

**FACULTAD DE CIENCIAS FISICAS Y
MATEMÁTICAS
ESCUELA PROFESIONAL DE COMPUTACIÓN E
INFORMÁTICA**

TESIS

“SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE
INSCRIPCIÓN DE POSTULANTES DE LA UNIVERSIDAD
NACIONAL AUTÓNOMA DE CHOTA - 2015”.

*Para optar el Título Profesional de Ingeniero en
Computación e Informática.*

AUTORES

Bach. Luis Edgardo Pérez Bonilla

Bach. Katherine Vanessa Quintos Estrada

ASESOR

Ing. Roger Ernesto Alarcón García

Lambayeque, Agosto 2016 - Perú

TESIS

“SISTEMA BAJO ENTORNO WEB PARA EL
PROCESO DE INSCRIPCIÓN DE
POSTULANTES DE LA UNIVERSIDAD
NACIONAL AUTÓNOMA DE CHOTA - 2015”.

AUTORES

Bach. Luis Edgardo Pérez Bonilla

Bach. Katherine Vanessa Quintos Estrada

ASESOR

Ing. Roger Ernesto Alarcón García

TESIS

“SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCIÓN DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA - 2015”.

JURADO

Ing. Gisella Luisa Elena Maquén Niño
(Presidente)

Ing. Franklin Edinson Terán Santa Cruz
(Secretario)

Ing. Denny John Fuentes Adrianzén
(Vocal)

TESIS

“SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCIÓN DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA - 2015”.

Autores

Bach. Luis Edgardo Pérez Bonilla

Bach. Katherine Vanessa Quintos Estrada

Ing. Roger Ernesto Alarcón García
(Asesor)

Pérez Bonilla Luis Edgardo - Quintos Estrada Katherine Vanessa

DEDICATORIA

Dedico esta tesis a mi familia en especial a mi padre José y a mi madre Luz María quienes me dieron la vida y a mi hermana que me da su apoyo en todo momento, y son ellos quienes me dan el aliento para seguir adelante.

Luis

Dedicado a mis padres Camilo y Rosario por su confianza y apoyo absoluto y a mis hermanas Evelyn y Paola por estar siempre a mi lado y ser mis compañeras y cómplices incondicionales.

Katherine

A aquellas personas que no están en este momento con nosotros y desde el cielo nos siguen bendiciendo para seguir adelante.

AGRADECIMIENTOS

Queremos agradecer en la presente tesis a Dios y a la Virgen por brindarnos cada día su bendición y permitirnos cumplir uno de nuestros sueños tan anhelados, como también a nuestra alma mater, a nuestros docentes que aportaron con un granito de arena en nuestra formación universitaria y han logrado en nosotros poder terminar nuestros estudios con éxito, en especial a nuestro asesor, el Ing. Roger Ernesto Alarcón García, por su esfuerzo y dedicación.

A la Universidad Nacional Autónoma de Chota en especial a la Presidenta Dra. Florencia Adelina Arteaga Torres, al Vicerrector Académico Dr. Edin Edgardo Alva Plasencia y al Vicerrector de Investigación Dr. Edgar Carlos Quispe Peña, quienes integran la Comisión Organizadora e hicieron posible el desarrollo de nuestra Tesis.

A nuestros abuelos, Rosario, Araminda, Blanca, Agustina, Adelfina, Pedro, Segundo Felipe, Artemio y Pedro, quienes en estos momentos gozan del descanso eterno pero siguen viviendo en nuestros corazones, sin importar en donde estén queremos darles las gracias por formar parte de nosotros.

A Davicito, por regalarme cuatro años de inmensa alegría, ahora desde el cielo sé que me estás cuidando y guiando mis pasos, eres mi motivación para seguir adelante y ser mejor persona, por siempre, mi sobrino favorito.

Son muchas las personas que han formado parte de nuestra vida profesional a las que nos encantaría agradecerles por su amistad, consejos, apoyo y ánimo. Para ellos muchas gracias y que Dios los bendiga.

RESUMEN

En la actualidad la Universidad Nacional Autónoma de Chota no cuenta con un sistema que optimice su proceso de inscripción de postulantes, porque los registros se realizan manualmente y mediante el programa “Microsoft Excel”. Los postulantes deben acercarse a las instalaciones de la universidad con su voucher para poder inscribirse y obtener su carnet que le permitirá ingresar al examen de admisión o centro pre, lo que ocasiona que el proceso requiera una mayor inversión de tiempo y se torne más complejo y ante una inadecuada administración puede afectar a la entidad en mención.

En la propuesta de solución se desarrolló un sistema web para optimizar el proceso de inscripción, con la finalidad que los postulantes accedan al sistema en forma virtual y puedan lograr su inscripción desde la comodidad de su casa o una cabina de internet.

ABSTRACT

At present, the National Autonomous University of Chota does not have a system to optimize the registration process of applicants, because the records are performed manually and using "Microsoft Excel". Applicants should approach the university facilities with their voucher to register and obtain their carnet that will allow them to enter the entrance examination or pre center, which causes the process requires greater time, becomes more complex, and to improper management can affect the entity.

In the proposed solution, a web system to optimize the registration process was developed, in order that applicants have access to the system virtually their registration from the comfort of your home or Internet cabins.

ÍNDICE

DEDICATORIA	v
AGRADECIMIENTOS	vi
RESUMEN	vii
ABSTRACT	viii
INTRODUCCIÓN	1
1. ASPECTO INFORMATIVO	4
1.2. Autor(es)	4
1.3. Centro o Instituto de Investigación	4
1.4. Área de investigación	4
1.5. Lugar de ejecución	5
1.6. Duración estimada	5
1.7. Fecha de Inicio	5
1.8. Fecha de Finalización	5
1.9. Beneficiarios del Proyecto	5
2. BASE LEGAL Y ASPECTO DE LA ORGANIZACIÓN	7
2.1. Base Legal	7
2.2. Aspecto de la Organización	10
2.2.1. Nombre	10
2.2.2. Ubicación Geográfica	10
2.2.3. Visión	10
2.2.4. Misión	10
2.2.5. Objetivos	10
2.2.6. Funciones	11
2.2.7. Principios	11
2.2.8. Organigrama	13
2.2.9. Personal	14
2.2.9.1. Presidencia	14
2.2.9.2. Vicepresidencia Académica	14
2.2.9.3. Vicepresidencia de Investigación	14
3. ASPECTO DE LA INVESTIGACIÓN	16
3.1. Marco Lógico	16
3.1.1. Planteamiento del Problema Científico.	16
3.1.2. Revisión Bibliográfica.	18
3.1.2.1. Antecedentes	18
3.1.2.2. Base Teórica	19

3.1.2.2.1.	Sistemas Web	20
3.1.2.2.2.	Lenguaje de Desarrollo	21
	JAVA	21
3.1.2.2.3.	Gestor de Base de Datos	23
	POSTGRESQL	23
3.1.2.2.4.	Metodología de Modelado	24
	METODOLOGÍA ÁGIL	24
	PROGRAMACIÓN EXTREMA	28
3.1.2.2.5.	Arquitectura Cliente / Servidor.....	45
	ARQUITECTURA MVC	45
3.1.2.2.6.	Complementos	50
	JAVASCRIPT	50
	JDK (JAVA DEVELOPMENT KIT).....	52
	AJAX	52
	JQUERY.....	53
	HTML 5	53
	CSS 3.....	54
	APPLET	54
3.1.3.	Formulación del Problema Científico.....	56
3.1.4.	Hipótesis	56
3.1.5.	Operacionalización de Variables.....	56
3.1.6.	Objetivos	57
3.1.6.1.	Objetivo General	57
3.1.6.2.	Objetivo Específicos.....	57
3.1.7.	Justificación e Importancia	57
3.1.7.1.	Justificación.....	57
3.1.7.2.	Importancia	58
3.2.	Marco Metodológico	58
3.2.1.	Diseño de contrastación de Hipótesis	58
3.2.2.	Tipo de investigación.....	58
4.	ASPECTO ADMINISTRATIVO.....	60
4.1.	Cronogramas de actividades.....	60
4.2.	Presupuesto	62
4.2.1.	Bienes	62
4.2.2.	Inversión.....	63
4.2.3.	Costo de Operación	65
4.3.	Financiación	66
4.4.	Análisis de Factibilidad.....	66
4.4.1.	Factibilidad Operativa.....	66
4.4.2.	Factibilidad Técnica.....	67
4.4.3.	Factibilidad Económica	68
4.4.3.1.	Beneficios.....	68
4.4.3.1.1.	Beneficios tangibles	68

4.4.3.1.2. Beneficios Intangibles	68
4.4.3.1.3. Determinación de Factibilidad.....	68
5. DESARROLLO DEL SISTEMA.....	71
5.1. Fase I: Exploración.....	71
5.1.1. Historias de Usuario	71
5.2. Fase 2: Planeación.....	80
5.2.1. Planificación de Iteraciones.....	80
5.3. Fase 3: Iteraciones.....	81
5.3.1. Primera Iteración	81
5.3.2. Segunda Iteración	87
5.3.3. Tercera Iteración	97
5.3.4. Cuarta Iteración.....	104
5.3.5. Quinta Iteración.....	107
5.3.6. Sexta Iteración	111
5.3.7. Séptima Iteración	117
5.4. Fase 4: Producción	129
6. PRUEBAS DE USO DEL SISTEMA	131
6.1. Objetivos	131
6.2. Estrategias de Prueba.....	131
6.3. Actividades a realizar	131
6.3.1. Pruebas Unitarias.....	131
6.3.2. Pruebas de Integración	131
6.3.3. Pruebas de Implantación.....	132
6.3.4. Pruebas de Aceptación	132
6.4. Características que no serán probadas.....	173
6.5. Personal administrativo a quienes fue aplicado la capacitación y pruebas de funcionalidad del sistema	173
7. SEGURIDAD	175
7.1. Sistema Administrativo	175
7.2. Sistema de Inscripción	178
8. MANTENIMIENTO Y ACTUALIZACIÓN.....	183
8.1. Vacuum:	183
8.2. Reindexacion:	183
CONCLUSIONES Y RECOMENDACIONES	184
DEFINICIÓN DE TÉRMINOS.....	185
REFERENCIAS BIBLIOGRÁFICAS.....	187
Anexos	189
A. Documentos	189

B. Instalación y configuración de Apache Tomcat	194
C. Instalación y Configuración del Lector de Huella Digital	210
D. Permisos del Applet.....	217

ÍNDICE DE TABLAS

Tabla 3.1: Comparación entre metodologías ágiles y metodologías tradicionales	28
Tabla 3.2: Operacionalización de variables.....	56
Tabla 4.1: Bienes	62
Tabla 4.2: Resumen de bienes.....	62
Tabla 4.3: Infraestructura - Licencia de Software	63
Tabla 4.4: Infraestructura - Materiales de Oficina.....	63
Tabla 4.5: Resumen de infraestructura	64
Tabla 4.6: Servicios	64
Tabla 4.7: Capacitación en el uso del sistema	64
Tabla 4.8: Resumen Servicios.....	65
Tabla 4.9: Resumen inversión	65
Tabla 4.10: Costo fijo	65
Tabla 4.11: Costo variable.....	66
Tabla 4.12: Resumen	66
Tabla 4.13: Factibilidad Técnica.....	67
Tabla 4.14: Estado de ganancias y pérdidas.....	68
Tabla 4.15: Flujo de efectivo	69
Tabla 5.1: Personas entrevistadas en la fase de exploración del proyecto	71
Tabla 5.2: Historia de Usuario 1. Inscripción de Postulante	71
Tabla 5.3: Historia de Usuario 2. Registro de Fotos	72
Tabla 5.4: Historia de Usuario 3. Registro de Huellas	72
Tabla 5.5: Historia de Usuario 4. Verificar Datos de Postulante	73
Tabla 5.6: Historia de Usuario 5. Impresión de carnet	73
Tabla 5.7: Historia de Usuario 6. Cambio de modalidad (De extraordinario a ordinario).....	74
Tabla 5.8: Historia de Usuario 7. Cambio de modalidad (De CPU a Ingresante CPU).....	74
Tabla 5.9: Historia de Usuario 8. Listado de postulantes	75
Tabla 5.10: Historia de Usuario 9. Total de postulantes por modalidad y escuela	75
Tabla 5.11: Historia de Usuario 10. Gestión de personal	76
Tabla 5.12: Historia de Usuario 11. Iniciar sesión	76
Tabla 5.13: Historia de Usuario 12. Modificar contraseña	77
Tabla 5.14: Historia de Usuario 13. Asignar permisos a usuarios	77

Tabla 5.15: Historia de Usuario 14. Gestión de modalidad	78
Tabla 5.16: Historia de Usuario 15. Gestión de escuela	78
Tabla 5.17: Historia de Usuario 16. Gestión de tarifario	79
Tabla 5.18: Historia de Usuario 17. Gestión de proceso de admisión	79
Tabla 5.19: Iteraciones del proyecto	80
Tabla 5.20: Historia 10 - Tarea 1. Diseño de la Base de Datos.....	81
Tabla 5.21: Historia 10 - Tarea 2. Diseño de la interfaz del sistema administrativo.....	83
Tabla 5.22: Historia 10 - Tarea 3. Diseño de interfaz de Gestión de Personal. 84	
Tabla 5.23: Historia 10 - Tarea 4. Desarrollo de las principales funciones en base de datos	84
Tabla 5.24: Historia 10 - Tarea 5. Conexión a base de datos y registro de Personal.....	85
Tabla 5.25: Historia 14 - Tarea 1. Diseño de interfaz de Gestión de modalidad	87
Tabla 5.26: Historia 14 - Tarea 2. Desarrollo de las funciones en base de datos para Modalidad y Tipo de Modalidad	87
Tabla 5.27: Historia 14 - Tarea 3. Conexión del formulario Gestión de modalidad con la base de datos	88
Tabla 5.28: Historia 15 - Tarea 1. Diseño de interfaz de Gestión de escuela ..	89
Tabla 5.29: Historia 15 - Tarea 2. Desarrollo de las funciones en base de datos para Escuela y Grupo	90
Tabla 5.30: Historia 15 - Tarea 3. Conexión del formulario Gestión de escuela con la base de datos.....	90
Tabla 5.31: Historia 16 - Tarea 1. Diseño de interfaz de Gestión de tarifario...	92
Tabla 5.32: Historia 16 - Tarea 2. Desarrollo de las funciones en base de datos para Tarifario	92
Tabla 5.33: Historia 16 - Tarea 3. Conexión del formulario Gestión de tarifario con la base de datos.....	93
Tabla 5.34: Historia 17 - Tarea 1. Diseño de interfaz de Gestión de Proceso de Admisión	95
Tabla 5.35: Historia 17 - Tarea 2. Desarrollo de las funciones en base de datos para Proceso de Admisión.....	95
Tabla 5.36: Historia 17 - Tarea 3. Conexión del formulario Gestión de Proceso de Admisión con la base de datos	96
Tabla 5.37: Historia 1 - Tarea 1. Diseño de interfaz del sistema de inscripción (examen de admisión)	97
Tabla 5.38: Historia 1 - Tarea 2. Diseño de interfaz del sistema de inscripción (centro pre)	97
Tabla 5.39: Historia 1 - Tarea 3. Desarrollo de las funciones en base de datos para Postulante.....	98
Tabla 5.40: Historia 1 - Tarea 4. Conexión del formulario de inscripción (examen de admisión) con la base de datos	98

Tabla 5.41: Historia 1 - Tarea 5. Conexión del formulario de inscripción (centro pre) con la base de datos	99
Tabla 5.42: Historia 2 - Tarea 1. Diseño de interfaz de Registro de Fotos	104
Tabla 5.43: Historia 2 - Tarea 2. Instalación y conexión de la cámara fotográfica con el sistema	104
Tabla 5.44: Historia 2 - Tarea 3. Guardar fotos en el sistema	105
Tabla 5.45: Interfaz de registrar foto cuando no existe un proceso de admisión activo	107
Tabla 5.46: Historia 3 - Tarea 1. Diseño del applet para Registro de Huellas	107
Tabla 5.47: Historia 2 - Tarea 2. Instalación y conexión del huellero con el sistema	108
Tabla 5.48: Historia 2 - Tarea 3. Guardar imágenes de huellas en el sistema	109
Tabla 5.49: Historia 4 - Tarea 1. Diseño de interfaz de Verificar datos de postulante	111
Tabla 5.50: Historia 4 - Tarea 2. Desarrollo de las funciones en base de datos para Postulante	111
Tabla 5.51: Historia 4 - Tarea 3. Conexión del formulario Verificar Datos de Postulante con la base de datos	112
Tabla 5.52: Historia 5 - Tarea 1. Diseño del carnet	115
Tabla 5.53: Historia 5 - Tarea 2. Desarrollar la función para la impresión de carnet	116
Tabla 5.54: Historia 11 - Tarea 1. Diseño de interfaz de Iniciar Sesión	117
Tabla 5.55: Historia 11 - Tarea 2. Conexión del formulario Iniciar Sesión con la base de datos	117
Tabla 5.56: Historia 12 - Tarea 1. Diseñar interfaz de Perfil de Usuario	118
Tabla 5.57: Historia 12 - Tarea 2. Conexión del formulario Perfil de Usuario con la base de datos	119
Tabla 5.58: Historia 13 - Tarea 1. Diseño de la interfaz Asignar Permisos	120
Tabla 5.59: Historia 13 - Tarea 2. Conexión del formulario Asignar Permisos con la base de datos	121
Tabla 5.60: Historia 6 - Tarea 1. Diseño de interfaz de Cambio de modalidad (De extraordinario a ordinario)	122
Tabla 5.61: Historia 6 - Tarea 2. Conexión del formulario Cambio de modalidad (De extraordinario a ordinario) con la base de datos	123
Tabla 5.62: Historia 7 - Tarea 1. Diseño de Interfaz de Cambio de modalidad (De CPU a Ingresante CPU)	124
Tabla 5.63: Historia 7 - Tarea 2. Conexión del formulario Cambio de modalidad (De CPU a Ingresante CPU) con la base de datos	124
Tabla 5.64: Historia 8 - Tarea 1. Diseño de interfaz Lista de postulantes y conexión a base de datos	125
Tabla 5.65: Historia 8 - Tarea 2. Mostrar datos en el formulario	126
Tabla 5.66: Reporte - Lista de postulantes	126
Tabla 5.67: Historia 9 - Tarea 1. Diseño de interfaz de Postulantes por modalidad y escuela y conexión a base de datos	127

Tabla 5.68: Historia 9 - Tarea 2. Mostrar datos en el formulario	127
Tabla 6.1: Prueba de Aceptación 1. Registro de Postulante con datos erróneos.	133
Tabla 6.2: Prueba de Aceptación 2. Registro de Postulante	135
Tabla 6.3: Prueba de Aceptación 3. Registrar foto de postulante	135
Tabla 6.4: Prueba de Aceptación 4. Registrar huella de postulante	136
Tabla 6.5: Prueba de Aceptación 5. Verificar datos de postulante	137
Tabla 6.6: Prueba de Aceptación 6. Modificar datos incorrectos del postulante	138
Tabla 6.7: Prueba de Aceptación 7. Imprimir carnet de postulante	139
Tabla 6.8: Prueba de Aceptación 8. Imprimir carnet de postulante	140
Tabla 6.9: Prueba de Aceptación 9. Cambiar modalidad de postulante a ordinario.....	141
Tabla 6.10: Prueba de Aceptación 10. Cambiar modalidad de postulante a ingresante CPU	142
Tabla 6.11: Prueba de Aceptación 11. Listado total de postulantes.....	143
Tabla 6.12: Prueba de Aceptación 12. Listar postulantes por modalidad y escuela	144
Tabla 6.13: Prueba de Aceptación 13. Registrar Personal.....	145
Tabla 6.14: Prueba de Aceptación 14. Modificar Personal.....	146
Tabla 6.15: Prueba de Aceptación 15. Resetear contraseña de personal	147
Tabla 6.16: Prueba de Aceptación 16. Registrar rol.....	148
Tabla 6.17: Prueba de Aceptación 17. Modificar rol.....	149
Tabla 6.18: Prueba de Aceptación 18. Eliminar rol	150
Tabla 6.19: Prueba de Aceptación 19. Ingreso al sistema con usuario o contraseña incorrectos.....	151
Tabla 6.20: Prueba de Aceptación 20. Ingreso al sistema con usuario y contraseña correctos	152
Tabla 6.21: Prueba de Aceptación 21. Modificar contraseña de usuario	153
Tabla 6.22: Prueba de Aceptación 22. Registrar permisos de usuario.....	154
Tabla 6.23: Prueba de Aceptación 23. Registrar modalidad	155
Tabla 6.24: Prueba de Aceptación 24. Modificar modalidad	156
Tabla 6.25: Prueba de Aceptación 25. Eliminar modalidad.....	157
Tabla 6.26: Prueba de Aceptación 26. Registrar tipo de modalidad.....	158
Tabla 6.27: Prueba de Aceptación 27. Modificar tipo de modalidad.....	159
Tabla 6.28: Prueba de Aceptación 28. Eliminar tipo de modalidad	160
Tabla 6.29: Prueba de Aceptación 29. Registrar escuela	161
Tabla 6.30: Prueba de Aceptación 30. Gestión de escuela.....	162
Tabla 6.31: Prueba de Aceptación 31. Eliminar escuela	163
Tabla 6.32: Prueba de Aceptación 32. Registrar grupo.....	164
Tabla 6.33: Prueba de Aceptación 33. Modificar grupo.....	165
Tabla 6.34: Prueba de Aceptación 34. Eliminar grupo	166
Tabla 6.35: prueba de Aceptación 35. Registrar tarifario	167
Tabla 6.36: Prueba de Aceptación 36. Modificar tarifario	168

Tabla 6.37: Prueba de Aceptación 37. Eliminar tarifario	169
Tabla 6.38: Prueba de Aceptación 38. Registrar proceso de admisión	170
Tabla 6.39: Prueba de Aceptación 39. Modificar proceso de admisión	171
Tabla 6.40: Prueba de Aceptación 40. Eliminar proceso de admisión	172
Tabla 6.41: Personal administrativo que realizó las pruebas de los sistemas	173

