

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE ING. EN COMPUTACIÓN E
INFORMÁTICA



TESIS

“Sistema informático web de control de recorrido de la empresa “El Chasqui S.A.” Chiclayo -2017”

**Presentado para obtener el título profesional de ingeniero en
computación e informática**

Autores:

Br. Solis Vera Gabriel Leonardo

Br. Zamora Granados Joel Alfredo

Asesor:

Ing. Martín Manuel Leiva Castillo.

Lambayeque, 2020

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE ING. EN COMPUTACIÓN E
INFORMÁTICA



TESIS

“Sistema informático web de control de recorrido de la empresa “El Chasqui S.A.” Chiclayo -2017”

**Presentado para obtener el título profesional de ingeniero en
computación e informática**

Aprobado Por:

Dr. Ing. Moreno Heredia Armando Jose.

Presidente

Dra. Ing. Lecca Orrego Giuliana Fiorella.

Secretario

Mg. Ing. Valdivia Salazar Carlos Alberto.

Vocal

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE ING. EN COMPUTACIÓN E
INFORMÁTICA



TESIS

“Sistema informático web de control de recorrido de la empresa “El Chasqui S.A.” Chiclayo -2017”

**Presentado para obtener el título profesional de ingeniero en
computación e informática**

Elaborado Por:

Br. Solis Vera Gabriel Leonardo

Br. Zamora Granados Joel Alfredo

Asesorado Por:

Ing. Martín Manuel Leiva Castillo.

Dr. Ing.

Presidente: Dr. Ing. Moreno Herdia Armando Jose

Dra. Ing.

Secretario: Dra. Ing. Lecca Orrego Giuliana Fiorella

Mg. Ing.

Vocal: Mg. Ing. Valdivia Salazar Carlos Alberto

Ing.

Asesor: Ing. Martín Manuel Leiva Castillo



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DECANATO
 Ciudad Universitaria - Lambayeque



ACTA DE SUSTENTACIÓN N° 004-2020-D/FACFyM

(Sustentación Autorizada por Resolución N° 1714-2019-D/FACFyM)

En la ciudad de Lambayeque, siendo las 12:00 m. del día 14 de enero de 2020 se reunieron en la videoteca del laboratorio de Física de la FACFyM los miembros del Jurado designados mediante Resolución N° 020-2018-D/FACFyM, los docentes:

Dr. Ing. Armando José Moreno Heredia	Presidente
Dra. Ing. Giuliana Fiorella Lecca Orrego	Secretario
M.Sc. Ing. Carlos Alberto Valdivia Salazar	Vocal

Para recibir la tesis titulada:

Sistema informático web de control de recorridos de la Empresa "El Chasqui S.A.A" Chiclayo 2017

desarrollada por los Bachilleres en Computación e Informática, **Solis Vera Gabriel Leonardo y Zamora Granados Joel Alfredo**

Después de escuchar la exposición y las respuestas a las preguntas formuladas por los miembros del Jurado, se acordó aprobar el trabajo por unanimidad con el calificativo de Muy Bueno.

En consecuencia, los Bachilleres en referencia quedan aptos para recibir el Título Profesional de **Ingeniero en Computación e Informática**, de acuerdo a la Ley Universitaria, el Estatuto y Reglamento de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque.

Observaciones:

Para constancia del hecho firman.

 Dr. Ing. Armando José Moreno Heredia
 Presidente

 Dra. Ing. Giuliana Fiorella Lecca Orrego
 Secretario

 M.Sc. Ing. Carlos Alberto Valdivia Salazar
 Vocal

CER: ICO: Que, copia fiel del original
 FECHA: 10/08/20

 Dra. Iris Margarita Teresa Romero
 SECRETARÍA DOCENTE - FACFyM
 VÁLIDO PARA TRÁMITE INTERNOS DE LA UNPRG

Declaración Jurada de Originalidad

Nosotros, Solis Vera Gabriel Leonardo, Zamora Granados Joel Alfredo, y Ing. Martín Manuel Leiva Castillo declaramos bajo juramento que este trabajo no ha sido plagiado, ni contiene datos falsos. En caso se demostrar lo contrario, asumo responsablemente la anulación de este informe y por ende el proceso administrativo a que hubiera lugar.

Que puede conducir a la anulación del título o grado emitido como consecuencia de este informe.

Lambayeque, 1 de Octubre de 2020

Br. Solis Vera Gabriel Leonardo

Autor

Br. Zamora Granados Joel Alfredo

Autor

Ing. Martín Manuel Leiva Castillo.

Asesor

Dedicatoria

A mis padres Segundo Solis Bautista y
Sivla Vera Santoyo quienes con su amor,
paciencia y esfuerzo me han permitido
llegar a cumplir una meta más, gracias por
inculcar en mi la idea que, con esfuerzo,
estudio y Dios las metas siempre se hacen
realidad.

A mis hermanos Paul y Cristina por su
cariño y apoyo incondicional, durante todo
este proceso, por estar conmigo en todo
momento gracias. A toda mi familia por el
apoyo a lo largo de este largo camino
universitario.

Finalmente quiero dedicar esta tesis a
Brenda por su paciencia y amor en los
momentos alegres y difíciles que me
permitieron seguir adelante.

A mi Asesor por su tiempo y paciencia,
Profesores, Autoridades de la FACFYM y
quienes nos brindaron apoyo para poder
lograr y culminar este objetivo.

Br. Gabriel Leonardo Solis Vera

Dedico este trabajo a Dios por darme la vida, salud y fuerzas para permitirme llegar a este momento tan especial en mi vida. Por los logros y los obstáculos presentados los cuales me enseñan a seguir adelante.

A mis Padres Antonio Zamora Flores y Marleni del Pilar Granados Ramos, por ser siempre un apoyo incondicional en la parte moral y económica para llegar a ser un profesional, por apoyarme constantemente a lograr mis objetivos, y a su inmenso cariño que todos los días me brindan.

A mis hermanos, sobrinas, sobrinos porque me han brindado su apoyo incondicional y por compartir buenos y malos momentos.

A Arielita que me animo con su gran carácter de no dejarme vencer ante las adversidades y su motivación.

A mi Asesor por su tiempo y paciencia, Profesores, Autoridades de la FACFYM y quienes nos brindaron apoyo para poder lograr y culminar este objetivo.

Br. Joel Alfredo Zamora Granados

Agradecimiento

Agradecemos a Dios por habernos ayudado maravillosamente en cada paso de este proyecto, facilitando y abriendo camino a los obstáculos presentados y sobre todo por habernos inspirado día tras día.

A nuestro Asesor Ing. Martin Leiva Castillo por habernos ayudado a enfocarnos en la investigación, por su apoyo incondicional, experiencia compartida y consejos que nos brindó durante el desarrollo del presente trabajo de investigación.

A las Autoridades y Trabajadores de la Facultad de Ciencias Físicas y Matemáticas que nos brindaron su apoyo para agilizar y finalizar el presente trabajo de investigación.

Agradecer también a la señora Olivia quien nos dio todas las facilidades para el desarrollo completo del sistema.

A nuestras familias por brindarnos el apoyo, la confianza, el impulso, la perseverancia y la motivación para poder seguir adelante y lograr nuestros objetivos.

Índice General

Declaración Jurada de Originalidad.....	vi
Dedicatoria.....	vii
Agradecimiento.....	ix
Índice General.....	x
Índice de Tablas.....	xv
Índice de Figuras.....	xx
Resumen.....	xxvi
Abstract.....	xxvii
Introducción.....	1
Capítulo 1 Diseño Teórico.....	3
1.1 Realidad Problemática.....	3
1.1.1 Planteamiento del Problema.....	3
1.2 Antecedentes.....	4
1.2.1 Antecedentes en el Contexto Internacional.....	4
1.2.2 Antecedentes en el Contexto Nacional.....	5
1.3 Base Teórica.....	7
1.4 Variables.....	26
Variables dependientes.....	26
Variable independiente.....	28

1.5	Conceptos y Definiciones.....	28
Capitulo 2	Métodos y Materiales.....	31
2.1	Formulación del Problema	31
2.2	Objetivos de la Investigación	31
2.2.1	Objetivo General.....	31
2.2.2	Objetivos Específicos.	31
2.3	Justificación e Importancia de la Investigación	32
2.3.1	Justificación Social.	32
2.3.2	Justificación Económica.	32
2.3.3	Justificación Tecnológica.....	32
2.3.4	Justificación Operativa.	32
2.3.5	Importancia.	32
2.4	Hipótesis.....	33
2.5	Tipo de Investigación	33
2.6	Limitaciones de la Investigación.....	33
2.7	Análisis de Costos y Beneficios	33
2.7.1	Análisis de Costos.....	33
2.7.2	Análisis de Beneficios.	38
2.7.3	Flujo de Caja Incremental.....	41
2.8	Periodo de Recuperación de la Inversión.....	48
Capitulo 3	Resultados y Discusión.....	50
3.1	Sprint N° 0.....	50

3.1.1	Fase de entrevistas	50
3.1.2	Identificación de Stakeholders o Interesados.....	51
3.1.3	Elaborar Modelo de Procesos.	52
3.1.4	Requerimientos del Negocio.....	54
3.1.5	Requerimientos no funcionales.....	57
3.1.6	Definir usuarios del sistema.....	59
3.1.7	Product Backlog o Pila del Producto	60
3.1.8	Plan de versiones.	65
3.1.9	Organización del equipo de trabajo.	68
3.1.10	Entorno de trabajo.	68
3.2	Sprint N° 01.....	68
3.2.1	Planificación.	68
3.2.2	Tablero SCRUM.....	107
3.2.3	Diagrama de Burndown del Sprint N°01.	108
3.2.4	Retrospectiva.	108
3.3	Sprint N° 02.....	111
3.3.1	Planificación	111
3.3.2	Tablero SCRUM	134
3.3.3	Diagrama de Burndown N° 02.....	134
3.3.4	Retrospectiva	135
3.4	Sprint N°03.....	136
3.4.1	Planificación.	136

3.4.2	Tablero SCRUM.....	151
3.4.3	Diagrama de Burndown del Sprint N° 03	152
3.4.4	Retrospectiva	152
3.5	Sprint N°04.....	154
3.5.1	Planificación.	154
3.5.2	Tablero SCRUM.....	169
3.5.3	Diagrama de Burndown del Sprint N° 04.	170
3.5.4	Retrospectiva.	170
3.5.5	Planificación.	172
3.5.6	Tablero SCRUM.....	186
3.5.7	Diagrama de Burndown del Sprint N° 03	187
3.5.8	Retrospectiva	187
3.6	Sprint N°05.....	189
3.6.1	Planificación.	189
3.6.2	Tablero SCRUM.....	201
3.6.3	Diagrama de Burndown del Sprint N° 05	202
3.6.4	Retrospectiva	202
3.7	Diccionario de la base de datos	206
3.8	Creación de Funciones	226
3.9	Resultado de pruebas.....	244
Capitulo 4	Conclusiones	252
Capitulo 5	Recomendaciones	255

Capitulo 6	Bibliografía	256
Capitulo 7	Apendices.....	263
7.1	Apéndice A: Desarrollo de Historias de usuario	263
7.2	Apéndice B: Preguntas del Sprint 0	272
7.3	Apéndice C: Desarrollo de pruebas automatizadas.....	278
7.4	Apendice D: Entorno de trabajo.....	282

Índice de Tablas

Tabla 01 Costo de Materiales de Oficina para el Desarrollo del Proyecto	34
Tabla 02 Costo de Suministros de Cómputo y Otros para el Desarrollo del Proyecto ..	34
Tabla 03 Costo Total de Bienes para el Desarrollo del Proyecto	35
Tabla 04 Costo de Equipos para el Desarrollo del Proyecto	35
Tabla 05 Costo de Software para el Desarrollo del Proyecto	36
Tabla 06 Costo de Servicios para el Desarrollo del Proyecto.....	37
Tabla 07 Costo Total para el Desarrollo del Proyecto	37
Tabla 08 Costo del Material Desperdiciado en los Procesos sin Uso del Sistema Informático	39
Tabla 09 Costo de los Rubros de Ahorro Generados con Uso del Sistema Informático	40
Tabla 10 Desarrollo del Flujo de Caja Incremental	44
Tabla 11 Interpretación de los Resultados	47
Tabla 12 Flujo de Efectivo.....	48
Tabla 13 Lista de Stakeholders o Interesados.....	51
Tabla 14 Lista Inicial de requerimientos	55
Tabla 15 Lista de Requerimientos no Funcionales	57
Tabla 16 Gerente.....	59
Tabla 17 Secretaria	59
Tabla 18 Controlador	60
Tabla 19 Product Backlog.....	61
Tabla 20 Plan de versiones	66
Tabla 21 Sprint Backlog – Sprint N° 01	69
Tabla 22 HU: Registrar socio	70
Tabla 23 HU: Modificar socio	72

Tabla 24 HU: Dar de baja socio.....	75
Tabla 25 HU: Visualizar socio.....	77
Tabla 26 HU: Registro de unidades por socio	79
Tabla 27 HU: Modificar datos de unidad	81
Tabla 28 HU: Dar de baja unidad	84
Tabla 29 HU: Visualizar unidad	87
Tabla 30 HU: Registro de ruta.....	89
Tabla 31 HU: Modificar ruta	92
Tabla 32 HU: Dar de baja ruta.....	94
Tabla 33 HU: Visualizar ruta.....	96
Tabla 34 HU: Registro de puntos de control	98
Tabla 35 HU: Modificar punto de control	101
Tabla 36 HU: Activar punto de control	104
Tabla 37 HU: Listar puntos de control	106
Tabla 38 Cierre del Sprint N° 01	109
Tabla 39 Sprint backlog – Sprint N° 02.....	111
Tabla 40 HU: Dar de baja puntos de control de ruta	112
Tabla 41 HU: Registro de tiempos establecidos.....	114
Tabla 42 HU: Modificar tiempos establecidos	116
Tabla 43 HU: Listar tiempos	118
Tabla 44 HU: Crear usuarios	120
Tabla 45 HU: Modificar contraseña de usuarios	122
Tabla 46 HU: Dar de baja usuarios.....	124
Tabla 47 HU: Registrar personal	126
Tabla 48 HU: Modificar personal.....	128

Tabla 49 HU: Dar de baja personal	130
Tabla 50 HU: Visualizar personal	132
Tabla 51 Cierre de Sprint N°02	135
Tabla 52 Sprint Backlog – Sprint N°03	137
Tabla 53 HU: Justificación de no disponibilidad de la unidad	138
Tabla 54 HU: Asignación de tarjetas a unidad	140
Tabla 55 HU: Dar de baja a la tarjeta	142
Tabla 56 HU: Visualizar tarjeta	144
Tabla 57 HU: Asignar recorrido a unidad	147
Tabla 58 HU: Dar de baja recorrido asignado a unidad	149
Tabla 59 Cierre de Sprint N°02	153
Tabla 60 Sprint Backlog – Sprint N° 04	154
Tabla 61 HU: Registrar sanciones	155
Tabla 62 HU: Aplicar sanciones a unidad	157
Tabla 63 HU: Registrar justificación por unidad	158
Tabla 64 HU: Visualizar de recorrido por unidad	160
Tabla 65 HU: Reporte de unidades trabajando por fecha	162
Tabla 66 HU: Listar sanciones por unidad	164
Tabla 67 HU: Reporte de retrasos de tiempo.....	166
Tabla 68 HU: Reporte de recorrido diario	168
Tabla 69 Cierre de Sprint N°03	171
Tabla 70 Sprint Backlog – Sprint N°03	172
Tabla 71 HU: Justificación de no disponibilidad de la unidad	173
Tabla 72 HU: Asignación de tarjetas a unidad	175
Tabla 73 HU: Dar de baja a la tarjeta	177

Tabla 74 HU: Visualizar tarjeta	179
Tabla 75 HU: Asignar recorrido a unidad	182
Tabla 76 HU: Dar de baja recorrido asignado a unidad	184
Tabla 77 Cierre de Sprint N°02	188
Tabla 78 Sprint Backlog - Sprint N°05.....	189
Tabla 79 HU: Registro de GPS.....	190
Tabla 80 HU: Pago de unidad.....	192
Tabla 81 HU: Comparación de recorridos	194
Tabla 82 HU: Reporte de promedio de recorrido de rutas.....	196
Tabla 83 HU: Listado de GPS	198
Tabla 84 HU: Autenticación de usuarios	200
Tabla 85 Cierre de Sprint N°05	202
Tabla 86 Base de datos – Tabla calleruta	206
Tabla 87 Base de datos – Tabla detalletarjeta.....	207
Tabla 88 Base de datos – Tabla GPS.....	208
Tabla 89 Base de datos – Tabla GPSunidad.....	209
Tabla 90 Base de datos – Tabla persona.....	210
Tabla 91 Base de datos – Tabla personal.....	211
Tabla 92 Base de datos – Tabla puntocontrol.....	212
Tabla 93 Base de datos – Tabla registrounidad	213
Tabla 94 Base de datos – Tabla ruta	215
Tabla 95 Base de datos – Tabla sanción	216
Tabla 96 Base de datos – Tabla socio.....	217
Tabla 97 Base de datos – Tabla tarjeta	219
Tabla 98 Base de datos – Tabla tiempo detalletarjeta	220

Tabla 99 Base de datos – Tabla tiempoestablecido	222
Tabla 100 Base de datos – Tabla unidad	223
Tabla 101 Base de datos – Tabla usuario	225
Tabla 102 Base de datos – Creación de funciones.....	226
Tabla 103 Escenario N° 2 – Edad de socio.....	247
Tabla 104 Escenario N° 2 – Registrar punto de control	248
Tabla 105 Escenario N° 1 – Motivo en blanco.....	249
Tabla 106 Escenario N° 1 – Listar por tipo	250

Índice de Figuras

Figura 1 Prototipo del Sprint 01 – HU: Registrar Socio.....	71
Figura 2 Prototipo del Sprint 01 – HU: Modificar Socio	73
Figura 3 Prototipo del Sprint 01 – HU: Modificar Socio	74
Figura 4 Prototipo del Sprint 01 – HU: Dar de baja Socio.....	76
Figura 5 Prototipo del Sprint 01 – HU: Visualizar Socio.....	78
Figura 6 Prototipo del Sprint 01 – HU: Visualizar Socio.....	78
Figura 7 Prototipo del Sprint 01 – HU: Registro de unidades por socio	80
Figura 8 Prototipo del Sprint 01 – HU: Modificar datos de unidad	82
Figura 9 Prototipo del Sprint 01 – HU: Modificar datos de unidad	83
Figura 10 Prototipo del Sprint 01 – HU: Modificar datos de unidad	83
Figura 11 Prototipo del Sprint 01 – HU: Dar de baja unidad	85
Figura 12 Prototipo del Sprint 01 – HU: Dar de baja unidad	86
Figura 13 Prototipo del Sprint 01 – HU: Visualizar unidad	88
Figura 14 Prototipo del Sprint 01 – HU: Visualizar unidad	88
Figura 15 Prototipo del Sprint 01 – HU: Visualizar unidad	89
Figura 16 Prototipo del Sprint 01 – HU: Registro de Ruta.....	91
Figura 17 Prototipo del Sprint 01 – HU: Registro de Ruta.....	91
Figura 18 Prototipo del Sprint 01 – HU: Modificar Ruta.....	93
Figura 19 Prototipo del Sprint 01 – HU: Modificar Ruta.....	93
Figura 20 Prototipo del Sprint 01 – HU: Dar de baja ruta.....	95
Figura 21 Prototipo del Sprint 01 – HU: Visualizar ruta.....	97
Figura 22 Prototipo del Sprint 01 – HU: Visualizar ruta.....	97
Figura 23 Prototipo del Sprint 01 – HU: Registro de punto de control	99
Figura 24 Prototipo del Sprint 01 – HU: Registro de punto de control	100

Figura 25 Prototipo del Sprint 01 – HU: Registro de punto de control	100
Figura 26 Prototipo del Sprint 01 – HU: Modificar punto de control	102
Figura 27 Prototipo del Sprint 01 – HU: Modificar punto de control	103
Figura 28 Prototipo del Sprint 01 – HU: Modificar punto de control	103
Figura 29 Prototipo del Sprint 01 – HU: Activar punto de control	105
Figura 30 Prototipo del Sprint 01 – HU: Listar puntos de control	107
Figura 31 Tablero SCRUM del Sprint 01	108
Figura 32 Diagrama de Burndown del Sprint 01	108
Figura 33 Prototipo del Sprint 02 – HU: Dar de baja puntos de control de ruta	113
Figura 34 Prototipo del Sprint 02 – HU: Registro de tiempos establecidos	115
Figura 35 Prototipo del Sprint 02 – HU: Registro de tiempos establecidos	115
Figura 36 Prototipo del Sprint 02 – HU: Modificar tiempos establecidos	117
Figura 37 Prototipo del Sprint 02 – HU: Modificar tiempos establecidos	117
Figura 38 Prototipo del Sprint 02 – HU: Listar Tiempos	119
Figura 39 Prototipo del Sprint 02 – HU: Crear usuarios	121
Figura 40 Prototipo del Sprint 02 – HU: Crear usuarios	121
Figura 41 Prototipo del Sprint 02 – HU: Modificar contraseña de usuario	123
Figura 42 Prototipo del Sprint 02 – HU: Dar de baja usuario	125
Figura 43 Prototipo del Sprint 02 – HU: Registrar personal	127
Figura 44 Prototipo del Sprint 02 – HU: Modificar personal	129
Figura 45 Prototipo del Sprint 02 – HU: Dar de baja personal.....	131
Figura 46 Prototipo del Sprint 02 – HU: Visualizar personal.....	133
Figura 47 Prototipo del Sprint 02 – HU: Visualizar personal.....	133
Figura 48 Tablero SCRUM del Sprint 02.....	134
Figura 49 Diagrama de Burndown del Sprint 02.....	134

Figura 50 Prototipo del Sprint 03 – HU: Justificación de no disponibilidad de unidad	139
Figura 51 Prototipo del Sprint 03 – HU: Asignación de tarjetas a unidad	141
Figura 52 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta	143
Figura 53 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta	144
Figura 54 Prototipo del Sprint 03 – HU: Visualizar tarjeta	145
Figura 55 Prototipo del Sprint 03 – HU: Visualizar tarjeta	146
Figura 56 Prototipo del Sprint 03 – HU: Visualizar tarjeta	146
Figura 57 Prototipo del Sprint 03 – HU: Asignar recorrido a unidad.....	148
Figura 58 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	150
Figura 59 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	150
Figura 60 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	151
Figura 61 Tablero SCRUM del Sprint 03.....	151
Figura 62 Diagrama de Burndown del Sprint 03	152
Figura 63 Prototipo del Sprint 04 – HU: Registrar sanción.....	156
Figura 64 Prototipo del Sprint 04 – HU: Aplicar sanciones a unidad	158
Figura 65 Prototipo del Sprint 04 – HU: Registrar justificación por unidad	159
Figura 66 Prototipo del Sprint 04 – HU: Visualizar recorrido por unidad	161
Figura 67 Prototipo del Sprint 04 – HU: Reporte de unidades trabajando por fecha...	163
Figura 68 Prototipo del Sprint 04 – HU: Listar sanciones por unidad	165
Figura 69 Prototipo del Sprint 04 – HU: Reporte de retrasos de tiempo.....	167
Figura 70 Prototipo del Sprint 04 – HU: Reporte de recorrido diario	169
Figura 71 Tablero SCRUM del Sprint 04.....	169
Figura 72 Diagrama de Burndown del Sprint 04.....	170
Figura 73 Prototipo del Sprint 03 – HU: Justificación de no disponibilidad de unidad	174
Figura 74 Prototipo del Sprint 03 – HU: Asignación de tarjetas a unidad	176

Figura 75 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta	178
Figura 76 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta	179
Figura 77 Prototipo del Sprint 03 – HU: Visualizar tarjeta	180
Figura 78 Prototipo del Sprint 03 – HU: Visualizar tarjeta	181
Figura 79 Prototipo del Sprint 03 – HU: Visualizar tarjeta	181
Figura 80 Prototipo del Sprint 03 – HU: Asignar recorrido a unidad.....	183
Figura 81 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	185
Figura 82 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	185
Figura 83 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad.....	186
Figura 84 Tablero SCRUM del Sprint 03	186
Figura 85 Diagrama de Burndown del Sprint 03	187
Figura 86 Prototipo del Sprint 05 – HU: Registro de GPS	191
Figura 87 Prototipo del Sprint 05 – HU: Pago de unidad.....	193
Figura 88 Prototipo del Sprint 05 – HU: Pago de unidad.....	193
Figura 89 Prototipo del Sprint 05 – HU: Pago de unidad.....	193
Figura 90 Prototipo del Sprint 05 – HU: Comparación de recorridos	195
Figura 91 Prototipo del Sprint 05 – HU: Reporte de promedio de recorrido de rutas..	197
Figura 92 Prototipo del Sprint 05 – HU: Listado de GPS	199
Figura 93 Prototipo del Sprint 05 – HU: Autenticación de usuarios.....	201
Figura 94 Tablero SCRUM del Sprint 05	201
Figura 95 Diagrama de Burndown del Sprint 05	202
Figura 96 Diagrama de Base de Datos.....	204
Figura 97 Diagrama de Componentes	205
Figura 98 Test verificar placa – UnidadTest	244
Figura 99 Test obtener unidad – UnidadTest.....	244

Figura 100 Test listar unidades activas – UnidadTest	245
Figura 101 UnidadTest	245
Figura 102 Test verificar tarjeta– TarjetaTest	245
Figura 103 Test listar tarjetas por estado- TarjetaTest	245
Figura 104 Test listar tarjetas– TarjetaTest	246
Figura 105 TarjetaTest.....	246
Figura 106 Diseñar la interfaz – HU 001 – Registrar Socio –Apendice A.....	263
Figura 107 Diseñar y crear tabla de BD– HU 001 – Registrar Socio – Apendice A..	264
Figura 108 Construir Interfaz 1– HU 001– Registrar Socio – Apendice A.....	265
Figura 109 Construir Interfaz 2– HU 001 – Registrar Socio – Apendice A.....	265
Figura 110 Crear PA en la Base de Datos– HU 001– Registrar Socio – Apendice A.	266
Figura 111 Crear Código Parte 1 -HU 001 -Registrar Socio - Apendice A	266
Figura 112 Crear Código Parte 2 -HU 001 -Registrar Socio - Apendice A	266
Figura 113 Diseñar la interfaz -HU 043 –Registro de GPS - Apendice A	267
Figura 114 Cuonstruir la interfaz -HU 043–Registro de GPS-Apendice A	267
Figura 115 Interfaz Tracar -HU 043 –Registro de GPS - Apendice A.....	268
Figura 116 Crear Tablas y Procedimiento Almacenado en la BD -HU 043 –Registro de GPS - Apendice A.....	268
Figura 117 Crear código correspondiente -HU 043 –Registro de GPS - Apendice A .	269
Figura 118 Diseñar la interfaz -HU 035 –Registrar Sanciones-Apendice A.....	270
Figura 119 Crear Tablas y Procedimiento Almacenado en la BD -HU 035 –Registrar Sanciones - Apendice A.....	271
Figura 120 Construir interfaz -HU 035 –Registrar Sanciones - Apendice A.....	271
Figura 121 Crear código correspondiente -HU 035 –Registrar Sanciones - Apendice A	272

Figura 122 Criterio de aceptación –Prueba automatizada - Registro de socio – Apendice C.....	278
Figura 123 Clases de prueba – Prueba automatizada – Registro de socio – Apendice C	279
Figura 124 Resultado –Prueba automatizada – Registro de socio – Apendice C... 280	
Figura 125 Criterio de aceptación –Prueba automatizada – Registro de punto de control – Apendice C	280
Figura 126 Clase de prueba –Prueba automatizada – Registro de punto de control– Apendice C	281
Figura 127 Resultado –Prueba automatizada – Registro de punto de control– Apendice C.....	281
Figura 128 Configuracion de variables de entorno JAVA_HOME – JDK 8– Entorno de trabajo	282
Figura 129 Configuracion de variables de entorno MAVEN_HOME – Entorno de trabajo	283
Figura 130 GPS TRACKER – Entorno de trabajo	283
Figura 131 Spring Tool Suite (STS) – Entorno de trabajo	283
Figura 132 Postgresql (PGAdmin4) – Entorno de trabajo.....	284
Figura 133 Configuracion de Tomcat en Digital Ocean – Entorno de trabajo	285
Figura 134 Dashboard de Heroku – Entorno de trabajo	302
Figura 135 Dashboard de Digital Oceab – Entorno de trabajo.....	302

Resumen

El presente proyecto consiste en desarrollar un sistema informático web de control de recorridos que permita regularizar las rutas y tiempos de las unidades de transporte público de la empresa “El Chasqui S.A”.

La problemática se enfoca con mayor proporción en el proceso de registro y control de rutas y tarjetas de recorrido, donde vemos una serie de inconvenientes, entre ellos tenemos, el malestar entre el personal ya que muchas veces no se respeta los tiempos asignados de cada vuelta por unidad debido a que este proceso se realiza en forma manual permitiendo además saboteos en las tarjetas por parte de los choferes de las unidades afectando el servicio que busca brindar la empresa, así como también demora e inconvenientes en el registro y aplicación de sanciones por retrasos o cambios de ruta para las unidades que vienen desempeñando su labor diariamente. En el desarrollo de este proyecto se aplicó la metodología ágil XP, utilizando el marco de trabajo SCRUM, apoyándonos en este enfoque ágil se permitió obtener los requerimientos de forma más clara y precisa, debido a las reuniones con el cliente; también se realizaron pruebas automatizadas, mejorando la calidad del sistema a lo largo del desarrollo de cada sprint. Con la realización de este sistema informático web se normalizará las rutas y el control de tiempos en las unidades así como un almacenamiento de datos seguros y rápida consulta de reportes, así como el ahorro de tiempo en la ejecución de las actividades y la obtención de información confiable y oportuna ayudando así a mejorar el ambiente laboral y control de rutas y tiempos de las unidades de la empresa permitiendo con esto brindar una mejor calidad de servicio al personal y pasajeros.

Palabras clave: metodología ágil XP, marco de trabajo SCRUM, tarjetas de recorrido, unidades.

Abstract

The present project consists in developing a computerized system to control the routes that allow to regularize the routes and public transport times of the company "El Chasqui S.A".

The present project consists of developing a computerized system of engraving control that allows to regulate the routes and times of the public transport units of the company "El Chasqui S.A".

The problem focuses with greater proportion on the process of registration and control of routes and travel cards, where we see a series of inconveniences, among them we have, the male among the staff since many times the assigned times of each lap are not respected per unit because this process is carried out manually, in addition to sabotages on the cards by the drivers of the units that affect the service that the company seeks to provide, as well as delay and inconvenience in the registration and application of delays for delays or changes of route for the units that have been performing their daily work. In the development of this project the agile XP methodology was applied, using the SCRUM framework, based on this agile approach, the requirements will be obtained more clearly and precisely, due to the meetings with the client; Automated tests were also performed, improving the quality of the system throughout the development of each sprint. With the realization of this web computer system, the routes and the control of time in the units will be standardized, as well as a secure and fast data storage to consult reports, as well as saving time in the execution of activities and obtaining information Reliable and timely helping to improve the work environment and the control of routes and times of the units of the company affected with this provide a better quality of service to staff and passengers.

Keywords: agile XP methodology, SCRUM framework, travel cards, units.

Introducción

Este proyecto de tesis tiene como finalidad presentar una solución web dirigida a la problemática actual en la gestión de información y regularización del proceso de rutas y recorridos de la empresa “EL CHASQUI S.A”.

En la empresa a diario se registran manualmente tarjetas en las cuales se colocan información de rutas, tiempos y sanciones pertenecientes a cada unidad laborando, posteriormente la información de cada tarjeta se pasa a una plantilla la cual sirve para recaudar la información pertinente para la junta de accionistas, en todo este trayecto intervienen los procesos de asignar tarjeta, proceso de asignar ruta, proceso de asignar sanción, proceso de registrar socio y el proceso de registro de tiempos, los cuales al ser realizados de forma manual y encontrándose la información de las tarjetas y planillas expuesta a los trabajadores presentan lentitud en el procesamiento de información y pérdida o alteración de la misma. Además, el control de rutas y tiempos presenta deficiencias debido a que los relojes controladores que se usan en el proceso son fácilmente manipulables por los trabajadores generando información imprecisa, gastos en reparaciones y afectando muchas veces directamente el servicio brindado por la empresa.

El objetivo principal es implementar un sistema informático web de control de recorridos que permita regularizar las rutas y tiempos de las unidades de transporte público de la empresa “El Chasqui S.A” minimizando considerablemente los problemas que ocurren actualmente

Con el cumplimiento de este objetivo general, el proyecto ya realizado provocará un gran impacto positivo en la empresa, de tal manera que se mejorará la calidad del servicio de transporte público brindado por la empresa el Chasqui S.A además de reducir el esfuerzo en las actividades diarias que realizan los trabajadores administrativos, también permitirá un

mejor control de tiempos y recorridos, generados por las unidades, minimizando costos y tiempo al momento de realizar reportes.

En el capítulo 1 se enumeran diversos trabajos desarrollados a nivel nacional e internacional por otros autores que permitirán dar una noción de como abordar la problemática existente en la empresa “El Chasqui S.A”, además se describen los diversos términos propios del negocio que son claves para el desarrollo de la investigación.

En el capítulo 2 se describen los objetivos del proyecto, importancia y justificación del desarrollo de la investigación, así como también la metodología a usar y el costo total de la solución.

En el capítulo 3 se detalla el desarrollo de la solución pasando por las etapas de configuración de las herramientas a utilizar, desarrollos de sprints, resultado de pruebas e implementación de la solución.

En el capítulo 4 y 5 se detallan las conclusiones a las que se llegaron como resultado de la investigación, así como también las recomendaciones para futuras investigaciones.

En el capítulo 6 mencionamos la documentación, libros, lecturas y demás información que nos fue útil para el desarrollo exitoso del proyecto.

En el capítulo 7 se incluyen 3 apéndices en donde se agrupan diversos procesos y herramientas utilizados para el desarrollo del proyecto. El apéndice A muestra el desarrollo detallado de diversas historias de usuario. EL apéndice B contiene las preguntas realizadas al personal de la empresa “El Chasqui S.A.” para obtener información clave para el desarrollo de la investigación. Por último, el Apendic C contiene el desarrollo detallada de algunas pruebas automatizadas necesarias para la correcta validación del desarrollo e implementación de la solución.

Capítulo 1 Diseño Teórico

1.1 Realidad Problemática

1.1.1 Planteamiento del Problema.

La empresa “EL CHASQUI S.A” que inicio sus servicios el 1 de abril del 1997 ubicada en el distrito de La Victoria, provincia de Chiclayo, región Lambayeque dedicada principalmente a ofrecer el servicio de transporte urbano, tiene como objetivo principal posicionarse entre una de las empresas líderes en su rubro y mejorar su servicio para bienestar y comodidad de sus clientes.

Actualmente la empresa cuenta con 100 unidades de combis para el servicio de transporte público, controladas diariamente por dos personas las cuales registran en tarjetas: las rutas de cada unidad, el tiempo de salida y llegada por cada vuelta y los retrasos de las unidades junto a sus justificaciones, al final del día cada controlador entrega las tarjetas a la oficina central siendo la secretaria la encargada de procesar y archivar las tarjetas en planillas, este proceso ha sido aprovechado por los choferes de las unidades para poder sabotear estas tarjetas, evitando la ruta e infracciones establecidas ya que carecen de algún mecanismo de control para el correcto recorrido de rutas, generando malestar entre el personal ya que muchas veces no se respeta los tiempos asignados de cada vuelta por unidad y afectando así el servicio que busca brindar la empresa.

Para el mantenimiento y crecimiento de la empresa los 80 socios que la conforman realizan una aportación de 10 soles por cada día laborado y por cada unidad que poseen, estas aportaciones diarias son registradas y archivadas por la secretaria en carpetas causando un retraso de una hora y media aproximadamente y por otro lado un retraso de

una semana en promedio para generar reportes como: informes de aportación de cada socio, balances de ingresos por aportaciones de la empresa o informes de aportaciones a la junta de socios.

Por ello, debido a los problemas ya mencionados que amenazan a la empresa se desarrollará un sistema informático web que permita llevar un correcto control de tiempos en las unidades, supervisión de recorrido de rutas en tiempo real, así como un almacenamiento de datos seguro y rápida consulta de reportes.

1.2 Antecedentes

1.2.1 Antecedentes en el Contexto Internacional.

Antecedente 01.

Mendoza Orozco, X. A. (2013), En su proyecto de tesina en la Universidad Carlos III de Madrid, Madrid, “*Mejoramiento del servicio de transporte urbano colectivo en la pista Juan Pablo II, Managua-Nicaragua*”, presenta como principal objetivo el mejorar la calidad del servicio de transporte urbano colectivo a lo largo de la Pista Juan Pablo II en Managua. El autor finaliza su investigación concluyendo que se acondicionara un nuevo carril exclusivo para el transporte urbano colectivo con el fin de satisfacer la demanda total de servicios a través de la construcción de infraestructura adecuada, así como implementara en cada parada mapas con información relativa a los horarios de los buses, origen y destino de su recorrido; pantallas que informen sobre los tiempos de espera de cada autobús, horarios de llegada y destino de cada línea, así como información sobre las diferentes líneas de trasbordo hacia otros lugares, además se promovió el transporte público sostenible y seguro brindando a la población un sistema de transporte seguro, eficiente y funcional que atraiga no solo a la población de bajos recursos económicos por ser un sistema accesible sino también a otros grupos sociales al comprobar los beneficios en calidad y tiempo que este sistema presenta.

Antecedente 02.

Vega Bueno, M. A. (2014) En su tesis para optar al grado de magíster en gestión y dirección de empresas en la Universidad de Chile, Santiago de Chile, *Gestión de flota para una empresa distribuidora de pizzas*, presenta como principal objetivo el gestionar la flota de transporte de una empresa distribuidora de pizzas de modo tal de optimizar el proceso de distribución a tiendas y locales. El autor finaliza su investigación concluyendo que el nivel de servicio a tiendas aumentará por medio de entregas de pedidos a tiempo y entrega de información en tiempo real de los pedidos, recomendando el uso de un sistema GPS para el control de rutas debido a su menor costo y calidad de información en comparación con otros métodos de control.

1.2.2 Antecedentes en el Contexto Nacional.

Antecedente 03.

Bocanegra Ureta, R. G. (2012). En sus tesis para optar por el título profesional de ingeniero informático en la Universidad Ricardo Palma, Lima, *Desarrollo de una aplicación Web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de telecomunicaciones*, plantea como principal objetivo desarrollar una aplicación web que realice el monitoreo de vehículos con dispositivos GPS y agilice la implantación del servicio a los clientes. El autor finaliza su investigación concluyendo que se implementó una solución web de monitoreo vehicular que da servicio a múltiples clientes sin necesidad de instancias personales, solo con el uso de un navegador web, los usuarios pueden gozar de las más representativas funcionalidades que una solución de monitoreo puede brindar de manera inmediata. Además, compara y muestra las funcionalidades más representativas que tienen las aplicaciones de monitoreo vehicular.

Antecedente 04.

Borjas Giraldo, G. (2013). En su tesis para optar por el título de Ingeniero informático en la Universidad Católica del Perú, Lima, *Análisis, diseño e implementación de un sistema de información para la administración de horarios y rutas en empresas de transporte público*, plantea como principal objetivo analizar, diseñar e implementar un sistema de información en un entorno web, que permita la administración de rutas y la generación manual o automatizada de los horarios del sistema de transporte público por medio de un algoritmo meta heurístico GRASP Construcción. El autor finaliza su investigación concluyendo que el sistema de información propuesto para implementarse en el sistema de transporte de Lima Metropolitana no supone solucionar los problemas que se presentan en la actualidad a nivel organizacional. Además con respecto a los tiempos de demora establecidos por tramo según el día y el rango de horas, es necesario que exista el compromiso por parte de las líneas de transporte en cumplir tales, ya que si bien, los tiempos de demora son aproximaciones, cada tiempo de demora real no debería diferir demasiado. Con el presente sistema de información web solo se plantea administrar óptimamente los datos de las rutas y de los horarios, con el fin de poder unificarla y utilizarla para mejorar el servicio del sistema de transporte en Lima Metropolitana, pero no busca implementar infraestructura, o un sistema integral de transporte tal como es el caso del Metropolitano. El fin de administrar información, es permitir facilitar la organización, tanto para los administradores como para los usuarios.

Antecedente 05.

Guanilo Barreto, R. (2013) En su tesis desarrollada en la Universidad Señor de Sipán, Chiclayo *Implementación de un sistema web y aplicación móvil bajo plataforma Android utilizando tecnología GPS para el control y monitoreo de las unidades de la empresa transporta S.A.C. de la ciudad de Chiclayo*, presenta como principal objetivo implantar un sistema web y aplicación móvil bajo plataforma Android utilizando tecnología GPS para el

control y monitoreo de las unidades de la empresa transporta S.A.C. de la ciudad de Chiclayo. El autor finaliza su investigación realzando que el desarrollo de la interfaz web, permitió una mejor administración del servicio y atención rápida a las consultas realizadas a través del celular, reflejadas en el sistema web mediante la gestión adecuada de los procesos siendo estos visualizados en reportes.

1.3 Base Teórica

Sistema de Transporte Público Urbano

El transporte público es el término aplicado al transporte colectivo de pasajeros. A diferencia del transporte privado, los viajeros de transporte público tienen que adaptarse a los horarios y a las rutas que ofrezca el operador, puede ser proporcionado por una o varias empresas privadas o consorcios de transporte público. Los servicios se mantienen mediante cobro directo de pasajeros y normalmente son servicios regulados y subvencionados por autoridades locales o nacionales, aunque en algunas ciudades existen servicios completamente subvencionados, cuyo costo para el viajero es gratuito.

Según el estudio realizado en el año 2014, por la consultora internacional Arthur D. Little y la Unión Internacional de Transporte Público (UITP) (citado en BBC, 2017) que incluyó a 9 ciudades de latinoamérica, de las 84 ciudades participantes, latinoamérica se encuentra en la mitad de la tabla resaltando ciudades como Santiago, Bogotá y Sao Paulo.

En el estudio se tomaron en cuenta factores como la integración entre los distintos modos de transporte, el planeamiento de la red de transporte, el cumplimiento y la puntualidad del servicio, la posibilidad para los usuarios de contar con información sobre el sistema en términos de congestionamiento, horarios, tiempos, espera, mapas, etc. En el estudio Lima ocupó el puesto 44.

En nuestro país el sistema de transporte público urbano más grande y congestionado se encuentra en Lima, la ciudad con mayor tráfico y accidentes vehiculares en todo el Perú,

una de las causas es el exceso de vehículos ya que según la Asociación Automotriz del Perú (citado en Perú21, 2017) en Lima existen el doble de vehículos de lo que requiere la demanda real de pasajeros, solo en taxis actualmente hay más de 200 mil cuando solo con 70 mil se podría cubrir la demanda. Asimismo, solo se necesitan 15 mil buses de transporte público pero circulan más de 36 mil unidades entre buses, cústers, y combis distribuidos en más de 308 empresas de transporte que recorren más de 652 rutas. Para resolver estos problemas Lima ha implementado sistemas de transporte alternos como el tren eléctrico de Lima, corredor azul, el Metropolitano y otras obras aún en gestión, pero aún falta mucho por hacer para solucionar estos y otros problemas que aquejan al sistema de transporte público de nuestra capital.

La ciudad de Chiclayo vive una situación similar ya que según datos de Coordinadora Regional de Transporte Público en Lambayeque (Cortpul) (citado en Chirinos, 2015), en Chiclayo circulan 7 mil unidades de combi, que cubren las diferentes rutas, tanto en el sector urbano, interurbano e interprovincial. Se suma a ello las unidades del servicio de colectivos, que llegan a los 3 mil 500; lo que suma en su totalidad las 10 mil 500 unidades que cubren el transporte público, esta cantidad de vehículos generan congestión en las diferentes rutas de la ciudad debido también a que la ciudad no cuenta con un plan regulador de rutas el cual se viene gestionando desde el año 2015 pero aún no ha sido aceptado ni implementado por la municipalidad.

Sistema de Posicionamiento Global (G.P.S)

El sistema de posicionamiento global, G.P.S, es un sistema de navegación desarrollado por el departamento de los Estados Unidos. Actualmente este sistema de 24 satélites artificiales y sus respectivas estaciones en tierra proporcionan información para el posicionamiento las 24 horas del día sin importar las condiciones del tiempo.

El sistema de posicionamiento global, G.P.S., se basa en la medición de distancias a partir de señales de radio transmitidas por un grupo de satélites artificiales cuya órbita se conoce con precisión y captadas y decodificadas por receptores ubicados en los puntos cuya posición se desea determinar.

Debido al constante desarrollo del sistema G.P.S., día a día se incrementa la aplicación de los mismos en las actividades científicas, profesionales, deportivas, recreacionales, etc.

Sería muy largo describir las actividades en las cuales el G.P.S. es una herramienta fundamental, Huerta, Mangiaterra y Noguera (2005) mencionan algunas de las áreas dentro del campo de la ingeniería en las cuales el G.P.S. está siendo utilizado:

Los geólogos, geógrafos e ingenieros forestales utilizan los G.P.S., en combinación con los sistemas de información geográfica (SIG) para la elaboración de mapas temáticos, captando en forma rápida y precisa la posición de puntos y asociando información y atributos a dichos puntos.

En cuanto a la planificación del transporte urbano, constituye un método rápido de levantamiento de la red de transporte, ya que recorriendo las calles y avenidas del sistema con un receptor G.P.S. se puede elaborar automáticamente el plano de la red.

En el mantenimiento vial, (drenajes, pavimentos, puentes, etc.), se pueden ubicar los puntos o sectores de la vía que necesitan mantenimiento, asociándole como atributo el tipo de estructura y mantenimiento a realizar.

En el campo de la topografía, se están aplicando los G.P.S. en el levantamiento de grandes extensiones y de zonas de difícil acceso ya que se requiere menos personal, debido a lo complicado del traslado de equipos tradicionales y a que la captura de datos con el sistema G.P.S. acelera o acorta el tiempo de recolección de los mismos.

Extreme Programming (XP):

Según Borja López (s.f) XP es una metodología ágil para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo, además promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo.

Borja López (s.f) indica que si bien el ciclo de vida de un proyecto XP es muy dinámico, se puede separar en las siguientes fases o etapas:

Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. El resultado de esta fase es un Plan de Entregas.

Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración

se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son:

Historias de usuario no abordadas

Velocidad del proyecto

Pruebas de aceptación no superadas en la iteración anterior

Tareas no terminadas en la iteración anterior.

Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.

Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Muerte del Proyecto.

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Scrum

Radigan (s.f) afirma que Scrum es uno de los marcos de trabajo para implementar una metodología ágil. Con Scrum, se crean una serie de iteraciones de longitud fija (dos a cuatro semanas) llamadas sprints que aportan a los equipos un marco para lanzar software con regularidad, lo que conlleva una sensación de progreso tangible en cada ciclo que centra a los trabajadores y les da energía.

Un equipo Scrum consta de tres cargos específicos: el propietario del producto cuya funcionalidad se basa en entender los requisitos empresariales y del mercado, para luego priorizar el trabajo que el equipo de ingeniería debe realizar para cumplirlos, el experto en Scrum que se basa en la funcionalidad de orientar al equipo, al propietario de producto y a la empresa sobre el proceso de Scrum y buscan cómo refinar su puesta en práctica y por último el equipo de desarrollo que tiene como funcionalidad el desarrollo sostenible ayudándose entre sí para asegurar una finalización satisfactoria del sprint.

Por otro lado, Scrum propone cuatro protocolos que articulan cada sprint:

Planificación de sprint

Reunión rápida diaria

Demostración del sprint

Retrospectiva del sprint

Generando como beneficios de seguir este marco de trabajo el cumplimiento de expectativas, la flexibilidad a cambios y generar una mayor calidad de software entre otros.

Combinando Scrum y XP

Según Lasa Gómez, C., Álvarez Garcia, A., & De las Heras del Dedo (2017) nos mencionan que Scrum y XP son dos métodos basados en valores y principios Agile y coinciden en su filosofía de trabajo. Ambos métodos comparten también numerosas practicas que un equipo de trabajo de desarrollo deberían aplicar a su trabajo diario para mejorar no solo desde el punto de vista de la productividad, si no también desde un punto de vista mas personal.

Se puede afirmar, sin correr demasiados riesgos a equivocarse, que scrum y xp pueden convivir en armonía en el caso en el que el proyecto que se este llevando a cabo contenga un desarrollo de software.

