá en una unidad la cantidad del articulo seleccionado.
- ✓ Dentro de la barra de acción, verificar que al presionar el botón con el icono de un bote de basura el artículo seleccionado será removido del pedido.
- ✓ Verificar que al seleccionar la acción “Cancelar”, y luego de confirmar la acción, el pedido será eliminado, el estado de la mesa pasara a “Libre” y el usuario será redirigido a donde seleccionó la mesa, ya sea “Seleccionar Mesa” o a “Buscar Reserva”

Observaciones: Previamente se debe haber realizado el proceso de sincronización que comprende el registro de: <ul style="list-style-type: none"> • Artículos • Carta • Categorías • Mesas • Clientes 	
Tareas	Responsable
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 67. 6° Historia de Usuario - Sexto Sprint

HISTORIA DE USUARIO	
Número: 7	Usuario: Mozo
Nombre de Historia: Envío del pedido al servidor para su procesamiento	
Prioridad en Negocio: baja	
Puntos estimados: 10	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
Descripción: Esta opción permitirá guardar el pedido en el dispositivo y enviarlo al servidor para su procesamiento.	

<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la acción “Enviar”, se iniciará el proceso de envío del pedido. ✓ Verificar que, al término del envío del pedido, este aparecerá dentro de los pedidos listados en la opción de menú “Restaurante/Seguimiento de Pedidos”. ✓ Verificar que, al término del envío del pedido, el usuario será redirigido a donde seleccionó la mesa, ya sea “Seleccionar Mesa” o a “Buscar Reserva” 	
<p>Observaciones: Para realizar el envío de un pedido, este debe contar con al menos un ítem.</p>	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 68. 7° Historia de Usuario - Sexto Sprint

4.6.6.4. DIAGRAMA DE CLASES SPRINT 6

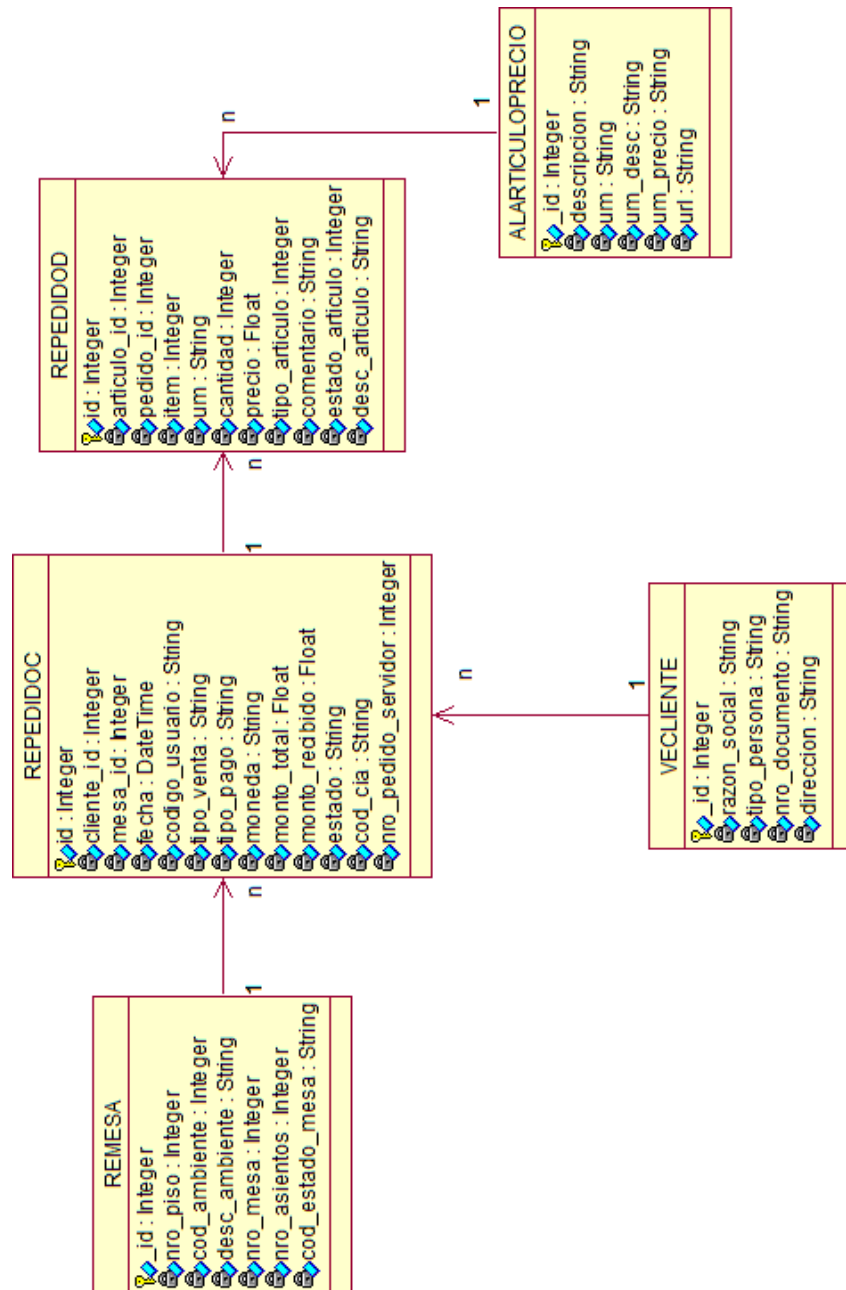


Ilustración 155 Diagrama de Clases – Sexto Sprint

4.6.6.5. DIAGRAMA LOGICO Y FISICO SPRINT 6

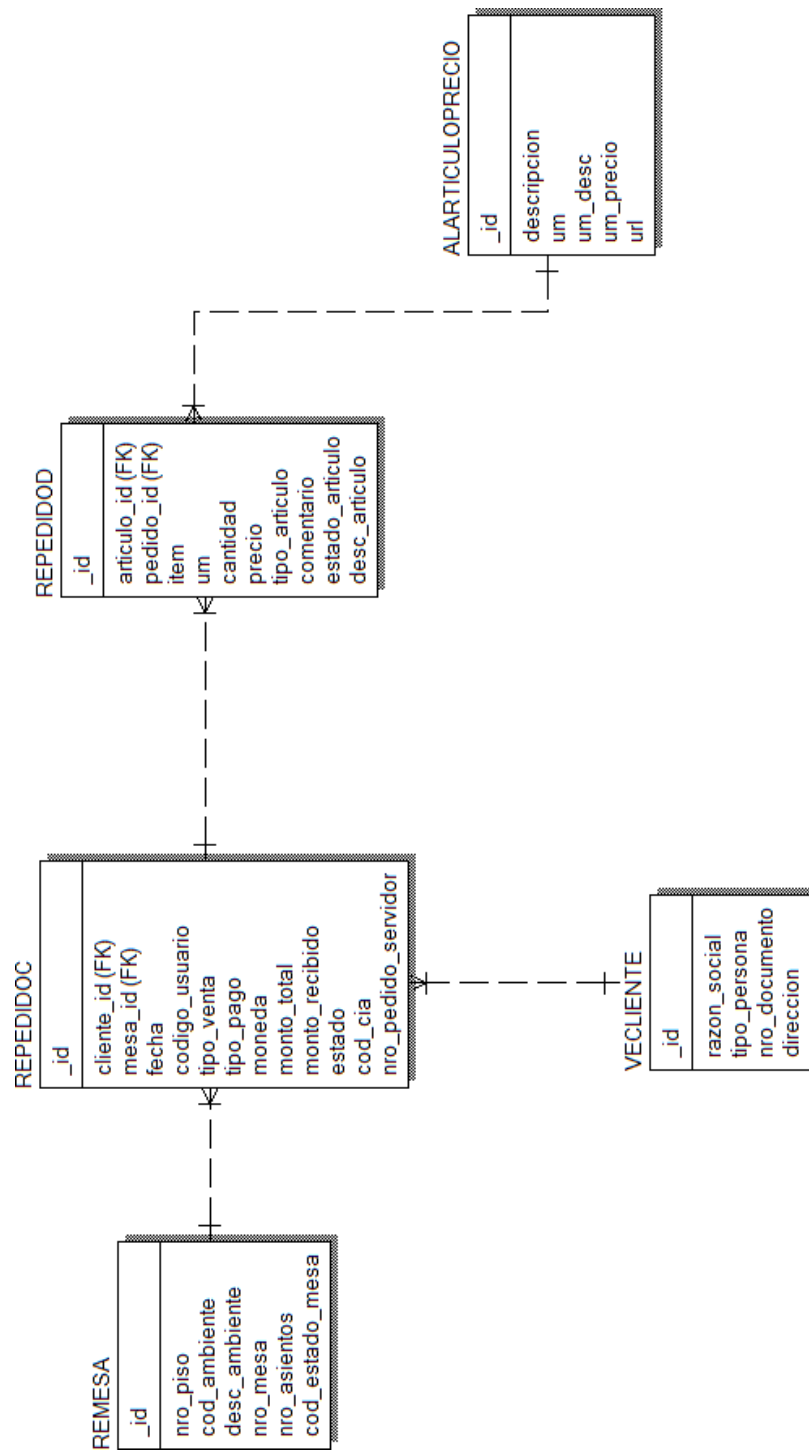


Ilustración 156 Diagrama Físico - Sexto Sprint

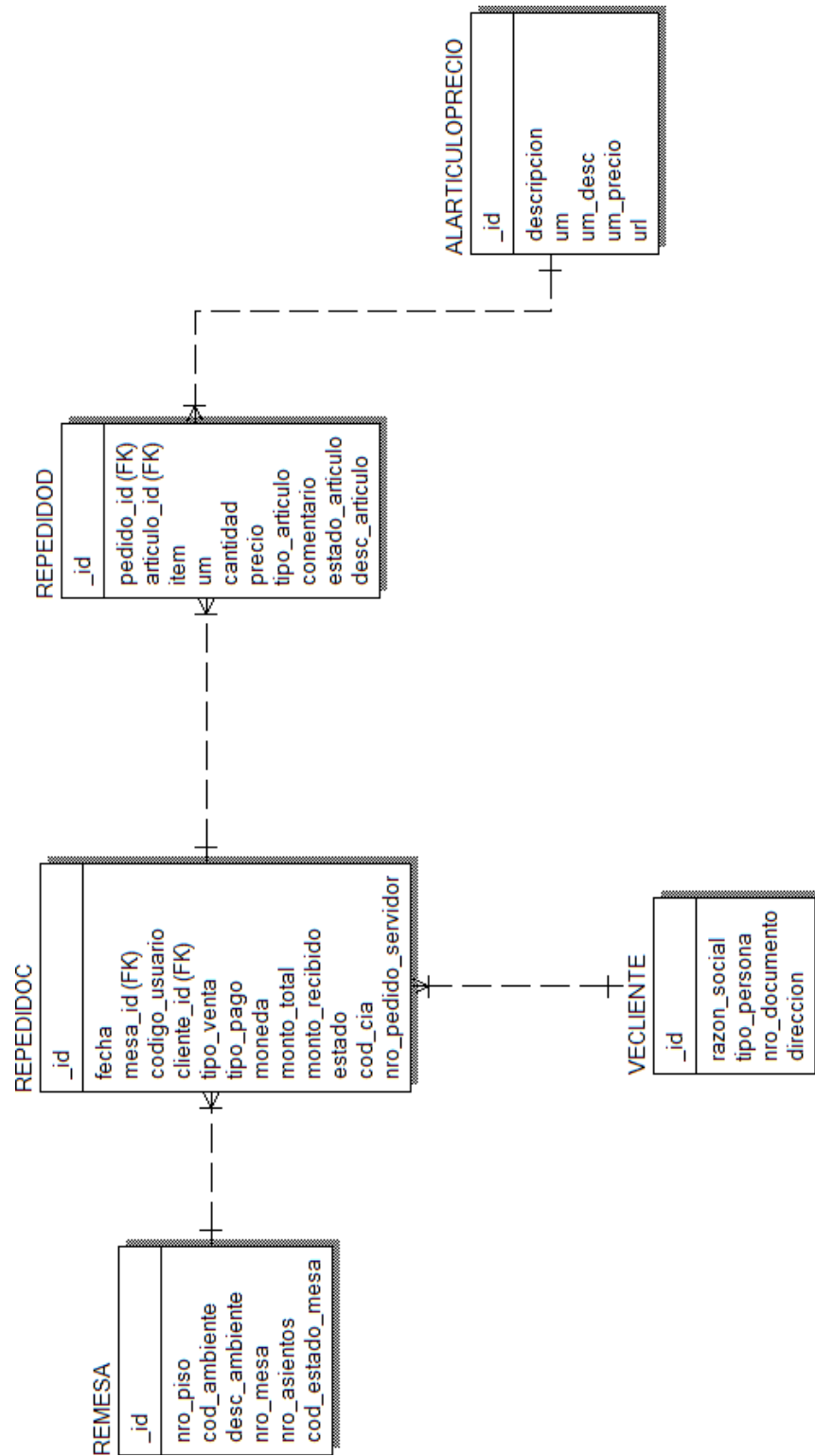


Ilustración 157 Diagrama Lógico - Sexto Sprint

4.6.6.6. DIAGRAMA DE COMPONENTES SPRINT 6

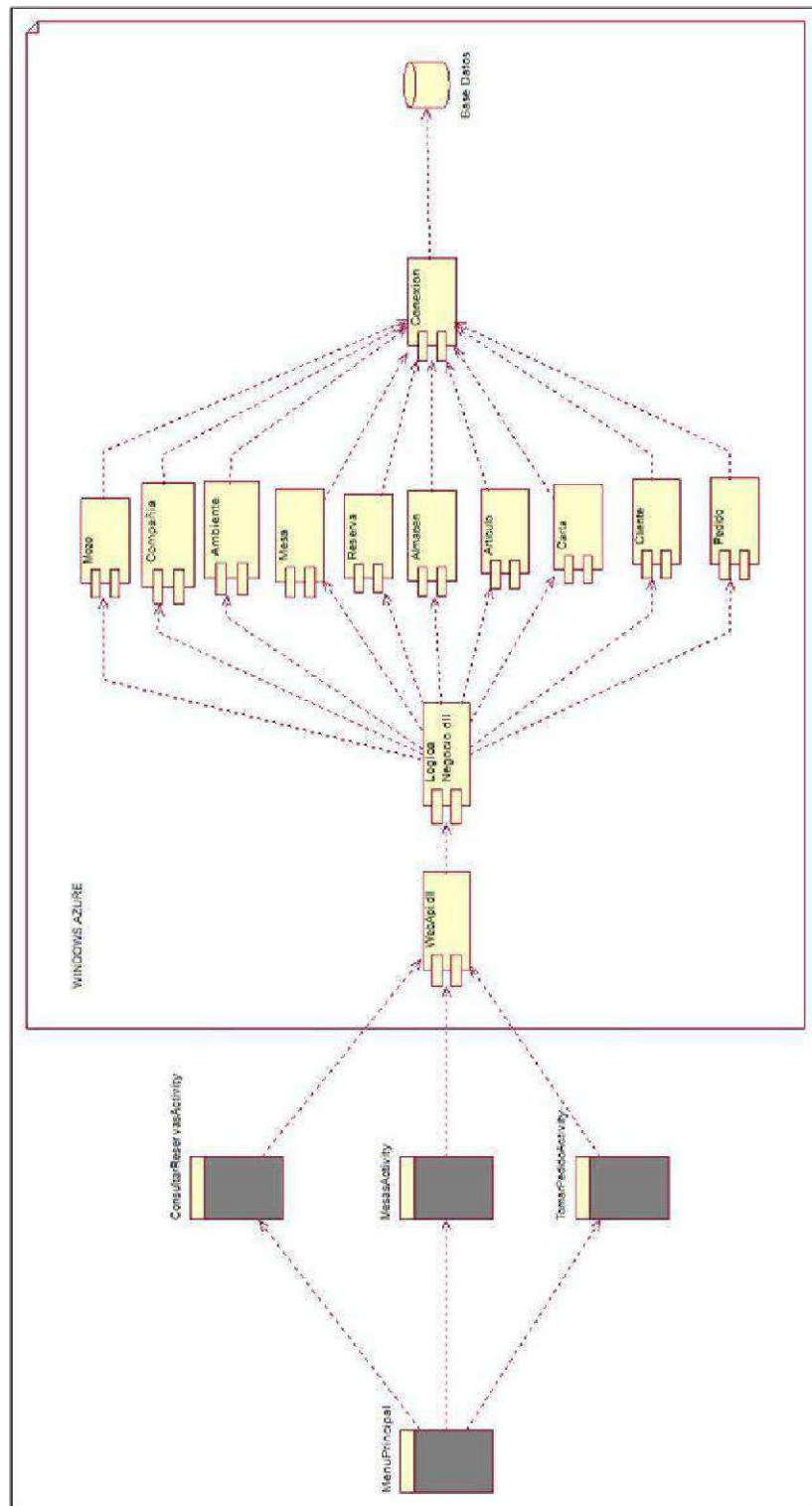


Ilustración 158 Diagrama de Componentes – Sexto Sprint

4.6.6.7. INTERFAZ DE LAS HISTORIAS DE USUARIO

- ✓ Listado general actualizado de mesas:

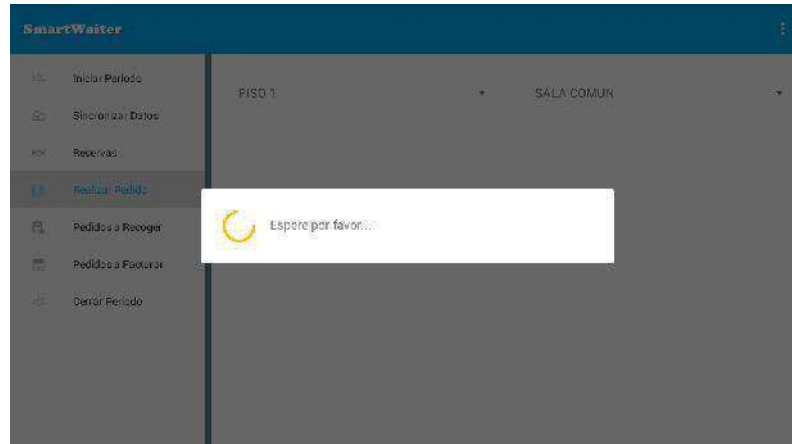


Ilustración 159 Listado general actualizado de mesas. – Sexto Sprint

- ✓ Búsqueda de reservas y listado actual de mesas reservadas:

The interface shows a sidebar with navigation options: Iniciar Período, Sincronizar Datos, **Reservas**, Realizar Pedido, Pedidos a Recoger, Pedidos a Facturar, and Cerrar Período. The main area contains a search form with the following fields:

- Código Reserva
- ID Cliente: 20486245027
- Razón Social
- DNI/RUC

The interface shows the same search form as above, but with a loading spinner overlay in the center that says "Espere por favor!!".

The interface shows the same search form as above, but with a table of reserved tables displayed below the form.

2 RESERVADO 27	3 RESERVADO 28	4 RESERVADO 29
2 RESERVADO 30	5 RESERVADO 31	6 RESERVADO 32

Ilustración 160 Búsqueda de reservas y listado actual de mesas reservadas – Sexto Sprint

- ✓ Selección de mesa y actualización de su estado en el dispositivo y en el servidor

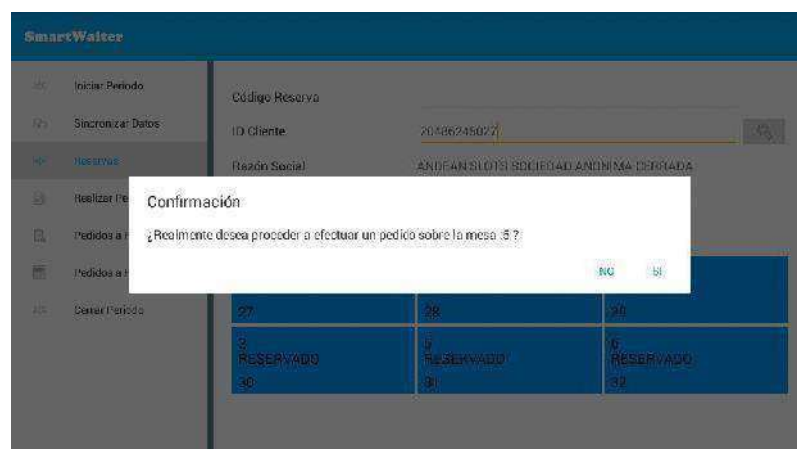


Ilustración 161 Interfaz Selección de mesa y actualización de su estado en el dispositivo y en el servidor– Sexto Sprint

✓ Listado de categorías



Ilustración 162 Listado de categorías– Sexto Sprint

✓ Listado de artículos de la categoría seleccionada

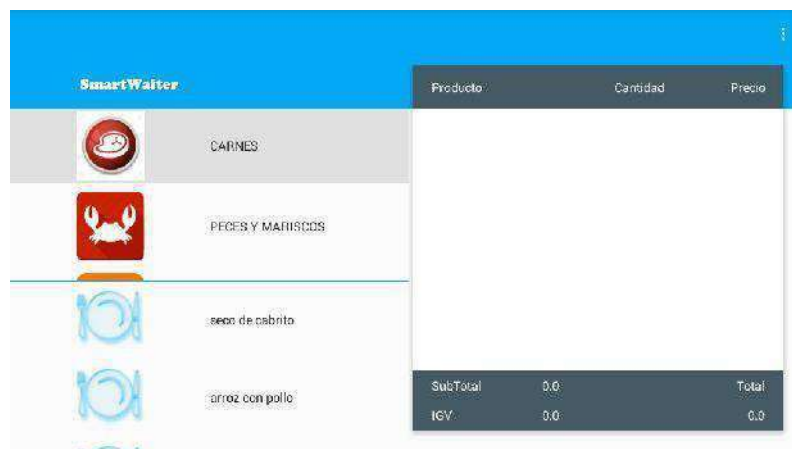






Ilustración 163 Listado de artículos de la categoría seleccionada– Sexto Sprint

✓ Mantenimiento de pedido en el dispositivo

SmartWaiter		Producto	Cantidad	Precio
	CARNES	seco de cabrito	1.0	14.0
		arroz con pollo	1.0	24.0
	PECES Y MARISCOS			
	seco de cabrito			
	arroz con pollo			
		SubTotal	38.0	Total
		IGV	7.22	45.22

SmartWaiter		Producto	Cantidad	Precio
	CARNES	seco de cabrito	1.0	14.0
		arroz con pollo	1.0	24.0
	PECES Y MARISCOS			
	seco de cabrito			
	arroz con pollo			
		SubTotal	38.0	Total
		IGV	7.22	45.22





SmartWaiter		Producto	Cantidad	Precio
	CARNES	seco de cabrito	1.0	14.0
		arroz con pollo	1.0	24.0
	PECES Y MARISCOS			
	seco de cabrito			
	arroz con pollo			
		SubTotal	38.0	Total
		IGV	7.22	45.22