ÍNDICE DE FIGURAS

Figura 2.2.1: Organigrama de la Institución.....	13
Figura 1.1: Prácticas técnicas de XP	38
Figura 1.2: Arquitectura Modelo Vista Controlador.....	46
Figura 1.3: Ciclo de vida de un Java Servlet	49
Figura 1.4: Diferencias entre Java y JavaScript	51
Figura 1.1: Cronograma del Proyecto	61
Figura 3.1: Diagrama entidad relación	82
Figura 3.2: Interfaz del sistema administrativo	83
Figura 3.3: Interfaz de Gestión de personal	85
Figura 3.4: Listado de personal	86
Figura 3.5: Modal para agregar un rol.....	86
Figura 3.6: Interfaz de Gestión de modalidad.....	88
Figura 3.7: Modal para agregar los tipo de modalidad	89
Figura 3.8: Interfaz de Gestión de escuela.....	91
Figura 3.9: Modal para registrar grupos	91
Figura 3.10: Interfaz de Gestión de tarifario	94
Figura 3.11: Interfaz de Gestión de proceso de admisión	96
Figura 3.12: Interfaz de sistema de inscripción de postulantes	100
Figura 3.13: Datos de validación para iniciar registro de postulante	100
Figura 3.14: Interfaz de inscripción - Información básica de postulante	101
Figura 3.15: Interfaz de inscripción - Información académica	102
Figura 3.16: Interfaz de inscripción - Verificar datos	103
Figura 3.17: Interfaz de Registrar foto.....	106
Figura 3.18: Mensaje de seguridad cuando se accede al applet.....	109
Figura 3.19: Applet para registrar huellas	110
Figura 3.20: Interfaz de registrar huella cuando no existe un proceso de admisión activo	110
Figura 3.21: Interfaz de verificar datos de postulante.....	113
Figura 3.22: Modal para modificar datos de postulante.....	114
Figura 3.23: Interfaz de Verificar datos de postulante cuando no existe un proceso de admisión activo	115
Figura 3.24: Carnet de postulante	116
Figura 3.25: Interfaz de sistema administrativo - Iniciar sesión	118

Figura 3.26: Perfil de usuario	119
Figura 3.27: Modal para modificar contraseña de usuario	120
Figura 3.28: Información que aparece en la cabecera de cada formulario.....	120
Figura 3.29: Interfaz de Asignar permisos.....	121
Figura 3.30: Interfaz de acceso denegado	122
Figura 3.31: Interfaz de cambio de modalidad (de extraordinario a ordinario)	123
Figura 3.32: Interfaz de Cambio de modalidad (De CPU a Ingresante CPU).	125
Figura 3.33: Reporte - Total de postulantes por modalidad	128
Figura 3.34: Reporte - Total de postulantes por escuela.....	128
Figura 3.35: Reporte - Total de postulantes por modalidad y escuela	129
Figura 3.1: Manual de usuario.....	132
Figura 1.1: Iniciar sesión	175
Figura 1.2: Modificar contraseña de usuario	175
Figura 1.3: Asignar permisos.....	176
Figura 1.4: Acceso denegado.....	176
Figura 1.5: Menú para el usuario con rol de Fotos	177
Figura 2.1: Sistema de inscripción de postulantes	178
Figura 2.2: Ingreso a la inscripción de postulantes	178
Figura 2.3: Paso1 - Campos Validados I.....	179
Figura 2.4: Paso 1 - Campos Validados II – Mensaje de error	179
Figura 2.5: Campos Validados III	180
Figura 2.6: Paso 3 - Graduados y titulados - Cambio de Universidad.....	181
Figura 2.7: Paso 3 - Campos Validados.....	181
Figura A.1: Resolución N° 218-2016-C.O/UNACH - Pág. 1	189
Figura A.2: Resolución N° 218-2016-C.O/UNACH Pág. 2	190
Figura A.3: Certificado de aprobación del sistema web otorgado por la Universidad Nacional Autónoma de Chota	191
Figura A.4: Informe sobre el sistema para el proceso de inscripción de postulantes enviado por la Unidad de Soporte Informático a la presidenta de la Comisión Organizadora – UNACH. Pág. 1.	192
Figura A.5: Informe sobre el sistema para el proceso de inscripción de postulantes enviado por la Unidad de Soporte Informático a la presidenta de la Comisión Organizadora – UNACH. Pág. 2.	193

INTRODUCCIÓN

Hoy en día existe la necesidad de optimizar las actividades manuales que se desarrollan diariamente en las empresas, casa de estudios, entre otras y que ha venido incrementando a la par con el despliegue de nuevas tecnologías para brindar un mejor control de forma rápida y sencilla. La Universidad Nacional Autónoma de Chota tenía la necesidad de desarrollar un sistema que le permitiera el registro de postulantes, por lo cual se planteó elaborar un proyecto de investigación que lleva como título SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCIÓN DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA – 2015, cabe mencionar que con el desarrollo de sistema web la universidad estaría a la vanguardia con otras casas de estudio que se encuentran frecuentemente innovando.

La presente Tesis consta de 8 capítulos, los cuales se encuentran estructurados de la siguiente manera:

- Aspecto Informativo: en esta sección presentamos toda la información básica de los integrantes.
- Base Legal y Aspecto de la Organización: en esta parte detallamos como nace la idea y como fue creada la casa de estudios con sus principios, misión y visión corporativa y quienes la integran.
- Aspecto de la investigación: en esta parte exponemos toda la información y las herramientas de apoyo como el lenguaje de desarrollo, las metodologías de desarrollo, la herramienta de modelado, los servidores web que hicieron posible el desarrollo del sistema, así como el problema científico y sus objetivos.
- Aspecto Administrativo: en esta sección costeamos el proyecto.

- Desarrollo del sistema: en este capítulo describimos las fases del proyecto, desde la recopilación de información (historias de usuario), diseño y programación, hasta la fase de producción.
- Pruebas de uso del sistema: en este capítulo comprobamos el correcto funcionamiento del sistema y verificamos si presenta errores o fallas para luego realizar sus correcciones.
- Seguridad: mostramos las medidas de seguridad que hemos implementado en el sistema para mantener la confiabilidad de la información y evitar que personas extrañas manipulen los datos.
- Mantenimiento y actualización: Existe un encargado que es el administrador de la base de datos y en este capítulo describimos las actividades que debe realizar para preservar la información y mantener el correcto funcionamiento del sistema.

CAPITULO I

ASPECTO INFORMATIVO



1. ASPECTO INFORMATIVO

1.1. Título

Sistema bajo entorno web para el proceso de inscripción de postulantes de la Universidad Nacional Autónoma de Chota - 2015.

1.2. Autor(es)

Nombres y Apellidos: Pérez Bonilla Luis Edgardo

Código Universitario: 075552-A

Dirección: Res. Augusto B. Leguía Edif. XXI Dpto. 203 - Chiclayo

Correo Electrónico: lperezb@unprg.edu.pe

Teléfono Fijo: 074 - 266198

Celular: 979880146

Nombres y Apellidos: Quintos Estrada Katherine Vanessa

Código Universitario: 089037 G

Dirección: Prolongación Huáscar 128 - Lambayeque

Correo Electrónico: kquintos@unprg.edu.pe

Teléfono Fijo: 074 - 281468

Celular: 979494145

1.3. Centro o Instituto de Investigación

Centro de Investigación de la Facultad de Ciencias Físicas y Matemáticas.

1.4. Área de investigación

Área: Computación e Informática

Sub-Área: Aplicada

Línea: Desarrollo de Software

Programas: Informática aplicada a la calidad total

1.5. Lugar de ejecución

Departamento de Cajamarca, Provincia de Chota, Ciudad de Chota,
Universidad Nacional Autónoma de Chota.

1.6. Duración estimada

10 meses

1.7. Fecha de Inicio

30 de julio 2015

1.8. Fecha de Finalización

24 de mayo 2016

1.9. Beneficiarios del Proyecto

Los administrativos que laboran en los procesos de Admisión en la
Oficina Central de Informática – OCI, de la Universidad Nacional
Autónoma de Chota.

CAPITULO II

BASE LEGAL Y

ASPECTO DE LA ORGANIZACIÓN



2. BASE LEGAL Y ASPECTO DE LA ORGANIZACIÓN

2.1. Base Legal

El Presente Reglamento norma el proceso de Admisión a la Universidad Nacional Autónoma de Chota, teniendo como base legal los dispositivos que a continuación se indica:

1.1. Ley Universitaria N° 30220.

1.2. Ley de CONAFU N° 26439,29780 y su Estatuto.

Que, por Ley N° 26439, se crea el Consejo Nacional para la Autorización de Funcionamiento de Universidades - CONAFU, como órgano autónomo de la Asamblea Nacional de Rectores, teniendo como una de sus atribuciones: Evaluar los proyectos y solicitudes de autorización de funcionamiento de las nuevas universidades a nivel nacional, emitiendo resoluciones autorizando o denegando el funcionamiento provisional, previa verificación de los requisitos y condiciones establecidos; en concordancia con el artículo 3º inciso a) del Estatuto del CONAFU, aprobado por Resolución N° 189-2006-CONAFU de fecha 13 de julio de 2006.

Ley que modifica la Ley 26439, crean el Consejo Nacional para la Autorización de Funcionamiento de Universidades (CONAFU) LEY N° 29780

La Comisión Permanente del Congreso de la República ha dado la Ley siguiente:

Artículo único. Incorporación del literal i) al artículo 2 de la Ley 26439, Crean el Consejo Nacional para la Autorización de Funcionamiento de Universidades (Conafu) Incorpórase el literal i) al artículo 2 de la Ley 26439, Crean el Consejo Nacional para la Autorización de Funcionamiento de Universidades (Conafu), conforme al texto siguiente:

"Artículo 2. Son atribuciones del CONAFU:

- a) Evaluar los proyectos y solicitudes de autorización de funcionamiento de las nuevas universidades a nivel nacional, y emitir resoluciones autorizando o denegando el funcionamiento provisional, previa verificación del cumplimiento efectivo de los requisitos y condiciones establecidos.
- b) Autorizar la fusión de universidades, previa evaluación del proyecto, así como la supresión de las mismas.
- c) Evaluar en forma permanente y durante el tiempo que estime conveniente el funcionamiento de las universidades hasta autorizar o denegar su funcionamiento definitivo. La autorización de funcionamiento definitivo no puede ser concedida antes de transcurridos cinco años, contados a partir de la fecha de la autorización provisional de funcionamiento.
- d) Autorizar, denegar, ampliar o suprimir facultades, carreras o escuelas, así como limitar el número de vacantes en las universidades con funcionamiento provisional.
- e) Reconocer a las comisiones organizadoras a propuesta de los promotores.
- f) Elaborar sus propios estatutos.
- g) Elaborar la reglamentación que señale los requisitos, procedimientos y plazos para la autorización provisional o definitiva de funcionamiento y para la evaluación de las universidades con autorización provisional.
- h) Autorizar o denegar el cambio de denominación de las universidades a solicitud de su máximo órgano de gobierno,

cualquiera que haya sido el instrumento legal o la fecha de su creación.

- 1.3. Ley General de Educación N° 28044.
- 1.4. Ley N° 29531, Ley de creación de la Universidad Nacional Autónoma de Chota (UNACH).
- 1.5. Ley del Procedimiento Administrativo General N° 27444.
- 1.6. Resolución N° 208-2013-CONAFU, que autoriza el funcionamiento provisional de la UNACH.
- 1.7. Estatuto de la UNACH.

A través de la Resolución N° 218-2016-C.O./UNACH de fecha 17 de mayo de 2016, la Comisión Organizadora en pleno resolvió APROBAR el Estatuto de la Universidad Nacional Autónoma de Chota, el mismo que consta de quince (15) capítulos, doscientos ochenta (280) artículos, ocho (8) disposiciones finales y transitorias y dos (2) disposiciones complementarias, documento que otorga una base legal al cumplimiento de funciones en todos los ámbitos universitarios y que permitirá un desempeño concreto frente a las múltiples situaciones que devendrán en adelante.

La Comisión Organizadora manifestó que la aprobación de este documento constituye uno de los pilares fundamentales y constituye un requisito clave para concretar el proceso de licenciamiento, el Estatuto es el protagonista de toda vida universitaria y ahora la UNACH ya cuenta con él. Se ha dispuesto poner de conocimiento el presente documento a las Oficinas correspondientes de la Universidad, al Ministerio de

Educación (MINEDU) y a la Superintendencia Nacional de Educación Superior Universitaria (SUNEDU).

1.8. Reglamento General de la UNACH.

2.2. Aspecto de la Organización

2.2.1. Nombre

Universidad Nacional Autónoma de Chota.

2.2.2. Ubicación Geográfica

La Universidad Nacional Autónoma de Chota se encuentra ubicada en colpa matara – Chota – Cajamarca.

2.2.3. Visión

Universidad que cumple altos estándares de calidad, reconocida en el ámbito nacional e internacional, generadora de conocimientos que promueve la investigación científica y tecnológica, líder en la formación académica y humanística al servicio de la sociedad, contribuyendo al desarrollo sostenible de la región y el país.

2.2.4. Misión

Somos una universidad con compromiso social que desarrolla investigación científica y tecnológica, formadora de profesionales competitivos y emprendedores, con sólidos valores que contribuyen al desarrollo de la región y el país.

2.2.5. Objetivos

Formar profesionales con una concepción humanista, con capacidades críticas e innovadoras.

2.2.6. Funciones

Las funciones de la Universidad Nacional Autónoma de Chota son:

- Formación Profesional.
- Investigación.

2.2.7. Principios

La Universidad Nacional Autónoma de Chota, se basa en los siguientes principios:

- Integración hacia el crecimiento y desarrollo de la Región como base esencial en la generación de conocimientos. Su aplicación y difusión en los diversos campos de la producción y la prestación de servicios. La innovación del conocimiento, constituye la base de desarrollo.
- Aporte al desarrollo económico sostenible de la Región y del país en las actuales condiciones de cambio, a través de la dotación de profesionales competitivos en áreas específicas. La solución prioritaria de los problemas de la provincia y el aprovechamiento potencial de sus recursos naturales regionales para una mejor redistribución social y ecológica.
- La eficiencia, creatividad y eficacia en la productividad. La calidad del servicio institucional deberá ser su paradigma y el desempeño de sus integrantes, su inspiración. El turismo hacia toda la Región, más aún a la Provincia de Chota, que con su historia y tradición requieren de una decidida promoción.
- Calidad y competitividad en la oferta de sus servicios así como en la dotación de los medios técnicos y científicos de avanzada, lo cual conlleve a garantizar

la solución de los problemas más apremiantes de la Región y de la Provincia de Chota, propiciando la diversificación y una mejora de la calidad de vida.

- Estrategias, consolidación institucional, a través de la aplicación del Planeamiento Estratégico y sus principios, bondades y resultados que consoliden los criterios de organización y desarrollo.
- Responsabilidad y adecuación de la universidad al contexto social, fomentando la investigación, acordemos avances científicos, tecnológicos de cada especialidad, donde sus resultados repercutan en los procesos de producción y desarrollo provincial y Regional.
- Desarrollo de una conciencia crítica de la juventud, orientada a valorar y conservar la identidad regional y nacional; el aprovechamiento de los recursos naturales, culturales con plena capacidad creadora y mística, desterrando el conformismo, dando pase al dinamismo e integración, con criterios de sostenibilidad regional.
- El Concurso de Admisión es el proceso que permite a la Universidad seleccionar a los estudiantes que deseen iniciar o continuar sus estudios en ésta; evaluando los intereses vocacionales, aptitudes y rasgos de personalidad para acceder a una de las vacantes aprobadas mediante resolución del CONAFU.

2.2.8. Organigrama

ORGANIGRAMA

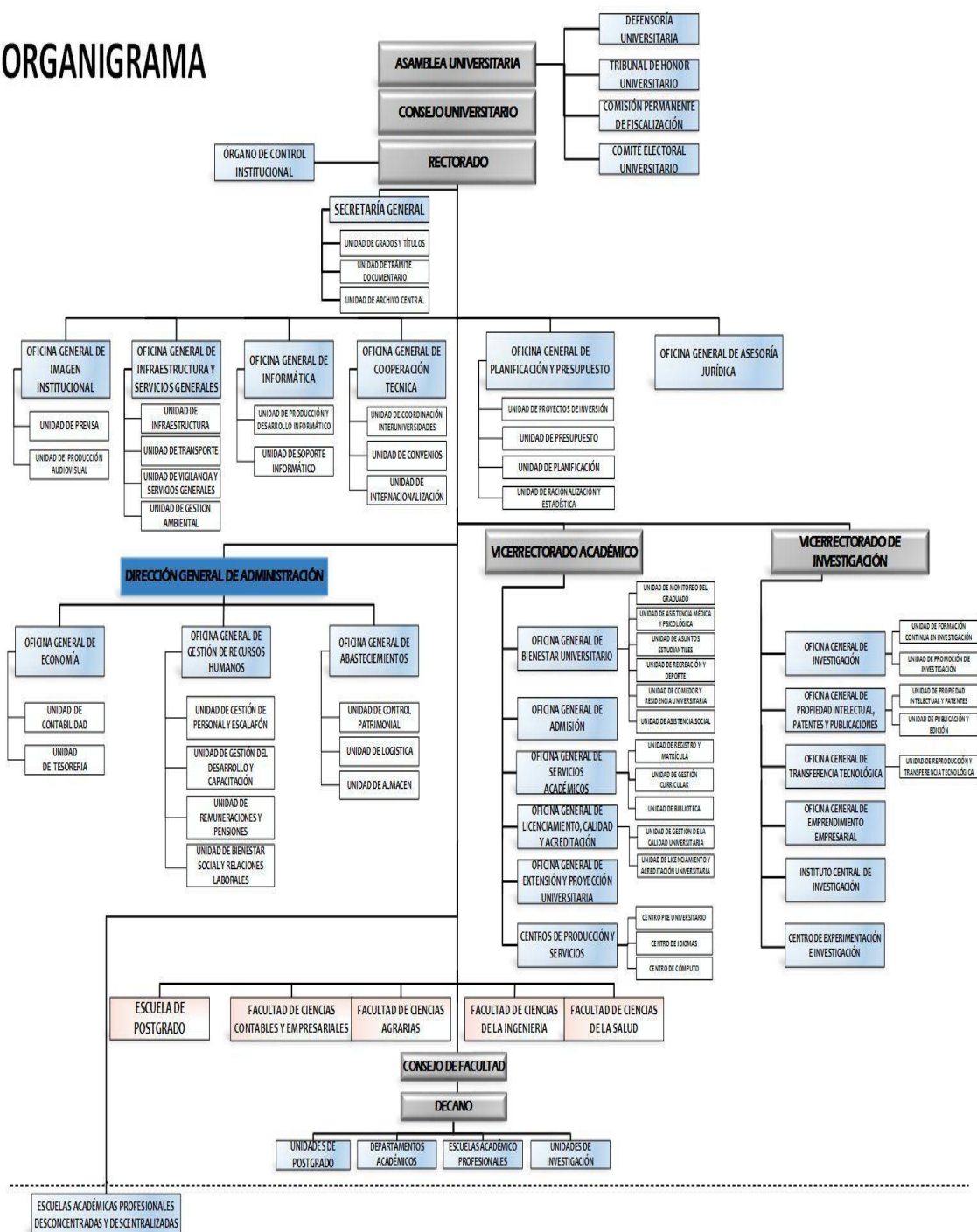


Figura 2.2.1: Organigrama de la Institución
Fuente: Universidad Nacional Autónoma de Chota

2.2.9. Personal

2.2.9.1. Presidencia

La Dra. Florencia Adelina Arteaga Torres nació el 23 de mayo de 1953, realizó sus estudios de pregrado en la Universidad Nacional de Trujillo obteniendo su título de Licenciada en Estadística y una segunda especialidad en Psicopedagogía, sus estudios de maestría en Ciencias de la Educación con mención en Docencia Universitaria e Investigación Educativa, posteriormente obtuvo su doctorado en Gestión y Ciencias de la Educación. Actualmente asume el cargo de Presidenta de la Comisión Organizadora de la Universidad Nacional Autónoma de Chota.

2.2.9.2. Vicepresidencia Académica

El Dr. Edin Edgardo Alva Plasencia nació el 19 de octubre de 1960 en la provincia de Contumazá – Cajamarca, sus estudios universitarios los realizó en la Universidad de Cajamarca siguiendo la carrera de Ing. Agrónomo, estudió su post grado en la Universidad Agraria de la Molina donde obtuvo el grado de Magister en Sciente con especialidad en suelos, posteriormente realizó su doctorado en la Universidad de Cajamarca con mención en Gestión Ambiental y Recursos Naturales. Actualmente asume el cargo de Vicepresidente Académico de la Comisión Organizadora de la Universidad Nacional Autónoma de Chota.

2.2.9.3. Vicepresidencia de Investigación

EL Dr. Edgar Carlos Quispe Peña nació en Cerro de Pasco y estudio pregrado en la Universidad Nacional del Centro del Perú obteniendo el título de Ing. Zootecnista en 1990. En la Universidad Nacional Agraria la Molina obtiene el Grado de M.sc en Producción Animal, en la actualidad asume el cargo de Vicepresidente de Investigación de la Comisión Organizadora de la Universidad Nacional Autónoma de Chota.

CAPITULO III

ASPECTO DE LA INVESTIGACIÓN



3. ASPECTO DE LA INVESTIGACIÓN

3.1. Marco Lógico

3.1.1. Planteamiento del Problema Científico.

La Universidad Nacional Autónoma de Chota es una institución pública destinada a impartir educación superior, promover el estudio de la realidad nacional, la investigación científica, la difusión del saber y la cultura, la extensión y la proyección social para contribuir al desarrollo local, regional y nacional.

Fue creada por Ley N° 29531 del 11 de mayo de 2010, es persona jurídica de derecho público sin fines de lucro, con autonomía académica, económica, normativa y administrativa conforme a ley. Se identifica con su sigla UNACH para todos los efectos legales y oficiales propios de la institución.

La historia de la Universidad en Chota se remonta a hace 20 años. Un grupo de notables chotanos, con mucha voluntad y un local disponible, lograron que la Universidad Nacional de Cajamarca llegara a Chota con el programa de Enfermería. Sin embargo, la UNC prestó poca atención a las constantes demandas de la sociedad chotana, de modo que los estudiantes se sentían desplazados y poco atendidos por la institución. Por este motivo, y con la colaboración de las autoridades locales, se inició un proyecto ambicioso en beneficio de la comunidad universitaria. Se recibió aportes de diversa índole: donación de local, terrenos, materiales de construcción, etc. La sede obtuvo un local en la Plaza de Armas, un pabellón en Colpa Matara y otras propiedades.

Hasta hace 4 años, aproximadamente, tener una Universidad con Autonomía era un tema de conversación como otros tantos: se consideraba que era derecho del pueblo pero no llegaba. El sueño que parecía lejano empezó a tomar forma con el proyecto de la Universidad Tecnológica de Chota.

Desde hace un año la universidad desarrolla procesos de admisión semestrales, en cada proceso registra un promedio de 1,000 postulantes repartidos en sus cinco especialidades, de la misma forma se lleva el proceso de inscripción del centro Pre Universitario de dicha casa de estudios, este proceso actualmente no es llevado de manera óptima, generando demasiado trabajo para manipular la información que contiene los requisitos y las fichas de inscripción que corren el riesgo de extraviarse o alterar su contenido.