Algunos de los aspectos que comparten Scrum y XP y que hacen que su convivencia sea posible en perfecta armonía son los siguientes:

- **Revision del trabajo:** Ambos métodos se basan en revisar de forma periódica y frecuente el trabajo realizado. Estar dispuestos constantemente a ver que se puede cambiar y mejorar en todos los niveles del trabajo y llevarlos a la practica. Ejemplo de ello son la programación por parejas, la automaizacion de pruebas, la retrospectivas y la revisión del sprint.

- **Colaboración:** Una buena relación entre las personas que estran trabajando con un objetivo común lleva consigo buenos productos.
- **Implicacion directa del cliente.** La colaboración constante del cliente es clave para asegurar en todo momento que se esta construyendo con calidad y de manera eficaz lo que realmente se esta esperando.
- **Simplicidad:** Hacer exactamente lo que se necesita y no perder tiempo haciendo algo que no aporta valor. Por ejemplo, Scrum limita el tiempo de las reuniones para evitar el “análisis-parálisis”. La practica de simplicidad en el código de XP seria su equivalente desde el punto de vista de la programación.
- **Trabajo en iteraciones o ciclos cortos:** Trabajar con itereaciones permite, a su vez, volver a planificar en función de las necesidades de cada momento. Las iteraciones de XP pueden convivir con la facilidad con los sprints de scrum.
- **Respeto:** Sin respeto ninguna práctica de XP ni de Scrum podrá aplicarse, ya que ambos métodos se basan en la profesionalidad y confianza entre los integrantes de los equipos.
- **Valentia:** Es necesario ser valiente para cambiar la forma de trabajar y relacionarse.
- **Priorización:** Tanto Scrum como XP necesitan unos requisitos bien elaborados y con una prioridad bien establecida por parte del cliente para asegurar que se esta construyendo lo que interesa en cada momento.

Diseño guiado por dominio (DDD)

El Diseño Guiado por el Dominio (DDD) es un enfoque de diseño de software que trabaja en conjunto el modelado del dominio y el diseño del software, con el objetivo de crear un modelo del dominio compacto que evolucione progresivamente a través de sucesivas iteraciones del diseño.

Se entiende por dominio del software a todo aquello que está relacionado con alguna actividad relevante para el usuario y su negocio. De esta manera, es posible encontrar una creciente gran variedad de dominios pertenecientes al mundo físico tangible como un asiento en un ómnibus de larga distancia o intangible como el valor de una imagen acompañada de sonidos.

Echeverría et al. (2010) indica que en aplicaciones empresariales, las necesidades y expectativas de los usuarios provocan una gran complejidad en el manejo de información, el cumplimiento de funcionalidades requeridas, así como varias características no funcionales. Es decir que las actividades de los usuarios para el funcionamiento del negocio deben ser captadas y realizadas lo más ajustadamente posible por el equipo de desarrollo de manera que el producto resuelva de manera óptima las necesidades de manejo del negocio. Esto resulta, a veces, extremadamente complejo para ser resuelto mediante sistemas basados en computadora.

El enfoque Diseño Guiado por Dominio (DDD) plantea una serie de buenas prácticas y patrones para el desarrollo del software orientado al dominio, entre ellas tenemos:

Arquitectura de capas

El objetivo principal de DDD es aislar el dominio del resto de la aplicación. La separación de la capa de dominio de las capas infraestructura y de la interfaz de usuario da lugar a un diseño mucho más limpio de cada capa. Al tratar las capas por separado se obtienen un menor costo de mantenimiento y desarrollo puesto que cada capa evoluciona de manera diferente y responde a diferentes necesidades. La separación también ayuda al despliegue en sistemas distribuidos, permitiendo que diferentes capas se ubiquen en diferentes sitios físicos, servidores o clientes, de manera de minimizar las comunicaciones y optimizar el rendimiento.

Siguiendo a Evans autor del libro *Domain-Driven Design: Tackling Complexity in the Heart of Software* (citado en Echeverría et al., 2010), se considera el siguiente conjunto de capas:

Capa de presentación o interfaz de usuario,

Capa de aplicación.

Capa de dominio o capa del modelo.

Capa de infraestructura.

Lenguaje ubicuo

Para el buen modelado del dominio se debe hacer uso de un mismo lenguaje el cual será utilizado en las diferentes fases de modelado y por todos los involucrados del mismo. A este lenguaje común, DDD lo denomina lenguaje ubicuo o lenguaje omnipresente.

Paredes (2009) afirma que el alcance del lenguaje omnipresente de un proyecto deberá considerar:

Nombres de clases y sus operaciones destacadas.

Términos para discutir reglas de negocio que se han incorporado al modelo.

Nombres de patrones de análisis y de diseño que se aplican al modelo de dominio.

Patrón de Entidades (Entities)

Como asegura Paredes (2009) cualquier objeto del dominio que mantiene un estado y comportamiento más allá de la ejecución de la aplicación y que necesita ser distinguido de otro que tenga las mismas propiedades y comportamientos, es una Entidad. Cada objeto instanciado debe poseer un identificador único. DDD recomienda que las entidades modeladas sean lo más pequeñas y sencillas posibles, reflejando solo el comportamiento y atributos necesarios.

Patrón de objetos de valor (Value objects)

Avram y Marinescu (2006) definen a los objetos de valor como todo objeto que describe aspectos del dominio, pero no tiene identidad.

Los objetos de valor dentro de una aplicación son indistinguibles unos de otros. Consisten únicamente en un conjunto de propiedades y comportamientos pero no mantienen identidad alguna. Evans (citado en Paredes, 2009) recomienda que los objetos de valor sean inmutables y que si se requiere actualizar su valor solo se descarte el objeto de valor actual y se reemplace por uno nuevo.

Patrón de Servicios (Services)

Dentro del lenguaje ubicuo encontraremos verbos, muchos de estos verbos serán parte de los objetos de la capa de negocios en forma de métodos sin embargo hay verbos o comportamientos que no encajarán en un solo objeto si no en varios o en ninguno, dichos comportamientos suelen ser en realidad Servicios. Avram y Marinescu (2006) recomiendan distinguir servicios que pertenecen al dominio, de los servicios que puedan encontrarse en otras capas. Paredes (2009) define algunas claves para distinguirlos:

Son comportamientos que no pertenecen a ninguna entidad.

Pueden intercambiarse con facilidad por servicios de similares características.

Van a ser usados por varias entidades en distintos puntos de la aplicación.

Patrón de Módulos (Modules)

Los Módulos son mecanismos de comunicación. Cuando se decide colocar una clase dentro de algún módulo, es una forma de explicarle al próximo programador que mire el diseño, que el concepto y la importancia que representa esa clase sólo se puede entender dentro del contexto del módulo en que fue ubicada.

Paredes (2009) afirma que es importante modularizar agrupando conceptos o importancia de negocio logrando una alta cohesión, pero bajo acoplamiento.

Patrón de Agregados (Aggregates)

Una agregación es un grupo de objetos asociados que deben tratarse como una unidad a la hora de manipular sus datos. Paredes (2009) describe que cada agregación cuenta con una entidad raíz y una frontera. La frontera define qué está dentro de la agregación y que no. La entidad raíz es una entidad contenida en la agregación y será el único punto de entrada a la agregación. Esto significa que:

La entidad raíz es responsable de chequear las reglas de consistencia entre los distintos objetos de la agregación en la carga, modificación y eliminación de datos.

Fuera del contexto de la Agregación, no puede existir ninguna referencia a un miembro que no sea la Entidad Raíz. Los demás miembros de la Asociación no existen más allá de la Frontera.

Sólo la Entidad Raíz puede ser obtenida directamente a través de una consulta a la base de datos. Los demás objetos de la Agregación son obtenidos navegando a través de las Asociaciones.

Patrón de Fábrica (Factory)

Las Factorías sirven para encapsular la creación de objetos complejos. La creación de una entidad, por ejemplo, podría ser tan compleja que involucre a muchas otras entidades y objetos de valor.

Erick Evans (citado en Paredes, 2009) presenta este patrón clasificándolo como componente fundamental de un *Model-Driven Design*. El patrón nos dice que traslademos la responsabilidad de crear instancias de objetos complejos y Agregaciones a un objeto externo, que puede no tener responsabilidad en el modelo, pero que sigue siendo parte de la capa de dominio.

Patrón de Repositorio

Paredes (2009) asegura que los repositorios son similares al patrón fábrica (factory) utilizado para la creación de objetos la diferencia radica en que los repositorios ayudan a la reconstrucción de objetos ya existentes no crea objetos nuevos. Los repositorios reciben una petición para obtener un objeto o una lista de objetos que satisfagan un cierto criterio y ellos se encargan de reconstruir los objetos de donde sea que haya que hacerlo, de una base de datos, de un archivo de texto, de memoria, de una red, de un web service, etc.

Refactorización Continua

Refactorizar es importante en todo software pero muchas veces refactorizar resulta ser una tarea difícil pero necesaria. Avram y Marinescu (2006) indican que refactorizar debe ser una tarea continua y delicada sobre todo si se pretende refactorizar el dominio, cada vez que el experto de dominio utilice términos que no formen parte del lenguaje ubicuo es una señal clara que se debe refactorizar.

No siempre será claro el momento cuando se debe refactorizar Avram y Marinescu (2006) aseguran que se debe tener en cuenta tres conceptos para poder detectar conceptos implícitos en el negocio estos son: restricción, proceso y especificación.

Paredes (2009) nos brinda algunas recomendaciones de cuando refactorizar y cuando no.

Es adecuado refactorizar:

El diseño no expresa el entendimiento actual del equipo sobre el dominio.

Conceptos importantes del dominio están implícitos en el diseño (y descubrimos una forma de hacerlos explícitos).

Encontramos una oportunidad de hacer más flexible una porción importante del diseño.

No es adecuado refactorizar:

No debemos introducir mejoras cuando no podemos convencer al experto del dominio de usarlas.

No debemos introducir modificaciones que sólo demuestran virtuosismo técnico y no aportan valor al modelo.

Arquitectura basada en microservicios

El modelo de arquitectura basado en microservicios tiene como idea principal dividir el software de la aplicación para que sean lo más independiente posible entre ellos (cabe la posibilidad en función de la aplicación desarrollada que algunos servicios deben estar relacionados entre sí de manera irrefutable) y que sean fácilmente reemplazables y actualizables. No obstante, sigue siendo necesario que las aplicaciones que sigan esa estructura posean un lado cliente tal y como conocemos ahora en el que de algún modo confluyan los resultados de los procesos llevados a cabo en el lado del servidor y muestren al usuario final la funcionalidad de la aplicación de manera correcta.

La definición más general es la dada por Martín Fowler y James Lewis (citado en Pérez-Herrera, 2015), en su artículo “Microservices”. En este artículo se definen microservicios como un “estilo arquitectural en el que múltiples servicios, cada uno corriendo de manera individual y desplegados de la forma más automatizada posible, se comunicaran entre sí mediante mecanismos ligeros, generalmente un recurso API basado en HTTP”

Ventajas que brinda la arquitectura basada en microservicios:

Servicios pequeños e independientes (principio de responsabilidad única).

Unidades de despliegue pequeñas.

Reducción de tiempo de desarrollo.

Multitecnología.

Fácil escalado.

PostgreSQL

Según Denzer (2002) PostgreSQL es un avanzado sistema de bases de datos relacionales basado en Open Source, distribuido bajo licencia BSD, con su código fuente disponible libremente.

EcuRed (s.f) Afirma que PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Además de soportar casi toda la sintaxis SQL contando también a su disposición con varios lenguajes de programación incluyendo C, C++, Java, Perl, Tcl, y Python.

Según PostgreSQL (2012) destaca entre sus ventajas:

Completa documentación.

Ampliamente popular - Ideal para tecnologías Web.

Su sintaxis SQL es estándar y fácil de aprender.

Multiplataforma.

Adicionalmente según Boundless (s.f) existe PosGIS creado para dar soporte de objetos geográficos los cuales permitirán trabajar con tipos de datos especiales como límites y dimensiones convirtiendo a PostgreSQL en una base de datos relacional de datos espaciales libre.

Eureka

Rodríguez López (2015) indica que Eureka es un servidor para el registro y localización de microservicios, balanceo de carga y tolerancia a fallos. La función de Eureka es registrar las diferentes instancias de microservicios existentes y sus diferentes datos como localización y estado.

Cada microservicio al iniciar enviara datos al servidor Eureka quien registrara y mantendrá actualizada estos datos, cada 30 segundos Eureka comprobará el estado del microservicio si luego de tres periodos el microservicio no devuelve ningún dato Eureka lo eliminara de su registro.

De la misma forma una vez recibido tres notificaciones con datos Eureka considerará el microservicio disponible.

Las ventajas que brinda Eureka son:

Eureka devuelve la localización y puerto del microservicio a utilizar conociendo solamente su identificador.

Eureka nos muestra una interfaz que permite visualizar los microservicios existentes actualmente en el registro y sus estados.

Se puede configurar como una agrupación de microservicios incrementando notablemente su tolerancia a fallos.

Eureka fue inicialmente desarrollado por Netflix, quien utiliza AWS como IaaS, así Eureka fue diseñado para integrarse fácilmente con los servicios de Amazon.

Cloud-config

Rodríguez López (2015) define a Cloud-config como un servidor de configuración pensado para sistemas distribuidos, cuya principal función son las propiedades de configuración de los microservicios.

Cloud-config aporta las siguientes ventajas:

Integración con Eureka.

Proporciona un repositorio de configuración centralizado y un historial de el mismo.

Proporciona encriptado opcional para los valores de las propiedades.

Zuul

Zuul es un servidor compuesto de filtros cada uno con una funcionalidad especial ya sea enrutamiento dinámico, balanceo de carga, monitorización y segurización de peticiones. En una arquitectura basada en microservicios normalmente Zuul será el punto de entrada y se encargará de redirigir, balancear y brindar seguridad a las diversas peticiones que reciban los microservicios.

Rodríguez López (2015) menciona algunas ventajas de Zuul:

Proporciona un sistema que puede reaccionar de forma rápida cambiando su comportamiento ante diferentes situaciones problemáticas: esto se consigue a través de los diferentes filtros de los que está compuesto Zuul.

Permite la implementación de filtros personalizados y la carga de estos cuando ya está trabajando, lo que permite reequilibrar carga, redirigir peticiones, cancelarlas etc.

RestFul

Cabe aclarar que cuando hablamos de Rest y RestFul no hablamos de conceptos diferentes, sino que es lo mismo, salvo ciertos matices.

Tyagi (2006) en su artículo afirma que REST es una arquitectura cliente-servidor basada en estándares web y utiliza el protocolo HTTP para la comunicación de datos. En la que los servicios web se ven como recursos y pueden identificarse mediante sus URL. Además, TutorialesPoint (s.f) asegura que el recurso se accede mediante una interfaz común que utiliza métodos estándar HTTP. REST fue presentado por primera vez por Roy Fielding en el año 2000.

Según Wajser, D (2015) para dejar en claro que es RESTful debemos tener en cuenta los siguientes principios:

El primero de estos principios es saber que todo lo que se mueve a través de las comunicaciones web es un recurso.

El segundo principio que tenemos que tener en cuenta, es que cada uno de estos recursos debe tener un identificador único, el cual va a estar dado por su URL

En tercer lugar, es que este protocolo de transmisión de datos debe utilizar los verbos estándares de HTTP, que están definidos en el protocolo nativo, donde cada uno de estos verbos significa una acción diferente. Hay definidas 8 acciones principales: GET, POST, PUT, DELETE, HEAD, OPTIONS, TRACE, CONNECT.

El cuarto principio y clave en este tipo de arquitectura es, que cada recurso puede tener múltiples representaciones, independientemente de cómo esté almacenado. Un ejemplo sencillo para ilustrar este principio sería tener un recurso en formato XML y poder solicitarlo en JSON.

El quinto principio es clave para entender las comunicaciones cliente-servidor, se trata de comunicaciones que se denominan sin estado (stateless), lo que significa que cada petición al servidor es tratada de manera totalmente independiente.

Un diseño RESTful promete un servicio que sea fácil de implementar, sostenible, extensible y escalable. (Vaqqas ,2014)

Maven

Según García Oterino (2014) Maven es una herramienta open-source para la gestión de proyectos de software, que se creó en 2001. Por otro lado Pérez Garcia (2006) basado en el concepto de POM (Project Object Model) nos dice que con Maven vamos a poder compilar, empaquetar, generar documentación, pasar los test, preparar las builds (compilar y generar ejecutables a partir del código fuente), etc.

Permitiendo a los desarrolladores moverse entre proyectos sin la necesidad de aprender como compilar o empaquetar. Esto mejora el mantenimiento y la reusabilidad.

Según Maven (2017) el objetivo principal de Maven es permitir que un desarrollador comprenda el estado completo de un esfuerzo de desarrollo en el período más corto de tiempo. Para alcanzar este objetivo hay varias áreas de preocupación que Maven intenta tratar las cuales son:

Hace el proceso de construcción fácil

Proporcionar un sistema de construcción uniforme, utilizando su modelo de objetos de proyecto (POM).

Proporcionar información de proyectos de calidad

Proporcionar directrices para el desarrollo de mejores prácticas, sugiriendo algunas pautas sobre cómo diseñar la estructura de directorios de su proyecto para que una vez que aprenda el diseño, pueda navegar fácilmente por cualquier otro proyecto que utilice Maven y los mismos valores predeterminados. Además de ayudar en el flujo de trabajo del proyecto, como la gestión de versiones y el seguimiento de problemas.

Permitir la migración transparente a nuevas funciones.

JAVA

Java es un lenguaje de programación orientada a objetos, popular y de amplio uso que está desarrollado por Sun Microsystems a principios de los años 90.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (Raico y Linares, 2012, p.57)

Según Rubio y Castillo (2016) Java es un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera “bytecodes” es interpretado

por una máquina virtual. Además, con la programación en Java se puede desarrollar aplicaciones que se ejecutan en el entorno de un navegador web.

Actualmente con Java podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que deseemos hacer con acceso a través web se puede hacer utilizando Java.

Bootstrap

Según Fontela (2015) Bootstrap es un framework desarrollado inicialmente en el año 2011 por Twitter, liberado bajo licencia MIT que permite dar forma a un sitio web mediante librerías CSS y JavaScript que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web.

Bootstrap es una excelente herramienta para crear interfaces de usuario totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Por otro lado según Rodríguez (2012) las características principales de Bootstrap es ofrecer una serie de plantillas CSS y ficheros Javascript que nos permiten integrar el framework de forma sencilla en nuestros proyectos webs. No obstante, el desarrollo de Bootstrap continúa en un repositorio de GitHub.

1.4 Variables

A continuación, se muestra el análisis de variables dependiente e independiente respectivamente

Variables dependientes.

Rutas

Definición conceptual: Trayectos establecidos que va de un sitio a otro utilizados por la empresa.

Definición operacional: Trayecto de calles y avenidas que deben ser recorridas por las unidades de transporte en cada día de trabajo.

Dimensiones:

Rutas con servicio: Rutas por las cuales las unidades de transporte público de la empresa están obligadas a recorrer.

Rutas sin servicio: Rutas por las cuales temporalmente las unidades de transporte no recorren.

Indicadores:

Número de rutas recorridas.

Cantidad de recorridos por ruta.

Tiempos de recorrido

Definición conceptual: Duración del recorrido de un trayecto.

Definición operacional: Tiempo que demora en recorrer una ruta cada unidad de transporte público.

Dimensiones:

Tiempo establecido de recorrido: Tiempo estimado de duración de recorrido establecido por la empresa

Tiempo de retraso de recorrido: Tiempo excedente con respecto al tiempo establecido de recorrido.

Indicadores:

Promedio de tiempo de recorrido por ruta.

Promedio de tiempo de retraso de recorrido por ruta.

Promedio de tiempo de intervalos de salida.

Variable independiente.

El sistema informático web de control de recorridos

Definición conceptual: Sistema que es utilizado accediendo por medio de internet o un servidor web local mediante un navegador.

Definición operacional: Sistema que podrá ser utilizado en cualquier navegador web (chrome, Firefox, etc) que controlará diariamente los recorridos de cada vehículo brindando información si cumple las normas establecidas por la empresa.

1.5 Conceptos y Definiciones

- **Unidad**

Combis que brindan servicio de transporte público en la empresa las cuales pertenecen a un socio específico.

- **Socio**

Hace referencia a los socios que conforman la empresa para especificar el número de acciones con los que cuentan, su fecha de ingreso y retiro de la empresa, así como algunos datos de algún pariente cercano.

- **GPS**

Sistema de Posicionamiento Global. Dicho sistema fue creado por el Departamento de Defensa de los Estados Unidos y permite, a través de una red de 24 satélites, indicar la posición de un cuerpo en la superficie terrestre con gran precisión. El GPS recurre al método matemático conocido como trilateración para trabajar con la información que aportan los satélites y así determinar la ubicación del objeto. (Julián Pérez Porto,2018)

- **Punto de control**

Hace referencia a los relojes que marcan el tiempo al pasar una unidad.

- **Tarjeta**

Tarjeta que se le otorga diariamente a cada unidad en donde se registrarán cada recorrido de la unidad en el día.

- **Tiempo establecido**

Tiempo que establece la empresa la cual debe cumplir cada unidad al pasar por cada punto de control asignado a una ruta.

- **Metodología**

Hace referencia al conjunto de técnicas, procedimientos y soportes documentales empleados en el diseño de sistemas de información. Su objetivo principal es exponer una serie de técnicas clásicas y modernas de modelado de sistemas que permitan desarrollar un software de calidad, que incluyen heurísticas de construcción y criterios de comparación de modelos de sistemas. (concepto definiciones, 2014)

- **Requerimiento**

Un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.

Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. (Leando Alegsa, 2016)

- **Servidor**

Los cloud servers son unas alternativas para llevar la herramienta de los servidores al mundo virtual. La infraestructura en la nube se consigue gracias a la existencia de diversos servidores físicos controlados mediante un software, que es el encargado de virtualizar la plataforma.

Los servidores en la nube ofrecen a las empresas la posibilidad de tener un servidor a medida de sus necesidades, cuyos recursos y capacidades puedan ir incrementándose a

conforme aumentan el tamaño y la actividad de la empresa, lo que permite un considerable ahorro para el presupuesto de las distintas corporaciones. (infortelecom,2016)

- **VPS**

Un servidor virtual privado (VPS), forma parte de un servidor físico, que se encuentra dividido en diferentes servidores virtuales. Esta partición permite a cada VPS trabajar para un cliente como si de una sola máquina se tratase, con su propio disco duro, memoria RAM y velocidad de transmisión, entre otros aspectos. Sin embargo, el servidor real es compartido con otros clientes del resto de VPS alojados en él. Por tanto, un VPS se encuentra a mitad camino entre un servidor dedicado y un alojamiento compartido, ya que trabaja como una máquina privada para un cliente, pero comparte espacio físico con otros VPS. (ValorTop, 2017)

- **PaaS**

Las soluciones PaaS son plataformas de software para las cuales la herramienta de desarrollo en si misma está alojada en la nube y se accede a través de un navegador web. Con PaaS, los desarrolladores pueden construir aplicaciones web sin tener que instalar ninguna herramienta adicional en sus computadoras, y luego despliegan estas aplicaciones sin necesidad de tener ningún conocimiento administrativo especializado. (Seta, 2009).

Capítulo 2 Métodos y Materiales

2.1 Formulación del Problema

¿El desarrollo del Sistema informático web de control de recorridos permitirá regular las rutas y tiempos de las unidades de transporte público de la empresa “El Chasqui S.A”?

2.2 Objetivos de la Investigación

2.2.1 Objetivo General.

Desarrollar un sistema informático web de control de recorridos que permita regularizar las rutas y tiempos de las unidades de transporte público de la empresa “El Chasqui S.A”

2.2.2 Objetivos Específicos.

- Obtener información de la empresa para definir los requerimientos.
- Comprender el flujo de sus procesos de la empresa.
- Establecer un lenguaje común con el personal de la empresa para el entendimiento fluido de los procesos.
- Aplicar un enfoque ágil para un desarrollo organizado y estandarizado del sistema.
- Generar pruebas para la detección de errores del sistema.
- Brindar soporte a las primeras versiones del sistema durante el desarrollo del mismo.

2.3 Justificación e Importancia de la Investigación

2.3.1 Justificación Social.

Podemos señalar que se mejorará la calidad de atención a los clientes ya que gracias al sistema se podrá tener un mejor control a los choferes de las unidades para que brinden el servicio a las rutas correspondientes, por otro lado, se reducirá el esfuerzo en las actividades diarias que realizan los usuarios.

2.3.2 Justificación Económica.

Como justificación económica, una vez el proyecto de investigación esté realizado, permitirá a la empresa el Chasqui S.A mejorar sus ingresos de forma óptima ya que el dinero invertido en el mantenimiento de los relojes controladores y las tarjetas diarias será sustituido por el sistema el cual también ayudara en el control de unidades que deben laborar debido a su aportación diaria hacia la empresa.

2.3.3 Justificación Tecnológica.

El desarrollo de un sistema web mejorará la seguridad y disposición de la información cuando se necesite. Las pruebas que se realizarán a cada módulo del sistema informático permitirán contar con el nivel técnico necesario para que sea calificado como un sistema robusto, confiable y seguro. Permitiendo un aprovechamiento de las tecnologías que hoy en día son de gran importancia en las organizaciones para mejorar sus procesos.

2.3.4 Justificación Operativa.

El sistema web mejorará el modo de trabajo, acelerando los procesos existentes a la vez que la información será procesada más eficientemente, lo cual ahorrará tiempo en generación de reportes de forma oportuna y confiable para la empresa El Chasqui S.A.

2.3.5 Importancia.

Por lo tanto, implementamos este sistema web, con el cual se mejore el proceso de control de tiempos en las unidades y supervisión de recorrido de rutas en tiempo real, así como un

almacenamiento de datos seguros, de manera que genere información confiable de forma oportuna cuando se requiera. Por otro lado, se tendrá una mejor seguridad, organización y rapidez de la información que maneja la empresa.

2.4 Hipótesis

Si se desarrolla el sistema informático web de control de recorridos entonces permitirá regularizar las rutas y tiempos de las unidades de transporte público en la empresa “El Chasqui S.A.”

2.5 Tipo de Investigación

Tecnológica Formal

2.6 Limitaciones de la Investigación

A continuación, se presentan las limitaciones o restricciones que surgieron durante la realización de esta investigación:

- Falta de disponibilidad de tiempo del personal de la empresa.
- Existe una desconfianza al brindar información por parte del personal.
- Falta de equipos necesarios para la implementación de un sistema que funcionara bajo tecnología web.
- El sistema tendrá un presupuesto de S/. 3417.00.
- Falta de internet por parte de la empresa.
- Resistencia al cambio por parte del personal de la empresa hacia el sistema web.

2.7 Análisis de Costos y Beneficios

2.7.1 Análisis de Costos.

- ✓ **Duración del Proyecto:** 7 meses

✓ **Bienes**

- **Materiales de Oficina** (Ver tabla 01)

Tabla 01

Costo de Materiales de Oficina para el Desarrollo del Proyecto

INSUMOS	UNIDAD DE MEDIDA	CANTIDAD	VALOR UNITARIO (S/.)	SUBTOTAL (S/.)
Paquete de hojas bond A4	Millar	1	12.00	12.00
Lapiceros	Unidad	5	5.00	5.00
Plumones gruesos	Unidad	5	2.00	10.00
Papel sabana	Unidad	6	0.50	3.00
Cinta masking tape	Unidad	2	2.50	5.00
Limpiatipo	Unidad	2	3.00	6.00
CD's	Unidad	5	2.00	10.00
Cuaderno	Unidad	1	4.00	4.00
	Total			55.00

Fuente: Elaboración Propia.

- **Suministros de Cómputo y otros** (Ver tabla 02)

Tabla 02

Costo de Suministros de Cómputo y Otros para el Desarrollo del Proyecto

INSUMOS	UNIDAD DE MEDIDA	CANTIDAD	VALOR UNITARIO (S/.)	SUBTOTAL (S/.)
Memoria USB 16G	Unidad	1	29.00	29.00
	Total			29.00

Fuente: Elaboración Propia.

- **Costo Total de Bienes** (Ver tabla 03)

Tabla 03

Costo Total de Bienes para el Desarrollo del Proyecto

DESCRIPCIÓN	SUBTOTAL (S/.)
Materiales de Oficina	55.00
Suministros de Cómputo y otros	29.00
Total	84.00

Fuente: Elaboración Propia.

- ✓ **Equipos** (Ver tabla 04)

Tabla 04

Costo de Equipos para el Desarrollo del Proyecto

DESCRIPCIÓN	CANTIDAD	VALOR UNITARIO (S/.)	DEPRECIACIÓN (%)	SUBTOTAL (S/.)
PC de escritorio	2	1500.00	25 %	750.00
Dispositivo GPS	100	52.42	25%	1310.50
	Total			2060.50

Fuente: Elaboración Propia.

(*) Cálculo de la depreciación de PC de escritorio: para este cálculo se utilizará la depreciación constante. El porcentaje de la depreciación es de 25% y la vida útil para equipos de hardware es de 4 años.

(*) Cálculo de la depreciación de dispositivo GPS: para este cálculo se utilizará la depreciación constante. El porcentaje de la depreciación es de 25% y la vida útil para dispositivos Gps es de 4 años.

La fórmula es la siguiente:

$$D = \frac{P - VS}{N}$$

Donde:

P: Inversión Inicial

VS: Valor de Salvamiento

N: Nivel de Vida Útil expresada en años.

✓ **Software** (Ver tabla 05)

Es el costo que se gastó en el desarrollo del software, en este caso como todo es software libre el costo es cero soles para el desarrollo del sistema.

Tabla 05

Costo de Software para el Desarrollo del Proyecto

DESCRIPCIÓN	CANTIDAD	VALOR	
		UNITARIO (S/.)	SUBTOTAL (S/.)
S.O Windows 10 Pro Gratuito	1	0.00	0.00
Servicio Heroku	1	0.00	0.00
Servicio Traccar	1	0.00	0.00
Servicio Digital Ocean	1	16.50	16.50
Spring Tool Suite	1	0.00	0.00
pgAdmin 4	1	0.00	0.00
Apache Tomcat	1	0.00	0.00
	Total		16.50

Fuente: Elaboración Propia.

✓ **Servicios** (Ver tabla **¡Error! No se encuentra el origen de la referencia.06**)

Tabla 06

Costo de Servicios para el Desarrollo del Proyecto

DESCRIPCIÓN	SUBTOTAL (S/.)
Movilidad	392.00
Internet	360.00
Luz	300.00
Fotocopiado	21.00
Impresiones	20.00
Anillado	15.00
Empastado	45.00
Quemado de CD's	15.00
Total	1256.00

Fuente: Elaboración Propia.

✓ **Consolidado de costos para el Desarrollo del Proyecto** (Ver tabla 07)

Tabla 07

Costo Total para el Desarrollo del Proyecto

DESCRIPCIÓN	SUBTOTAL (S/.)
Bienes	84.00
Equipos	2060.50
Software	16.50
Servicios	1256.00
Total	3417.00

Fuente: Elaboración Propia.

2.7.2 Análisis de Beneficios.

Los beneficios por lo general aumentan las ganancias o reducen los costes; ambas características muy importantes para todo nuevo sistema informático. Se pueden clasificar en tangibles e intangibles

BENEFICIOS TANGIBLES

a) Reducción de horas extras laborales en labor de reportes:

La generación de reportes tales como las planillas diarias de los recorridos de las unidades se obtiene estrictamente por el cumplimiento efectivo de labores adicionales ya sea más allá del horario habitual o por mayor carga. El monto de las horas extras asciende a la suma máxima de S/. 5.59 soles diarios, con un máximo de 26 días laborables al mes.

Este costo de horas extras se ahorrará con la implementación del sistema, ya que el trabajador podrá agilizar sus actividades y entregar sus reportes a tiempo sin necesidad de venir en horario fuera de oficina.

El costo del ahorro calculado al mes es:

$S/. 5.59 \times 26 \text{ días} \times 1 \text{ trabajador} = S/. 145.37 \text{ al mes.}$

b) Ahorro en el material utilizado para el proceso:

Con la implantación y uso del sistema se minimizará el costo del material que se desperdiciaba en los procesos manuales, todo ello convirtiéndose en ahorro para la Empresa. A continuación, se detalla en la tabla 08 cuales son estos materiales con la cantidad y el costo equivalente para cada uno de ellos.

Tabla 08

Costo del Material Desperdiciado en los Procesos sin Uso del Sistema Informático

MATERIAL	UNIDAD DE MEDIDA	CANTIDAD DE PÉRDIDA X MES	PRECIO UNITARIO	SUBTOTA L
Papel utilizado para Reportes e Informes	Paquete	0,5	S/. 10,00	S/. 5,00
Cartulinas para Tarjetas	Millar	0,05	S/. 260,00	S/. 13,00
Cinta para reloj marcador	Unidad	0,25	S/. 275,00	S/. 68,75
Pilas para reloj marcador	Unidad	0,25	S/. 160,00	S/. 40,00
Lapiceros	Unidad	0,375	S/. 8.00	S/. 3,00
Total, Mensual				S/. 129,75

Fuente: Elaboración Propia.

En la tabla 09 se muestra el costo total ahorrado con el uso del sistema informático.

Tabla 09

Costo de los Rubros de Ahorro Generados con Uso del Sistema Informático

RUBROS DE AHORRO GENERADOS CON EL SISTEMA INFORMÁTICO	RRHH	SIN EL USO DE UN SISTEMA INFORMÁTICO COSTO MENSUAL S/S	CON EL USO DE		
			UN SISTEMA INFORMÁTICO COSTO MENSUAL C/S	AHORRO MENSUAL	AHORRO ANUAL
Ahorro en pago de horas extras de trabajadores en labor de reportes	1	145,37	0,00	145,37	1744,44
Ahorro en el mantenimiento de los relojes controladores	1	166,67	0,00	166,67	2000,00
Ahorro en el material utilizado para el proceso		129,75	0,00	129,75	1557,00
Total, de Beneficio				831,79	5301,44

Fuente: Elaboración Propia.

Nota: Los datos del ahorro fueron proporcionados por la empresa “El chasqui S.A”

BENEFICIOS INTANGIBLES

- Mejora de la imagen de la Empresa “El Chasqui S.A”.
- Mejora del nivel de satisfacción de los trabajadores ya que recibirían pagos puntuales por labores extraordinarias al tener reportes actualizados del estado situacional de la Empresa.
- Mejora del nivel de satisfacción de los socios por mejora en los procesos de sus unidades de transporte público.
- Mejora en el control de las unidades por los controladores para brindar un mejor servicio por las rutas establecidas por la empresa.
- Mejora el Acceso rápido a la información para la toma de decisiones.
- Ahorra tiempo y esfuerzo enorme en el control de tiempos y recorridos por parte de las unidades, igualmente en la realización de reportes.
- Apoya al crecimiento de la Empresa.
- Mejora en la productividad de los trabajadores.
- Permite una diferenciación ante las demás Empresa de transporte público de la ciudad de Chiclayo.

2.7.3 Flujo de Caja Incremental.

El flujo de caja incremental es el ingreso adicional que se genera cuando una empresa u otro tipo de organización ponen en marcha un nuevo proyecto. (Prucommercialre, 2016) , así como también es una técnica de medición de retorno de la inversión que ayuda a la toma de decisiones con respecto a hacer una inversión o un cambio en las políticas de administración (Shadunsky A. , 2018) , lo cual nos permitirá evaluar si este proyecto es factible o no.

a) Ingresos Incrementales:

En los ingresos incrementales se incluye los rubros que generan ahorro como son los Beneficios Tangibles descritos anteriormente, obtenidos a partir del primer año.

b) Egresos:

✓ Inversión para el desarrollo; Costo total de los rubros que conlleva el desarrollo del proyecto como son:

- Bienes
- Equipos
- Software
- Servicios.

✓ Inversión para la operación; costo total de los equipos y dispositivos que se utilizarán a partir de la implantación del sistema informático en la Empresa:

Hardware

- 2 PC de escritorio para mejoras del sistema informático
- 100 dispositivos GPS para la obtención del posicionamiento de las unidades para su control de recorridos dentro del sistema.

Software

- Servicio Heroku
- Servicio Traccar
- Servicio Digital Ocean
- Spring Tool Suite
- pgAdmin 4
- Apache Tomcat
- SO Windows 10 Pro

- Costo para la operación; son aquellos costos que se invierten cada año a partir del año 1 con el uso del sistema, aquí consideramos lo siguiente:
 - Mantenimiento de Software; se estableció como bolsa de trabajo, ya que puede ser realizado por un practicante de sistemas o computación informática, el costo por el mantenimiento es de S/. 930.00 mensuales y se podrá realizar cada cuatro meses.

A continuación, en la tabla 10 presentamos el detalle con los costos para el Flujo de Caja Incremental para este proyecto.

Tabla 10

Desarrollo del Flujo de Caja Incremental

Flujo De Caja Incremental						
Rubros / Tiempo	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Ingresos Incrementales		5301,44	5301,44	5301,44	5301,44	5301,44
Ahorro en pago de horas extras de trabajadores en labor de reportes.		1744,44	1744,44	1744,44	1744,44	1744,44
Ahorro en el mantenimiento de los relojes controladores.		2000,00	2000,00	2000,00	2000,00	2000,00
Ahorro en el material utilizado para el proceso		1557,00	1557,00	1557,00	1557,00	1557,00
Egresos	6417,00	2988,00	2988,00	2988,00	2988,00	2988,00
Inversión (Para Desarrollo)	3417,00					
Bienes	84,00					
Equipos	2060,50					
Software	16,50					
Servicios	1256,00					

Fuente: Elaboración Propia.

Tabla 10 Continuación

Desarrollo del Flujo de Caja Incremental

Flujo De Caja Incremental						
Rubros / Tiempo	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Inversión (Para La Operación)	3000,00					
Hardware						
PC de escritorio	3000,00					
Software						
S.O Windows 10 Pro Gratuito	0,00					
Servicio Heroku	0,00					
Servicio Traccar	0,00					
Servicio Digital Ocean		198.00	198.00	198.00	198.00	198.00
Spring Tool Suite	0.00					
pgAdmin 4	0.00					
Apache Tomcat	0.00					
Costos De Operación	0,00	2790,00	2790,00	2790,00	2790,00	2790,00
Mantenimiento de software (bolsa de trabajo)		2790,00	2790,00	2790,00	2790,00	2790,00
Flujo De Caja	-6417,00	2313,44	2313,44	2313,44	2313,44	2313,44

Fuente: Elaboración Propia.

a) Análisis del VAN

Es un indicador financiero que sirve para determinar la viabilidad de un proyecto. Si tras medir los flujos de los futuros ingresos y egresos y descontar la inversión inicial queda alguna ganancia, el proyecto es viable. (Santa Cruz E., 2017).

- Tasa social de descuento: representa el costo de oportunidad en que incurre el país cuando utiliza recursos para financiar sus proyectos; actualmente ésta Tasa es de 8% para proyectos de inversión pública. (Diario Oficial El Peruano, 2017)
- Inversión inicial: **S/. 6417,00**
- Flujo de caja por año: **S/.2313,44**

El V.A.N. se obtuvo en base a la siguiente fórmula usando Excel:

$$\text{VAN} = -I_0 + \sum_{t=1}^n \frac{FC_t}{(1+Td)^t}$$

$$\text{VAN} = -I_0 + \frac{FC_1}{(1+Td)^1} + \frac{FC_2}{(1+Td)^2} + \frac{FC_3}{(1+Td)^3} + \frac{FC_4}{(1+Td)^4} + \frac{FC_5}{(1+Td)^5}$$

$$\text{VAN} = -6417.00 + \frac{2313.44}{(1+0.08)^1} + \frac{2313.44}{(1+0.08)^2} + \frac{2313.44}{(1+0.08)^3} + \frac{2313.44}{(1+0.08)^4} + \frac{2313.44}{(1+0.08)^5}$$

$$\text{VAN} = \text{S/. 2819.90}$$

V.A.N. > 0: La inversión producirá ganancias por encima de la rentabilidad exigida, es decir que es un proyecto viable.

b) Análisis de la tasa interna de retorno (T.I.R.)

Es un indicador de la rentabilidad de un proyecto, que se lee a mayor TIR, mayor rentabilidad. Por esta razón, se utiliza para decidir sobre la aceptación o rechazo de un proyecto de inversión. (nvindi , 2018).

$$VAN = 0 = \sum_{t=1}^n \frac{FC_t}{(1 + TIR)^t}$$

$$0 = \frac{2313.44}{(1 + TIR)^1} + \frac{2313.44}{(1 + TIR)^2} + \frac{2313.44}{(1 + TIR)^3} + \frac{2313.44}{(1 + TIR)^4} + \frac{2313.44}{(1 + TIR)^5}$$

$$TIR = 0.24 = 24\%$$

T.I.R. (24%) > Tasa Social de Descuento (8%): Proyecto financieramente viable sin que genere pérdidas.

Tabla 11

Interpretación de los Resultados

Tasa Social de Descuento	8 %	
Valor Actual Neto (Van)	S/2819,90	Se acepta el proyecto por tener un VAN positivo
Tasa Interna De Retorno (TIR)	24%	Se acepta el proyecto por tener un TIR mayor a la tasa de descuento social (8%)

Fuente: Elaboración Propia.

2.8 Periodo de Recuperación de la Inversión

El período de recuperación de la inversión (PRI) es un indicador que mide en cuánto tiempo se recuperará el total de la inversión a valor presente. Puede revelarnos con precisión, en años, meses y días, la fecha en la cual será cubierta la inversión inicial. (Santa Cruz E., 2017).

La fórmula es la siguiente:

$$\text{PRI} = \frac{a + (b - c)}{d}$$

d

Donde:

a = Año inmediato anterior en que se recupera la inversión.

b = Inversión Inicial.

c = Flujo de Efectivo Acumulado del año inmediato anterior en el que se recupera la inversión.

d = Flujo de efectivo del año en el que se recupera la inversión (Ver tabla 12).

Tabla 12

Flujo de Efectivo

AÑOS	FLUJO DE EFECTIVO A VALOR PRESENTE
0	S/. 6417,00
1	S/. 2313,44
2	S/. 2313,44

Fuente: Elaboración Propia.

Así, los datos son:

a = 1 año

b = 6417,00

c = 2313,44

d = 2313,44

Reemplazando los valores en la fórmula:

$$\text{PRI} = 1 + \frac{(6417,00 - 2313,44)}{2313,44} = \mathbf{2,773791 \text{ años}}$$

Para determinar el número de meses se resta el número entero y luego se multiplica por 12:

$$0,773791 \times 12 = \mathbf{9,29 \text{ meses}}$$

Para expresar el número de días restamos nuevamente el número entero y multiplicamos la fracción por 30 que es el número de días:

$$0,29 \times 30 = \mathbf{8,56 \text{ días}}$$

Así, en este ejemplo tenemos que la recuperación de la inversión tomará **2 años, 9 meses y 8 días aproximadamente.**

Capítulo 3 Resultados y Discusión

3.1 Sprint N° 0

En este Sprint se realiza un estudio de los procesos de la empresa por medio de reuniones con los usuarios que intervienen en los procesos importantes del sistema, obteniendo así una lista inicial de requerimientos, el modelo de procesos, acta de constitución del proyecto y el plan de versiones.

3.1.1 Fase de entrevistas

Conforme al cronograma del proyecto se realizó una fase inicial con las personas involucradas en los procesos del negocio relevantes para el sistema con la finalidad de obtener un alcance inicial:

Fase N°01 de entrevista:

Junta Directiva de la empresa “El Chasqui S.A.”

Se realizó la entrevista con los 5 miembros de la junta directiva presididos por el Sr. Lluncor Gonzales Fausto, cada uno de los miembros expresó la situación actual de la empresa, así como también los diversos inconvenientes que se presentan en algunos procesos de la empresa.

Funciones:

- Plantear el recorrido de las rutas a recorrer en la empresa.
- Plantear los tiempos de recorrido de cada ruta de la empresa.
- Plantear el monto de sanción por retraso.
- Plantear el monto de aportación diaria y sus conceptos.
- Convocar juntas de socios de manera excepcional.

Posteriormente se realizó la entrevista con la secretaria y el controlador de la empresa.

Secretaria de la empresa “El Chasqui S.A.”: Srta. Solano Bravo Olivia

Función:

- Registrar el pago diario de cada unidad que laboró.
- Generar reporte de recorridos de unidades y sus tiempos.
- Generar reporte de unidades que laboraron en el día.
- Generar reporte de aportaciones de los socios.
- Registro de socios, unidades y personal.

Controlador de la empresa “El Chasqui S.A.”: Sr. Enrique Colchado Correa

Función:

- Asignar las tarjetas a cada unidad que labora.
- Registrar rutas y tiempos de recorrido por cada vuelta de la unidad.
- Aplicar sanciones a las unidades.
- Asignar minutos de tolerancia según la situación.
- Reportar cualquier incidente suscitado en el terminal.

3.1.2 Identificación de Stakeholders o Interesados

A continuación, se detallan los involucrados y el rol que desempeñan en este proyecto, en la siguiente Tabla 13

Tabla 13

Lista de Stakeholders o Interesados

ID	NOMBRE	PUESTO	ORGANIZACIÓN	ROL
A	Montenegro Barboza Agustín	Gerente	Empresa “Chasqui S.A.”	Sponsor del proyecto
B	Correa Saldaña Olivia	Secretaria	Empresa “Chasqui S.A.”	Usuario del sistema

C	Enrique Colchado Correa	Controlador	Empresa “Chasqui S.A.”	Usuario del sistema
D	Gil Estela Fortunato	Controlador	Empresa “Chasqui S.A.”	Usuario del sistema

Fuente: Elaboración Propia

3.1.3 Elaborar Modelo de Procesos.

Con la información recaudada en las entrevistas iniciales se hizo una descripción de cada proceso relevante con el sistema con el fin de dar una mejor comprensión de cómo se trabaja en cada uno de los procesos y las personas que intervienen ayudando también a identificar las tareas que serían optimizadas por el sistema. A continuación, describiremos los procesos más importantes para el sistema.

Proceso de asignar tarjeta:

Este proceso inicia en la ventanilla del controlador, el chofer llega a solicitar el primer recorrido que realizará junto a su hora de partida, el controlador identifica la unidad y verifica si tal unidad tiene alguna sanción para no poder laborar, si no tiene ninguna sanción para no poder laboral, el controlador otorga su respectiva tarjeta la cual contiene: número, placa de la unidad, ruta y tiempo que recorrerá generando así el proceso de asignación de ruta; por lo contrario si tiene alguna sanción que no le permite laborar a la unidad se termina el proceso para esta unidad.

Proceso de asignar ruta:

El proceso de asignar ruta inicia en la ventanilla del controlador, el chofer llega a solicitar el recorrido que realizara junto a su hora de partida entregando la tarjeta de control, el controlador identifica el vehículo por medio de la tarjeta y verifica que número de vuelta a realizar le corresponde para designarle la ruta correspondiente; si es su primer recorrido de

la unidad el controlador designa la ruta que realizara la unidad junto a su hora de partida; si no es el primer recorrido a realizar por la unidad, entonces se verifica sus minutos de retraso que obtuvo en cada punto de control, si no obtuvo minutos de retraso se le asigna la nueva ruta y la hora de partida a la unidad con normalidad; si obtuvo minutos de retraso entonces interviene el proceso de asignar sanción.

Proceso de asignar sanción

Este proceso inicia en la ventanilla del controlador, el chofer llega a solicitar el recorrido que realizara junto a su hora de partida entregando la tarjeta de control después de haber realizado su primera ruta o recorrido, el controlador identifica la unidad y se verifica sus minutos de retraso que obtuvo en cada punto de control, si obtuvo minutos de retraso entonces se ejecuta la sanción según los minutos de retraso, si en caso hay justificación fundamentada por los minutos de retraso se levanta la sanción y se le asigna la nueva ruta con su hora de partida a la unidad normalmente, sea el caso que no haya justificación para la sanción asignada la unidad tendrá que acatar la sanción de permanencia de la unidad en el terminal.

Proceso de registrar socio:

El proceso de registrar socio inicia en la oficina de la empresa, el socio llega a realizar el registro de sus datos personales y de su unidad que laborara en la empresa, la secretaria identifica al socio y verifica si realmente pertenece a la sociedad y cuantas acciones posee, si se confirma la pertenencia del socio para la empresa entonces se procede al llenado de sus datos personales y de su o sus unidades; si no perteneciera a la sociedad el proceso finaliza de inmediato; si en caso se realice el traspaso de la sociedad a una persona externa pero con la misma unidad que ya labora en la empresa, entonces la secretaria realizara la modificación del socio y dueño de la unidad.

Proceso de registro de tiempos:

Este proceso inicia con la junta directiva que en consenso y decisión unánime otorgan los tiempos obligatorios y sus respectivos minutos de tolerancia a cada punto de control según la ruta que debe recorrer una unidad, una vez tomada la decisión el gerente designa a la secretaria a cargo a registrar los tiempos establecidos y esta a su vez le designa al controlador para tenerlos en cuenta en el proceso de asignación de ruta, si se da el caso de que una calle este averiada dentro del recorrido de alguna ruta, el controlador verifica que tal hecho es real, si la calle esta averiada entonces se procederá a brindar minutos adicionales de tolerancia.