Ilustración 164 Mantenimiento de pedido en el dispositivo– Sexto Sprint

- ✓ Registrar cliente envío del pedido al servidor para su procesamiento

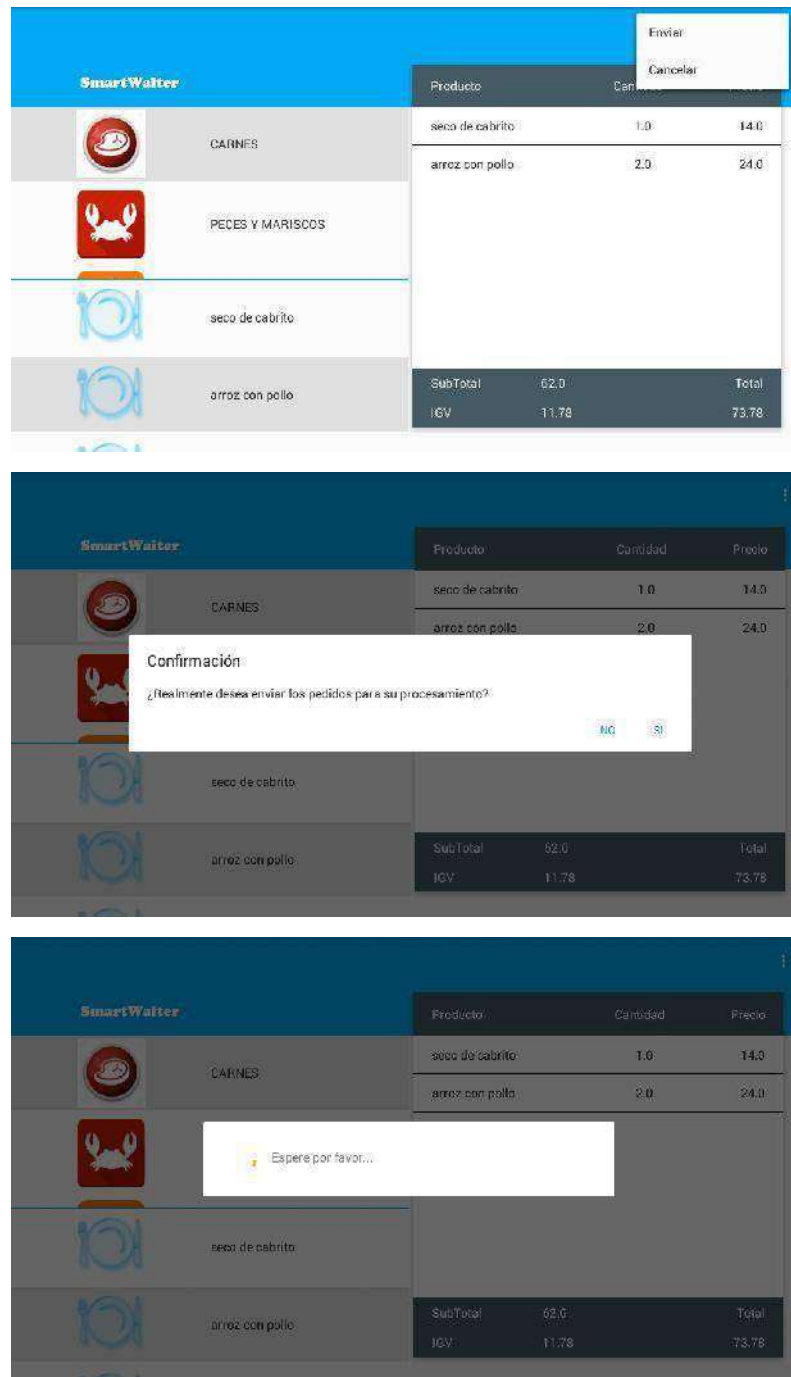


Ilustración 165 Registrar cliente envío del pedido al servidor para su procesamiento – Sexto Sprint

4.6.6.8. GRAFICANDO EL TABLÓN DE TAREAS

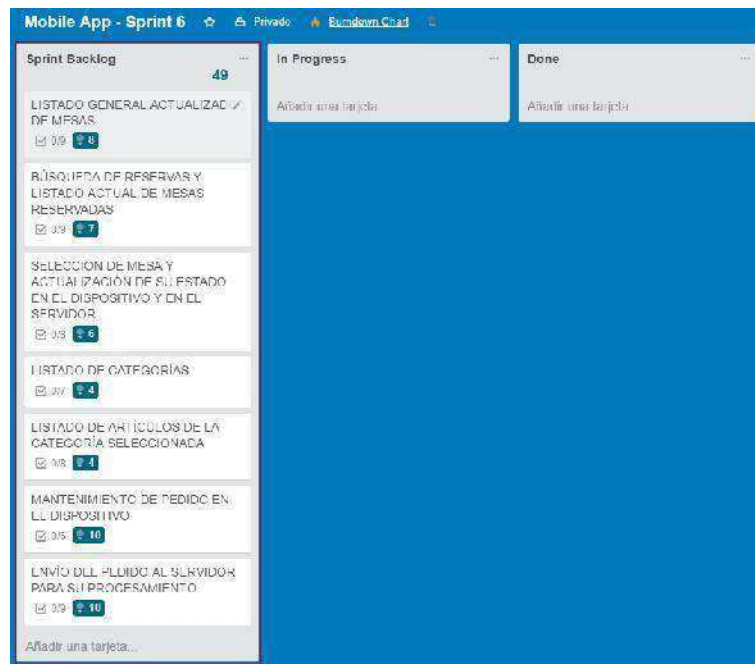


Ilustración 166 Historias de Usuario –Sexto Sprint

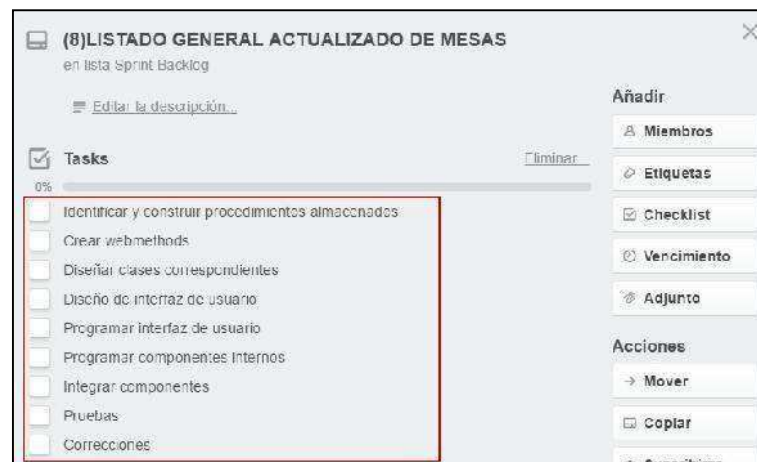


Ilustración 167 H. de Usuario Listado General Actualizado de Mesas

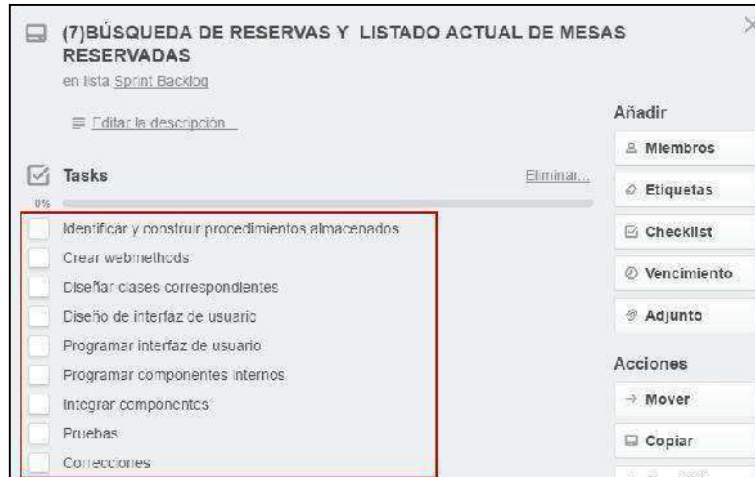


Ilustración 168 H. de Usuario Búsqueda de Reservas y Listado Actual de Mesas Reservadas

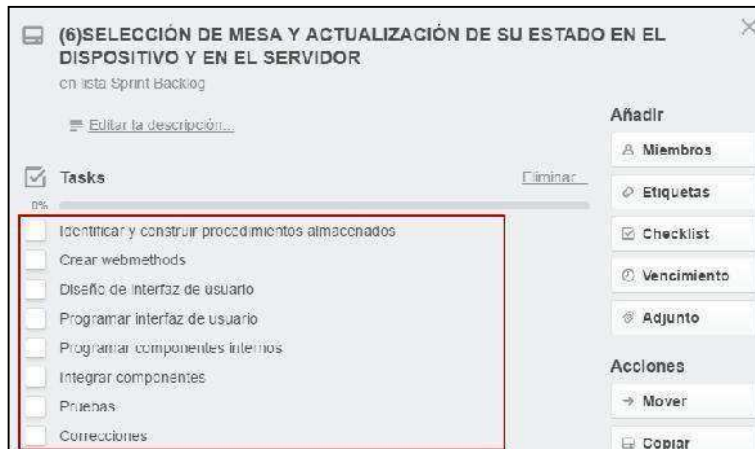


Ilustración 169 H. de Usuario Selección de Mesa y Actualización de su Estado en el Dispositivo y en el Servidor



Ilustración 170 H. de Usuario Listado de Categorías

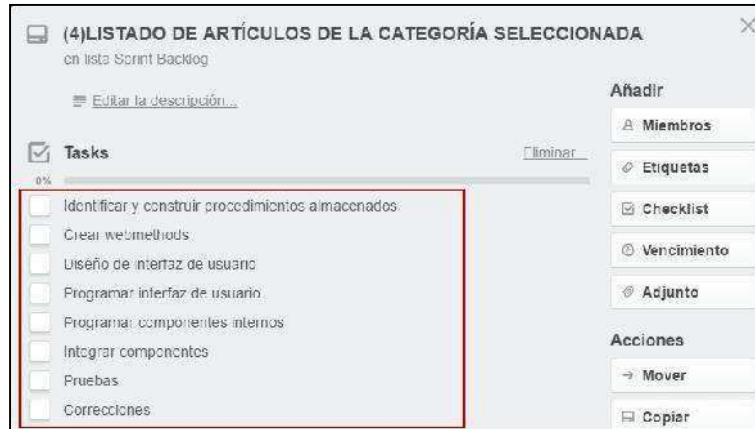


Ilustración 171 H. de Usuario Listado de Artículos de La Categoría Seleccionada

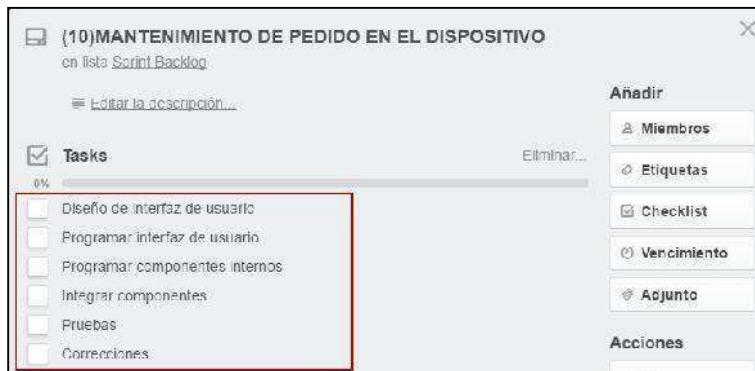


Ilustración 172 H. de Usuario Mantenimiento de Pedido en el Dispositivo

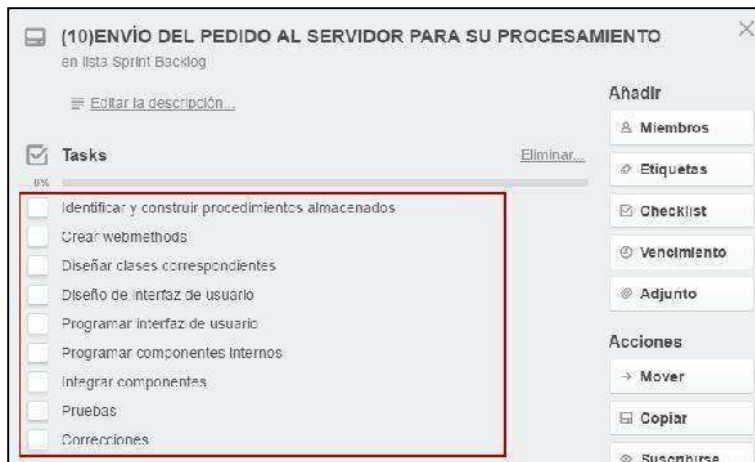


Ilustración 173 H. de Usuario Envío del Pedido al Servidor para su Procesamiento

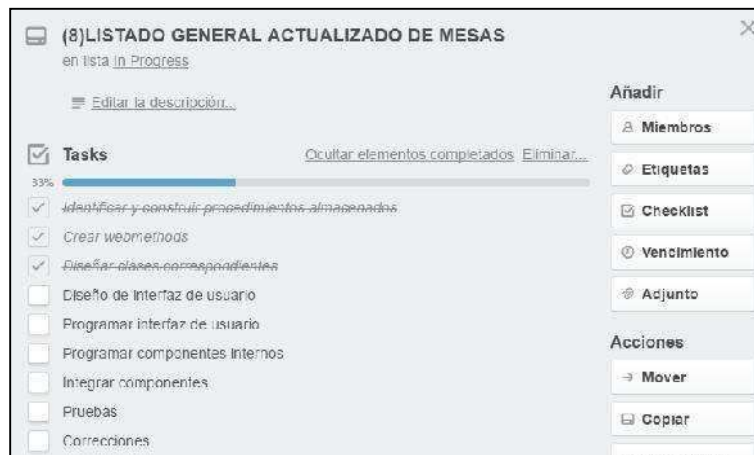
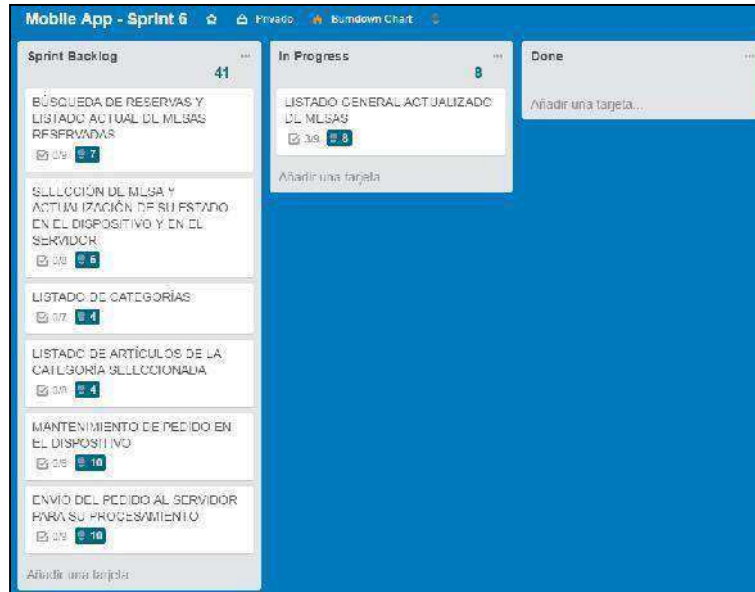


Ilustración 174 Proceso de Tareas Historia 1 – Sexto Sprint

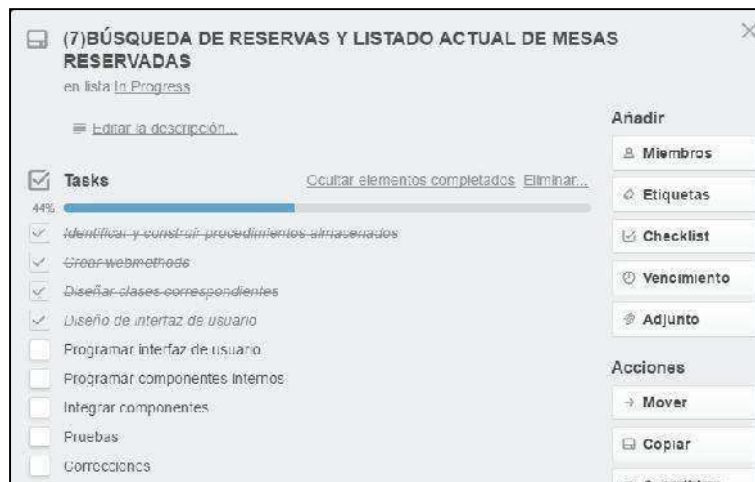
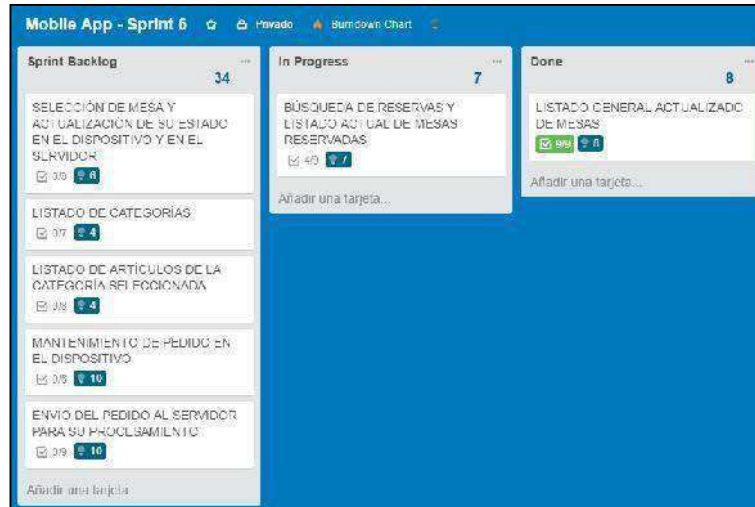
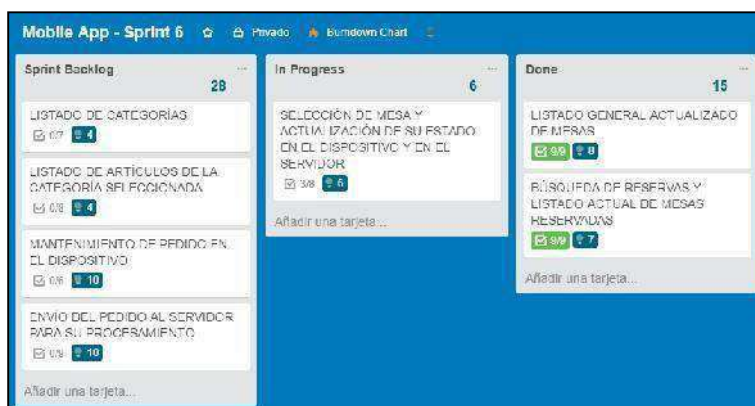


Ilustración 175 Proceso de Tareas Historia 2 – Sexto Sprint



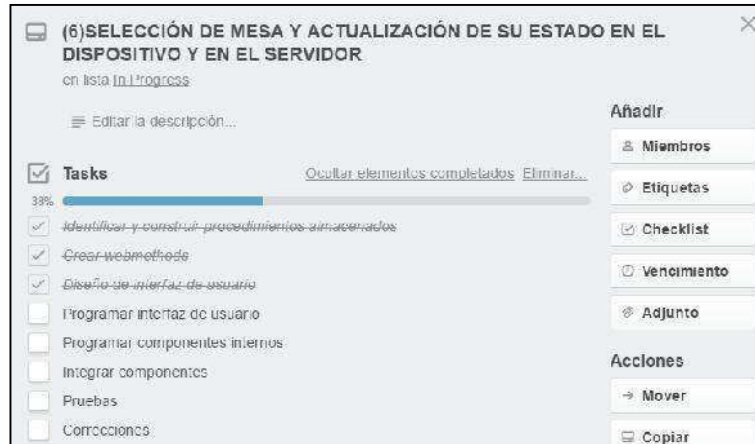


Ilustración 176 Proceso de Tareas Historia 3 – Sexto Sprint



Ilustración 177 Proceso de Tareas Historia 4 – Sexto Sprint

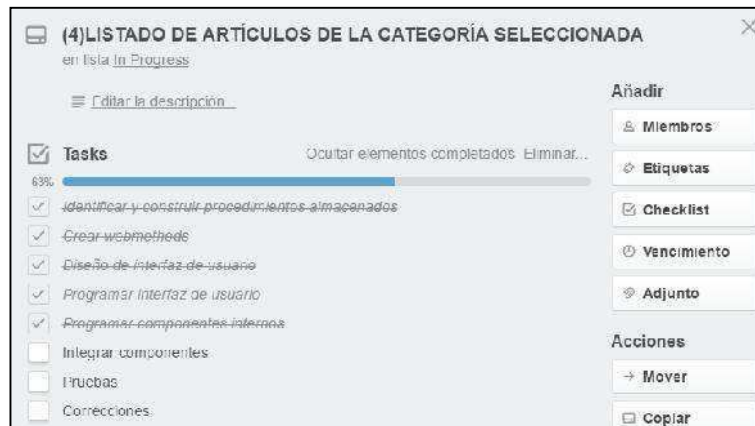
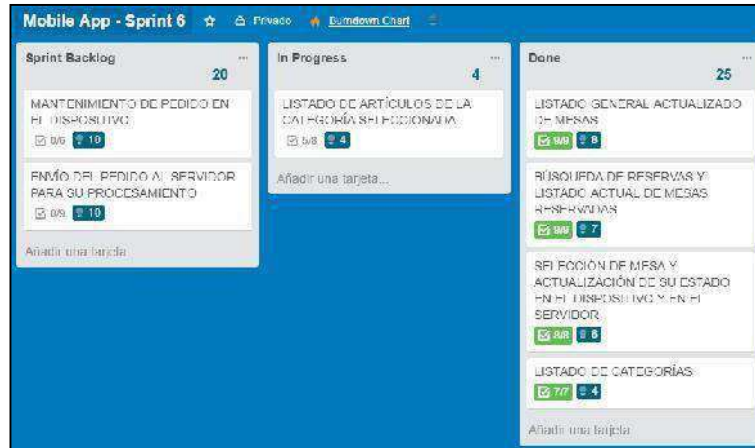


Ilustración 178 Proceso de Tareas Historia 5 – Sexto Sprint

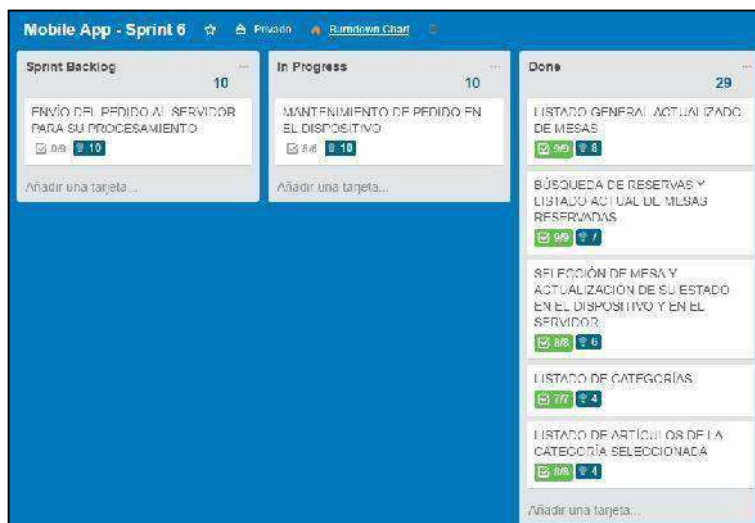




Ilustración 179 Proceso de Tareas Historia 6 – Sexto Sprint

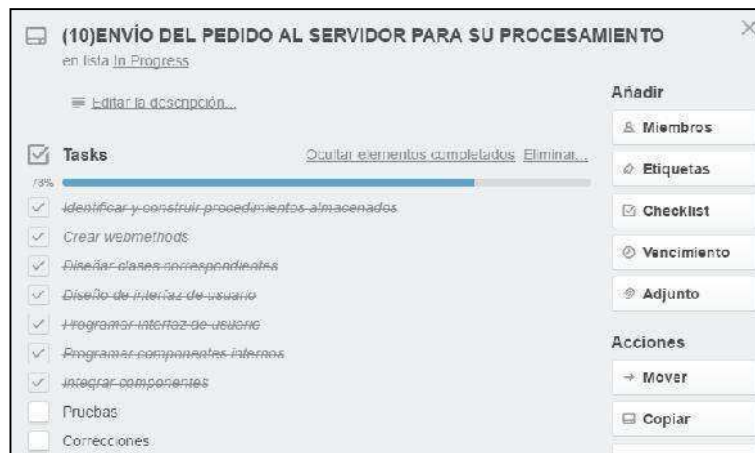
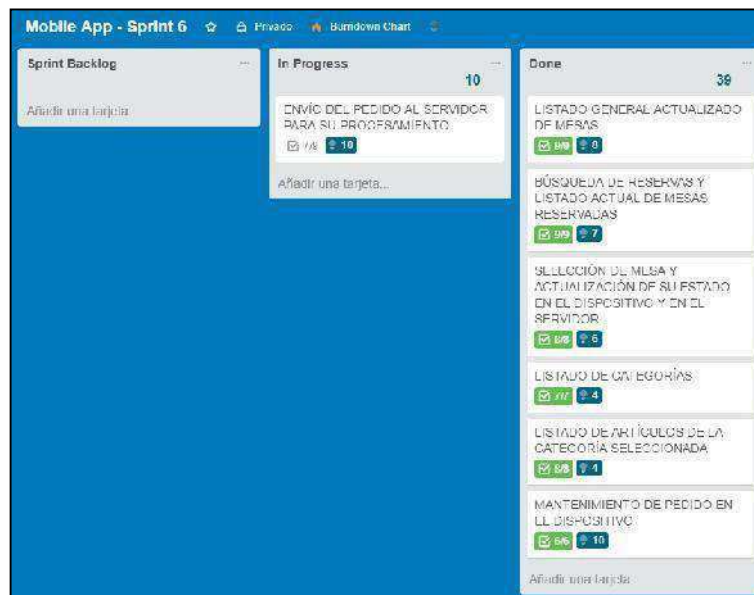