El proceso de inscripción se lleva a cabo de forma manual, donde se verifica la totalidad de los documentos, se recolectan los datos a través de una ficha desarrollada en Microsoft Excel con el fin de crear expedientes de cada postulante y se anexan los documentos exigidos: partida de nacimiento, certificado de estudios, copia de DNI.

Entre las dificultades que presenta el proceso actual tenemos: El trabajo se torna demasiado pesado durante el proceso, el carnet no es entregado al instante como prueba que el postulante ha sido inscrito, algunas de las actividades hechas por la institución se atrasan generando malestar entre las personas que laboran en la institución, la información que se procesa es muy lenta; todas estas dificultades repercuten en las diferentes áreas de la institución, esto ocasiona ineficiencia en el proceso de inscripción y el personal trabaja demasiadas horas extras. Por estas razones es necesario un cambio estructural, a través de un sistema de web que evite errores y contratiempos, controle la duplicación y pérdida de documentos y que agilice el proceso de inscripción de tal manera que sea confiable, seguro y de fácil manejo para el usuario.

3.1.2. Revisión Bibliográfica.

3.1.2.1. Antecedentes

Álvarez A. (2007). **“Desarrollo de un Sistema de Inscripciones utilizando AJAX”** concluye que las nuevas tecnologías web brindan ventajas tangibles sobre las aplicaciones web tradicionales, además de poseer características que se acoplan perfectamente a los requerimientos de los sistemas de inscripciones, por ello se obtiene un mejor rendimiento seguridad y accesibilidad, reduciendo el tráfico de datos.

Palacios D., Chiluisa H. (2011). **“Desarrollo de un sistema de inscripción y matriculación para el Centro Regional de Formación Industrial del Norte”** llegaron a la conclusión que el sistema mejoró los procesos de inscripción y matrícula, brindando agilidad, organización y un mayor control de la información, además, gracias a las herramientas utilizadas, y al sistema basado en web, permitió una rápida y confiable implantación del sistema.

Norabuena A. (2011). En su tesis **“Análisis, diseño e implementación de un Sistema de Información para la Gestión Académica de un Instituto Superior Tecnológico”** sugiere como una alternativa de solución, para incrementar la calidad del servicio que brindan los centros de educación a los alumnos, el uso de tecnologías de la información porque sirven de soporte a las actividades que se realizan en estas instituciones y aseguran el manejo eficiente de la información y su disponibilidad en el momento oportuno.

Arcaya C. (2012). **“Sistema de Información Cliente/Servidor con Tecnología Web para los procesos de matrículas y trámites de certificación de la escuela**

Nacional de Estadística e Informática del INEI –Tacna – 2011” concluye que el sistema de información cliente/servidor desarrollado con tecnología Web, si mejora los procesos de matrículas para el Trámite de certificación en la Escuela nacional de Estadísticas e Informática del INEI - TACNA, disminuyendo los tiempos y previniendo los errores que causa la redundancia de datos .Además este sistema cumplió con la mayor parte de la funcionalidad requerida y establecida por los usuarios.

Vera H. (2009). **“Desarrollo de una Aplicación web para la gestión de matrículas y notas en la I.E. N°11015 Comandante ‘Elías Aguirre’ - Chiclayo”** afirma que el uso de Ajax en los sistemas permite mejorar completamente la interacción del usuario con la aplicación web, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en segundo plano. Además la aplicación web cumple con los objetivos trazados porque permite un acceso más fácil y rápido al registro y publicación de notas de los alumnos y mayor seguridad porque los padres de familia y apoderados ingresan a la web con una clave que ellos pueden cambiar.

3.1.2.2. Base Teórica

El sistema web será diseñado siguiendo el patrón de arquitectura Modelo Vista Controlador (MVC) , desarrollado con el lenguaje de programación Java orientada a web y el gestor de base de datos PostgreSQL.

Se construirá un applet Java, que será la aplicación encargada de la captura de huellas de los postulantes.

3.1.2.2.1. Sistemas Web

Las aplicaciones Web para Internet e Intranet presentan una serie de ventajas y beneficios con respecto al software de escritorio, con lo cual se logra aprovechar y acoplar los recursos de una empresa de una forma mucho más práctica que el software tradicional.

Entre los beneficios que las aplicaciones desarrolladas para la web tienen respecto a las aplicaciones de escritorio se encuentran:

- El trabajo a distancia se realiza con mayor facilidad.
- Para trabajar en la aplicación Web solo se necesita un computador con un buen navegador Web y conexión a internet.
- Las aplicaciones Web no necesitan conocimientos previos de informática. Con una aplicación Web tendrá total disponibilidad en cuanto a hora y lugar, podrá trabajar en ella en cualquier momento y en cualquier lugar del mundo siempre que tenga conexión a internet.
- Las aplicaciones Web le permiten centralizar todas las áreas de trabajo.

Ventajas de las aplicaciones Web:

- Compatibilidad multiplataforma. Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables.
- Actualización. Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento.
- Inmediatez de acceso. Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online a trabajar sin importar cuál es su configuración o su hardware.

- Menos requerimientos de memoria. Las aplicaciones basadas en web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente.
- Menos Bugs. Las aplicaciones basadas en web deberán ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.
- Múltiples usuarios concurrentes. Las aplicaciones basadas en web pueden realmente ser utilizada por múltiples usuarios al mismo tiempo. (INTERNET YA – Soluciones Web, s.f.)

3.1.2.2.2. Lenguaje de Desarrollo

JAVA

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes. (Java, s.f.)

Un objetivo clave de Java es poder escribir programas que se ejecuten en una gran variedad de sistemas computacionales y dispositivos controlados por computadora. A esto se le conoce algunas veces como “escribir una vez, ejecutar en cualquier parte”.

La popularidad del servicio Web se intensificó en 1993; en ese entonces Sun vio el potencial de usar Java para agregar contenido dinámico, como interactividad y animaciones, a las páginas Web. Java generó la atención de la comunidad de negocios debido al fenomenal interés en Web. En la actualidad, Java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de los servidores Web (las computadoras que proporcionan el contenido que vemos en nuestros exploradores Web), para proporcionar aplicaciones para los dispositivos de uso doméstico (como teléfonos celulares, teléfonos inteligentes, receptores de televisión por Internet y mucho más) y para muchos otros propósitos. (Deitel & Deitel, 2012, pág. 18)

Ventajas y Desventajas

❖ Ventajas:

- **Simplicidad:** Proporciona una interfaz muy fácil para los usuarios como para los desarrolladores.
- **Portable:** Java es independiente de la plataforma es decir que proporciona la facilidad de escribir una vez el código y ser ejecutada en cualquier browsers o navegador web que sea compatible con java o tenga la máquina virtual de java.
- **Robusto:** Java crea aplicaciones altamente fiables, por ello proporciona numerosas comprobaciones en ejecución y en tiempo real.
- **Interpretado:** Java corre en máquina virtual.
Se puede acceder a la base de datos mediante JDBC.

❖ **Desventajas:**

- **Velocidad:** los programas desarrollados en java no suelen ser muy rápidos.
- Algunas herramientas tienen costo adicional.

3.1.2.2.3. Gestor de Base de Datos

POSTGRESQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

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. (Martinez, 2010)

Ventajas y Desventajas

❖ **Ventajas:**

- Fácil de Administrar.
- Ideal para tecnologías web.
- Multiplataforma (incluyendo Linux, Windows, Unix, Solaris y MacOS X).
- Recopilación de Datos.
- Aun no se presentan caídas de base de datos.
- Gran capacidad de almacenamiento.

❖ **Desventajas:**

- Reducida cantidad de tipo de datos.
- Consume más recursos que MYSQL, por lo que necesita más características de hardware para instalarlo.

3.1.2.2.4. Metodología de Modelado

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta sistematización nos indica cómo dividiremos un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, nos ayuda a definir entradas y salidas para cada una de las etapas y, sobretodo, normaliza el modo en que administraremos el proyecto. Entonces, una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado. (Cantone, 2006, pág. 16)

METODOLOGÍA ÁGIL

El término de metodologías Ágiles se hizo popular cuando diecisiete especialistas en desarrollo de software, presentaron los métodos Extreme Programming (XP), Scrum, Feature Driven Development FDD, entre otros, establecieron principios comunes compartidos por todos. El resultado fue la creación de la Alianza Ágil (Agile Alliance) y el establecimiento del Manifiesto Ágil o Agile Manifiesto, en el año de 2004. Las metodologías ágiles varían en sus prácticas y en sus fases, sin embargo, comparten algunas características, tales como: desarrollo iterativo e incremental, comunicación y reducción de productos intermediarios y de la comunicación intensiva. (Ramos, Noriega, Laínez, & Durango, 2015, pág. 212)

El manifiesto ágil expone 4 valores sobre estas metodologías:

1. **Individuos e interacciones** sobre procesos y herramientas.
2. **Software funcionando** sobre documentación extensiva.
3. **Colaboración con el cliente** sobre negociación contractual.
4. **Respuesta ante el cambio** sobre seguir un plan.
(Manifiesto Ágil, 2001)

De estos valores parten 12 principios:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y

usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
 10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
 11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
 12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.
- (Manifiesto Ágil, 2001)

Los firmantes de los valores y principios de este Manifiesto son:

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas.

(Manifiesto Ágil, 2001)

Las principales metodologías ágiles

Entre las metodologías ágiles más usadas se encuentran:

- **SCRUM**. Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante

iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Schwaber, Sutherland , & Beedle, 2001)

- **KANBAN**. Básicamente, Kanban se apoya en una producción a demanda: se produce solo lo necesario, de manera tal que el ritmo de la demanda sea quien controle al ritmo de la producción (se necesitan N partes, entonces solo se producen N partes), limitando el trabajo en curso a una cantidad predefinida. (Bahit, 2012, pág. 156)
- **XP**: Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. (Raya, 2014)

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 3.1: Comparación entre metodologías ágiles y metodologías tradicionales

Fuente: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

PROGRAMACIÓN EXTREMA

eXtreme Programming (programación extrema) también llamado XP, es una metodología que tiene su origen en 1996, de la mano de Kent Beck, Ward Cunningham y Ron Jeffries.

A diferencia de Scrum, XP propone solo un conjunto de prácticas técnicas, que aplicadas de manera simultánea,

pretenden enfatizar los efectos positivos de *[sic]* en un proyecto de desarrollo de Software. (Bahit, 2012, pág. 70)

“XP” es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Hernández, 2014, pág. 11)

Base de la Programación Extrema

eXtreme Programming se apoya en cinco valores, los cuales enfatizan la esencia colaborativa del equipo. Estos valores son:

- **Comunicación**

En XP, todo es trabajado en equipo: desde el relevamiento y análisis hasta el código fuente desarrollado. Todo se conversa cara a cara, procurando hallar soluciones en conjunto a los problemas que puedan surgir.

- **Simplicidad**

Se pretende desarrollar solo lo necesario y no perder tiempo en detalles que no sean requeridos en el momento. En este aspecto, se asemeja a otra metodología ágil, denominada Kanban, en la cual, un

proceso “anterior” solo produce lo que el proceso posterior demanda.

- **Retroalimentación**

El objetivo de eXtreme Programming es entregar lo necesario al cliente, en el menor tiempo posible. A cambio, demanda al cliente, un feedback continuo - retroalimentación-, a fin de conocer sus requerimientos e implementar los cambios tan pronto como sea posible.

- **Respeto**

El equipo respeta la idoneidad del cliente como tal (solo este, es quien conoce el valor para el negocio) y el cliente, a la vez, respeta la idoneidad del equipo (confiando en ellos profesionalmente para definir y decidir el “como” se llevara a cabo el desarrollo de lo requerido).

- **Coraje**

Se dice que en XP un equipo debe tener el valor para decir la verdad sobre el avance del proyecto y las estimaciones del mismo, planificando el éxito en vez de buscar excusas sobre los errores. (Bahit, 2012, págs. 70-71)

Prácticas Técnicas del XP

eXtreme Programming propone 12 prácticas técnicas, simples, de fácil comprensión y que aplicadas en forma conjunta, garantizan un mejor resultado del proyecto. Estas doce prácticas, se describen a continuación.

- **Práctica #1: Cliente presente o in situ**

En XP se requiere que el cliente esté dispuesto a participar activamente del proyecto, contando con la

disponibilidad suficiente y necesaria, para interactuar con el equipo en todas las fases del proyecto. (...)

La comunicación cara a cara con el cliente es fundamental, ya que a partir de ésta, el equipo avanzará más rápidamente en el proyecto, puesto que:

- Evacuará todas sus dudas sobre el proyecto, con el cliente y en el momento que éstas surjan.
- Se establecerán las prioridades a desarrollar, en tiempo real, así como la resolución de conflictos.

Las personas asignadas por el cliente, deben ser:

- Conocedores expertos de lo que se pretende producir.
- En lo posible, se pretende que sean usuarios reales del software.
- Personas que cuenten con la información suficiente sobre la aplicación y los objetivos del negocio, a fin de poder actuar con autoridad en la toma de decisiones.

- **Práctica #2: Semana de 40 horas (40 hour week)**

eXtreme Programming asegura la calidad del equipo, considerando que este, no debe asumir responsabilidades que le demanden mayor esfuerzo del que humanamente se puede disponer.

Esto marca una diferencia radical con otras metodologías, sobre todo, con aquellas que siguen una línea tradicional donde las estimaciones deficientes obligan al equipo de desarrollo, a sobre-esfuerzo significativo. (...)

Un equipo “descansado”, sin esfuerzos desmedidos, logra un mejor resultado.

- **Práctica #3: Metáfora (Metaphor)**

A fin de evitar los problemas de comunicación que suelen surgir en la práctica, entre técnicos y usuarios, eXtreme Programming propone el uso de metáforas, intentando hallar un punto de referencia que permita representar un concepto técnico con una situación en común con la vida cotidiana [*sic*] y real.

Para que una metáfora cumpla este objetivo, es fundamental:

- Hacer un paralelismo entre una funcionalidad del sistema y la vida real.
- Si ese paralelismo (metáfora) no se entiende inmediatamente, se deberá buscar otra metáfora (una metáfora solo sirve cuando es comprendida de forma inmediata).

Muchas veces, puede ser necesario emplear más de una metáfora para explicar una misma característica del sistema. Incluso, en algunas oportunidades, pueden no encontrarse metáforas que describan con precisión toda la característica que se intenta explicar. Son momentos, en los que se requiere de un gran ingenio y creatividad. (...)

- **Práctica #4: Diseño simple (Simple Design)**

Esta práctica, deriva de un famoso principio técnico del desarrollo de software: KISS. Cuyas siglas derivan del inglés “Keep it simple, stupid!” - ¡Mantenlo simple, estúpido!-.

Dado que frecuentemente puede resultar “chocante” el término “stupid”, a lo largo de los años, se han ido produciendo variantes, como “silly” -tonto- e incluso, muchos de nosotros, a veces, preferimos reemplazarlo por un nombre propio, como si

estuviésemos dirigiendo la frase a una persona en particular: Keep it Simple, Susan!

Básicamente, el principio KISS, se trata de mantener un diseño sencillo, estandarizado, de fácil comprensión y refactorización. Puede resumirse en “hacer lo mínimo indispensable, tan legible como sea posible”.(...)

- **Práctica #5: Refactorización (Refactoring)**

La refactorización (o refactoring) es una técnica que consiste en modificar el código fuente de un software sin afectar a su comportamiento externo. (...)

Dado que eXtreme Programming propone “desarrollar lo mínimamente indispensable, en el desarrollo de cada nueva funcionalidad, puede (y debe) ser necesario, realizar un refactoring del código, a fin de lograr una mayor cohesión de este, impidiendo redundancias.

Por ello, la refactorización es la principal técnica propuesta por XP, para evolucionar el código de la aplicación con facilidad en cada iteración.

- **Práctica #6: Programación de a pares (Pair Programming)**

Otra de las practicas técnicas fundamentales de eXtreme Programming, es la programación de a pares.

Se estila (aunque no de forma constante) programar en parejas de dos desarrolladores, los cuales podrán ir intercambiando su rol, en las sucesivas Pair Programming.

¿En qué consiste la programación de a pares?
Sencillo.

Dos programadores, sentados frente a una misma computadora, cada uno cumpliendo un rol diferente. Las combinaciones y características de este rol, no solo son variables, sino que además, son inmensas, permitiendo la libertad de “ser originalmente creativos”.

Algunas de estas combinaciones (tal vez las más frecuentes), suelen ser:

- Uno escribe el código, mientras que otro lo va revisando;
- Un programador más avanzado, programa, mientras va explicando lo desarrollado a otro menos experto;
- Los dos programadores piensan en cómo resolver el código, y uno de ellos lo escribe.

De esa forma y según las necesidades de cada equipo, las funciones de cada programador, podrán variar ilimitadamente.

- **Práctica #7: Entregas cortas (Short Releases)**

Se busca hacer entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración. Esto conlleva a que el cliente pueda tener una mejor experiencia con el Software, ya que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras con mayor facilidad de implementación. (...)

- **Práctica #8: Testing**

En esta práctica, podemos encontrar tres tipos de test -pruebas- propuestos por eXtreme Programming: (...)

✓ **Test Unitarios**

Consisten en testear el código de manera unitaria (individual) mientras se va programando. Técnica conocida como Unit Testing, la cual forma parte de la técnica TDD (Test- driven development - desarrollo guiado por pruebas).

✓ **Test de aceptación**

Están más avocados a las pruebas de funcionalidad, es decir al comportamiento funcional del código. Estas pruebas, a diferencia de los Test Unitarios, son definidas por el cliente y llevadas a cabo mediante herramientas como Selenium, por ejemplo, y basadas en Casos de Uso reales que definirán si la funcionalidad desarrollada, cumple con los objetivos esperados (criterios de aceptación). (...)

✓ **Test de integración**

Como su nombre lo indica, tienen por objeto, integrar todos los test que conforman la aplicación, a fin de validar el correcto funcionamiento de la misma, evitando que nuevos desarrollos, dañen a los anteriores.

• **Práctica #9: Código estándar (Coding Standards)**

Los estándares de escritura del código fuente, son esenciales a la hora de programar. Permiten hacer más legible el código y más limpio a la vez de proveer a otros programadores, una rápida visualización y entendimiento del mismo.

Algunos lenguajes de programación como Python, poseen sus propias reglas de estilo, estandarizando la escritura del código fuente, por medio de las PEP. Otros lenguajes “menos prolijos” como PHP, no tienen reglas de estilo o estándares definidos (aunque Zend Framework, está realizando un esfuerzo por proponerlos), y es aquí donde el equipo deberá ponerse de acuerdo, definir y documentar sus propias reglas. (...)

- **Práctica #10: Propiedad colectiva**

Para eXtreme Programming no existe un programador “dueño” de un determinado código o funcionalidad. La propiedad colectiva del código tiene por objetivo que todos los miembros del equipo conozcan “que” y “como” se está desarrollando el sistema, evitando así, lo que sucede en muchas empresas, que existen “programadores dueños de un código” y cuando surge un problema, nadie más que él puede resolverlo, puesto que el resto del equipo, desconoce cómo fue hecho y cuales han sido sus requerimientos a lo largo del desarrollo.

- **Práctica #11: Integración continua (Continuous Integracion)**

La integración continua de XP propone que todo el código (desarrollado por los miembros del equipo) encuentren un punto de alojamiento común en el cual deban enviarse los nuevos desarrollos, diariamente, previo correr los test de integración, a fin de verificar que lo nuevo, no “rompa” lo anterior.

Estos puntos de alojamiento en común, suelen ser repositorios, los cuales pueden manejarse ampliando las ventajas de la integración, mediante software de

control de versiones como Bazaar, Git o SVN (entre otros). (...)

- **Práctica #12: Juego de planificación (Planning Game)**

Con diferencias pero grandes similitudes con respecto a la planificación del Sprint en Scrum, la dinámica de planificación de eXtreme Programming llevada a cabo al inicio de la iteración, suele ser la siguiente:

- 1) El cliente presenta la lista de las funcionalidades deseadas para el sistema, escrita con formato de “Historia de Usuario”, en la cual se encuentra definido el comportamiento de la misma con sus respectivos criterios de aceptación.
- 2) El equipo de desarrollo estima el esfuerzo que demanda desarrollarlas, así como el esfuerzo disponible para el desarrollo (las iteraciones en XP, suelen durar 1 a 4 semanas). Estas estimaciones pueden hacerse mediante cualquiera de las técnicas de estimación.
- 3) El cliente decide que Historias de Usuario desarrollar y en qué orden. (Bahit, 2012, págs. 72-82)

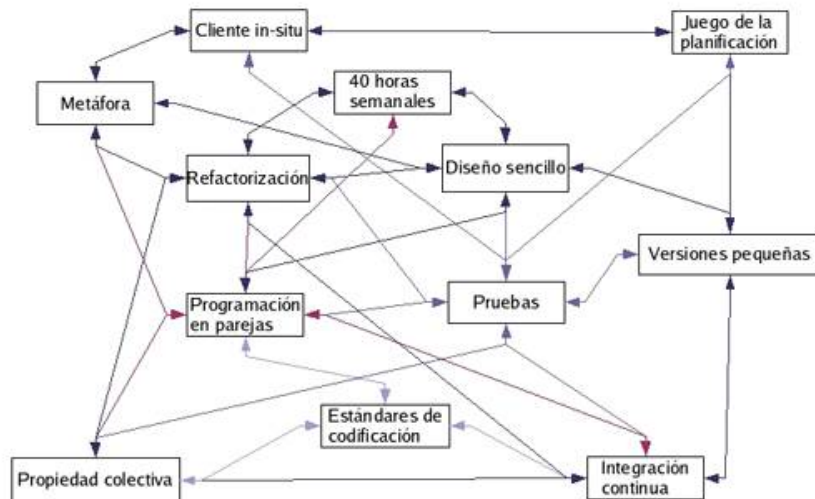


Figura 1.1: Prácticas técnicas de XP
Fuente: (Beck, 2000)

Fases de XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1. (Jeffries, Anderson, & Hendrickson, 2001)

El ciclo de vida ideal de XP consiste de seis fases:

- **Fase I: 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. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- **Fase II: 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.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

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. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto,

determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

- **Fase III: 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 y 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.

- **Fase IV: 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).

- **Fase V: 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.

- **Fase VI: 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. (Beck, 2000)

Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (Jeffries, Anderson, & Hendrickson, 2001)

Roles

Hay muchos roles que las personas deben desempeñar en los proyectos de desarrollo de XP, e incluso se requerirá que algunas personas desempeñen múltiples roles durante el esfuerzo. Los siete roles son: **programador, cliente, probador, rastreador, entrenador, consultor** y (medio en broma) "**gran jefe**". (...)