3.1.4 Requerimientos del Negocio.

Los requerimientos del negocio fueron desarrollados a partir de las entrevistas con los usuarios y las necesidades presentadas en cada proceso de la empresa relevante para el sistema, a continuación, se muestran en la siguiente Tabla 14.

Tabla 14

Lista Inicial de requerimientos

ID	REQUERIMIENTO	DESCRIPCIÓN
R01	Gestionar Socios	Registrar, modificar, dar de baja, listar los socios pertenecientes a la empresa.
R02	Gestionar Unidades	Registrar, modificar, dar de baja y listar las unidades que laboran en la empresa.
R03	Gestionar Rutas	Registrar, modificar, dar de baja y listar las rutas pertenecientes a la empresa.
R04	Gestionar Puntos de control	Registrar, modificar, dar de baja y listar los puntos donde se controlarán los tiempos de recorrido de cada ruta.
R05	Gestionar Tiempos establecidos	Registrar, modificar, dar de baja y listar los tiempos de recorrido de cada ruta.
R06	Gestionar Usuarios	Registrar, modificar, dar de baja los usuarios que manejaran el sistema.
R07	Gestionar Personal	Registrar, modificar, dar de baja y listar los trabajadores de la empresa.
R08	Autenticar Usuarios	Autenticar usuario y dar acceso a las funcionalidades correspondientes.
R09	Gestionar Tarjetas	Registrar, modificar, eliminar y listar las tarjetas diarias para cada unidad.
R10	Asignar ruta	Asignar la ruta, hora de inicio y número de vuelta a cada unidad según corresponda.
R11	Reporte de recorrido diario	Visualizar el listado de unidades, sus recorridos y tiempos por día
R12	Registrar Sanción	Registrar sanciones por retraso en el recorrido de una unidad.
R13	Registrar Justificación	Registrar la justificación por retraso en el recorrido de una unidad.
R14	Gestionar Dispositivo GPS	Registrar, modificar, eliminar y listar dispositivos GPS con su unidad correspondiente.

R15	Visualizar de recorrido por unidad	Visualizar el recorrido de la unidad de acuerdo al número de vuelta.
R16	Reporte de unidades laborando.	Generar e imprimir reporte de las unidades que trabajaron cierta fecha.
R17	Recorrido de unidad	Visualizar en tiempo real el recorrido de la unidad seleccionada.
R18	Reporte de rutas y unidades laborando	Visualizar la cantidad de unidades laborando por ruta según hora y fecha indicada.
R19	Reporte promedio de recorrido	Visualizar el promedio de recorrido de cada ruta por día, semanas y meses.
R20	Comparación de recorridos	Visualizar el recorrido original de la ruta y el recorrido real de la unidad según el número de vuelta.
R21	Reporte de retrasos de unidades	Visualizar e imprimir el listado de retrasos de una unidad.
R22	Asignar pago de tarjeta	Asignar el pago a una tarjeta.

Fuente: Elaboración Propia.

3.1.5 Requerimientos no funcionales.

En este apartado se especifican los requerimientos no funcionales para el desarrollo y puesta en marcha del sistema web.

Tabla 15

Lista de Requerimientos no Funcionales

ID	REQUERIMIENTO	DESCRIPCIÓN
RNF01	Fácil manejo de los usuarios con el sistema	El sistema debe ser fácil de usar, con ayuda de interfaces intuitivas.
RNF02	Estabilidad del sistema cuando se procese grandes cantidades de información	El sistema deberá dar soporte a gran cantidad de información durante su proceso.
RNF03	Seguridad del acceso al sistema	El sistema permitirá a los usuarios el ingreso hacia las diversas funcionalidades, permitiendo el filtrado de datos de acuerdo al rol o perfil del usuario.
RNF04	Verificación de campos obligatorios	El sistema debe verificar que los datos correspondientes a los campos obligatorios de los formularios fueron ingresados por el usuario final.
RNF05	Verificación de campos numéricos	El sistema debe verificar que los campos exclusivamente numéricos no contengan caracteres de texto o caracteres especiales tales como comas, puntos, asteriscos.
RNF06	Verificación de reglas de negocio	El sistema debe hacer uso de estas reglas para validar la integridad de la información ingresada.
RNF07	Navegación en el sistema	El usuario solo podrá navegar entre las páginas del sistema a través de las opciones que le presenta la aplicación y solo podrá acceder a aquellas autorizadas para el rol correspondiente.
RNF08	Manejo de contraseñas	Pantalla para autenticación de usuarios (login y password).

		Retroalimentación hacia el usuario si se producen errores durante el proceso de autenticación.
RNF09	Poseer un diseño responsivo	El sistema web debe tener diseño “Responsivo” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivo tableta y teléfonos inteligentes.
		Los requerimientos mínimos son:
RNF10	Software para el equipo cliente web	Java 8.
		Navegador Chrome 66.0 , Firefox Quantum
		Los requerimientos mínimos son:
RNF11	Hardware para el equipo cliente web	CPU CORE I3
		Memoria RAM 4GB
		Conexión a internet ADSL 256 Kbps.

Fuente: Elaboración Propia.

3.1.6 Definir usuarios del sistema.

En este apartado se definen y describen los siguientes usuarios para el sistema, todos estos intervienen en el desarrollo de las Historias de Usuario y serán los que obtendrán el beneficio real con el uso del sistema, a continuación se detalla desde la Tabla 16 a la Tabla 18.

Tabla 16

Gerente

USUARIO	GERENTE
Descripción	Encargado de asignar los usuarios del sistema, y visualizar todos los reportes del sistema. Listado de socios. Listado de unidades. Visualizar recorridos.
Requerimientos	Comparar de recorridos. Promedio de recorrido de rutas. Reporte de retrasos. Gestión de rutas.

Fuente: Elaboración Propia.

Tabla 17

Secretaria

USUARIO	SECRETARIA
Descripción	Encargada de registrar los datos básicos de la empresa como socios, unidades, rutas, tiempos, puntos de control, aportaciones diarias y generar reportes diarios.
Requerimientos	Gestión de socios.

	Gestión de unidades.
	Gestión de GPS.
	Gestión de rutas.
	.Gestión de puntos de control.
	Gestión de tiempos.
	Reporte de planilla de recorrido diario.
	Reporte de unidades laborando.

Fuente: Elaboración Propia.

Tabla 18

Controlador

USUARIO	CONTROLADOR
Descripción	Encargado de asignar las tarjetas y recorridos a cada unidad por vuelta, aplicar las sanciones y generar reportes diarios.
Requerimientos	Gestionar Tarjetas. Gestionar Sanciones. Reporte de retrasos. Registrar justificación. Reporte de posición actual. Comparación de recorridos.

Fuente: Elaboración Propia.

3.1.7 Product Backlog o Pila del Producto

En esta fase se llevó a cabo la priorización de la lista de requerimientos funcionales del producto, a esta lista la llamamos Product Backlog.

Considerando las siguientes prioridades:

- Alta.
- Media.
- Baja.

A continuación se muestra la pila del producto en la Tabla 19

Tabla 19

Product Backlog

ID	REQUERIMIENTO	DESCRIPCIÓN	USUARIO	PRIORIDAD
R01-01	Registrar Socio	Registrar a todos los socios vigentes de la empresa	Secretaria	Alta
R01-02	Modificar Socio	Modificar los datos del socio para rectificar o actualizar datos.	Secretaria	Alta
R01-03	Dar de baja Socio	Dar de baja al socio por retiro de la empresa.	Secretaria	Alta
R01-04	Visualizar Socio	Visualizar, imprimir los datos personales de cada socio.	Secretaria	Alta
R02-05	Registro de unidades	Registro de las unidades que laboran en la empresa.	Secretaria	Alta
R02-06	Modificar datos de unidad	Modificar datos de la unidad para rectificar o actualizar datos.	Secretaria	Alta
R02-07	Dar de baja unidad	Dar de baja a la unidad por retiro de la empresa.	Secretaria	Alta
R02-08	Visualizar Unidad	Visualizar datos de unidad	Secretaria	Alta
R03-09	Registrar rutas	Registrar todas las rutas vigentes de la empresa.	Secretaria	Alta

		Modificar datos de la		
R03-10	Modificar ruta	ruta para rectificar o actualizar.	Secretaria	Alta
R03-11	Dar de baja ruta	Dar de baja a la ruta por inactividad.	Secretaria	Alta
R03-12	Visualizar ruta	Visualizar el nombre y calles de la ruta.	Secretaria	Alta
R04-13	Registro puntos de control	Registrar todos los puntos de control vigentes.	Secretaria	Alta
R04-14	Modificar puntos de control	Modificar los datos de un punto de control.	Secretaria	Alta
R04-15	Dar de baja punto de control	Dar de baja al punto de control.	Secretaria	Alta
R04-16	Listar puntos de control	Visualizar los puntos de control vigente con sus datos respectivos	Secretaria	Alta
R04-17	Activar punto de control	Activar punto de control	Secretaria	Alta
R05-18	Registro de tiempos establecidos	Registrar los tiempos establecidos por ruta y punto de control	Secretaria	Alta.
R05-19	Modificar tiempos establecidos	Modificar los datos del tiempo establecido	Secretaria	Alta.
R05-20	Listar tiempos	Visualizar los tiempos establecidos con sus respectivos datos según ruta.	Secretaria	Alta.
R06-21	Crear usuarios	Generar usuarios para el personal que labora en la empresa.	Secretaria	Alta.

R06-22	Modificar Usuario	Modificar contraseña de usuario	Secretaria	Media
R06-23	Dar de baja a usuario	Dar de baja a un usuario	Gerente	Alta
R06-24	Listar usuario	Listar usuarios y sus cargos	Gerente	Media
R07-25	Registrar personal	Registro del personal que labora en la empresa	Secretaria	Alta
R07-26	Modificar personal	Modificar datos del personal	Secretaria	Alta
R07-27	Dar de baja personal	Dar de baja al personal que ya no labora en la empresa	Secretaria	Media.
R07-28	Visualizar personal	Visualizar, imprimir los datos del personal.	Secretaria	Media
R02-29	Justificación de no disponibilidad de la unidad	Registrar el motivo por el cual la unidad no labora	Secretaria	Media
R09-30	Asignación de tarjeta a unidad	Asignar una tarjeta a cada unidad que labora por día.	Controlador	Alta
R09-31	Dar de baja tarjeta	Dar de baja a una tarjeta	Controlador	Media
R09-32	Visualizar tarjeta	Visualizar los datos de una tarjeta	Controlador	Alta
R10-33	Asignar recorrido a unidad	Asignar el recorrido y número de vuelto a una unidad	Controlador	Alta

	Dar de baja a	Dar de baja a un		
R10-34	recorrido asignada a unidad	recorrido asignado a una unidad	Controlador	Alta
R12-35	Registrar sanción	Registrar sanción de unidad por retraso.	Controlador	Media
R12-36	Aplicar sanción a unidad	Aplicar sanción a unidad por retraso.	Controlador	Media
R13-37	Registrar justificación	Registrar justificación de unidad por retraso	Controlador	Media.
R15-39	Visualizar de recorrido por unidad	Visualizar el recorrido de una unidad por número de vuelta	Controlador	Media.
R18-39	Reporte de unidades trabajando por fecha	Visualizar las unidades laborando por día	Gerente	Media
R12-40	Listar sanciones por unidad	Listar las sanciones registradas por unidad	Gerente	Media
R21-41	Reporte de retrasos de unidades	Listar los retrasos por unidad	Gerente	Baja
R11-42	Reporte de recorrido diario	Visualizar el listado de unidades, sus recorridos y tiempos por día	Gerente	Baja
R14-43	Registro GPS	Registrar GPS a su respectiva unidad	Secretaria	Media
R20-44	Comparación de recorridos	Visualizar el recorrido original de la ruta y el recorrido real de la unidad según el número de vuelta.	Controlador	Media
R22-45	Pago de unidad	Asignar un pago a la tarjeta de una unidad	Gerente	Baja

R19-46	Reporte promedio de recorrido	Visualizar el promedio de recorrido de cada ruta por día, semanas y meses.	Gerente	Media
R14-47	Listar GPS	Listar los datos de un GPS con su respectiva unidad	Secretaria	Media
R08-48	Autenticar Usuarios	Autenticar usuario y dar acceso a las funcionalidades correspondientes.	Gerente	Baja

Fuente: Elaboración Propia.

Las historias de usuario no fueron escritas por los clientes, pero sí definieron su contenido en la redacción y eso debido a que no tenían conocimiento de cómo elaborarlas, pero finalmente no se vieron afectadas y se mantuvieron clara en términos del cliente en su contenido.

3.1.8 Plan de versiones.

El plan de versiones o también llamado plan de entregas presenta un conjunto de requerimientos agrupados por sprint y release. Cada sprint tendrá la duración de 4 semanas, a continuación, se detalle el plan de versiones en la Tabla 20.

Tabla 20

Plan de versiones

SPRINT	REQUERIMIENTO	PRIORIDAD	RELEASE
SPRINT 01	Registrar socio	ALTA	
	Modificar socio	ALTA	
	Dar de baja socio	ALTA	
	Visualizar socio	ALTA	
	Registro de unidades	ALTA	
	Modificar unidad	ALTA	
	Dar de baja a unidad	ALTA	
	Visualizar unidad	ALTA	
	Registrar ruta	ALTA	
	Modificar ruta	ALTA	
	Dar de baja ruta	ALTA	
	Visualizar ruta	ALTA	
	Registro puntos de control	ALTA	Versión 1.0
	Modificar puntos de control	ALTA	
	Listar puntos de control	ALTA	
	Activar punto de control	ALTA	
Dar de baja punto de control	ALTA		
Registro de tiempos establecidos	ALTA		
Modificar tiempos establecidos	ALTA		
Listar tiempos	ALTA		
SPRINT 02	Crear usuarios	ALTA	
	Modificar Usuario	ALTA	
	Dar de baja a usuario	ALTA	
	Listar usuario	MEDIA	
	Registrar personal	ALTA	
	Modificar personal	ALTA	

	Dar de baja personal	MEDIA	
	Visualizar personal	MEDIA	
	Justificación de no disponibilidad de la unidad	MEDIA	
	Asignación de tarjeta a unidad	ALTA	
SPRINT 03	Dar de baja tarjeta	MEDIA	
	Visualizar tarjeta	ALTA	
	Asignar recorrido a unidad	ALTA	
	Dar de baja a recorrido asignada a unidad	ALTA	
	Registrar sanción	Media	
	Aplicar sanción a unidad	Media	
	Registrar justificación	Media.	
	Visualizar de recorrido por unidad	Media.	Versión 2.0
SPRINT 04	Reporte de unidades trabajando por fecha	Media	
	Listar sanciones por unidad	Media	
	Reporte de retrasos de unidades	Baja	
	Reporte de recorrido diario	Baja	
	Registro GPS	Media	
	Comparación de recorridos	Media	
SPRINT 05	Pago de unidad	Baja	
	Reporte promedio de recorrido	Media	
	Listar GPS	Media	
	Autenticar Usuarios	Baja	

Fuente: Elaboración Propia.

3.1.9 Organización del equipo de trabajo.

En este apartado se describe el equipo de trabajo conformado por el dueño del producto o cliente, maestro SCRUM y miembros del equipo, quienes son responsables de llevar a cabo este proyecto hasta su finalización y entrega

El equipo para este proyecto fue conformado de la siguiente manera:

- **Dueño del producto o cliente:** Sr. Montenegro Barboza Agustín.
- **Maestro SCRUM:** Ing. Martín Leiva Castillo.
- **Miembros del equipo SCRUM:**
 - Gabriel Leonardo Solis Vera.
 - Joel Alfredo Zamora Granados.

Siendo los comprometidos: el dueño del producto o cliente, maestro SCRUM y miembros del equipo. Y los implicados, la secretaria, junta directiva y controladores.

3.1.10 Entorno de trabajo.

Dirigirse al Apéndice D donde indicaremos la instalación y configuración de las herramientas usadas a lo largo del proyecto.

3.2 Sprint N° 01

3.2.1 Planificación.

Al inicio de cada Sprint el equipo SCRUM llevó a cabo una reunión con la finalidad de poder definir las historias de usuarias correspondientes al Sprint Backlog obtenidas del Product Backlog ya definido, con la finalidad de establecer los tiempos de desarrollo, el diseño de prototipos, delegar los encargados de su desarrollo y estimar las historias de usuario. La estimación de cada historia de usuario se realizó por medio de la técnica Planning Pocker con la siguiente serie de números cada tarjeta 0, 1, 2, 3, 5, 8, 13, 20, 40, 100, se tuvo en cuenta los prototipos para reconocer la complejidad de la historia y el posible tiempo de desarrollo.

Para este Sprint se consideraron las HU que muestra la tabla 21.

Tabla 21

Sprint Backlog – Sprint N° 01

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-001	Registrar socio	Alta	20
HU-002	Modificar socio	Alta	08
HU-003	Dar de baja socio	Alta	08
HU-004	Visualizar socio	Alta	20
HU-005	Registro de unidades	Alta	20
HU-006	Modificar unidad	Alta	8
HU-007	Dar de baja a unidad	Alta	5
HU-008	Visualizar Unidad	Alta	20
HU-009	Registrar ruta	Alta	8
HU-010	Modificar ruta	Alta	8
HU-011	Dar de baja ruta	Alta	8
HU-012	Visualizar ruta	Alta	20
HU-013	Registros de punto de control	Alta	40
HU-014	Modificar punto de control	Alta	20
HU-015	Activar punto de control	Alta	8
HU-016	Listar punto de control	Alta	8
Total, puntos estimados			229

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 22 hasta la Tabla 37

Tabla 22

HU: Registrar socio

Desarrollo de la historia de usuario (Ver apéndice A).

HISTORIA DE USUARIO			
IDENTIFICADOR:	001		
NOMBRE:	Registrar socio		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar a los socios de la empresa.</p> <p>Para: llevar un control adecuado de ellos.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registro de socio.			
<p>Dado: que el usuario quiera registrar un DNI de un socio ya registrado.</p> <p>Cuando: el usuario de clic en el botón “Registrar socio”.</p> <p>Entonces: La aplicación mostrará un mensaje de error de DNI.</p>			

Escenario 2: Edad de socio

Dado: que el usuario ingrese una fecha de nacimiento y la edad sea menor de 18 años.

Cuando: este de clic en el botón “Registrar socio”.

Entonces: se mostrará un mensaje “El socio debe ser mayor de edad”.

Registro Socios

Nombres:
Ingrese nombres del socio

Apellido Paterno:
Ingrese apellido paterno del socio

Apellido Materno:
Ingrese apellido materno del socio

DNI:
Ingrese DNI del socio

Fecha de Nacimiento:
dd/mm/aaaa

Sexo:
 Masculino Femenino

Fecha de Registro:
dd/mm/aaaa

Estado Civil:
Casado

Dirección:
Ingrese dirección del socio

Correo:
Ingrese correo del socio

Celular:
Ingrese celular del socio

Datos Referencia

Nombres y apellidos:
Ingrese nombres y apellidos completos del pariente

Celular:
Ingrese celular del pariente

Registrar Socio

Figura 1 Prototipo del Sprint 01 – HU: Registrar Socio

Fuente. Elaboración Propia

Tabla 23

HU: Modificar socio

HISTORIA DE USUARIO			
IDENTIFICADOR:	002		
NOMBRE:	Modificar socio		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	08	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Modificar a los socios de la empresa.</p> <p>Para: poder corregir los datos errados de los socios.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Modificar atributos de tablas de base de datos	1 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Modificar fecha de nacimiento			
<p>Dado: que el usuario quiera modificar la fecha de nacimiento y la edad sea menor a 18 años</p> <p>Cuando: el usuario seleccione la opción para modificar al socio.</p> <p>Entonces: La aplicación mostrará un mensaje de alerta indicando que el socio debe ser mayor de edad.</p>			
Escenario 2: Modificar DNI del socio			

<p>Dado: que el usuario quiera modificar el DNI del socio</p> <p>Cuando: el usuario quiera ingresar al campo DNI.</p> <p>Entonces: La aplicación no le permitirá modificar el número de DNI del socio.</p>
<p>Escenario 3: Socios no registrados en el sistema</p>
<p>Dado: que el usuario quiera listar a los socios de la empresa y no haya socios registrados.</p> <p>Cuando: ingrese el filtro correspondiente y de clic en el botón “Listar”</p> <p>Entonces: el sistema mostrara un mensaje “No hay socios registrados actualmente”</p>
<p>Escenario 4: Búsqueda del socio</p>
<p>Dado: que el usuario quiera buscar un socio por apellido.</p> <p>Cuando: ingrese el “apellido” del socio en el campo correspondiente y de clic en el botón “Listar”</p> <p>Entonces: el sistema realizara la búsqueda por similitud de apellidos y mostrara a todos los socios con el mismo apellido.</p>



Figura 2 Prototipo del Sprint 01 – HU: Modificar Socio
Fuente. Elaboración Propia

Modificar Socio

Nombres:
Jose Luis

Apellido Paterno:
Vera

Apellido Materno:
Sandoval

DNI:
75896532

Fecha de Nacimiento:
01/01/1980

Sexo:
 Masculino Femenino

Fecha de Registro:
24/03/2018

Estado Civil:
Casado

Dirección:
Chiclayo-La victoria

Correo:
jose_luis@gmail.com

Celular:
958635241

Datos Referencia

Nombres y apellidos:
Joel Vera Sandoval

Celular:
988856362

[Modificar Socio](#)

Figura 3 Prototipo del Sprint 01 – HU: Modificar Socio
Fuente. Elaboración Propia

Tabla 24

HU: Dar de baja socio

HISTORIA DE USUARIO			
IDENTIFICADOR:	003		
NOMBRE:	Dar de baja socio		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	08	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja a los socios de la empresa.</p> <p>Para: poder retirar a los socios que ya no son parte de la empresa.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir interfaz		2 h	
Modificar atributo de tablas de base de datos		1/2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Dar de baja con unidades trabajando			
<p>Dado: que el usuario quiera dar de baja a un socio con unidades trabajando</p> <p>Cuando: el usuario seleccione la opción correspondiente.</p>			

<p>Entonces: La aplicación mostrará un mensaje de alerta indicando que no se puede dar de baja al socio con unidades trabajando.</p>
<p>Escenario 2: Dar de baja a un socio</p>
<p>Dado: que el usuario quiera dar de baja a un socio</p> <p>Cuando: el usuario seleccione el botón “Desactivar”.</p> <p>Entonces: La aplicación pasara al socio a estado “Inactivo” y el botón mostrara la opción de “Activar”</p>
<p>Escenario 3: Activar a socio dado de baja anteriormente</p>
<p>Dado: que el usuario quiera activar a un socio</p> <p>Cuando: el usuario seleccione el botón “Activar”.</p> <p>Entonces: La aplicación pasara al socio a estado “Activo” y el botón mostrara la opción de “Desactivar”.</p>



Figura 4 Prototipo del Sprint 01 – HU: Dar de baja Socio
Fuente. Elaboración Propia

Tabla 25

HU: Visualizar socio

HISTORIA DE USUARIO			
IDENTIFICADOR:	004		
NOMBRE:	Visualizar socio		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de un socio</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	4 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar Datos			
<p>Dado: que el usuario quiera visualizar los datos de un socio</p> <p>Cuando: el usuario de clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con todos los datos del socio</p>			
Escenario 2: Imprimir datos del socio			
<p>Dado: Que el usuario quiera imprimir los datos de un socio.</p> <p>Cuando: el usuario de clic en el botón “Imprimir”</p> <p>Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos del socio.</p>			

Listado de Socios

Filtro de socios: Imprimir

Buscar socio por apellido: Listar

Número	Nombres y Apellidos	DNI	Numero de acciones	Estado	Opciones
1	Jose Luis Vera Sandoval	78596541	24	Activo	Ver Modificar Inactivar
2	Maria Gutierrez Sanchez	71236548	46	Inactivo	Ver Modificar Activar
3	Gabriel Solis Vera	79865425	24	Activo	Ver Modificar Inactivar

Figura 6 Prototipo del Sprint 01 – HU: Visualizar Socio
Fuente. Elaboración Propia

Datos del socio

Nombres y apellidos:

DNI:

Fecha de Nacimiento:

Sexo:

Dirección:

Estado Civil:

Estado:

Fecha de Registro:

Fecha de Retiro:

Celular:

Correo:

Numero de acciones:

Pariente:

Celular de pariente:

Imprimir Aceptar

Figura 5 Prototipo del Sprint 01 – HU: Visualizar Socio
Fuente. Elaboración Propia

Tabla 26

HU: Registro de unidades por socio

HISTORIA DE USUARIO			
IDENTIFICADOR:	005		
NOMBRE:	Registro de unidades por socio		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar las unidades de cada socio de la empresa</p> <p>Para: llevar un mejor control de las unidades de la empresa</p>			
OBSERVACIONES			
Luego de que las unidades del socio, se registren las acciones del socio aumentaran en 23 acciones.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Unidades para un solo socio			
<p>Dado: que el usuario quiera registrar una unidad que pertenece ya a un socio</p> <p>Cuando: el usuario intente registrarlo</p> <p>Entonces: el sistema mostrará un mensaje “La placa ya ha sido registrada”.</p>			
Escenario 2: Unidades para un socio no registrado			

Dado: Que el usuario quiera registrar una unidad de un socio que no está registrado en el sistema.

Cuando: el usuario intente buscar al socio.

Entonces: El sistema solo mostrara un mensaje indicando que no se encuentra el socio y solo mostrara los socios registrados.

Registro de unidades

Buscar Socio por Apellido

Socio

▼

Placa

Marca

Modelo

Fecha de Registro:

Color

Capacidad

Año de Fabricación

Tipo de unidad

▼

Registrar Unidad

Figura 7 Prototipo del Sprint 01 – HU: Registro de unidades por Fuente. Elaboración Propia

Tabla 27

HU: Modificar datos de unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	006		
NOMBRE:	Modificar datos de unidad		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Modificar las unidades de cada socio de la empresa</p> <p>Para: realizar ajustes en los datos de las unidades de la empresa</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributos de tablas de base de datos		1 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Actualizar o corregir datos			
<p>Dado: que el usuario quiera actualizar o corregir un dato ya registrado de una unidad.</p> <p>Cuando: el usuario seleccione la opción correspondiente</p> <p>Entonces: se visualizarán los datos de la unidad con la opción de poder modificarlos.</p>			
Escenario 2: Unidades para un socio no registrado			

Dado: Que el usuario quiera registrar una unidad de un socio que no está registrado en el sistema.

Cuando: el usuario intente buscar al socio.

Entonces: El sistema solo mostrara un mensaje indicando que no se encuentra el socio y solo mostrara los socios registrados.

Escenario 3: Unidades no registradas

Dado: que el usuario quiera listar las unidades de la empresa y no haya unidades registradas

Cuando: ingrese el filtro correspondiente y seleccione el botón “Listar”

Entonces: el sistema mostrara un mensaje “No hay unidades registradas actualmente”

Escenario 4: Socio no registrado

Dado: que el usuario quiera listar las unidades de un socio e ingrese un socio no registrado

Cuando: ingrese el “apellido” del socio en el campo correspondiente y de clic en el botón “Listar”

Entonces: el sistema mostrara un mensaje “Socio no registrado”

Número	Socio	Placa	Estado	Opciones
1	Jose luis Vera Sandoval	MPL-000, MPL-001	Activo	Vizualizar datos
2	Maria Gutierrez Sanchez	ABC-124	Inactivo	Vizualizar datos
3	Gabriel Solis Vera	QWE-123,QWE-234	Activo	Vizualizar datos

Figura 8 Prototipo del Sprint 01 – HU: Modificar datos de unidad

Fuente. Elaboración Propia

Datos del socio y sus vehiculos

Socio:

Vehiculo 1:

Vehiculo 2:

Figura 9 Prototipo del Sprint 01 – HU: Modificar datos de unidad
Fuente. Elaboración Propia

Modificar Unidad

Buscar Socio por Apellido

Socio

▼

Placa

Marca

Modelo

Fecha de Registro:

Color

Capacidad

Año de Fabricación

Tipo de unidad

▼

Figura 10 Prototipo del Sprint 01 – HU: Modificar datos de unidad
Fuente. Elaboración Propia

Tabla 28

HU: Dar de baja unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	007		
NOMBRE:	Dar de baja unidad		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	5	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: dar de baja las unidades de la empresa</p> <p>Para: poder retirar del sistema las unidades que ya no están activas.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		1/2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: única unidad del socio			
<p>Dado: que el usuario quiera dar de baja a la única unidad asignada a un socio</p> <p>Cuando: el usuario seleccione la opción correspondiente</p> <p>Entonces: el sistema mostrara un mensaje “No se puede dar de baja a esta unidad por ser la única unidad del socio”</p>			

Escenario 2: Dar de baja a una unidad
<p>Dado: que el usuario quiera dar de baja a una unidad</p> <p>Cuando: el usuario seleccione el botón “Desactivar”.</p> <p>Entonces: La aplicación pasara a la unidad a estado “Inactivo” y el botón mostrara la opción de “Activar”</p>
Escenario 3: Activar a una unidad dado de baja anteriormente
<p>Dado: que el usuario quiera activar a una unidad</p> <p>Cuando: el usuario seleccione el botón “Activar”.</p> <p>Entonces: La aplicación pasara a la unidad a estado “Activo” y el botón mostrara la opción de “Desactivar”.</p>

Listado de Unidades

Filtro de unidades:

Buscar unidad por apellido del socio:

Número	Socio	Placa	Estado	Opciones
1	Jose Luis Vera Sandoval	MPL-000, MPL-001	Activo	<input type="button" value="Vizualizar datos"/>
2	Maria Gutierrez Sanchez	ABC-124	Inactivo	<input type="button" value="Vizualizar datos"/>
3	Gabriel Solis Vera	QWE-123, QWE-234	Activo	<input type="button" value="Vizualizar datos"/>

Figura 11 Prototipo del Sprint 01 – HU: Dar de baja unidad

Fuente. Elaboración Propia

Datos del socio y sus vehiculos

Socio:

Vehiculo 1:

Vehiculo 2:

Figura 12 Prototipo del Sprint 01 – HU: Dar de baja unidad
Fuente. Elaboración Propia

Tabla 29

HU: Visualizar unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	008		
NOMBRE:	Visualizar unidad		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de la unidad de un socio</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar Datos			
<p>Dado: que el usuario quiera visualizar los datos de la unidad de un socio</p> <p>Cuando: el usuario de clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con todos los datos de la unidad</p>			
Escenario 2: Imprimir datos de la unidad			

Dado: Que el usuario quiera imprimir los datos de la unidad de un socio.

Cuando: el usuario de clic en el botón “Imprimir”

Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos del socio.

Listado de Unidades

Filtro de unidades:
Todos

Buscar unidad por apellido del socio:
Ingrese apellido del socio

Número	Socio	Placa	Estado	Opciones
1	Jose Luis Vera Sandoval	MPL-000, MPL-001	Activo	<input type="button" value="Vizualizar datos"/>
2	Maria Gutierrez Sanchez	ABC-124	Inactivo	<input type="button" value="Vizualizar datos"/>
3	Gabriel Solis Vera	QWE-123,QWE-234	Activo	<input type="button" value="Vizualizar datos"/>

Figura 13 Prototipo del Sprint 01 – HU: Visualizar unidad
Fuente. Elaboración Propia

Datos del socio y sus vehiculos

Socio:

Vehiculo 1:

Vehiculo 2:

Figura 14 Prototipo del Sprint 01 – HU: Visualizar unidad
Fuente. Elaboración Propia

Datos del vehiculo

Socio:

Placa:

Marca:

Modelo:

Fecha de registro:

Color:

Capacidad:

Año de fabricación:

Tipo:

Figura 15 Prototipo del Sprint 01 – HU: Visualizar unidad

Fuente. Elaboración Propia

Tabla 30

HU: Registro de ruta

HISTORIA DE USUARIO			
IDENTIFICADOR:	009		
NOMBRE:	Registro de ruta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			

<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: registrar las rutas que recorren las unidades de la empresa.</p> <p>Para: llevar un control de cuál es el recorrido que genera cada ruta.</p>	
OBSERVACIONES	
Ninguna.	
TAREAS	
Descripción	Duración (horas)
Diseñar la interfaz	2 h
Diseñar y crear tablas de la base de datos	2 h
Construir la interfaz	2 h
Crear procedimientos almacenados en la base de datos	2 h
Crear el código correspondiente	8 h
Realizar Pruebas y Corregir Errores	4 h
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Ruta ya registrada	
<p>Dado: que el usuario quiera registrar una ruta ya registrada</p> <p>Cuando: de clic en el botón “Agregar Ruta”</p> <p>Entonces: se mostrará un mensaje diciendo “Ruta ya está registrada”</p>	
Escenario 2: Agregar calles	
<p>Dado: que el usuario quiera agregar las calles de una ruta</p> <p>Cuando: de clic en la ruta que se quiera agregar la calle y se dé anticlic en la calle deseada dentro del mapa.</p> <p>Entonces: se mostrará un punto de referencia de la calle con las opciones: “Agregar calle”, “Visualizar punto” y “Retirar punto de referencia”</p>	



Figura 16 Prototipo del Sprint 01 – HU: Registro de Ruta
Fuente. Elaboración Propia



Figura 17 Prototipo del Sprint 01 – HU: Registro de Ruta
Fuente. Elaboración Propia

Tabla 31

HU: Modificar ruta

HISTORIA DE USUARIO			
IDENTIFICADOR:	010		
NOMBRE:	Modificar ruta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: reorganizar el recorrido de las unidades de la empresa.</p> <p>Para: actualizar una ruta y brindar un mejor servicio al cliente.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		1 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Actualizar o corregir ruta			
<p>Dado: que el usuario quiera actualizar o corregir una ruta ya registrada</p> <p>Cuando: el usuario seleccione el botón “Modificar”</p> <p>Entonces: se visualizará los datos de la ruta con la opción de poder modificarlos.</p>			
Escenario 2: Ruta no registrada			

Dado: que no haya rutas registradas según filtro seleccionado

Cuando: se dé clic en el botón “Listar”

Entonces: se mostrará el mensaje: “No hay rutas registradas”

Listado de Rutas

Filtro de Rutas:

Número	Ruta	Calles	Tiempo de recorrido (min)	Estado	Opciones
1	A	Av Augusto B. Leguía 10, Chiclayo 14001, Perú / Av Augusto B, Leguía 1689, Chiclayo 14009, Perú	163	Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar baja"/>
2	B	Amaru Inca Yupanqui 989-1029 - Los Andes / Panamericana Nte. 1296 - Inca Yupanqui	62	Sin Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar alta"/>
3	C	Union 1520, La Victoria, Perú / Av. Grau 1643, La Victoria, Perú - Unión	22	Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar baja"/>

Figura 18 Prototipo del Sprint 01 – HU: Modificar Ruta

Fuente. Elaboración Propia

Datos de ruta

Letra de ruta:

Estado de ruta:

Figura 19 Prototipo del Sprint 01 – HU: Modificar Ruta

Fuente. Elaboración Propia

Tabla 32

HU: Dar de baja ruta

HISTORIA DE USUARIO			
IDENTIFICADOR:	011		
NOMBRE:	Dar de baja ruta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja rutas</p> <p>Para: poder retirar del sistema las rutas que ya no se le brindara servicio.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		1/2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Dar de baja por equivocación			
<p>Dado: que se intente dar de baja una ruta que no es la deseada.</p> <p>Cuando: se dé clic en el botón “Dar de baja ruta”</p> <p>Entonces: se mostrara el mensaje de confirmación: “Desea eliminar esta ruta”</p>			
Escenario 2: Confirmación para dar de baja a una ruta			

Dado: que el usuario confirme dar de baja a una ruta

Cuando: el usuario seleccione el botón “Desactivar”.

Entonces: La aplicación pasara la ruta a estado “Sin servicio”

Escenario 3: Activar ruta dada de baja anteriormente

Dado: que el usuario quiera activar una ruta para volver a brindarle servicio

Cuando: el usuario seleccione el botón “Activar”.

Entonces: La aplicación pasara la ruta a estado “Activo”.

Número	Ruta	Calles	Tiempo de recorrido (min)	Estado	Opciones
1	A	Av Augusto B. Leguía 10, Chiclayo 14001, Perú / Av Augusto B. Leguía 1689, Chiclayo 14009, Perú	163	Servicio	Ver Modificar Dar baja
2	B	Amaru Inca Yupanqui 989-1029 - Los Andes / Panamericana Nte. 1296 - Inca Yupanqui	62	Sin Servicio	Ver Modificar Dar alta
3	C	Union 1520, La Victoria, Perú / Av. Grau 1643, La Victoria, Perú - Unión	22	Servicio	Ver Modificar Dar baja

*Figura 20 Prototipo del Sprint 01 – HU: Dar de baja ruta
Fuente. Elaboración Propia*

Tabla 33

HU: Visualizar ruta

HISTORIA DE USUARIO			
IDENTIFICADOR:	012		
NOMBRE:	Visualizar ruta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de la unidad de un socio</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar Datos			
<p>Dado: que el usuario quiera visualizar los datos de una ruta</p> <p>Cuando: el usuario dé clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con la ruta y todas sus calles que recorrerá.</p>			
Escenario 2: Imprimir datos de ruta			

Dado: Que el usuario quiera imprimir los datos de una ruta.

Cuando: el usuario dé clic en el botón “Imprimir”

Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos de la ruta.

Listado de Rutas

Filtro de Rutas:

Todos

Número	Ruta	Calles	Tiempo de recorrido (min)	Estado	Opciones
1	A	Av Augusto B. Leguia 10, Chiclayo 14001, Perú / Av Augusto B. Leguia 1689, Chiclayo 14009, Perú	163	Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar baja"/>
2	B	Amanu Inca Yupanqui 989-1029 - Los Andes / Panamericana Nte. 1296 - Inca Yupanqui	62	Sin Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar alta"/>
3	C	Union 1520, La Victoria, Perú / Av. Grau 1643, La Victoria, Perú - Unión	22	Servicio	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Dar baja"/>

Figura 21 Prototipo del Sprint 01 – HU: Visualizar ruta

Fuente. Elaboración Propia

Datos de ruta

Ruta: Tiempo: Estado:

CALLES

Número	Calle
1	Av Augusto B. Leguia 10, Chiclayo 14001, Perú
2	Av Augusto B. Leguia 1689, Chiclayo 14009, Perú

Figura 22 Prototipo del Sprint 01 – HU: Visualizar ruta

Fuente. Elaboración Propia

Tabla 34

HU: Registro de puntos de control

HISTORIA DE USUARIO			
IDENTIFICADOR:	013		
NOMBRE:	Registro de puntos de control		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	40	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: registrar puntos de control</p> <p>Para: poder llevar una mejor administración de los puntos de control.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	2 h		
Construir la interfaz	2 h		
Investigar sobre geolocalización	6 h		
Investigar sobre API Google maps	8 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Punto de control fuera de chiclayo			
<p>Dado: que el usuario quiera registrar un punto de control fuera de la provincia de Chiclayo</p> <p>Cuando: el usuario seleccione la opción de registrar punto de control</p> <p>Entonces: La aplicación mostrará un mensaje de alerta “Punto de control fuera de rango”</p>			

Escenario 2: Registrar punto de control mediante clic derecho

Dado: que el usuario quiera registrar un punto de control

Cuando: de clic derecho en la ubicación dentro del mapa de Chiclayo

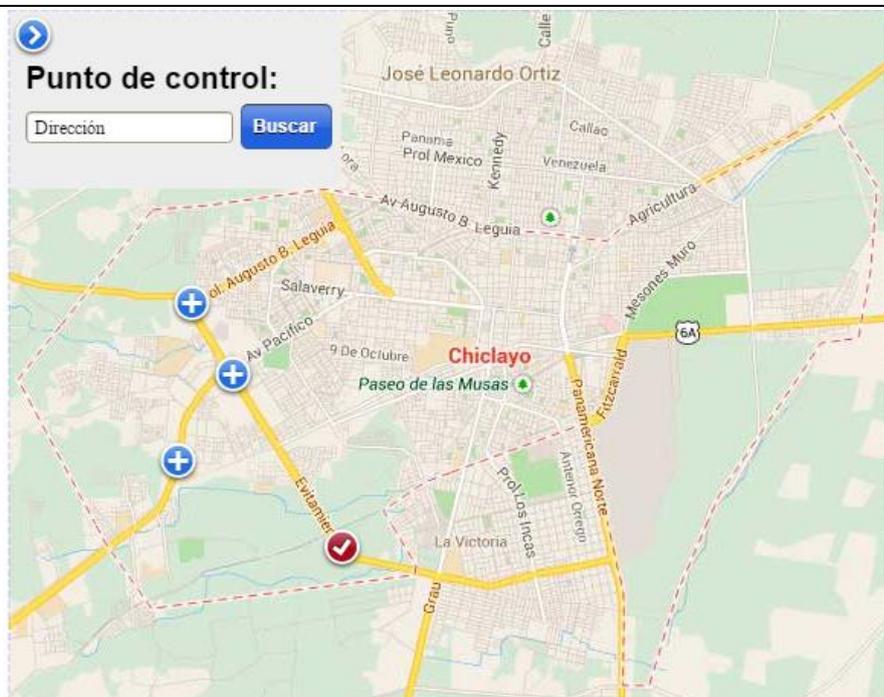
Entonces: La aplicación mostrará un modal para confirmar el registro del punto mostrando la dirección seleccionada.

Escenario 3: Registrar punto de control ingresando dirección

Dado: que el usuario quiera registrar un punto de control

Cuando: ingrese la dirección en el campo “Dirección” y de clic en el botón “Buscar”

Entonces: La aplicación mostrará un icono de color rojo en el punto buscado dando las opciones de “Agregar punto de control”, “Visualizar punto” y “eliminar búsqueda”



*Figura 23 Prototipo del Sprint 01 – HU: Registro de punto de control
Fuente. Elaboración Propia*



Figura 24 Prototipo del Sprint 01 – HU: Registro de punto de control
Fuente. Elaboración Propia



Figura 25 Prototipo del Sprint 01 – HU: Registro de punto de control
Fuente. Elaboración Propia

Tabla 35

HU: Modificar punto de control

HISTORIA DE USUARIO			
IDENTIFICADOR:	014		
NOMBRE:	Modificar punto de control		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Modificar los puntos de control</p> <p>Para: poder trasladarlos de un lugar a otro dado algún inconveniente en el lugar establecido inicialmente.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributos de tablas de base de datos		1 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Aceptar modificar punto de control			
<p>Dado: que el punto de control no haya sido modificado con anterioridad.</p> <p>Cuando: El usuario dé clic en el botón aceptar de la ventana modificar punto</p> <p>Entonces: El punto guardara las nuevas coordenadas como su nueva posición y guardaras las antiguas por si desea volver a la posición anterior.</p>			
Escenario 2: Retornar a ubicación anterior			

Dado: que el punto de control ya haya sido modificado con anterioridad.

Cuando: el usuario seleccione la opción volver a ubicación anterior

Entonces: La aplicación mostrará un mensaje de ¿Desea volver a la ubicación anterior? y mostrará la dirección de referencia y coordenadas antiguas.

Escenario 3: Modal de confirmación de retornar a ubicación anterior

Dado: que el punto de control no haya sido modificado con anterioridad.

Cuando: El usuario dé clic en el botón aceptar de la ventana modificar punto.

Entonces: El sistema mostrará el mensaje: "No existe historial de ubicación de este punto"



Figura 26 Prototipo del Sprint 01 – HU: Modificar punto de control
Fuente. Elaboración Propia

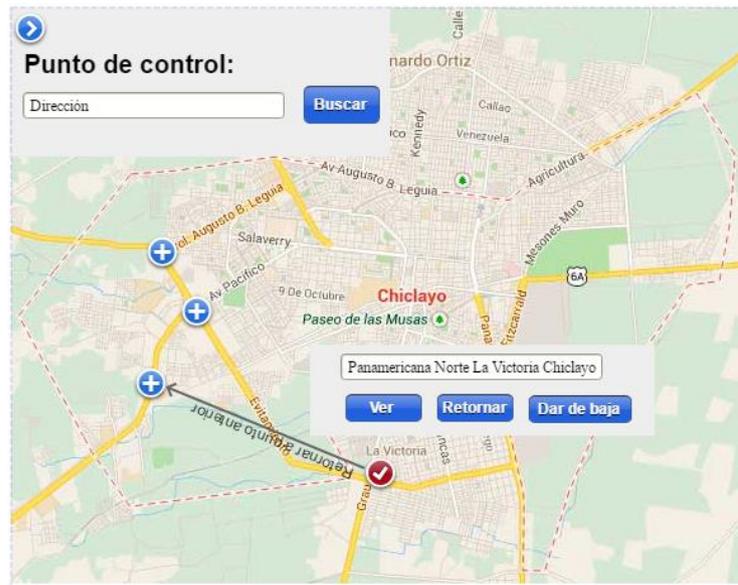


Figura 27 Prototipo del Sprint 01 – HU: Modificar punto de control
Fuente. Elaboración Propia

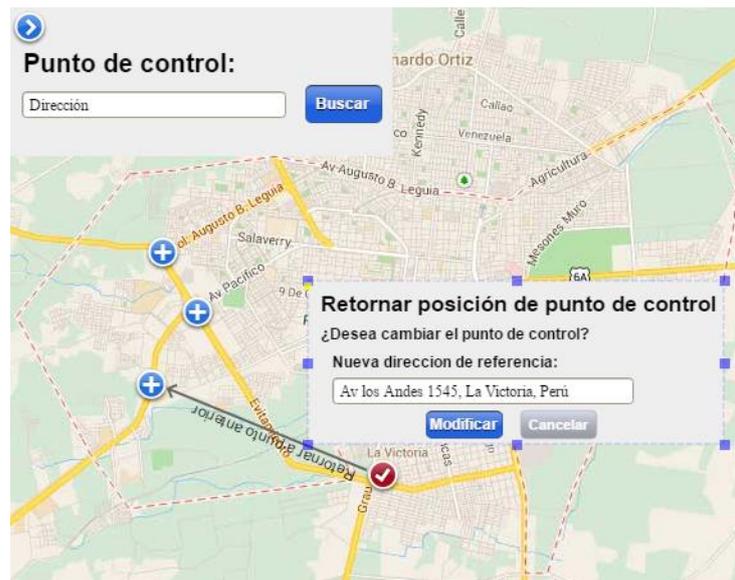


Figura 28 Prototipo del Sprint 01 – HU: Modificar punto de control
Fuente. Elaboración Propia

Tabla 36

HU: Activar punto de control

HISTORIA DE USUARIO			
IDENTIFICADOR:	015		
NOMBRE:	Activar punto de control		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: activar los puntos de control</p> <p>Para: poder ser utilizados de nuevo en alguna ruta.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		½ h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Mapa de puntos retirados			
<p>Dado: Que existan puntos retirados</p> <p>Cuando: el usuario de clic en la opción de puntos retirados.</p> <p>Entonces: El sistema mostrará un mapa con los puntos retirados en forma de marcadores.</p>			
Escenario 2: Activar Punto			
<p>Dado: que el usuario de clic en el botón activar.</p> <p>Cuando: la operación se haya realizado</p> <p>Entonces: el punto de control inactiva cambiara su estado y se borraría del mapa de puntos retirados.</p>			

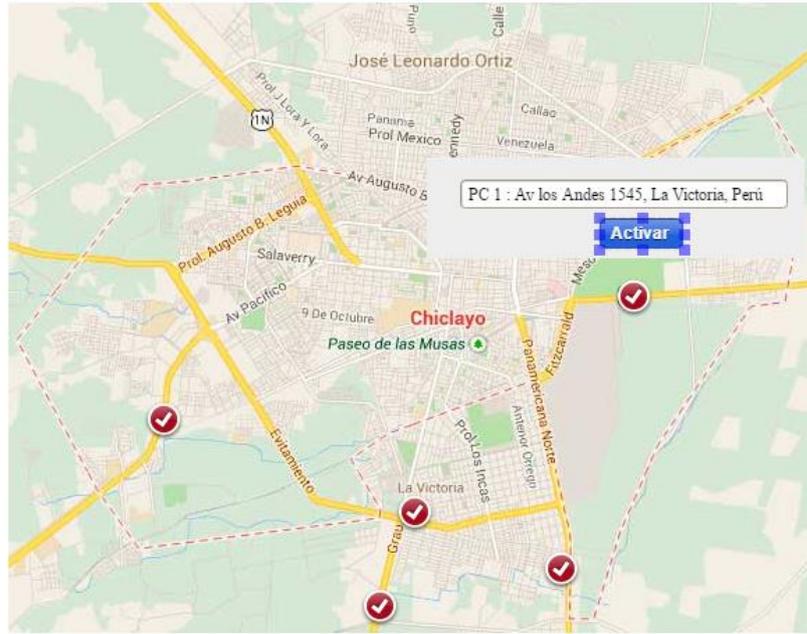


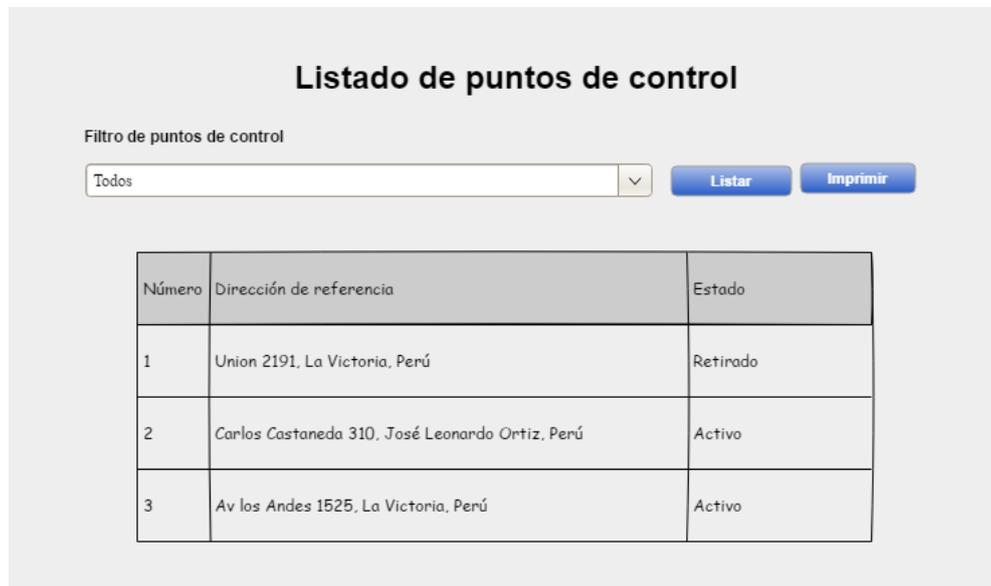
Figura 29 Prototipo del Sprint 01 – HU: Activar punto de control
Fuente. Elaboración Propia

Tabla 37

HU: Listar puntos de control

HISTORIA DE USUARIO			
IDENTIFICADOR:	016		
NOMBRE:	Listar puntos de control		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	01
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: visualizar los puntos de control registrados.</p> <p>Para: poder llevar un mejor manejo de ellos.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: No hay puntos de control registrados			
<p>Dado: que no hayan puntos de control registrados.</p> <p>Cuando: el usuario seleccione la opción de listar puntos de control.</p> <p>Entonces: La aplicación mostrará un mensaje de “No existen puntos de control registrados”</p>			
Escenario 2: Visualizar puntos de control por estado			
<p>Dado: que el usuario quiera visualizar solo socios por un determinado estado</p> <p>Cuando: el usuario seleccione algún filtro de los mencionados a continuación: “Activos”, “Retirados”, “Todos”.</p>			

Entonces: La aplicación mostrará solo los puntos de control según lo filtrado.