Ilustración 180 Proceso de Tareas Historia 7 – Sexto Sprint

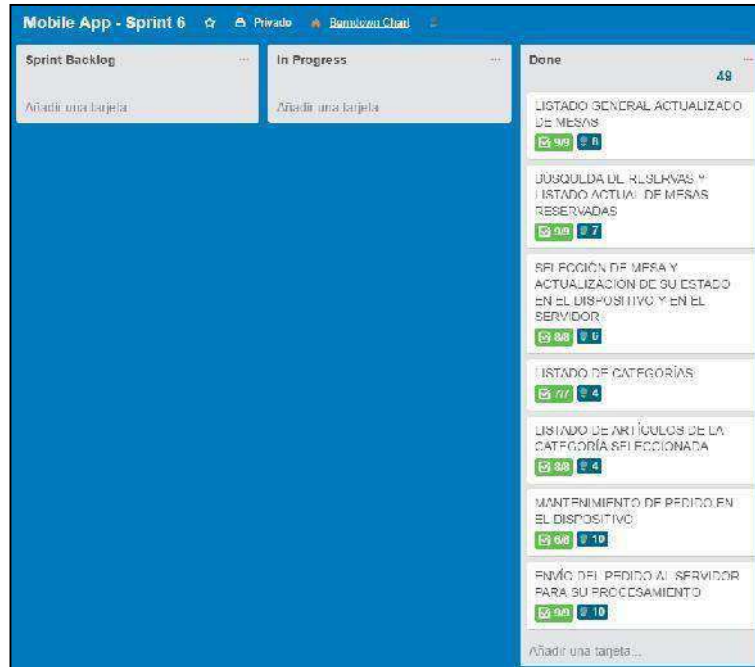


Ilustración 181 Historias de Usuario Concluidas – Sexto Sprint

Historias de Usuario Concluidas – Sexto Sprint

4669. DIAGRAMA DE BURN DOWN

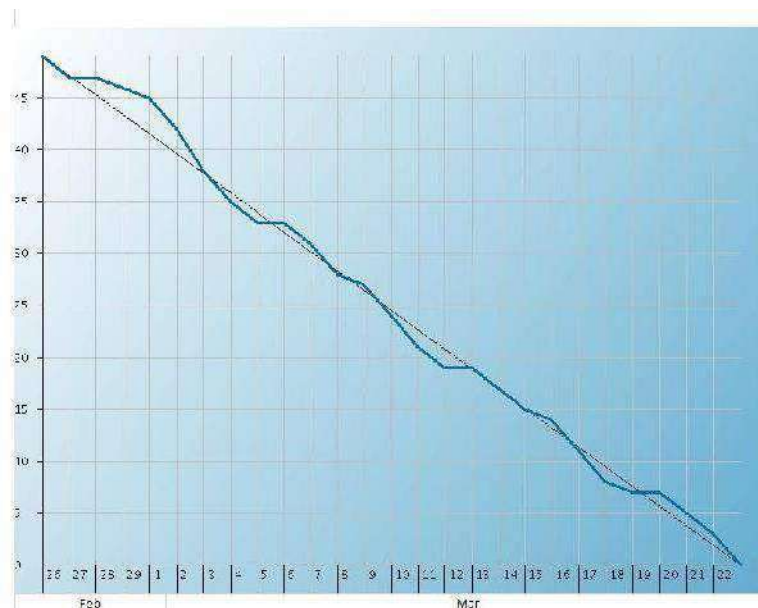


Ilustración 182 Diagrama de Burn Down –Sexto Sprint

4.6.6.10. DESARROLLO DE HISTORIAS

Seleccionar Mesa

➤ activity_mesas.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.neversoft.smartwaiter.ui.MesasActivity">

    <include
        layout="@layout/toolbar"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="7">

        <include layout="@layout/navigation_view"/>

        <android.support.v7.widget.GridLayout
            android:id="@+id/mainLinearLayout"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="4"
            android:background="@color/detail_accent_pane_background_sec"
            android:paddingLeft="@dimen/form_horizontal_padding_small"
            app:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:background="@color/detail_accent_pane_background"
                android:gravity="center_vertical"
                android:padding="@dimen/abc_list_item_padding_horizontal_material"
                app:layout_rowWeight="1">

                <Spinner
                    android:id="@+id/pisos_spinner"
                    android:layout_width="0dp"
                    android:textColor="@android:color/white"
                    android:layout_height="wrap_content"
                    android:background="@android:color/transparent"

                    android:layout_marginRight="@dimen/abc_list_item_padding_horizontal_material"

                    android:layout_weight="1"/>

                <Spinner
                    android:id="@+id/ambientes_spinner"
                    android:layout_width="0dp"
```

```

        android:textColor="@android:color/white"
        android:layout_height="wrap_content"
        android:background="@android:color/transparent"

    android:layout_marginLeft="@dimen/abc_list_item_padding_horizontal_material"
        android:layout_weight="1"/>
</LinearLayout>

<GridView
    android:id="@+id/mesasGridView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:background="@color/detail_accent_pane_background"
    android:columnWidth="200dp"
    android:horizontalSpacing="4dp"
    android:numColumns="auto_fit"
    android:padding="@dimen/abc_list_item_padding_horizontal_material"

    android:paddingTop="@dimen/abc_list_item_padding_horizontal_material"
    android:stretchMode="columnWidth"
    android:verticalSpacing="4dp"
    app:layout_rowWeight="4"
    tools:listitem="@layout/mesa_item"/>

</android.support.v7.widget.GridLayout>

</LinearLayout>

</LinearLayout>

```

➤ MesasActivity.java

```

public class MesasActivity extends AppCompatActivity
    implements AdapterView.OnItemClickListener,
        NavigationView.OnNavigationItemSelectedListener {
    public static final String EXTRA_CANTIDAD_MESAS_ACTUALIZADOS =
        "cantidad_mesas_actualizadas";
    private Spinner mPisosSpinner;
    private Spinner mAmbienteSpinner;
    private GridView mMesasGridView;
    private MesaPisoEE mMesaPisoSeleccionado;

    private ArrayList<SpinnerEE> mListaAmbientes;
    private ArrayList<SpinnerEE> mListaPisos;
    private ArrayList<MesaPisoEE> mListaMesas;
    private NavigationView mNavigationView;
    private MaterialDialog mProgress;

    private MesaPisoDAO mDataHelper;
    private SharedPreferences mPrefPedidoExtras;

    private BroadcastReceiver onEventRefrescarListadoMesas = new
    BroadcastReceiver() {

```

```

        @Override
        public void onReceive(Context context, Intent intent) {
            int cantMesasActualizadas =
intent.getIntExtra(EXTRA_CANTIDAD_MESAS_ACTUALIZADOS, 0);
            Toast.makeText(MesasActivity.this, "Nro de mesas actualizadas : " +
cantMesasActualizadas, Toast.LENGTH_SHORT).show();
            int nroPiso =
Integer.parseInt((mListaPisos.get(mPisosSpinner.getSelectedItemPosition()).getCo
digo()));
            int codAmbiente =
Integer.parseInt((mListaAmbientes.get(mAmbienteSpinner.getSelectedItemPositio
n()).getCodigo()));
            loadMesas(nroPiso, codAmbiente);
            showProgressIndicator(false);
        }
    };

    private BroadcastReceiver onEventActualizarEstadoMesa = new
BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            int resultadoOperacion =
intent.getIntExtra(ActualizarEstadoMesaService.EXTRA_RESULTADO_ACTUALI
ZACION, 0);
            String mensajeOperacion =
intent.getStringExtra(ActualizarEstadoMesaService.EXTRA_MENSAJE_ACTUALI
ZACION);

            if (resultadoOperacion > 0) {
                Log.d(DBHelper.TAG, "Resultado de Actualizar Estado de Mesa: " +
resultadoOperacion);

                Intent intentTo = new Intent(MesasActivity.this,
TomarPedidoActivity.class);
                startActivity(intentTo);
                finish();
                showProgressIndicator(false);
            } else {
                Log.d(DBHelper.TAG, "Se produjo la excepción: " + mensajeOperacion);
                Toast.makeText(MesasActivity.this, mensajeOperacion,
Toast.LENGTH_LONG).show();
                showProgressIndicator(false);
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mesas);
        overridePendingTransition(0, 0);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(false);
    }
}

```

```

getSupportActionBar().setDisplayShowTitleEnabled(false);

mDataHelper = new MesaPisoDAO(this);
mNavigationView = (NavigationView) findViewById(R.id.navigation_view);
mNavigationView.setNavigationItemSelectedListener(this);

mNavigationView.getMenu().getItem(SmartWaiter.OPCION_TOMAR_PEDIDO).
setChecked(true);

mPisosSpinner = (Spinner) findViewById(R.id.pisos_spinner);
mAmbienteSpinner = (Spinner) findViewById(R.id.ambientes_spinner);

mMesasGridView = (GridView) findViewById(R.id.mesasGridView);
mMesasGridView.setOnItemClickListener(this);

mPrefPedidoExtras = getSharedPreferences(PedidoExtraSharedPref.NAME,
MODE_PRIVATE);

loadPisosSpinner();
}

@Override
protected void onResume() {
    super.onResume();
    IntentFilter filterNotificarRefrescarListado = new
IntentFilter(ObtenerListaMesasService.ACTION_GET_TABLES_STATUS);
    LocalBroadcastManager.getInstance(this)
        .registerReceiver(onEventRefrescarListadoMesas,
filterNotificarRefrescarListado);
    IntentFilter filterNotificarActualizarEstadoMesa = new
IntentFilter(ActualizarEstadoMesaService.ACTION_UPDATE_TABLE_STATUS);
    LocalBroadcastManager.getInstance(this)
        .registerReceiver(onEventActualizarEstadoMesa,
filterNotificarActualizarEstadoMesa);
}

@Override
protected void onPause() {

LocalBroadcastManager.getInstance(this).unregisterReceiver(onEventRefrescarLis
tadoMesas);

LocalBroadcastManager.getInstance(this).unregisterReceiver(onEventActualizarEs
tadoMesa);
    super.onPause();
}

public Spinner getPisosSpinner() {
    return mPisosSpinner;
}

public Spinner getAmbienteSpinner() {
    return mAmbienteSpinner;
}

```

```

public ArrayList<SpinnerEE> getListaAmbientes() {
    return mListaAmbientes;
}

public ArrayList<SpinnerEE> getListaPisos() {
    return mListaPisos;
}

public GridView getMesasGridView() {
    return mMesasGridView;
}

public ArrayList<MesaPisoEE> getListaMesas() {
    return mListaMesas;
}

private void loadPisosSpinner() {
    mListaPisos = new ArrayList<SpinnerEE>();
    WeakReference<Activity> weakActivity = new
WeakReference<Activity>(this);
    mDataHelper.getPisosAsync(weakActivity);
}

public void loadAmbienteSpinner(final int nroPiso) {
    mListaAmbientes = new ArrayList<SpinnerEE>();
    WeakReference<Activity> weakActivity = new
WeakReference<Activity>(this);
    mDataHelper.getAmbientesAsync(weakActivity, nroPiso);
}

public void loadMesas(int nroPiso, int codAmbiente) {
    mListaMesas = new ArrayList<MesaPisoEE>();
    WeakReference<Activity> weakActivity = new
WeakReference<Activity>(this);
    mDataHelper.getMesasAsync(weakActivity, nroPiso, codAmbiente, "LIB");
}

public void startActualizarEstadoMesas() {
    Intent serviceIntent = new Intent(MesasActivity.this,
ObtenerListaMesasService.class);
    Log.d(DBHelper.TAG, "Antes de startService
ObtenerListaMesasServiceService");
    showProgressIndicator(true);
    startService(serviceIntent);
}

private void showProgressIndicator(boolean showValue) {
    if (showValue) {
        mProgress = new MaterialDialog.Builder(MesasActivity.this)
            .content("Espere por favor...")
            .cancelable(false)
            .progress(true, 0)
            .show();
    } else {

```

```

        if (mProgress != null) {
            mProgress.dismiss();
        }
    }
}

private void confirmarActualizarEstadoMesa() {

    new AlertDialog.Builder(this)
        .setTitle("Confirmación")
        .setMessage("¿Realmente desea proceder a efectuar un pedido sobre la
mesa número " + mMesaPisoSeleccionado.getNroMesa() + " ?")
        .setPositiveButton("Si", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int id) {

                dialog.cancel();
                Gson gson = new Gson();
                String mesaString = gson.toJson(mMesaPisoSeleccionado);
                Intent serviceIntent = new Intent(MesasActivity.this,
ActualizarEstadoMesaService.class);

                serviceIntent.putExtra(ActualizarEstadoMesaService.EXTRA_NUEVO_ESTADO_
MESA, "OCU");

                serviceIntent.putExtra(ActualizarEstadoMesaService.EXTRA_NUEVO_ESTADO_
RESERVA, "EFE");

                PedidoExtraSharedPref.save(mPrefPedidoExtras, mesaString,
MesasActivity.this.getClass().getName());

                Log.d(DBHelper.TAG, "Antes de startService
ActualizarEstadoMesaService");
                showProgressIndicator(true);
                startService(serviceIntent);

            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        }).show();

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_mesas, menu);
    return true;
}

@Override

```



```

public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_update) {
        startActualizarEstadoMesas();
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@Override
public void onItemClick(AdapterView<?> parent, View v,
    int position, long id) {
    if (parent.getId() == R.id.mesasGridView) {
        mMesaPisoSeleccionado = getListaMesas().get(position);
        confirmarActualizarEstadoMesa();
    }
}

@Override
public boolean onNavigationItemSelected(MenuItem menuItem) {
    if (menuItem.getOrder() != SmartWaiter.OPCION_TOMAR_PEDIDO) {
        WeakReference<AppCompatActivity> weakActivity = new
WeakReference<AppCompatActivity>(MesasActivity.this);
        Funciones.selectMenuOption(weakActivity, menuItem.getOrder());
        return true;
    }
    return true;
}
}

```

➤ menu_mesas.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.neversoft.smartwaiter.ui.MesasActivity">
    <item
        android:id="@+id/action_update"
        android:icon="@drawable/ic_update"
        android:orderInCategory="100"
        android:title="@string/menu_mesas_update"
        app:showAsAction="never"/>
</menu>

```

Realizar Pedido

➤ activity_tomar_pedido.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.neversoft.smartwaiter.ui.TomarPedidoActivity">

    <RelativeLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:background="?attr/colorPrimary"
        android:elevation="@dimen/appbar_elevation"
        android:orientation="vertical">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"

            app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>

        <android.support.v4.widget.Space
            android:id="@+id/appLogoAlign"
            android:layout_width="@dimen/list_icon"
            android:layout_height="0dp"
            android:layout_alignParentLeft="true"
            android:layout_below="@id/toolbar"

            android:layout_marginLeft="@dimen/abc_list_item_padding_horizontal_material"

            android:layout_marginStart="@dimen/abc_list_item_padding_horizontal_material"
            "/>

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="?attr/listPreferredItemHeight"
            android:layout_below="@id/appLogoAlign"
            android:layout_gravity="center_vertical"

            android:layout_marginLeft="@dimen/abc_list_item_padding_horizontal_material"

            android:layout_marginStart="@dimen/abc_list_item_padding_horizontal_material"
            "

            android:layout_toRightOf="@id/appLogoAlign"
            android:contentDescription="@string/app_name"
            android:src="@drawable/ic_logo"/>
    </RelativeLayout>
```

```

<android.support.v4.widget.Space
    android:id="@+id/layout_center"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_centerInParent="true"/>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_alignEnd="@id/layout_center"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@id/layout_center"
    android:layout_below="@id/appbar"
    android:layout_weight="1"
    android:orientation="vertical">

    <ListView
        android:id="@+id/categorias_listview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="8"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="@dimen/form_horizontal_padding_small"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="@color/primary"/>

    <ListView
        android:id="@+id/articulos_listview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="8"
        android:choiceMode="singleChoice"
        android:divider="@android:color/transparent"
        android:dividerHeight="4dp"/>
</LinearLayout>

<android.support.v7.widget.Space
    android:id="@+id/cardviewAligner"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignLeft="@id/layout_center"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignStart="@id/layout_center"
    android:layout_marginEnd="@dimen/list_item_extra_padding"
    android:layout_marginRight="@dimen/list_item_extra_padding"
    android:layout_marginTop="?attr/actionBarSize"
    android:elevation="@dimen/appbar_elevation"/>

```

```

<android.support.v7.widget.CardView
    android:id="@+id/pedido_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignBottom="@id/cardviewAligner"
    android:layout_alignLeft="@id/cardviewAligner"
    android:layout_alignRight="@id/cardviewAligner"
    android:layout_alignTop="@id/cardviewAligner"
    android:layout_marginBottom="@dimen/detail_container_bottom_margin"

    android:layout_marginEnd="@dimen/abc_list_item_padding_horizontal_material"

    android:layout_marginRight="@dimen/abc_list_item_padding_horizontal_material"
"
    app:cardElevation="@dimen/form_card_elevation"
    app:cardPreventCornerOverlap="false">

    <include layout="@layout/pedido_cliente"/>
</android.support.v7.widget.CardView>

</RelativeLayout>

```

➤ TomarPedidoActivity.java

```

public class TomarPedidoActivity extends AppCompatActivity
    implements OnItemClickListener, EditarCantidadItemFragment.Contract {

    private ListView mCategoriasListView;
    private ListView mArticulosListView;
    private ListView mPedidoListView;
    private ArrayList<CategoriaEE> mListaCategorias;
    private ArrayList<ArticuloEE> mListaArticulos;
    private CategoriaDAO mCategoriaDAO;
    private ArticuloDAO mArticuloDAO;
    private ArrayList<DetallePedidoEE> mItems;
    private ActionMode mActionMode;
    private int mSelectedItemsCount;
    private boolean mIsInActionMode = false;
    private MesaPisoEE mMesaPisoEE;

    private TextView mSubTotalPedidoTextView;
    private TextView mIGVPedidoTextView;
    private TextView mTotalPedidoTextView;

    private MaterialDialog mProgress;
    private float mTotal = 0;
    private String mPrevClassName;
    private SharedPreferences mPrefPedidoExtras;
    private BroadcastReceiver onEventActualizarEstadoMesa = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            int resultadoOperacion =
intent.getIntExtra(ActualizarEstadoMesaService.EXTRA_RESULTADO_ACTUALIZACION,
0);

            String mensajeOperacion =
intent.getStringExtra(ActualizarEstadoMesaService.EXTRA_MENSAJE_ACTUALIZACION)

```

```

;

    if (resultadoOperacion > 0) {
        Log.d(DBHelper.TAG, "Resultado de Actualizar Estado de Mesa: " +
resultadoOperacion);
        Class<?> clase = MesasActivity.class;
        try {
            clase = Class.forName(mPrevClassName);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        PedidoSharedPref.clear(TomarPedidoActivity.this);
        PedidoExtraSharedPref.remove(mPrefPedidoExtras);

        Intent intentTo = new Intent(TomarPedidoActivity.this, clase);
        startActivity(intentTo);
        finish();
        showProgressIndicator(false);
    } else {
        Log.d(DBHelper.TAG, "Se produjo la excepción: " + mensajeOperacion);
        Toast.makeText(TomarPedidoActivity.this, mensajeOperacion,
Toast.LENGTH_LONG).show();
        showProgressIndicator(false);
    }
}
};
private BroadcastReceiver sendDataReceiver = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d(DBHelper.TAG, "sendData broadcast received");
        boolean exito =
intent.getBooleanExtra(EnviarPedidoService.EXTRA_RESULTADO_EXITO, false);
        String mensaje =
intent.getStringExtra(EnviarPedidoService.EXTRA_RESULTADO_MENSAJE);

        abortBroadcast();

        if (exito) {
            mensaje = "Pedidos enviados correctamente.";
            Log.d(DBHelper.TAG, "Success from BroadcastReceiver within
EnviarDatosActivity : "
+ mensaje);
            Class<?> clase = MesasActivity.class;
            try {
                clase = Class.forName(mPrevClassName);
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
            PedidoSharedPref.clear(TomarPedidoActivity.this);
            PedidoExtraSharedPref.remove(mPrefPedidoExtras);
            Intent toIntent = new Intent(TomarPedidoActivity.this, clase);
            startActivity(toIntent);
            finish();
            showProgressIndicator(false);
        } else {
            showProgressIndicator(false);
            Log.d(DBHelper.TAG, "Exception from BroadcastReceiver within
EnviarDatosActivity : " + mensaje);
        }
    }
}

```

```

        Toast.makeText(TomarPedidoActivity.this, mensaje,
Toast.LENGTH_LONG).show();
    }

};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    overridePendingTransition(0, 0);
    setContentView(R.layout.activity_tomar_pedido);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(false);
    getSupportActionBar().setDisplayShowTitleEnabled(false);

    mPrefPedidoExtras = getSharedPreferences(PedidoExtraSharedPref.NAME,
MODE_PRIVATE);
    mPrevClassName =
mPrefPedidoExtras.getString(PedidoExtraSharedPref.STARTING_ACTIVITY,
MesasActivity.class.getClass().getName());
    String mesaString =
mPrefPedidoExtras.getString(PedidoExtraSharedPref.SELECTED_TABLE_JSON, null);

    Gson gson = new Gson();
    mMesaPisoEE = gson.fromJson(mesaString, MesaPisoEE.class);

    mCategoriaDAO = new CategoriaDAO(getApplicationContext());
    mArticuloDAO = new ArticuloDAO(getApplicationContext());

    mCategoriasListView = (ListView) findViewById(R.id.categorias_listview);
    mCategoriasListView.setOnItemClickListener(this);

    mArticulosListView = (ListView) findViewById(R.id.articulos_listview);
    mArticulosListView.setOnItemClickListener(this);

    mSubTotalPedidoTextView = (TextView)
findViewById(R.id.subTotalPedidoTextView);
    mIGVPedidoTextView = (TextView) findViewById(R.id.igvPedidoTextView);
    mTotalPedidoTextView = (TextView) findViewById(R.id.totalPedidoTextView);
    mPedidoListView = (ListView) findViewById(R.id.detallePedidoListView);
    mPedidoListView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE_MODAL);
    mPedidoListView.setMultiChoiceModeListener(new ActionModeCallbacks());

    loadCategorias();
    mItems = PedidoSharedPref.getItems(TomarPedidoActivity.this);
    showItems();
}