Con frecuencia, a los **programadores** se les considera como el corazón del desarrollo de XP. Sin embargo, las fortalezas de los programadores de XP son bastante diferentes a las de otros programadores, porque a los primeros se les exige ante todo que sean excelentes comunicadores. Las habilidades de comunicación entran en juego tan pronto como empieza un esfuerzo de desarrollo, porque la XP exige el trabajo en parejas de programación, en el cual cada programador codifica, aunque suele haber un programador principiante y uno experimentado en una tarea determinada. Como

programador, usted también necesita contar con excelentes habilidades técnicas para programar, refactorizar y realizar pruebas unitarias al código que escriba. (...)

El siguiente rol que describiremos es el de **cliente**. La mejor forma de describirlo es que el cliente debe adoptar nuevas cualidades, algunas de ellas muy similares a las de un programador, conservando al mismo tiempo la esencia de lo que necesita el sistema. El cliente más adecuado para el equipo de XP es alguien que será usuario del sistema y que conoce la funcionalidad de negocios que éste requiere. Con esto en mente, el cliente debe aprender a escribir relatos de usuario, aprender a escribir pruebas funcionales para las aplicaciones que generen los programadores, y tomar decisiones acertadas sobre las características esenciales del sistema e incluso sobre ajustes a la programación del proyecto y a las fechas de entrega. Los clientes también necesitan demostrar que tienen valor para enfrentar problemas graves referentes a la programación del proyecto al funcionamiento del sistema.

Un tercer rol es el de **probador** de un equipo de desarrollo de XP. A los programadores se les pide que realicen pruebas unitarias y de funcionamiento del nuevo código que se haya escrito. El programador también necesita comunicarse con el cliente sobre las pruebas de funcionamiento, realizar pruebas con regularidad, dar mantenimiento a las herramientas de prueba y elaborar informes precisos acerca de los resultados de las pruebas.

Otro rol en el equipo de desarrollo de XP es el de **rastreador**. Éste da seguimiento al progreso general del grupo calculando el tiempo que toman sus tareas y el

progreso general hacia sus metas. El rastreador realiza estimaciones de tiempo, pero también da retroalimentación acerca de las estimaciones del equipo. ¿Eran demasiado bajas o demasiado altas? ¿En qué porcentaje fueron incorrectas? En el rol de rastreador usted podrá ofrecer esta valiosa información al equipo, con el fin de que mejoren la precisión de sus estimaciones. Los rastreadores también fungen como memoria del equipo, al dar seguimiento a los resultados de todas las pruebas de funcionamiento. Asimismo, registran los defectos reportados y el nombre del programador a cargo del manejo de cada defecto. Además, dan seguimiento a los casos de prueba agregados para resolver los defectos.

A continuación consideraremos el rol de **entrenador**, quien con frecuencia es una mano invisible que guía el proceso general. Dado que una de las características distintivas del desarrollo de XP es que cada persona acepta la responsabilidad por sus acciones, un entrenador podría parecer innecesario. Sin embargo, los entrenadores son muy importantes. Por ejemplo, ellos conservan la calma cuando todo el equipo está asustado. Ellos moldean las situaciones de manera indirecta (la mayor parte del tiempo), y sólo de vez en cuando necesitan retirar con firmeza el mando a un programador errático, volver a encarrilarlo y devolverle las riendas otra vez. (...) Los entrenadores procuran resaltar las mejores cualidades de todos los demás miembros del equipo y mantenerse ellos en segundo plano la mayor parte del tiempo.

El próximo rol que examinaremos es el de **consultor**. El rol de un experto en consultoría técnica es muy singular. Si usted se está desempeñando como consultor de un equipo, sus miembros le pedirán que resuelva los

problemas junto con ellos, fastidiándolo a cada momento y cuestionando cualquier suposición que usted haga. Lo que los equipos de desarrollo de XP esperan de un consultor es que éste les enseñe a resolver sus propios problemas. (...)

El último rol que consideraremos para el equipo de desarrollo de XP es el de **gran jefe o líder**. El equipo espera que el gran jefe confíe en ellos, demuestre disposición para apegarse a los valores y principios a los que ellos se apeguen, y que tenga la capacidad de señalar un error si el equipo se desvía del camino. El equipo requerirá mantenerse en comunicación con usted (incluso los pequeños cambios al diseño o las desviaciones de otras metas). Su tarea es conseguir que la comunicación fluya sin convertirse en una ola de rumores. Ante todo, usted no desea ser un obstáculo para cualquier cosa razonable que el equipo intente realizar. Éste es un rol que exige una total convicción en el enfoque de XP y de que si todos en el equipo se apegan a sus valores y principios básicos, es muy probable que lograrán algo que valga la pena. (Kendall & Kendall, 2005, págs. 73-75)

3.1.2.2.5. Arquitectura Cliente / Servidor

ARQUITECTURA MVC

Fue creada en 1979 por Trygve Reenskaug. Es un patrón que permite separar la GUI, de los datos y de la lógica apoyándose en tres componentes. A saber:

- **Modelo**

Esta es la representación de los datos y reglas de negocio (mundo del problema). Es el encargado de manejar un registro de las vistas y de los controladores que existen en el sistema.

- **Vista**

Permite mostrar la información del modelo en un formato adecuado que permita que se de la interacción Además de poseer un registro acerca del controlador asociado y brinda el servicio de update que puede ser usado tanto por el controlador como por el modelo.

- **Controlador**

Responde a los eventos provocados por el usuario (se da un clic, se digita un texto, etc) que implican cambios en el modelo y la vista, dando una correcta gestión a las entradas del usuario. (Jaramillo, Cardona, & Villa, 2008, págs. 120-121)

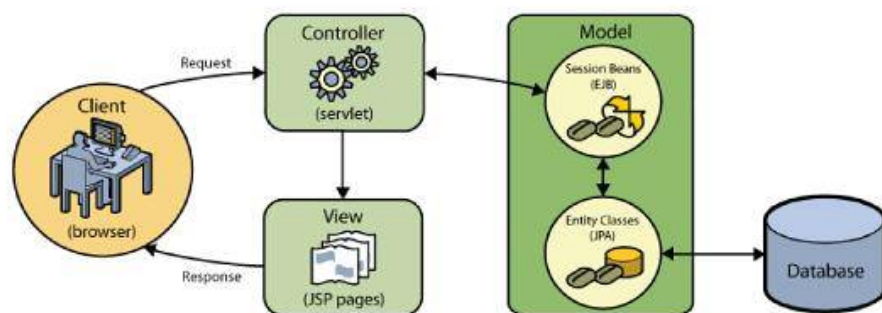


Figura 1.2: Arquitectura Modelo Vista Controlador
Fuente: (Ordax & Ocaña, 2012, pág. 71)

En un sistema web, el modelo está formado por las clases Java que contienen la lógica de negocio y la conexión a la base de datos. La vista son los JSP y el controlador está formado por los servlets.

JavaServer Pages

Una JavaServer Page (JSP) es un fichero de extensión *.jsp que utiliza principalmente lenguajes de etiquetado como el HTML u otros menos extendidos como WML,

SVG, cHTML, etc., en el que se puede embeber mediante unas etiquetas especiales, código Java.

A la hora de su ejecución, el servidor Java EE convertirá estos ficheros en un Java Servlet. Es decir, generará de forma automática, un Java Servlet en el que se incluirá todos los fragmentos de Java embebidos, y añadirá sentencias de impresión con el lenguaje de etiquetado de presentación estático. (Ordax & Ocaña, 2012, pág. 118)

- **Ciclo de vida de un JavaServer Page (JSP)**

Los JavaServer Page siguen un ciclo de vida definido por el API de la especificación Java EE. En concreto el interface:

```
javax.servlet.jsp.HttpJspBase
```

Dicho ciclo de vida es el mismo que el de los JavaServlets ya que (...) los JavaServer Page terminan siendo transformados por el Servidor de Aplicaciones Java EE en un Java Servlet. La única diferencia radica en los nombres.

Los métodos del ciclo de vida de un JavaServer Page son:

```
Public void _jspInit();
```

```
Public _jspService(HttpServletRequest request,  
    HttpServletResponse response);
```

```
Public void _jspDestroy;
```

Si quisiéramos sobrescribir el comportamiento de inicialización o finalización de un JSP implica siempre sobrescribir el método `_jspService`. (Ordax & Ocaña, 2012, págs. 127-128)

Servlet

Los Java Servlets son un tipo de componente Java EE que se empaquetan en un módulo web (WAR) y se

despliegan y ejecutan en el contenedor web de un Servidor de Aplicaciones Java EE. Adicionalmente, dentro del patrón de diseño MVC ocupan el espacio del Controlador, como receptor/validador de peticiones, seleccionador de la lógica a ejecutar y seleccionador de la vista a presentar como resultado. (Ordax & Ocaña, 2012)

Un Servlet interacciona con un cliente Web mediante el protocolo HTTP, a través de un mecanismo de petición/respuesta.

Los servlets permiten extender las posibilidades de un servidor Web, aportando la posibilidad de generar contenido dinámico en respuesta a las peticiones de los clientes. (Aumaille, 2002, pág. 149)

- **Ciclo de vida de un Java Servlet**

Un Java Servlet tiene definido un ciclo de vida, forzado por la interface `javax.servlet.Servlet`, por tanto siempre que queramos implementar un Java Servlet, nuestra clase deberá implementar dicho interface. Es una especie de contrato entre el Java Servlet y el contenedor web del Servidor de Aplicaciones Java EE para poder funcionar.

Dicho ciclo de vida tiene los siguientes estados:

- 1) Cuando se recibe una petición de la ejecución de un Java Servlet en el Servidor de Aplicaciones Java EE y este no ha sido invocado nunca, el contenedor web instancia la clase del Java Servlet (llama a su constructor) y acto seguido invoca el método:

Public void init(ServletConfig config);

Dicho método, solo se ejecuta una vez en la vida del `JavaServlet` y se suele utilizar para llevar a

cabo todas las labores de inicialización que necesitemos para su posterior ejecución. (...)

- 2) Una vez ya está inicializado, entonces invoca el método:

```
Public void service(ServletRequest request,  
ServletResponse response);
```

Dicho método se ejecuta cada vez que se recibe una petición para este Java Servlet. En su implementación, colocaremos la lógica para la que se creó dicho Java Servlet. (...)

- 3) Si por alguna razón el contenedor web necesita eliminar el Java Servlet (se detiene el servidor de aplicaciones Java EE, se queda sin recursos de memoria y necesita liberar espacio, etc...), entonces invoca el método:

```
Public void destroy();
```

Dicho método solo se ejecuta una vez en la vida del Java Servlet y se suele utilizar para liberar todos aquellos recursos que pudiéramos estar utilizando, para dejar el sistema lo más limpio posible. (Ordax & Ocaña, 2012, págs. 84-85)

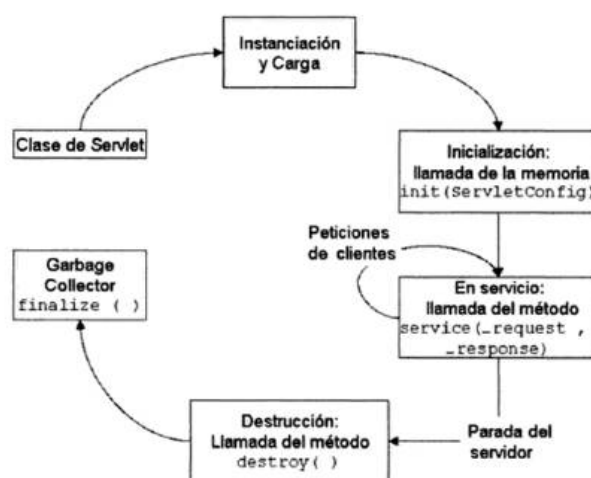


Figura 1.3: Ciclo de vida de un Java Servlet
Fuente: (Aumaille, 2002, pág. 150)

- **Ventajas de los Servlets**

- Es portable y evolutivo, al estar escrito en Java, y sólo necesita un contenedor web para su ejecución.
- Es potente, porque es cargado en memoria desde su primera llamada o desde el inicio del servidor de aplicación. Si se actualiza el Servlet será cargado de nuevo automáticamente por el contenedor Web, sin que debamos reiniciarlo.
- Está disponible, porque se ejecuta en un contexto multitarea. Se crea una sola instancia que crea una hebra para cada petición cliente. (Aumaille, 2002, págs. 149-150)

3.1.2.2.6. Complementos

JAVASCRIPT

Se presenta como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de internet.

El programa en JavaScript tiene la particularidad de estar insertado dentro mismo del documento HTML que lo presenta al usuario y no es por ello un programa aparte (...) El programa en JavaScript reconoce eventos, son acciones de JavaScript (...) creados por el usuario, definiendo así un sistema interactivo. Podemos por ello crear formularios que verifiquen la validez de la información e interpreten ésta en el mismo programa contenido en el documento HTML sin necesidad de comunicación por la red. (Sánchez, 2001, pág. 9)

Diferencias entre Java Y JavaScript

Muchas personas no entienden la diferencia entre Java y JavaScript y generalmente tienden a confundirlos pensando que se trata del mismo lenguaje. Java fue desarrollado por Sun Microsystems como un lenguaje

orientado a objetos opuesto al lenguaje de bolsillo de JavaScript cuya estructura ésta basada en objetos. En Java se pueden crear aplicaciones independientes para Web llamadas applets. (...)

JavaScript es muy parecido a Java en su sintaxis, pero ahí acaban muchas de sus similitudes. JavaScript es un lenguaje interpretado, lo que significa que el código JavaScript se ejecuta directamente en el navegador y no requiere compilación anterior a la ejecución, además JavaScript tiene un conjunto de tipos de datos más reducido. (...)

Java define sus programas sobre un conjunto de clases y sus métodos. Estas clases y métodos así como la característica de herencia convergen en una jerarquía de clases lo suficientemente compleja. Por el otro lado JavaScript pretende ser un lenguaje más pequeño más similar a lenguajes como HyperTalk o dBase. (Sánchez, 2001, págs. 13-14)

<i>Java</i>	<i>JavaScript</i>
<p>Se compila y como resultado el compilador genera un archivo binario que se da al cliente para su ejecución</p>	<p>Sólo se requiere un archivo de texto que será interpretado por el browser</p>
<p>Un applet es referenciado desde un documento HTML, pero se transfiere al cliente como un archivo diferente</p>	<p>La codificación de los scripts se coloca dentro del documento HTML</p>
<p>Java requiere la declaración de variables</p>	<p>JavaScript no requiere la declaración de variables</p>
<p>La etiqueta para llamar a un applet es: <APPLET></p>	<p>La etiqueta para declarar un guión es: <SCRIPT></p>
<p>Las referencias a los objetos se verifican a tiempo de compilación</p>	<p>Las referencias del objeto se verifican durante el tiempo de ejecución</p>

Figura 1.4: Diferencias entre Java y JavaScript
Fuente: (Sánchez, 2001, pág. 15)

JDK (JAVA DEVELOPMENT KIT)

Se trata de un paquete de software que puede utilizar para desarrollar aplicaciones basadas en Java.

Java Development Kit es necesario para desarrollar aplicaciones de Java.

El JDK necesita más espacio en el disco (que el JRE) porque contiene el JRE junto con varias herramientas de desarrollo. (Java, s.f.)

AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas, estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y Document Object Model (DOM). (Universidad de Alicante, s.f.)

JQUERY

Es la librería web más popular disponible en estos días. La librería jQuery es gratuita y fue diseñada para simplificar la creación de sitios web modernos. Facilita la selección de elementos HTML, la creación de animaciones y efectos, y también controla eventos y ayuda a implementar Ajax en nuestras aplicaciones. (Gauchat, 2012, pág. 101)

HTML 5

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto (...).

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio. (Gauchat, 2012, pág. 1)

Las novedades de Html5 y sobretodo del trinomio conceptual “Html5 + CSS3 + aplicaciones JavaScript” son numerosas. (...)

- Un nuevo doctype simplificado y unificado.
- Supresión de etiquetas y atributos de presentación.
- Nuevas etiquetas semánticas o de organización.
- Nuevas etiquetas de audio y de video que ya no necesitan más plugins dedicados.
- Profusión de formularios innovadores como por ejemplo cursores o calendarios, y la integración de forma nativa en los navegadores de la validación de datos. (Van Lancker, 2012, pág. 15)

CSS 3

Cascading style sheets. Hoja de estilo en cascada que define los estilos para dar formato al contenido html. (Van Lancker, 2012, pág. 177)

Css es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño color, fondo, bordes, etc. (Gauchat, 2012, pág. 32)

APPLET

Un applet es un tipo de aplicación Java específica que permite ejecutar código Java en el interior de una página Web. El principal objetivo de los applets es añadir interactividad y dinamismo a una página web.

La llamada del applet se incorpora al código html de la página. Cuando el navegador analiza la página html que el servidor web acaba de transmitirle y encuentra una etiqueta que corresponde a un applet, descarga el código del applet y lanza su ejecución. La ventaja principal de un applet en comparación con una aplicación es sin duda la ausencia de instalación

necesaria en los puestos clientes. Otras ventajas de la utilización de un applet son:

- Los clientes siempre dispondrán de la última versión del código. A cada carga de la página web por el navegador, el código del applet se descarga también desde el servidor web. Si una nueva versión del código está disponible, ésta sólo se debe desplegar en el servidor web para que todos los clientes se puedan beneficiar de ello.
- Todos los recursos útiles para el buen funcionamiento de la aplicación estarán disponibles. Como el código del applet, los recursos que necesita se transmiten también del servidor web al navegador. Por lo tanto, sólo deben estar presentes en el servidor web.
- No hay riesgo de que la ejecución del código del applet provoque daños en el puesto cliente. La máquina virtual Java del navegador se encarga de la ejecución del código, y, dado que lo considera potencialmente peligroso, lo ejecuta con un conjunto limitado de derechos.

Sin embargo los applets comportan una pequeña desventaja ya que la máquina virtual Java del jdk no los puede ejecutar de manera autónoma. Los applets deben estar incorporados obligatoriamente a una página html para que la máquina virtual Java del navegador se encargue de ellos. Naturalmente, este navegador debe disponer de una máquina virtual. (Groussard, 2012, pág. 171)

3.1.3. Formulación del Problema Científico.

¿De qué manera se optimizará el proceso de inscripción de postulantes de la Universidad Nacional Autónoma de Chota?

3.1.4. Hipótesis

Si se implementa un sistema web, entonces se optimizará el proceso de inscripción de postulantes de la Universidad Nacional Autónoma de Chota.

3.1.5. Operacionalización de Variables

	Conceptualización de Variables	Dimensiones	Indicadores
I N D E P E N D I E N T E	Sistema bajo entorno web	<ul style="list-style-type: none"> ✓ Usabilidad como producto. ✓ Usabilidad como proceso. ✓ Funcionalidad. 	Flexible
			Adaptabilidad
			Exactitud
			Eficiencia
D E P E N D I E N T E	Proceso de inscripción de postulantes de la Universidad Nacional Autónoma de Chota	<ul style="list-style-type: none"> ✓ Información en tiempo real. 	Tiempo de respuesta de información
			Tiempo de proceso de información
			Rapidez

Tabla 3.2: Operacionalización de variables
Fuente: Elaboración propia

3.1.6. Objetivos

3.1.6.1. Objetivo General

Desarrollar un sistema bajo entorno web que permita optimizar el proceso de inscripción de postulantes para administrar de manera segura, precisa y eficiente la información registrada.

3.1.6.2. Objetivo Específicos

- Determinar las ventajas de automatizar los formatos manuales de inscripción de postulantes.
- Diseñar una base de datos para el registro de la información que sea adaptable en caso de futuras actualizaciones.
- Determinar la mejor metodología ágil para el desarrollo del sistema web encargado del control del proceso de inscripción de postulantes.
- Desarrollar los sistemas web (inscripción y administrativo), siguiendo el patrón de diseño MVC (Modelo Vista Controlador).
- Desarrollar un sistema que permita de manera precisa y eficiente el registro de huellas digitales y la captura de fotos de los postulantes inscritos en el sistema.

3.1.7. Justificación e Importancia

3.1.7.1. Justificación

El desarrollo del sistema se orienta a la mejora de los servicios prestados por la Universidad Autónoma de Chota para el proceso de inscripción de los postulantes logrando el manejo eficiente de la información.

Con la innovación del sistema en plataforma WEB los postulantes podrán inscribirse accediendo al portal de la página principal de la Universidad Autónoma de Chota. Los postulantes podrán tener acceso al sistema en línea (Internet)

o desde la sala de cómputo de la misma universidad (intranet). Esto significa una mejora en cuanto al tiempo de atención y reducción de errores al momento de la inscripción, además de garantizar la seguridad e integridad de los datos.

Para el personal encargado, el sistema será de mucha ayuda porque le brindará un acceso más rápido a los datos y contará con información confiable y actualizada para la generación de reportes.

Para la universidad el proyecto se presenta como una alternativa para estar a la vanguardia con sus competidores.

3.1.7.2. Importancia

La importancia de esta investigación usando tecnología WEB radica en proporcionar soluciones a corto, mediano y largo plazo ante la imperiosa necesidad de mejorar las actividades relacionadas con el proceso de inscripción de los postulantes.

3.2. Marco Metodológico

3.2.1. Diseño de contrastación de Hipótesis

Cuasi - Experimental

3.2.2. Tipo de investigación

Tecnológico formal. Se enmarca dentro del campo de programación y el análisis y diseño de sistemas.

CAPITULO IV

ASPECTO ADMINISTRATIVO



4. ASPECTO ADMINISTRATIVO

4.1. Cronogramas de actividades

En la Figura 4.1 podemos observar el cronograma del proyecto detallado por actividades.

Hemos seguido las fases de la programación extrema, por ello aparecen las historias de usuario y las tareas que se realizaron. En cada una se especifica la fecha de inicio, la fecha de fin y la duración en días.