Listado de puntos de control

Filtro de puntos de control

Todos

Número	Dirección de referencia	Estado
1	Union 2191, La Victoria, Perú	Retinado
2	Carlos Castaneda 310, José Leonardo Ortiz, Perú	Activo
3	Av los Andes 1525, La Victoria, Perú	Activo

Figura 30 Prototipo del Sprint 01 – HU: Listar puntos de control
Fuente. Elaboración Propia

3.2.2 Tablero SCRUM.

El tablero SCRUM nos permite un orden de las historias de historias realizadas, las faltantes por sprint y el Product Backlog, además de poder visualizar alguna observación o cambio sugerido por el cliente y comunicarlo a los miembros del equipo, durante el desarrollo del sistema hemos utilizado la herramienta Trello. A continuación se muestra como estuvo diseñado el tablero.

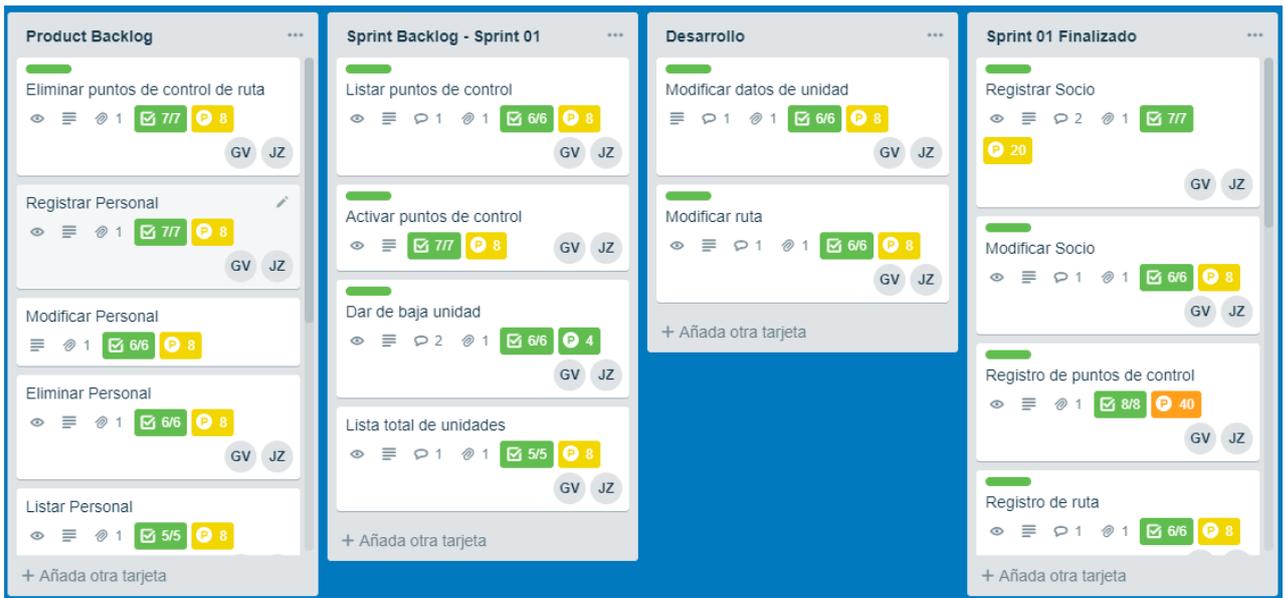


Figura 31 Tablero SCRUM del Sprint 01

Fuente. Elaboración Propia

3.2.3 Diagrama de Burndown del Sprint N°01.

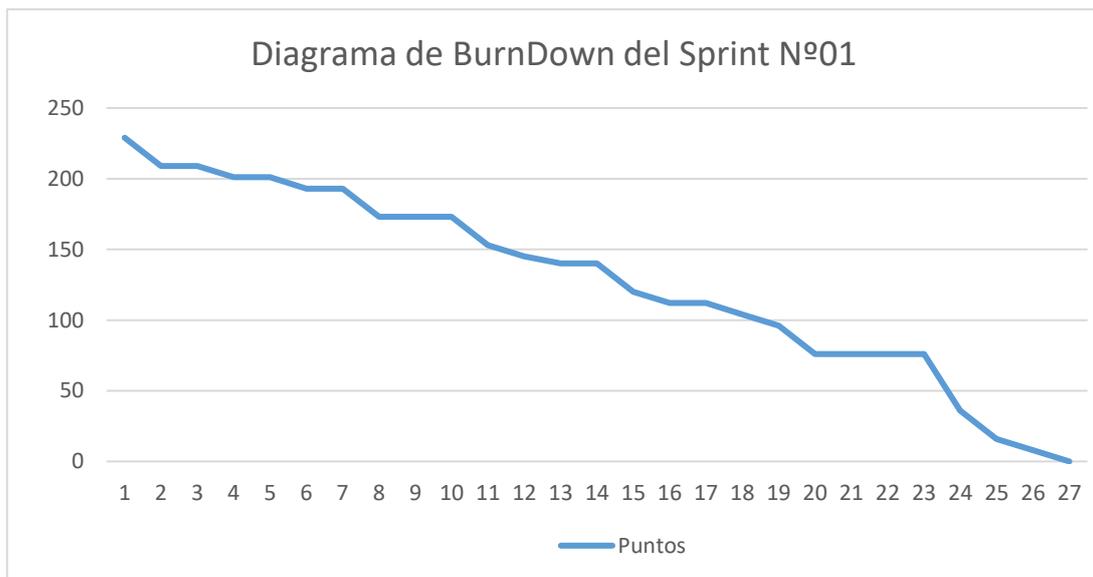


Figura 32 Diagrama de Burndown del Sprint 01

Fuente. Elaboración Propia

3.2.4 Retrospectiva.

En esta etapa se evaluaron tres aspectos importantes; buenos, malos y aspectos de mejora:

¿Qué podríamos seguir manteniendo?

- ✓ Se logró comunicación constante con los usuarios y se cumplieron las visitas quincenales.

- ✓ Buen ritmo de desarrollo por parte del equipo.
- ✓ Cumplimiento total de las historias de usuario.

¿Qué pudimos hacer de manera diferente?

- ✓ Ser un equipo equitativo con respecto a la teoría con lo práctico para no tener dificultades al implementar la nueva tecnología.
- ✓ Ser un equipo con más puntualidad en reuniones de equipo.
- ✓ Puntuación de historias de usuario más realistas.

¿Qué debemos mejorar?

- ✓ Analizar bien las funcionalidades antes de estimarlas, para ser más precisos en los tiempos estimados.
- ✓ Dedicar más tiempo a leer la documentación de la tecnología que usaremos en el sistema.
- ✓ Ser más puntuales con el equipo de trabajo cuando se acuerda una reunión.

Finalmente vemos en la Tabla 38, el cierre del proyecto para este Sprint.

Cierre del Proyecto con Scrum

Tabla 38

Cierre del Sprint N° 01

ID	HISTORIA DE USUARIO	RESPONSABLE	ESTADO	REVISIÓN
HU-001	Registrar socio	Solis Vera	Realizado	Conforme
		Gabriel Leonardo		
HU-002	Modificar socio	Solis Vera	Realizado	Conforme
		Gabriel Leonardo		
HU-003	Dar de baja socio	Solis Vera	Realizado	Conforme
		Gabriel Leonardo		

HU-004	Visualizar socio	Zamora Granados Joel	Realizado	Conforme
HU-005	Registro de unidades	Zamora Granados Joel	Realizado	Conforme
HU-006	Modificar unidad	Zamora Granados Joel	Realizado	Conforme
HU-007	Dar de baja a unidad	Zamora Granados Joel	Realizado	Conforme
HU-008	Visualizar Unidad	Zamora Granados Joel	Realizado	Conforme
HU-009	Registrar ruta	Zamora Granados Joel	Realizado	Conforme
HU-010	Modificar ruta	Zamora Granados Joel	Realizado	Conforme
HU-011	Dar de baja ruta	Zamora Granados Joel	Realizado	Conforme
HU-012	Visualizar ruta	Zamora Granados Joel	Realizado	Conforme
HU-013	Registros de punto de control	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-014	Modificar punto de control	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-015	Activar punto de control	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-016	Listar punto de control	Solis Vera Gabriel Leonardo	Realizado	Conforme

Fuente: Elaboración Propia.

3.3 Sprint N° 02

3.3.1 Planificación

Para este sprint se consideraron las historias de usuario que muestran la Tabla 39 y los prototipos

Tabla 39

Sprint backlog – Sprint N° 02

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-017	Dar de baja punto de control	Alta	08
HU-018	Registro de tiempos establecidos	Alta	20
HU-019	Modificar tiempos establecidos	Alta	08
HU-020	Listar tiempos	Alta	08
HU-021	Crear usuarios	Alta	20
HU-022	Modificar Usuario	Alta	13
HU-023	Dar de baja a usuario	Alta	5
HU-024	Listar usuario	Media	5
HU-025	Registrar personal	Alta	8
HU-026	Modificar personal	Alta	8
HU-027	Dar de baja personal	Media	8
HU-028	Visualizar personal	Media	20
Total, puntos estimados			131

Fuente: Elaboración Propia.

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 40 a la Tabla 50

Tabla 40

HU: Dar de baja puntos de control de ruta

HISTORIA DE USUARIO			
IDENTIFICADOR:	017		
NOMBRE:	Dar de baja puntos de control de ruta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja tiempos de recorrido a cada ruta por punto de control</p> <p>Para: retirar los puntos de control que ya no pertenecen a la ruta.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		3 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Dar de baja punto de control.			
<p>Dado: que el usuario desee dar de baja un punto de control de una ruta.</p> <p>Cuando: el usuario dé clic en el botón eliminar.</p> <p>Entonces: Le saldrá un modal mostrando el tiempo establecido del punto a dar de baja y la pregunta: ¿Desea eliminar el punto de control de la ruta?</p>			
Escenario 2: Aceptar eliminación.			
<p>Dado: el usuario desee dar de baja un punto de control de la ruta.</p> <p>Cuando: el usuario acepte el modal de eliminación.</p> <p>Entonces: Se mostrará un modal de éxito y se actualizará el tiempo total de recorrido de la ruta.</p>			

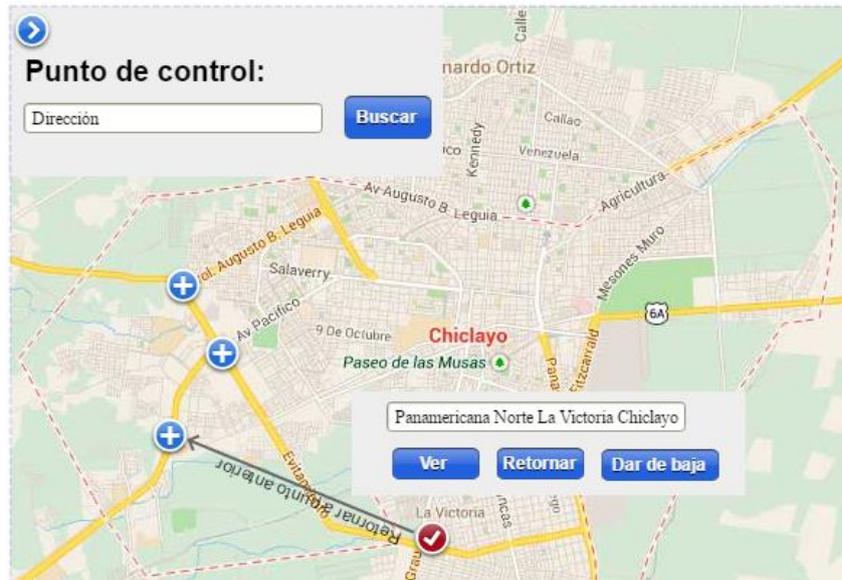


Figura 33 Prototipo del Sprint 02 – HU: Dar de baja puntos de control de ruta Fuente. Elaboración Propia

Tabla 41

HU: Registro de tiempos establecidos

HISTORIA DE USUARIO			
IDENTIFICADOR:	018		
NOMBRE:	Registro de tiempos establecidos		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	02
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Asignar tiempos de recorrido a cada ruta por punto de control</p> <p>Para: que las unidades sepan el tiempo definido de recorrido.</p>			
OBSERVACIONES			
Se debe registrar tiempo establecido por cada punto de control agregado a la ruta.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	6 h		
Realizar Pruebas y Corregir Errores	4 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Asignar tiempo			
<p>Dado: que la unidad intente asignar un punto de control a una ruta</p> <p>Cuando: el controlador de clic en el botón agregar punto de control.</p> <p>Entonces: La aplicación mostrará un modal para el registro del tiempo establecido para ese punto de control.</p>			
Escenario 2: Tiempos registrados			

Dado: que la ruta tenga tiempos ya registrados.

Cuando: el controlador seleccione una ruta específica.

Entonces: La aplicación mostrará los puntos de control que no estén asignados a la ruta para ser agregados.

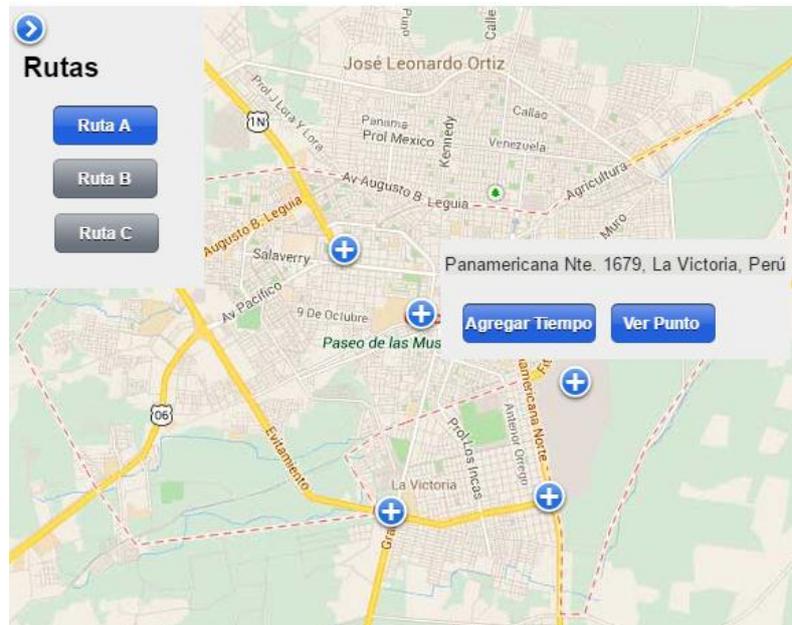


Figura 34 Prototipo del Sprint 02 – HU: Registro de tiempos establecidos
Fuente. Elaboración Propia

Agregar tiempo establecido

¿Desea agregar un nuevo tiempo establecido?

Ruta A

Punto de control:

Tiempo establecido (min):

Orden:

Figura 35 Prototipo del Sprint 02 – HU: Registro de tiempos establecidos
Fuente. Elaboración Propia

Tabla 42

HU: Modificar tiempos establecidos

HISTORIA DE USUARIO			
IDENTIFICADOR:	019		
NOMBRE:	Modificar tiempos establecidos		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Modificar tiempos de recorrido a cada ruta por punto de control.</p> <p>Para: que actualizar o corregir errores en los tiempos establecidos.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		3 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Modificar tiempo			
<p>Dado: que el usuario no coloque nada el tiempo establecido.</p> <p>Cuando: el usuario de clic en el botón modificar tiempo.</p> <p>Entonces: La aplicación mostrará un mensaje “Debe llenar todos los campos”</p>			

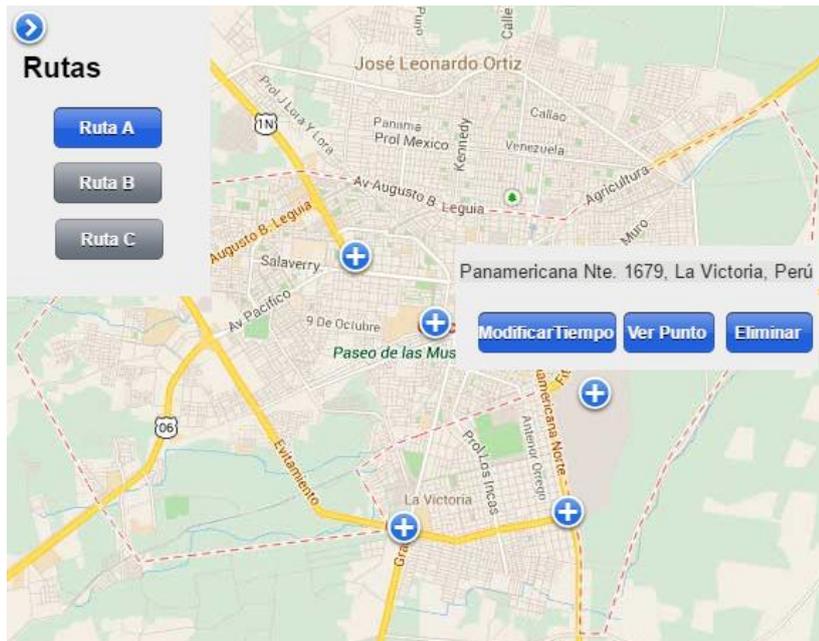


Figura 36 Prototipo del Sprint 02 – HU: Modificar tiempos establecidos
Fuente. Elaboración Propia

Agregar tiempo establecido

¿Desea agregar un nuevo tiempo establecido?

Ruta A

Punto de control:

Tiempo establecido (min):

Orden:

Figura 37 Prototipo del Sprint 02 – HU: Modificar tiempos establecidos
Fuente. Elaboración Propia

Tabla 43

HU: Listar tiempos

HISTORIA DE USUARIO			
IDENTIFICADOR:	020		
NOMBRE:	Listar tiempos		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: listar tiempos de recorrido.</p> <p>Para: poder visualizar los datos de los tiempos establecidos en las rutas.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos en la base de datos		1 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Listar tiempos establecidos			
<p>Dado: que el usuario desee visualizar los tiempos establecidos de cada ruta.</p> <p>Cuando: el usuario dé clic en el botón listar.</p> <p>Entonces: el sistema mostrará un modal con la lista de los tiempos establecidos de la ruta seleccionada</p>			

Listado de tiempos establecidos

Filtro de rutas:

Ruta A



Listar

Imprimir

Número	Ruta	Punto de control	Tiempo (min)
1	A	Panamericana Nte. 1679, La Victoria, Perú	15
2	A	Av los Andes 1525, La Victoria, Perú	9

Figura 38 Prototipo del Sprint 02 – HU: Listar Tiempos

Fuente. Elaboración Propia

Tabla 44

HU: Crear usuarios

HISTORIA DE USUARIO			
IDENTIFICADOR:	021		
NOMBRE:	Crear usuarios		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Generar usuarios administrativos</p> <p>Para: generar acceso a cada parte del sistema respecto al cargo de los trabajadores administrativos.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2	
Diseñar y crear tablas de la base de datos		2	
Construir la interfaz		2	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8	
Realizar Pruebas y Corregir Errores		4	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Mostrar cuenta autogenerada			
<p>Dado que los datos ingresados sean correctos</p> <p>Cuando la secretaria de clic en “Guardar y generar usuario”</p> <p>Entonces se mostrará la cuenta autogenerada (usuario y contraseña).</p>			

Listado de Personal

Filtro de Personal:

Buscar personal por apellido:

Número	Nombres y Apellidos	DNI	Cargo	Usuario	Estado	Opciones
1	Juan Carlos Perez Sandoval	73652458	Controlador	JPerezS	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>
2	Marta Sanchez Solis	74529685	Secretaria	MSanchezS	Inactivo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Activar"/> <input type="button" value="Generar usuario"/>
3	Oscar Vera Cruz	74986512	Gerente	OVeraC	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>

Figura 39 Prototipo del Sprint 02 – HU: Crear usuarios
Fuente. Elaboración propia

Generar Usuario

Personal:

Usuario:

Contraseña:

Fecha de registro

Figura 40 Prototipo del Sprint 02 – HU: Crear usuarios
Fuente. Elaboración Propia

Tabla 45

HU: Modificar contraseña de usuarios

HISTORIA DE USUARIO			
IDENTIFICADOR:	022		
NOMBRE:	Modificar contraseña de usuarios		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	13	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa</p> <p>Quiero: que los usuarios administrativos puedan modificar su contraseña</p> <p>Para: Colocar una contraseña con más seguridad y más confiable</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Modificar atributo de tablas de base de datos	½ h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	3 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: CONTRASEÑA NO SEGURA			
<p>Dado: que el usuario ingrese la contraseña demasiado corta o sin caracteres de seguridad</p> <p>Cuando: quiera modificar la contraseña</p> <p>Entonces: se le mostrara un mensaje indicando que la contraseña debe ser una contraseña con más seguridad (con un mínimo 6 y un máximo de 15 caracteres)</p>			

Modificar Usuario

Usuario:

Contraseña Actual:

Nueva Contraseña:

Repetir Contraseña:

Figura 41 Prototipo del Sprint 02 – HU: Modificar contraseña de usuario
Fuente. Elaboración Propia

Tabla 46

HU: Dar de baja usuarios

HISTORIA DE USUARIO			
IDENTIFICADOR:	023		
NOMBRE:	Dar de baja usuarios		
PRIORIDAD:	Alta	USUARIO:	Gerente
ESTIMACIÓN:	5	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja a los usuarios administrativos</p> <p>Para: sacar del sistema a los usuarios que ya no pertenecerán a la empresa.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		½ h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Dar de baja a usuario			
<p>Dado que haya personal con usuarios registrados</p> <p>Cuando el gerente de clic en “Inactivar”</p> <p>Entonces se mostrará un mensaje “El personal con su usuario ha cambiado de estado a inactivo” y el botón mostrara la opción de “Activar”.</p>			

Escenario 2: Activar usuario dado de baja anteriormente

Dado: que se quiera activar el usuario de un personal

Cuando: se seleccione el botón “Activar”.

Entonces: La aplicación pasara al usuario del personal a estado “Activo” y el botón mostrara la opción de “Desactivar”.

Listado de Personal

Filtro de Personal:
Todos

Buscar personal por apellido:
Ingrese apellido del personal

Número	Nombres y Apellidos	DNI	Cargo	Usuario	Estado	Opciones
1	Juan Carlos Perez Sandoval	73652458	Controlador	JPerezS	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>
2	Marta Sanchez Solis	74529685	Secretaria	MSanchezS	Inactivo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Activar"/> <input type="button" value="Generar usuario"/>
3	Oscar Vera Cruz	74986512	Gerente	OVeraC	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>

Figura 42 Prototipo del Sprint 02 – HU: Dar de baja usuario
Fuente. Elaboración Propia

Tabla 47

HU: Registrar personal

HISTORIA DE USUARIO			
IDENTIFICADOR:	025		
NOMBRE:	Registrar personal		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Registrar el personal de la empresa</p> <p>Para: Llevar un mejor control del personal que labora en mi empresa.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registro de Personal			
<p>Dado que el usuario quiera registrar un DNI ya registrado</p> <p>Cuando el usuario dé clic en “Registrar Personal”</p> <p>Entonces se mostrará un mensaje “Error de DNI”.</p>			
Escenario 2: Edad del personal			

Dado que el usuario registre la fecha de nacimiento y la edad sea menor a 18 años

Cuando el usuario dé clic en “Registrar personal”

Entonces se mostrará un mensaje “El personal debe ser mayor de edad”.

Registro de personal

Nombres:
Ingrese nombres del personal

Apellido Paterno:
Ingrese apellido paterno del personal

Apellido Materno:
Ingrese apellido materno del personal

DNI:
Ingrese DNI del personal

Fecha de Nacimiento:
dd/mm/aaaa

Sexo:
 Masculino Femenino

Fecha de Registro:
dd/mm/aaaa

Dirección:
Ingrese dirección del personal

Estado Civil:
Casado

Cargo:
Controlador

Correo:
Ingrese correo del personal

Celular:
Ingrese celular del personal

Registrar Personal

Figura 43 Prototipo del Sprint 02 – HU: Registrar personal
Fuente. Elaboración Propia

Tabla 48

HU: Modificar personal

HISTORIA DE USUARIO			
IDENTIFICADOR:	026		
NOMBRE:	Modificar personal		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Modificar los datos del personal de la empresa</p> <p>Para: Corregir datos ingresados de manera errónea.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Modificar atributo de tablas de base de datos		1 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Modificar Fecha de Nacimiento			
<p>Dado que el usuario quiera modificar la fecha de nacimiento y la edad sea menor a 18 años</p> <p>Cuando el usuario de clic en “Modificar Personal”</p> <p>Entonces se mostrará un mensaje “El personal debe ser mayor de edad”.</p>			
Escenario 2: Modificar DNI de Personal			

Dado que el usuario quiera modificar un DNI e ingrese un nro. de DNI ya registrado

Cuando el usuario de clic en “Modificar Personal”

Entonces se mostrará un mensaje “Error de DNI”.

Modificar personal

Nombres:
Juan carlos

Apellido Paterno:
Perez

Apellido Materno:
Sandoval

DNI:
73985624

Fecha de Nacimiento:
08/08/1989

Sexo:
 Masculino Femenino

Fecha de Registro:
05/05/2018

Dirección:
constituicion#181 - JLO

Estado Civil:
Casado

Cargo:
Controlador

Correo:
juan_sandoval@gmail.com

Celular:
986536285

Modificar Personal

Figura 44 Prototipo del Sprint 02 – HU: Modificar personal
Fuente. Elaboración Propia

Tabla 49

HU: Dar de baja personal

HISTORIA DE USUARIO			
IDENTIFICADOR:	027		
NOMBRE:	Dar de baja personal		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	8	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja al personal de la empresa</p> <p>Para: sacar del sistema al personal que ya no pertenece a la empresa.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2	
Modificar atributo de tablas de base de datos		1 h	
Construir la interfaz		2	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Dar de baja personal con usuario			
<p>Dado que el usuario quiera dar de baja a un personal y este tenga un usuario</p> <p>Cuando el usuario dé clic en “Dar de baja Personal”</p> <p>Entonces se pasará a estado “Inactivo” el personal y su usuario correspondiente mostrando un mensaje “Eliminación exitosa”.</p>			

Escenario 2: Activar personal dado de baja anteriormente

Dado: que el usuario quiera activar un personal que se a dado de baja anteriormente.

Cuando: el usuario seleccione el botón “Activar”.

Entonces: La aplicación pasara al personal y su usuario correspondiente a estado “Activo”.

Listado de Personal

Filtro de Personal:
Todos

Buscar personal por apellido:
Ingrese apellido del personal

Número	Nombres y Apellidos	DNI	Cargo	Usuario	Estado	Opciones
1	Juan Carlos Perez Sandoval	73652458	Controlador	JPerezS	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>
2	Marta Sanchez Solis	74529685	Secretaria	MSanchezS	Inactivo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Activar"/> <input type="button" value="Generar usuario"/>
3	Oscar Vera Cruz	74986512	Gerente	OVeraC	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>

Figura 45 Prototipo del Sprint 02 – HU: Dar de baja personal
Fuente. Elaboración Propia

Tabla 50

HU: Visualizar personal

HISTORIA DE USUARIO			
IDENTIFICADOR:	028		
NOMBRE:	Visualizar personal		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	02
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de un personal</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar Datos			
<p>Dado: que el usuario quiera visualizar los datos de un personal</p> <p>Cuando: el usuario de clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con todos los datos del personal</p>			
Escenario 2: Imprimir datos del socio			

Dado: Que el usuario quiera imprimir los datos de un personal.

Cuando: el usuario de clic en el botón “Imprimir”

Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos del personal.

Listado de Personal

Filtro de Personal:
Todos

Buscar personal por apellido:
Ingrese apellido del personal

Número	Nombres y Apellidos	DNI	Cargo	Usuario	Estado	Opciones
1	Juan Carlos Perez Sandoval	73652458	Controlador	JPerezS	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>
2	Marta Sanchez Solis	74529685	Secretaria	MSanchezS	Inactivo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Activar"/> <input type="button" value="Generar usuario"/>
3	Oscar Vena Cruz	74986512	Gerente	OVenaC	Activo	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Inactivar"/> <input type="button" value="Generar usuario"/>

Figura 46 Prototipo del Sprint 02 – HU: Visualizar personal
Fuente. Elaboración Propia

Datos del personal

Nombres y apellidos:

DNI:

Fecha de Nacimiento:

Sexo:

Dirección:

Estado Civil:

Estado:

Fecha de Registro:

Fecha de Retiro:

Celular:

Correo:

Cargo:

Figura 47 Prototipo del Sprint 02 – HU: Visualizar personal
Fuente. Elaboración Propia

3.3.2 Tablero SCRUM

A continuación se muestra el tablero SCRUM donde se ordenó las historias de usuario con sus respectivas tareas, conforme se iban desarrollando en el sprint.

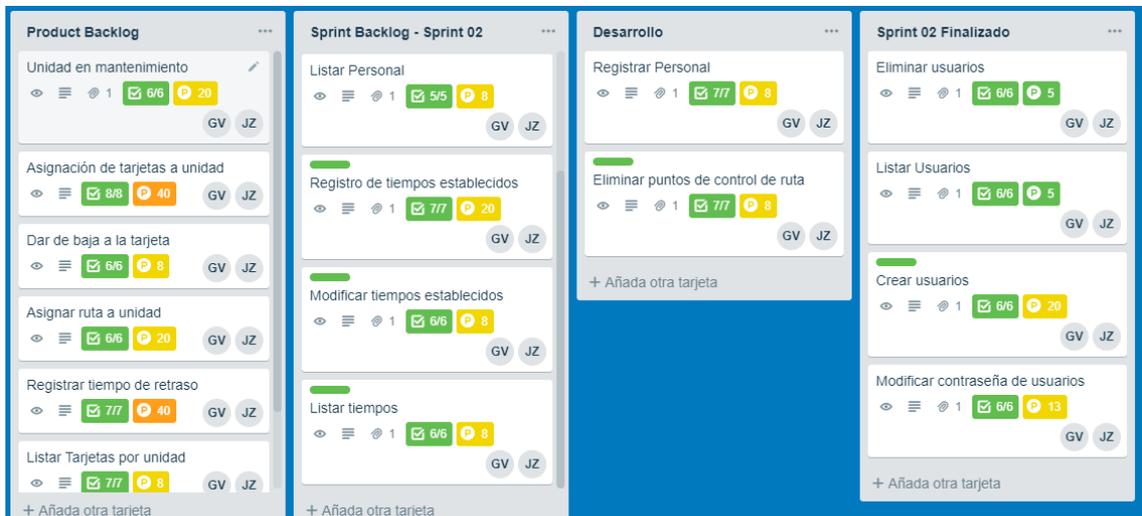


Figura 48 Tablero SCRUM del Sprint 02
Fuente. Elaboración Propia

3.3.3 Diagrama de Burndown N° 02.

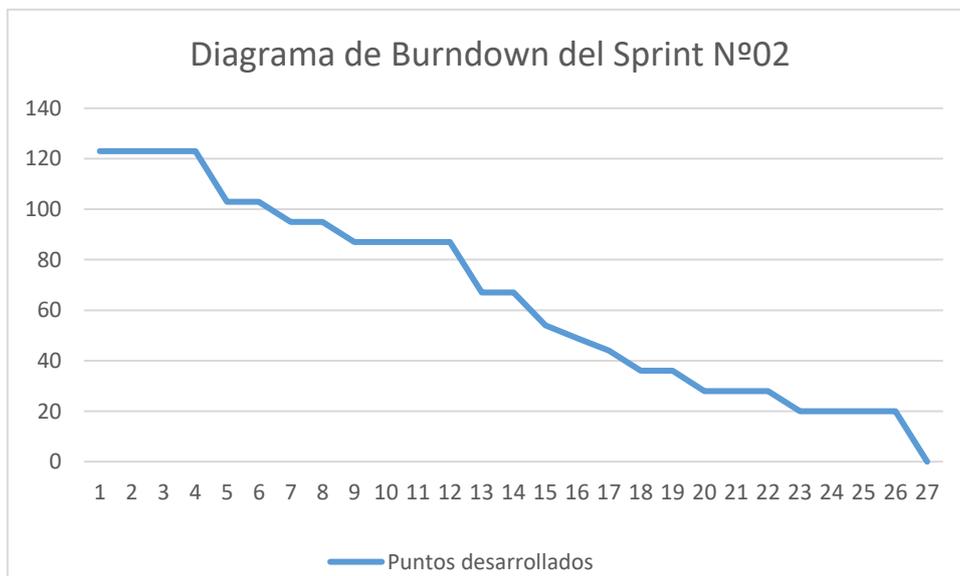


Figura 49 Diagrama de Burndown del Sprint 02
Fuente. Elaboración Propia

3.3.4 Retrospectiva

En esta etapa se evaluaron tres aspectos importantes; buenos, malos y aspectos de mejora:

¿Qué podríamos seguir manteniendo?

- ✓ Se logró comunicación constante con los usuarios y se cumplieron las visitas quincenales.
- ✓ Buen ritmo de desarrollo por parte del equipo.
- ✓ Cumplimiento total de las historias de usuario.
- ✓ Puntuación de historias de usuario más realistas.
- ✓ Se obtuvo puntualidad en las reuniones designadas.

¿Qué pudimos hacer de manera diferente?

- ✓ Aprender mejor el uso del Framework Spring.
- ✓ Tener una constante actualización del tablero Scrum.

¿Qué debemos mejorar?

- ✓ Dedicar más tiempo a leer información sobre el Framework.
- ✓ Actualizar el tablero Scrum.

Finalmente vemos en la Tabla 51, el cierre del proyecto para este Sprint.

Tabla 51

Cierre de Sprint N°02

ID	HISTORIA DE USUARIO	RESPONSABLE	ESTADO	REVISIÓN
HU-017	Dar de baja punto	Solis Vera	Realizado	Conforme
	de control	Gabriel Leonardo		
HU-018	Registro de	Solis Vera	Realizado	Conforme
	tiempos establecidos	Gabriel Leonardo		
HU-019	Modificar	Solis Vera	Realizado	Conforme
	tiempos establecidos	Gabriel Leonardo		

		Solis Vera		
HU-020	Listar tiempos	Gabriel Leonardo	Realizado	Conforme
HU-021	Crear usuarios	Zamora Granados Joel	Realizado	Conforme
HU-022	Modificar Usuario	Zamora Granados Joel	Realizado	Conforme
HU-023	Dar de baja a usuario	Zamora Granados Joel	Realizado	Conforme
HU-024	Listar usuario	Zamora Granados Joel	Realizado	Conforme
HU-025	Registrar personal	Zamora Granados Joel	Realizado	Conforme
HU-026	Modificar personal	Zamora Granados Joel	Realizado	Conforme
HU-027	Dar de baja personal	Zamora Granados Joel	Realizado	Conforme
HU-028	Visualizar personal	Zamora Granados Joel	Realizado	Conforme

Fuente: Elaboración Propia.

3.4 Sprint N°03

3.4.1 Planificación.

Para este sprint se consideraron las historias de usuario que muestran la Tabla 52 y los prototipos.

Tabla 52

Sprint Backlog – Sprint N°03

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-029	Justificación de no disponibilidad de la unidad	Media	20
HU-030	Asignación de tarjeta a unidad	Alta	40
HU-031	Dar de baja tarjeta	Media	08
HU-032	Visualizar tarjeta	Alta	20
HU-033	Asignar recorrido a unidad	Alta	20
HU-034	Dar de baja a recorrido asignada a unidad	Media	20
Total, puntos estimados			128

Fuente: Elaboración Propia

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 53 a la Tabla 58

Tabla 53

HU: Justificación de no disponibilidad de la unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	029		
NOMBRE:	Justificación de no disponibilidad de la unidad		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar el motivo de porque una unidad no está trabajando</p> <p>Para: otorgarle su respectivo plazo de tiempo o sanción al no ser justificado.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Justificación			
<p>Dado que el usuario quiera poner inactivo una unidad por algún motivo de daño en la unidad.</p> <p>Cuando el usuario de clic en el botón “Mantenimiento”</p> <p>Entonces se mostrará un modal de registro de justificación y fecha de inactivo</p>			

Datos de Mantenimiento

Justificación:

Vehiculo tiene que realizar revisión tecnica y reparación de motor por lo cual se le brinda un plazo de 5 dias

Fecha de Mantenimiento

15/08/2018

Dias de Plazo:

5

Figura 50 Prototipo del Sprint 03 – HU: Justificación de no disponibilidad de unidad
Fuente. Elaboración Propia

Tabla 54

HU: Asignación de tarjetas a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	030		
NOMBRE:	Asignación de tarjetas a unidad		
PRIORIDAD:	Alta	USUARIO:	Controlador
ESTIMACIÓN:	40	SPRINT:	03
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Asignar una tarjeta a la unidad.</p> <p>Para: que la unidad pueda conocer que rutas debe recorrer y sus tiempos.</p>			
OBSERVACIONES			
<p>*La tarjeta se debe entregar una sola vez, de otra manera se debe aplicar una sanción.</p> <p>* Datos a registrar: Placa de la unidad, fecha y hora de registro, rutas y tiempos a recorrer.</p>			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registro de tarjeta			
<p>Dado que la unidad ya tenga una tarjeta activa registrada ese día</p> <p>Cuando el controlador de clic en el botón buscar placa</p>			

Entonces La aplicación mostrará un mensaje informando:” ya se ha registrado una tarjeta con anterioridad y no permitirá el registro“.

Escenario 2: Búsqueda de placa.

Dado que el controlador ingresa una placa de una unidad no registrada.

Cuando el controlador de clic en buscar placa.

Entonces La aplicación mostrará un mensaje de placa no registrada.

Asignar tarjeta

Fecha actual: Número de tarjeta:

Placa:

Vehiculo: Socio:

Ruta:

Hora de inicio de control

Punto de control	Hora de control	Tiempo desde inicio	Tiempo comodin
1	10:00 p.m.	15	2
2	10:15 p.m.	10	0

Figura 51 Prototipo del Sprint 03 – HU: Asignación de tarjetas a unidad
Fuente. Elaboración Propia

Tabla 55

HU: Dar de baja a la tarjeta

HISTORIA DE USUARIO			
IDENTIFICADOR:	031		
NOMBRE:	Dar de baja a la tarjeta		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	8	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja a las tarjetas de control.</p> <p>Para: Poder deshabilitar una tarjeta registrada con errores.</p>			
OBSERVACIONES			
<p>* Se debe guardar el usuario que ha realizado la baja de la tarjeta.</p> <p>* Se debe registrar el motivo de la baja.</p>			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Motivo en blanco			
<p>Dado que se el campo de motivo de baja este en blanco</p> <p>Cuando el controlador de clic en el botón aceptar del modal dar de baja</p> <p>Entonces La aplicación mostrará un mensaje informando: “Que el motivo no puede estar en blanco”.</p>			
Escenario 2: Dar de baja a tarjeta			

<p>Dado: que el usuario quiera dar de baja a una tarjeta</p> <p>Cuando: el usuario seleccione el botón “Desactivar”.</p> <p>Entonces: La aplicación pasara la tarjeta a estado “Inactiva”, la cual será su estado final.</p>
<p>Escenario 3: Listar tarjetas por unidad</p>
<p>Dado que el usuario desee visualizar las tarjetas que posea una unidad.</p> <p>Cuando el usuario haga clic en el botón listar.</p> <p>Entonces el sistema listará la lista de tarjetas.</p>
<p>Escenario 4: Rango de fechas</p>
<p>Dado que el usuario desee visualizar las tarjetas que posea una unidad.</p> <p>Cuando el usuario haga clic en el botón listar.</p> <p>Entonces el sistema listará la lista de tarjetas de acuerdo con el rango de fechas ingresado.</p>

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 52 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta
Fuente. Elaboración Propia

Dar de baja tarjeta

Motivo:

Dar de baja
Cancelar

Figura 53 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta
Fuente. Elaboración Propia

Tabla 56

HU: Visualizar tarjeta

HISTORIA DE USUARIO			
IDENTIFICADOR:	004		
NOMBRE:	Visualizar tarjeta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de una tarjeta</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			

TAREAS	
Descripción	Duración (horas)
Diseñar la interfaz	2 h
Construir la interfaz	2 h
Crear procedimientos almacenados en la base de datos	2 h
Crear el código correspondiente	8 h
Realizar Pruebas y Corregir Errores	4 h
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Visualizar Datos	
<p>Dado: que el usuario quiera visualizar los datos de una tarjeta</p> <p>Cuando: el usuario de clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con todos los datos de la tarjeta</p>	
Escenario 2: Imprimir datos de tarjeta	
<p>Dado: Que el usuario quiera imprimir los datos de una tarjeta.</p> <p>Cuando: el usuario de clic en el botón “Imprimir”</p> <p>Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos de la tarjeta.</p>	

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 54 Prototipo del Sprint 03 – HU: Visualizar tarjeta
Fuente. Elaboración Propia



Figura 55 Prototipo del Sprint 03 – HU: Visualizar tarjeta
 Fuente. Elaboración Propia

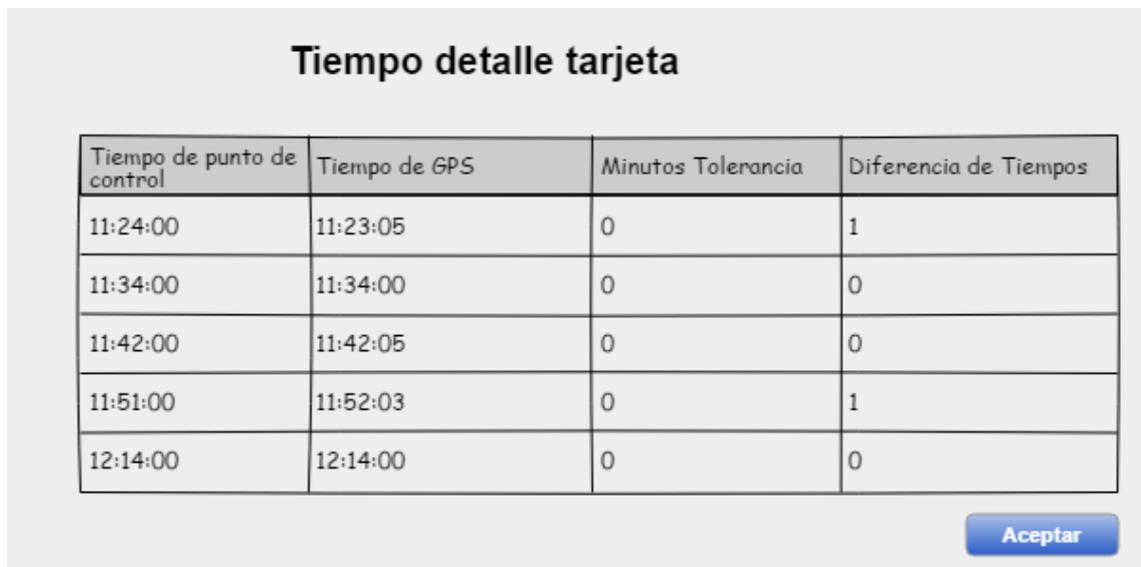


Figura 56 Prototipo del Sprint 03 – HU: Visualizar tarjeta
 Fuente. Elaboración Propia

Tabla 57

HU: Asignar recorrido a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	033		
NOMBRE:	Asignar recorrido a unidad		
PRIORIDAD:	Alta	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Asignar rutas a la unidad.</p> <p>Para: Poder controlar los tiempos de cada unidad de acuerdo a cada ruta asignada.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Reasignación de ruta			
<p>Dado que el usuario desee asignar una ruta ya asignada a la unidad y esta aun no la recorre.</p> <p>Cuando el usuario dé clic en el botón “asignar ruta”.</p> <p>Entonces el sistema mostrara un mensaje de alerta indicando que no se puede reasignar una ruta que aún no se ha recorrido.</p>			

Asignar Ruta

Fecha actual:
 Número de tarjeta:
 Número de vuelta:

Placa:

Vehículo:
 Socio:

Ruta:

Hora de inicio de control

Punto de control	Hora de control	Tiempo desde inicio	Tiempo comodin
1	10:00 p.m.	15	2
2	10:15 p.m.	10	0

Figura 57 Prototipo del Sprint 03 – HU: Asignar recorrido a unidad
 Fuente. Elaboración Propia

Tabla 58

HU: Dar de baja recorrido asignado a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	034		
NOMBRE:	Dar de baja recorrido asignado a unidad		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: poder modificar la ruta asignada a una unidad.</p> <p>Para: poder cambiar su recorrido y tiempo por causa de error</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Especificar motivo			
<p>Dado Que se desee cambiar la ruta asignada de unidad</p> <p>Cuando el controlador de clic en el botón dar de baja ruta en el listado de rutas asignadas a unidad</p> <p>Entonces La aplicación mostrará un modal para escribir el motivo del cambio</p>			

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 58 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia

Visualizar tarjeta

Ruta	Número Vuelta	Opciones
A	1	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>
B	2	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>
C	3	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>

Figura 59 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia



Figura 60 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia

3.4.2 Tablero SCRUM.

A continuación se muestra el tablero SCRUM donde se ordenó las historias de usuario con sus respectivas tareas, conforme se iban desarrollando en el sprint.