@Override
public void onBackPressed() {
    confirmarCancelarPedido();
}

private void showProgressIndicator(boolean showValue) {
    if (showValue) {
        mProgress = new MaterialDialog.Builder(TomarPedidoActivity.this)
            .content("Espere por favor...")
            .cancelable(false)
            .progress(true, 0)
            .show();
    }
}

```

```

    } else {
        if (mProgress != null) {
            mProgress.dismiss();
        }
    }
}

public ListView getCategoriasListView() {
    return mCategoriasListView;
}

public ListView getArticulosListView() {
    return mArticulosListView;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_tomar_pedido, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_save_send:
            sendData();
            break;
        case R.id.action_cancelar:
            confirmarCancelarPedido();
            break;
    }
    return true;
}

private void confirmarCancelarPedido() {
    new AlertDialog.Builder(this)
        .setTitle("Confirmación")
        .setMessage("¿Realmente desea cancelar el pedido actual?")
        .setPositiveButton("Si", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int id) {

                dialog.cancel();

                Intent serviceIntent = new Intent(TomarPedidoActivity.this,
                ActualizarEstadoMesaService.class);

                serviceIntent.putExtra(ActualizarEstadoMesaService.EXTRA_NUEVO_ESTADO_MESA,
                "LIB");

                serviceIntent.putExtra(ActualizarEstadoMesaService.EXTRA_NUEVO_ESTADO_RESERVA,
                "CAN");

                Log.d(DBHelper.TAG, "Antes de startService ActualizarEstadoMesaService
                (Cancelar Pedido)");
                showProgressIndicator(true);
                startService(serviceIntent);

            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {

```

```

        @Override
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();
        }
    }).show();
}

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int position, long id)
{
    if (adapterView.getId() == R.id.categorias_listview) {
        int familiaId = Integer.parseInt(mListaCategorias.get(position).getCodigo().trim());
        loadArticulos(familiaId);
    } else if (adapterView.getId() == R.id.articulos_listview) {
        DetallePedidoEE itemDetalle = new
DetallePedidoEE(mListaArticulos.get(position));
        PedidoSharedPref.addItem(this, itemDetalle);
        mItems = PedidoSharedPref.getItems(this);
        showItems();
    }
}

private void showItems() {
    float subTotal = 0;
    float igv;
    ArrayList<HashMap<String, String>> data = new ArrayList<HashMap<String,
String>>();
    if (mItems == null) {
        mItems = new ArrayList<>();
    }
    for (DetallePedidoEE item : mItems) {
        HashMap<String, String> map = new HashMap<String, String>();
        map.put("articuloDescripcion", item.getDescArticulo());
        map.put("articuloCantidad", String.valueOf(item.getCantidad()));
        map.put("articuloPrecio", String.valueOf(item.getPrecio()));
        data.add(map);

        subTotal += item.getCantidad() * item.getPrecio();
    }
    int resource = R.layout.order_item;
    String[] from = {"articuloDescripcion", "articuloCantidad", "articuloPrecio"};
    int[] to = {R.id.productoPedidoDescTextView, R.id.cantidadPedidoTextView,
R.id.precioPedidoTextView};
    SimpleAdapter adapter = new SimpleAdapter(this, data, resource, from, to);
    mPedidoListView.setAdapter(adapter);
    igv = subTotal * 0.19f;
    mTotal = subTotal + igv;
    mSubTotalPedidoTextView.setText(String.valueOf(subTotal));
    mIGVPedidoTextView.setText(String.valueOf(igv));
    mTotalPedidoTextView.setText(String.valueOf(mTotal));
}

private void sendData() {
    new AlertDialog.Builder(this)
        .setTitle("Confirmación")
        .setMessage("¿Realmente desea enviar los pedidos para su procesamiento?")
        .setPositiveButton("Si",
            new DialogInterface.OnClickListener() {

```



```

        @Override
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();
            Gson gson = new Gson();
            PedidoEE pedido;
            pedido = createPedido();

            String pedidosString = gson.toJson(pedido);
            Intent iniciarServiceIntent = new Intent(TomarPedidoActivity.this,
            EnviarPedidoService.class);
            iniciarServiceIntent.putExtra(EnviarPedidoService.EXTRA_PEDIDO_JSON,
            pedidosString);

            Log.d(DBHelper.TAG, "Antes de startService SendDataService");
            showProgressIndicator(true);
            startService(iniciarServiceIntent);

        }
    })
    .setNegativeButton("No",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();

        }
    }).show();
}

@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(sendDataReceiver);

    LocalBroadcastManager.getInstance(this).unregisterReceiver(onEventActualizarEstadoMesa);
}

Log.d(DBHelper.TAG, "Entre a onPause - TomarPedidoActivity");

@Override
protected void onResume() {
    super.onResume();

    Log.d(DBHelper.TAG, "Entre a onResume - TomarPedidoActivity");
    IntentFilter filter = new IntentFilter(EnviarPedidoService.ACTION_SEND_DATA);
    filter.setPriority(2);
    registerReceiver(sendDataReceiver, filter);
    IntentFilter filterNotificarActualizarEstadoMesa = new
    IntentFilter(ActualizarEstadoMesaService.ACTION_UPDATE_TABLE_STATUS);

    LocalBroadcastManager.getInstance(this).registerReceiver(onEventActualizarEstadoMesa,
    filterNotificarActualizarEstadoMesa);

}

public ArrayList<CategoriaEE> getListaCategorias() {
    return mListaCategorias;
}

public ArrayList<ArticuloEE> getListaArticulos() {
    return mListaArticulos;
}

```

```

    }

    public void loadCategorias() {
        mListaCategorias = new ArrayList<CategoriaEE>();
        WeakReference<Activity> weakActivity = new WeakReference<Activity>(this);
        mCategoriaDAO.getCategoriasAsync(weakActivity);
    }

    public void loadArticulos(int familiaId) {
        mListaArticulos = new ArrayList<ArticuloEE>();
        WeakReference<Activity> weakActivity = new WeakReference<Activity>(this);
        mArticuloDAO.getArticuloPorFamiliaAsync(weakActivity, familiaId);
    }

    private PedidoEE createPedido() {
        SharedPreferences prefConfig = getApplicationContext().getSharedPreferences(
            LoginActivity.PREF_CONFIG, Context.MODE_PRIVATE);
        String codUsuario = prefConfig.getString("Usuario",
            "").toUpperCase(Locale.getDefault());
        String codCia = prefConfig.getString("CodCia", "");

        PedidoEE pedido = new PedidoEE();

        pedido.setFecha(Funciones.getCurrentDate("yyyy/MM/dd"));
        pedido.setNroMesa(mMesaPisoEE.getNroMesa());
        pedido.setNroPiso(mMesaPisoEE.getNroPiso());
        pedido.setCantRecogida("");
        pedido.setAmbiente(mMesaPisoEE.getCodAmbiente());
        pedido.setCodUsuario(codUsuario);
        pedido.setCodCliente(0);
        pedido.setTipoVenta("");
        pedido.setTipoPago("");
        pedido.setMoneda("SOL");
        pedido.setMontoTotal(mTotal);
        pedido.setMontoRecibido(0);
        pedido.setEstado("020");
        pedido.setCodCia(codCia);
        pedido.setDetalle(mItems);
        return pedido;
    }

    @Override
    public void OnEditarCantidadItemClick(DetallePedidoEE item, int which) {
        if (which == DialogInterface.BUTTON_POSITIVE) {
            PedidoSharedPref.updateItem(TomarPedidoActivity.this, item);
            mItems = PedidoSharedPref.getItems(TomarPedidoActivity.this);
            showItems();
        }
        mActionMode.finish();
    }

    public class ActionModeCallbacks implements AbsListView.MultiChoiceModeListener {

        @Override
        public boolean onCreateActionMode(ActionMode mode, Menu menu) {

            TomarPedidoActivity.this.getMenuInflater().inflate(R.menu.menu_tomar_pedido_contextua
            l, menu);
            mActionMode = mode;
            mIsInActionMode = true;
        }
    }

```

```

        mode.setTitle(String.format("%d Selected", mSelectedItemsCount));
        return true;
    }

    @Override
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        if (mSelectedItemsCount == 1) {
            showOptionsOnSingleSelectedItem(true, menu);
            return true;
        } else {
            showOptionsOnSingleSelectedItem(false, menu);
            return true;
        }
    }

    private void showOptionsOnSingleSelectedItem(boolean showOnSingle, Menu menu)
    {
        MenuItem item = menu.findItem(R.id.menu_plus);
        item.setVisible(showOnSingle);
        item = menu.findItem(R.id.menu_minus);
        item.setVisible(showOnSingle);
        item = menu.findItem(R.id.menu_edit);
        item.setVisible(showOnSingle);
    }

    private void modifyProductQuantity(int accion, SparseBooleanArray
checkedItemPositions) {
        if (checkedItemPositions.valueAt(0)) {
            int position = checkedItemPositions.keyAt(0);
            DetallePedidoEE itemDetalle = mItems.get(position);
            if (accion == 0) {
                itemDetalle.setCantidad(itemDetalle.getCantidad() + 1);
            } else {
                if (itemDetalle.getCantidad() > 1) {
                    itemDetalle.setCantidad(itemDetalle.getCantidad() - 1);
                } else {
                    mItems.remove(itemDetalle);
                }
            }
        }
        PedidoSharedPref.saveItems(TomarPedidoActivity.this, mItems);
        showItems();
    }

    private void removeSelectedItems(SparseBooleanArray checkedItemPositions) {
        for (int i = (checkedItemPositions.size() - 1); i >= 0; i--) {
            if (checkedItemPositions.valueAt(i)) {
                int position = checkedItemPositions.keyAt(i);
                DetallePedidoEE itemDetalle = mItems.get(position);
                mItems.remove(itemDetalle);
            }
        }
        PedidoSharedPref.saveItems(TomarPedidoActivity.this, mItems);
        showItems();
    }

    @Override
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        SparseBooleanArray checkedItemPositions =
mPedidoListView.getCheckedItemPositions();
        switch (item.getItemId()) {
            case R.id.menu_delete:

```

```

        removeSelectedItems(checkedExceptionPositions);
        mode.finish();
        return true;
    case R.id.menu_edit:
        int position = checkedItemPositions.keyAt(0);
        DetallePedidoEE itemDetalle = mItems.get(position);

        EditarCantidadItemFragment.newInstance(itemDetalle).show(TomarPedidoActivity.this.get
FragmentManager(), "EditarCantidadFragment");
        return true;
    case R.id.menu_plus:
        modifyProductQuantity(0, checkedItemPositions);
        mode.finish();
        return true;
    case R.id.menu_minus:
        modifyProductQuantity(1, checkedItemPositions);
        mode.finish();
        return true;
    default:
        return false;
    }
}

@Override
public void onDestroyActionMode(ActionMode mode) {
    mIsInActionMode = false;
}

@Override
public void onItemCheckedStateChanged(ActionMode mode, int position, long id,
boolean checked) {
    mSelectedItemsCount = mPedidoListView.getCheckedItemCount();
    mode.setTitle(String.format("%d Seleccionados", mSelectedItemsCount));
    mode.invalidate();
}
}
}

```

➤ menu_tomar_pedido.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context="com.idealsolution.smartwaiter.ui.CategoryActivity">
    <item
        android:id="@+id/action_save_send"
        android:icon="@drawable/ic_action_send"
        android:orderInCategory="10"
        android:title="@string/action_save_send"
        app:showAsAction="never"/>
    <item
        android:id="@+id/action_cancelar"
        android:icon="@drawable/ic_action_cancel"
        android:orderInCategory="20"
        android:title="@string/action_cancelar_pedido"
    >

```

```

        app:showAsAction="never"/>
    </menu>

```

➤ EnviarPedidoService.java

```

public class EnviarPedidoService extends IntentService {
    public static final String ACTION_SEND_DATA =
"com.neversoft.smartwaiter.ENVIAR_PEDIDO";
    public static final String EXTRA_PEDIDO_JSON = "pedido_json";

    public static final String EXTRA_RESULTADO_EXITO = "resut_exito";
    public static final String EXTRA_RESULTADO_MENSAJE = "result_mesnaje";
    private static final String NAME = "EnviarPedidoService";
    private SharedPreferences mPrefConfig;
    private SharedPreferences mPrefConexion;

    private String mAmbiente;
    private String mCodMozo;
    private String mCodCia;
    private String mUsuario;

    private PedidoDAO mPedidoDAO;

    public EnviarPedidoService() {
        super(NAME);
        setIntentRedelivery(false);
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        String mensaje = "";
        int idOrderFromServer = 0;
        boolean exito = false;
        long idPedido = 0;

        mPrefConfig =
getApplicationContext().getSharedPreferences(LoginActivity.PREF_CONFIG,
Context.MODE_PRIVATE);
        mPrefConexion =
getApplicationContext().getSharedPreferences(ConexionSharedPref.NAME,
Context.MODE_PRIVATE);

        mAmbiente = mPrefConexion.getString(ConexionSharedPref.AMBIENTE,
        "");
        mCodCia = mPrefConfig.getString("CodCia", "");
        mCodMozo = mPrefConfig.getString("CodMozo", "");
        mUsuario = mPrefConfig.getString("Usuario",
        "").toUpperCase(Locale.getDefault());

        try {
            if (Funciones.hasActiveInternetConnection(getApplicationContext())) {

```

```

        ArrayList<PedidoEE> listaPedidosRegistrados = new ArrayList<>();
        mPedidoDAO = new PedidoDAO(getApplicationContext());
        String stringPedido = intent.getStringExtra(EXTRA_PEDIDO_JSON);
        Gson gson = new Gson();
        PedidoEE pedido = gson.fromJson(stringPedido, PedidoEE.class);
        try {
            idPedido = mPedidoDAO.savePedido(pedido, 1);
        } catch (Exception e) {
            throw new Exception("No se pudo guardar el pedido. Excepcion: " +
e.getMessage());
        }
        listaPedidosRegistrados.add(pedido);
        if (idPedido > 0) {
            String dataToSend = getEnvio(listaPedidosRegistrados, idPedido);
            idOrderFromServer = sendDataToServer(dataToSend, idPedido);
            if (idOrderFromServer > 0) {
                exito = true;
            } else {
                throw new Exception("Pedido enviado pero no guardado");
            }
        }
    }
} catch (Exception e) {
    mensaje = e.getMessage();
    exito = false;
}
Intent broadcastIntent = new Intent();
broadcastIntent.setAction(EnviarPedidoService.ACTION_SEND_DATA);
broadcastIntent.putExtra(EXTRA_RESULTADO_EXITO, exito);
broadcastIntent.putExtra(EXTRA_RESULTADO_MENSAJE, mensaje);
Log.d(DBHelper.TAG, "El id del pedido generado en el servidor es : " +
idOrderFromServer);

sendOrderedBroadcast(broadcastIntent, null);

}

private String getEnvio(ArrayList<PedidoEE> pedidosRegistrados, long
idPedido) throws Exception {
    String result;
    JSONObject jsonObjEnvio = new JSONObject();

    jsonObjEnvio.addProperty("cadenaConexion", mAmbiente);
    JSONArray jsArrayPedidos = new JSONArray();
    for (PedidoEE pedido : pedidosRegistrados) {
        JSONObject jsPedido = getPedido(pedido, idPedido);
        jsArrayPedidos.add(jsPedido);
    }
    jsonObjEnvio.add("pedidos", jsArrayPedidos);
    result = jsonObjEnvio.toString();
    return result;
}

```

```

private int sendDataToServer(String dataToSend, long idPedido) throws
Exception {
    int procesoOK = 0;
    Object requestObject = null;
    String resultado;

    String urlServer = RestUtil.obtainURLServer(getApplicationContext());
    String POST_URI = urlServer + "restaurante/EnviarListaPedidoMV/";

    Log.d(DBHelper.TAG, POST_URI);
    List<NameValuePair> parameters = new ArrayList<NameValuePair>();
    parameters.add(new BasicNameValuePair("dataToSend", dataToSend));
    if (Funciones.hasActiveInternetConnection(getApplicationContext())) {
        RestConnector restConnector = RestUtil
            .obtainFormPostConnection(POST_URI, parameters);
        requestObject = restConnector.doRequest(POST_URI);
        if (requestObject instanceof String) {
            mPedidoDAO.updateEstadoEnviado(idPedido, "1");
            resultado = (String) requestObject;
            procesoOK = Integer.parseInt(resultado);

        } else if (requestObject instanceof Exception) {
            throw new Exception(((Exception) requestObject).getMessage());
        }
    }
    return procesoOK;
}

private JSONObject getPedido(PedidoEE ped, long idPedido) throws Exception {
    JSONObject jsonObjPed = new JSONObject();
    JSONArray jsonArrayPedDetalle;
    jsonObjPed.addProperty("id", idPedido);
    jsonObjPed.addProperty("fecha", ped.getFecha());
    jsonObjPed.addProperty("nroMesa", ped.getNroMesa());
    jsonObjPed.addProperty("nroPiso", ped.getNroPiso());
    jsonObjPed.addProperty("ambiente", ped.getAmbiente());
    jsonObjPed.addProperty("codMozo", mCodMozo);
    jsonObjPed.addProperty("codUsuario", ped.getCodUsuario());
    jsonObjPed.addProperty("codCliente", ped.getCodCliente());
    jsonObjPed.addProperty("tipoVenta", ped.getTipoVenta());
    jsonObjPed.addProperty("tipoPago", ped.getTipoPago());
    jsonObjPed.addProperty("moneda", ped.getMoneda());
    jsonObjPed.addProperty("montoTotal", ped.getMontoTotal());
    jsonObjPed.addProperty("montoRecibido", ped.getMontoRecibido());
    jsonObjPed.addProperty("estado", String.valueOf(ped.getEstado()));
    jsonObjPed.addProperty("codcia", mCodCia);
    jsonArrayPedDetalle = getDetallePedido(ped.getDetalle(), idPedido);
    jsonObjPed.add("detalle", jsonArrayPedDetalle);

    return jsonObjPed;
}

private JSONArray getDetallePedido(ArrayList<DetallePedidoEE>
listaDetallePedido, long idPedido) throws Exception {

```

```

        JSONArray jsonArrayDetalle = new JSONArray();
        JSONObject jsonObjItem;
        for (int i = 0; i < listaDetallePedido.size(); i++) {
            jsonObjItem = new JSONObject();
            jsonObjItem.addProperty("id", idPedido);
            jsonObjItem.addProperty("idPedido", (i + 1));
            jsonObjItem.addProperty("codArticulo",
listaDetallePedido.get(i).getCodArticulo());
            jsonObjItem.addProperty("um", listaDetallePedido.get(i).getUm());
            jsonObjItem.addProperty("cantidad",
listaDetallePedido.get(i).getCantidad());
            jsonObjItem.addProperty("precio", listaDetallePedido.get(i).getPrecio());
            jsonObjItem.addProperty("tipoArticulo",
listaDetallePedido.get(i).getTipoArticulo());
            jsonObjItem.addProperty("codArticuloPrincipal",
listaDetallePedido.get(i).getCodArticuloPrincipal());
            jsonObjItem.addProperty("comentario",
listaDetallePedido.get(i).getComentario());
            jsonObjItem.addProperty("estadoArticulo",
listaDetallePedido.get(i).getEstadoArticulo());
            jsonArrayDetalle.add(jsonObjItem);
        }

        return jsonArrayDetalle;
    }
}

```

➤ Web Service

```

public int EnviarListaPedidoMV([FromBody]
    EnvioMovilRestaurante_API envioMovil)
{
    string cadenaConexion;
    DataSet dsEnvioPedido = new DataSet();
    PedidoMovilRestaurante pedidoMovilRestauranteObj;
    DataTable dtCabecera;
    DataRow drCabecera;
    DataTable dtDetalle;
    DataRow drDetalle;
    int nroPedido;

    try
    {
        Configuracion.PrepararAppConfig();
        Configuracion.CadenaServidor =
            ConfigurationManager.ConnectionStrings["C
onexionServidor"].ConnectionString;
        cadenaConexion =
            Configuracion.CompletarCredenciales(envioMo
vil.cadenaConexion);

        pedidoMovilRestauranteObj = new
            PedidoMovilRestaurante(cadenaConexion);
    }
}

```



```

dtCabecera =
    pedidoMovilRestauranteObj.ConsultarCabeceraP
    edidoMovil(0);
dtDetalle =
    pedidoMovilRestauranteObj.ConsultarDetallePe
    didoMovil(0);

foreach (var pedido in envioMovil.pedidos)
{
    drCabecera = dtCabecera.NewRow();
    drCabecera["nroped"] = pedido.Id;
    drCabecera["fecped"] = pedido.fecha;
    drCabecera["nromesa"] = pedido.nroMesa;
    drCabecera["codambiente"] = pedido.ambiente;
    drCabecera["nropiso"] = pedido.nropiso;
    drCabecera["codMozo"] = pedido.codMozo;
    drCabecera["coduserori"] = pedido.codUsuario;
    drCabecera["codcli"] = pedido.codCliente;
    drCabecera["tipcomp"] = pedido.tipoVenta;
    drCabecera["codpag"] = pedido.tipoPago;
    drCabecera["codmon"] = pedido.moneda;
    drCabecera["imptotal"] = pedido.montoTotal;
    drCabecera["imprecibido"] = pedido.montoRecibido;
    drCabecera["estado"] = pedido.estado;
    drCabecera["codcia"] = pedido.codcia;

    foreach (var detalle in pedido.detalle)
    {
        drDetalle = dtDetalle.NewRow();
        drDetalle["nroped"] = detalle.id;
        drDetalle["item"] = detalle.idPedido;
        drDetalle["codart"] = detalle.codArticulo;
        drDetalle["um"] = detalle.um;
        drDetalle["cantped"] = detalle.cantidad;
        drDetalle["preuni"] = detalle.precio;
        drDetalle["tipart"] = detalle.tipoArticulo;
        drDetalle["codartalt"] =
            detalle.codArticuloPrincipal;
        drDetalle["glosa"] = detalle.comentario;
        drDetalle["estart"] = detalle.estadoArticulo;
        dtDetalle.Rows.Add(drDetalle);
    }
    dtCabecera.Rows.Add(drCabecera);
}

dtCabecera.TableName = "TablaCabecera";
dsEnvioPedido.Tables.Add(dtCabecera.Copy());

dtDetalle.TableName = "TablaDetalle";
dsEnvioPedido.Tables.Add(dtDetalle.Copy());

pedidoMovilRestauranteObj = new
    PedidoMovilRestaurante(cadenaConexion);
nroPedido =
    pedidoMovilRestauranteObj.AdicionarPedidoMo
    vil(dsEnvioPedido);

```

```
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    return nroPedido;  
}
```

4.6.7. SPRINT 7

4.6.7.1. CALCULANDO LA VELOCIDAD ESTIMADA PARA EL SÉPTIMO SPRINT

- ✓ Calculando duración del sprint:
5 semanas= 25 días laborables.
- ✓ Calculando días hombres disponibles:

PROGRAMADOR	DIAS
Programador 1	25
Programador 2	25
Total	50 días hombres disponible

Tabla 69. Calculo Días Hombre – Séptimo Sprint

- ✓ Calculando factor de dedicación (FD):
 $FD = \text{Puntos de historia (último sprint)} / \text{días hombres disponible}$
 $FD = 49/50 = 0.98$
- ✓ Velocidad estimada: $FD \times \text{Días –Hombre- Disponible}$
 $0.98 \times 50 = 49$

4.6.7.2. DEFINIENDO LOS PRODUCTOS A DESARROLLAR EN EL SÉPTIMO SPRINT

Nº	SPRINT3	PUNTO S	SITUACIÓN
1	ACTUALIZACIÓN DE PEDIDOS A RECOGER	10	TERMINADO
2	SELECCIÓN DE PEDIDO Y DE LOS ÍTEMS	10	TERMINADO

	ESPECÍFICOS RECOGIDOS		
3	ACTUALIZACIÓN DE ESTADO DE PEDIDO E ÍTEMS RECOGIDOS TANTO EN EL SERVIDOR COMO EN EL DISPOSITIVO.	10	TERMINAD O
4	LISTADO DE PEDIDOS A FACTURAR	6	TERMINAD O
5	SELECCIÓN DE PEDIDO A FACTURAR Y BÚSQUEDA DE CLIENTE	4	TERMINAD O
6	ENVÍO DE DATOS DEL PEDIDO FACTURADO Y ACTUALIZACIÓN DEL ESTADO DEL PEDIDO EN EL DISPOSITIVO	13	TERMINAD O
7	REGISTRO DE CIERRE DE PERIODO Y ELIMINACIÓN DE DATA LOCAL	10	TERMINAD O
*	TOTAL DE PUNTOS DE HISTORIA	63	63
Objetivo	<ul style="list-style-type: none"> Listar pedidos listos para recoger. Actualizar el estado de los pedidos recogidos. 		