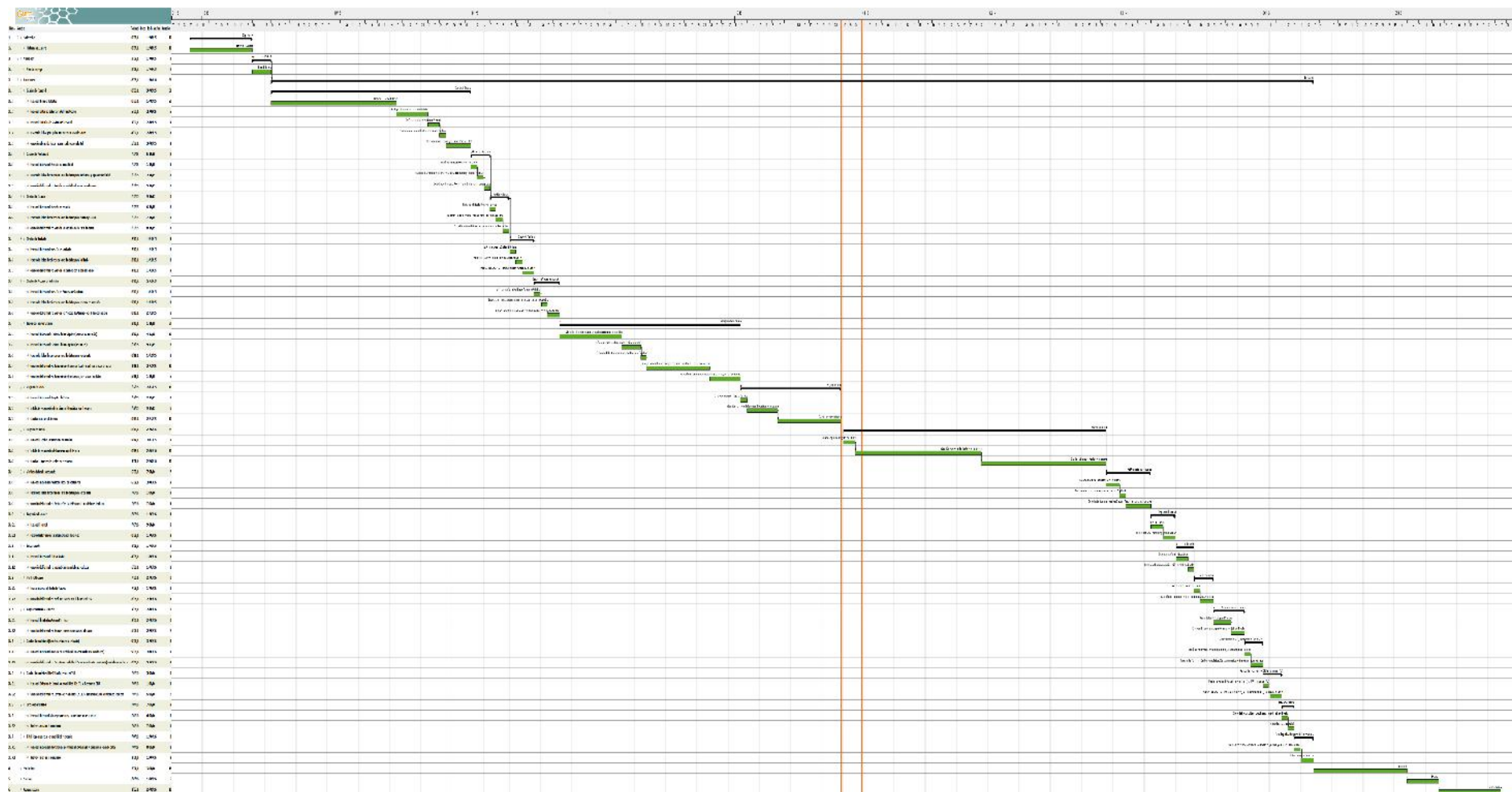


Figura 1.1: Cronograma del Proyecto
Fuente: Elaboración propia

4.2. Presupuesto

4.2.1. Bienes

Bienes	Cantidad	Precio Unitario (S/.)
Lector de huella digital	1	S/. 150.00
Laptop Lenovo	1	S/. 3,500.00
Cámara Profesional Canon	1	S/. 1,500.00
Impresora HP	1	S/. 1,500.00
Sub Total		S/. 6,650.00

*Tabla 4.1: Bienes
Fuente: Elaboración propia*

Resumen Bienes

Bienes	Total (S/.)
Total S/.	S/. 6,650.00

*Tabla 4.2: Resumen de bienes
Fuente: Elaboración propia*

4.2.2. Inversión

4.2.2.1. Infraestructura

Licencia de Software	Observación	Total (S/.)
PgModeler	Todos los programas mencionados son brindados por la universidad.	S/.0.00
PostgresSQL		S/.0.00
Windows 7 64 bit		S/.0.00
Microsoft Office 2013		S/.0.00
Gantt proyect	Adquirido por los programadores	S/.0.00
sublime		S/.0.00
Netbeans 8.0		S/.0.00
Sub Total		S/.0.00

*Tabla 4.3: Infraestructura - Licencia de Software
Fuente: Elaboración propia*

Materiales de Oficina	Cantidad	Precio Unitario (S/.)	Total (S/.)
Papel Bond A4	8	S/. 15.00	S/.120.00
Tinteros para impresora	4	S/. 20.00	S/.80.00
Lapiceros	4	S/. 2.50	S/.10.00
Resaltador	2	S/. 3.00	S/.6.00
Total			S/.216.00

*Tabla 4.4: Infraestructura - Materiales de Oficina
Fuente: Elaboración propia*

Resumen Infraestructura

Descripción	Costo (S/.)
Licencia de Software	S/.0.00
Materiales de Oficina	S/.216.00
Total S/.	S/.216.00

*Tabla 4.5: Resumen de infraestructura
Fuente: Elaboración propia*

4.2.2.2. Servicios

Servicios	Precio (S/.)
Desarrolladores	20,000.00
Transporte	1000.00
Internet	800.00
Energía Eléctrica	1000.00
viáticos	1000.00
Fotocopias	100.00
otros	500.00
Sub Total	S/. 24,400.00

*Tabla 4.6: Servicios
Fuente: Elaboración propia*

Capacitación en el uso del sistema	Total (S/.)
Capacitación por 3 días de 1 hora por día	S/.300.00
Sub Total	S/.300.00

*Tabla 4.7: Capacitación en el uso del sistema
Fuente: Elaboración propia*

Resumen Servicios

Descripción	Total (S/.)
Servicios	S/. 24,400.00
Capacitación en el uso del sistema	S/.300.00
Total S/.	S/.24,700.00

*Tabla 4.8: Resumen Servicios
Fuente: Elaboración propia*

Resumen Inversión

Descripción	Total (S/.)
Resumen Infraestructura	S/.216.00
Resumen Servicios	S/.24,700.00
Total S/.	S/.24,916.00

*Tabla 4.9: Resumen inversión
Fuente: Elaboración propia*

4.2.3. Costo de Operación

4.2.3.1. Costo Fijo

Personal	Observación	Total (S/.)
Administrador Base de Datos	Todos los mencionados son brindados por la universidad.	S/.0.00
Administrador del servidor Web		S/.0.00
Administrador del Sistema Web		S/.0.00
Administrador del Sistema Administrativo		S/.0.00
Sub Total		S/.0.00

*Tabla 4.10: Costo fijo
Fuente: Elaboración propia*

4.2.3.2. Costo Variable

Materiales de Oficina	Cantidad	Precio Unitario (S/.)	Total (S/.)
Papel Bond	8	S/. 15.00	S/.120.00
Tinteros para impresora	4	S/. 20.00	S/.80.00
Lapiceros	4	S/. 2.50	S/.10.00
Resaltador	2	S/. 3.00	S/.6.00
Total			S/.216.00

*Tabla 4.11: Costo variable
Fuente: Elaboración propia*

Resumen

Descripción	Total (S/.)
Costo fijo	S/.0.00
Costo Variable	S/.216.00
Total	S/.216.00

*Tabla 4.12: Resumen
Fuente: Elaboración propia*

4.3. Financiación

Los bienes y los materiales de oficina en cuanto al desarrollo del software son financiados por los autores. En lo que respecta a la infraestructura, servicios y remuneraciones en la capacitación para el uso del sistema serán asumidos por la Universidad.

4.4. Análisis de Factibilidad

4.4.1. Factibilidad Operativa

- Existe gran predisposición por parte de la universidad para el desarrollo del proyecto brindando la información necesaria para el cumplimiento de los objetivos del sistema web.

- Al personal que utilizará el sistema web se le brindará la capacitación necesaria para el uso correcto del mismo.
- La universidad cuenta con el hardware necesario para la implementación del sistema web.
- Por eso es factible operacionalmente el desarrollo del sistema ya que el personal se encuentra poco conforme de cómo se maneja actualmente dichos procesos.

4.4.2. Factibilidad Técnica

- Para seguridad del funcionamiento del sistema web, estará alojado en un ordenador de la oficina Central de Informática que tendrá la función de servidor.
- En cuanto al sistema Administrativo se les otorgará los roles y los permisos respectivos a los usuarios para evitar el acceso no autorizado al sistema.
- El hardware y software necesario con lo que debe contar la universidad son:

Cantidad	Nombre	Descripción
4	Computadora de Escritorio - PC	Procesador core-i7 de 5ta generación con disco duro de 1tb, memoria ram de 8gb, con puertos usb de 3.0, monitor de 20".
3	Supresor a picos	
5	Ups	Debe estar conectada a cada computadora y a la impresora, para posibles bajas de tensión eléctrica.
5	Estabilizador solido	Estarán conectados a cada computadora y a la impresora para posibles bajas de tensión eléctrica.

Tabla 4.13: Factibilidad Técnica
Fuente: Elaboración propia

- Por lo tanto es factible Técnicamente el desarrollo del sistema Web porque existen equipos que hacen posible su óptimo funcionamiento.

4.4.3. Factibilidad Económica

4.4.3.1. Beneficios

4.4.3.1.1. Beneficios tangibles

- La información se procesará más rápido.
- Generación de reporte.
- Acceso a la información.

4.4.3.1.2. Beneficios Intangibles

- Optimización de los procesos semestrales
- Fiabilidad de la información.
- Mejor toma de decisiones.
- Aumenta la satisfacción del postulante.

4.4.3.1.3. Determinación de Factibilidad

Estado de Ganancias y Pérdidas

Concepto		Periodo					
		0	1	2	3	4	5
1	Ingreso por Venta del proyecto		7000	7000	7000	7000	7000
2	Costos Variables		216	216	216	216	216
3	Costos Fijos	-	-	-	-	-	-
	Utilidad Bruta (1) - [(2)+(3)]		6784	6784	6784	6784	6784
4	Impuestos		910	910	910	910	910
5	Utilidad después de impuestos		5874	5874	5874	5874	5874

Tabla 4.14: Estado de ganancias y pérdidas
Fuente: Elaboración propia

Flujo de Efectivo

	0	1	2	3	4	5
Inversión	24,916					
Valor de Rescate	-	-	-	-	-	-
Utilidad después de impuestos		5,874	5,874	5,874	5,874	5,874
Flujo Neto de Efectivo	24,916	5,874	5,874	5,874	5,874	5,874

*Tabla 4.15: Flujo de efectivo
Fuente: Elaboración propia*

- **VAN:** Valor Actual Neto, es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. Cuya fórmula es la siguiente:

$$\text{VAN} = \text{FC}_0 + \text{FC}_1 / (1+r) + \dots + \text{FC}_{n-1} / (1+r)^{n-1} + \text{FC}_n / (1+r)^n$$

$$\text{VAN} = -24,916 + 5,874 / (1+0.035) + 5,874 / (1+0.035)^2 + 5,874 / (1+0.035)^3 + 5,874 / (1+0.035)^4 + 5,874 / (1+0.035)^5$$

$$\text{VAN} = 1605$$

- **TIR:** Tasa Interna de Retorno o de Rentabilidad, es un método de valoración de inversiones que mide la rentabilidad de los cobros y los pagos actualizados, generados por una inversión, es decir en porcentaje.
 $x = \text{TIR}; \text{VAN} = 0$

$$\text{VAN} = \text{FC}_0 + \text{FC}_1 / (1+\text{TIR}) + \dots + \text{FC}_{n-1} / (1+\text{TIR})^{n-1} + \text{FC}_n / (1+\text{TIR})^n$$

$$0 = -24,916 + 5,874 / (1+x) + 5,874 / (1+x)^2 + 5,874 / (1+x)^3 + 5,874 / (1+x)^4 + 5,874 / (1+x)^5$$

$$x = \text{TIR} = 6\%$$

CAPITULO V

DESARROLLO DEL SISTEMA



5. DESARROLLO DEL SISTEMA

5.1. Fase I: Exploración

En esta fase nos entrevistamos con las personas responsables del proceso de admisión de postulantes (ver Tabla 5.1) y recopilamos información mediante las historias de usuario.

	Nombres y Apellidos	Cargo
1	Mg. Richart Hernández Fiesta	Jefe de la Oficina de Admisión – Unach.
2	Tc. Johan Jeampier Briones Delgado	Unidad de Soporte Informático – Unach.
3	Lic. Gilder Cieza Altamirano	Encargado del Centro Pre – Unach.

Tabla 5.1: Personas entrevistadas en la fase de exploración del proyecto
Fuente: Elaboración propia

5.1.1. Historias de Usuario

HISTORIAS DE USUARIO	
Número: 1	Usuario: Postulante
Nombre: Inscripción de Postulante	
Prioridad en negocio: Alta	Iteración asignada: 3
Riesgo en desarrollo: Medio	Puntos estimados: 6
Desarrollador encargado:	
Descripción: El postulante ingresa a la página web de inscripción y proporcionará datos como su número de voucher, su dni y la modalidad a la que postula, el sistema debe verificar que los datos sean correctos, si es así le dará acceso a un formulario para que registre sus datos personales, el colegio o universidad donde estudió y la carrera a la que postula.	
Observación:	

Tabla 5.2: Historia de Usuario 1. Inscripción de Postulante
Fuente: Elaboración propia

HISTORIAS DE USUARIO	
Número: 2	Usuario: Fotos
Nombre: Registro de Fotos	
Prioridad en negocio: Alta	Iteración asignada: 4
Riesgo en desarrollo: Medio	Puntos estimados: 3
Desarrollador encargado:	
Descripción: El postulante se acerca a la OCI para la toma de la foto que aparecerá en su carnet.	
Observación: El postulante previamente se debe haber registrado en la página web.	

*Tabla 5.3: Historia de Usuario 2. Registro de Fotos
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 3	Usuario: Huellas
Nombre: Registro de Huellas	
Prioridad en negocio: Alta	Iteración asignada: 5
Riesgo en desarrollo: Alta	Puntos estimados: 8
Desarrollador encargado:	
Descripción: El postulante se acerca a la OCI para la toma de la huella que aparecerá en su carnet.	
Observación: El postulante previamente se debe haber registrado en la página web.	

*Tabla 5.4: Historia de Usuario 3. Registro de Huellas
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 4	Usuario: Carnet
Nombre: Verificar datos de postulante	
Prioridad en negocio: Media	Iteración asignada: 6
Riesgo en desarrollo: Medio	Puntos estimados: 2
Desarrollador encargado:	
Descripción: El postulante se acerca a la OCI y verifica que los datos que ha registrado en la página web estén correctos, sino el personal encargado actualizará sus datos (excepto DNI y su escuela).	
Observación:	

*Tabla 5.5: Historia de Usuario 4. Verificar Datos de Postulante
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 5	Usuario: Carnet
Nombre: Impresión de carnet	
Prioridad en negocio: Alta	Iteración asignada: 6
Riesgo en desarrollo: Bajo	Puntos estimados: 1
Desarrollador encargado:	
Descripción: El postulante se acerca a la OCI para la impresión y entrega de su carnet.	
Observación: El postulante debe tener registrado su foto y huella y haber verificado que sus datos sean correctos.	

*Tabla 5.6: Historia de Usuario 5. Impresión de carnet
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 6	Usuario: Administrador
Nombre: Cambio de modalidad (De extraordinario a ordinario)	
Prioridad en negocio: Media	Iteración asignada: 8
Riesgo en desarrollo: Medio	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: El postulante registrado bajo el tipo de modalidad extraordinario que no haya alcanzado vacante, tiene el derecho de canjear su carnet para volver a postular al examen ordinario.	
Observación:	

Tabla 5.7: Historia de Usuario 6. Cambio de modalidad (De extraordinario a ordinario)
Fuente: Elaboración propia

HISTORIAS DE USUARIO	
Número: 7	Usuario: Administrador
Nombre: Cambio de modalidad (De CPU a Ingresante CPU)	
Prioridad en negocio: Media	Iteración asignada: 8
Riesgo en desarrollo: Medio	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: El postulante que haya alcanzado vacante en el concurso de admisión bajo la modalidad CPU, deberá acercarse al banco de la Nación a realizar el pago correspondiente para poder canjear su carnet de CPU a Ingresante CPU.	
Observación:	

Tabla 5.8: Historia de Usuario 7. Cambio de modalidad (De CPU a Ingresante CPU)
Fuente: Elaboración propia

HISTORIAS DE USUARIO	
Número: 8	Usuario: Administrador
Nombre: Listado de postulantes	
Prioridad en negocio: Media	Iteración asignada: 8
Riesgo en desarrollo: Bajo	Puntos estimados: 0.4
Desarrollador encargado:	
Descripción: Se requiere un reporte que muestre una lista con los datos personales de los postulantes, procedencia, modalidad y escuela en la que se registraron.	
Observación:	

*Tabla 5.9: Historia de Usuario 8. Listado de postulantes
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 9	Usuario: Administrador
Nombre: Total de postulantes por modalidad y escuela	
Prioridad en negocio: Media	Iteración asignada: 8
Riesgo en desarrollo: Bajo	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: Necesitamos un reporte que muestre el total de postulantes que se han registrado por modalidad y escuela.	
Observación:	

*Tabla 5.10: Historia de Usuario 9. Total de postulantes por modalidad y escuela
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 10	Usuario: Administrador
Nombre: Gestión de personal	
Prioridad en negocio: Alta	Iteración asignada: 1
Riesgo en desarrollo: Bajo	Puntos estimados: 5
Desarrollador encargado:	
Descripción: Se registrarán sus datos personales para que puedan tener acceso al sistema administrativo y se le asignará un usuario y el rol a desempeñar (administrador u otro usuario).	
Observación: Solo un administrador podrá registrar datos de un nuevo personal.	

*Tabla 5.11: Historia de Usuario 10. Gestión de personal
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 11	Usuario: Administrador, Fotos, Huellas, Carnet
Nombre: Iniciar sesión	
Prioridad en negocio: Alto	Iteración asignada: 7
Riesgo en desarrollo: Bajo	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: El personal encargado podrá acceder al sistema ingresando su usuario y contraseña.	
Observación:	

*Tabla 5.12: Historia de Usuario 11. Iniciar sesión
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 12	Usuario: Administrador, Fotos, Huellas, Carnet
Nombre: Modificar contraseña	
Prioridad en negocio: Baja	Iteración asignada: 7
Riesgo en desarrollo: Bajo	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: El personal que tiene acceso al sistema puede cambiar la contraseña asignada por una de su preferencia.	
Observación:	

*Tabla 5.13: Historia de Usuario 12. Modificar contraseña
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 13	Usuario: Administrador
Nombre: Asignar permisos a usuarios	
Prioridad en negocio: Media	Iteración asignada: 7
Riesgo en desarrollo: Bajo	Puntos estimados: 1
Desarrollador encargado:	
Descripción: Solo el administrador del sistema puede realizar registros o ver reportes, los demás usuarios como el de fotos o huellas solo pueden acceder al formulario en que se les ha encargado trabajar.	
Observación:	

*Tabla 5.14: Historia de Usuario 13. Asignar permisos a usuarios
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 14	Usuario: Administrador
Nombre: Gestión de modalidad	
Prioridad en negocio: Media	Iteración asignada: 2
Riesgo en desarrollo: Bajo	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: Se debe tener un registro de las modalidades y tipo de modalidad con las que cuenta la universidad.	
Observación:	

*Tabla 5.15: Historia de Usuario 14. Gestión de modalidad
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 15	Usuario: Administrador
Nombre: Gestión de escuela	
Prioridad en negocio: Media	Iteración asignada: 2
Riesgo en desarrollo: Bajo	Puntos estimados: 0.6
Desarrollador encargado:	
Descripción: Se debe tener un registro de las escuelas con las que cuenta la universidad, y la posibilidad de agregar nuevas.	
Observación:	

*Tabla 5.16: Historia de Usuario 15. Gestión de escuela
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 16	Usuario: Administrador
Nombre: Gestión de tarifario	
Prioridad en negocio: Media	Iteración asignada: 2
Riesgo en desarrollo: Bajo	Puntos estimados: 0.8
Desarrollador encargado:	
Descripción: Se debe tener una lista de las tarifas que deben pagar los postulantes para el concurso de admisión, de acuerdo a la modalidad y la procedencia (nacional o particular), y tener la opción de modificar estas tarifas.	
Observación:	

*Tabla 5.17: Historia de Usuario 16. Gestión de tarifario
Fuente: Elaboración propia*

HISTORIAS DE USUARIO	
Número: 17	Usuario: Administrador
Nombre: Gestión de proceso de admisión	
Prioridad en negocio: Media	Iteración asignada: 2
Riesgo en desarrollo: Bajo	Puntos estimados: 0.8
Desarrollador encargado:	
Descripción: Semestralmente se registra un proceso de admisión y otro de CPU, indicando los plazos disponibles para registrarse en cada modalidad.	
Observación:	

*Tabla 5.18: Historia de Usuario 17. Gestión de proceso de admisión
Fuente: Elaboración propia*

5.2. Fase 2: Planeación

5.2.1. Planificación de Iteraciones

Nº	Nombre	Puntos Estimados	Duración	Iteración
10	Gestión de personal	6.0	32 días	1
14	Gestión de modalidad	0.6	14 días	2
15	Gestión de escuela	0.6		
16	Gestión de tarifario	0.8		
17	Gestión de proceso de admisión	0.8		
1	Inscripción de Postulante	6.0	29 días	3
2	Registro de Fotos	5.0	16 días	4
3	Registro de Huella	8.0	42 días	5
4	Verificar datos de postulante	1.4	11 días	6
5	Impresión de carnet	0.8		
11	Iniciar sesión	0.6	11 días	7
12	Modificar Contraseña	0.6		
13	Asignar permisos a usuarios	1		
6	Cambio de modalidad (De extraordinario a ordinario)	0.6	11 días	8
7	Cambio de modalidad (De CPU a Ingresante CPU)	0.6		
8	Listado de postulantes	0.4		
9	Total de postulantes por modalidad y escuela	0.6		

Tabla 5.19: Iteraciones del proyecto
Fuente: Elaboración propia

5.3. Fase 3: Iteraciones

5.3.1. Primera Iteración

TAREA	
Número de tarea: 1	Número de Historia: 10
Nombre de Tarea: Diseño de la Base de Datos	
Tipo de Tarea: Diseño	Puntos estimados: 4
Fecha Inicio: 18/08/15	Fecha Fin: 14/09/15
Desarrollador encargado:	
Descripción: Se analizará la información proporcionada por el usuario y se diseñará la base de datos conforme a los requerimientos.	