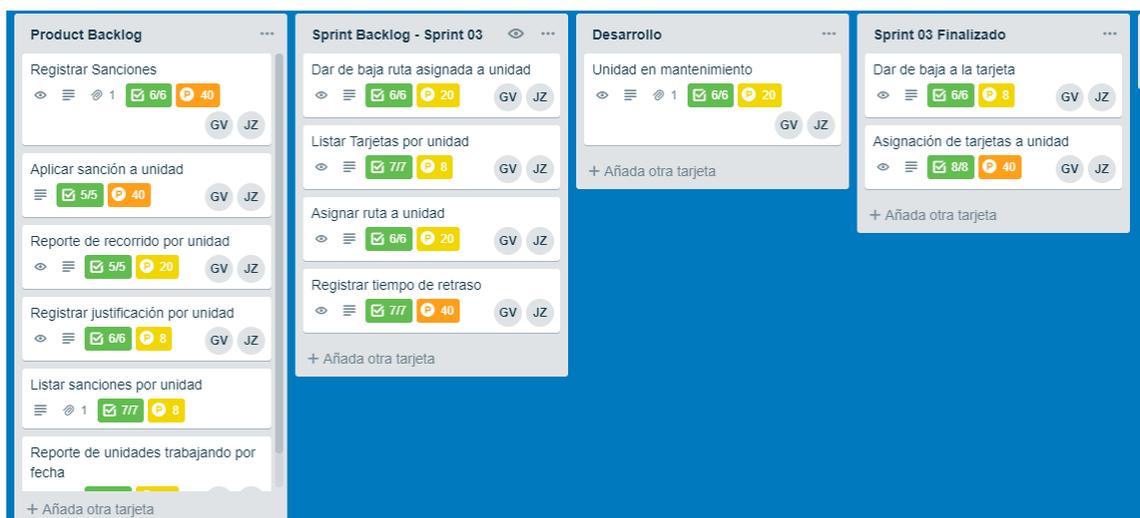


Figura 61 Tablero SCRUM del Sprint 03
Fuente. Elaboración Propia

3.4.3 Diagrama de Burndown del Sprint N° 03

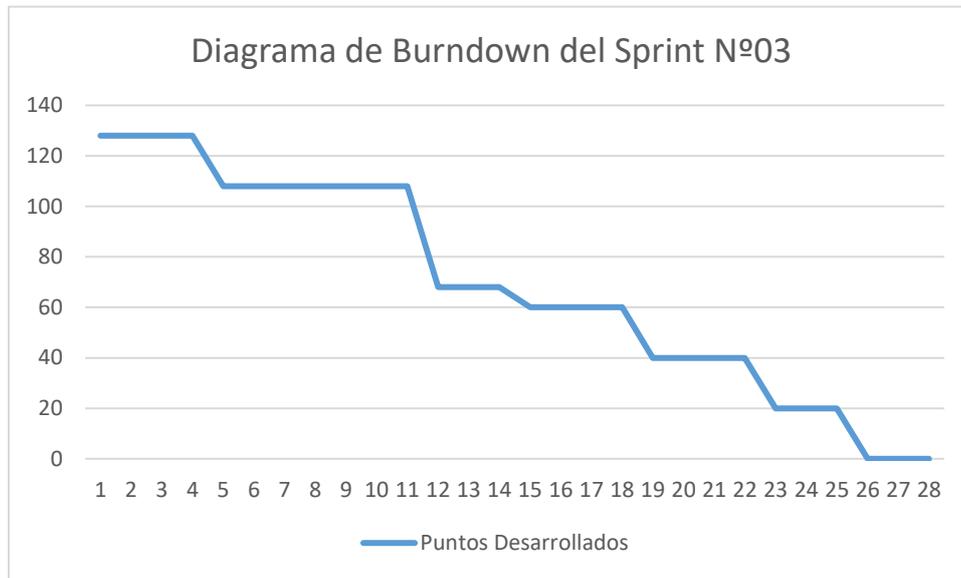


Figura 62 Diagrama de Burndown del Sprint 03
Fuente. Elaboración Propia

3.4.4 Retrospectiva

En esta etapa se evaluaron tres aspectos importantes; buenos, malos y aspectos de mejora:

¿Qué podríamos seguir manteniendo?

- ✓ Se logró comunicación constante con los usuarios y se cumplieron las visitas quincenales.
- ✓ Buen ritmo de desarrollo por parte del equipo.
- ✓ Cumplimiento total de las historias de usuario.
- ✓ Puntuación de historias de usuario más realistas.
- ✓ Se obtuvo puntualidad en las reuniones designadas.
- ✓ Se logró actualizar constantemente el tablero Scrum.

¿Qué pudimos hacer de manera diferente?

- ✓ Coordinar tiempos para la configuración y mantenimiento del sistema

- ✓ Avisar los inconvenientes por lo cual no se logró avanzar correctamente las tareas asignadas.

¿Qué debemos mejorar?

- ✓ Dedicar más tiempo para la configuración y mantenimiento del sistema.
- ✓ Comunicar con tiempo los problemas que se van teniendo con el desarrollo de algunas tareas para podernos ayudar en equipo.

Finalmente vemos en la Tabla 59, el cierre del proyecto para este Sprint.

Tabla 59

Cierre de Sprint N°02

ID	HISTORIA DE USUARIO	RESPONSABLE	ESTADO	REVISIÓN
HU-029	Justificación de no disponibilidad de la unidad	Zamora Granados Joel	Realizado	Conforme
HU-030	Asignación de tarjeta a unidad	Zamora Granados Joel	Realizado	Conforme
HU-031	Dar de baja tarjeta	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-032	Visualizar tarjeta	Zamora Granados Joel	Realizado	Conforme
HU-033	Asignar recorrido a unidad	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-034	Dar de baja a recorrido asignada a unidad	Solis Vera Gabriel Leonardo	Realizado	Conforme

Fuente: Elaboración Propia

3.5 Sprint N°04

3.5.1 Planificación.

Para este sprint se consideraron las historias de usuario que muestran la Tabla 60 y los prototipos

Tabla 60

Sprint Backlog – Sprint N° 04

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-035	Registrar sanción	Media	40
HU-036	Aplicar sanción a unidad	Media	40
HU-037	Registrar justificación	Media.	08
HU-038	Visualizar de recorrido por unidad	Media.	20
HU-039	Reporte de unidades trabajando por fecha	Media	20
HU-040	Listar sanciones por unidad	Media	08
HU-041	Reporte de retrasos de unidades	Baja	20
HU-042	Reporte de recorrido diario	Baja	20
Total, puntos estimados			176

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 61 a la Tabla 68

Tabla 61

HU: Registrar sanciones

Desarrollo de la historia de usuario (Ver apéndice A).

HISTORIA DE USUARIO			
IDENTIFICADOR:	035		
NOMBRE:	Registrar Sanciones		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	40	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar las sanciones a tener en cuenta para el personal de la empresa</p> <p>Para: aplicar las respectivas sanciones según las faltas cometidas del personal de la empresa.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registrar sanción ya registrada			

Dado que el usuario quiera registrar una sanción ya registrada

Cuando el usuario de clic en el botón “Registrar Sanción”

Entonces se mostrará un mensaje “Sanción ya registrada”.

Registro de sanción

Tiempo de retraso total

Minutos sanción

Motivo

Justificación

Aplicar Sanción Cancelar

Figura 63 Prototipo del Sprint 04 – HU: Registrar sanción
Fuente. Elaboración Propia

Tabla 62

HU: Aplicar sanciones a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	036		
NOMBRE:	Aplicar sanciones a unidad		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	40	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: aplicar una sanción a la unidad.</p> <p>Para: Poder llevar el correcto control en los tiempos de recorridos de las unidades.</p>			
OBSERVACIONES			
<p>* Se debe guardar el usuario que ha realizado la baja de la tarjeta.</p> <p>* Se debe registrar el motivo de la baja.</p>			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Tiempo acumulado			
<p>Dado Que se desee registrar una nueva ruta a recorrer por una unidad</p> <p>Cuando el controlador de clic en el registrar ruta a tarjeta</p> <p>Entonces se visualizará los tiempos de la ruta más los tiempos por sanción.</p>			



Figura 64 Prototipo del Sprint 04 – HU: Aplicar sanciones a unidad
Fuente. Elaboración Propia

Tabla 63

HU: Registrar justificación por unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	037		
NOMBRE:	Registrar justificación por unidad		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	8	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: aplicar una justificación por una sanción</p> <p>Para: Poder mantener registrada la razón por la cual no se le aplicó una sanción a una unidad.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	1 h		
Construir la interfaz	2 h		

Crear procedimientos almacenados en la base de datos	2 h
Crear el código correspondiente	8 h
Realizar Pruebas y Corregir Errores	2 h
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Justificación registrada	
Dado Que el controlador desee registrar una justificación	
Cuando De clic en registrar justificación	
Entonces El sistema no aplicará la sanción respectiva.	

Registro de justificación

Tiempo de retraso total

Justificación

Registrar Justificación Cancelar

Figura 65 Prototipo del Sprint 04 – HU: Registrar justificación por unidad
Fuente. Elaboración Propia

Tabla 64

HU: Visualizar de recorrido por unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	038		
NOMBRE:	Visualizar de recorrido por unidad		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar el recorrido en tiempo real de las unidades.</p> <p>Para: Poder aplicar alguna sanción si es necesario.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Recorrido de unida			
<p>Dado Que el usuario desee visualizar el recorrido actual de una unidad</p> <p>Cuando el usuario de clic en el botón visualizar recorrido</p> <p>Entonces El sistema mostrará marcada de un color la ruta actual de la unidad y con un marcador la posición actual de la unidad.</p>			



Figura 66 Prototipo del Sprint 04 – HU: Visualizar recorrido por unidad
Fuente. Elaboración Propia

Tabla 65

HU: Reporte de unidades trabajando por fecha

HISTORIA DE USUARIO			
IDENTIFICADOR:	039		
NOMBRE:	Reporte de unidades trabajando por fecha		
PRIORIDAD:	Media	USUARIO:	Gerente
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar el listado de unidades que trabajaron en el día.</p> <p>Para: Poder cobrar las contribuciones diarias a las unidades que trabajaron.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		2 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Listar por tipo			
<p>Dado que el usuario quiera visualizar las unidades que trabajaron por día , semana o mes</p> <p>Cuando el usuario de clic en el botón listar</p> <p>Entonces La aplicación listará las unidades según el criterio de búsqueda</p>			
Escenario 2: Listar unidad específica			

Dado que el usuario quiera visualizar los días trabajados de una unidad en específico.

Cuando el usuario ingrese la placa de la unidad a buscar y de clic en el botón listar.

Entonces La aplicación mostrara los días trabajados por la unidad.

Reporte de unidades trabajando por fecha

Fecha : Placa :

Filtro de tarjetas Unidades Trabajando Unidades sin trabajar

Socio	Vehículo
Oracio Flores Martinez	MPL-123
Jose Campos Cabrera	PLQ-987

Figura 67 Prototipo del Sprint 04 – HU: Reporte de unidades trabajando por fecha
Fuente. Elaboración Propia

Tabla 66

HU: Listar sanciones por unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	40		
NOMBRE:	Listar sanciones por unidad		
PRIORIDAD:	Media	USUARIO:	Gerente
ESTIMACIÓN:	8	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Conocer la lista de sanciones que posee una unidad</p> <p>Para: Poder tomar las medidas correctivas necesarias</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar pruebas y corregir errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Rango de fechas			
<p>Dado Que se quiera listar las sanciones de una unidad</p> <p>Cuando el usuario de clic en el botón “buscar sanciones”</p> <p>Entonces el sistema validara que se deba llenar los filtros de fecha de inicio, fecha final y placa de unidad.</p>			
Escenario 2: Sin sanciones			

Dado que la unidad no cuente con sanciones registradas entre las fechas de filtro

Cuando el usuario de clic en buscar sanciones

Entonces Se mostrará un mensaje "La unidad no posee sancione registradas en la fecha ingresada".

Listar sanciones por unidad

Desde Fecha: Hasta Fecha: Placa:

Filtro de tarjetas

Sanciones

Fecha	Número de vuelta	Ruta	Motivo	Tiempo de retraso	Tiempo sanción	justificación
05/08/2018	1	A	Retraso	5 min	15 min	Ninguna
06/08/2018	2	B	Retraso	10 min	20 min	Se malogro el carro

Figura 68 Prototipo del Sprint 04 – HU: Listar sanciones por unidad

Fuente. Elaboración Propia

Tabla 67

HU: Reporte de retrasos de tiempo

HISTORIA DE USUARIO			
IDENTIFICADOR:	41		
NOMBRE:	Reporte de retrasos de tiempo		
PRIORIDAD:	Baja	USUARIO:	Gerente
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: conocer los retrasos por día de cada unidad.</p> <p>Para: Poder tomar las decisiones necesarias para evitar los retrasos</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2		
Diseñar y crear tablas de la base de datos	2		
Crear procedimientos almacenados en la base de datos	2 h		
Construir la interfaz	2		
Crear el código correspondiente	8		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Listado de retraso por unidad			
<p>Dado Que el usuario quiera visualizar los retrasos de una unidad</p> <p>Cuando el usuario llene el filtro de fecha actual ,placa de la unidad y de clic en el botón listar retrasos</p> <p>Entonces se mostrará la unidad con sus rutas recorridos y sus retrasos.</p>			

Reporte de retrasos de tiempo

Fecha :

Generar Reporte

Unidad	Vuelta 1	Vuelta 2
MPL-001	A - 4 min	D - 5 min
MPL-000	B - 1 min	C - 3 min
ABC-123	C - 0 min	A - 5min

Figura 69 Prototipo del Sprint 04 – HU: Reporte de retrasos de tiempo
Fuente. Elaboración Propia

Tabla 68

HU: Reporte de recorrido diario

HISTORIA DE USUARIO			
IDENTIFICADOR:	42		
NOMBRE:	Reporte de recorrido diario		
PRIORIDAD:	Baja	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: obtener una planilla diaria de recorrido</p> <p>Para: Poder dar informe de cuales unidades han trabajado, las rutas que han recorrido y el número de vueltas que han realizado</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Día sin unidades laborando			
<p>Dado que el usuario quiera obtener una planilla de un día que no laboraron unidades.</p> <p>Cuando seleccione la opción correspondiente</p> <p>Entonces se mostrará un mensaje informativo “No se encontraron unidades que laboraron este día”</p>			

Escenario 2: Planilla con casilleros en blanco

Dado que el usuario quiera obtener una planilla de todas las unidades

Cuando seleccione la opción correspondiente

Entonces se la planilla con casilleros en blanco debido a que el sistema tomara como referencia la unidad que realizo más vueltas en el día laborado.

Reporte de recorrido diario

Fecha : Generar Reporte

Unidad	Vuelta 1	Vuelta 2
MPL-001	A 11:30:00 - 11:35:00	B 12:30:00 - 12:35:00
MPL-000	B 12:25:00 - 12:25:00	A 13:25:00 - 13:25:00
ABC-123	C 10:20:00 - 10:20:00	B 11:20:00 - 11:20:00

Figura 70 Prototipo del Sprint 04 – HU: Reporte de recorrido diario
Fuente. Elaboración Propia

3.5.2 Tablero SCRUM.

A continuación se muestra el tablero SCRUM donde se ordenó las historias de usuario con sus respectivas tareas, conforme se iban desarrollando en el sprint.

The image shows a Scrum board interface with three columns:

- Sprint Backlog - Sprint 04:**
 - Reporte de recorrido por unidad: 5/5 (green), 20 (yellow), assignees GV, JZ.
 - Registrar justificación por unidad: 6/6 (green), 8 (yellow), assignees GV, JZ.
 - + Añada otra tarjeta
- Desarrollo:**
 - Reporte de unidades trabajando por fecha: 6/6 (green), 20 (yellow), assignees GV, JZ.
 - Listar sanciones por unidad: 1 (grey), 7/7 (green), 8 (yellow), assignees GV, JZ.
 - + Añada otra tarjeta
- Sprint 4 Finalizado:**
 - Registrar Sanciones: 1 (grey), 6/6 (green), 40 (orange), assignees GV, JZ.
 - Aplicar sanción a unidad: 5/5 (green), 40 (orange), assignees GV, JZ.
 - + Añada otra tarjeta

Figura 71 Tablero SCRUM del Sprint 04
Fuente. Elaboración Propia

3.5.3 Diagrama de Burndown del Sprint N° 04.

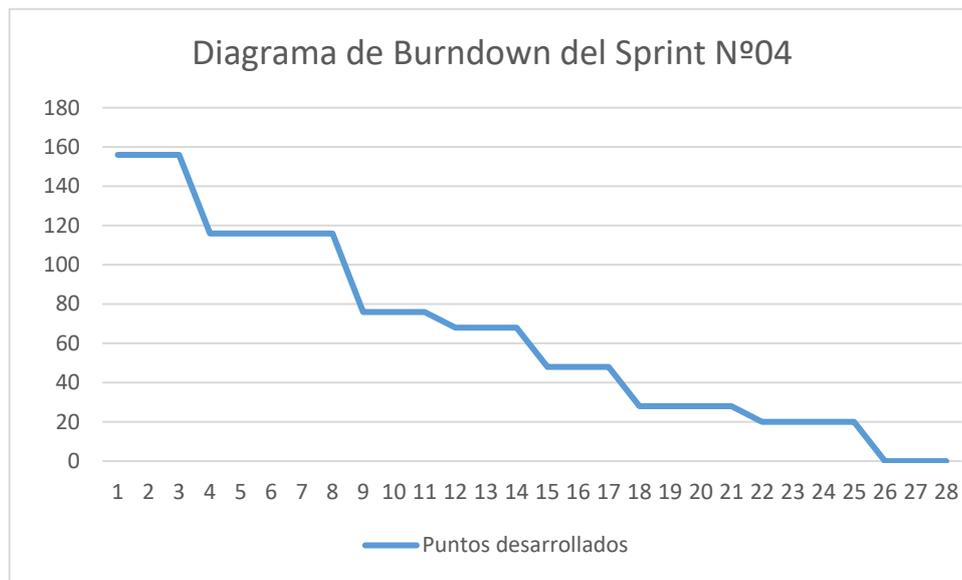


Figura 72 Diagrama de Burndown del Sprint 04
Fuente. Elaboración Propia

3.5.4 Retrospectiva.

En esta etapa se evaluaron tres aspectos importantes; buenos, malos y aspectos de mejora:

¿Qué podríamos seguir manteniendo?

- ✓ Puntuación de historias de usuario más realistas.
- ✓ Corrección y mantenimientos del sistema.
- ✓ Mejor comunicación entre el equipo de trabajo para ayudarnos en las dificultades que se presentan en las tareas designadas

¿Qué pudimos hacer de manera diferente?

- ✓ Mejorar la comunicación y reuniones con el usuario ya que en este sprint no se estuvo presente por parte del equipo de trabajo.
- ✓ Mejorar el ritmo de desarrollo.
- ✓ Mejorar el retraso en las entregas de las historias de usuario

¿Qué debemos mejorar?

- ✓ Mejor comunicación con los usuarios.
- ✓ Mejorar el ritmo de desarrollo.
- ✓ Cumplir con las historias de usuario propuestas en el sprint.

Finalmente vemos en la Tabla 69, el cierre del proyecto para este Sprint.

Tabla 69

Cierre de Sprint N°03

ID	HISTORIA DE USUARIO	RESPONSABLE	ESTADO	REVISIÓN
HU-035	Registrar sanción	Solis Vera Gabriel Leonard	Realizado	Conforme
HU-036	Aplicar sanción a unidad	Solis Vera Gabriel Leonard	Realizado	Conforme
HU-037	Registrar justificación	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-038	Visualizar de recorrido por unidad	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-039	Reporte de unidades trabajando por fecha	Zamora Granados Joel	Realizado	Conforme
HU-040	Listar sanciones por unidad	Zamora Granados Joel	Realizado	Conforme
HU-041	Reporte de retrasos de unidades	Zamora Granados Joel	Realizado	Conforme
HU-042	Reporte de recorrido diario	Zamora Granados Joel	No Realizado	Incompleto

Fuente: Elaboración Propia

La HU-042 no se llegó a realizar por la que fue considerada en el desarrollo del sprint 05.

3.5.5 Planificación.

Para este sprint se consideraron las historias de usuario que muestran la Tabla 70 y los prototipos.

Tabla 70

Sprint Backlog – Sprint N°03

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-029	Justificación de no disponibilidad de la unidad	Media	20
HU-030	Asignación de tarjeta a unidad	Alta	40
HU-031	Dar de baja tarjeta	Media	08
HU-032	Visualizar tarjeta	Alta	20
HU-033	Asignar recorrido a unidad	Alta	20
HU-034	Dar de baja a recorrido asignada a unidad	Media	20
Total, puntos estimados			128

Fuente: Elaboración Propia

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 71 a la Tabla 76

Tabla 71

HU: Justificación de no disponibilidad de la unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	029		
NOMBRE:	Justificación de no disponibilidad de la unidad		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar el motivo de porque una unidad no está trabajando</p> <p>Para: otorgarle su respectivo plazo de tiempo o sanción al no ser justificado.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Diseñar y crear tablas de la base de datos	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	8 h		
Realizar Pruebas y Corregir Errores	2 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Justificación			
<p>Dado que el usuario quiera poner inactivo una unidad por algún motivo de daño en la unidad.</p> <p>Cuando el usuario de clic en el botón “Mantenimiento”</p> <p>Entonces se mostrará un modal de registro de justificación y fecha de inactivo</p>			

Datos de Mantenimiento

Justificación:

Vehiculo tiene que realizar revisión tecnica y reparación de motor por lo cual se le brinda un plazo de 5 dias

Fecha de Mantenimiento

Dias de Plazo:

Figura 73 Prototipo del Sprint 03 – HU: Justificación de no disponibilidad de unidad
Fuente. Elaboración Propia

Tabla 72

HU: Asignación de tarjetas a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	030		
NOMBRE:	Asignación de tarjetas a unidad		
PRIORIDAD:	Alta	USUARIO:	Controlador
ESTIMACIÓN:	40	SPRINT:	03
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Asignar una tarjeta a la unidad.</p> <p>Para: que la unidad pueda conocer que rutas debe recorrer y sus tiempos.</p>			
OBSERVACIONES			
<p>*La tarjeta se debe entregar una sola vez, de otra manera se debe aplicar una sanción.</p> <p>* Datos a registrar: Placa de la unidad, fecha y hora de registro, rutas y tiempos a recorrer.</p>			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registro de tarjeta			
<p>Dado que la unidad ya tenga una tarjeta activa registrada ese día</p> <p>Cuando el controlador de clic en el botón buscar placa</p> <p>Entonces La aplicación mostrará un mensaje informando:” ya se ha registrado una tarjeta con anterioridad y no permitirá el registro “</p>			

Escenario 2: Búsqueda de placa.

Dado que el controlador ingresa una placa de una unidad no registrada.

Cuando el controlador de clic en buscar placa.

Entonces La aplicación mostrará un mensaje de placa no registrada.

Asignar tarjeta

Fecha actual: Número de tarjeta:

Placa:

Vehículo: Socio:

Ruta:

Hora de inicio de control

Punto de control	Hora de control	Tiempo desde inicio	Tiempo comodín
1	10:00 p.m.	15	2
2	10:15 p.m.	10	0

Figura 74 Prototipo del Sprint 03 – HU: Asignación de tarjetas a unidad
Fuente. Elaboración Propia

Tabla 73

HU: Dar de baja a la tarjeta

HISTORIA DE USUARIO			
IDENTIFICADOR:	031		
NOMBRE:	Dar de baja a la tarjeta		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	8	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Dar de baja a las tarjetas de control.</p> <p>Para: Poder deshabilitar una tarjeta registrada con errores.</p>			
OBSERVACIONES			
<p>* Se debe guardar el usuario que ha realizado la baja de la tarjeta.</p> <p>* Se debe registrar el motivo de la baja.</p>			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Motivo en blanco			
<p>Dado que se el campo de motivo de baja este en blanco</p> <p>Cuando el controlador de clic en el botón aceptar del modal dar de baja</p> <p>Entonces La aplicación mostrará un mensaje informando: “Que el motivo no puede estar en blanco”.</p>			
Escenario 2: Dar de baja a tarjeta			

<p>Dado: que el usuario quiera dar de baja a una tarjeta</p> <p>Cuando: el usuario seleccione el botón “Desactivar”.</p> <p>Entonces: La aplicación pasará la tarjeta a estado “Inactiva”, la cual será su estado final.</p>
<p>Escenario 3: Listar tarjetas por unidad</p>
<p>Dado que el usuario desee visualizar las tarjetas que posea una unidad.</p> <p>Cuando el usuario haga clic en el botón listar.</p> <p>Entonces el sistema listará la lista de tarjetas.</p>
<p>Escenario 4: Rango de fechas</p>
<p>Dado que el usuario desee visualizar las tarjetas que posea una unidad.</p> <p>Cuando el usuario haga clic en el botón listar.</p> <p>Entonces el sistema listará la lista de tarjetas de acuerdo con el rango de fechas ingresado.</p>

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 75 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta
Fuente. Elaboración Propia

Dar de baja tarjeta

Motivo:

Figura 76 Prototipo del Sprint 03 – HU: Dar de baja a la tarjeta
Fuente. Elaboración Propia

Tabla 74

HU: Visualizar tarjeta

HISTORIA DE USUARIO			
IDENTIFICADOR:	004		
NOMBRE:	Visualizar tarjeta		
PRIORIDAD:	Alta	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	01
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Visualizar los datos de una tarjeta</p> <p>Para: realizar alguna gestión importante con sus datos.</p>			
OBSERVACIONES			
Ninguna			
TAREAS			
Descripción		Duración (horas)	

Diseñar la interfaz	2 h
Construir la interfaz	2 h
Crear procedimientos almacenados en la base de datos	2 h
Crear el código correspondiente	8 h
Realizar Pruebas y Corregir Errores	4 h
CRITERIOS DE ACEPTACIÓN	
Escenario 1: Visualizar Datos	
<p>Dado: que el usuario quiera visualizar los datos de una tarjeta</p> <p>Cuando: el usuario de clic en el botón “Visualizar datos”</p> <p>Entonces: el sistema mostrará un modal con todos los datos de la tarjeta</p>	
Escenario 2: Imprimir datos de tarjeta	
<p>Dado: Que el usuario quiera imprimir los datos de una tarjeta.</p> <p>Cuando: el usuario de clic en el botón “Imprimir”</p> <p>Entonces: El sistema generar un archivo PDF para la posterior impresión de los datos de la tarjeta.</p>	

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 77 Prototipo del Sprint 03 – HU: Visualizar tarjeta

Fuente. Elaboración Propia



Figura 78 Prototipo del Sprint 03 – HU: Visualizar tarjeta
 Fuente. Elaboración Propia

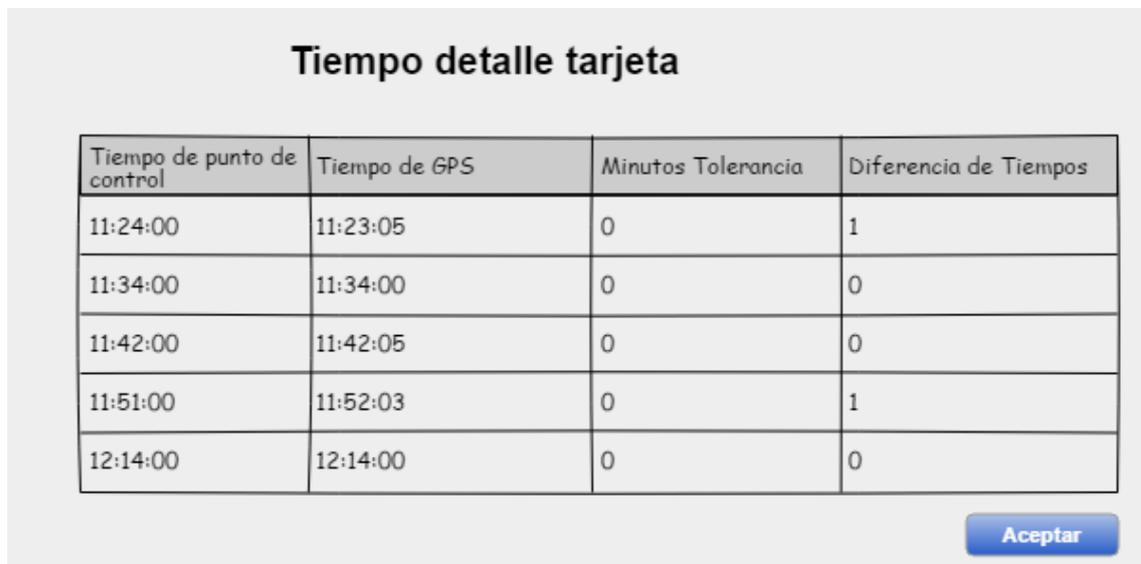


Figura 79 Prototipo del Sprint 03 – HU: Visualizar tarjeta
 Fuente. Elaboración Propia

Tabla 75

HU: Asignar recorrido a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	033		
NOMBRE:	Asignar recorrido a unidad		
PRIORIDAD:	Alta	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: Asignar rutas a la unidad.</p> <p>Para: Poder controlar los tiempos de cada unidad de acuerdo con cada ruta asignada.</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Diseñar y crear tablas de la base de datos		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Reasignación de ruta			
<p>Dado que el usuario desee asignar una ruta ya asignada a la unidad y esta aun no la recorre.</p> <p>Cuando el usuario dé clic en el botón “asignar ruta”.</p> <p>Entonces el sistema mostrara un mensaje de alerta indicando que no se puede reasignar una ruta que aún no se ha recorrido.</p>			

Asignar Ruta

Fecha actual:
 Número de tarjeta:
 Número de vuelta:

Placa:

Vehículo:
 Socio:

Ruta:

Hora de inicio de control

Punto de control	Hora de control	Tiempo desde inicio	Tiempo comodin
1	10:00 p.m.	15	2
2	10:15 p.m.	10	0

Figura 80 Prototipo del Sprint 03 – HU: Asignar recorrido a unidad
 Fuente. Elaboración Propia

Tabla 76

HU: Dar de baja recorrido asignado a unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	034		
NOMBRE:	Dar de baja recorrido asignado a unidad		
PRIORIDAD:	Media	USUARIO:	Controlador
ESTIMACIÓN:	20	SPRINT:	03
DESARROLLADORES:	Solis Vera Gabriel Leonardo		
DESCRIPCIÓN			
<p>Como: Controlador de la empresa el Chasqui S.A</p> <p>Quiero: poder modificar la ruta asignada a una unidad.</p> <p>Para: poder cambiar su recorrido y tiempo por causa de error</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción	Duración (horas)		
Diseñar la interfaz	2 h		
Construir la interfaz	2 h		
Crear procedimientos almacenados en la base de datos	2 h		
Crear el código correspondiente	6 h		
Realizar Pruebas y Corregir Errores	4 h		
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Especificar motivo			
<p>Dado Que se desee cambiar la ruta asignada de unidad</p> <p>Cuando el controlador de clic en el botón dar de baja ruta en el listado de rutas asignadas a unidad</p> <p>Entonces La aplicación mostrará un modal para escribir el motivo del cambio</p>			

Listar tarjetas por unidad

Desde Fecha: Hasta Fecha: Placa:

Tarjeta	Vehiculo	Socio	Fecha	Estado	Opciones
1	MPL001	Jorge Luis Cabrejos Gonzales	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Dar de baja"/>

Figura 81 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia

Visualizar tarjeta

Ruta	Número Vuelta	Opciones
A	1	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>
B	2	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>
C	3	<input type="button" value="Ver"/> <input type="button" value="Dar de baja ruta"/>

Figura 82 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia



Figura 83 Prototipo del Sprint 03 – HU: Dar de baja recorrido asignado a unidad
Fuente. Elaboración Propia

3.5.6 Tablero SCRUM.

A continuación se muestra el tablero SCRUM donde se ordenó las historias de usuario con sus respectivas tareas, conforme se iban desarrollando en el sprint.

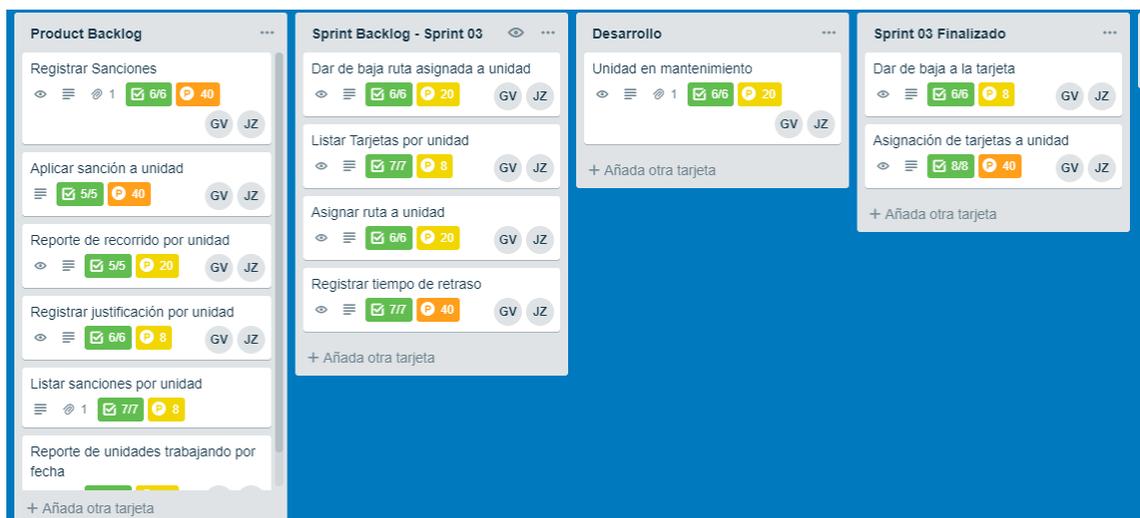


Figura 84 Tablero SCRUM del Sprint 03
Fuente. Elaboración Propia

3.5.7 Diagrama de Burndown del Sprint N° 03

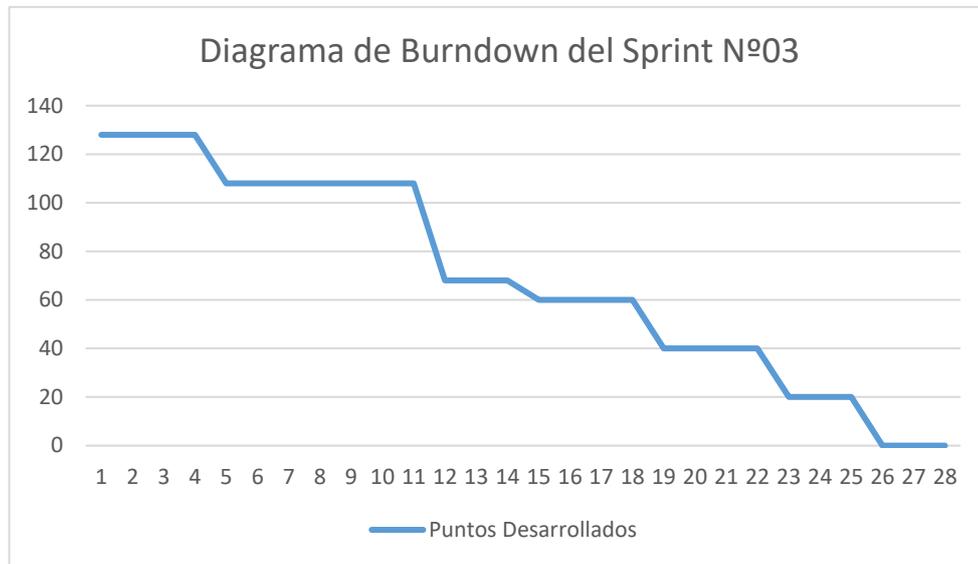


Figura 85 Diagrama de Burndown del Sprint 03

Fuente. Elaboración Propia

3.5.8 Retrospectiva

En esta etapa se evaluaron tres aspectos importantes; buenos, malos y aspectos de mejora:

¿Qué podríamos seguir manteniendo?

- ✓ Se logró comunicación constante con los usuarios y se cumplieron las visitas quincenales.
- ✓ Buen ritmo de desarrollo por parte del equipo.
- ✓ Cumplimiento total de las historias de usuario.
- ✓ Puntuación de historias de usuario más realistas.
- ✓ Se obtuvo puntualidad en las reuniones designadas.
- ✓ Se logró actualizar constantemente el tablero Scrum.

¿Qué pudimos hacer de manera diferente?

- ✓ Coordinar tiempos para la configuración y mantenimiento del sistema
- ✓ Avisar los inconvenientes por lo cual no se logró avanzar correctamente las tareas asignadas.

¿Qué debemos mejorar?

- ✓ Dedicar más tiempo para la configuración y mantenimiento del sistema.
- ✓ Comunicar con tiempo los problemas que se van teniendo con el desarrollo de algunas tareas para podernos ayudar en equipo.

Finalmente vemos en la Tabla 77, el cierre del proyecto para este Sprint.

Tabla 77

Cierre de Sprint N°02

ID	HISTORIA DE USUARIO	RESPONSABLE	ESTADO	REVISIÓN
HU-029	Justificación de no disponibilidad de la unidad	Zamora Granados Joel	Realizado	Conforme
HU-030	Asignación de tarjeta a unidad	Zamora Granados Joel	Realizado	Conforme
HU-031	Dar de baja tarjeta	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-032	Visualizar tarjeta	Zamora Granados Joel	Realizado	Conforme
HU-033	Asignar recorrido a unidad	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-034	Dar de baja a recorrido asignada a unidad	Solis Vera Gabriel Leonardo	Realizado	Conforme

Fuente: Elaboración Propia

3.6 Sprint N°05

3.6.1 Planificación.

Para este sprint se consideraron las historias de usuario que muestran la Tabla 78 y los prototipos

Tabla 78

Sprint Backlog - Sprint N°05

ID	HISTORIA DE USUARIO	PRIORIDAD	ESTIMACIÓN
HU-042	Reporte de recorrido diario	Baja	20
HU-043	Registro GPS	Media	20
HU-044	Pago de unidad	Baja	13
HU-045	Comparación de recorridos	Media	20
HU-046	Reporte promedio de recorrido	Media	20
HU-047	Listar GPS	Media	05
HU-048	Autenticar Usuarios	Baja	08
Total, puntos estimados			106

Fuente: Elaboración Propia

A continuación, se detalla cada historia de usuario para este Sprint. Ver Tabla 79 a la Tabla 84

Tabla 79

HU: Registro de GPS

Desarrollo de la historia de usuario (Ver apéndice A).

HISTORIA DE USUARIO			
IDENTIFICADOR:	43		
NOMBRE:	Registro de GPS		
PRIORIDAD:	Medio	USUARIO:	Secretaria
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Registrar el GPS de cada unidad</p> <p>Para: Poder llevar un correcto control de cada GPS</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Registrar GPS			
<p>Dado: que el usuario quiera registrar un IME de un GPS ya registrado.</p> <p>Cuando: el usuario de clic en el botón “Registrar GPS”.</p> <p>Entonces: La aplicación mostrará un mensaje de error de GPS.</p>			

Registro de GPS

Nombre GPS:

ID del GPS:

Unidad:

 ▼

Figura 86 Prototipo del Sprint 05 – HU: Registro de GPS
Fuente. Elaboración Propia

Tabla 80

HU: Pago de unidad

HISTORIA DE USUARIO			
IDENTIFICADOR:	44		
NOMBRE:	Pago de unidad		
PRIORIDAD:	Baja	USUARIO:	Secretaria
ESTIMACIÓN:	13	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: saber que unidad a cancelado su cuota diaria</p> <p>Para: Poder habilitar a la unidad para que pueda trabajar</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: No se canceló cuota diaria			
<p>Dado Que la unidad no haya cancelado su cuota diaria</p> <p>Cuando se le quiera asignar una tarjeta para un nuevo día</p> <p>Entonces el sistema no permitirá asignarle una tarjeta a la unidad.</p>			

Listar tarjetas

Desde Fecha: Hasta Fecha:

Filtro de tarjetas:

Tarjetas Activas

Tarjeta	Vehiculo	Fecha	Estado	Opciones
1	MPL001	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Registrar Pago"/>
2	ABC123	15/05/2018	Activa	<input type="button" value="Ver"/> <input type="button" value="Eliminar Pago"/>

Figura 87 Prototipo del Sprint 05 – HU: Pago de unidad
Fuente. Elaboración Propia

Actualizar pago de tarjeta

Figura 88 Prototipo del Sprint 05 – HU: Pago de unidad
Fuente. Elaboración Propia

Actualizar pago de tarjeta

Figura 89 Prototipo del Sprint 05 – HU: Pago de unidad
Fuente. Elaboración Propia

Tabla 81

HU: Comparación de recorridos

HISTORIA DE USUARIO			
IDENTIFICADOR:	45		
NOMBRE:	Comparación de recorridos		
PRIORIDAD:	Media	USUARIO:	Gerente
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Saber que unidades se encuentran recorriendo cada ruta</p> <p>Para: Poder tomar las decisiones correspondientes con esa información</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		8 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Visualizar más de una unidad			
<p>Dado: que usuario quiera visualizar dos unidades con sus respectivos recorridos simultáneamente.</p> <p>Cuando seleccione el botón visualizar recorrido</p> <p>Entonces el sistema solo permitirá la visualización de recorrido por unidad.</p>			

Reporte de servicio de rutas

Fecha : Hora: [Listar recorridos actuales](#)

Ruta A
M1W-792 - 10:00:00 - 10:48:00
Ruta B
ABC-123 - 10:20:00 - 12:00:00
Ruta C
MPL-456 - 10:00:00 - 11:30:00

Figura 90 Prototipo del Sprint 05 – HU: Comparación de recorridos
Fuente. Elaboración Propia

Tabla 82

HU: Reporte de promedio de recorrido de rutas

HISTORIA DE USUARIO			
IDENTIFICADOR:	46		
NOMBRE:	Reporte de promedio de recorrido de rutas		
PRIORIDAD:	Media	USUARIO:	Gerente
ESTIMACIÓN:	20	SPRINT:	04
DESARROLLADORES:	Zamora Granados Joel Alfredo		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Conocer el promedio de recorridos por cada ruta</p> <p>Para: Poder distribuir las unidades a fin de brindar un mejor servicio</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Promedio según opción ingresada			
<p>Dado que el usuario quiera visualizar el promedio de un día, mes o año específico</p> <p>Cuando se presiones el botón correspondiente</p> <p>Entonces se mostrará el índice de recorrido según la opción elegida.</p>			
Escenario 2: Promedio solo de rutas vigentes.			

Dado que el usuario quiera visualizar el promedio de las rutas y una de ellas este dada de baja

Cuando se presiones el botón correspondiente

Entonces se mostrará solo el índice de recorrido de las rutas vigentes.



Figura 91 Prototipo del Sprint 05 – HU: Reporte de promedio de recorrido de rutas

Fuente. Elaboración Propia

Tabla 83

HU: Listado de GPS

HISTORIA DE USUARIO			
IDENTIFICADOR:	47		
NOMBRE:	Listado de GPS		
PRIORIDAD:	Media	USUARIO:	Secretaria
ESTIMACIÓN:	5	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel		
DESCRIPCIÓN			
<p>Como: Secretaria de la empresa el Chasqui S.A</p> <p>Quiero: Conocer el listado de GPS y la unidad que lo posee</p> <p>Para: Poder llevar un correcto control</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: unidades sin GPS registrado			
<p>Dado que el usuario quiera visualizar las unidades con sus GPS registrados y no haya ninguna unidad con GPS</p> <p>Cuando seleccione la opción correspondiente</p> <p>Entonces se mostrará un mensaje “No hay unidades con GPS registradas”</p>			

Listar GPS

Filtro de GPS:

GPS Asignados ▼ Listar

Número	Vehiculo	GPS	Estado
1	MPL001	23654178	Activa
2	ABC123	98563218	Inactivo

Figura 92 Prototipo del Sprint 05 – HU: Listado de GPS
Fuente. Elaboración Propia

Tabla 84

HU: Autenticación de usuarios

HISTORIA DE USUARIO			
IDENTIFICADOR:	48		
NOMBRE:	Autenticación de Usuarios		
PRIORIDAD:	Baja	USUARIO:	Gerente
ESTIMACIÓN:	8	SPRINT:	04
DESARROLLADORES:	Solis Vera Gabriel		
DESCRIPCIÓN			
<p>Como: Gerente de la empresa el Chasqui S.A</p> <p>Quiero: Que el sistema autentique usuarios</p> <p>Para: Que cada uno de ellos trabajen con las funciones que les correspondan</p>			
OBSERVACIONES			
Ninguna.			
TAREAS			
Descripción		Duración (horas)	
Diseñar la interfaz		2 h	
Construir la interfaz		2 h	
Crear procedimientos almacenados en la base de datos		2 h	
Crear el código correspondiente		6 h	
Realizar Pruebas y Corregir Errores		4 h	
CRITERIOS DE ACEPTACIÓN			
Escenario 1: Usuario autenticado			
<p>Dado que el usuario sea autenticado correctamente</p> <p>Cuando ingrese a la pantalla principal</p> <p>Entonces se mostrará en el menú solo las opciones que le correspondan según su tipo de usuario</p>			

Sistema de registro de unidades y rutas

Usuario:

Contraseña:

Ingresar

Recuperar contraseña

Figura 93 Prototipo del Sprint 05 – HU: Autenticación de usuarios
Fuente. Elaboración Propia

3.6.2 Tablero SCRUM.

A continuación se muestra el tablero SCRUM donde se ordenó las historias de usuario con sus respectivas tareas, conforme se iban desarrollando en el sprint.



Figura 94 Tablero SCRUM del Sprint 05
Fuente. Elaboración Propia

3.6.3 Diagrama de Burndown del Sprint N° 05



Figura 95 Diagrama de Burndown del Sprint 05

Fuente. Elaboración Propia

3.6.4 Retrospectiva

En el proceso de desarrollo de este sprint, no se encontraron inconvenientes; también se resalta que el tiempo de estimación con el tiempo real de implementación han logrado ser bastantes similares a diferencia de los anteriores sprint, debido a que se ha ido ganando experiencia en la estimación y se conoce mejor el tiempo que tarda en realizarse una funcionalidad. Finalmente vemos en la Tabla 85, el cierre del proyecto para este Sprint.