	<ul style="list-style-type: none"> • Facturar pedidos. • Cerrar periodo de atención. 		
--	--	--	--

Tabla 70. Productos en el Séptimo Sprint

4.6.7.3 DEFINIENDO LAS HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Número: 1	Usuario: Mozo
Nombre de Historia: Actualización de pedidos a recoger	
Prioridad en Negocio: baja	
Puntos estimados: 10	Iteración asignada: 2
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá sincronizar la relación de los pedidos listos para recoger de acuerdo a la información disponible en el servidor.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Pedidos a recoger”, se iniciará el proceso de consulta y actualización del estado de los pedidos. ✓ Verificar que en la lista de pedidos solo aparecerán los pedidos que tengan por lo menos un ítem listo para ser recogido. 	
Observaciones:	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases de conexión a la base de datos	Alex Ventura Chero

Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 71. 1° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 2	Usuario: Mozo
Nombre de Historia: Selección de pedido y de los ítems específicos recogidos	
Prioridad en Negocio: baja	
Puntos estimados: 10	Iteración asignada: 2
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá seleccionar los ítems específicos de un pedido para posteriormente establecerlos como recogidos.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar un pedido en la lista de la derecha se mostrarán los ítems de ese pedido que están listos para ser recogidos. ✓ Verificar que al mantener presionado un ítem, en la parte superior aparecerá la barra de acción con la acción 	

“Enviar” que permitirá establecer el estado de los ítems como recogidos.	
Observaciones:	
Tareas	Responsable
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 72. 2° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 3	Usuario: Mozo
Nombre de Historia: Actualización de estado de pedido e ítems recogidos tanto en el servidor como en el dispositivo.	
Prioridad en Negocio: baja	
Puntos estimados: 10 Iteración asignada: 1	
Programador responsable: José Huamán Valqui	
Descripción: Esta opción permitirá establecer el estado de los ítems seleccionados como recogidos tanto el dispositivo como en el servidor.	

Pruebas:	
<ul style="list-style-type: none"> ✓ Verificar que al seleccionar la acción “Enviar” de la barra de acción, todos los ítems seleccionados en el servidor en la tabla “REPEDIDOMOVILD”, cambiaran del estado “2” al estado “3”. ✓ Verificar que todos los ítems seleccionados y procesados mediante la acción “Enviar”, ya no aparecerán en la lista de haber resultado exitoso el proceso de actualización. ✓ Verificar que, si todos los ítems de un pedido fueron seleccionados y procesados exitosamente, el pedido desaparecerá de la lista de pedidos por recoger. 	
Observaciones:	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 73. 3° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 4	Usuario: Mozo
Nombre de Historia: Listado de pedidos a facturar	
Prioridad en Negocio: alta	
Puntos estimados: 6	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá mostrar la relación completa de los pedidos que están listos para facturar.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Pedidos a facturar”, se iniciará el proceso de consulta y actualización del estado de los pedidos. ✓ Verificar que en la lista de pedidos solo aparecerán los pedidos en los que por lo menos se haya recogido un ítem. 	
Observaciones:	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui

Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 74. 4° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 5	Usuario: Mozo
Nombre de Historia: Selección de pedido a facturar y búsqueda de cliente	
Prioridad en Negocio: alta	
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá seleccionar un pedido, buscar un cliente y asignar al pedido que se está por facturar.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al seleccionar un pedido en el panel de la derecha se mostrará el total del pedido. ✓ Verificar que solo si el tipo de pago es “Efectivo”, se tendrá que indicar el monto “Recibido”. ✓ Verificar que solo si el tipo de venta es “Factura”, se solicitará indicar el cliente. <p>Verificar que para buscar un cliente se debe indicar su RUC</p>	
Observaciones:	

Tareas	Responsable
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 75. 5° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 6	Usuario: Mozo
Nombre de Historia: Envío de datos del pedido facturado y actualización del estado del pedido en el dispositivo	
Prioridad en Negocio: alta	
Puntos estimados: 13	Iteración asignada: 1
Programador responsable: José Huamán Valqui	
<p>Descripción:</p> <p>Esta opción permitirá enviar los datos del pedido que se facturará. Además, este proceso se encargará de actualizar el estado del pedido tanto en el servidor como en el dispositivo.</p>	
<p>Pruebas:</p> <ul style="list-style-type: none"> ✓ Verificar que al presionar el botón “Aceptar”, después de haber completado los campos de acuerdo al tipo de venta y tipo de pago, los datos del pedido serán enviados para su facturación. ✓ Verificar que una vez terminado el proceso de envío se mostrará un mensaje de éxito y el pedido desaparecerá de la lista de pedidos por facturar. A su vez, del panel de la derecha será restablecido a su estado inicial. 	

Observaciones:	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 76. 6° Historia de Usuario – Séptimo Sprint

HISTORIA DE USUARIO	
Número: 7	Usuario: Mozo
Nombre de Historia: Registro de cierre de periodo y eliminación de data local	
Prioridad en Negocio: alta	
Puntos estimados: 10 Iteración asignada: 1	
Programador responsable: José Huamán Valqui	
Descripción: Esta opción permitirá registrar el cierre de periodo del mozo (tanto en el dispositivo como en el servidor) y la vez permitirá eliminar todo registro de información del periodo que fue cerrado.	
Pruebas:	
<ul style="list-style-type: none"> ✓ Verificar que al seleccionar la opción “Cerrar Periodo”, se mostrará la interfaz necesaria para dicho propósito. ✓ Verificar que al presionar el botón “Cerrar Periodo”, en el servidor, en tabla “ppp” se actualizará el campo “ttt” 	

<p>insertado al iniciar el periodo y pasará del valor “a” al valor “b”.</p> <ul style="list-style-type: none"> ✓ Verificar que una vez culminado el proceso se mostrará un mensaje de confirmación y el botón “Cerrar Periodo” quedará deshabilitado. ✓ Verificar que al cerrar y abrir la aplicación una vez culminado el proceso se volverá a solicitar las credenciales de usuario. 	
<p>Observaciones: No se podrá iniciar el proceso de cierre de periodo si haber sincronizado los datos.</p>	
Tareas	Responsable
Identificar y construir procedimientos almacenados	Alex Ventura Chero
Crear webmethods	José Huamán Valqui
Diseñar clases correspondientes	José Huamán Valqui
Diseño de interfaz de usuario	Alex Ventura Chero
Programar interfaz de usuario	José Huamán Valqui
Programar componentes internos	José Huamán Valqui
Integrar componentes	José Huamán Valqui
Pruebas	Alex Ventura Chero
Correcciones	José Huamán Valqui
Firma:	

Tabla 77. 7° Historia de Usuario – Séptimo Sprint

4.6.7.4. DIAGRAMA DE CLASES SPRINT 7

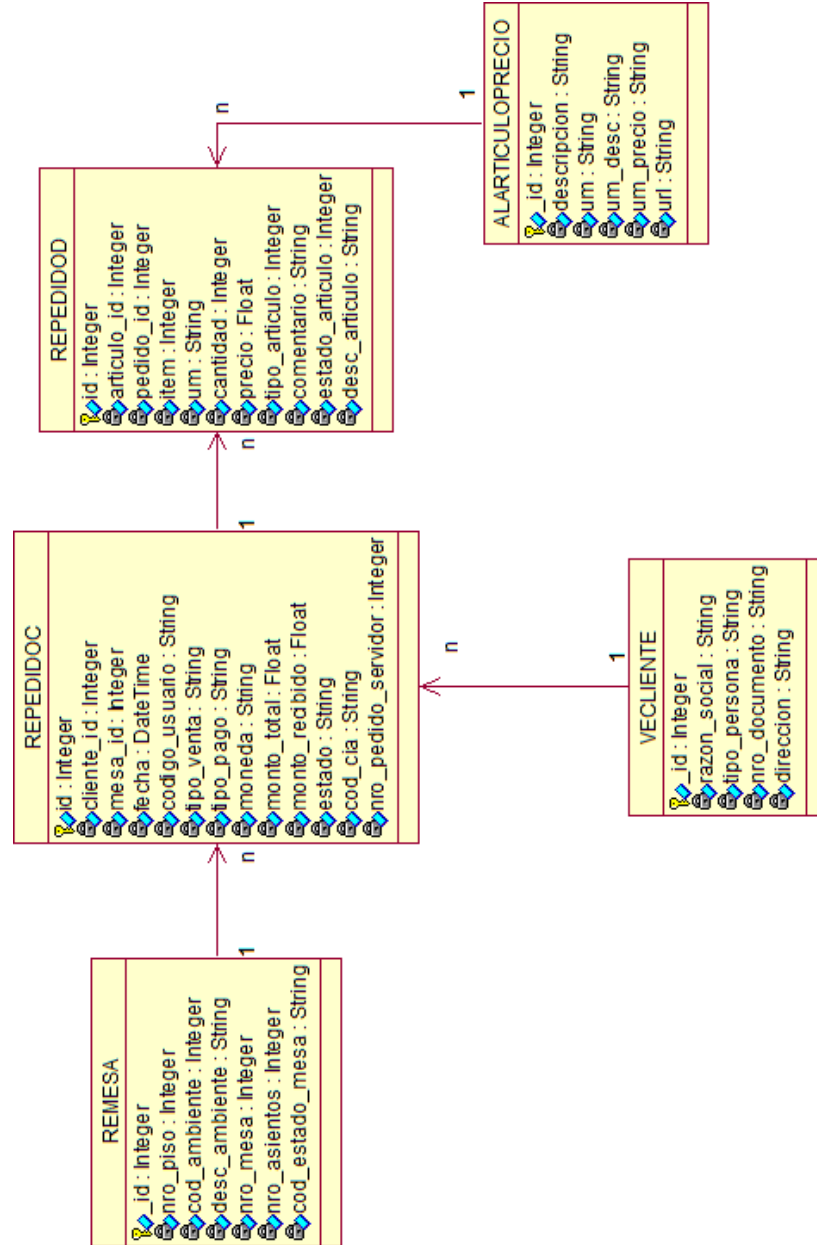


Ilustración 183 Diagrama de Clases – Séptimo Sprint

4.6.7.5. DIAGRAMA LOGICO Y FISICO SPRINT 7

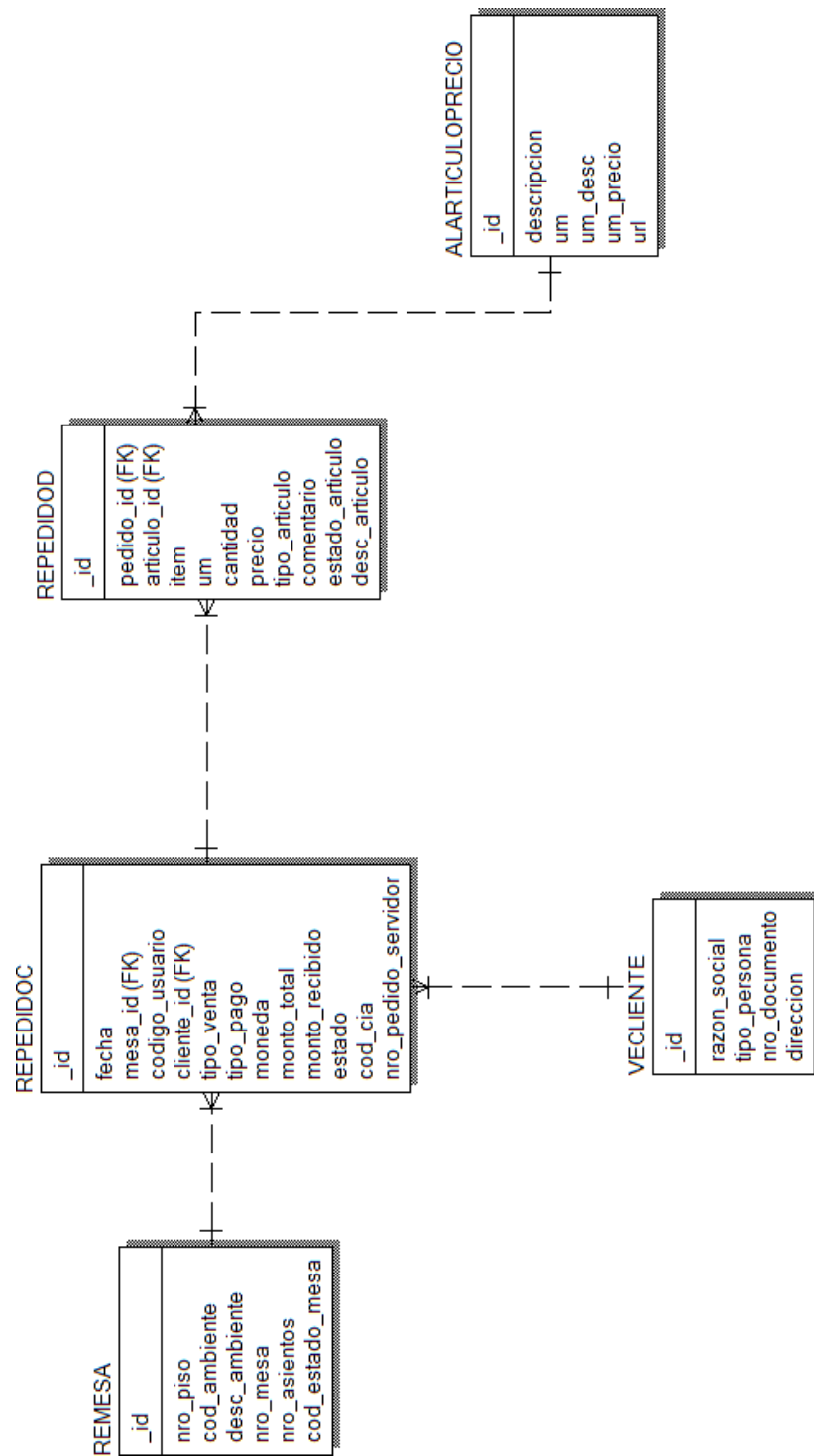


Ilustración 184 Diagrama Lógico - Séptimo Sprint

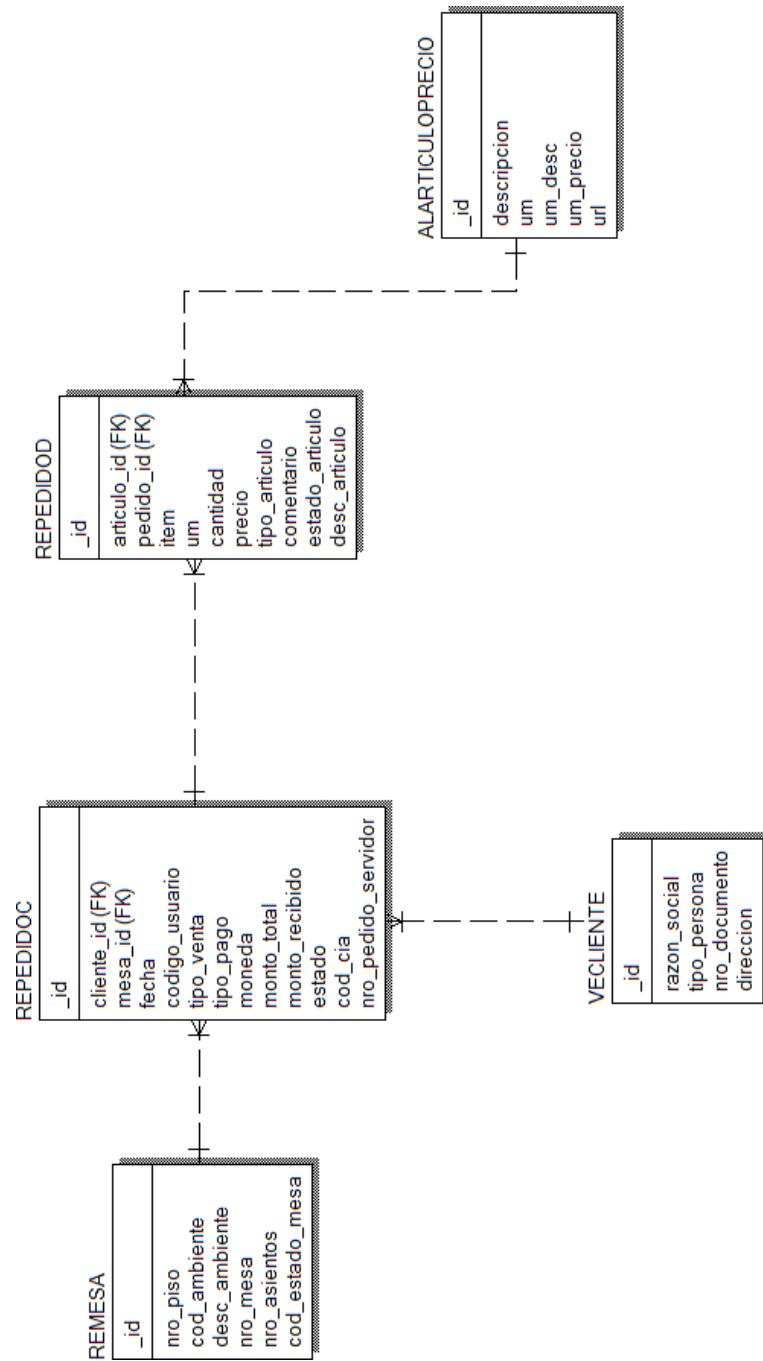


Ilustración 185 Diagrama Físico – Séptimo Sprint

4.6.7.6. DIAGRAMA DE COMPONENTES SPRINT 7

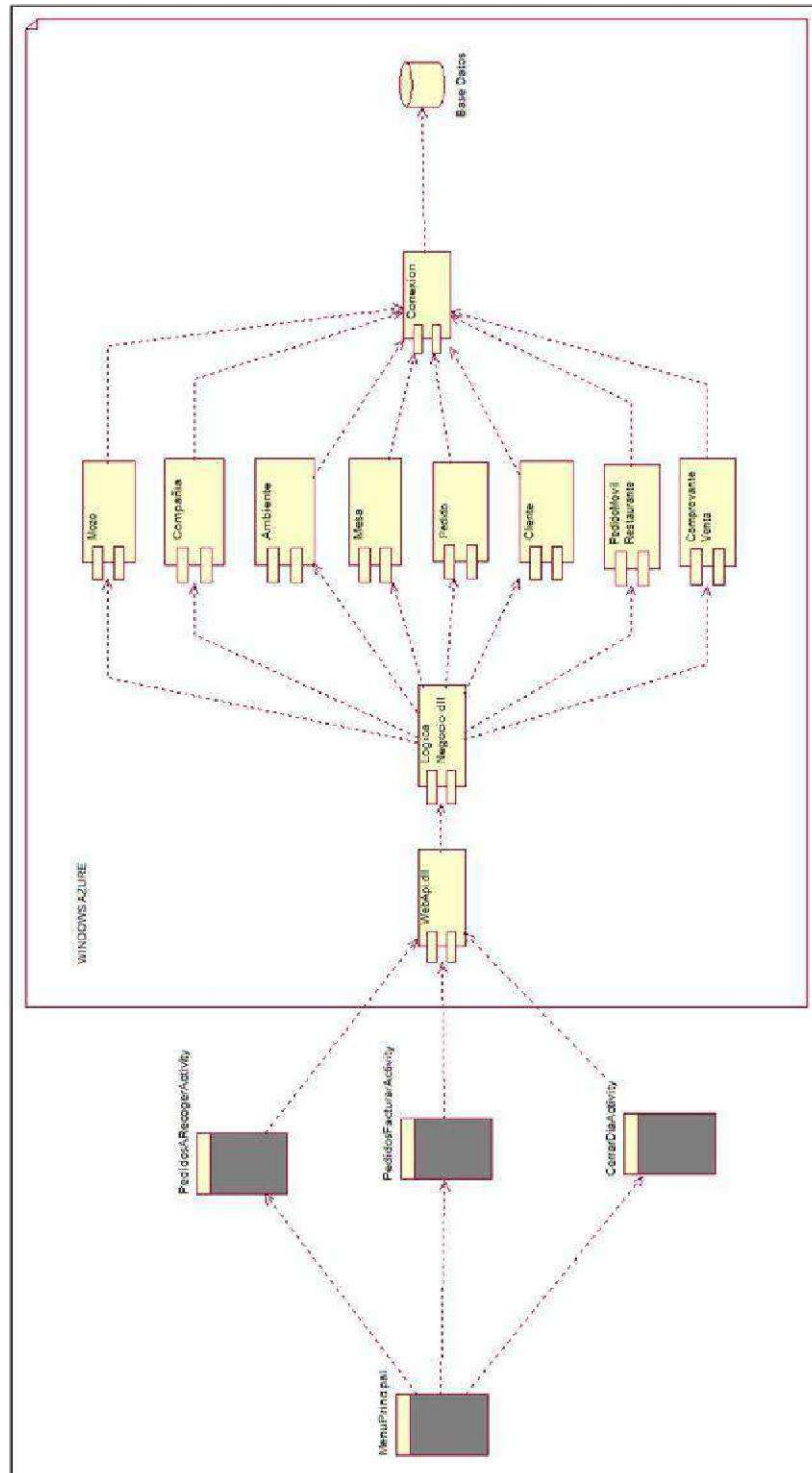
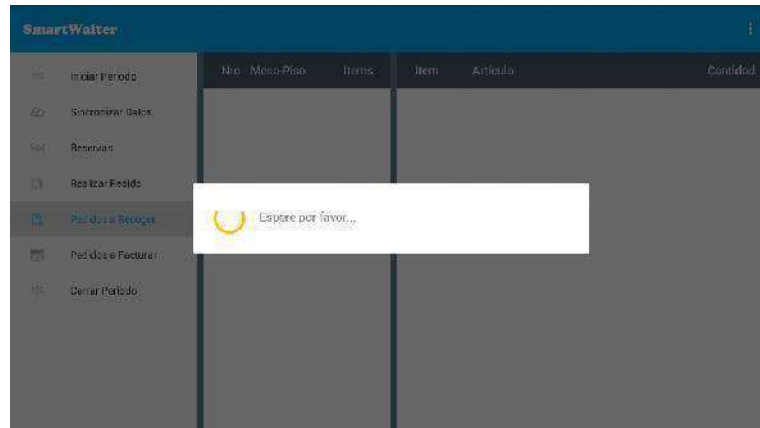


Ilustración 186 Diagrama de Componentes – Séptimo Sprint

4.6.7.7. INTERFAZ DE LAS HISTORIAS DE USUARIO

✓ Actualización de pedidos a recoger



The screenshot shows the SmartWaiter application interface with the table data loaded. The table has columns: Nro., Mesa-Piso, Items, Item, Artículo, and Cantidad. The data rows are as follows:

Nro.	Mesa-Piso	Items	Item	Artículo	Cantidad
1	25-1	0/2			
2	33-1	0/3			
3	29-1	0/7			

Ilustración 187 Actualización de pedidos a recoger. – Séptimo Sprint

✓ Selección de pedido y de los ítems específicos recogido

SmartWaiter						
	Nro.	Mesa-Piso	Items	Item	Artículo	Cantidad
Inicio Período	1	25-1	0/2	1	seco de cabrito	1.0
Sincronizar Datos	2	33-1	0/3	2	milanesa de pollo	1.0
Reservas	3	29-1	0/7	3	arroz con pato	1.0
Realizar Pedido				4	arroz con cuy	1.0
Pedidos a Recoger				5	pollo a la coracola	1.0
Pedidos a Facturar						
Cerrar Período						

← 3 Selected						
	Nro.	Mesa-Piso	Items	Item	Artículo	Cantidad
Inicio Período	1	25-1	0/2	1	seco de cabrito	1.0
Sincronizar Datos	2	33-1	0/3	2	milanesa de pollo	1.0
Reservas	3	29-1	0/7	3	arroz con pato	1.0
Realizar Pedido				4	arroz con cuy	1.0
Pedidos a Recoger				5	pollo a la coracola	1.0
Pedidos a Facturar						
Cerrar Período						

Ilustración 188 Selección de pedido y de los ítems específicos recogidos. – Séptimo Sprint

- ✓ Actualización de estado de pedido e ítems recogidos tanto en el servidor como en el dispositivo

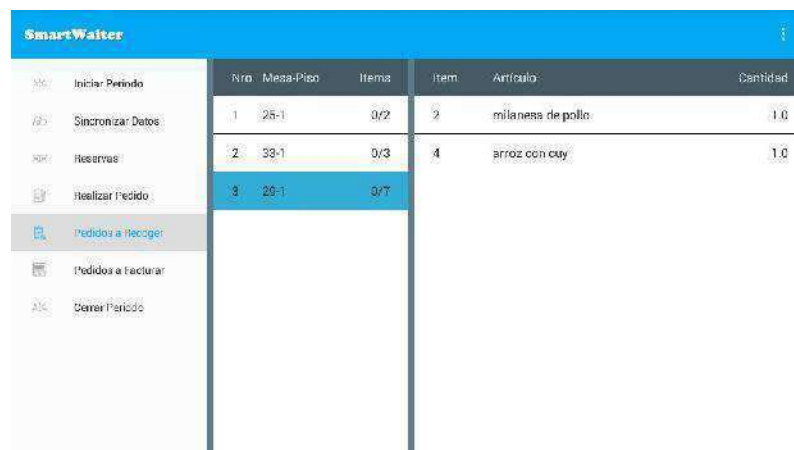
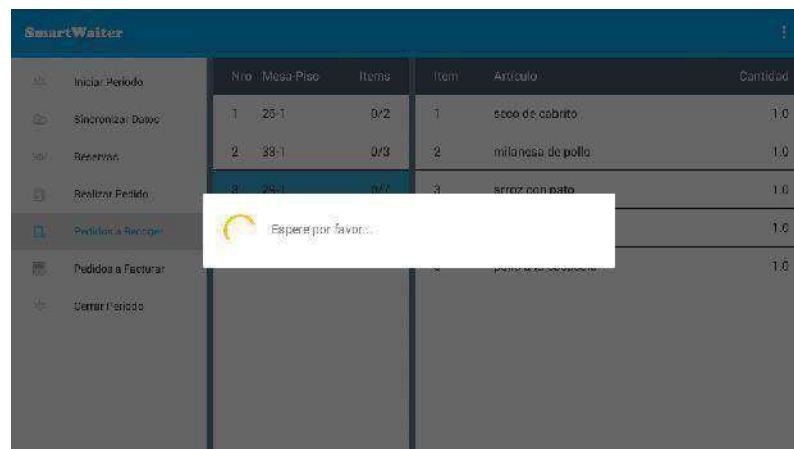
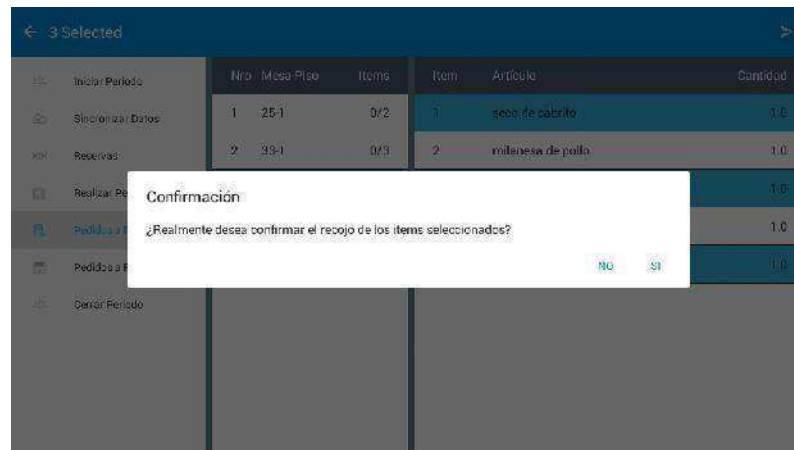


Ilustración 189 Actualización de estado de pedido e ítems recogidos tanto en el servidor como en el dispositivo – Séptimo Sprint

✓ Listado de pedidos a facturar

The screenshot shows the 'SmartWaiter' application interface. On the left is a sidebar with navigation icons and labels: 'Iniciar Período', 'Sincronizar Datos', 'Reservas', 'Realizar Pedido', 'Pedidos a Recoger', 'Pedidos a facturar' (highlighted), and 'Cerrar Período'. The main area features a table with columns 'Nro', 'Mesa', 'Piso', and 'Importe'. The first row contains the values '3', '29', '1', and '228.48'. To the right of the table is a form with fields for 'Tipo Venta' (Factura), 'Tipo Pago' (Efectivo), 'Cliente', 'RUC', 'Total' (0.00), 'Recibido' (0.00), and 'Restante' (0.00). At the bottom right are 'ACEPTAR' and 'CANCELAR' buttons.

Ilustración 190 Listado de pedidos a facturar– Séptimo Sprint

✓ Selección de pedido a facturar y búsqueda de cliente

This screenshot shows the same 'SmartWaiter' application interface as the previous one, but with the first row of the table selected. The 'Tipo Venta' field now shows 'Factura' and 'Tipo Pago' shows 'Efectivo'. The 'Cliente' field contains the text 'andean s.r.l.s sociedad anónima cerrada'. The 'RUC' field is populated with '20485245127'. The 'Total' field now displays '228.48', while 'Recibido' and 'Restante' remain at '0.00'. The 'ACEPTAR' and 'CANCELAR' buttons are still present at the bottom right.

Ilustración 191 Selección de pedido a facturar y búsqueda de cliente – Séptimo Sprint

- ✓ Envío de datos del pedido facturado y actualización del estado del pedido en el dispositivo

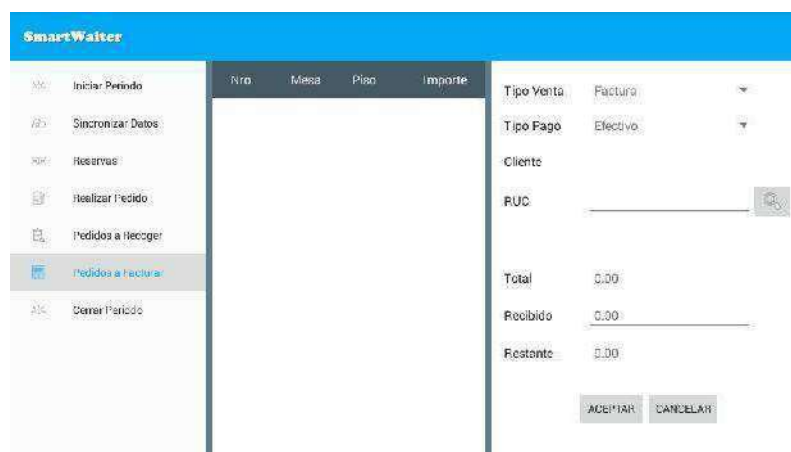
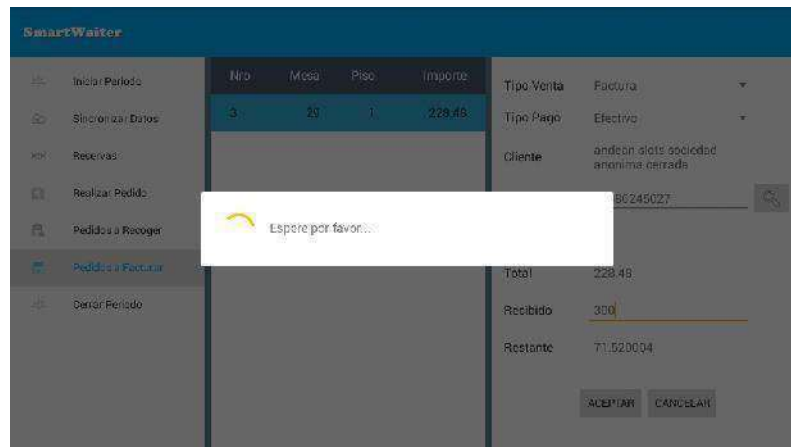


Ilustración 192 Envío de datos del pedido facturado y actualización del estado del pedido en el dispositivo– Séptimo Sprint

✓ Registro de cierre de periodo y eliminación de data local

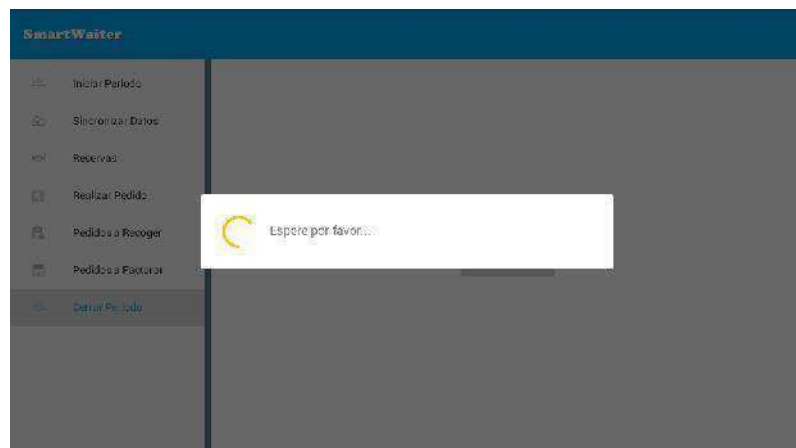


Ilustración 193 Registro de cierre de periodo y eliminación de data local– Séptimo Sprint

4.6.7.8. GRAFICANDO EL TABLÓN DE TAREAS

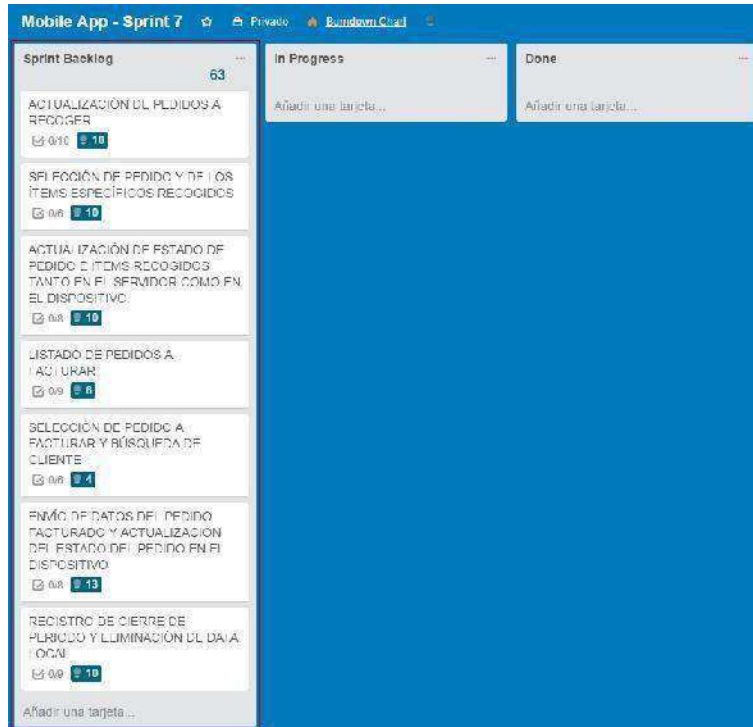


Ilustración 194 Historias de Usuario –Séptimo Sprint

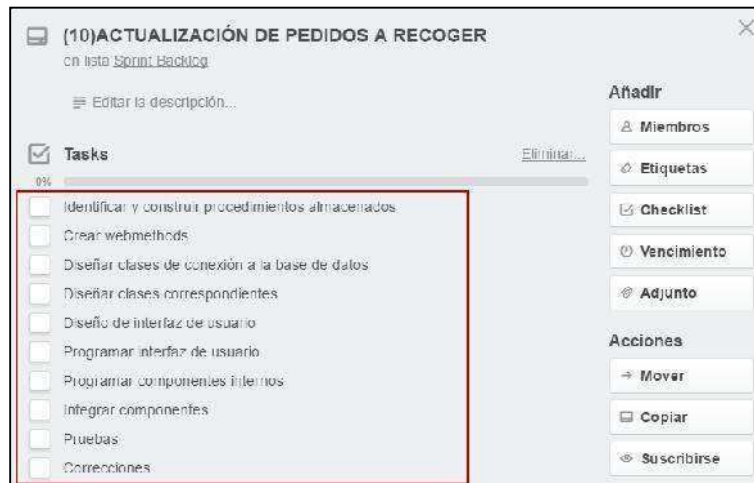


Ilustración 195 H. de Usuario Actualización de Pedidos a Recoger

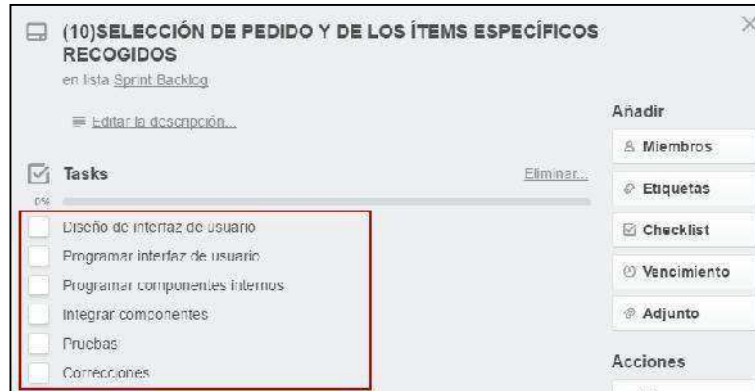


Ilustración 196 H. de Usuario Selección de Pedido y de los Ítems Específicos Recogidos

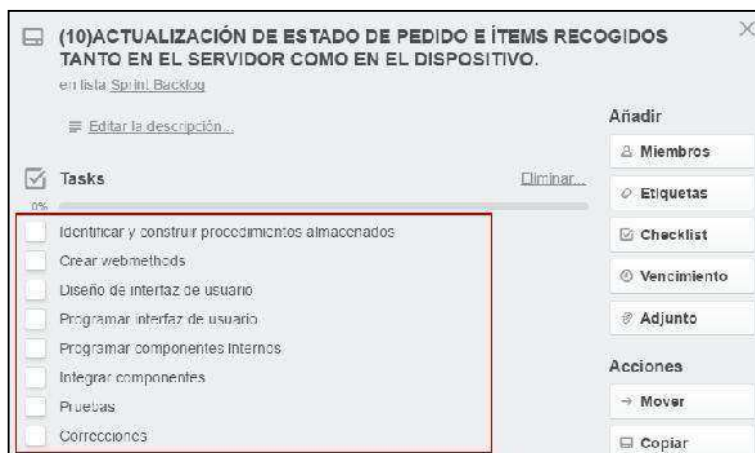


Ilustración 197 H. de Usuario Actualización de estado de Pedido e Ítems Recogidos tanto en el Servidor como en el Dispositivo

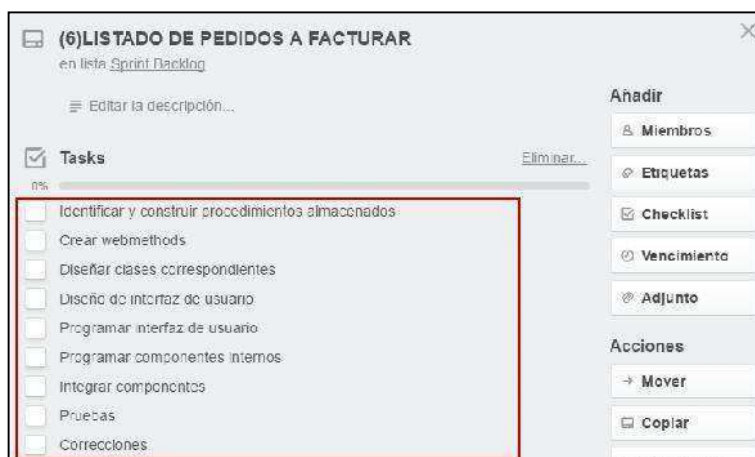


Ilustración 198 H. de Usuario Listado de Pedidos a Facturar

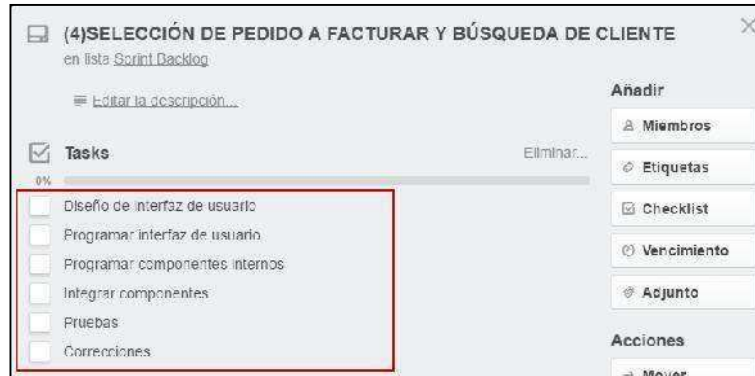


Ilustración 199 H. de Usuario Selección de Pedido a Facturar y Búsqueda de Cliente

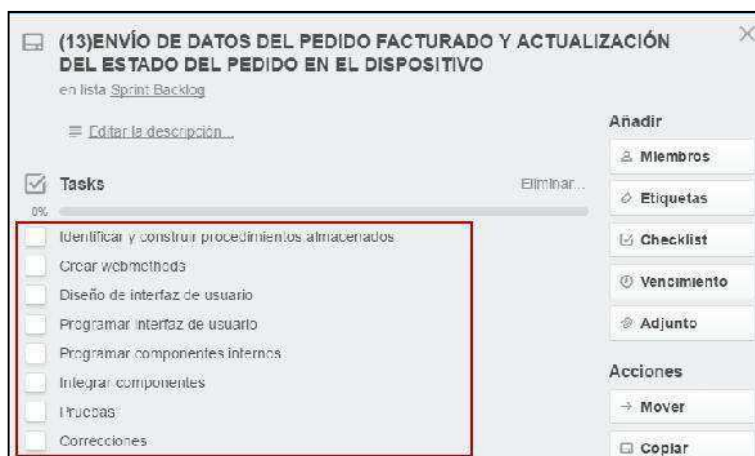


Ilustración 200 H. de Usuario Envío de Datos del Pedido Facturado y Actualización del Estado del Pedido en el Dispositivo

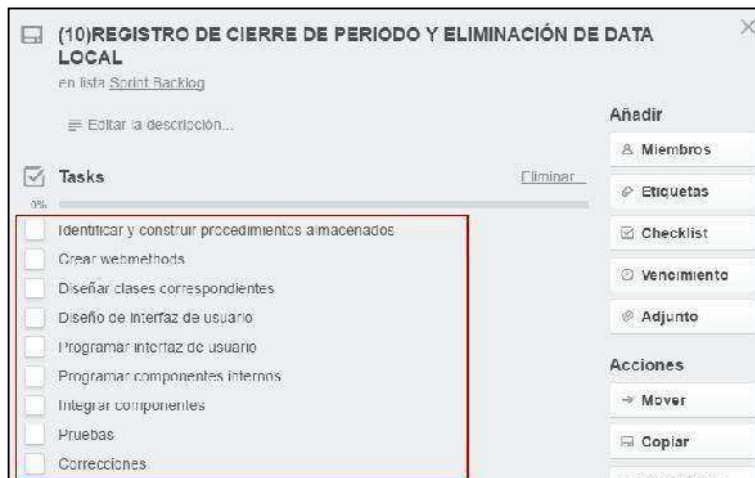


Ilustración 201 . H. de Usuario Registro de Cierre De Periodo y Eliminación de Data Local



Ilustración 202 Proceso de Tareas Historia 1 – Séptimo Sprint

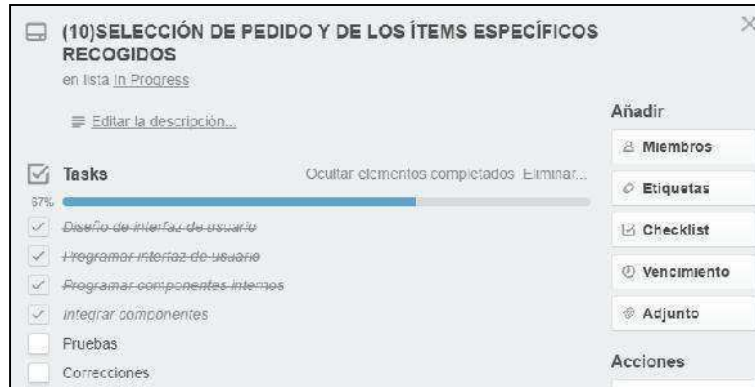
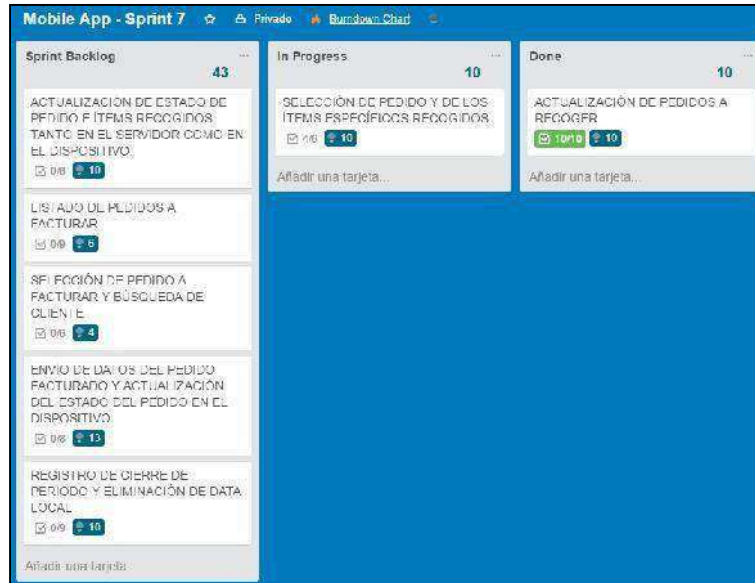
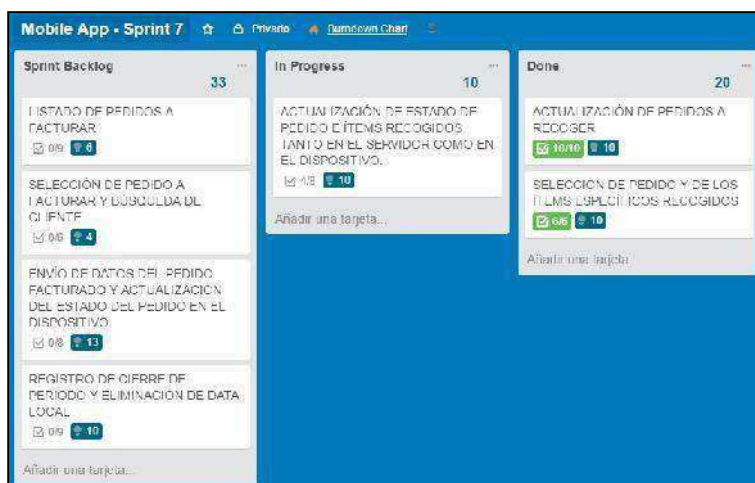