Tabla 5.20: Historia 10 - Tarea 1. Diseño de la Base de Datos
Fuente: Elaboración propia

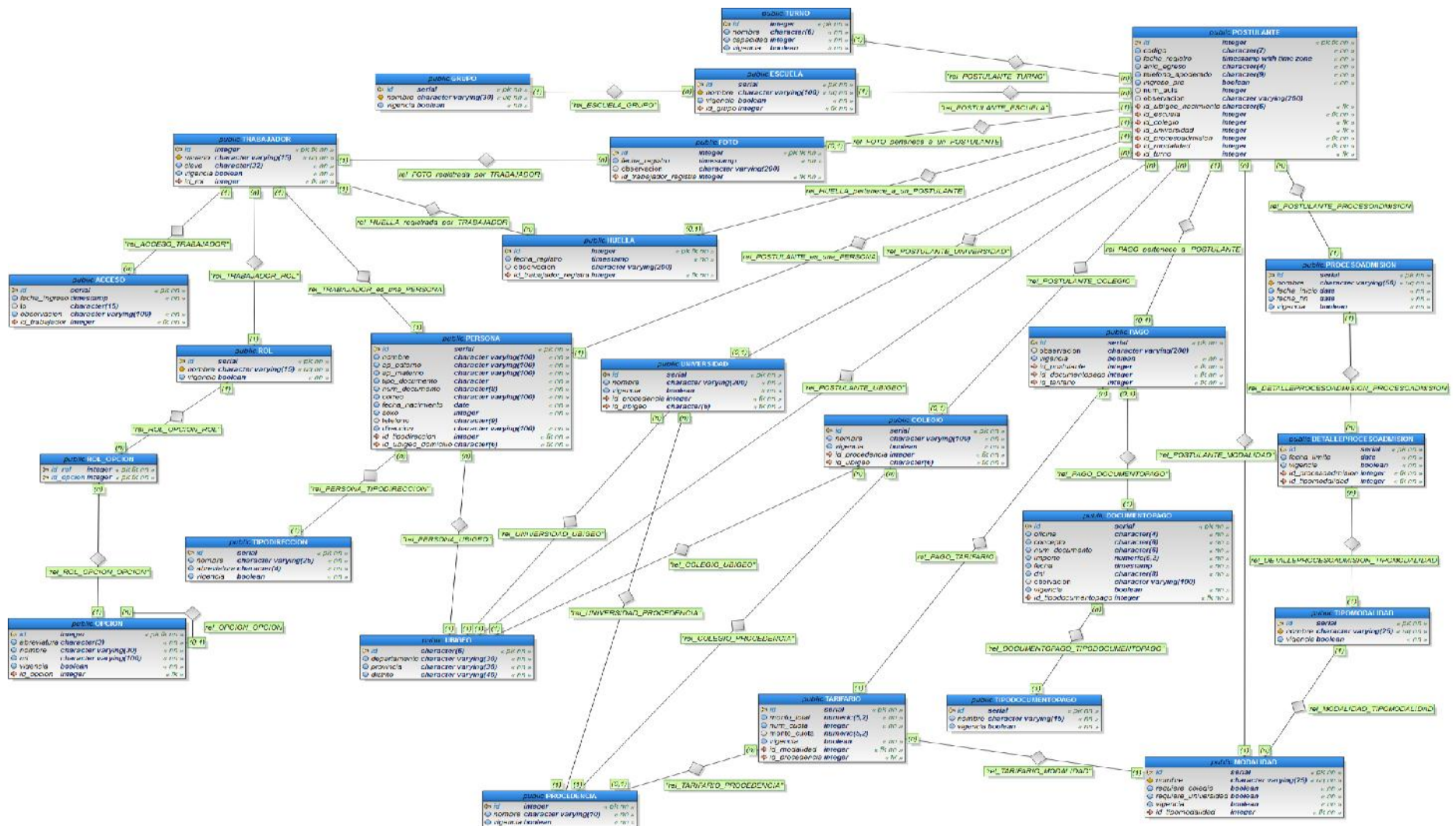


Figura 3.1: Diagrama entidad relación
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 10
Nombre de Tarea: Diseño de Interfaz del Sistema Administrativo	
Tipo de Tarea: Diseño	Puntos estimados: 1
Fecha Inicio: 15/09/15	Fecha Fin: 21/09/15
Desarrollador encargado:	
Descripción: Se diseñará la interfaz principal del sistema con un menú de opciones para que el usuario pueda acceder a los diferentes formularios.	

Tabla 5.21: Historia 10 - Tarea 2. Diseño de la interfaz del sistema administrativo
Fuente: Elaboración propia

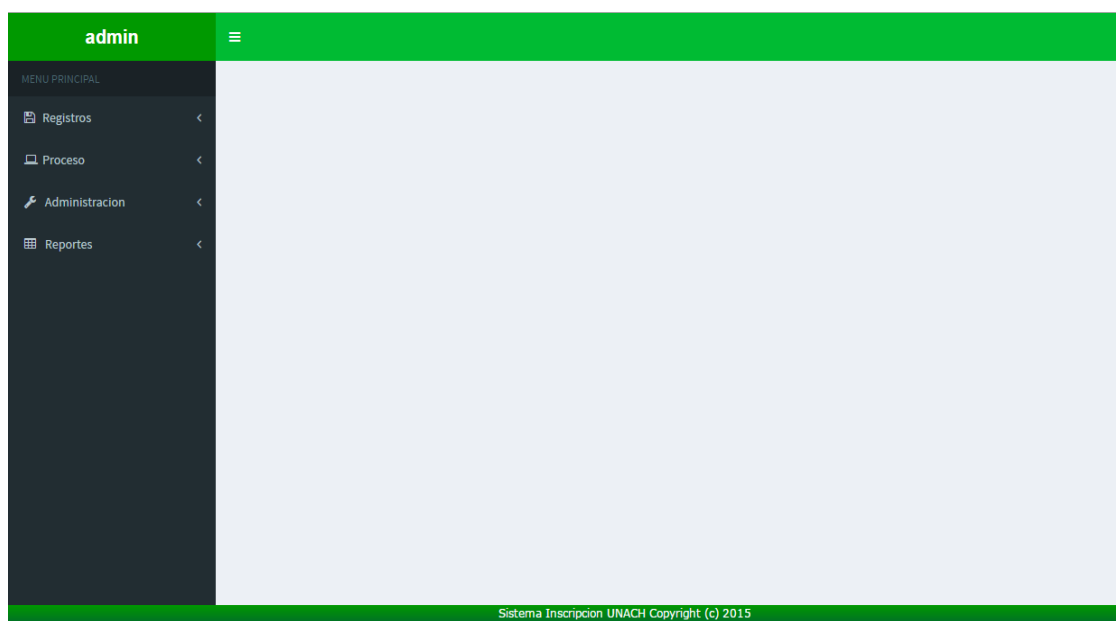


Figura 3.2: Interfaz del sistema administrativo
Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 10
Nombre de Tarea: Diseño de Interfaz de Gestión de Personal	
Tipo de Tarea: Diseño	Puntos estimados: 0.4
Fecha Inicio: 22/09/15	Fecha Fin: 23/09/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará una interfaz con los controles necesarios donde se ingresará los datos personales del nuevo personal, como su nombre completo, dni, teléfono, correo, dirección, su nombre de usuario y el rol que va a desempeñar. - Se mostrará una lista de todo el personal registrado con la opción de editar datos. - También se incluirá una opción para agregar roles de usuario. 	

*Tabla 5.22: Historia 10 - Tarea 3. Diseño de interfaz de Gestión de Personal
Fuente: Elaboración propia*

TAREA	
Número de tarea: 4	Número de Historia: 10
Nombre de Tarea: Desarrollo de las principales funciones en base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 24/09/15	Fecha Fin: 24/09/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se escribirá los scripts de las principales funciones como listado de la tabla Ubigeo y el registro, actualización y listado de la tabla Personal y Rol. 	

*Tabla 5.23: Historia 10 - Tarea 4. Desarrollo de las principales funciones en base de datos
Fuente: Elaboración propia*

TAREA	
Número de tarea: 5	Número de Historia: 10
Nombre de Tarea: Conexión a base de datos y registro de Personal	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 25/09/15	Fecha Fin: 30/09/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se creará una clase para la conexión a base de datos y se escribirá el código que permita que los datos ingresados en el formulario de Personal sean registrados y también puedan ser visualizados. - Solo el nombre, dni, usuario y su rol serán datos obligatorios, los demás serán opcionales, para permitir un registro más rápido. Además se establecerá como contraseña predeterminada el número de su dni. - En la lista inferior aparecerá una opción al lado de cada personal registrado, que al seleccionarla mostrará los datos en las cajas de texto correspondientes para poder modificarlos. 	

Tabla 5.24: Historia 10 - Tarea 5. Conexión a base de datos y registro de Personal
Fuente: Elaboración propia

Figura 3.3: Interfaz de Gestión de personal
Fuente: Elaboración propia

admin

Lista

Mostrar: 10 Entradas

Buscar:

Apellidos y Nombres	N° de Documento	Correo	telefono	Direccion	Usuario	Rol	Vigente	Accion
ADMIN, ADMIN, ADMIN	46660263	ADMIN@UNACH.COM	579494145	AV	admin	ADMINISTRADOR	SI	
PEREZ BONILLA, LUIS	43564253	LUIS@GMAIL.COM	579630146	RESOLCQUIA	lperez	ADMINISTRADOR	SI	
QUINTOS ESTRADA, KATHERINE	46660263	KAT@GMAIL.COM	579494145	PROHIBIDASCAR 128	kquintos	ADMINISTRADOR	SI	
JORGES SANCHEZ, JORGE	17254852	JORGES@HOTMAIL.COM	686444120	CAJ	jorges	HUELLAS	SI	
VALDEERA KUIZ, MARTHA	70458825	MWALDEERA@HOTMAIL.COM	579474649	CALLE LOS ALAMOS 456	mvaldeera	FOTOS	SI	

Mostrando 1 - 5 de 5 entradas

Anterior 1 Siguiente

Sistema Inscripción UNACH Copyright (C) 2015

Figura 3.4: Listado de personal
Fuente: Elaboración propia

Agregar Rol

Datos

Nombre

Registrar Cancelar

Lista

Nombre	Accion
ADMINISTRADOR	
FOTOS	
HUELLAS	

Cerrar

Sistema Inscripción UNACH Copyright (C) 2015

Figura 3.5: Modal para agregar un rol
Fuente: Elaboración propia

5.3.2. Segunda Iteración

TAREA	
Número de tarea: 1	Número de Historia: 14
Nombre de Tarea: Diseño de interfaz de Gestión de modalidad	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 1/10/15	Fecha Fin: 1/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará la interfaz con los controles necesarios para ingresar el nombre de modalidad y seleccionar el tipo de modalidad, además de un modal para el registro de los datos de tipo de modalidad. - En la parte inferior aparecerá una lista con todas las modalidades registradas. 	

Tabla 5.25: Historia 14 - Tarea 1. Diseño de interfaz de Gestión de modalidad
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 14
Nombre de Tarea: Desarrollo de las funciones en base de datos para Modalidad y Tipo de Modalidad	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 2/10/15	Fecha Fin: 2/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el script de las funciones registrar, modificar, listar, leer y eliminar, en la base de datos para modalidad y tipo de modalidad. - En la función Eliminar, sólo se realizará un borrado lógico de datos, teniendo en cuenta que no haya datos asociados para poder ejecutar esta función. 	

Tabla 5.26: Historia 14 - Tarea 2. Desarrollo de las funciones en base de datos para Modalidad y Tipo de Modalidad
Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 14
Nombre de Tarea: Conexión del formulario Gestión de modalidad con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 5/10/15	Fecha Fin: 5/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el código para que los datos ingresados en el formulario sean registrados y puedan ser visualizados por el usuario. - En la lista, al lado de cada modalidad aparecerán las opciones de editar datos y eliminar modalidad. 	

Tabla 5.27: Historia 14 - Tarea 3. Conexión del formulario Gestión de modalidad con la base de datos

Fuente: Elaboración propia

The screenshot shows a web application for managing modalities. The sidebar menu includes options like 'Registros', 'Personal', 'Modalidad', 'Proceso', 'Administración', and 'Reportes'. The main form allows users to register a new modality by entering a name and selecting a type (Extraordinario). It also provides options to register for a school or university. A table below the form displays existing modalities with their names, types, and action buttons for editing or deleting.

Figura 3.6: Interfaz de Gestión de modalidad
Fuente: Elaboración propia

Agregar Tipo de Modalidad

Datos

Nombre

Registrar **Limpiar**

Lista

Nombre	Accion
ORDINARIO	
EXTRAORDINARIO	
CPU	

Cerrar

Figura 3.7: Modal para agregar los tipo de modalidad
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 15
Nombre de Tarea: Diseño de interfaz de Gestión de escuela	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 6/10/15	Fecha Fin: 6/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará la interfaz con los controles necesarios para registrar y mostrar los datos de las escuelas que existen en la universidad y los grupos en que están clasificadas. - También aparecerá una lista con todas las escuelas que han sido registradas. - Se diseñará un modal para Grupos, donde se podrá registrar, editar y listar sus datos. 	

Tabla 5.28: Historia 15 - Tarea 1. Diseño de interfaz de Gestión de escuela
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 15
Nombre de Tarea: Desarrollo de las funciones en base de datos para Escuela y Grupo	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 7/10/15	Fecha Fin: 7/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el script de las funciones registrar, modificar, listar, leer y eliminar, en la base de datos para escuela y grupo. - En la función Eliminar, sólo se realizará un borrado lógico de datos, teniendo en cuenta que no haya datos asociados para poder ejecutar esta función. 	

Tabla 5.29: Historia 15 - Tarea 2. Desarrollo de las funciones en base de datos para Escuela y Grupo

Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 15
Nombre de Tarea: Conexión del formulario Gestión de escuela con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 9/10/15	Fecha Fin: 9/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se registrará en la base de datos el nombre de escuela ingresado y el grupo seleccionado, verificando que los nombres no se repitan. - En la parte inferior aparecerá la lista de Escuelas con su respectivo Grupo y las opciones de Editar y Eliminar datos. - Además un modal donde se podrá registrar, editar y eliminar los nombres de los Grupos 	

Tabla 5.30: Historia 15 - Tarea 3. Conexión del formulario Gestión de escuela con la base de datos

Fuente: Elaboración propia

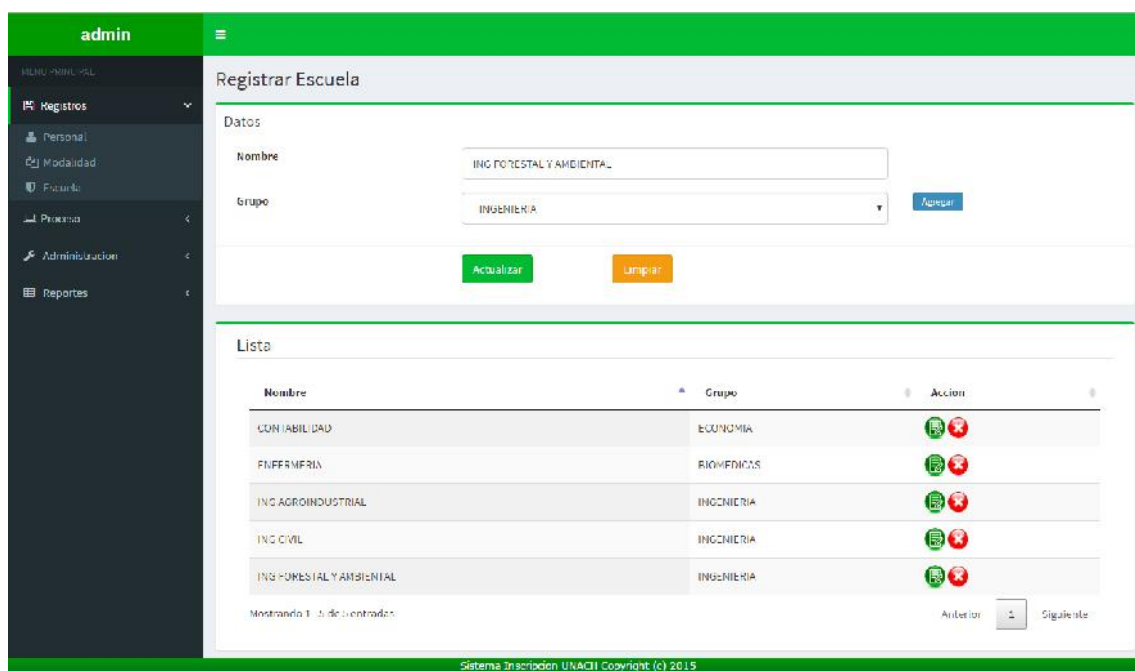


Figura 3.8: Interfaz de Gestión de escuela
Fuente: Elaboración propia

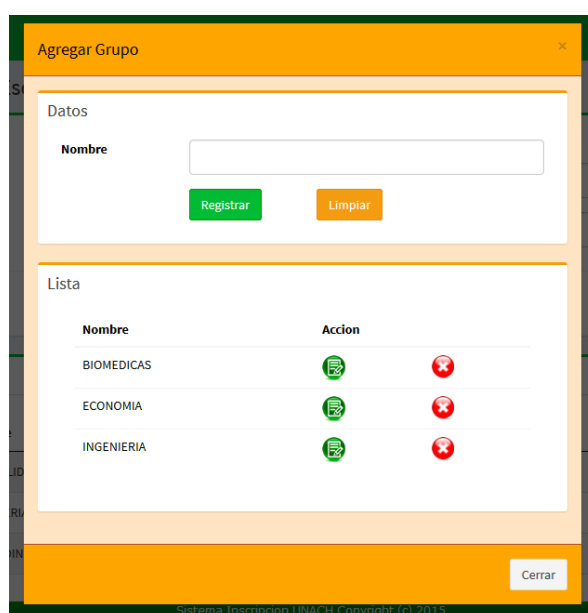


Figura 3.9: Modal para registrar grupos
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 16
Nombre de Tarea: Diseño de interfaz de Gestión de tarifario	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 12/10/15	Fecha Fin: 12/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará una interfaz que permita ingresar los datos para el registro de los tarifarios. Deberá aparecer las opciones para seleccionar la modalidad, procedencia, el monto de la tarifa e indicar si se puede realizar pagos en cuotas. 	

Tabla 5.31: Historia 16 - Tarea 1. Diseño de interfaz de Gestión de tarifario
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 16
Nombre de Tarea: Desarrollo de las funciones en base de datos para Tarifario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 13/10/15	Fecha Fin: 13/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el script de las funciones registrar, modificar, listar, leer y eliminar, en la base de datos para tarifario. - Se realizarán validaciones para evitar que una tarifa con respecto a la modalidad y procedencia se repita. 	

Tabla 5.32: Historia 16 - Tarea 2. Desarrollo de las funciones en base de datos para Tarifario
Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 16
Nombre de Tarea: Conexión del formulario Gestión de tarifario con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio:	Fecha Fin:
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el código para que los datos ingresados en el formulario sean registrados, esto incluye la modalidad, procedencia y el monto de la tarifa, la opción de cuotas es opcional. - En la parte inferior aparecerá una lista de todos los Tarifarios con sus respectivos datos y botones con la opción de editar y eliminar datos. 	

Tabla 5.33: Historia 16 - Tarea 3. Conexión del formulario Gestión de tarifario con la base de datos
Fuente: Elaboración propia

admin

ADMISSION 2015 I

Menu Principal

Registros

Principal

Modalidad

Escuela

Tarifario

Proceso de Admisión

Historia

Administración

Reportes

Registrar Tarifario

Datos

Modalidad

CPU

Procedencia

NACIONAL

Monte Total (\$.)

360.00

La tarifa está vigente

☒

Pago por cuotas

☒

N. de cuotas

2





















Monte de primera cuota

180.00

Actualizar

Cancelar

Lista

Modalidad	Procedencia	Monte Total	N. de Cuotas	Monte de Cuota	Vigencia	Acción
CAMBIO DE UNIVERSIDAD	NACIONAL	400.00	1	--	Vigente	 
CAMBIO DE UNIVERSIDAD	PARTICULAR	450.00	1	--	Vigente	 
CPU	NACIONAL	360.00	2	180.00	Vigente	 
CPU	PARTICULAR	460.00	2	230.00	Vigente	 
DEPORTISTA CALIFICADOS	NACIONAL	180.00	1	--	Vigente	 
DEPORTISTA CALIFICADOS	PARTICULAR	280.00	1	--	Vigente	 
GRADUADOS Y TITULADOS	--	600.00	1	--	Vigente	 
INGRESANTE CPU	NACIONAL	180.00	1	--	Vigente	 
INGRESANTE CPU	PARTICULAR	280.00	1	--	Vigente	 
ORDINARIO	NACIONAL	180.00	1	--	Vigente	 

Mostrando 1 - 10 de 15 entradas

Anterior

1

2

Siguiente

Sistema Inscritión UNACH Copyright (c) 2015

Figura 3.10: Interfaz de Gestión de tarifario
Fuente: Elaboración propia

Pérez Bonilla Luis Edgardo - Quintos Estrada Katherine Vanessa

94

TAREA	
Número de tarea: 1	Número de Historia: 17
Nombre de Tarea: Diseño de interfaz de Gestión de Proceso de Admisión	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 16/10/15	Fecha Fin: 16/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Cada vez que va a iniciar un proceso de admisión se deberá recurrir a este formulario para registrarlo. - Este formulario contendrá los controles necesarios para ingresar el nombre del proceso de admisión, seleccionar la fecha de inicio y de fin. - También contará con una sección para agregar los detalles del proceso de admisión, esto incluye las modalidades y las fechas límites de inscripción de cada una. 	

Tabla 5.34: Historia 17 - Tarea 1. Diseño de interfaz de Gestión de Proceso de Admisión
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 17
Nombre de Tarea: Desarrollo de las funciones en base de datos para Proceso de Admisión	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 19/10/15	Fecha Fin: 19/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el script de las funciones registrar, modificar, listar, leer y eliminar, en la base de datos para proceso de admisión. - Se validará que los nombres asignados sean únicos y que las fechas en que transcurre cada proceso no se crucen. 	

Tabla 5.35: Historia 17 - Tarea 2. Desarrollo de las funciones en base de datos para Proceso de Admisión
Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 17
Nombre de Tarea: Conexión del formulario Gestión de Proceso de Admisión con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 20/10/15	Fecha Fin: 21/10/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se ingresa el nombre, fecha de inicio y fecha de fin del proceso de admisión, luego en la sección Detalles se van agregando las modalidades de este proceso, estos detalles irán apareciendo en la parte inferior con la opción de editarlos o eliminarlos. Cuando se haya ingresado toda la información necesaria, se dará clic en el botón Registrar para que se guarde en la base de datos. - En la parte inferior aparecerá una lista con todos los procesos de admisión registrados, pero si estos procesos ya finalizaron o están en transcurso, no podrán ser editados ni eliminados. 	

Tabla 5.36: Historia 17 - Tarea 3. Conexión del formulario Gestión de Proceso de Admisión con la base de datos

Fuente: Elaboración propia

Figura 3.11: Interfaz de Gestión de proceso de admisión

Fuente: Elaboración propia

5.3.3. Tercera Iteración

TAREA	
Número de tarea: 1	Número de Historia: 1
Nombre de Tarea: Diseño de interfaz del sistema de inscripción (examen de admisión)	
Tipo de Tarea: Diseño	Puntos estimados: 1
Fecha Inicio: 22/10/15	Fecha Fin: 4/11/15
Desarrollador encargado:	
Descripción: Siempre que el postulante inicie el proceso de Inscripción deberá ingresar al sistema web que contendrá formularios para poder ingresar sus datos personales, para eso los formularios están debidamente validados.	