Tabla 85

Cierre de Sprint N°05

ID	Historia de Usuario	Responsable	Estado	Revisión
HU-042	Reporte de recorrido diario	Zamora Granados Joel	Realizado	Conforme
HU-043	Registro GPS	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-044	Pago de unidad	Zamora Granados Joel	Realizado	Conforme

HU-045	Comparación de recorridos	Zamora Granados Joel	Realizado	Conforme
HU-046	Reporte promedio de recorrido	Zamora Granados Joel	Realizado	Conforme
HU-047	Listar GPS	Solis Vera Gabriel Leonardo	Realizado	Conforme
HU-048	Autenticar Usuarios	Solis Vera Gabriel Leonardo	Realizado	Conforme

Fuente: Elaboración Propia

Diagrama de componentes

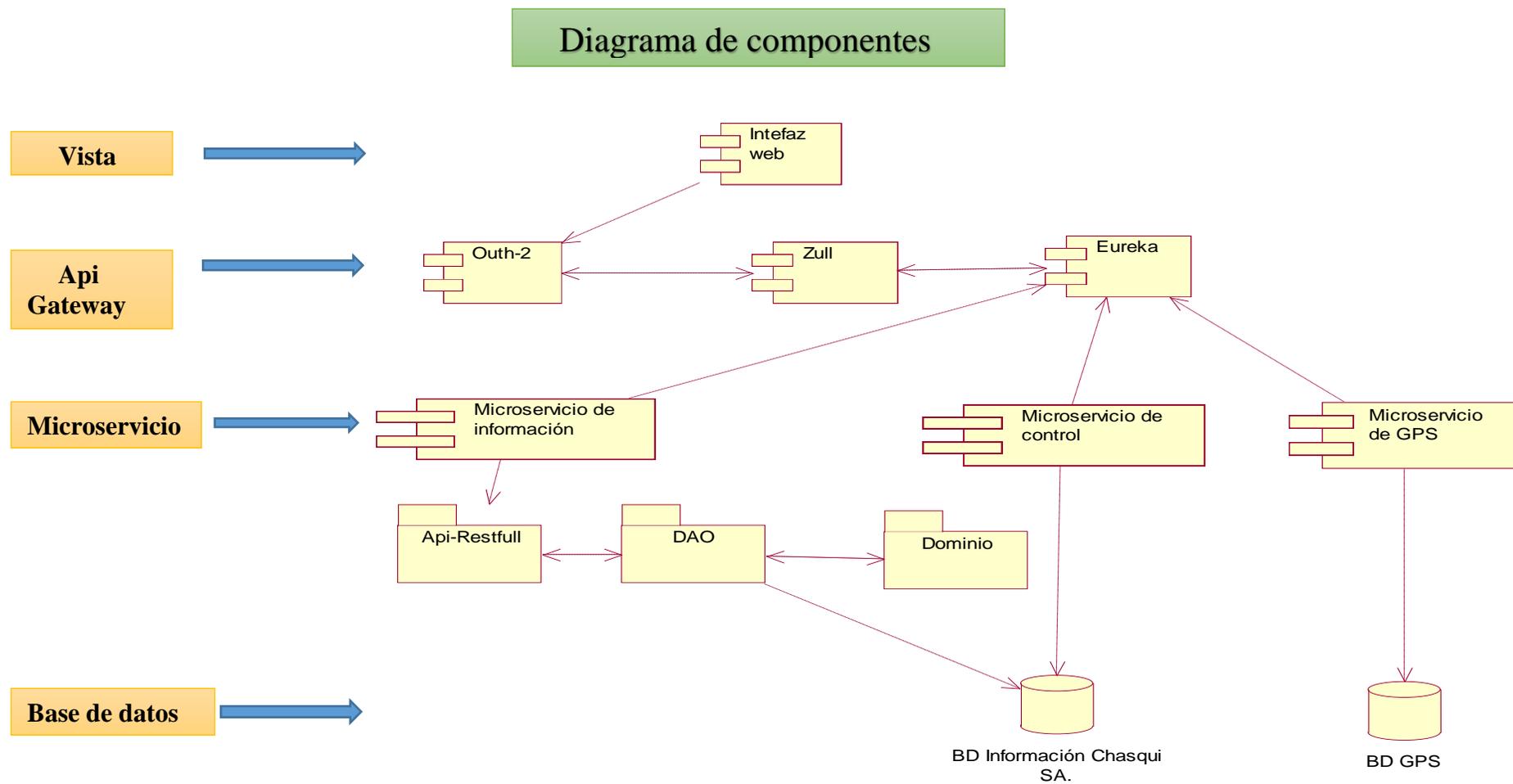


Figura 97 Diagrama de Componentes

Fuente. Elaboración Propia

3.7 Diccionario de la base de datos

A continuación, se describen los campos que integran el diccionario de base de datos. Ver Tabla 86 hasta Tabla 101

Tabla 86

Base de datos – Tabla calleruta

Tabla: calleruta				
Tabla en la que se registran todos los puntos geográficos que permiten trazar la ruta así como datos informativos				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que hace referencia al punto geográfico por donde se traza la ruta.	Not null
codigoruta	smallint	Foránea	Código que hace referencia a la ruta que pertenece el punto geográfico.	Not null
numeroorden	smallint		Campo que permitirá trazar el recorrido.	Not null
direccion	varchar(200)		Dirección aproximada de la calle.	Null

puntoinicio	point	Latitud y longitud de la calle.	Not null
vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 87

Base de datos – Tabla detalletarjeta

Tabla: detalletarjeta

Tabla en la que se registran la ruta que recorrerá la unidad

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	serial	Primaria	Código con el que se identifica el detalle de la tarjeta.	Not null
codigotarjeta	integer	Foránea	Código de la tarjeta al que corresponde el detalle.	Not null
codigoruta	smallint	Foránea	Código que permite identificar la ruta recorrida.	Not null
horainicio	time		Hora en la que se empezó el recorrido.	Not null

mensajebaja	varchar(300)	Detalle de el por qué se dio de baja la tarjeta. El valor guardado por defecto es null.	Null
nombreruta	varchar	Nombre de la ruta que se está recorriendo.	Not null
numerovuelta	smallint	Numero de vuelta o recorrido en el día.	Not null
vigencia	boolean	Activa (true) – Inactiva (false). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 88

Base de datos – Tabla GPS

Tabla: GPS

Tabla en la que se registran todos los puntos geográficos que permiten trazar la ruta así como datos informativos

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que permite identificar el gps.	Not null

codigoreferencia	varchar(7)	Código generado al registrar el GPS en traccar. El valor guardado por defecto es null.	Null
vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 89

Base de datos – Tabla GPSunidad

Tabla: Gpsunidad

Tabla intermedia entre unidad y GPS en la que se asignará un GPS activo a cada unidad

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que permite identificar la asignación entre unidad y gps.	Not null
codigounidad	smallint	Foránea	Código que permite identificar la unidad asignada al gps.	Not null
codigogps	smallint	Foránea	Código que permite identificar el gps asignado a la unidad.	Not null

vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null
-----------------	---------	--	----------

Fuente: Elaboración Propia.

Tabla 90

Base de datos – Tabla persona

Tabla: persona

Tabla base en la que se registran los datos principales de una persona

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
Código	smallserial	Primaria	Código con que se identifica a la persona	Not null
nombres	varchar(32)		Nombre de la persona.	Not null
apellidopaterno	varchar(32)		Apellido paterno de la persona.	Not null
apellidomaterno	varchar(32)		Apellido materno de la persona.	Not null
correo	varchar(76)		Correo de la persona que sirve de contacto. El valor guardado por defecto es null.	Null

dirección	varchar(76)	Dirección.	Not null
sexo	char(1)	Sexo.	Not null
dni	char(8)	DNI.	Not null
fechanacimiento	date	Fecha de nacimiento.	Not Null
estadocivil	varchar(10)	Estado civil.	Not Null
celular	char(9)	Celular. El valor guardado por defecto es null.	Null.

Fuente: Elaboración Propia.

Tabla 91

Base de datos – Tabla personal

Tabla: personal				
Se registrará el personal que labora en la empresa				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	Smallint	Primaria - Foránea	Código con que se identifica al personal	Not null

fecharegistro	date	Fecha en la que ingresó a la empresa	Not null
fecharetiro	date	Fecha en la que se retiró de la empresa. El valor por defecto es el valor que se guarda en el campo fecharegistro de la misma tabla.	Not null
Cargo	varchar(32)	Cargo en la empresa	Not null
vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 92

Base de datos – Tabla puntocontrol

Tabla: puntocontrol

Tabla en donde se registran todos los puntos de control que posea la empresa con su ubicación geográfica

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que permite identificar el punto de control.	Not null

puntogeografico	point	Latitud y longitud de la ubicación del punto de control.	Not null
punto2	point	Campo que guarda la posición anterior del punto de control. El valor guardado por defecto es null.	Null
direccion	varchar(100)	Dirección aproximada del punto de control.	Not null
vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 93

Base de datos – Tabla registrounidad

Tabla: registrounidad

Tabla intermedia entre unidad y socio

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
-----------------	---------------------	--------------	--------------------	-----------------

codigo	smallserial	Primaria	Código que permite identificar la asignación de unidad y socio.	Not null
codigounidad	smallint	Foránea	Código que permite identificar a la unidad asignada.	Not null
codigosocio	smallint	Foránea	Código que permite identificar al socio asignado.	Not null
fecharegistro	date		Fecha en la que se registró la unidad	Not null
fecharetiro	date		Fecha en la que se retiró la unidad. El valor guardado por defecto es el mismo valor que se guarda en el campo fecharegistro de la misma tabla.	Not null
vigencia	boolean		True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 94

Base de datos – Tabla ruta

Tabla: ruta				
Tabla en donde se registran todas las rutas que posee la empresa				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que permite identificar la ruta	Not null
letra	char(1)		Nombre de la ruta.	Not null
tiempo	smallint		Tiempo total de recorrido de la ruta - el tiempo es calculado por la suma de los tiempos establecidos.	Null
estado	char(1)		A (Con servicio) – I (Sin servicio). Las rutas sin servicio son aquellas rutas adquiridas por la empresa pero que aún no son recorridas por las unidades.	Not null
vigencia	boolean		True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 95

Base de datos – Tabla sanción

Tabla: sanción				
Se registrarán las sanciones y justificaciones por retraso de una unidad				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	smallserial	Primaria	Código que permite identificar una sanción.	Not null
codigodetalletarjeta	integer	Foránea	Código que permite identificar en que detalle sucedió la sanción.	Not null
codigoregistrounidad	smallint	Foránea	Código que permite identificar a que unidad se le aplicará la sanción.	Not null
estado	varchar(1)		A (Aplicada) – I (No aplicada).	Not null
justificacion	varchar(300)		Justificación de la sanción. El valor guardado por defecto es null.	Null
motivo	varchar(300)		Motivo por el cual se registra la sanción. El valor guardado por defecto es null.	Null

sanciontiempo	real	Tiempo de penalidad registrado por el controlador.	Null
vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 96

Base de datos – Tabla socio

Tabla: socio

Tabla en la que se registran todos los socios de la empresa

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigopersona	smallint	Primaria - Foránea	Código que identifica la persona que es socio.	Not null

Número de acciones por cada socio			
numeroacciones	smallint	aumenta cada vez que se registre una nueva unidad.	Not null
fecharegistro	date	Fecha de registro del socio en la empresa.	Not null
fecharetiro	date	Fecha en la que se retiró de la empresa.	Not null
nombresapellidospariente	varchar(120)	Nombres y apellidos de familiar. El valor guardado por defecto es null.	Null
telefonopariente	char(9)	Teléfono pariente. El valor guardado por defecto es null.	Null
estado	varchar(1)	A (Activo) – (I) Retirado. El valor guardado por defecto es A.	
vigencia	boolean	True (Datos válidos) – False (Datos inválidos).	Not null

Fuente: Elaboración Propia.

Tabla 97

Base de datos – Tabla tarjeta

Tabla: tarjeta				
Tabla en la que se registra todas las rutas recorridas en un día por una unidad, está asociada a una unidad				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	serial	Primaria	Código que permite identificar la tarjeta.	Not null
codigoregistrounidad	integer	Foránea	Código que permite identificar a que unidad pertenece la tarjeta.	Not null
codigousuario	smallint	Foránea	Código que permite conocer el usuario que registra y realiza operaciones en la tarjeta.	Not null
fecha	date		Fecha de registro.	Not null
numero	varchar		Número de la tarjeta registrada por el controlador.	Not null
mensaje	varchar(200)		Campo utilizado para detallar el por qué se le dio de baja a la tarjeta. El valor guardado por defecto es null.	Null

vigencia	boolean	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null
-----------------	---------	--	----------

Fuente: Elaboración Propia.

Tabla 98

Base de datos – Tabla tiempodetalletarjeta

Tabla: tiempodetalletarjeta

Tabla en la que se registran los tiempos calculados de los puntos de control, los retrasos y los tiempos que capta el GPS según los puntos de control

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
Código	serial	Primaria	Código que identifica el tiempo del recorrido de la unidad.	Not null

Codigodetalletarjeta	integer	Foránea	Código que permite conocer a que detalle pertenece el tiempo.	Not null
codigotiempoestablecido	smallint	Foránea	Código que permite conocer el tiempo establecido.	Not null
Diferencia	real		Hora de control menos hora GPS. El valor guardado por defecto es null.	Null
horagps	time		Hora que marcó el GPS en la posición del punto de control. El valor guardado por defecto es null.	Null
Horacontrol	time		Hora actual más tiempo de punto control establecido.	Not null
Minutostolerancia	smallint		Minutos de tolerancia para el retraso.	Not null

Fuente: Elaboración Propia.

Tabla 99

Base de datos – Tabla tiempoestablecido

Tabla: tiempoestablecido				
Tabla intermedia entre ruta y punto de control, se registra los tiempos de recorrido según la ruta				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	Smallserial	Primaria	Código que permite identificar el tiempo establecido.	Not null
codigoruta	smallint	Foránea	Código que permite conocer la ruta a la que pertenece el tiempo establecido.	Not null
codigopuntocontrol	smallint	Foránea	Código que permite identificar el punto de control al que pertenece el tiempo establecido.	Not null
tiempoestablecido	smallint		Tiempo en minutos que debe demorar la unidad en llegar desde el inicio de la ruta hasta el punto de control.	Not null

nombresapellidospariente	varchar(120)	Nombres y apellidos de familiar.	Not null
orden	smallint	Orden en la que se toma en cuenta el punto de control.	Not null
vigencia	bool	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 100

Base de datos – Tabla unidad

Tabla: unidad

Tabla intermedia entre ruta y punto de control, se registra los tiempos de recorrido según la ruta

ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigo	serial	Primaria	Código que permite identificar a la unidad	Not null
placa	varchar(7)		Placa de la unidad	Not null

marca	varchar(60)	Marca de la unidad	Not null
modelo	varchar(15)	Modelo de la unidad	Not null
capacidad	smallint	Capacidad	Not null
aniofabricacion	smallint	Año de fabricación	Not null
color	varchar(10)	Color de la unidad	Not null
estado	char(1)	Activo(A) - Inactivo(I) - Mantenimiento(M)	Not null
vigencia	bool	True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

Tabla 101

Base de datos – Tabla usuario

Tabla: usuario				
Tabla de registro de usuarios de personal para el uso de la plataforma web				
ATRIBUTO	TIPO DE DATO	CLAVE	DESCRIPCIÓN	NULLABLE
codigopersonal	smallint	Primaria – Foránea	Código que permite identificar a que persona pertenece el usuario	Not null
usuario	varchar(32)		Nombre o nick del usuario	Not null
contrasenia	varchar(60)		Contraseña del usuario	Not null
vigencia	bool		True (Activo) – False (Inactivo). El valor guardado por defecto es true.	Not null

Fuente: Elaboración Propia.

3.8 Creación de Funciones

Tabla 102

Base de datos – Creación de funciones

ID	NOMBRE	TIPO DE DATOS RESULTANTE	ARGUMENTOS	DESCRIPCIÓN
P01	pr_acalleruta	void	p_codigocalle integer, p_direccion character varying, p_numeroorden integer	Se actualizan los datos de una calle perteneciente a una ruta
P02	pr_apersonal	void	p_codigo integer, p_nombres character varying, p_apellidopaterno character varying, p_apellidomaterno character varying, p_correo character varying, p_direccion character varying, p_dni character varying, p_sexo character varying, p_fechanacimiento character varying, p_celular character varying, p_estadocivil character varying, p_fecharegistro character varying, p_cargo character varying	Actualiza los datos del personal
P03	pr_apersonalestad o	void	p_codigo integer, p_vigencia boolean, p_fecharetiro character varying	Actualiza el estado de un personal y si cambia a

				false se guarda la fecha de registro
				Se actualizan los datos del punto de control y si se cambia de posición se guarda la posición antigua en el punto2
	pr_apuntocontrol	void	p_codigo integer, p_latitud double precision, p_longitud double precision, p_direccion character varying, p_punto2 character varying, p_vigencia boolean	
P04	pr_apuntocontrolv igencia	void	p_codigo integer, p_vigencia boolean	Se activa o se da de baja al punto de control
P05	pr_arestrounida d	void	p_codigo integer, p_codigosocio integer, p_fecharegistro character varying, p_placa character varying, p_soat character varying, p_marca character varying, p_modelo character varying, p_capacidad integer, p_aniofabricacion integer, p_color character varying, p_estado character	Actualiza los datos de una unidad y la tabla restrounidad
P06	pr_arestrounida dvigencia	void	p_codigo integer, p_vigencia boolean, p_fechaotiro character varying	Actualiza la vigencia de una unidad (True-false)
P07	pr_aruta	void	p_codigo ruta integer, p_letra character varying, p_estado character varying	Se actualizan los datos de la ruta

P08	pr_asocio	void	p_codigo integer, p_nombres character varying, p_apellidopaterno character varying, p_apellidomaterno character varying, p_correo character varying, p_direccion character varying, p_dni character, p_sexo character, p_fechanacimiento character varying, p_celular character, p_estadocivil character varying, p_fecharegistro character varying, p_nombrespriente character varying, p_telefonopriente character, p_estado character	Actualizará los datos de un socio
P09	pr_asocioacciones	void	p_codigo integer, p_numeroacciones integer	Se actualizan las acciones del socio por el código
P10	pr_asocioestado	void	p_codigo integer, p_estado character, p_fecha retiro character varying	Actualiza el estado del socio A(activo) - R(retirado) si es retirado registra la fecha de retiro
P11	pr_atiempoestable cido	void	p_codigo integer, p_codigoruta integer, p_tiempoestablecido integer, p_orden integer	Actualiza los tiempos establecidos

P12	pr_atiempogspsde talletarjeta	void	p_codigotiempodetalle integer, p_horagps character varying, p_diferencia double precision	Actualiza la tabla tiempoDetalleTarjeta para poder registrar los tiempos GPS y la diferencia
P13	pr_aumentartiemp oruta	void	p_codigoruta integer, p_tiempo integer	Aumenta el tiempo total o tiempo de recorrido de una ruta
P14	pr_ausuario	void	p_codigo integer, p_nuevacontrasenia character varying	Se actualiza la contraseña del usuario
P15	pr_buscarusuari ombre	TABLE (p_codigo integer, p_usuario character varying, p_clave character varying, p_cargo character varying, p_indicador smallint, p_vigencia boolean)	p_usuarioe character varying	Lista a un usuario según su nombre
P16	pr_comprobardni	boolean	p_dni character	Comprueba si un DNI ya ha sido registrado con anterioridad

P17	pr_comprobarplaca	boolean	p_placa character varying	Comprueba si una placa se encuentra ya registrada
P18	pr_comprobarruta	boolean	p_letra character	Comprueba si el nombre de la ruta ya se encuentra registrado
P19	pr_ecalleruta	void	p_codigocalle integer	Da de baja a una calle perteneciente a una ruta
P20	pr_edetalletarjeta	void	p_codigodetalletarjeta integer, p_mensajebaja character varying	Da de baja a un detalle tarjeta
P21	pr_eruta	void	p_codigoruta integer	Da de baja a una ruta
P22	pr_etarjeta	void	codigotarjeta integer, mensajebaja character varying, codigousuario integer	Da de baja a una tarjeta

P23	pr_etiempoestable cido	void	p_codigo integer, p_codigoruta integer	Se da de baja a un tiempo establecido
P24	pr_eusuario	void	p_codigo smallint, p_vigencia boolean	Se da de baja a un usuario
P25	pr_icalleruta	smallint	p_codigoruta integer, p_direccion character varying, p_latitud double precision, p_longitud double precision, p_numeroorden integer, p_vigencia boolean	Registra las calles del recorrido de la ruta
P26	pr_idetalletarjeta	TABLE (p_codigodetalletarjeta integer)	p_codigotarjeta integer, p_numerovuelta integer, p_codigoruta integer, p_horainicio character varying, p_nombreruta character varying, p_vigencia boolean	Registra el detalle de la tarjeta
P27	pr_igps	TABLE (codigo smallint, codigoreferencia character varying)	p_codigounidad integer	Se registra la asignación de un dispositivo GPS a una unidad
P28	pr_ipersonal	void	p_nombres character varying, p_apellidopaterno character varying, p_apellidomaterno character varying, p_correo character varying, p_direccion character varying, p_dni character, p_sexo character, p_fechanacimiento character	Registra personal que labora en la empresa

			varying, p_celular character, p_estadocivil character varying, p_fecharegistro character varying, p_cargo character varying	
P29	pr_ipuntocontrol	integer	p_latitud double precision, p_longitud double precision, p_direccion character varying	Registrará los puntos de control
P30	pr_iregistrounidad	void	p_codigosocio integer, p_codigounidad integer, p_fecharegistro character varying, p_fecharetiro character varying, p_vigencia boolean	Registrará datos en la tabla intermedia registro unidad
P31	pr_iruta	integer	p_letra character varying, p_estado character varying, p_tiempo integer, p_vigencia boolean	Registro de ruta
P32	pr_isancion	void	p_codigodetalletarjeta integer, p_codigoregistrounidad integer, p_motivo character varying, p_tiemprotraso integer, p_sanciontiempo real, p_sancionmonto real	Registra una sanción o una justificación
P33	pr_isocio	void	p_nombres character varying, p_apellidopaterno character varying, p_apellidomaterno character varying, p_correo character varying, p_direccion character varying, p_dni character, p_sexo character, p_fechanacimiento character varying, p_celular character, p_estadocivil character varying,	Procedimiento que registrará los datos en la tabla persona y socio

			p_fecharegistro character varying, p_nombrespaciente character varying, p_telefonopaciente character, p_estado character	
P34	pr_itarjeta	integer	p_numero character varying, p_fecha character varying, p_usuario character varying, p_codigoru integer, p_vigencia boolean	Registrará la tarjeta asignada a una unidad
P35	pr_itiempoestablecido	void	p_codigoruta integer, p_codigopunto integer, p_tiempo integer, p_orden integer, p_vigencia boolean	Insertar tiempo establecido
P36	pr_itiemposdetalletarjeta	void	p_codigodetalletarjeta integer, p_codigotiempoestablecido integer, p_minutostolerancia integer, p_horacontrol character varying	Registro de los tiempos calculados para el detalle tarjeta
P37	pr_iunidad	integer	p_placa character varying, p_soat character varying, p_marca character varying, p_modelo character varying, p_capacidad integer, p_aniofabricacion integer, p_color character varying, p_estado character, p_vigencia boolean	Registro de unidad
P38	pr_iusuario	character varying	p_codigopersonal integer, p_vigencia boolean	Registra un usuario

		TABLE (p_codigo smallint,			Lista todas las calles
P39	pr_licallesrutavig entes	p_punto point, p_direccion character varying, p_numeroorden smallint, p_vigencia boolean)		p_codigoruta integer	rutas activas
		TABLE (codigotarjeta integer, codigoruta smallint, numerovuelta			Lista todos los detalle de
P40	pr_lidetalletarjeta	smallint, nombreruta character varying, codigodetalletarjeta integer, horainicio time without time zone)		p_codigotarjeta integer	la tarjeta
		TABLE (codigopersona smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, dni			Lista personal según la
P41	pr_lipersonalestad oapellido	character, direccion character varying, cargo character varying, vigencia boolean)		p_vigencia character varying, p_apellidos character varying	vigencia y apellido
		TABLE (p_codigo integer, p_punto			Lista los puntos de
P42	pr_lipuntoscontrol	point, p_direccion character varying, p_punto2 point, p_vigencia boolean)		p_estado character varying	control por estado

P43	pr_lipuntoscontrol vigentes	TABLE (p_codigo integer, p_punto point, p_direccion character varying, p_punto2 point, p_vigencia boolean)	Lista los puntos de control vigentes
P44	pr_lirutasestado	TABLE (p_codigo smallint, p_tiempo smallint, p_letra character, p_estado character, p_vigencia boolean) p_estadorecibido character varying	Lista todas las rutas según el estado recibido
P45	pr_lirutasvigentes	TABLE (p_codigo smallint, p_letra character, p_estado character, p_vigencia boolean)	Lista todas las calles rutas activas
P46	pr_lisocios	TABLE (codigopersona smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, correo character varying, dni character, direccion character varying, sexo character, estadocivil character varying, celular character, fechanacimiento date, fecharegistro	Listará todos los socios

		date, fecharetiro date, numeroacciones smallint, nombrespariente character varying, estado character, vigencia boolean) TABLE (codigopersona smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, dni character, direccion character varying, numeroacciones smallint, estado character) TABLE (p_codigo integer, p_codigoru integer, p_numerotarjeta character varying, p_fechatarjeta date, p_mensaje character varying, p_vigencia boolean) TABLE (p_codigo integer, p_codigoru integer, p_numerotarjeta character varying, p_fechatarjeta date,			
P47	pr_lisocioestado orapellido		p_estado character, p_apellidos character varying	Listará todos los socios según el estado(A-R) o el apellido	
P48	pr_litarjeta		p_codigorunidad integer, p_fecha character varying	Lista solo las tarjetas activas por fecha y codigo registro unidad	
P49	pr_litarjetaestado		fechatarjetainicio character varying, fechatarjetafin character varying, vigenciatarjeta boolean	Lista las tarjetas segun el estado recibido	

		p_mensaje character varying, p_vigencia boolean) TABLE (p_codigo integer, p_codigoru integer, p_numerotarjeta		
P50	pr_litarjetafechas	character varying, p_fechatarjeta date, p_mensaje character varying, p_vigencia boolean) TABLE (codigo integer, codigodetalletarjeta integer, codigotiempoestablecido smallint,	p_codigorunidad integer, p_desdefecha character varying, p_hastafecha character varying	Lista las tarjetas según la unidad y el rango de fechas
P51	pr_litiemposdetall etarjeta	horacontrol time without time zone, horagps time without time zone, horainicio time without time zone, diferencia real, minutostolerancia smallint) TABLE (p_codigo smallint,	p_codigodetalletarjeta integer	Lista los tiempos del detalle tarjeta de un detalle tarjeta especifico
P52	pr_litiemposvigen tesruta	p_codigoruta smallint, p_codigopuntocontrol smallint, p_letra character, p_direccion character	e_codigoruta integer	Lista de tiempos vigentes según la ruta.

		varying, p_tiempoestablecido smallint, p_orden smallint, p_punto point, p_vigencia boolean) TABLE (p_codigotarjeta integer, p_codigodetalletarjeta integer, p_codigotiempodetalletarjeta integer, p_codigotiempoestablecido smallint, p_numerovuelta smallint, p_horacontrol time without time zone, p_horagps time without time zone, p_diferencia real, p_minutostolerancia smallint, p_codigoruta smallint, p_horainicio time without time zone, p_nombreruta character varying) TABLE (codigoregistrounidad smallint, nombres character varying, apellidopaternal character varying, apellidomaternal character varying,		
P53	pr_liultimodetalle tarjeta		p_codigotarjetai integer	Lista el último detalle tarjeta activo de la tarjeta
P54	pr_liunidades			Lista todas las unidades

		placa character varying, estado character)		
P55	pr_liunidadestrab ajando	TABLE (p_codigotarjeta integer, p_fecha date, p_codigoregistrunidad integer)	p_fecha character varying, p_codigoregistrunidad integer	Se listan las unidades que han trabajado en una cierta fecha
P56	pr_liunidadestrab ajandofecha	SETOF tarjeta	p_fecha character varying, p_codigoregistrunidad integer, p_opcion character varying	Se listan las unidades que han trabajado en una cierta fecha, semana, mes, semana para el reporte gráfico.
P57	pr_liunidadesesta doapellido	TABLE (codigoregistrunidad smallint, codigosocio smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, placa character varying, estado character, numeroacciones smallint)	p_estado character, p_apellidos character varying	Lista las unidades por estado y apellidos del socio

		TABLE (p_codigo smallint, p_codigopersonal smallint, p_usuario character varying, p_nombres character varying, p_apellidopaterno character varying, p_materno character varying, p_cargo character varying, p_vigencia boolean)			
P58	pr_liusuario		p_estadoi character varying		Lista los usuarios según el estado recibido
		TABLE (p_codigo smallint, p_usuario character varying, p_clave character varying, p_cargo character varying, p_indicador smallint, p_vigencia boolean)			
P59	pr_login		p_usuarioe character varying, p_clavee character varying		Verifica el logueo de usuario
		TABLE (codigopersona smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, correo character varying, dni character, direccion character varying, sexo character, estadocivil character			
P60	pr_lpersonal		p_codigopersona integer		Lista el personal según el codigo de persona

		varying, celular character, fechanacimiento date, fecharegistro date, fecharetiro date, cargo character varying, vigencia boolean) TABLE(codigosocio smallint, codigounidad smallint, fecharegistro date, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, placa character varying, soat character varying, marca character varying, modelo character varying, capacidad smallint, aniofabricacion smallint, color character varying, estado character) TABLE (p_codigo smallint,			
P61	Lregistrounidades		p_codigoregistro integer		Lista las unidades utilizando las tablas registro unidad , unidad , socio
P62	pr_lruta	p_direccion character, estado character, p_vigencia boolean)	p_codigoi integer		Lista una ruta por su codigo

P63	pr_lsanciondetalle tarjeta	TABLE (p_codigo smallint, p_codigoregistrounidad smallint, p_sanciontiempo real, p_estado character varying, p_tiemporetraso real)	p_codigodetalletarjetai integer	Se lista las sanciones pertencientes a un detalle
P64	pr_lsocio	TABLE (codigopersona smallint, nombres character varying, apellidopaterno character varying, apellidomaterno character varying, correo character varying, dni character, direccion character varying, sexo character, estadocivil character varying, celular character, fechanacimiento date, fecharegistro date, fecharetiro date, numeroacciones smallint, nombrespaciente character varying, telefonopaciente character, estado character, vigencia boolean)	p_codigopersona integer	Lista el socio según el código de la persona

		TABLE (p_codigo smallint, p_codigopersonal smallint, p_usuario smallint)		
P65	pr_usuario	character varying, p_contrasenia character varying, p_indicador smallint)	p_codigoe integer	Lista el usuario según el código
P66	pr_verificarcontra senia	boolean	p_codigo integer, p_contrasenia character varying	Verifica si la contraseña es correcta
P67	pr_verificartarjeta	TABLE (p_codigotarjeta integer)	p_codigoru integer, p_fecha character varying	Verifica si la unidad posea una tarjeta registrada activa según la fecha recibida

Fuente: Elaboración Propia.

3.9 Resultado de pruebas

La metodología XP, la cual es la empleada en el proyecto, hace referencia a la ejecución de pruebas a lo largo del desarrollo del proyecto, para finalmente conseguir la realización de lo planificado en cada iteración. En este proceso hubo colaboración tanto del programador como del usuario. A continuación se presenta un breve resumen de las pruebas realizadas durante el proyecto.

Pruebas unitarias

Estas pruebas fueron desarrolladas durante la ejecución y desarrollo de cada historia de usuario. Para la implementación se utilizó JUnit Ver figura 98 a figura 105

```
@Test
public void verificarPlaca() throws Exception {
    unidad.setPlaca("ABC-123");
    assertTrue(dao.verificarPlaca(unidad.getPlaca()));
    unidad.setPlaca("MPL-001");
    assertFalse(dao.verificarPlaca(unidad.getPlaca()));
}
```

Figura 98
Test verificar placa – UnidadTest

```
@Test
public void obtenerUnidad() throws Exception {
    unidad.setCodigo(1);
    registroUnidad = dao.getRegistroUnidad(unidad.getCodigo());
    assertNotNull(registroUnidad);
}
```

Figura 99
Test obtener unidad – UnidadTest

```

@Test
public void listarUnidadesActivas() throws Exception{
    List<RegistroUnidad> lista;
    lista = dao.listarUnidadesActivas();
    assertEquals(0, lista.size());
}

```

Figura 100
Test listar unidades activas – UnidadTest

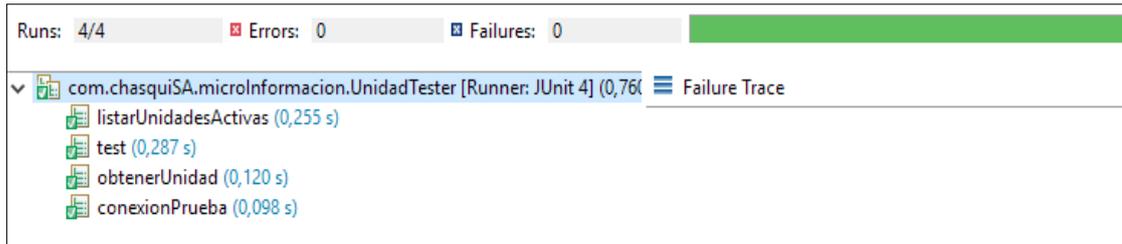


Figura 101
UnidadTest

```

@Test
public void verificarTarjeta() throws Exception{
    tarjeta.setCodigo(1);
    tarjeta.setCodigoRegistroUnidad(1);
    tarjeta.setFecha("2018-06-20");
    assertFalse(dao.verificarTarjeta(tarjeta));
}

```

Figura 102
Test verificar tarjeta– TarjetaTest

```

@Test
public void listarTarjetasEstado()throws Exception{

    tarjeta.setEstado("P");
    lista = dao.listarTarjetasEstado("2018-01-01","2018-06-20",tarjeta.getEstado());
    assertEquals(0,lista.size());
}

```

Figura 103
Test listar tarjetas por estado- TarjetaTest

```

@Test
public void listarTarjetas() throws Exception{
    tarjeta.setCodigo(1);
    tarjeta.setCodigoRegistroUnidad(1);
    lista = dao.listarTarjetas(tarjeta.getCodigoRegistroUnidad(),"2018-06-20");
    assertEquals(0,lista.size());
}

```

Figura 104
Test listar tarjetas– TarjetaTest

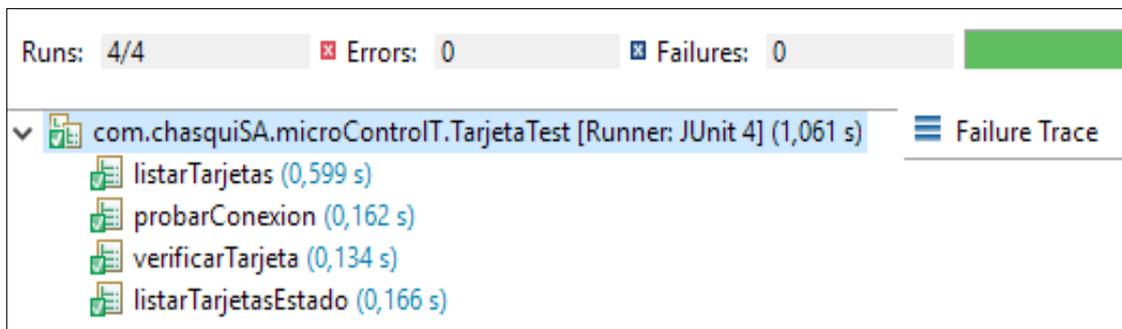


Figura 105
TarjetaTest

Pruebas de Aceptación

Las pruebas de aceptación han sido elaboradas a partir de los criterios de aceptación de cada historia de usuario, redactadas de acuerdo al formato del lenguaje Gherkin, donde cada prueba tiene distintos escenarios, estas fueron ejecutadas al final de cada iteración según las historias implementadas en el plan de entrega. Para la ejecución y automatización de estas pruebas se utilizó la herramienta Cucumber y Selenium web driver. A continuación, se muestra un resumen detallado de las pruebas de aceptación más relevantes por cada historia de usuario que fueron realizadas sobre el sistema integrando todos los módulos.

Registrar Socio

A continuación se detalla el escenario. Ver Tabla 103 – Implementación de prueba (Ver apéndice C).

Tabla 103

Escenario N° 2 – Edad de socio

ESCENARIO N° 2	EDAD DE SOCIO
	Dado que estoy en la página de autenticación Cuando inicio sesión como usuario administrador Y voy a la opción Registrar Socio
Descripción(Gherkin)	Y se ingrese una fecha de nacimiento cuya edad calculada sea menor a 18 años Y doy clic en el botón Registrar Socio Entonces deberá mostrarse un modal “Socio debe ser mayor de edad”

Resultado Esperado

The image shows a registration form for 'Shasqui S.A. | Empresa de transporte público'. The form includes fields for Sex (Masculino/Femenino), Date of Birth (13/09/2018), Civil Status (Soltero), Address (Calle Pedro Ruiz #165 - Chiclayo), Email (Perez@hotmail.com), and Cellphone (968857748). Below these fields is a section for 'Datos de referencia' with fields for Name and Surname (María Perez Sosa) and Cellphone (955263214). A modal dialog box titled 'Error al registrar socio' is displayed over the form, containing the message: 'El socio debe tener una edad mayor de 18 y menor que 100'. The modal has an 'Aceptar' button. The footer of the page reads 'Shasqui S.A. | Empresa de transporte público'.

Fuente: Elaboración Propia.

Registro de punto de control

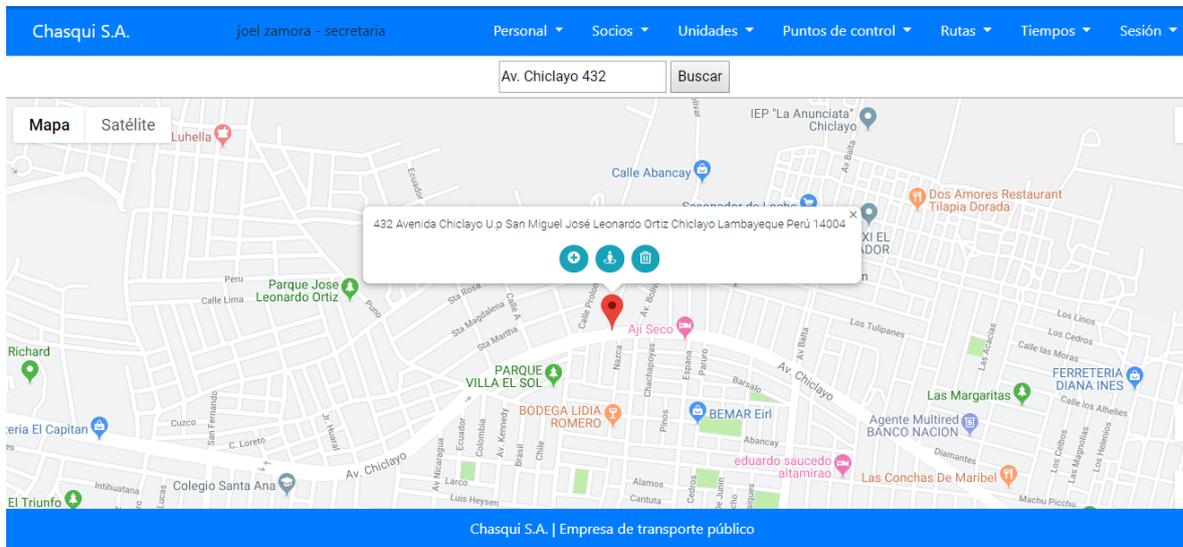
A continuación se detalla el escenario. Ver Tabla 104 – Implementación de prueba (Ver apéndice C).

Tabla 104

Escenario N° 2 – Registrar punto de control

DESCRIPCIÓN	REGISTRAR PUNTO DE CONTROL INGRESANDO DIRECCIÓN
	Dado que el usuario quiera registrar un punto de control
	Cuando ingrese la dirección en el campo Dirección y de clic en el botón
Descripción(Gherkin)	“Buscar”
)	Entonces la aplicación mostrará un icono de color rojo en el punto buscado dando las opciones de “Agregar punto de control”, “Visualizar punto” y “eliminar búsqueda”.

Resultado Esperado



Fuente: Elaboración Propia.

Dar de baja a la tarjeta

A continuación se detalla el escenario. Ver Tabla 105.

Tabla 105

Escenario N° 1 – Motivo en blanco

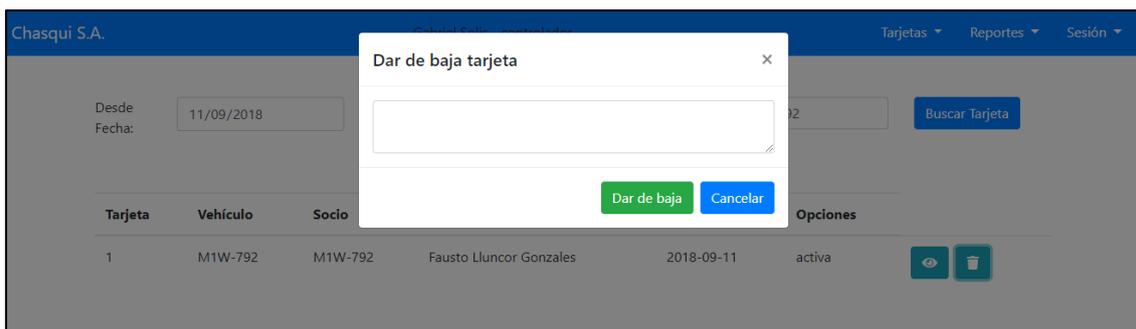
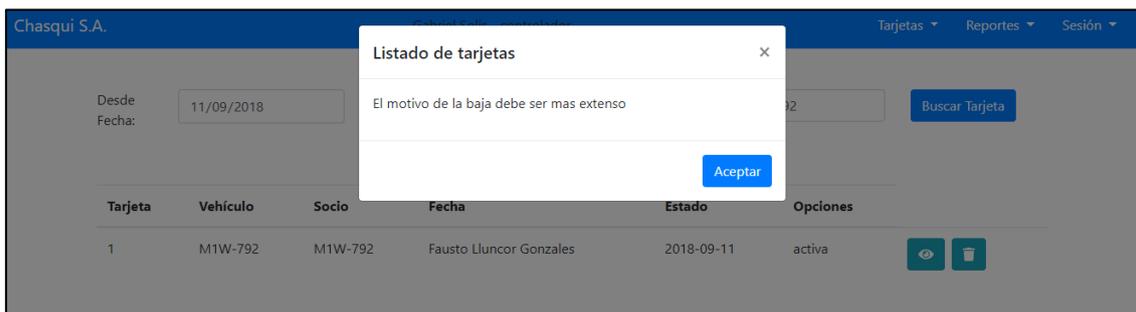
ESCENARIO N° 1	MOTIVO EN BLANCO
	Dado que estoy en la página de autenticación Cuando inicio sesión como usuario controlador Y voy a la opción Listado de tarjetas por unidad Y se ingrese la fecha de inicio y final junto con la placa de la unidad que se desea el reporte. Y se da clic en el botón Buscar Tarjeta
Descripción(Gherkin)	Y se da clic en el botón Eliminar de una tarjeta específica Entonces se mostrara un modal para ingresar el motivo del dar de baja a la tarjeta Y no se ingrese el motivo del porque eliminar la tarjeta Y se da clic en el botón “Dar de baja” Entonces se mostrara un modal indicando un mensaje “El motivo de la baja debe ser más extenso”

Resultado Esperado



The screenshot shows the Chasqui S.A. controlador interface. At the top, there is a navigation bar with the company name, the user name 'Gabriel Solís - controlador', and menu items for 'Tarjetas', 'Reportes', and 'Sesión'. Below the navigation bar, there is a search form with three input fields: 'Desde Fecha:' with the value '11/09/2018', 'Hasta Fecha:' with the value '14/09/2018', and 'Placa:' with the value 'M1W-792'. A blue button labeled 'Buscar Tarjeta' is positioned to the right of the search fields. Below the search form, there is a table with the following columns: 'Tarjeta', 'Vehículo', 'Socio', 'Fecha', 'Estado', and 'Opciones'. The table contains one row with the following data: '1', 'M1W-792', 'M1W-792', 'Fausto Lluncor Gonzales', '2018-09-11', and 'activa'. The 'Opciones' column for this row contains two icons: an eye icon and a trash can icon.

Tarjeta	Vehículo	Socio	Fecha	Estado	Opciones
1	M1W-792	M1W-792	Fausto Lluncor Gonzales	2018-09-11	activa



Fuente: Elaboración Propia.

Reporte de unidades trabajando por fecha

A continuación se detalla el escenario. Ver Tabla 106.

Tabla 106

Escenario N° 1 – Listar por tipo

ESCENARIO N° 1	LISTAR POR TIPO
	Dado que estoy en la página de autenticación
	Cuando inicio sesión como usuario administrados
	Y voy a la opción Reporte de unidades trabajando por fecha
Descripción(Gherkin)	Y se ingrese la fecha junto con el tipo de unidades “Unidades laborando”.
	Y se da clic en el botón Listar reporte
	Entonces se mostrará las unidades que han laborado el día especificado.
Resultado Esperado	

Chasqui S.A. Tarjetas ▾ Reportes ▾ Sesión ▾

Fecha: Placa: [Listar reporte](#)

Filtro de Tarjetas Unidades trabajando: Unidades sin trabajar:

Socio	Vehículo
Fausto Lluncor Gonzales	M1W-792

Fuente: Elaboración Propia.

Capítulo 4 Conclusiones

En este apartado se presentan las conclusiones que se obtuvieron a partir del presente proyecto de Tesis.

- Se realizó entrevistas iniciales con el cliente consideradas en forma técnica como Sprint 0 obteniendo documentos e informes relacionados con el proceso en estudio. Logrando definir e identificar la lista de requerimientos necesarios en los procesos de asignación de rutas y tarjetas que fueron desarrollándose en cada Sprint cumpliendo así las expectativas del cliente.
- Se realizó satisfactoriamente el análisis de los procesos de asignación de tarjetas, asignación de rutas, asignación de sanciones, registrar socios y asignación de tarjetas permitiendo identificar y conocer de manera más precisa cada uno de los procesos de negocio.
- Se implementó un método en el cual se trabajó con los usuarios para definir conceptos los cuales serían utilizados en todo el transcurso de cada sprint para un mayor entendimiento por parte de los usuarios y el equipo de trabajo.
- Se implementó el sistema informático web de control de recorridos y rutas permitiendo regularizar las rutas y tiempos de las unidades de transporte público en la empresa, la obtención de información de una forma más rápida y segura y la disminución de tiempos en el proceso de información. La incorporación de buenas prácticas de SCRUM y el seguimiento de la metodología XP complementada con una arquitectura basada en microservicios permitieron cumplir con los requerimientos técnicos propuestos y tiempos de entrega en cada uno de los Sprint.

- Se logró diseñar la base de datos, interfaces de usuario y componentes internos, durante el desarrollo de cada Sprint, a fin de obtener un sistema informático que cumpla con las características que se necesiten. Para ello se utilizó el modelador de Erwin Data Modeler que sirvió de mucha ayuda para el modelado de base de datos; para diseñar los prototipos de cada interfaz se usó Justinmind Prototyper, el cual permitió diseñar prototipos de manera rápida y personalizada a fin de obtener un prototipo final de acuerdo a lo que pidió el cliente. Para el diseño de tablas y procedimiento se usó el gestor pgAdmin el cual fue de gran utilidad en el manejo de Scripts.
- Se realizó la implementación de los módulos del sistema informático en el transcurso de cada sprint, para ello se utilizaron diversos módulos del framework Spring para java como Spring Rest, Spring security y el módulo orientado a microservicios Spring Cloud Netflix, los cuales nos facilitó la labor de desarrollo gracias a la disposición de las diversas funcionalidades y configuraciones que ofrece. Para las pruebas del sistema se procedió a desplegar la aplicación en un servidor embebido propio de Spring Boot, con lo que se pudo identificar falencias en el sistema para la posterior solución y retroalimentación de los miembros del proyecto para cada módulo del sistema.
- Se efectuaron pruebas unitarias con Junit y pruebas de aceptación con el uso de la herramienta Cucumber, pruebas necesarias que determinaron que el sistema cumplió con los requerimientos especificados; éstas herramientas permitieron mejorar la eficiencia de nuestro código, gracias a la ejecución de descripciones funcionales en texto plano como test automáticos de aceptación, ya que el lenguaje utilizado es un lenguaje común y sencillo, que es entendible por cualquiera de las partes que conforman el proyecto.

- Se logró poner en marcha el sistema informático web de control de recorridos y rutas a partir de la culminación de cada uno de los resultados esperados, detectando y corrigiendo inconvenientes presentados en entregables ya instalados, logrando así soporte y mejoras continuas a favor del usuario final.

Capítulo 5 Recomendaciones

- Se recomienda el uso librerías y mapas para conocer las rutas alternas y desvíos de rutas que puedan generarse día a día debido a accidentes, obras u otros casos.
- Se recomienda el uso de la tecnología docker para el almacenamiento de microservicios y la interconexión entre ellos.
- Se recomienda implementar una funcionalidad que permita a los usuarios de transporte público conocer la hora de llegada a los diversos paraderos.
- Se recomienda implementar la intreconexión con el Servicio de administración tributaria para el conocimiento de las multas pertenecientes a los trabajadores de las unidades de la empresa.
- Se recomienda incentivar el uso de pruebas automatizadas las cuales permitan disminuir el tiempo en la realización de pruebas y mejorar la calidad del software del producto con la detección oportuna de errores en él.