Ilustración 203 Proceso de Tareas Historia 2 – Séptimo Sprint



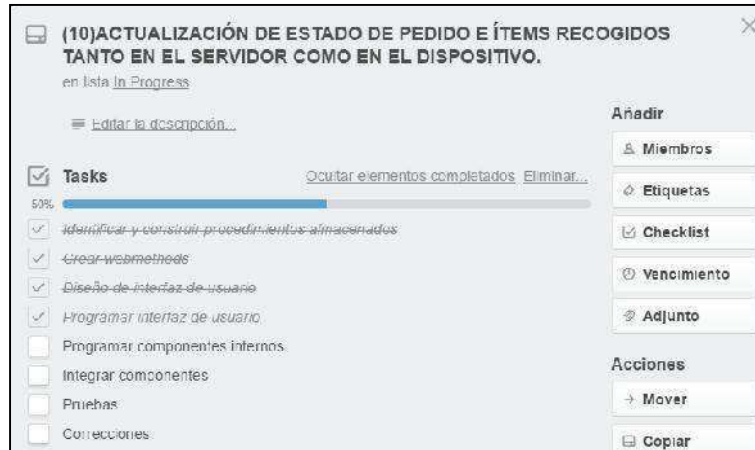


Ilustración 204 Proceso de Tareas Historia 3 – Séptimo Sprint

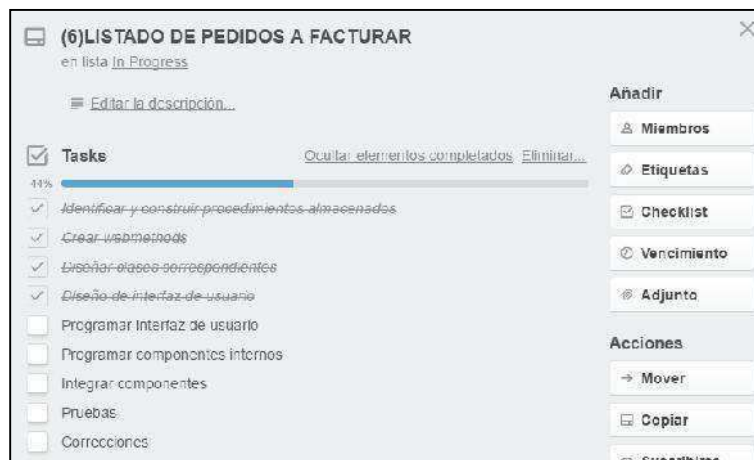


Ilustración 205 Proceso de Tareas Historia 4 – Séptimo Sprint

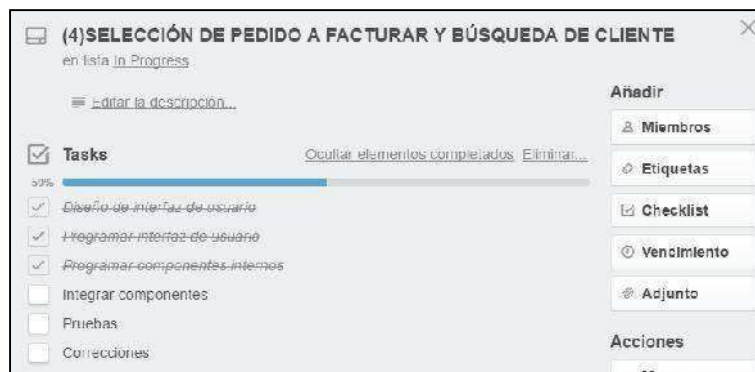


Ilustración 206 Proceso de Tareas Historia 5 – Séptimo Sprint

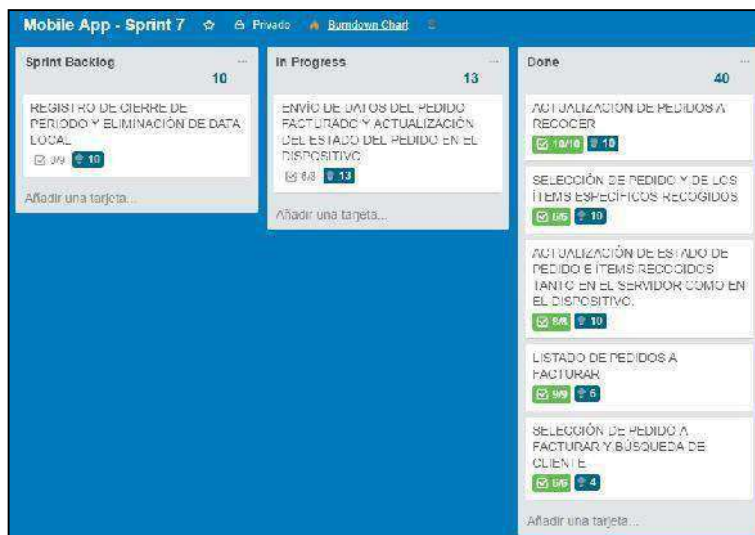
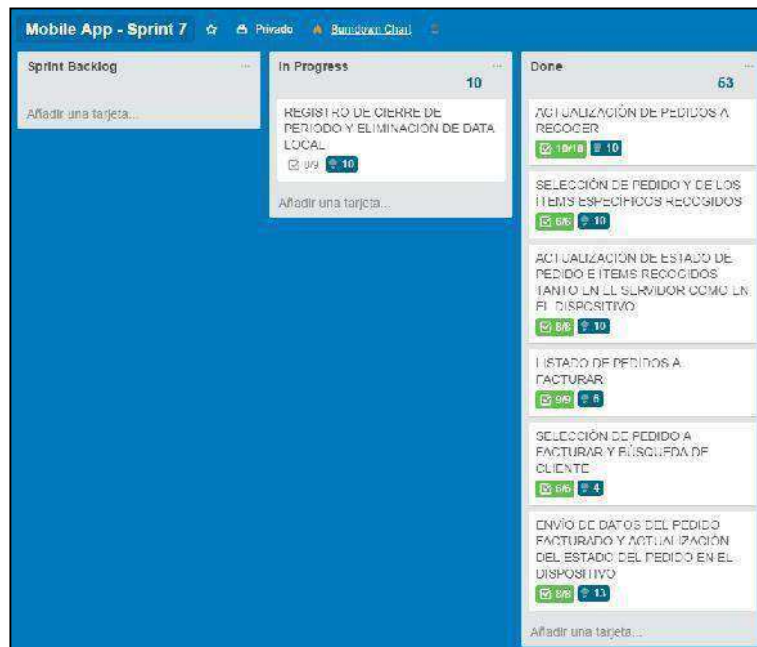




Ilustración 207 Proceso de Tareas Historia 6 – Séptimo Sprint



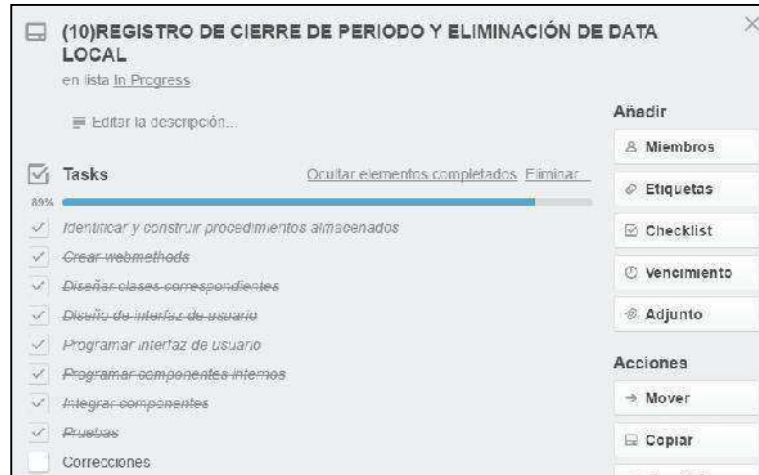


Ilustración 208 Proceso de Tareas Historia 7 – Séptimo Sprint

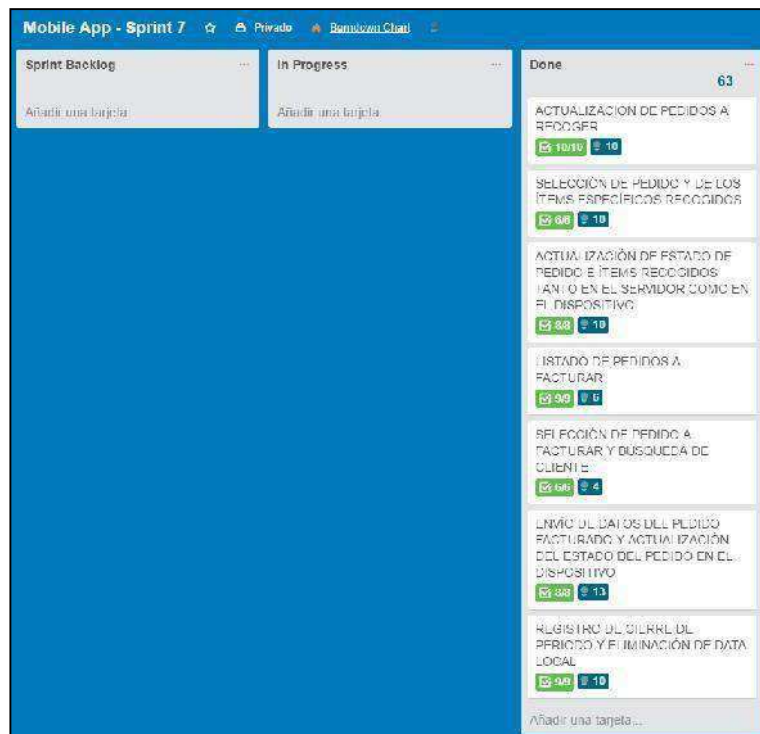


Ilustración 209 Historias de Usuario Concluidas –Séptimo Sprint

4.6.7.9. DIAGRAMA DE BURN DOWN



Ilustración 210 Diagrama de Burn Down – Séptimo Sprint

4.6.8. DIAGRAMA DE COMPONENTES

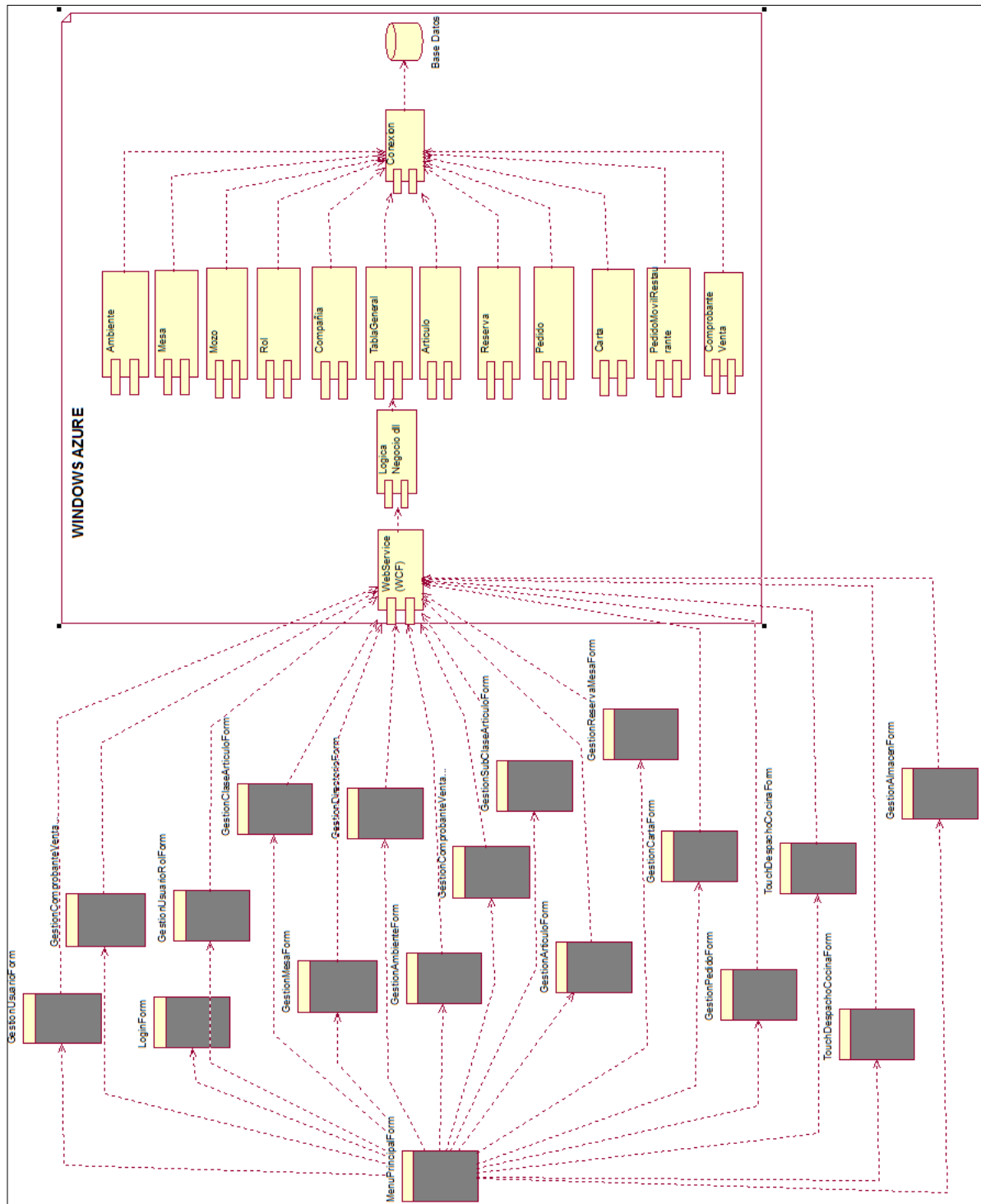


Ilustración 211 Diagrama de Componentes Escritorio

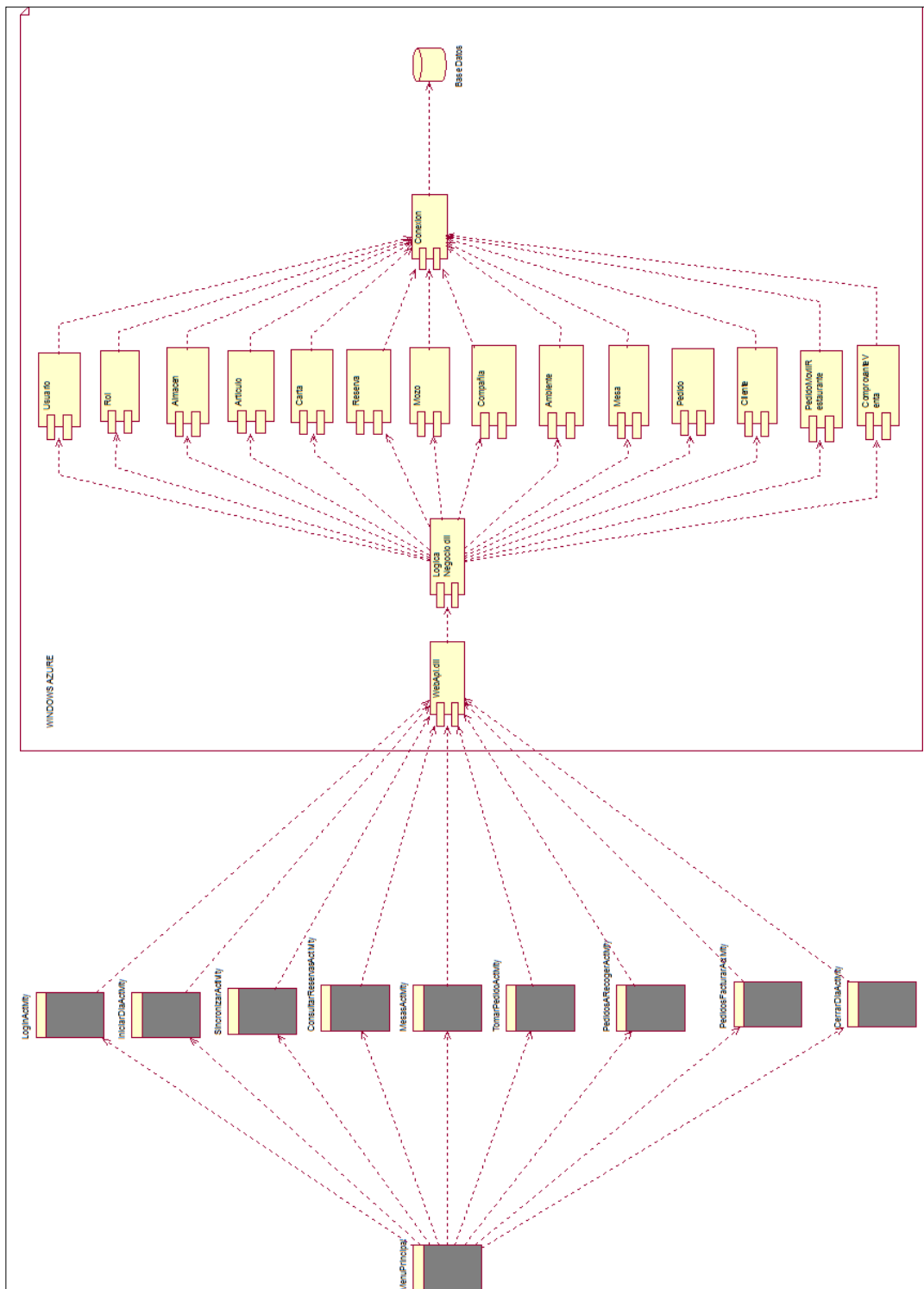


Ilustración 212 Diagrama de Componentes Movil

4.6.9. DIAGRAMA DE DESPLIEGUE

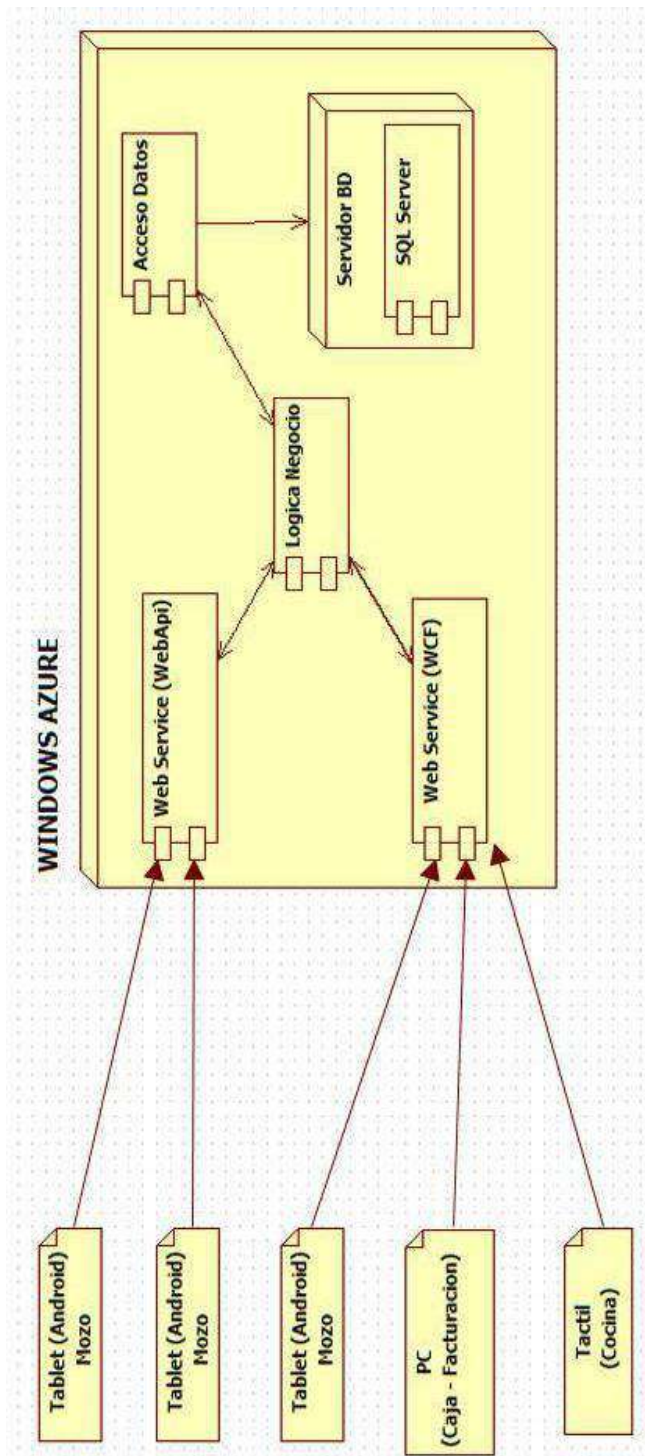


Ilustración 213 Diagrama de Despliegue

4.7. REUNIONES CON EL CLIENTE

FECHA	ACTIVIDAD	OBSERVACIONES
26/10/2015	DEFINICIÓN DEL SPRINT 1	
16/11/2015	REVISIÓN SPRINT 1	Se terminó satisfactoriamente el sprint 1.
18/11/2015	DEFINICIÓN SPRINT 2	
09/12/2015	REVISIÓN SPRINT 2	Se terminó satisfactoriamente el sprint 2.
11/12/2015	DEFINICIÓN SPRINT 3	
05/01/2016	DUDAS SPRINT 3	Las dudas fueron resueltas de manera clara.
06/01/2016	REVISIÓN SPRINT 3	Se terminó la revisión sin ningún inconveniente de por medio
07/01/2016	DEFINICIÓN SPRINT 4	
26/01/2016	REVISIÓN SPRINT 4	Se terminó satisfactoriamente el sprint 4.
01/02/2016	DEFINICIÓN SPRINT 5	
18/02/2016	REVISIÓN SPRINT 5	Se terminó satisfactoriamente el sprint 5.
22/02/2016	DEFINICIÓN SPRINT 6	
23/03/2016	REVISIÓN SPRINT 6	Se terminó satisfactoriamente el sprint 6.
28/03/2016	DEFINICIÓN SPRINT 7	
27/04/2016	REVISIÓN SPRINT 7	Se terminó satisfactoriamente el sprint 7.

Tabla 78. Reuniones con el cliente

4.8. PLAN DE INTEGRACION

4.8.1. Consideraciones de Seguridad de Base de Datos

✓ Inyección a la Base de Datos

Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.

La vulnerabilidad se puede producir automáticamente cuando un programa "arma descuidadamente" una sentencia SQL en tiempo de ejecución, o bien durante la fase de desarrollo, cuando el programador explicita la sentencia SQL a ejecutar en forma desprotegida

Manejando procedimientos almacenados, en el cual se envían los parámetros evita que la inyección SQL pueda ser usada.

4.8.2. Consideraciones de Seguridad de la Aplicación

✓ Autenticación de Usuarios

Cuando un usuario inicia sesión, dependiendo de su rol y los permisos que tiene asignados, se valida que solo se le permite acceder a las opciones de menú que tiene permiso, esto para evitar que cualquier usuario externo o interno que no tenga permisos pueda hacer o deshacer en la aplicación.

4.8.3. Consideraciones de Seguridad a Nivel de Opciones de Menú

Dado que el sistema elaborado cuenta con un mantenimiento de accesos por usuario, el usuario administrador tendrá la facilidad de otorgar dichos permisos / accesos a los demás usuarios.