Tabla 5.37: Historia 1 - Tarea 1. Diseño de interfaz del sistema de inscripción (examen de admisión)
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 1
Nombre de Tarea: Diseño de interfaz del sistema de inscripción (centro pre)	
Tipo de Tarea: Diseño	Puntos estimados: 0.6
Fecha Inicio: 5/11/15	Fecha Fin: 9/11/15
Desarrollador encargado:	
Descripción: Siempre que el postulante inicie el proceso de Inscripción deberá ingresar al sistema web que contendrá formularios para poder ingresar sus datos personales, como la modalidad que postula, su documento de identidad, y su voucher, para eso los formularios están debidamente validados.	

Tabla 5.38: Historia 1 - Tarea 2. Diseño de interfaz del sistema de inscripción (centro pre)
Fuente: Elaboración propia

TAREA	
Número de tarea: 3	Número de Historia: 1
Nombre de Tarea: Desarrollo de las funciones en base de datos para Postulante	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 10/11/15	Fecha Fin: 10/11/15
Desarrollador encargado:	
Descripción: se desarrollaran los script necesarios de cada función con la finalidad de registrar, listar, leer datos que se encuentren en la base de Datos	

Tabla 5.39: Historia 1 - Tarea 3. Desarrollo de las funciones en base de datos para Postulante
Fuente: Elaboración propia

TAREA	
Número de tarea: 4	Número de Historia: 1
Nombre de Tarea: Conexión del formulario de inscripción (examen de admisión) con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 11/11/15	Fecha Fin: 24/11/15
Desarrollador encargado:	
Descripción: Se desarrollarán las clases necesarias y los servlets que nos ayuden a procesar los datos ingresados por el postulante que este bajo la modalidad ordinaria y extra ordinaria para ser registrado correctamente.	

Tabla 5.40: Historia 1 - Tarea 4. Conexión del formulario de inscripción (examen de admisión) con la base de datos
Fuente: Elaboración propia

TAREA	
Número de tarea: 5	Número de Historia: 1
Nombre de Tarea: Conexión del formulario de inscripción (centro pre) con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 25/11/15	Fecha Fin: 1/12/15
Desarrollador encargado:	
Descripción: Se desarrollarán las clases necesarias y los servlets que nos ayuden a procesar los datos ingresados por el postulante que se inscribe bajo la modalidad centro pre y sean registrados correctamente.	

Tabla 5.41: Historia 1 - Tarea 5. Conexión del formulario de inscripción (centro pre) con la base de datos

Fuente: Elaboración propia



Figura 3.12: Interfaz de sistema de inscripción de postulantes
Fuente: Elaboración propia

Figura 3.13: Datos de validación para iniciar registro de postulante
Fuente: Elaboración propia

Ficha de Inscripción UNACH



Bienvenido:

Información Básica del postulante:

Modalidad:

Apellido Paterno:

Apellido Materno:

Nombres:

Fecha de Nacimiento:

Tipo de Documento de Identidad:

Ingrese Número de Documento:

Sexo:

Ingrese Número Telefónico:

Correo electrónico:

Lugar de Nacimiento:

Departamento:

Provincia:

Distrito:

Residencia Actual:

Departamento:

Provincia:

Distrito:

Tipo de vía/Zona:

Dirección actual:

Datos del Apoderado

Teléfono del Apoderado:



Figura 3.14: Interfaz de inscripción - Información básica de postulante
Fuente: Elaboración propia

Ficha de Inscripción UNACH



Información Académica :

(*) Datos necesarios de la Institución Educativa.

Departamento:	<input type="text"/>
Provincia:	<input type="text"/>
Distrito:	<input type="text"/>
Tipo de Institución Educativa:	<input type="text"/>
Nombre de la Institución Educativa:	<input type="text"/>
Año de Egreso de la Institución Educativa:	<input type="text"/>

Información ExtraOrdinaria :

(*) Esta información solo será llenada para los que postulan bajo la modalidad de Cambio de universidad y Graduados y titulados.

Tipo de Universidad:	<input type="text"/>
Departamento:	<input type="text"/>
Universidad:	<input type="text"/>

Universidad :

Escuela a que postula:	<input type="text"/>
------------------------	----------------------



Figura 3.15: Interfaz de inscripción - Información académica
Fuente: Elaboración propia

CARNET DEL POSTULANTE

UNACH

Datos del Postulante :



Documento de Identidad:

Apellido Paterno:

Apellido Materno:

Nombres:

Escuela Profesional:

Modalidad:



Figura 3.16: Interfaz de inscripción - Verificar datos
Fuente: Elaboración propia

5.3.4. Cuarta Iteración

TAREA	
Número de tarea: 1	Número de Historia: 2
Nombre de Tarea: Diseño de interfaz de Registro de Fotos	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 2/12/15	Fecha Fin: 2/12/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Este formulario contendrá unas cajas de texto en donde se ingresará el nombre o documento de identidad del postulante para proceder a su búsqueda. - También aparecerá una sección donde se mostrarán los datos del postulante seleccionado, su foto y una caja de texto para ingresar alguna observación si es necesario. 	

*Tabla 5.42: Historia 2 - Tarea 1. Diseño de interfaz de Registro de Fotos
Fuente: Elaboración propia*

TAREA	
Número de tarea: 2	Número de Historia: 2
Nombre de Tarea: Instalación y conexión de la cámara fotográfica con el sistema	
Tipo de Tarea: Configuración	Puntos estimados: 1
Fecha Inicio: 3/12/15	Fecha Fin: 9/12/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En una computadora designada para la toma de fotos se instalará el driver incluido con la cámara fotográfica para que las imágenes puedan ser descargadas. - Se desarrollará el código para que el sistema obtenga la foto recién tomada y pueda ser visualizada en el formulario. 	

*Tabla 5.43: Historia 2 - Tarea 2. Instalación y conexión de la cámara fotográfica con el sistema
Fuente: Elaboración propia*

TAREA	
Número de tarea: 3	Número de Historia: 2
Nombre de Tarea: Guardar fotos en el sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Fecha Inicio: 10/12/15	Fecha Fin: 23/12/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En los campos correspondientes se ingresarán datos de búsqueda de un postulante como su nombre o su documento de identidad y en la lista inferior irán apareciendo los resultados. - Cuando se seleccione un postulante de la lista, aparecerán sus datos en un cuadro ubicado en la parte inferior, se toma la foto al postulante y se obtiene esta imagen mediante el botón “Mostrar Foto”, se ingresará una observación de la foto si es necesario y se dará clic en el botón Registrar. - El sistema guardará la información en la base de datos y se encargará de recortar y redimensionar la foto a un tamaño estándar para luego guardarla en una carpeta interna del sistema y una copia en el disco local C. - En base de datos se guardarán datos referentes a la foto, como el identificador de postulante, el usuario que tomó la foto, la fecha de registro y una observación. - La foto no será guardada en la base de datos. - Si la foto de un postulante ya ha sido registrada, se mostrará en el sistema y la opción de registro será deshabilitada para evitar duplicados o incoherencias en los datos. - Si no existe un proceso de admisión activo no se podrá realizar la toma de fotos, por consiguiente no habrán postulantes y para indicarlo en el formulario aparecerá un mensaje de error. 	

Tabla 5.44: Historia 2 - Tarea 3. Guardar fotos en el sistema
Fuente: Elaboración propia

kquintos ADMISIÓN 2016 I

Numero de Documento: /3 Nombre / Apellidos:

Seleccione un postulante de la lista para registrar su foto

Código	Nombre	Tipo de Documento	Numero	Acción
0560004	BUSTAMANTE COLUNCHIE DEYSI ROSANA	DNI	73460652	
0400002	CAPUZA VASQUEZ ROSSANA ESTHER	DNI	761054097	
0560010	CAMPOS RIOS CI FVER	DNI	47372647	
0560011	CAMPOS TAAAY DILMER	DNI	75237777	
0560008	TARRILLO CIFUENA SILVIA	DNI	47325912	

Datos de Postulante

Mostrar Foto

Datos de Postulante

Código: 0560004
 Apellidos: BUSTAMANTE COLUNCHIE
 Nombres: DEYSI ROSANA
 Documento: DNI 73460652
 Escuela: CONTABILIDAD

Observación:

Registrar **Limpiar**

Sistema Inscripción UNACH Copyright (c) 2015

Figura 3.17: Interfaz de Registrar foto
Fuente: Elaboración propia

La carpeta predeterminada para guardar las fotos que tome la cámara fotográfica tiene la siguiente ruta: C:\SistemaUnach\FotosCamara. Cuando un usuario seleccione la opción “Mostrar Foto” el sistema obtendrá la imagen de la dirección antes mencionada para mostrarla en el formulario.

La carpeta predeterminada para guardar las copias de las fotos registradas tiene la siguiente ruta: C:\SistemaUnach\FyH\Fotos y dentro de esta se irán creando carpetas con el nombre del proceso de admisión, cada archivo tendrá como nombre el código del postulante al que pertenece.

Tabla 5.45: Interfaz de registrar foto cuando no existe un proceso de admisión activo
Fuente: Elaboración propia

5.3.5. Quinta Iteración

TAREA	
Número de tarea: 1	Número de Historia: 3
Nombre de Tarea: Diseño del applet para Registro de Huellas	
Tipo de Tarea: Diseño	Puntos estimados: 0.4
Fecha Inicio: 28/12/15	Fecha Fin: 29/12/15
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Para el registro de huellas se creará un applet que nos permitirá obtener las imágenes enviadas por el huellero. - En este applet aparecerán unas cajas de texto en donde se ingresará el nombre o documento de identidad del postulante para proceder a su búsqueda. - También aparecerá una sección donde se mostrarán los datos del postulante seleccionado, su huella, una caja de texto para ingresar alguna observación si es necesario y un área en donde aparecerán mensajes que nos indicarán sobre el estado del huellero (si ya está conectado, si ha tomado la huella, etc). 	

Tabla 5.46: Historia 3 - Tarea 1. Diseño del applet para Registro de Huellas
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 2
Nombre de Tarea: Instalación y conexión del huellero con el sistema	
Tipo de Tarea: Configuración	Puntos estimados: 4
Fecha Inicio: 30/12/15	Fecha Fin: 28/01/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En una computadora designada para la captura de huellas se instalará el driver incluido con el huellero. - Se desarrollará el código para que el sistema obtenga la huella capturada por el huellero y se visualice en el applet. 	

*Tabla 5.47: Historia 2 - Tarea 2. Instalación y conexión del huellero con el sistema
Fuente: Elaboración propia*

TAREA	
Número de tarea: 3	Número de Historia: 2
Nombre de Tarea: Guardar imágenes de huellas en el sistema	
Tipo de Tarea: Desarrollo	Puntos estimados: 4
Fecha Inicio: 29/01/16	Fecha Fin: 25/02/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En los campos correspondientes se ingresarán datos de búsqueda de un postulante como su nombre o su documento de identidad y en una tabla irán apareciendo los resultados. - Cuando se seleccione un postulante de la lista, aparecerán sus datos en un cuadro ubicado en la parte inferior, se captura la huella del postulante, se ingresa una observación de la huella si es necesario y se dará clic en el botón Registrar. - El sistema guardará la información en la base de datos y se encargará de recortar y redimensionar la huella a un tamaño estándar para luego guardarla en una carpeta interna del sistema y una copia en el disco local C. 	

- Si la huella de un postulante ya ha sido registrada, se mostrará en el sistema y la opción de registro será deshabilitada para evitar duplicados o incoherencias en los datos.
- En base de datos se guardarán datos referentes a la huella, como el identificador de postulante, el usuario que registró la huella, la fecha de registro y una observación.
- La imagen de la huella no será guardada en la base de datos.
- Si no existe un proceso de admisión activo no se podrá realizar la captura de huellas y en lugar del applet aparecerá un mensaje de error.

Tabla 5.48: Historia 2 - Tarea 3. Guardar imágenes de huellas en el sistema
Fuente: Elaboración propia

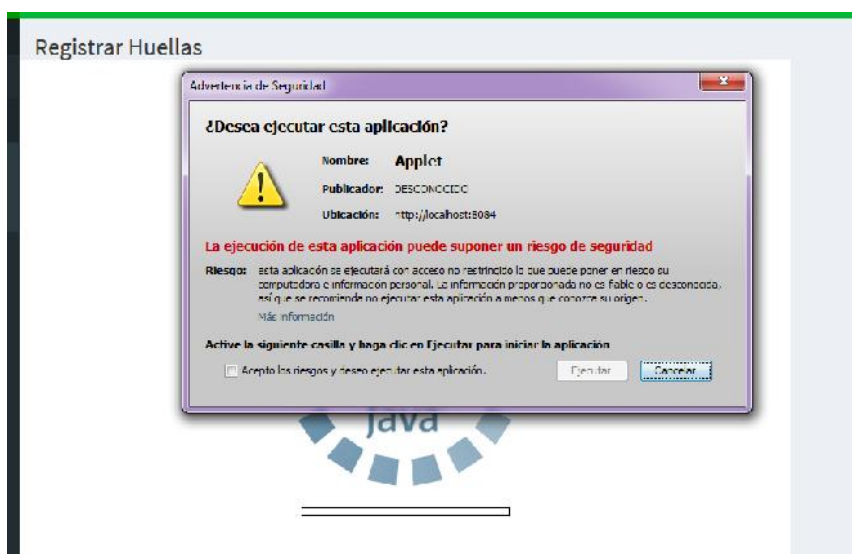


Figura 3.18: Mensaje de seguridad cuando se accede al applet
Fuente: Elaboración propia

Como un applet se ejecuta localmente, van a existir ciertos riesgos de seguridad para esta máquina cliente, por eso los navegadores restringen su acceso; para evitar esto debemos brindarle los permisos necesarios (ver anexo D) para que considere a nuestro applet como una aplicación segura. Cuando se ingresa al formulario de Registrar Huellas, nos aparecerá un mensaje de seguridad de Java, aún con los permisos concedidos siempre nos avisará que una aplicación desconocida está intentando ejecutarse, activamos la casilla “Acepto los riesgos y deseo ejecutar esta aplicación”, clic en el botón Ejecutar y ya tendremos acceso al applet.

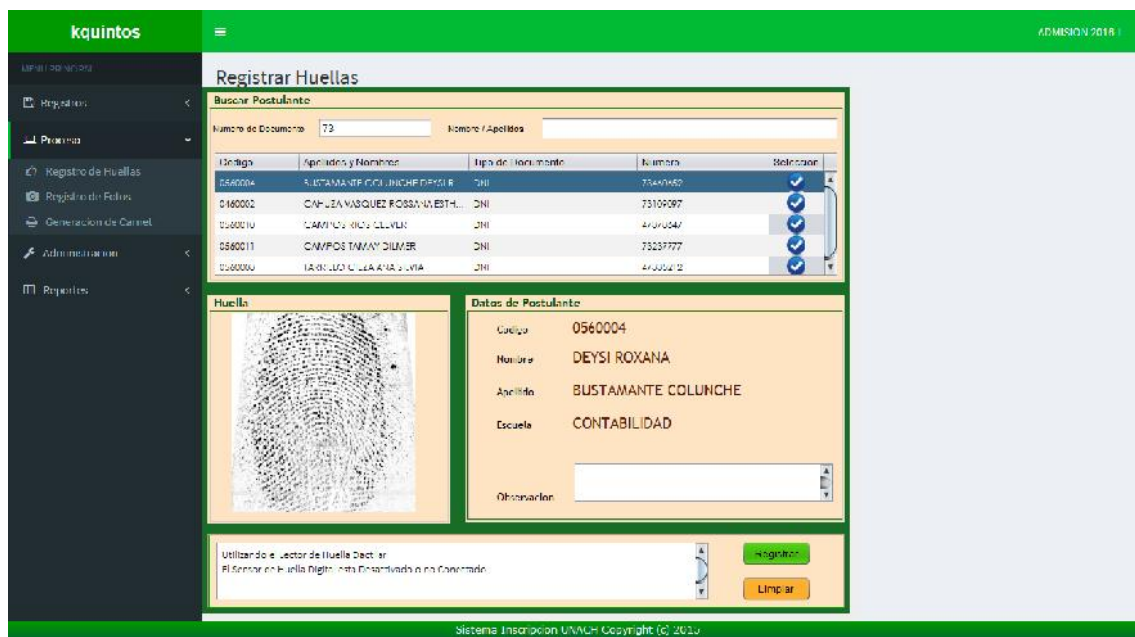


Figura 3.19: Applet para registrar huellas
Fuente: Elaboración propia

La carpeta predeterminada para guardar las copias de las huellas registradas tiene la siguiente ruta: C:\SistemaUnach\FyH\Huellas y dentro de esta se irán creando carpetas con el nombre del proceso de admisión, cada archivo tendrá como nombre el código del postulante al que pertenece.



Figura 3.20: Interfaz de registrar huella cuando no existe un proceso de admisión activo
Fuente: Elaboración propia

5.3.6. Sexta Iteración

TAREA	
Número de tarea: 1	Número de Historia: 4
Nombre de Tarea: Diseño de interfaz de Verificar datos de postulante	
Tipo de Tarea: Diseño	Puntos estimados: 0.4
Fecha Inicio: 26/02/16	Fecha Fin: 29/02/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Este formulario contendrá unas cajas de texto en donde se ingresará el nombre o documento de identidad del postulante para proceder a su búsqueda. - También aparecerá una sección en donde se mostrarán todos los datos del postulante seleccionado, así como su foto y su huella. - Este formulario tendrá la opción de modificar los datos del postulante e imprimir su carnet. 	

*Tabla 5.49: Historia 4 - Tarea 1. Diseño de interfaz de Verificar datos de postulante
Fuente: Elaboración propia*

TAREA	
Número de tarea: 2	Número de Historia: 4
Nombre de Tarea: Desarrollo de las funciones en base de datos para Postulante	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 1/03/16	Fecha Fin: 1/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se desarrollará el script de la funciones modificar y leer datos de postulante para permitir que se guarden los cambios efectuados en la base de datos. - Todos los datos del postulante pueden ser modificados con esta función, excepto su escuela y la modalidad a la que postula. 	

*Tabla 5.50: Historia 4 - Tarea 2. Desarrollo de las funciones en base de datos para Postulante
Fuente: Elaboración propia*

TAREA	
Número de tarea: 3	Número de Historia: 4
Nombre de Tarea: Conexión del formulario Verificar Datos de Postulante con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 2/03/16	Fecha Fin: 7/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En los campos correspondientes se ingresarán datos de búsqueda de un postulante como su nombre o su documento de identidad y en una tabla irán apareciendo los resultados. - Cuando se seleccione un postulante de la lista, aparecerán sus datos en una sección ubicada en la parte inferior, se pedirá al postulante que revise sus datos y si existe algún error se seleccionará la opción Modificar para corregirlos. - Si se selecciona la opción Modificar, aparecerá un modal que permitirá editar todos los datos del postulante. El usuario encargado también podrá ingresar alguna observación, en el campo correspondiente, si lo considera necesario. - Si no existe un proceso de admisión activo, aparecerá un mensaje de error en el formulario. 	

Tabla 5.51: Historia 4 - Tarea 3. Conexión del formulario Verificar Datos de Postulante con la base de datos

Fuente: Elaboración propia

Modificar Postulante

Datos Personales

Nombres	Apellido Paterno	Apellido Materno
DEYSI ROXANA	BUSTAMANTE	COLUNCHE

Tipo de Documento	Numero de Documento	Sexo	Fecha de Nacimiento
DNI	73460652	Femenino	05/05/2009

Telefono Postulante	Telefono Apoderado	Correo
976700114	976700114	DEYSI_05@HOTMAIL.COM

Lugar de Nacimiento

Departamento	Provincia	Distrito
LAMBAYEQUE	CHICLAYO	LA VICTORIA

Domicilio

Tipo de Direccion	Direccion
AVENIDA	TODOS LOS SANTOS

Departamento	Provincia	Distrito
CAJAMARCA	CHOTA	CHOTA

Colegio

Departamento	Provincia	Distrito
LAMBAYEQUE	CHICLAYO	CHICLAYO

Procedencia	Egreso	Colegio
PRIVADA	2015	BEATA IMELDA

Observacion

Modificar

Cerrar

Sistema Inscripción ONACH Copyright (c) 2015

Figura 3.22: Modal para modificar datos de postulante
Fuente: Elaboración propia

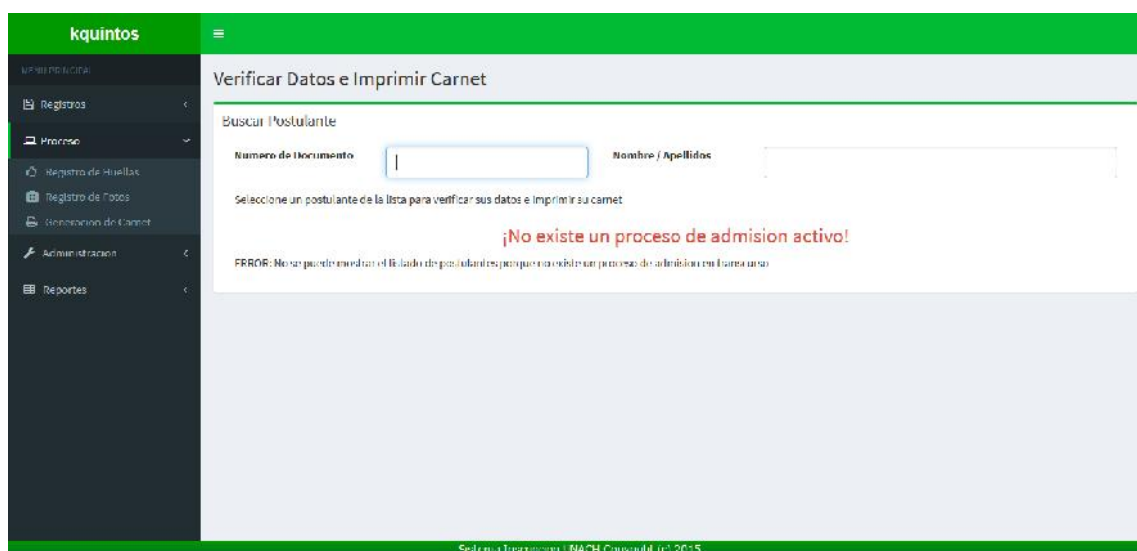


Figura 3.23: Interfaz de Verificar datos de postulante cuando no existe un proceso de admisión activo

Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 5
Nombre de Tarea: Diseño del carnet	
Tipo de Tarea: Diseño	Puntos estimados: 0.4
Fecha Inicio: 8/03/16	Fecha Fin: 9/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará el carnet que será entregado al postulante como constancia de su registro en el proceso de admisión. - Este carnet contendrá los datos más importantes como su nombre, apellidos, número de documento de identidad, escuela, modalidad y proceso de admisión en el que se encuentra registrado, así también aparecerá su foto y su huella. 	