Capítulo 6 Bibliografía

- Alvarez, M. (18 de Julio de 2001). *Qué es Java*. Recuperado el 06 de Octubre de 2017, de Desarrolloweb: <https://desarrolloweb.com/articulos/497.php>
- Apache. (07 de Octubre de 2017). *Maven*. Recuperado el 08 de Octubre de 2017, de Apache Maven Project: <http://maven.apache.org/what-is-maven.html>
- Avram, A., & Marinescu, F. (08 de Diciembre de 2006). *Domain-Driven Desing Quickly*. Recuperado el 02 de Octubre de 2017, de InfoQ: <https://www.infoq.com/minibooks/domain-driven-design-quickly>
- BBC Mundo. (10 de Febrero de 2017). *¿Cuáles son las ciudades con mejor y peor transporte público en América Latina?* Recuperado el 02 de Octubre de 2017, de BBC MUNDO: <http://www.bbc.com/mundo/noticias-america-latina-38927134>
- Bocanegra Ureta, R. G. (2012). *Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de telecomunicaciones (Tesis de pregrado) Universidad Ricardo Palma*. Lima. Recuperado el 02 de Octubre de 2017, de <http://cybertesis.urp.edu.pe/handle/urp/34>
- Borja López, Y. (S.F.). *Metodología Ágil de Desarrollo de Software – XP*. Recuperado el 06 de Octubre de 2017, de Proyecto EtnoMatemática - Runayupay Ecuador: http://www.runayupay.org/publicaciones/2244_555_COD_18_290814203015.pdf
- Borjas Giraldo, G. (2013). *Análisis, diseño e implementación de un sistema de información para la administración de horarios y rutas en empresas de transporte público (Tesis de pregrado) Universidad Católica del Perú*. Lima. Recuperado el 05 de Octubre de 2017, de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/4998>

- Boundless. (S.F). *Introducción a PostGIS*. Recuperado el 05 de Octubre de 2017, de Boundless: <http://workshops.boundlessgeo.com/postgis-intro/index.html>
- Chirinos, P. (20 de Julio de 2015). *Chiclayo: 85 mil vehículos hacen rebasar el transporte*. Recuperado el 13 de Octubre de 2017, de Diario Correo: <https://diariocorreo.pe/edicion/lambayeque/85-mil-vehiculos-hacen-rebasar-el-transporte-603657/>
- Denzer, P. (23 de Octubre de 2002). *PostgreSQL*. Recuperado el 05 de Octubre de 2017, de Departamento de Electrónica, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA: <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>
- Diario Oficial El Peruano. (22 de Abril de 2017). *PARÁMETROS DE EVALUACIÓN SOCIAL*. Recuperado el 06 de Octubre de 2018, de Diario Oficial El Peruano: <http://spij.minjus.gob.pe/Graficos/Peru/2017/Abril/22/RD-002-2017-EF-63.01-2.pdf>
- Echeverría, A., López , G., Grossi, M., Servetto, A., Jeder , I., Paredes, A., & Linares, P. (2010). *DDD (diseño dirigido por el dominio) y aplicaciones Enterprise: ¿fidelidad al*. Recuperado el 12 de Octubre de 2017, de SEDICI: http://sedici.unlp.edu.ar/bitstream/handle/10915/18328/Documento_completo.pdf?sequence=1&isAllowed=y
- EcuRed. (S.F). *PostGreSQL*. Recuperado el 05 de Octubre de 2017, de EcuRed Conocimientos con todos y para todos: <https://www.ecured.cu/PostGreSQL>
- Esau, A. (05 de Mayo de 2014). *Docker, Qué es y sus principales características*. Recuperado el 23 de Octubre de 2017, de OpenWebinars: <https://openwebinars.net/blog/docker-que-es-sus-principales-caracteristicas/>

- Figueredo Pérez , A. (03 de Mayo de 2011). *PostGreSQL*. Recuperado el 05 de Octubre de 2017, de EcuRed: <https://www.ecured.cu/PostGreSQL>
- Fontela, A. (2015). *¿Que es Bootstrap?* Recuperado el 06 de Octubre de 2017, de Raiola Networks: <https://raiolanetworks.es/blog/que-es-bootstrap/>
- García Oterino, A. M. (06 de Junio de 2014). *Simple y rápido. Entiende qué es Maven en menos de 10 min.* Recuperado el 08 de Octubre de 2017, de javiergarzas: <http://www.javiergarzas.com/2014/06/maven-en-10-min.html>
- Guanilo Barreto, R. (2013). *Implementación de un sistema web y aplicación móvil bajo plataforma android utilizando tecnología GPS para el control y monitoreo de las unidades de la empresa Transporta S.A.C. de la ciudad de Chiclayo (Proyecto de Pregrado) Universidad Señor de Sipán.* Recuperado el 06 de Octubre de 2017, de <http://repositorio.uss.edu.pe/handle/uss/404>
- Huerta, E., Mangiaterra, A., & Noguera, G. (2005). *GPS Posicionamiento Satelital.* Recuperado el 09 de Octubre de 2017, de <http://www.fceia.unr.edu.ar/gps/librogps/>
- Jorge. (17 de Septiembre de 2009). *Qué es Maven.* Recuperado el 08 de Octubre de 2017, de Tratando de entenderlo: <http://tratandodeentenderlo.blogspot.pe/2009/09/que-es-maven.html>
- Lasa Gómez, C., Álvarez Garcia, A., & De las Heras del Dedo, R. (2017). *Métodos Ágiles Scrum, Kanban, Lean.* (J. I. Tena., Ed.) Madrid: EDICIONES ANAYA MULTIMEDIA(GRUPO ANAYA, S.A.).
- Maven.* (07 de Octubre de 2017). Recuperado el 08 de Octubre de 2017, de Apache Maven Projct: <http://maven.apache.org/what-is-maven.html>
- Mendoza Orozco, X. A. (2013). *Mejoramiento del Servicio de Transporte Urbano Colectivo en la Pista Juan Pablo II, Managua-Nicaragua (Proyecto de Tesina)*

- Universidad Carlos III de Madrid*. Managua. Recuperado el 03 de Octubre de 2017, de <https://e-archivo.uc3m.es/handle/10016/18142>
- nvindi. (05 de Enero de 2018). *TIR y VAN: Cálculo y concepto*. Recuperado el 06 de Octubre de 2018, de Todo Productos Financieros:
<http://todoproductosfinancieros.com/tir-calculo-y-concepto/>
- Paredes, A. M. (Febrero de 22 de 2009). *DDD - Parte 1: Poniendo el Modelo de Dominio a Trabajar*. Recuperado el 12 de Octubre de 2017, de El blog de Adrián M. Paredes: <http://elblogdelfrasco.blogspot.pe/2009/02/ddd-parte-1-poniendo-el-modelo-de.html>
- Paredes, A. M. (15 de Marzo de 2009). *DDD - Parte 2: Los Componentes Fundamentales de un Modelo Guiado por el Dominio*. Recuperado el 12 de Octubre de 2017, de El blog de Adrián M. Paredes: <http://elblogdelfrasco.blogspot.pe/2009/03/ddd-parte-2-los-componentes.html>
- Paredes, A. M. (29 de Diciembre de 2009). *DDD - Parte 3: Refactorizando Hacia una Visión más Profunda*. Recuperado el 12 de Octubre de 2017, de El blog de Adrián M. Paredes: <http://elblogdelfrasco.blogspot.pe/2009/12/ddd-parte-3-refactorizando-hacia-una.html>
- Pérez García, A. (16 de Septiembre de 2006). *Maven, nunca antes resultó tan fácil compilar, empaquetar*. Recuperado el 08 de Octubre de 2017, de Adictos Al Trabajo: <https://www.adictosaltrabajo.com/tutoriales/maven/#mozTocId933973>
- Pérez-Herrera Cuadrillero, M. (2015). *Arquitectura basadas en microservicios (Tesis de grado) Universidad politécnica de Madrid*. Madrid. Recuperado el 14 de Octubre de 2017, de <http://oa.upm.es/37346/>

- Perú21. (22 de Setiembre de 2017). *Vehículos desbordan la demanda en Lima*. Recuperado el 13 de Octubre de 2017, de Perú 21: <https://peru21.pe/lima/vehiculos-desbordan-demanda-lima-376994>
- PostgreSQL. (2012). *Características, limitaciones y ventajas*. Recuperado el 05 de Octubre de 2017, de PostgreSQL: <http://postgresql-dbms.blogspot.pe/p/limitaciones-puntos-de-recuperacion.html>
- Prucommercialre. (2016). *¿Qué es el flujo de caja incremental?* Recuperado el 05 de Octubre de 2018, de Prucommercialre: <http://www.prucommercialre.com/que-es-el-flujo-de-caja-incremental/>
- Radigan, D. (S.F.). *Scrum Breve análisis sobre el uso del marco de trabajo scrum en el desarrollo de software*. Recuperado el 07 de Octubre de 2017, de ATlassian: <https://es.atlassian.com/agile/scrum>
- Raico, R., & Linares, E. (20 de Febrero de 2012). *MERCADO VIRTUAL AL SERVICIO DE LAS MYPES EN LA REGIÓN LAMBAYEQUE (Tesis de Pregrado) Universidad Nacional Pedro Ruiz Gallo*. Lambayeque. Recuperado el 02 de Octubre de 2017
- Rodríguez López, J. A. (10 de Noviembre de 2015). *Eureka: Registro de microservicios*. Recuperado el 20 de Octubre de 2017, de Paradigma Digital: <https://www.paradigmadigital.com/dev/el-microservicio-eureka-a-fondo-12/>
- Rodríguez López, J. A. (28 de Octubre de 2015). *Microservicios Spring Cloud: Arquitectura*. Recuperado el 17 de Octubre de 2017, de Paradigma Digital: <https://www.paradigmadigital.com/dev/quien-es-quien-en-la-arquitectura-de-microservicios-spring-cloud-22/>
- Rodríguez López, J. A. (26 de Octubre de 2015). *Microservicios Spring Cloud: Arquitectura*. Recuperado el 17 de Octubre de 2017, de Paradigma Digital:

<https://www.paradigmadigital.com/dev/quien-es-quien-en-la-arquitectura-de-microservicios-spring-cloud-12/>

Rodríguez López, J. A. (02 de Diciembre de 2015). *Ribbon: el balanceador de carga de spring-cloud-netflix*. Recuperado el 19 de Octubre de 2017, de Paradigma Digital: [https://www.paradigmadigital.com/dev/ribbon-balanceador-carga-spring-cloud-netflix-](https://www.paradigmadigital.com/dev/ribbon-balanceador-carga-spring-cloud-netflix-13/?utm_source=Site&utm_campaign=Related_Posts_Right&utm_medium=Blog)

[13/?utm_source=Site&utm_campaign=Related_Posts_Right&utm_medium=Blog](https://www.paradigmadigital.com/dev/ribbon-balanceador-carga-spring-cloud-netflix-13/?utm_source=Site&utm_campaign=Related_Posts_Right&utm_medium=Blog)

Rodríguez, T. (16 de Junio de 2012). *Bootstrap*. Recuperado el 06 de Octubre de 2017, de GENBETA: dev: <https://www.genbetadev.com/frameworks/bootstrap>

Rubio, E., & Carrillo, A. (11 de Agosto de 2016). *El lenguaje de Programación Java™*. Recuperado el 06 de Octubre de 2017, de Geofisica UNAM: <http://mmc.geofisica.unam.mx/femp/Herramientas/Java/JavaBasico/El%20lenguaje%20de%20programacion%20java.pdf>

Santa Cruz, E. (24 de Enero de 2017). *El PRI: uno de los indicadores que más llama la atención de los inversionistas*. Recuperado el 06 de Octubre de 2018, de Conexionesan: <https://www.esan.edu.pe/apuntes-empresariales/2017/01/el-pri-uno-de-los-indicadores-que-mas-llama-la-atencion-de-los-inversionistas/>

Santa Cruz, E. (24 de Enero de 2017). *Fundamentos financieros: el valor actual neto (VAN)*. Recuperado el 06 de Octubre de 2018, de Conexionesan: <https://www.esan.edu.pe/apuntes-empresariales/2017/01/fundamentos-financieros-el-valor-actual-neto-van/>

Shadunsky, A. (01 de Febrero de 2018). *Diferencia entre flujo de efectivo incremental y flujo total de efectivo*. Recuperado el 05 de Octubre de 2018, de Cuida tu dinero: <https://www.cuidatudinero.com/13104206/diferencia-entre-flujo-de-efectivo-incremental-y-flujo-total-de-efectivo>

- Softeng. (S.F). *Proceso y Roles de Scrum*. Recuperado el 07 de Octubre de 2017, de
SOFTENG YOUR COMPETITIVE ADVANTAGE: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>
- Tyagi, S. (Agosto de 2006). *Servicios Web RESTful*. Recuperado el 09 de Octubre de 2017, de Oracle: <http://www.oracle.com/technetwork/articles/javase/index-137171.html>
- Vaqqas, M. (23 de Septiembre de 2014). *Servicios web RESTful: un tutorial*. Recuperado el 09 de Octubre de 2017, de Dr.Dobb's: <http://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069>
- Vega Bueno, M. A. (2014). *Gestión de flota para una empresa distribuidora de pizza (Tesis de Maestría) Universidad de Chile*. Santiago de Chile. Recuperado el 04 de Octubre de 2017, de <http://repositorio.uchile.cl/handle/2250/116690>
- Wajser, D. (07 de Septiembre de 2017). *Rest - Restful: Ventajas y diferencias*. Recuperado el 09 de Octubre de 2017, de LatAm Digital: <http://latamdigital.softtek.co/rest-restful-ventajas-y-diferencias>

Capítulo 7 Apendices

7.1 Apéndice A: Desarrollo de Historias de usuario

- ✓ HU 001 – Registrar Socio
 - ✓ Diseñar la interfaz

The image shows a web form titled "Registro Socios". It contains several input fields and a button. The fields are organized into sections: "Nombres:" with a text input for "Ingrese nombres del socio"; "Apellido Paterno:" with a text input for "Ingrese apellido paterno del socio"; "Apellido Materno:" with a text input for "Ingrese apellido materno del socio"; "DNI:" with a text input for "Ingrese DNI del socio"; "Fecha de Nacimiento:" with a date input for "dd/mm/aaaa"; "Sexo:" with radio buttons for "Masculino" (selected) and "Femenino"; "Fecha de Registro:" with a date input for "dd/mm/aaaa"; "Estado Civil:" with a dropdown menu showing "Casado"; "Dirección:" with a text input for "Ingrese dirección del socio"; "Correo:" with a text input for "Ingrese correo del socio"; "Celular:" with a text input for "Ingrese celular del socio"; "Datos Referencia" section with "Nombres y apellidos:" (text input for "Ingrese nombres y apellidos completos del pariente") and "Celular:" (text input for "Ingrese celular del pariente"). At the bottom is a blue button labeled "Registrar Socio".

Figura 106 Diseñar la interfaz – HU 001 – Registrar Socio –Apendice A

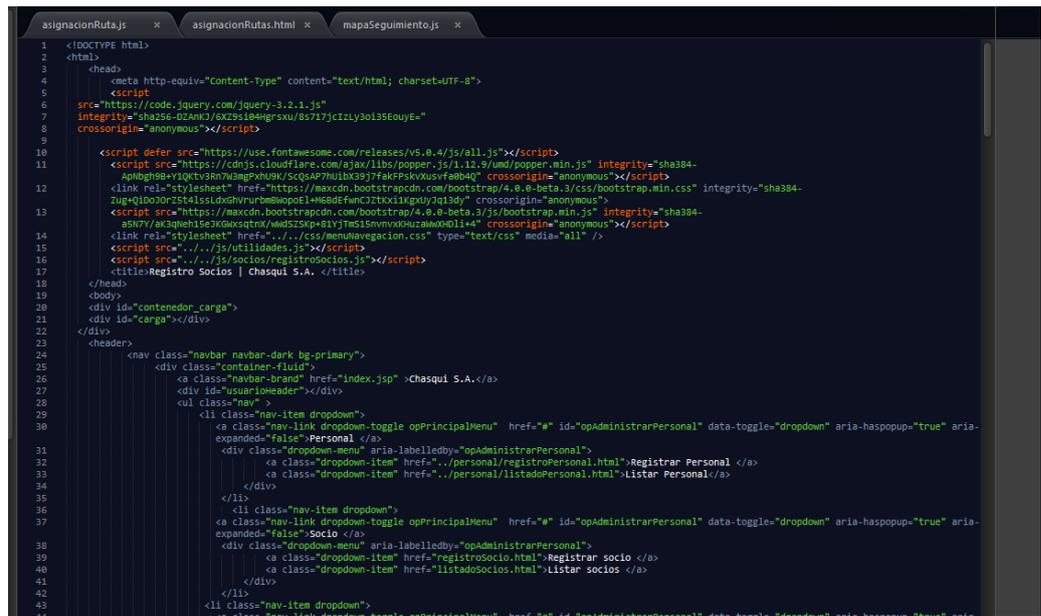
✓ Diseñar y crear tablas de la base de datos

```
1  ----- sprint 01 -----
2
3  --- TABLA Persona -----
4  CREATE TABLE persona /*Ya*/
5  (
6
7      codigo smallserial NOT NULL ,
8      nombres character varying(32) COLLATE pg_catalog."default" NOT NULL,
9      apellidoPaterno character varying(32) COLLATE pg_catalog."default" NOT NULL,
10     apellidoMaterno character varying(32) COLLATE pg_catalog."default" NOT NULL,
11     correo character varying(76) COLLATE pg_catalog."default",
12     direccion character varying(76) COLLATE pg_catalog."default" NOT NULL,
13     sexo character(1) NOT NULL,
14     dni character(8) unique COLLATE pg_catalog."default" NOT NULL,
15     fechaNacimiento date NOT NULL,
16     estadoCivil character varying(10),
17     celular character(9) COLLATE pg_catalog."default",
18
19     CONSTRAINT persona_pkey PRIMARY KEY (codigo)
20 )
21 --- TABLA SOCIO ---/*Ya*/
22 CREATE TABLE socio(CodigoPersona smallint primary key references persona(codigo),
23     numeroAcciones smallint NOT NULL,
24     fechaRegistro Date NOT NULL,
25     fechaRetiro Date NOT NULL,
26     nombresApellidosPariente character varying(120),
27     telefonoPariente character(9),
28     estado character(1),
29     vigencia boolean not null);
30
```

Figura 107 Diseñar y crear tabla de BD– HU 001 – Registrar Socio –

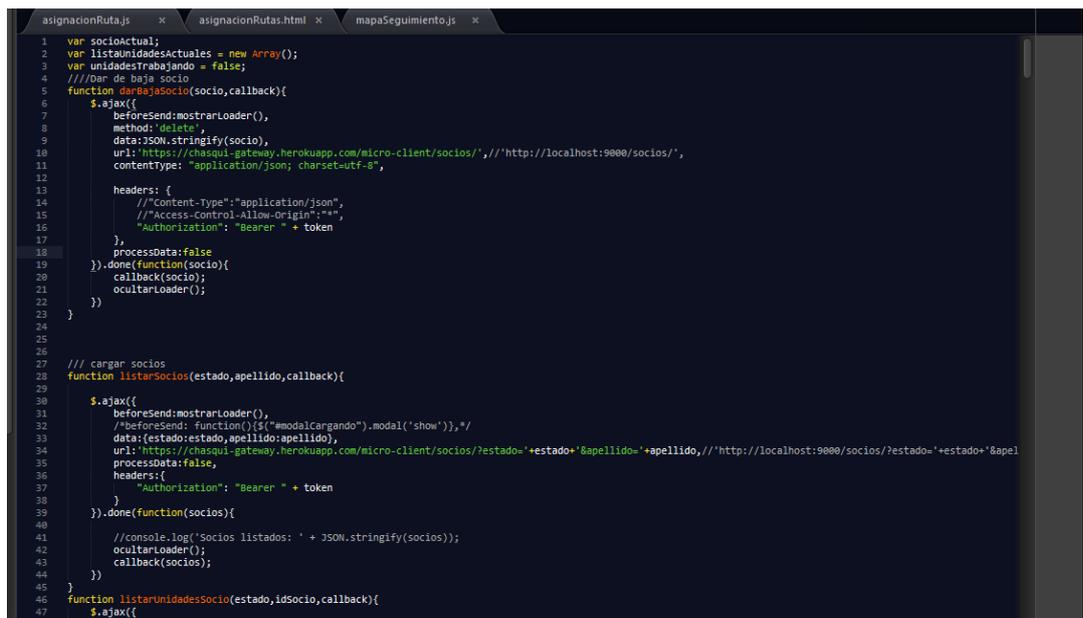
Apndice A

✓ Construir interfaz



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <script
6 src="https://code.jquery.com/jquery-3.2.1.js"
7 integrity="sha256-D2ANKK16XZ9S104grsxu/85717c121y30135Eouye="
8 crossorigin="anonymous"></script>
9
10 <script defer src="https://use.fontawesome.com/releases/v5.0.4/js/all.js"></script>
11 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
12 ApmBgh99+1Qkv3RN73mgPNUK/SCQAP7HuibX3977fakFPskvXusvfa8b4Q" crossorigin="anonymous"></script>
13 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min.css" integrity="sha384-
14 TuzqQq368W0xod6V0n2dbb7q13eDihvrn1wbM90p24dvr60g74f761422kkx11kguy3a33y" crossorigin="anonymous">
15 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/js/bootstrap.min.js" integrity="sha384-
16 a5N77/aK3qNeihSe3kG6xqtxN6dS2Skp+81y3Tms15nvnvXKhuZAW041i4" crossorigin="anonymous"></script>
17 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min.css" type="text/css" media="all" />
18 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/js/bootstrap.min.js" type="text/javascript" media="all" />
19 </head>
20 <div id="contenedor_carga">
21 <div id="carga"></div>
22 </div>
23 <header>
24 <nav class="navbar navbar-dark bg-primary">
25 <div class="container-fluid">
26 <a class="navbar-brand" href="index.jsp" >Chasqui S.A.</a>
27 <div id="navbarSupportedContent"></div>
28 <ul class="nav">
29 <li class="nav-item dropdown">
30 <a class="nav-link dropdown-toggle opPrincipalMenu" href="#" id="opAdministrarPersonal" data-toggle="dropdown" aria-haspopup="true" aria-
31 expanded="false">Personal </a>
32 <div class="dropdown-menu" aria-labelledby="opAdministrarPersonal">
33 <a class="dropdown-item" href="personal/registroPersonal.html">Registrar Personal </a>
34 <a class="dropdown-item" href="personal/listadoPersonal.html">Listar Personal </a>
35 </div>
36 </li>
37 <li class="nav-item dropdown">
38 <a class="nav-link dropdown-toggle opPrincipalMenu" href="#" id="opAdministrarPersonal" data-toggle="dropdown" aria-haspopup="true" aria-
39 expanded="false">Socio </a>
40 <div class="dropdown-menu" aria-labelledby="opAdministrarPersonal">
41 <a class="dropdown-item" href="registroSocio.html">Registrar socio </a>
42 <a class="dropdown-item" href="listadoSocios.html">Listar socios </a>
43 </div>
44 </li>
45 </ul>
46 </div>
47 </nav>
48 </header>
```

Figura 108 Construir Interfaz 1– HU 001– Registrar Socio – Apendice A



```
1 var socioActual;
2 var listaUnidadesActuales = new Array();
3 var unidadesTrabajando = false;
4 //Dar de baja socio
5 function darBajaSocio(socio, callback){
6 $.ajax({
7 beforeSend: mostrarLoader(),
8 method: 'delete',
9 data: JSON.stringify(socio),
10 url: 'https://chasqui-gateway.herokuapp.com/micro-client/socios/', // 'http://localhost:9000/socios/',
11 contentType: 'application/json; charset=utf-8',
12
13 headers: {
14 'Content-Type': 'application/json',
15 'Access-Control-Allow-Origin': '*',
16 'Authorization': 'Bearer ' + token
17 },
18 processData: false
19 }).done(function(socio){
20 callback(socio);
21 ocultarLoader();
22 })
23 }
24
25
26
27 // cargar socios
28 function listarSocios(estado, apellido, callback){
29 $.ajax({
30 beforeSend: mostrarLoader(),
31 //beforeSend: function(){ $('#modalCargando').modal('show') }, //
32 data: {estado: estado, apellido: apellido},
33 url: 'https://chasqui-gateway.herokuapp.com/micro-client/socios/?estado=' + estado + '&apellido=' + apellido, // 'http://localhost:9000/socios/?estado=' + estado + '&apel
34 processData: false,
35 headers: {
36 'Authorization': 'Bearer ' + token
37 },
38 }).done(function(socios){
39 //console.log('Socios listados: ' + JSON.stringify(socios));
40 ocultarLoader();
41 callback(socios);
42 })
43 }
44
45 // cargar unidades de un socio
46 function listarUnidadesSocio(estado, idSocio, callback){
47 $.ajax({
```

Figura 109 Construir Interfaz 2– HU 001 – Registrar Socio – Apendice A

✓ Crear procedimientos almacenados en la base de datos.

```
chaskiSA on postgres@PostgreSQL 10
1
2
3
4 ----- SOCIO -----
5 CREATE OR REPLACE FUNCTION pr_1Socio (p_Nombres character varying(32),p_ApellidoPaterno character varying(32),p_ApellidoMaterno character varying
6 p_Correo character varying(76),p_Direccion character varying(76),p_DNI character(9),p_Sexo character(1),p_F
7 p_Celular character(9),p_EstadoCivil character varying,p_FechaRegistro character varying,
8 p_NombresPariente character varying(128),p_telefonoPariente character(9),p_Estado character(1)) returns voi
9 AS $$
10
11 /*
12  * Procedimiento: pr_1Socio
13  * Autor: Gabriel Solis
14  * Fecha creación: 2018-07-12
15  * Versión: 1.0
16  * Parámetros: Nombres,apellido paterno, apellido materno, correo, direccion, dni, sexo , estado civil
17  * fecha de nacimiento , fecha de registro , datos de pariente , estado
18  * Descripción: Procedimiento que registrará los datos en la tabla persona y socio
19  */
20
21 DECLARE varCodigoPersona smallint;
22 BEGIN
23 INSERT INTO persona(nombres,apellidopaterno,apellidoMATerno,correo,direccion,dni,sexo,estadocivil,fechana
24 VALUES(p_Nombres,p_ApellidoPaterno,p_ApellidoMaterno,p_Correo,p_Direccion,p_DNI,p_Sexo,p_EstadoCivil,to_d
25
26 varCodigoPersona := (SELECT MAX(codigo) from persona);
27
28 INSERT INTO socio(codigopersona,numeroacciones,fecharegistro,fecharetiro,nombresApellidosPariente,telefon
29 values (varCodigoPersona,0,to_date(p_FechaRegistro,'yyyy-MM-DD'),to_date(p_FechaRegistro,'yyyy-MM-DD'),p_
30 p_telefonoPariente,p_estado,true);
31
32 END;
33 $$ LANGUAGE plpgsql
34
35 ----- Ejemplo de retorno de tablas unidas ----
```

Figura 110 Crear PA en la Base de Datos– HU 001– Registrar Socio – Apendice A

✓ Crear el código correspondiente.

```
workspace-sts-3.9.2.RELEASE - Microservicioinformacion/src/main/java/com/chaskiSA/microInformacion/Dominio/Socio.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Microservicioinformacion/pom.xml bootstrap.yml application.yml AuthServer/pom.xml EurekaPrueba/pom.xml Socio.java
5
6 public class Socio extends Persona {
7
8     int numeroAcciones;
9     String fechaRegistro;
10    String fechaRetiro;
11    String nombresApellidosPariente;
12    String estado;
13    String telefonoPariente;
14    boolean vigencia;
15
16    public Socio() {
17    }
18
19
20    public void setEstado(String estado) {
21        this.estado = estado;
22    }
23
24    public String getEstado() {
25        return estado;
26    }
27
28    public String getTelefonoPariente() {
29        return this.telefonoPariente;
30    }
31
32    public void setTelefonoPariente(String telefono) {
33        this.telefonoPariente = telefono;
34    }
35
36
37    public Socio(int numeroAcciones, String fechaRegistro, String fechaRetiro, String nombresApellidosPariente,
38        String estado, String telefonoPariente, boolean vigencia) {
39        super();
40        this.numeroAcciones = numeroAcciones;
41        this.fechaRegistro = fechaRegistro;
42        this.fechaRetiro = fechaRetiro;
43        this.nombresApellidosPariente = nombresApellidosPariente;
44        this.estado = estado;
45        this.telefonoPariente = telefonoPariente;
46        this.vigencia = vigencia;
47    }
48
49 }
```

Figura 111 Crear Código Parte 1 -HU 001 -Registrar Socio - Apendice A

```
workspace-sts-3.9.2.RELEASE - Microservicioinformacion/src/main/java/com/chaskiSA/microInformacion/restController/SocioController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Microservicioinformacion/pom.xml bootstrap.yml application.yml AuthServer/pom.xml EurekaPrueba/pom.xml Socio.java SocioController.java
1 package com.chaskiSA.microInformacion.restController;
2
3 import java.io.File;
4
5
6
7
8 @RestController
9 @RequestMapping("/socios")
10 public class SocioController {
11
12     @RequestMapping(value = "/", method = RequestMethod.POST)
13     // @ResponseStatus(HttpStatus.CREATED)
14     public ResponseEntity<?> nuevoSocio(@RequestBody Socio socio){
15         SocioDAO dao = new SocioDAO();
16         try {
17             if(!dao.verificarDNI(socio.getDni())) {
18                 //socio.asignarFechaRegistro();
19                 socio.asignarEstado();
20                 dao.registrarSocio(socio);
21                 return new ResponseEntity<>("Creado",HttpStatus.OK);
22             }
23             return new ResponseEntity<>("Dni ya registrado",HttpStatus.CONFLICT);
24         } catch (Exception e) {
25             System.out.println(e.getMessage());
26             return new ResponseEntity<>(e.getMessage(),HttpStatus.BAD_REQUEST);
27         }
28     }
29 }
```

Figura 112 Crear Código Parte 2 -HU 001 -Registrar Socio - Apendice A

✓ H043: Registro de GPS

Para el registro de los dispositivos GPS hemos utilizado una herramienta llamada Traccar la cual brinda una interfaz, un API REST y una estructura de base de datos para el manejo y administración de los dispositivos GPS.

Para el enlace entre el dispositivo GPS y la unidad se diseñó otra interfaz.

✓ Diseñar la interfaz.



Figura 113 Diseñar la interfaz -HU 043 –Registro de GPS - Apendice A

✓ Construir la interfaz.

```
asignacionRutas.js x asignacionRutas.html x mapaSeguimiento.js x
66 </div>
67 </nav>
68 </header>
69 <div class="container">
70 <h2 Registro de GPS</h2>
71 <div class="row mt-5 justify-content-center">
72 <div class="col-md">
73 <select class="form-control" id="cmbGPS">
74 <option value="0">Seleccionar GPS </option>
75 </select>
76 </div>
77 <div class="col-md">
78 <select class="form-control" id="cmbUnidades">
79 <option value="0">Seleccionar Unidades </option>
80 </select>
81 </div>
82 </div>
83 <div>
84 <button class="btn btn-primary" id="btn-registrar">Registrar GPS</button>
85 </div>
86 </div>
87 <div class="modal fade" id="modalRespuesta">
88 <div class="modal-dialog" role="document">
89 <div class="modal-content">
90 <div class="modal-header">
91 <h5 class="modal-title" id="tituloRespuesta">Registro GPS</h5>
92 <button type="button" class="close" data-dismiss="modal"
93 aria-label="close">
94 <span aria-hidden="true">&times;</span>
95 </button>
96 </div>
97 <div class="modal-body">
98 <p id="contenidoRespuesta"></p>
99 </div>
100 <div class="modal-footer">
101 <button type="button" class="btn btn-primary"
102 id="btn-aceptar">Aceptar</button>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </body>
111 </html>
```

Figura 114 Cuonstruir la interfaz -HU 043–Registro de GPS-Apendice A

Interfaz Traccar.

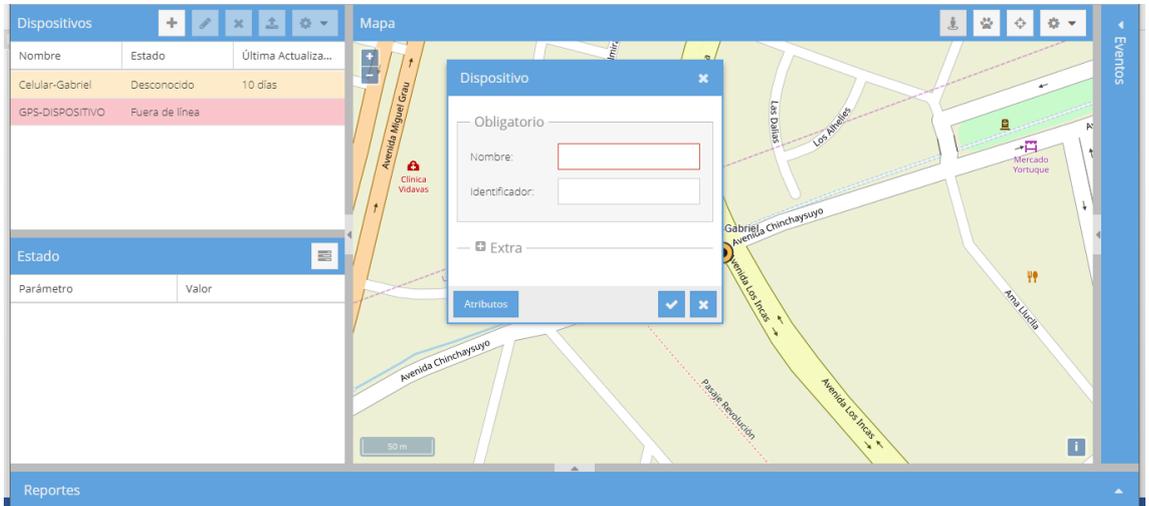


Figura 115 Interfaz Traccar -HU 043 –Registro de GPS - Apendice A

- ✓ Crear tablas y procedimientos almacenados en la base de datos.

```

179
180 CREATE TABLE GPS(codigo smallint unique not null,
181                  codigoReferencia character varying(16) null,
182                  primary key(codigo),
183                  vigencia boolean);
184
185 CREATE TABLE GPSUnidad(codigo smallserial not null,
186                        codigoUnidad smallint not null references Unidad(codigo),
187                        codigoGPS smallint not null references GPS(codigo),
188                        vigencia boolean not null,
189                        primary key(codigo));
190

```

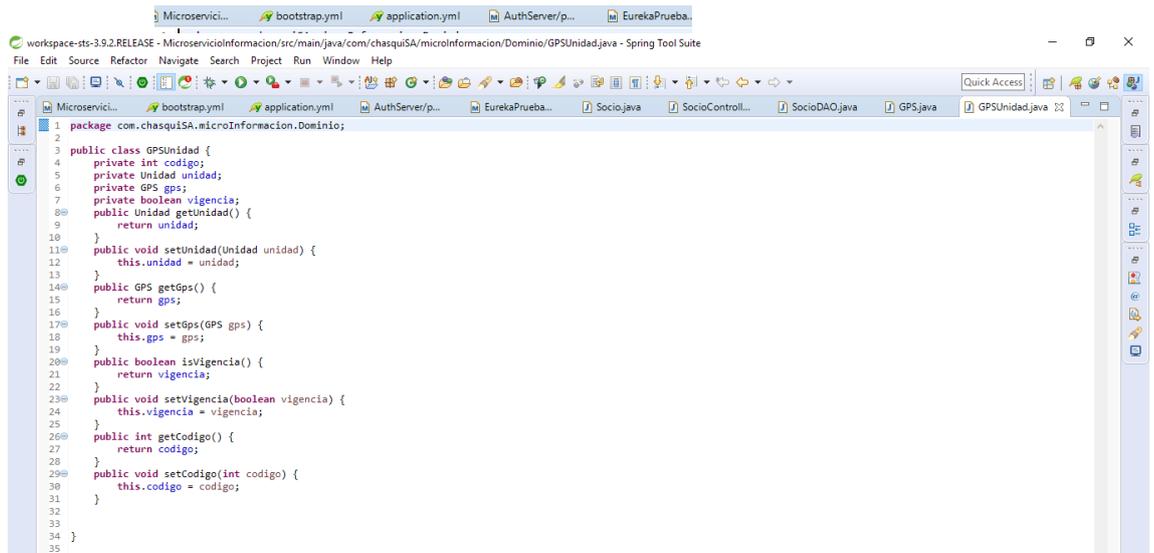
```

create or replace function pr_iGPS(p_codigoUnidad int) returns TABLE(codigo smallint , codigoReferencia character varying(7))
as
$$
/* Procedimiento: pr_iGPS()
 * Autor: Gabriel Solis
 * Fecha creación: 2018-07-12
 * Versión: 1.0
 * Parámetros: codigo unidad
 * Descripción: Se registra la asignación de un dispositivo GPS a una unidad
 */
begin
    return query(select G.codigo,G.codigoReferencia from gps G inner join GPSUnidad GU on G.codigo = GU.codigoGPS
                where GU.codigoUnidad = p_codigoUnidad);
end;
$$ LANGUAGE plpgsql;

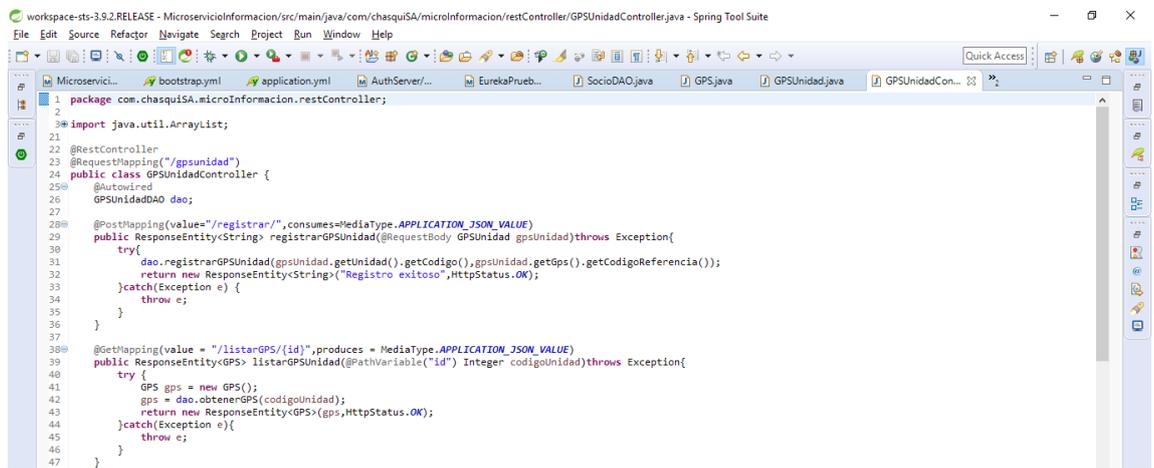
```

Figura 116 Crear Tablas y Procedimiento Almacenado en la BD -HU 043 –
Registro de GPS - Apendice A

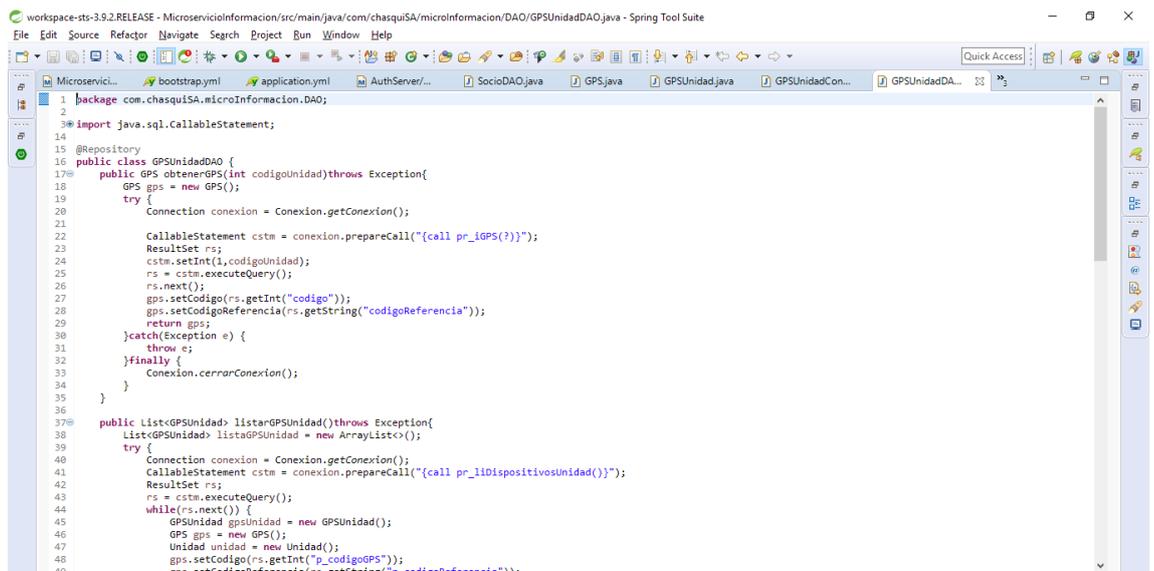
✓ Crear código correspondiente.



```
1 package com.chasquiSA.microInformacion.Dominio;
2
3 public class GPSUnidad {
4     private int codigo;
5     private Unidad unidad;
6     private GPS gps;
7     private boolean vigencia;
8     public Unidad getUnidad() {
9         return unidad;
10    }
11    public void setUnidad(Unidad unidad) {
12        this.unidad = unidad;
13    }
14    public GPS getGps() {
15        return gps;
16    }
17    public void setGps(GPS gps) {
18        this.gps = gps;
19    }
20    public boolean isVigencia() {
21        return vigencia;
22    }
23    public void setVigencia(boolean vigencia) {
24        this.vigencia = vigencia;
25    }
26    public int getCodigo() {
27        return codigo;
28    }
29    public void setCodigo(int codigo) {
30        this.codigo = codigo;
31    }
32
33 }
34
35
```



```
1 package com.chasquiSA.microInformacion.restController;
2
3 import java.util.ArrayList;
4
5 @RestController
6 @RequestMapping("/gpsunidad")
7 public class GPSUnidadController {
8     @Autowired
9     GPSUnidadDAO dao;
10
11     @PostMapping(value="/registrar", consumes="MediaType.APPLICATION_JSON_VALUE")
12     public ResponseEntity<String> registrarGPSUnidad(@RequestBody GPSUnidad gpsUnidad)throws Exception{
13         try{
14             dao.registrarGPSUnidad(gpsUnidad.getUnidad().getCodigo(),gpsUnidad.getGps().getCodigoReferencia());
15             return new ResponseEntity<String>("Registro exitoso",HttpStatus.OK);
16         }catch(Exception e){
17             throw e;
18         }
19     }
20
21     @GetMapping(value = "/listarGPS/{id}", produces = MediaType.APPLICATION_JSON_VALUE)
22     public ResponseEntity<GPS> listarGPSUnidad(@PathVariable("id") Integer codigoUnidad)throws Exception{
23         try {
24             GPS gps = new GPS();
25             gps = dao.obtenerGPS(codigoUnidad);
26             return new ResponseEntity<GPS>(gps,HttpStatus.OK);
27         }catch(Exception e){
28             throw e;
29         }
30     }
31 }
32
```



```
1 package com.chasquiSA.microInformacion.DAO;
2
3 import java.sql.CallableStatement;
4
5 @Repository
6 public class GPSUnidadDAO {
7     public GPS obtenerGPS(int codigoUnidad)throws Exception{
8         GPS gps = new GPS();
9         try {
10             Connection conexion = Conexion.getConexion();
11             CallableStatement cstm = conexion.prepareCall("{call pr_iGPS(?)}");
12             ResultSet rs;
13             cstm.setInt(1,codigoUnidad);
14             rs = cstm.executeQuery();
15             rs.next();
16             gps.setCodigo(rs.getInt("codigo"));
17             gps.setCodigoReferencia(rs.getString("codigoReferencia"));
18             return gps;
19         }catch(Exception e){
20             throw e;
21         }finally {
22             Conexion.cerrarConexion();
23         }
24     }
25
26     public List<GPSUnidad> listarGPSUnidad()throws Exception{
27         List<GPSUnidad> listaGPSUnidad = new ArrayList<>();
28         try {
29             Connection conexion = Conexion.getConexion();
30             CallableStatement cstm = conexion.prepareCall("{call pr_lidispositivosUnidad()}");
31             ResultSet rs;
32             rs = cstm.executeQuery();
33             while(rs.next()) {
34                 GPSUnidad gpsUnidad = new GPSUnidad();
35                 gps = new GPS();
36                 Unidad unidad = new Unidad();
37                 gps.setCodigo(rs.getInt("p_codigoGPS"));
38                 gps.setCodigoReferencia(rs.getString("p_codigoReferencia"));
39             }
40         }
41     }
42 }
43
```

Figura 117 Crear código correspondiente -HU 043 –Registro de GPS - Apendice A

✓ H035 – Registrar sanciones

En esta historia de usuario se realizó un API REST consultando los registros en base de datos brindados por traccar.

✓ Diseñar la interfaz.

Registro de sanción

Tiempo de retraso total

Minutos sanción

Motivo

Justificación

Aplicar Sanción Cancelar

Figura 118 Diseñar la interfaz -HU 035 –Registrar Sanciones-Apndice A

✓ Crear tabla y procedimiento almacenado en la base de datos.

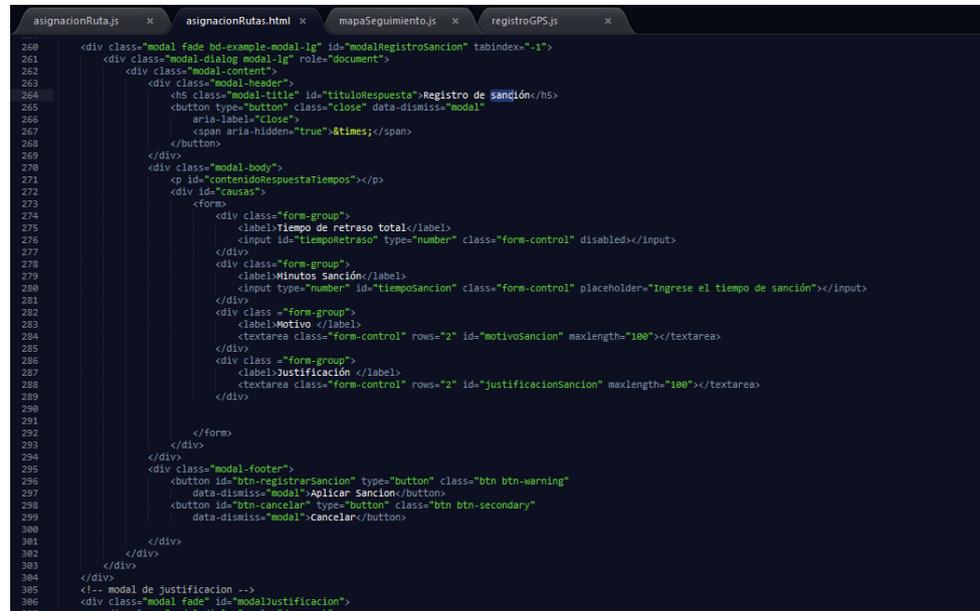
```
horaControl time not null,
minutosTolerancia smallint not null,
horaGPS time,
diferencia real,
codigoDetalleTarjeta integer not null references detalleTarjeta(codigo),
primary key(codigo);
/*codigoTiempoEstablecido smallint not null references tiempoestablecido(codigo)*/

CREATE TABLE sancion(codigo smallserial not null,
codigoRegistroUnidad smallint not null references registrounidad(codigo),
justificacion character varying(300),
motivo character varying(300),
estado character varying(1) not null,
tiempoRetraso real,
sancionTiempo real,
sancionMonto real,
vigencia boolean not null,
codigoDetalleTarjeta integer not null references detalleTarjeta(codigo),
primary key(codigo));

1 CREATE OR REPLACE FUNCTION pr_isancion(p_codigoRegistroUnidad integer,p_justificacion character varying,
2 p_motivo character varying, p_estado character varying, p_tiempoRetraso double precision,
3 p_sancionTiempo double precision,p_sancionMonto double precision,
4 p_vigencia boolean,p_codigoDetalleTarjeta integer) returns void
5 AS $$
6 /* Procedimiento: pr_isancion()
7 * Autor: Gabriel Solis
8 * Fecha creación: 2018-07-12
9 * Versión: 1.0
10 * Parámetros: codigo registro unidad , justificacion, motivo, estado , tiempo de retraso , vigencia, sancion tiempo
11 * Descripción: Se registra una sancion o justificacion
12 */
13 BEGIN
14 INSERT INTO sancion(codigoRegistroUnidad,justificacion,motivo,estado,
15 tiempoRetraso,sancionTiempo,sancionMonto,vigencia,codigoDetalleTarjeta)
16 VALUES(p_codigoRegistroUnidad,p_justificacion,p_motivo,p_estado
17 ,p_tiempoRetraso,p_sancionTiempo,p_sancionMonto,true,p_codigoDetalleTarjeta);
18
19 END;
20 $$ LANGUAGE plpgsql;
21
22
```

Figura 119 Crear Tablas y Procedimiento Almacenado en la BD -HU 035 –
Registrar Sanciones - Apendice A

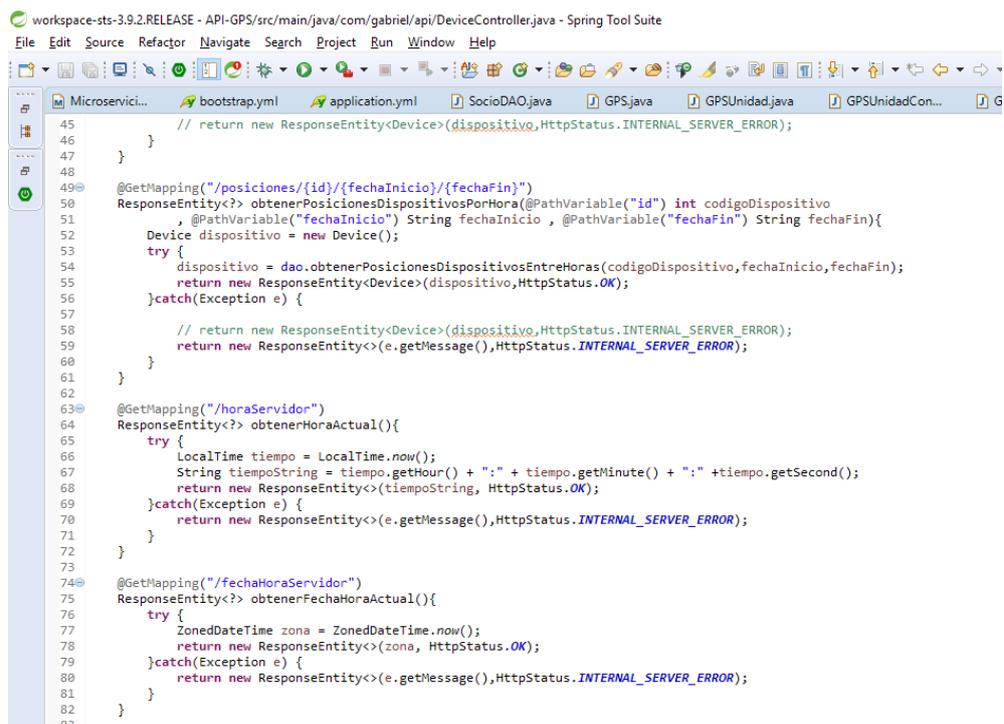
✓ Construir interfaz.