4.8.4. Consideraciones de Seguridad de la Aplicación Móvil

Los teléfonos inteligentes tienen la ventaja del uso de redes geográficamente distribuidas a nivel global. Lo cual los hace vulnerables a riesgos derivados por virus o ataques informáticos. Se han presentado diversas soluciones para dispositivos móviles, que ofrecen protección y una mayor seguridad en el uso diario de estos terminales. Sin embargo, las soluciones de seguridad, no importa cual, no son suficientes, puesto que una buena parte de la responsabilidad de la seguridad de dispositivos móviles recae en los propios usuarios que, es el eslabón más débil de la cadena de seguridad.

Si se unen las soluciones de seguridad que ofrecen mediante aplicativos junto con las buenas prácticas es posible conseguir un alto nivel de seguridad.

- a) Para evitar el acceso no autorizado a los dispositivos y a la información que contienen hay que protegerlos siempre con una clave.
- b) Cada vez es más frecuente la conexión a servicios de Internet desde dispositivos móviles, como el correo electrónico, redes sociales, entidades bancarias, etc. Para proteger las cuentas de acceso a los distintos servicios, conviene utilizar soluciones que permiten almacenar las cuentas de acceso de forma segura en un dispositivo, de forma que, además, se puedan utilizar cuando sea necesario, sin necesidad de tener que recordarlas.
- c) Además de lo dicho anteriormente, cada vez es más frecuente que los usuarios estén conectados de forma permanente a algunos servicios. Es conveniente se cerraran las sesiones y aplicaciones una vez que se ha terminado de utilizarlas, muy especialmente aquellas más sensibles como las aplicaciones para operaciones bancarias.

- d) Los dispositivos móviles suelen incorporar opciones que aportan mayor seguridad, que se pueden configurar en el dispositivo, como, por ejemplo, la activación de navegación privada, avisos de fraude, etc. Es conveniente configurar aquellas opciones que incorpora el dispositivo móvil y que permiten obtener un mayor nivel de seguridad.
- e) Las copias de seguridad son una medida de protección fundamental, que debe aplicarse a cualquier tipo de dispositivo. En el caso de los dispositivos móviles cobra más importancia puesto que, por su capacidad de traslado, están más expuestos a robo o pérdida. Por ello, es fundamental realizar copias de seguridad de forma regular.
- f) Al igual que ocurre en un ordenador de escritorio o portátil, los dispositivos móviles también deben pasar, de vez en cuando, una revisión y mantenimiento. Esto permite borrar cierta información generada internamente durante el uso del dispositivo y mejorar así su funcionamiento general, aumentando su velocidad y eliminando información que pudiera aprovechar un atacante.

4.9. Puesta En Marcha

4.9.1. Establecimiento del Plan de Implantación

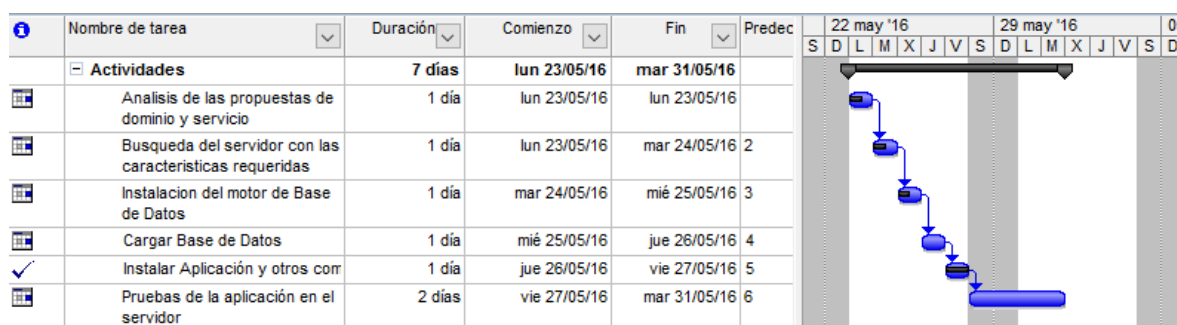


Ilustración 214 Establecimiento del Plan de Implantación

4.9.2. Especificaciones Técnicas

- PCs Clientes
 - ✓ Requerimientos de hardware:
 - ❖ Core i3 de 3.40 GHZ.
 - ❖ Memoria RAM 4 GB DDR3.
 - ❖ Disco Duro 500 GB SATA 6.0 GB/S / 7200 RPM.
 - ❖ Monitor de 15 pulgadas o superior.
 - ❖ Impresora.
 - ❖ Tarjeta de red 10 Mb.
 - ✓ Requerimientos de software:
 - ❖ Controladores de impresora.
 - ❖ Adobe Reader 11.0 o superior.
 - ❖ Sistema operativo Windows 7 Ultimate de 32 bits
- Tablet:
 - ✓ Requerimientos de hardware y software:
 - ❖ Pulgadas: 8"
 - ❖ Sistema Operativo: Android 4.1 (JELLY BEAN)
 - ❖ Batería: Li-ion 5000 mAh
 - ❖ Procesador: Dual Core Cortex A9 1.6 GHz
 - ❖ Memoria RAM: 1 Gb
 - ❖ Memoria Interna: 8 Gb

CAPITULO V

COSTOS Y BENEFICIOS

5. Costos y Beneficios

5.1. Análisis de Costos

5.1.1. Costo de Software

En el desarrollo se utilizó Visual Studio 2015 como herramienta de desarrollo, SQL 2014 como gestor de base de datos, Bizagi para diagramar los procesos, Erwin Data Modeler para el modelamiento de clases, todas estas herramientas se han utilizado en su versión express (gratuita) por lo que los costos en ese sentido han sido nulos.

TIPO	CANTIDAD	VALOR UNITARIO	SUBTOTAL
Visual Studio 2015	1	S/.0.0	S/. 0.00
SQL 2014	1	S/ 0.0	S/. 0.00
BIZAGI	1	S/. 0.00	S/. 0.00
ERWIN	1	S/. 1100.70	S/. 1100.70
TOTAL			S/. 1100.70

Tabla 79. Inversión en Software

Windows Azure

PRODUCTO	PRECIO
BD SQL Administrada x Mes Aprox.	10.34
Servicio de Aplicaciones x Mes Aprox.	115.20
TOTAL	125.54

Tabla 80. Inversión en Software

5.1.2. Costo de Personal

TIPO	CANTIDAD MES	PRECIO POR MES	SUBTOTAL
Analista	1	S/. 1500.00	S/. 1500.00
Programador	2	S/. 1800.00	S/. 3600.00
Control Calidad	1	S/. 1200.00	S/. 1200.00
TOTAL			S/. 6300.00

Tabla 81. Recursos Humanos para el Desarrollo

5.1.3. Costo de Servicio y Materiales

Recursos materiales

ARTICULO	PRECIO
Paquete de hojas bond x 5 meses	S/. 55.00
Recargas de cartucho x 5 meses	S/. 40.00
Fólder de Manila 40 unidades	S/. 20.00
Lapiceros x 5 meses	S/. 12.00
Total	S/. 127.00

Tabla 82. Recursos de Materiales para el Desarrollo

Servicios

PRODUCTOS	PRECIO
Luz x 5 meses	S/. 200.00
Movilidad x 5 meses	S/. 100.00
Alimentación x 5 meses	S/. 300.00
Internet x 5 meses	S/. 150.00

Teléfono x 5 meses	S/. 200.00
Total	S/. 950.00

Tabla 83. Costo Total por los Servicios

5.1.4. Costos de Hardware

En lo que respecta al hardware para el desarrollo del proyecto, fue necesario la adquisición de algunos dispositivos hardware como se detallan a continuación.

PRODUCTOS	PRECIO
2 laptops HP- Intel Core i5 Memoria RAM: 4 GB DDR3, Disco duro: 500 GB, 5400 RPM.	S/. 3600.00
1 Table Android 4.0 RAM: 512 MB	S/. 180.00
Total	S/. 3780.00

Tabla 84. Costos de Hardware

5.1.5. Costos de Implementación

PRODUCTOS	PRECIO
Capacitación	S/. 1000.00
Total	S/. 1000.00

Tabla 85. Costos de Implementación

5.1.6. Resumen de Costos

CONCEPTO	SUB TOTAL
Costo de Software	S/. 1226.24
Costos de Personal	S/. 6300.00
Costo de Servicio y Materiales	S/. 1077.00
Costos de Hardware	S/. 3780.00
Costos de Implementación	S/. 1000.00

	Total	S/. 13383.24
--	--------------	---------------------

Tabla 86. Resumen de Costos

5.2. Beneficios

5.2.1. Beneficios Tangibles

- ✓ Mejora la productividad de los procesos y el personal.
- ✓ Reducir el costo de los productos adquiridos.
- ✓ Automatizado de pedidos y pagos, el procesamiento de pagos y la reducción de comandas.
- ✓ Aumento de ingresos

5.2.2. Beneficios Intangibles

- ✓ Aumenta la transparencia organizativa y responsabilidad.
- ✓ Precisa y un acceso más rápido a los datos para tomar decisiones oportunas.
- ✓ Optimización del recurso tiempo en la preparación de pedidos para su despacho.
- ✓ Mejora satisfacción del cliente.
- ✓ Aumenta la calidad de servicio.
- ✓ Más control, lo que reduce el riesgo de mala utilización de los recursos.
- ✓ Facilita la planificación estratégica.

5.2.3. Beneficio de Proyecto

El beneficio del proyecto es anual el cual se ve reflejado:

✓ Reducción de costos en material

La implementación del proyecto implicará una reducción de costos con respecto a la gestión manual como se realiza actualmente.

PROCESO MANUAL	CANT.	P. UNIT.	TOTAL
Talonario de Comanda	300	S/. 20.00	S/. 6000.00
TOTAL			S/. 6000.00

Tabla 87. Costo con el proceso actual

SISTEMA	CANT.	P. UNIT.	BENEFICIO
Talonario de Comanda	0	S/. 20.00	S/. 0.00
TOTAL			S/. 0.00

Tabla 88. Costo con el sistema

MATERIAL DE OFICINA	TOTAL
PROCESO MANUAL	S/. 6000.00
SISTEMA	S/. 0.00
TOTAL, BENEFICIO	S/. 6000.00

Tabla 89. Beneficio del sistema

❖ Valor Actual Neto (VAN)

Llamado también valor presente neto, representa el excedente generado por un proyecto en términos absolutos después de haber cubierto los costos de inversión, de operación y de uso del capital. En resumen, el VAN es la suma algebraica de los costos beneficios generados por el proyecto.

Consideramos una tasa de interés bancaria anual del 15%.

	AÑO 0	AÑO 1	AÑO 2	AÑO 3
Costos de Inversión	S/. 5006.30	S/. 0.00	S/0.00	S/. 0.00
Costos de Desarrollo	S/. 7377.00	S/. 0.00	S/0.00	S/0.00
Beneficio Anual		S/. 6000.00	S/. 6000.00	S/. 6000.00
Total, VAN	S/. 12383.00	S/. 6000.00	S/. 6000.00	S/. 6000.00

Tabla 90. Comparativo: Inversión, Desarrollo, Operación, Beneficio

A continuación, aplicaremos la siguiente fórmula:

$$VAN = \sum_{i=0}^n \frac{B_i - C_{i_n}}{(1 + \frac{i}{100})^i}$$

$$VAN = -12383.00 + \frac{6000.00}{(1 + \frac{15}{100})^1} + \frac{6000.00}{(1 + \frac{15}{100})^2} + \frac{6000.00}{(1 + \frac{15}{100})^3}$$

$$VAN = 13676.04$$

Esto significa que se está generando más efectivo del que se necesita por lo que se deduce que este proyecto es rentable.

❖ Tiempo de la recuperación de la inversión (TIR)

Mide en cuanto tiempo se recuperará el total de la inversión a valor presente, es decirnos revela la fecha en la cual se

cubre la inversión inicial en años, meses y días, para calcularlo se utiliza la siguiente Fórmula:

$$TR = \frac{\text{Inversión Total}}{\text{Promedio Beneficio Neto}}$$

Dónde: promedio beneficio Neto= beneficio – Costo Operación

$$TR = \frac{12383.00}{6000.00}$$

$$TR = 2.06$$

Por lo tanto, el tiempo necesario para recuperar la inversión será en aproximadamente 2.06 años.

❖ Tasa interna de Retorno

Es la tasa de descuento que igual el valor actual de los beneficios y el valor de los costos. Representa la tasa de rendimiento a la cual el proyecto se hace diferente, es decir, cuando el VAN =0.

$$VAN = 0 = -12383.00 + \frac{6000.00}{(1+i)^1} + \frac{6000.00}{(1+i)^2} + \frac{6000.00}{(1+i)^3}$$

$$TIR > \text{TASA ACTUAL}$$

$$TIR = 21\% > 15\%$$

Dado que la tasa interna de retorno es muy superior a la tasa actual de efectividad, se concluye que el proyecto va a tener un rendimiento superior al esperado.

CAPITULO VI

CONCLUSIONES

6. CONCLUSIONES

- La recopilación precisa de la información de los procesos permitió implementar el registro del pedido integrado con los procesos de facturación y gestión de cocina.
- El diseño y arquitectura final de la aplicación permitió hacer posible la interacción del sistema y el usuario a través del uso de dispositivos móviles (tabletas)
- Las decisiones de plataformas tecnológicas tomadas para este proyecto fueron acertadas, tanto en el manejo de servicios como en la aplicación móvil.
- La metodología SCRUM utilizada para el manejo del proyecto resultó positiva de sobremanera, ya que permitió una mejor adaptación a los constantes cambios que se presentaban durante el tiempo de desarrollo del mismo.
- El diseño transaccional de la base de datos permitió cubrir las diferentes necesidades de datos que demandaban cada uno de los procesos que comprendió la implementación del sistema.
- El diseño sencillo y fácil de usar de la aplicación permitió cumplir los requerimientos identificados y solicitados por los usuarios.

CAPITULO VII

RECOMENDACIONES

7. RECOMENDACIONES

- Implementar la funcionalidad de notificaciones “push” en el sistema móvil. Esto permitirá que los dispositivos móviles en los que se encuentra instalada la aplicación sean alertados en tiempo real cuando se produzcan cambios en los datos relevantes para su funcionamiento, sin que sea necesario configurar alarmas o tener que refrescar continuamente la aplicación.
- Mantener una actualización frecuente de la información y productos que ofrece el restaurante ya que la aplicación está desarrollada para que brinde información actualizada haciendo peticiones al servidor.
- En caso de que existan nuevos requerimientos para ser implementados en la aplicación móvil, deberán desarrollar nuevas versiones y realizar las pruebas que sean necesarias para su correcto funcionamiento, antes de ser facilitadas a los usuarios.
- Teniendo en cuenta que los datos están expuestos a ser afectados sea por un virus informático, eliminación de datos accidentalmente o cualquier otro tipo de catástrofe informática es recomendable hacer respaldos periódicamente de la base de datos.
- Realizar un estudio periódico de satisfacción de los clientes del proceso de pedidos y facturación para obtener nuevos requerimientos que contribuirán con la mejora del negocio.
- Publicar la aplicación en la tienda de aplicaciones de Google (Google Play), ya que de esta manera los usuarios podrán tener acceso a la aplicación y sus actualizaciones de manera automática.

CAPITULO VIII

REFERENCIAS BIBLIOGRÁFICAS

8. Referencias Bibliográficas

8.1. Bibliografía

- ✓ **[LIB01]** Echeverría Tobón, Luis Miguel y Luz Elena Delgado Carmona (2007). *“Caso práctico de la metodología XP al desarrollo de software”*.
- ✓ **[LIB02]** Palacios Juan (2006). *“El Modelo Scrum”*, México: Editorial Coloriuris.
- ✓ **[LIB03]** Santi Caballé y Fatos Xhafa (2008). *“Aplicaciones Distribuidas en Java con tecnología RMI”*, España: Editorial Delta.
- ✓ **[LIB04]** Conesa Carlt, Jordi y Rius Gavidia, Angels (2011). *“Introducción a .NET”*, España: editorial UOC.
- ✓ **[LIB05]** Gabillaud, Jerome (2010). *“SQL Server 2008”*, México: Ediciones ENI.

8.2. Linografía

- ✓ **[WEB01]** WEBAPI.
[http://msdn.microsoft.com/es-es/library/hh833994\(v=vs.108\).aspx](http://msdn.microsoft.com/es-es/library/hh833994(v=vs.108).aspx)
- ✓ **[WEB02]** Microsoft Azure.
https://es.wikipedia.org/wiki/Microsoft_Azure
- ✓ **[WEB03]** Windows Communication Foundation (WCF).
<http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>
- ✓ **[WEB04]** Native Apps vs Web Apps.
[https://msdn.microsoft.com/es-es/library/ms731082\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ms731082(v=vs.110).aspx)
- ✓ **[WEB05]** BIZAGI.
Bizagi – 2012
<http://www.bizagi.com>
- ✓ **[WEB06]** SQLite.
SQLite – 2014
<http://www.sqlite.org/copyright.html>

✓ **[WEB07]** Oracle.

Oracle – 2014

<http://www.oracle.com/index.html>

✓ **[WEB08]** Android.

Android – 2014

<http://www.android.com/>

[https://msdn.microsoft.com/es-es/library/ms731082\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ms731082(v=vs.110).aspx)

ANEXOS

ANEXO 1. Acta N° 01 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	1
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	26/10/2015 – 11:00 am
FECHA FIN	26/10/2015 - 02:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 1
OBJETIVOS	
Loguear usuario.	
Registrar usuario	
Registrar rol	
Registrar ambiente	
Registrar mesa	
Registrar directorio	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 2. Acta N° 02 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	2
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	16/11/2015 – 09:00 am
FECHA FIN	16/11/2015 – 03:30 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	PRUEBAS SPRINT 1
OBJETIVOS	
Hacer pruebas de la entrega realizada por los desarrolladores	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 3. Acta N° 03 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	3
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	17/11/2015 – 03:00 pm
FECHA FIN	17/11/2015 – 06:00 pm
LUGAR	RESTAURANTE CASA DESCALZI

FASE DEL PROYECTO	CORRECCIONES SPRINT 1
OBJETIVOS	
Mostrar las correcciones realizadas y pasar al siguiente sprint	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 4. Acta N° 04 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	4
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	18/11/2015 – 09:00 am
FECHA FIN	18/11/2015 - 11:30 am
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 2
OBJETIVOS	
Registrar mozo.	
Registrar clase – subclase artículos	
Registrar almacén	
Registrar artículo	
Registrar carta	
Registrar reservación mesa	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 5. Acta N° 05 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	5
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	09/12/2015 – 09:00 am
FECHA FIN	09/12/2015 - 02:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	REVISIÓN SPRINT 2
OBJETIVOS	
Realizar la revisión por la entrega de las historias que comprende el sprint 2	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 6. Acta N° 06 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	6
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	11/12/2015 – 09:00 am
FECHA FIN	11/12/2015 – 12:00 am
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 3
OBJETIVOS	
Registrar a clientes	
Registrar pedido	
Registrar comprobante venta	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 7. Acta N° 07 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	7
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	05/01/2016
FECHA FIN	05/01/2016
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	REVISIÓN SPRINT 3
OBJETIVOS	
Verificar que las historias de usuario funcionen tal cual se ha indicado.	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 8. Acta N° 08 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	8
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	07/01/2016 – 08:30 am
FECHA FIN	07/01/2016 – 11:30 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 4
OBJETIVOS	

Mostrar y realizar el despacho de cocina después que el mozo realice el pedido desde el dispositivo móvil. Poder realizar el seguimiento de los pedidos.

PERSONA ENTREVISTADA

FIDEL ESPINOZA CASTRO

ANEXO 9. Acta N° 09 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	9
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	10/01/2016 - 09:00 am
FECHA FIN	10/01/2016 - 10:30 am
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	REVISIÓN SPRINT 4
OBJETIVOS	
Resolver las dudas de los desarrolladores con relación al proceso del despacho de cocina.	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 10. Acta N° 10 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	10
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	28/01/2016
FECHA FIN	29/01/2016
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	PRUEBAS SPRINT 4
OBJETIVOS	
Verificar la entrega final del sprint 4 y dar el visto bueno para el paso al siguiente sprint.	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 11. Acta N° 11 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	11
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	01/02/2016 – 09:00 am
FECHA FIN	01/02/2016 - 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 5
OBJETIVOS	
Configurar conexión	
Autenticar usuario	
Seleccionar empresa y registrar credenciales de usuario	
Registrar inicio de periodo en el servidor y en el dispositivo móvil	
Registro local de datos para toma de pedidos	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 12. Acta N° 12 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	12
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	18/02/2016 – 09:00 am
FECHA FIN	18/02/2016 – 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	PRUEBAS SPRINT 5
OBJETIVOS	
Hacer pruebas de la entrega realizada por los desarrolladores	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 13. Acta N° 13 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	13
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES

FECHA INICIO Y HORA	19/02/2016 – 05:00 pm
FECHA FIN	19/02/2016 – 07:30 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	CORRECCIONES SPRINT 5
OBJETIVOS	
Mostrar las correcciones realizadas y pasar al siguiente sprint	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 14. Acta N° 14 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	14
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	22/02/2016 – 09:00 am
FECHA FIN	22/02/2016 - 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 6
OBJETIVOS	
Listado general actualizado de mesas	
Búsqueda de reservas y listado actual de mesas reservadas	
Selección de mesa y actualización de su estado en el dispositivo y en el servidor	
Listado de categorías	
Listado de artículos de la categoría seleccionada	
Mantenimiento de pedido en el dispositivo	
Envío del pedido al servidor para su procesamiento	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 15. Acta N° 15 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	15
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES

FECHA INICIO Y HORA	23/03/2016 – 09:00 am
FECHA FIN	23/03/2016 – 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	PRUEBAS SPRINT 6
OBJETIVOS	
Hacer pruebas de la entrega realizada por los desarrolladores	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 16. Acta N° 16 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	16
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	25/03/2016 – 05:00 pm
FECHA FIN	25/03/2016 – 07:30 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	CORRECCIONES SPRINT 6
OBJETIVOS	
Mostrar las correcciones realizadas y pasar al siguiente sprint	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 17. Acta N° 17 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	17
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	28/03/2016 – 09:00 am
FECHA FIN	28/03/2016 - 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	DEFINICIÓN SPRINT 7
OBJETIVOS	
Actualización de pedidos a recoger	
Selección de pedido y de los ítems específicos recogidos	
Actualización de estado de pedido e ítems recogidos tanto en el servidor como en el dispositivo.	
Listado de pedidos a facturar	

Selección de pedido a facturar y búsqueda de cliente	
Envío de datos del pedido facturado y actualización del estado del pedido en el dispositivo	
Registro de cierre de periodo y eliminación de data local	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 18. Acta N° 18 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	18
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	27/04/2016 – 09:00 am
FECHA FIN	27/04/2016 – 12:00 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	PRUEBAS SPRINT 7
OBJETIVOS	
Hacer pruebas de la entrega realizada por los desarrolladores	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO

ANEXO 19. Acta N° 19 de reunión de coordinación del proyecto

REUNIÓN CON CLIENTE	19
PROYECTO	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO MÓVIL PARA MEJORAR LA ATENCIÓN DEL CLIENTE EN RESTAURANTES
FECHA INICIO Y HORA	29/04/2016 – 05:00 pm
FECHA FIN	29/04/2016 – 07:30 pm
LUGAR	RESTAURANTE CASA DESCALZI
FASE DEL PROYECTO	CORRECCIONES SPRINT 7
OBJETIVOS	
Mostrar las correcciones realizadas y pasar al siguiente sprint	
PERSONA ENTREVISTADA	FIDEL ESPINOZA CASTRO