```
260 <div class="modal fade bd-example-modal-lg id="modalRegistroSancion" tabindex="-1">
261 <div class="modal-dialog modal-lg role="document">
262 <div class="modal-content">
263 <div class="modal-header">
264 <h5 class="modal-title" id="tituloRespuesta">Registro de sanción</h5>
265 <button type="button" class="close" data-dismiss="modal"
266 aria-label="Close">
267 <span aria-hidden="true">&times;</span>
268 </button>
269 </div>
270 <div class="modal-body">
271 <p id="contenidoRespuestaTiempos"></p>
272 <div id="causas">
273 <form>
274 <div class="form-group">
275 <label>Tiempo de retraso total</label>
276 <input id="tiempoRetraso" type="number" class="form-control" disabled</input>
277 </div>
278 <div class="form-group">
279 <label>Minutos Sanción</label>
280 <input type="number" id="tiempoSancion" class="form-control" placeholder="Ingrese el tiempo de sanción"></input>
281 </div>
282 <div class="form-group">
283 <label>Motivo </label>
284 <textarea class="form-control" rows="2" id="motivosancion" maxlength="100"></textarea>
285 </div>
286 <div class="form-group">
287 <label>Justificación </label>
288 <textarea class="form-control" rows="2" id="justificacionSancion" maxlength="100"></textarea>
289 </div>
290 </form>
291 </div>
292 </div>
293 <div class="modal-footer">
294 <button id="btn-registrarSancion" type="button" class="btn btn-warning"
295 data-dismiss="modal">Aplicar Sancion</button>
296 <button id="btn-cancelar" type="button" class="btn btn-secondary"
297 data-dismiss="modal">Cancelar</button>
298 </div>
299 </div>
300 </div>
301 </div>
302 </div>
303 </div>
304 </div>
305 <!-- modal de justificacion -->
306 <div class="modal fade" id="modalJustificacion">
307 <div class="modal-dialog" role="document">
```

Figura 120 Construir interfaz -HU 035 –Registrar Sanciones - Apendice A

✓ Crear el código correspondiente.



```
workspace-sts-3.9.2.RELEASE - API-GPS/src/main/java/com/gabriel/api/DeviceController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help
Microservi... bootstrap.yml application.yml SocioDAO.java GPS.java GPSUnidad.java GPSUnidadCon...
45 // return new ResponseEntity<Device>(dispositivo,HttpStatus.INTERNAL_SERVER_ERROR);
46 }
47 }
48 }
49 @GetMapping("/posiciones/{id}/{fechaInicio}/{fechaFin}")
50 ResponseEntity<?> obtenerPosicionesDispositivosPorHora(@PathVariable("id") int codigoDispositivo
51 , @PathVariable("fechaInicio") String fechaInicio , @PathVariable("fechaFin") String fechaFin){
52 Device dispositivo = new Device();
53 try {
54 dispositivo = dao.obtenerPosicionesDispositivosEntreHoras(codigoDispositivo, fechaInicio, fechaFin);
55 return new ResponseEntity<Device>(dispositivo,HttpStatus.OK);
56 }catch(Exception e) {
57 // return new ResponseEntity<Device>(dispositivo,HttpStatus.INTERNAL_SERVER_ERROR);
58 return new ResponseEntity<>(e.getMessage(),HttpStatus.INTERNAL_SERVER_ERROR);
59 }
60 }
61 }
62 }
63 @GetMapping("/horaServidor")
64 ResponseEntity<?> obtenerHoraActual(){
65 try {
66 LocalTime tiempo = LocalTime.now();
67 String tiempoString = tiempo.getHour() + ":" + tiempo.getMinute() + ":" + tiempo.getSecond();
68 return new ResponseEntity<>(tiempoString, HttpStatus.OK);
69 }catch(Exception e) {
70 return new ResponseEntity<>(e.getMessage(),HttpStatus.INTERNAL_SERVER_ERROR);
71 }
72 }
73 }
74 @GetMapping("/fechaHoraServidor")
75 ResponseEntity<?> obtenerFechaHoraActual(){
76 try {
77 ZonedDateTime zona = ZonedDateTime.now();
78 return new ResponseEntity<>(zona, HttpStatus.OK);
79 }catch(Exception e) {
80 return new ResponseEntity<>(e.getMessage(),HttpStatus.INTERNAL_SERVER_ERROR);
81 }
82 }
83 }
```

7.2 Apéndice B: Preguntas del Sprint 0

El presente formulario contiene preguntas que nos permitirá obtener información detallada de los procesos resaltantes a tener en cuenta en el posterior desarrollo del proyecto el cual nos llevará a aclarar dudas con respecto a las reglas del negocio que maneja la empresa El chasqui S.A para el optimo y eficaz desempeño de los miembros del equipo de desarrollo.

Las preguntas realizadas fueron las siguientes:

- ✓ ¿Qué documentos piden para los choferes?
 - Copia de: DNI, licencia de conducir y recibo de luz o agua.
- ✓ ¿La boleta sale a nombre de los choferes?
 - No, Sale a nombre de la Placa del vehículo.
- ✓ ¿Cuántos trabajadores hay?
 - Hay 3 en planilla:
 - 2 Controladores.
 - 1 Administradora.
 - Hay 1 Trabajador que no está en planilla:
 - 1 Gerente.
 - Hay 1 trabajador Fuera de la empresa
 - 1 Contadora.
- ✓ ¿Cuál es su función?
 - 1 Administrativo: Pagos, Compras, etc.
 - 2 Controladores: Controlan la salida de los vehículos por las diferentes rutas.
 - 1 Gerente: Asigna a que turno irán los gerentes.

- ✓ ¿Cuántas unidades pueden tener un socio?
 - Un socio puede tener mínimo 1 Carro y máximo los que pueda adquirir .
- ✓ ¿Número de acciones como mínimo?
 - Un socio puede tener mínimo 1 Carro (Genera 23 acciones).
- ✓ ¿Una unidad es solo de un socio?
 - Si, Una unidad solo le pertenece a un socio (o se le registra solo a un socio si es que hay socios que estén casados).
- ✓ ¿Unidades de reemplazo?
 - Se le da de baja una unidad inoperativa y se registra la que quieren poner de reemplazo.
- ✓ ¿Qué datos piden de las unidades?
 - Nombre de Accionista y tarjeta de propiedad.
 - Para las unidades que se van a reemplazar o se venden piden el contrato de transferencia.
- ✓ ¿En el ahorro hay alguna tasa de interés?
 - No Hay ninguna tasa de interés, pero obtenemos beneficios como:
 - Depende del ahorro que tiene un socio para brindar una cantidad de préstamo que el socio solicite.
 - Son montos establecidos, por ejemplo:
 - Si tiene 1000 de ahorro se le presta el triple: 3000.
 - Máximo de préstamo es 5000.
- ✓ ¿Si se equivoca en un pago que se hace?
 - El pago se anula porque esas boletas no van a la sunat, Lo que va es solo el monto total que generan las boletas.
- ✓ ¿Los tickets se le entregan a la contadora?

- Si, Se le entregan los dos tikets.
- ✓ ¿Qué reportes se hacen con los pagos?
 - Resumen diario de ingresos.
 - Resumen mensual de ingresos.
 - Reporte mensual de cada concepto (Préstamos, ahorros, etc) Por socio y total.
 - Cuanto ahorro tal carro de cada socio por quincena.
- ✓ ¿Los conceptos se definen en asambleas?
 - Si, Los conceptos se definen en asamblea.
- ✓ ¿Los conceptos Suelen modificarse con el tiempo?
 - Si, en una asamblea.
- ✓ ¿Cuál es el proceso para tramitar un préstamo?
 - Presentar una solicitud a la secretaria.
 - Directivos Aprueban la solicitud.
 - Verificación de deudas (Si tiene una deuda no se presta).
 - Se genera un cheque por la secretaria.
 - El préstamo se puede dividir por la cantidad de carros que tiene el socio para ser cancelado, solo cuando el socio lo desea.
- ✓ ¿Quién aprueba la solicitud?
 - Directivos.
- ✓ ¿Se genera algún documento de préstamo?
 - Si, Un recibo de salida (Todo es interno).
- ✓ ¿Se lleva el control de los préstamos?
 - Si, Quincenal y mensual.
- ✓ ¿Hubo préstamos que nunca se pagaron?

- Si, en esta ocasión se hace:
 - Liquidar con el ahorro que tiene el socio deudor y si no alcanza se le paralizan las unidades que están debiendo.
- ✓ ¿La empresa se beneficia con los préstamos?
 - No, Solo es para ayuda mutua de los socios.
 - Pero el interés que se cobra pasado los 5 meses va a la cuenta general de socios y al final se reparten entre todos.
- ✓ ¿De qué depende que una ruta tenga servicio?
 - Depende de la asamblea.
- ✓ ¿Cuántas rutas cuenta la empresa?
 - Con 4 rutas: A, B, C y D.
- ✓ ¿Qué rutas son obligatorias?
 - C y D: Rutas Largas.
 - AyB: Rutas Cortas.
- ✓ Obligatoriamente debe pasar:
 - Dos vueltas por la D.
 - Una vuelta por la C.
 - Las restantes del día A y B.
- ✓ ¿Todas las rutas pasan por los 4 relojes o todo el reloj establecidos?
 - Las rutas C y D: No pasan por el primer reloj de las Rutas Ay B.
 - Las rutas A y B: No pasan por el primer reloj de las Rutas CyD.
- ✓ ¿Cómo se modifica una ruta? ¿Se acuerda en asamblea?
 - Se Modifica por medio de los directivos.
 - No se acuerda en asamblea.
- ✓ ¿Cómo se establecen los relojes?

- Los directivos establecen los lugares donde deben ir o si se cambian de lugar.
- ✓ ¿Qué pasa si la avenida donde está el punto de control no hay pase?
 - Se cambia de ubicación el reloj o no se marca por ese reloj según el tiempo que va a estar malograda la pista y se avisa a todos los choferes.
- ✓ ¿Qué pasa si el reloj esta malogrado?
 - Se deja de marcar.
 - Se informa al controlador para que tenga en cuenta.
- ✓ ¿Cómo se establecen los tiempos estimados?
 - Los directivos lo definen.
- ✓ ¿Cómo es el orden de salida?
 - Los controladores deciden la salida de los vehículos, pero se da más por cómo van llegando los choferes a laborar.
- ✓ ¿Qué problemas tienen en este proceso Rutas?
 - Problemas de tiempos y retrasos.
 - Problemas económicos para los carros que van atrás de un carro que se está retrasando porque le quitan pasajeros.
- ✓ Reportes que ayudarían a solucionar problemas
 - Reporte de cuantos minutos de retraso por vehículo en cada vuelta.
 - Reporte de cuantos carros trabajaron un día determinado.
 - Reporte de cuantas vueltas da un carro en un día determinado.
- ✓ ¿Quién se encarga de asignar la ubicación de los controladores?
 - Gerente.
- ✓ ¿Cómo se define una sanción?
 - Sanción de tiempo: Los Directivos.

- Suspensión a los choferes por todo un día, si suspende a un chofer todo el día puede ingresar otro chofer a manejar el carro.
- ✓ ¿Las sanciones se pueden justificar?
 - Si, depende del controlador si acepta la justificación de la sanción.
- ✓ ¿Cuántos tipos de sanciones hay actualmente?
 - Sanción por tiempo.
- ✓ ¿Cuánto es la sanción en tiempo?
 - A los choferes por horas y Días.
 - Si es por retraso paralizan el vehículo y al chofer por dos horas o según lo retrasado.
- ✓ ¿Cuántas tipos de asambleas hay?
 - Hay dos tipos:
 1. Obligatoria (Marzo):
 - Se ven los estados financieros del año.
 - Utilidad o pérdida de la empresa-
 - Lo que se va a pagar a la sunat-
 - Todos los socios deben asistir-
 2. Extraordinaria (Cualquier día del año):
 - Es obligatorio que todos los socios lleguen (50% + 1) de las acciones para que se de la junta.
 - La multa por no asistir es de 40 pasajes.
 - El recibo de la multa sale a nombre del socio.
 - No se declara a la sunat.
 - Si se puede justificar una falta a la asamblea.
 - Le puedes dar un poder a otro socio para ya no ir a la asamblea.

La dirige el directorio.

- ✓ ¿Quiénes tendrán acceso al sistema?
 - 1 Directivo.
 - 1 Secretaria.
 - 1 Gerente.
- ✓ ¿Qué beneficios tienen los socios de ahorrar en la empresa?
 - Que les den préstamos.
- ✓ ¿Quién es el autorizado de aceptar un préstamo?
 - Directivos.
- ✓ ¿Cuántos terminales tiene la empresa?
 - Tiene un terminal.

7.3 Apéndice C: Desarrollo de pruebas automatizadas

- Registro de socio.
 - ✓ Criterio de aceptación.

```
#language es
Feature: Socio
@SmokeTest
Scenario: Edad de socio menor a 18
Given el usuario ingrese una fecha de nacimiento y la edad sea menor a los 18 años
When el usuario haya completado los datos y haga click en el boton registrar socio
Then El socio debe tener una edad mayor de 18 y menor que 100
```

Figura 122 Criterio de aceptación –Prueba automatizada - Registro de socio –
Apendice C

✓ Clase de prueba.

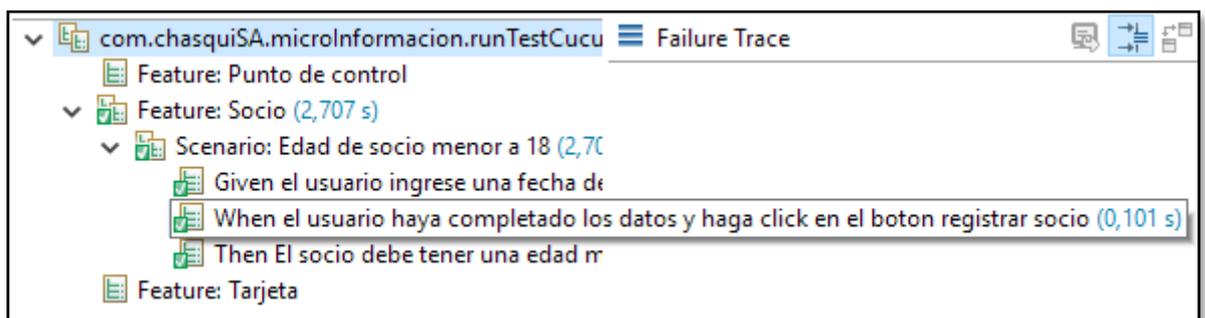
```
public class SocioTester {
    WebDriver driver;
    @Given("el usuario ingrese una fecha de nacimiento y la edad sea menor a los (\\d+) años$")
    public void fechaNacimientoSocio(int arg) {
        System.setProperty("webdriver.chrome.driver", Pathscte.PATH_CHROME_DRIVER);
        driver = new ChromeDriver();
        //operador = new OperacionesEnWeb(driver, reporte);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10000, TimeUnit.SECONDS);
        driver.get("https://chasqui-tracker.herokuapp.com/index.html");
        driver.findElement(By.id("txtUsuario")).sendKeys("gsolis");
        driver.findElement(By.id("txtClave")).sendKeys("123456");
        driver.findElement(By.id("btnIngresar")).click();
        driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
        List<WebElement> nav = driver.findElements(By.className("dropdown-item")); //.click();
        driver.navigate().to(nav.get(2).getAttribute("href"));
    }
}
```

```
@when("el usuario haya completado los datos y haga click en el boton registrar socios")
public void verificarDNI() {
    driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
    System.out.println("!Click en el botón registrar!");
    driver.findElement(By.id("txtNombres")).sendKeys("Gabriel Leonardo");
    driver.findElement(By.id("txtApellidoP")).sendKeys("Solis");
    driver.findElement(By.id("txtApellidoM")).sendKeys("Vera");
    driver.findElement(By.id("txtDNI")).sendKeys("73630607");
    driver.findElement(By.id("txtFechaNacimiento")).sendKeys("13-06-2019");
    driver.findElement(By.id("txtNombres")).sendKeys("Gabriel Leonardo");//aquí va el sexo
    driver.findElement(By.id("txtFechaRegistro")).sendKeys("20-08-2019");
    //driver.findElement(By.id("txtNombres")).sendKeys("Gabriel Leonardo");//aquí va el estado civil
    driver.findElement(By.id("txtDireccion")).sendKeys("Chicayo");
    driver.findElement(By.id("txtCorreo")).sendKeys("leo@gmail.com");
    driver.findElement(By.id("txtCelular")).sendKeys("982160177");
    driver.findElement(By.id("txtNombresPariente")).sendKeys("Paul Solis Vera");
    driver.findElement(By.id("txtTelefonoPariente")).sendKeys("932391831");
    WebElement boton = driver.findElement(By.id("btnRegistrar"));
    // TODO: handle exception
    WebDriverWait wait = new WebDriverWait(driver,60);
    boton = wait.until(ExpectedConditions.elementToBeClickable(By.id("btnRegistrar")));
    boton.sendKeys(Keys.RETURN);
}
```

```
@Then("El socio debe tener una edad mayor de 18 y menor que 100$")
public void mostrarMensajeEdad() {
    System.out.println("Holaa");
    WebElement mensaje = driver.findElement(By.id("mensajeError"));
    if(mensaje.getText().equals("El socio debe tener una edad mayor de 18 y menor que 100")) {
        System.out.println("Test Finalizado correctamente");
    }else {
        System.out.println("Criterio de aceptación incorrecto");
    }
}
```

Figura 123 Clases de prueba – Prueba automatizada – Registro de socio – Apendice C

✓ Resultado



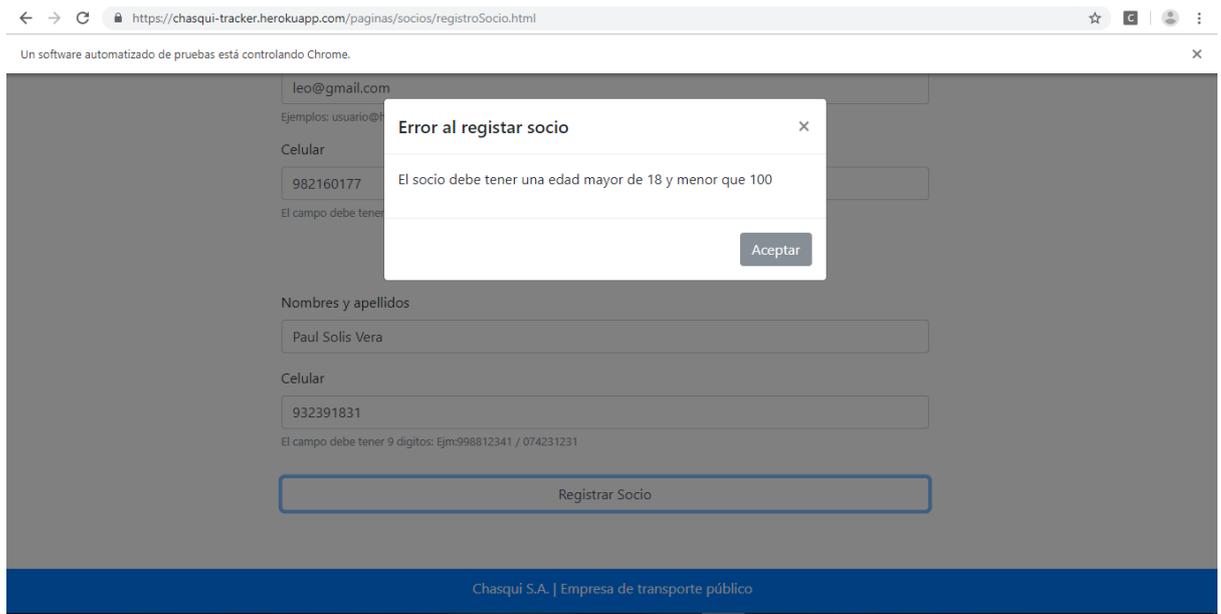


Figura 124 Resultado –Prueba automatizada– Registro de socio – Apendice C

- Registro de punto de control
 - ✓ Criterio de aceptación.

```
#language es
Feature: Punto de control
@SmokeTest
Scenario: Registrar punto de control
Given que el usuario quiera registrar un punto de control
When ingrese la dirección en el campo Dirección y de clic en el botón Buscar.
Then La aplicación mostrará un icono de color rojo en el punto buscado dando las opciones de Agregar punto de control, Visualizar punto y eliminar búsqueda
```

Figura 125 Criterio de aceptación –Prueba automatizada– Registro de punto de control – Apendice C

- ✓ Clase de prueba.

```
public class PuntoControlTester {
    WebDriver driver;

    @Given("^que el usuario quiera registrar un punto de control$")
    public void opcionRegistrarPuntoControl() {
        System.setProperty("webdriver.chrome.driver", PathsCte.PATH_CHROME_DRIVER);
        driver = new ChromeDriver();
        //operador = new OperacionesEnWeb(driver, reporte);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10000, TimeUnit.SECONDS);
        driver.get("https://chasqui-tracker.herokuapp.com/index.html");
        driver.findElement(By.id("txtUsuario")).sendKeys("gsolis");
        driver.findElement(By.id("txtClave")).sendKeys("123456");
        driver.findElement(By.id("btnIngresar")).click();
        driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
        List<WebElement> nav = driver.findElements(By.className.cClassname("dropdown-item"));
        driver.navigate().to(nav.get(6).getAttribute("href"));
    }
}
```

```

@When("ingrese la dirección en el campo Dirección y de clic en el botón Buscar$")
public void llenarCampo() {
    driver.findElement(By.id("address")).sendKeys("Colegio manuel pardo chichlayo");
    driver.findElement(By.id("submit")).click();
    driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
}

@Then("La aplicación mostrará un icono de color rojo en el punto buscado dando las opciones de Agregar punto de control, Visualizar punto y eliminar búsquedas$")
public void buscar() {
    String src = driver.findElement(By.xpath("/html/body/div[7]/div/div/div[1]/div[3]/div/div[3]/div[6]/map/area")).getAttribute("src");
    driver.findElement(By.xpath("/html/body/div[7]/div/div/div[1]/div[3]/div/div[3]/div[6]/map/area")).click();
    Assert.assertTrue(driver.findElement(By.xpath("/html/body/div[7]/div/div/div[1]/div[3]/div/div[3]/div[6]/map/area")).isEnabled());
}

```

Figura 126 Clase de prueba –Prueba automatizada – Registro de punto de control–

Apéndice C

✓ Resultado

```

com.chasquiSA.microlInformacion.runTestCucumber [Runner: JUnit 4]
  Feature: Punto de control (2,140 s)
    Scenario: Registrar punto de control (2,140 s)
      Given que el usuario quiera registrar un punto de control (0,764 s)
      When ingrese la dirección en el campo Dirección y de clic en el botón Buscar (1,372 s)
      Then La aplicación mostrará un icono de color rojo en el punto buscado dando las opciones de Agregar punto de control, Visua

```

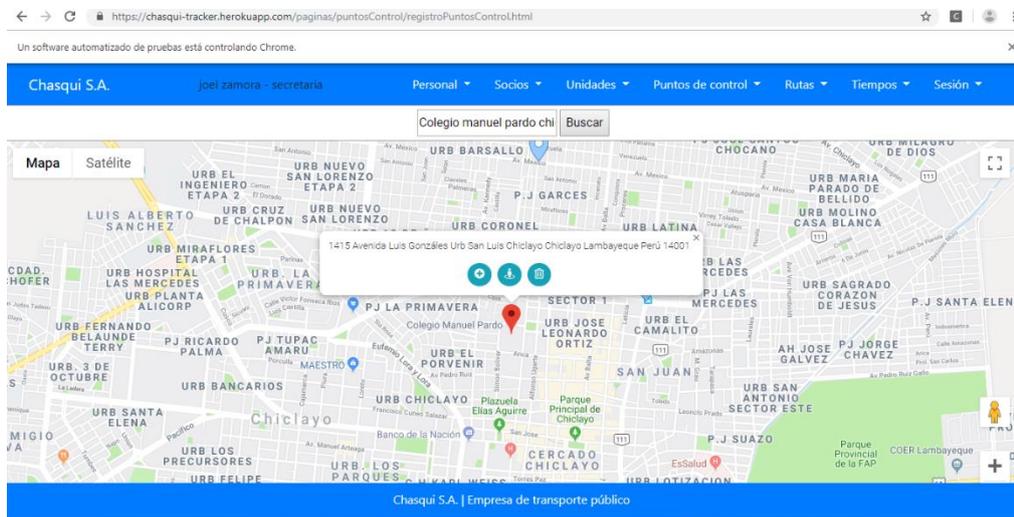


Figura 127 Resultado –Prueba automatizada – Registro de punto de control–

Apéndice C

7.4 Apéndice D: Entorno de trabajo

- Para el desarrollo con java se instaló el JDK 8 y configuramos las variables de entorno correspondientes.

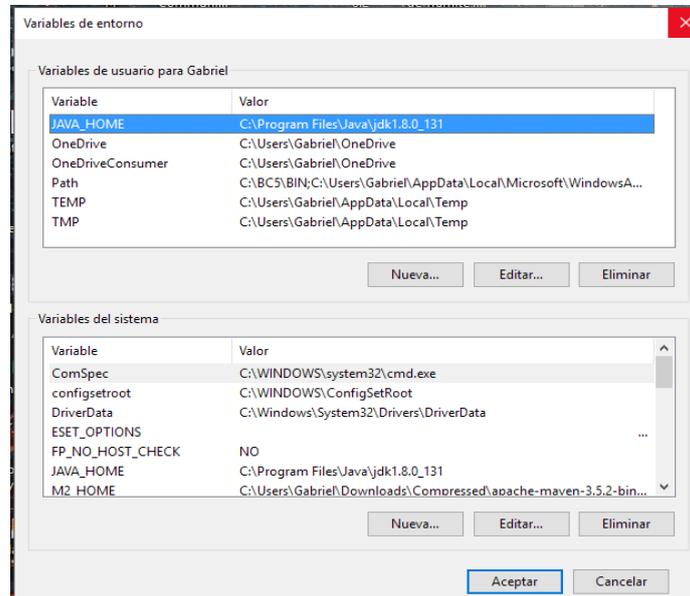
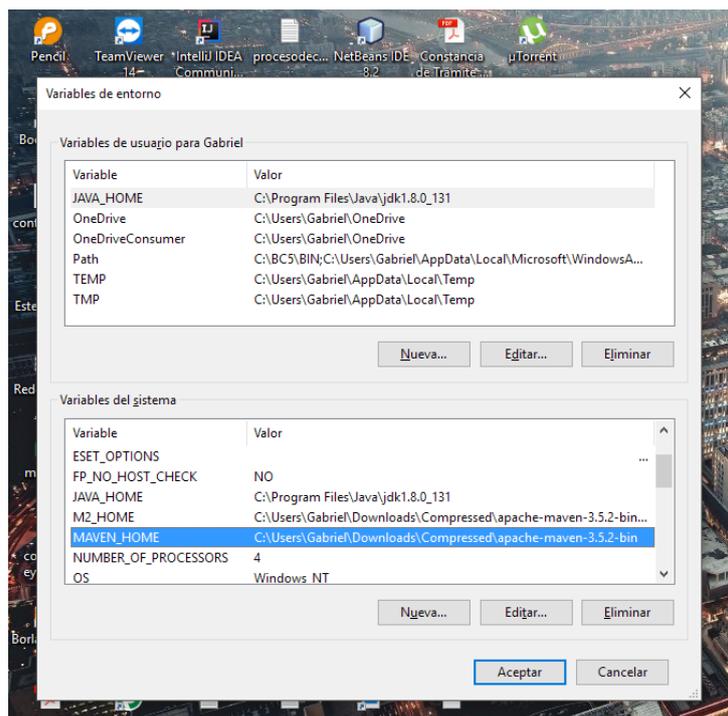


Figura 128 Configuración de variables de entorno JAVA_HOME – JDK 8 – Entorno de trabajo

- Para el manejo de dependencias y librerías en java instalamos maven y configuramos la variable de entorno respectiva.



*Figura 129 Configuración de variables de entorno MAVEN_HOME –
Entorno de trabajo*

- Para la obtención de las ubicaciones de gps las unidades utilizaron el siguiente dispositivo: “GPS TRACKER”



Figura 130 GPS TRACKER – Entorno de trabajo

- Spring Tool Suite (STS).

Para la codificación del proyecto utilizamos el IDE STS ide que ayuda a la creación y configuración de proyectos del ecosistema Spring.

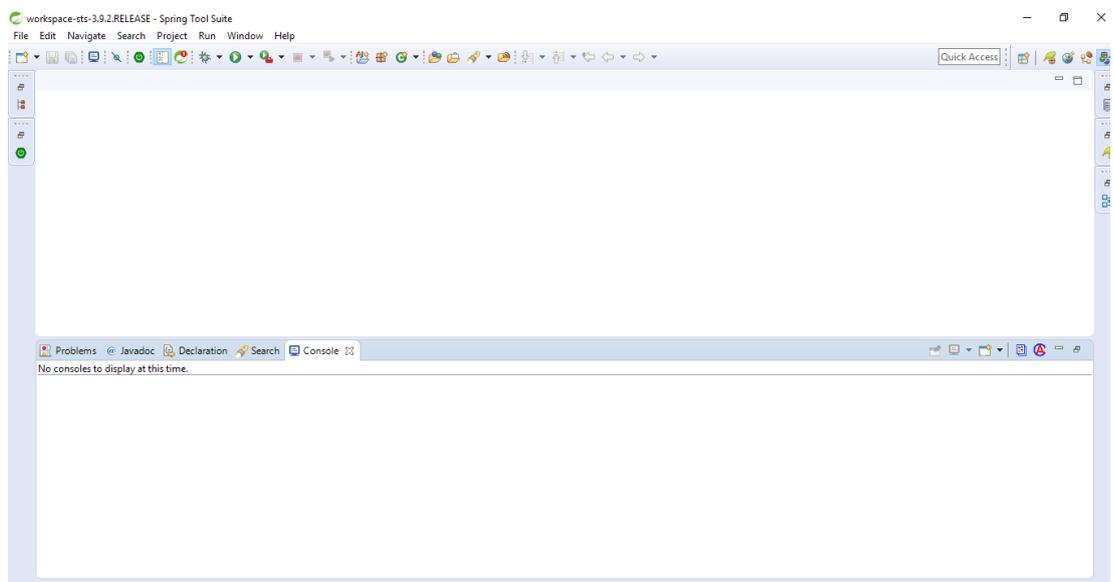


Figura 131 Spring Tool Suite (STS) – Entorno de trabajo

- Base de datos Postgresql.

Para el almacenamiento de los datos usamos el motor de base de datos

Postgresql con su cliente PGAdmin4.

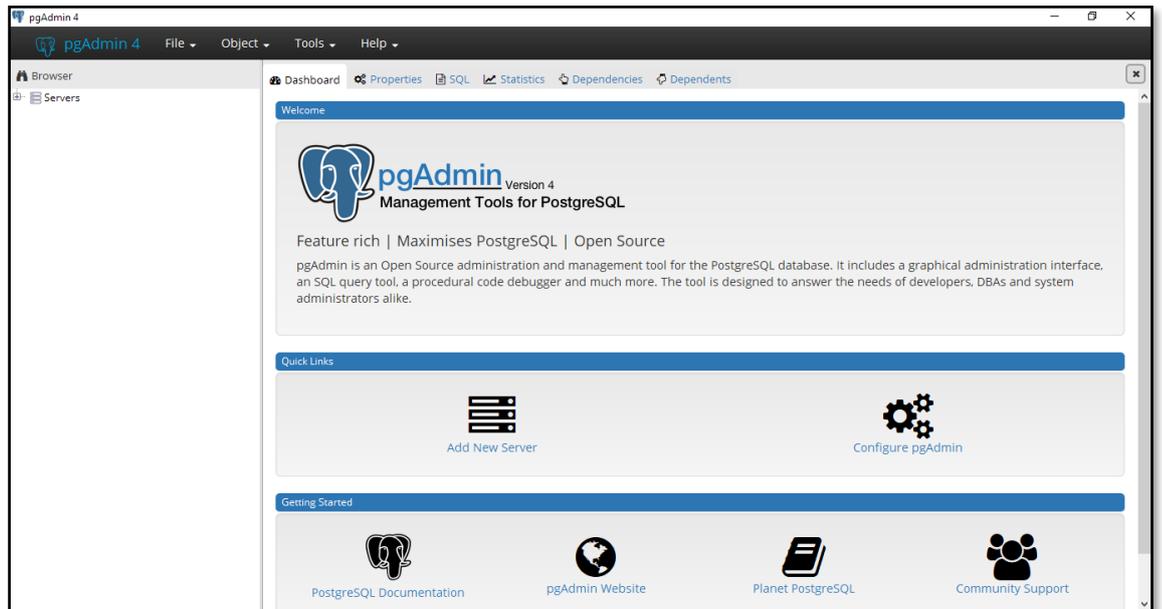


Figura 132 Postgresql (PGAdmin4) – Entorno de trabajo

- Configuración de Tomcat en Digital Ocean.

Para el despliegue del API REST que procesa y expone algunos datos de proceso instalamos el servidor de aplicaciones Tomcat 8.5.

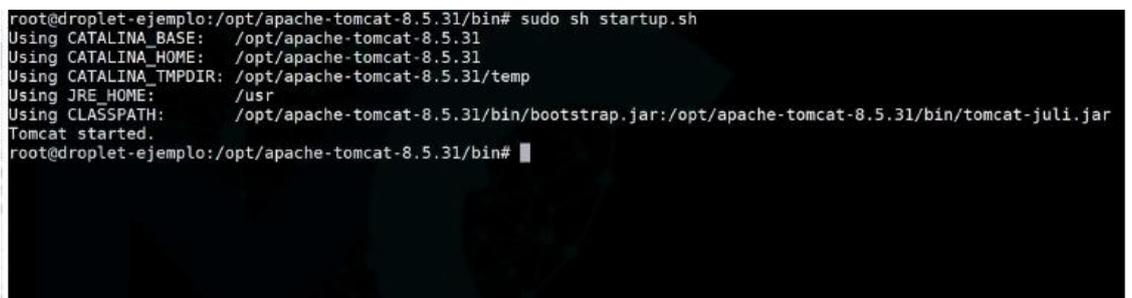
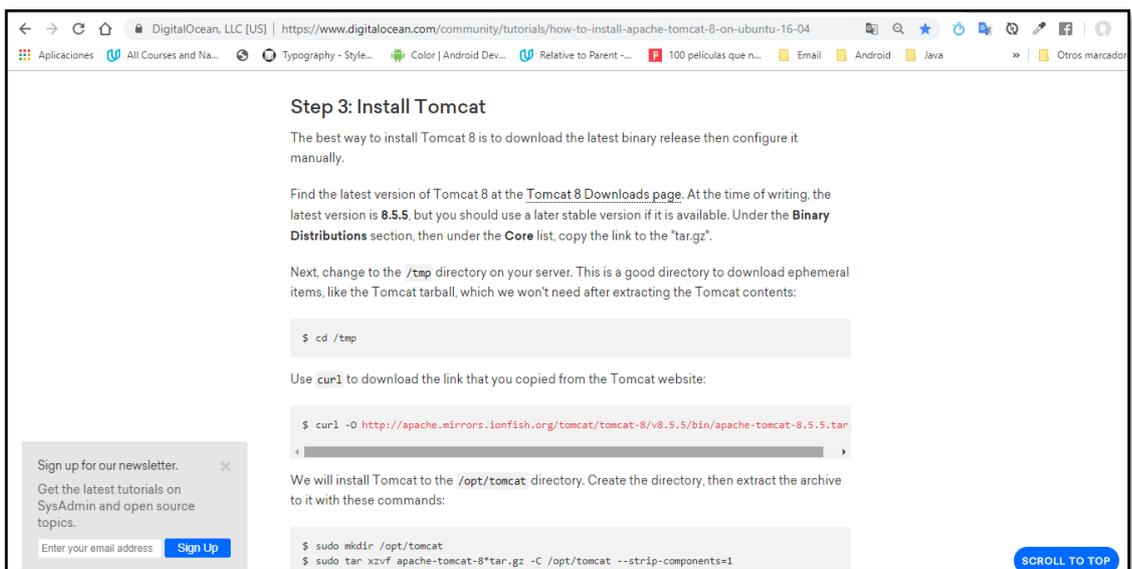
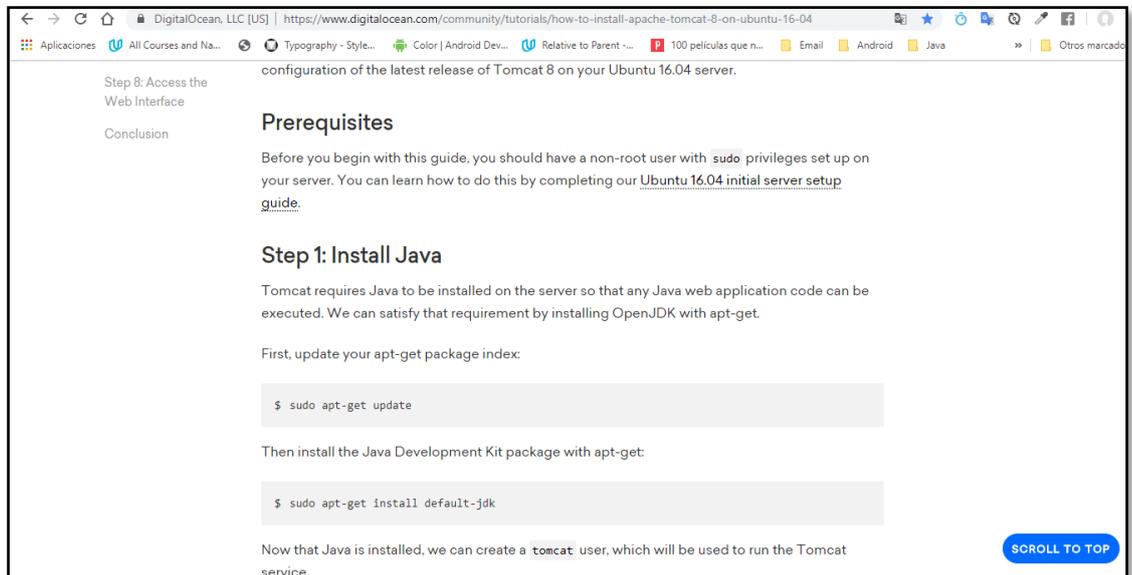


Figura 133 Configuración de Tomcat en Digital Ocean – Entorno de trabajo

- Configuración de microservicios.

Para la generación del servicio RestFul, conexión a base de datos, la implementación del módulo de seguridad Outh – 2 e interacción con Eureka y Gateway obtendremos sus dependencias por medio del archivo pom.xml y la configuración del proyecto por el archivo application.yml.

✓ Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>MicroservicioInformacion</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>MicroservicioInformacion</name>
  <description>Microservicio de informacion de la
empresa</description>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.9.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
```

```

        <java.version>1.8</java.version>
</properties>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-
dependencies</artifactId>
            <version>Edgware.SR2</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-eureka-
server</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
    </dependency>

```

```

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <scope>runtime</scope>
</dependency>
<!-- Spring OAUTH2 y Seguridad -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-oauth2</artifactId>
</dependency>
<dependency>
    <groupId>net.sf.jasperreports</groupId>
    <artifactId>jasperreports</artifactId>
    <version>6.5.1</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.lowagie/itext --
>
<dependency>
    <groupId>com.lowagie</groupId>
    <artifactId>itext</artifactId>
    <version>2.1.7</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.olap4j/olap4j --
>
<dependency>

```

```

        <groupId>org.olap4j</groupId>
        <artifactId>olap4j</artifactId>
        <version>1.2.0</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-security</artifactId>
    </dependency>
<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-jvm -->
    <dependency>
        <groupId>info.cukes</groupId>
        <artifactId>cucumber-jvm</artifactId>
        <version>1.2.5</version>
        <type>pom</type>
    </dependency>
<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-junit -
->
    <dependency>
        <groupId>info.cukes</groupId>
        <artifactId>cucumber-junit</artifactId>
        <version>1.2.5</version>
        <scope>test</scope>
    </dependency>
<!-- https://mvnrepository.com/artifact/info.cukes/cucumber-spring
-->
    <dependency>
        <groupId>info.cukes</groupId>
        <artifactId>cucumber-spring</artifactId>
        <version>1.2.5</version>
        <scope>test</scope>

```

```

        </dependency>
      <dependency>
        <groupId>org.scala-lang</groupId>
        <artifactId>scala-library</artifactId>
        <version>2.11.0</version>
      </dependency>
    </dependencies>
    <build>
      <plugins>
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-
plugin</artifactId>
        </plugin>
      </plugins>
    </build>
    <repositories>
      <repository>
        <id>spring-snapshots</id>
        <name>Spring Snapshots</name>
        <url>https://repo.spring.io/libs-snapshot</url>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </repository>
    </repositories>
  </project>

```

✓ Application.yml

server:

```

    port: ${PORT:9090}

eureka:
  client:
    service-url:
      defaultZone: https://chasqui-eureka.herokuapp.com/eureka/
#http://localhost:8761/eureka/ #https://eureka-
mito.herokuapp.com/eureka/
    instance:
      statusPageUrlPath: /my_info
      healthCheckUrlPath: /my_health
      metadataMap:
        appOwner: Tesis
        description: microservicio para el control de tiempos y
recorridos

#para seguridad
security:
  oauth2:
    resource:
      userInfoUri: https://chasqui-auth.herokuapp.com/user
#http://localhost:9999/user

```

- Configuración de proyecto AuthServer (Seguridad)

La configuración se realiza mediante los archivos pom.xml y application.yml

✓ Pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>AuthServer</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>AuthServer</name>
  <description>Prueba de eureka</description>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.10.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
    <spring-cloud.version>Edgware.SR2</spring-cloud.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>

```

```

</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-oauth2</artifactId>
</dependency>
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>

```

```

    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>${spring-cloud.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>

```

✓ Application.yml

```

server:
  port: ${PORT:9999}
spring:
  datasource:
    driverClassName: org.postgresql.Driver
    url: jdbc:postgresql://ec2-174-129-28-38.compute-
1.amazonaws.com:5432/d7ite1rtej9da
    username: epnvssgbxirdfv
    password:

```

```
eureka:  
  client:  
    serviceUrl:  
      defaultZone: #https://chasqui-eureka.herokuapp.com/eureka/
```

```
security:  
  oauth2:  
    client:  
      client-id: chasqui  
      client-secret: chasquisecret  
      authorized-grant-types:  
password,client_credentials,authorization_code,refresh_token  
      scope: read,write
```

- Configuración del proyecto Gateway

La configuración se realiza mediante los archivos pom.xml y application.yml.

✓ Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>com.example</groupId>  
  <artifactId>Gateway</artifactId>  
  <version>0.0.1-SNAPSHOT</version>  
  <packaging>jar</packaging>  
  <name>Gateway</name>
```

```

<description>Prueba de eureka</description>
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.10.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
  <java.version>1.8</java.version>
  <spring-cloud.version>Edgware.SR2</spring-cloud.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-zuul</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
<dependencyManagement>

```

```

        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-
dependencies</artifactId>
                <version>${spring-cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-
plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>

```

✓ Application.yml

server:

port: \${PORT:8099} #8099

eureka:

client:

registerWithEureka: false

serviceUrl:

```
    defaultZone: https://chasqui-eureka.herokuapp.com/eureka/
#http://localhost:8761/eureka/ #https://eureka-
mito.herokuapp.com/eureka/

zuul:
  host: #para seguridad
    connect-timeout-millis: 100000 #para seguridad
    socket-timeout-millis: 100000 #para seguridad
  routes:
    uaa: #para seguridad
      path: /uaa/**
      sensitiveHeaders: #para seguridad
      serviceId: AUTH-SERVER
      url: https://chasqui-auth.herokuapp.com/
    forma1:
      path: /micro-client/**
      sensitiveHeaders: #para seguridad
      url: https://chasqui-micro-basico.herokuapp.com/
#http://localhost:9000/ #https://spring-micro-mito.herokuapp.com/
estaba con 9090
    forma3:
      path: /micro-control/**
      sensitiveHeaders: #para seguridad
      url: https://chasqui-micro-
control.herokuapp.com/#http://localhost:9000/
    forma2:
      path: /micro-api/**
      sensitiveHeaders: #para seguridad
      service-id: MICRO-CLIENT
```

- Configuración de proyecto Eureka.

La configuración se realiza mediante los archivos pom.xml y application.yml.

✓ Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>EurekaPrueba</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>EurekaPrueba</name>
  <description>Prueba de eureka</description>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.10.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
```

```

        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-eureka-
server</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-
dependencies</artifactId>
            <version>Edgware.SR2</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-
plugin</artifactId>
        </plugin>
    </plugins>

```

```
</build>
<repositories>
  <repository>
    <id>spring-snapshots</id>
    <name>Spring Snapshots</name>
    <url>https://repo.spring.io/libs-snapshot</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
</project>
```

✓ Application.yml

```
server:
  port: ${PORT:8761}

spring:
  application:
    name: discovery-server

eureka:
  instance:
    hostname: localhost
```

Para el alojamiento de los microservicios el conjunto de herramientas se utilizó Digital Ocean y Heroku.

- Dashboard de Heroku.

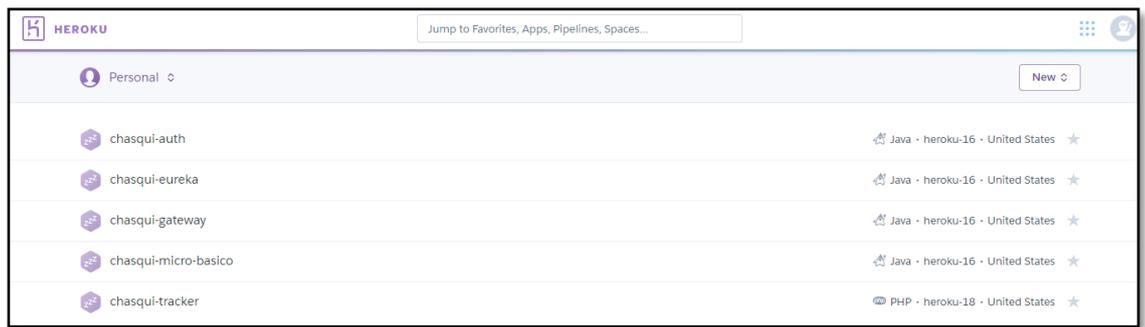


Figura 134 Dashboard de Heroku – Entorno de trabajo

- Dashboard de Digital Ocean.

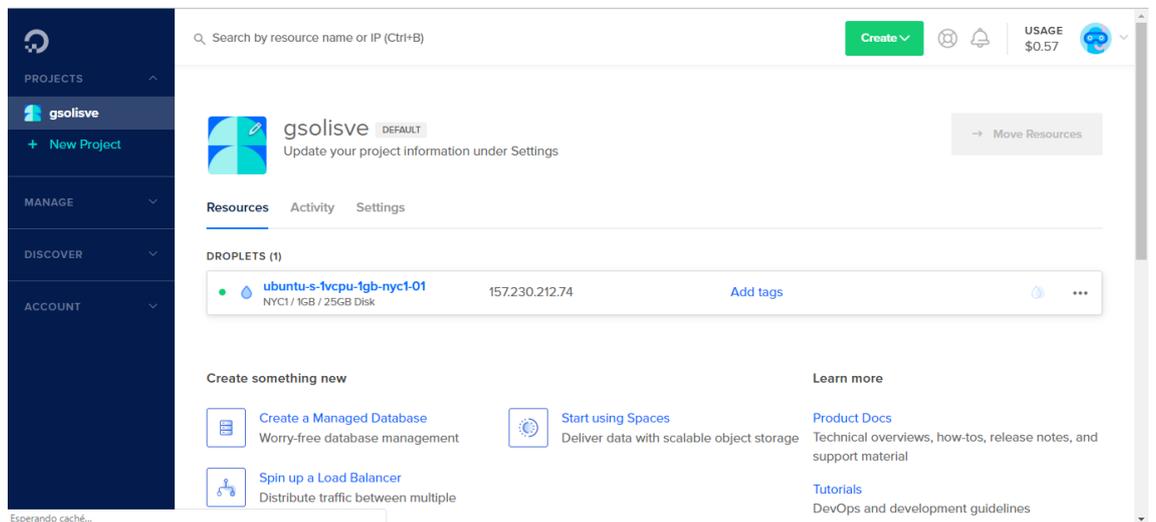


Figura 135 Dashboard de Digital Ocean – Entorno de trabajo