Tabla 5.52: Historia 5 - Tarea 1. Diseño del carnet

Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 5
Nombre de Tarea: Desarrollar la función para la impresión de carnet	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 10/03/16	Fecha Fin: 11/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Cuando el postulante ha verificado que todos sus datos son correctos y están completos (incluida su foto y huella) se seleccionará la opción de Imprimir Carnet. 	

*Tabla 5.53: Historia 5 - Tarea 2. Desarrollar la función para la impresión de carnet
Fuente: Elaboración propia*



*Figura 3.24: Carnet de postulante
Fuente: Elaboración propia*

5.3.7. Séptima Iteración

TAREA	
Número de tarea: 1	Número de Historia: 11
Nombre de Tarea: Diseño de interfaz de Iniciar Sesión	
Tipo de Tarea: Diseño	Puntos estimados: 0.4
Fecha Inicio: 14/03/16	Fecha Fin: 15/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará la interfaz de bienvenida al sistema. Esta contiene los controles necesarios para que el personal que accede al sistema ingrese su usuario y contraseña. 	

*Tabla 5.54: Historia 11 - Tarea 1. Diseño de interfaz de Iniciar Sesión
Fuente: Elaboración propia*

TAREA	
Número de tarea: 2	Número de Historia: 11
Nombre de Tarea: Conexión del formulario Iniciar Sesión con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 16/03/16	Fecha Fin: 16/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Una vez que se ha ingresado el usuario y contraseña en el formulario, el sistema validará que estos datos sean correctos, si es así, se le permitirá el acceso al sistema, sino aparecerá un mensaje de error indicando que la información ingresada es incorrecta. - En base de datos se guardará toda la información de los intentos de acceso al sistema, tanto válidos como inválidos. - Si un personal ha sido registrado, pero su vigencia en el sistema está inhabilitada, se le negará el acceso. 	

*Tabla 5.55: Historia 11 - Tarea 2. Conexión del formulario Iniciar Sesión con la base de datos
Fuente: Elaboración propia*



Figura 3.25: Interfaz de sistema administrativo - Iniciar sesión
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 12
Nombre de Tarea: Diseñar interfaz de Perfil de Usuario	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 17/03/16	Fecha Fin: 17/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - El perfil de usuario mostrará la información más importante del usuario que ha accedido al sistema. - El perfil también contiene una opción para modificar la contraseña. 	

Tabla 5.56: Historia 12 - Tarea 1. Diseñar interfaz de Perfil de Usuario
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 12
Nombre de Tarea: Conexión del formulario Perfil de Usuario con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 18/03/16	Fecha Fin: 18/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Cuando un usuario ingresa al sistema, el primer formulario que aparecerá será su Perfil de Usuario, aquí aparecerán sus datos personales, pero no los podrá modificar, excepto su contraseña. - Si se selecciona la opción “Cambiar Contraseña”, aparecerá un modal que le permitirá asignar una contraseña nueva, para ello tiene que ingresar su contraseña antigua y la nueva, el sistema validará que los datos sean correctos y realizará los cambios. - En la esquina superior derecha del formulario se mostrará el nombre del proceso de admisión en transcurso, para tenerlo como referencia, y un botón para cerrar sesión. Estos datos estarán visibles en todos los formularios. 	

Tabla 5.57: Historia 12 - Tarea 2. Conexión del formulario Perfil de Usuario con la base de datos
Fuente: Elaboración propia

Figura 3.26: Perfil de usuario
Fuente: Elaboración propia

Figura 3.27: Modal para modificar contraseña de usuario
Fuente: Elaboración propia

Figura 3.28: Información que aparece en la cabecera de cada formulario
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 13
Nombre de Tarea: Diseño de la interfaz Asignar Permisos	
Tipo de Tarea: Diseño	Puntos estimados: 0.6
Fecha Inicio: 22/03/16	Fecha Fin: 24/03/16
Desarrollador encargado:	
Descripción: - Este formulario contendrá una opción para seleccionar un rol de usuario y una lista con todas las opciones del menú del sistema.	

Tabla 5.58: Historia 13 - Tarea 1. Diseño de la interfaz Asignar Permisos
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 13
Nombre de Tarea: Conexión del formulario Asignar Permisos con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 25/03/16	Fecha Fin: 28/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En el formulario se selecciona un rol de usuario y en la lista de opciones del menú se activa la casilla de aquellos formularios a los que tendrá acceso. Clic en el botón Registrar para guardar los cambios. - En caso un usuario intente ingresar a un formulario al que no se le permita el acceso, le aparecerá una página con el error de “Acceso Denegado”. 	

Tabla 5.59: Historia 13 - Tarea 2. Conexión del formulario Asignar Permisos con la base de datos
Fuente: Elaboración propia

Figura 3.29: Interfaz de Asignar permisos
Fuente: Elaboración propia

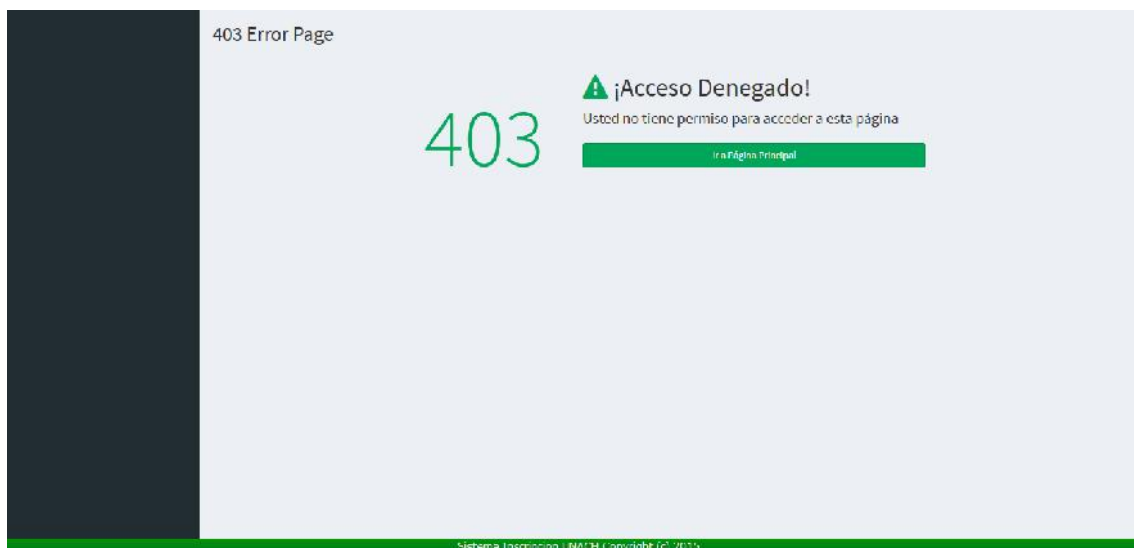


Figura 3.30: Interfaz de acceso denegado
Fuente: Elaboración propia

Octava Iteración

TAREA	
Número de tarea: 1	Número de Historia: 6
Nombre de Tarea: Diseño de interfaz de Cambio de modalidad (De extraordinario a ordinario)	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 29/03/16	Fecha Fin: 29/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará la interfaz con una caja de texto para ingresar el código del postulante que va a realizar cambio de modalidad (De extraordinario a ordinario). - En una sección del formulario aparecerán los datos del postulante junto a su foto y huella. 	

Tabla 5.60: Historia 6 - Tarea 1. Diseño de interfaz de Cambio de modalidad (De extraordinario a ordinario)

Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 6
Nombre de Tarea: Conexión del formulario Cambio de modalidad (De extraordinario a ordinario) con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 30/03/16	Fecha Fin: 31/03/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En la caja de texto correspondiente se ingresará el código del postulante que va a cambiar de modalidad, clic en el botón Buscar y si el dato es correcto, aparecerán los datos del postulante. - Si los datos del postulante están completos, incluida su foto y huella, se activará el botón de cambiar modalidad, al hacer clic en este botón se modificará en base de datos la modalidad a ordinario y el sistema procederá a imprimir un nuevo carnet. 	

Tabla 5.61: Historia 6 - Tarea 2. Conexión del formulario Cambio de modalidad (De extraordinario a ordinario) con la base de datos
Fuente: Elaboración propia

Figura 3.31: Interfaz de cambio de modalidad (de extraordinario a ordinario)
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 7
Nombre de Tarea: Diseño de Interfaz de Cambio de modalidad (De CPU a Ingresante CPU)	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 1/04/16	Fecha Fin: 1/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se diseñará la interfaz con una caja de texto para ingresar el código del postulante que va a cambiar de modalidad (De CPU a Ingresante CPU). - En una sección del formulario aparecerán los datos del postulante junto a su foto y huella. 	

Tabla 5.62: Historia 7 - Tarea 1. Diseño de Interfaz de Cambio de modalidad (De CPU a Ingresante CPU)

Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 7
Nombre de Tarea: Conexión del formulario Cambio de modalidad (De CPU a Ingresante CPU) con la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 4/04/16	Fecha Fin: 5/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En la caja de texto correspondiente se ingresará el código del postulante que va a cambiar de modalidad, clic en el botón Buscar y si el dato es correcto, aparecerán los datos del postulante. - Se ingresa el número del documento de pago del postulante y si sus datos están completos, incluida su foto y huella, se activará el botón de cambiar modalidad, al hacer clic en este botón se volverá a registrar al postulante con su nueva modalidad y generándole otro código, después de esto el sistema procederá a imprimir un nuevo carnet. 	

Tabla 5.63: Historia 7 - Tarea 2. Conexión del formulario Cambio de modalidad (De CPU a Ingresante CPU) con la base de datos

Fuente: Elaboración propia

Figura 3.32: Interfaz de Cambio de modalidad (De CPU a Ingresante CPU)
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 8
Nombre de Tarea: Diseño de interfaz Lista de postulantes y conexión a base de datos	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 6/04/16	Fecha Fin: 6/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Para este reporte el formulario contendrá las opciones de proceso de admisión, escuela y modalidad y de acuerdo a estos datos listar a los postulantes. 	

Tabla 5.64: Historia 8 - Tarea 1. Diseño de interfaz Lista de postulantes y conexión a base de datos
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 8
Nombre de Tarea: Mostrar datos en el formulario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Fecha Inicio: 7/04/16	Fecha Fin: 7/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - Se seleccionan los parámetros de búsqueda: proceso de admisión, modalidad o escuela, clic en el botón Buscar y el sistema se encargará de enviar esta información a la base de datos y mostrar los resultados ordenados en una tabla. 	

Tabla 5.65. Historia 8 - Tarea 2. Mostrar datos en el formulario
Fuente: Elaboración propia

k Quintos		ADMISION 2016-I					Salir
MENU PRINCIPAL		Lista de Postulantes					
<ul style="list-style-type: none"> Registros Proceso Administración Reportes Lista de Postulantes Lista Modalidad/Escuela 		Realizar búsqueda por Proceso de Admisión: ADMISION 2016-I Modalidad: Mostrar todos Escuela: Mostrar todos <input type="button" value="Buscar"/>					
Codigo	Apellidos y Nombres	DNI	Telefono	Correo	Modalidad	Escuela	
0560009	BENAVIDES IRIGOIN, JOSE IVAN	78020577	956674046	UNACH.ADMISION@HOTMAIL.COM	PRIMEROS PUESTOS	CONTABILIDAD	
0560004	BUSTAMANTE COLUNCHE, DEYSI ROXANA	73460652	976700114	DEYSI_05@HOTMAIL.COM	DEPORTISTAS CALIFICADOS	CONTABILIDAD	
0460002	CAHUZA VASQUEZ, ROSSANA ESTHER	73109097	956674046	UNACH.ADMISION@HOTMAIL.COM	GRADUADOS Y TITULADOS	ENFERMERIA	
0560010	CAMPOS RIOS, CLEVER	47378647	956674046	UNACH.ADMISION@HOTMAIL.COM	CAMBIO DE UNIVERSIDAD	CONTABILIDAD	
0560011	CAMPOS TAMAY, DILMER	73237777	956674046	UNACH.ADMISION@HOTMAIL.COM	CAMBIO DE UNIVERSIDAD	CONTABILIDAD	
0560005	GONZALEZ ARCELES, EDUAR ALBERTO	70202016	976700114	LUPEBOCLACAZA16@HOTMAIL.COM	DEPORTISTAS CALIFICADOS	CONTABILIDAD	
0560003	GONZALEZ HUAMAN, TREICY DAYANNE	74236834	948731872	LUPEBOCLACAZA16@HOTMAIL.COM	CAMBIO DE UNIVERSIDAD	CONTABILIDAD	
0560006	QUINTOS ESTRADA, KATHERINE VANESSA	00000000	979494145	ADMISION@UNACH.COM	CAMBIO DE UNIVERSIDAD	CONTABILIDAD	
0460003	QUINTOS ESTRADA, PAOLA	74940565	979494145	ADMISION.UNACH@HOTMAIL.COM	CAMBIO DE UNIVERSIDAD	ENFERMERIA	
0460001	SUXE QUIROZ, YANET	71910080	956674046	UNACH.ADMISION@HOTMAIL.COM	CAMBIO DE UNIVERSIDAD	ENFERMERIA	
Mostrando 1 - 10 de 14 entradas							Anterior 1 2 Siguiente
Sistema Inscripción UNACH Copyright (c) 2015							

Tabla 5.66: Reporte - Lista de postulantes
Fuente: Elaboración propia

TAREA	
Número de tarea: 1	Número de Historia: 9
Nombre de Tarea: Diseño de interfaz de Postulantes por modalidad y escuela y conexión a base de datos	
Tipo de Tarea: Diseño	Puntos estimados: 0.2
Fecha Inicio: 8/04/16	Fecha Fin: 8/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - El formulario se dividirá en tres secciones para mostrar una lista con el total de postulantes según su modalidad, escuela y por ambos datos. 	

Tabla 5.67: Historia 9 - Tarea 1. Diseño de interfaz de Postulantes por modalidad y escuela y conexión a base de datos
Fuente: Elaboración propia

TAREA	
Número de tarea: 2	Número de Historia: 9
Nombre de Tarea: Mostrar datos en el formulario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 11/04/16	Fecha Fin: 12/04/16
Desarrollador encargado:	
Descripción: <ul style="list-style-type: none"> - En cada sección del formulario de búsqueda se selecciona un proceso de admisión, clic en el botón Buscar y se mostrará una lista con el total de postulantes agrupados según la sección seleccionada. - Seguido de la lista se mostrará un cuadro estadístico para tener una información más detallada de los resultados obtenidos. 	

Tabla 5.68: Historia 9 - Tarea 2. Mostrar datos en el formulario
Fuente: Elaboración propia

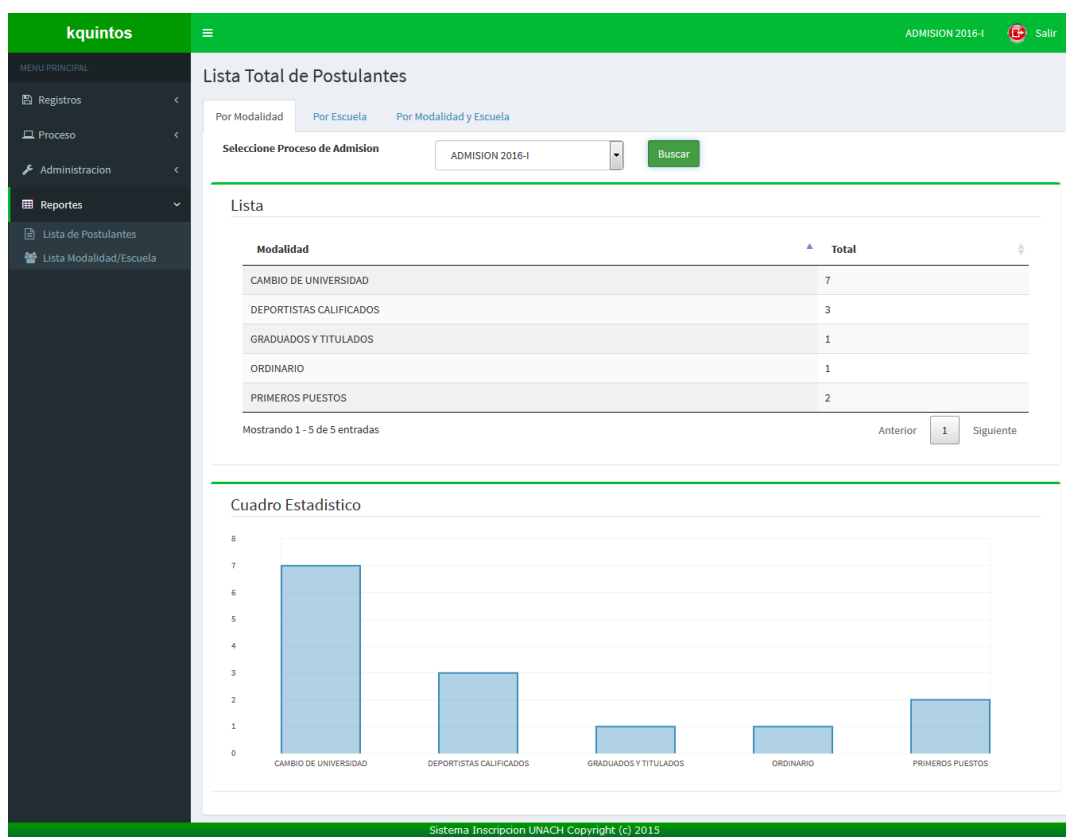


Figura 3.33: Reporte - Total de postulantes por modalidad
Fuente: Elaboración propia

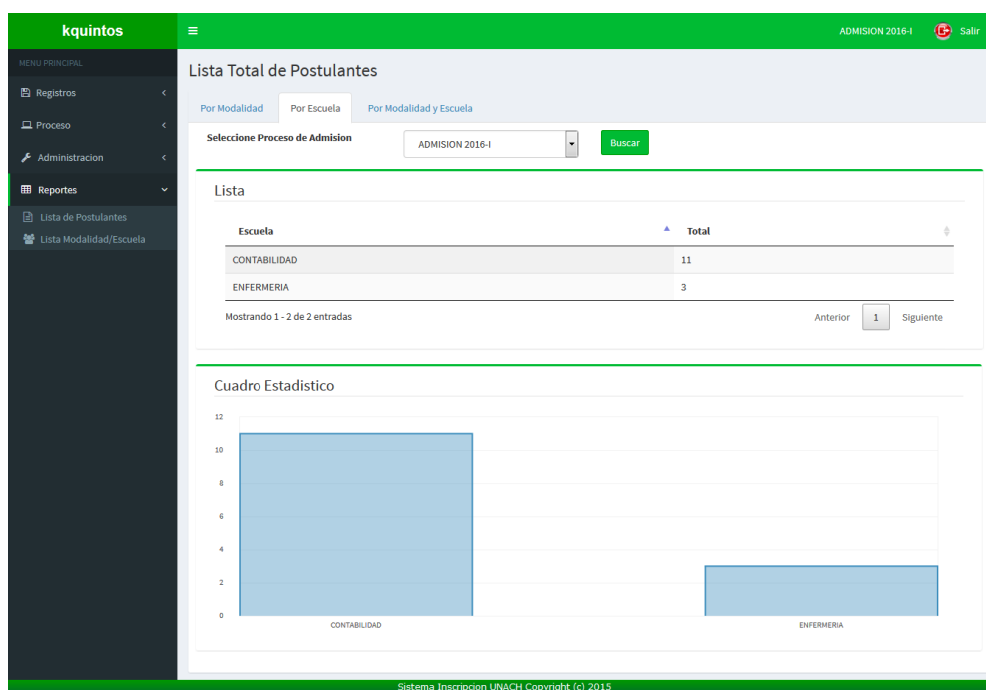


Figura 3.34: Reporte - Total de postulantes por escuela
Fuente: Elaboración propia

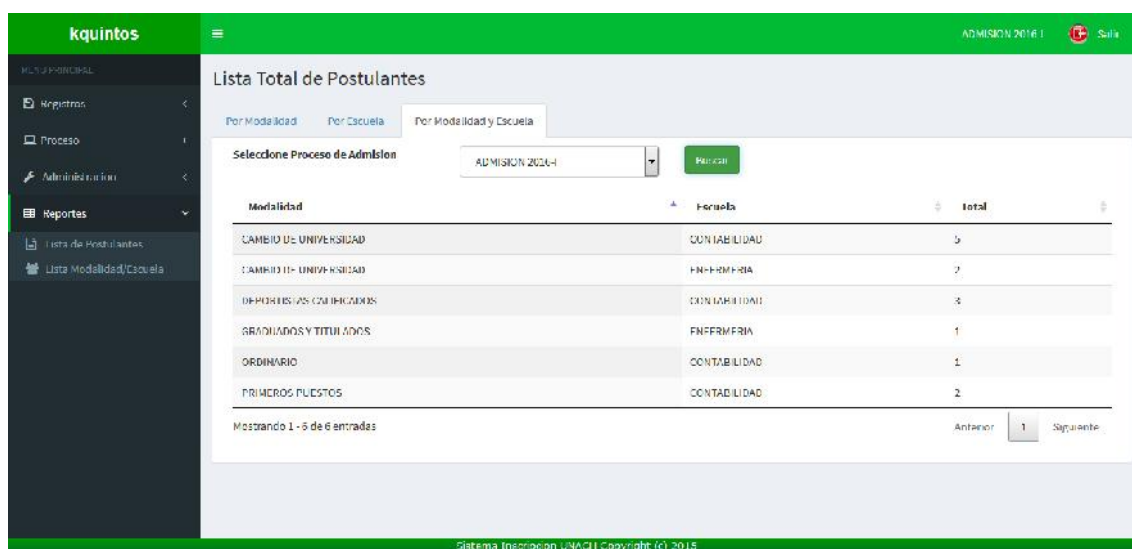


Figura 3.35: Reporte - Total de postulantes por modalidad y escuela
Fuente: Elaboración propia

5.4. Fase 4: Producción

Finalizada la fase de iteraciones es momento de integrar el sistema y comprobar su correcto funcionamiento.

Simulamos el entorno de trabajo utilizando dos computadoras conectadas a una red local. Una de ellas cumplirá la función de servidor y desde la otra accederemos a los sistemas (de inscripción y administrativo).

Utilizaremos Apache Tomcat como contenedor web, lo instalamos y configuramos en el servidor (ver Anexo B). Luego de realizar estos procedimientos ya podemos desplegar el archivo .war de los sistemas. En la otra computadora ingresamos al sistema y realizamos las pruebas respectivas, verificando que cada formulario cumpla con todas sus funciones.

Efectuado este proceso, realizamos una actualización en los sistemas, mejoramos interfaces y corregimos errores, especialmente en el applet para capturar huellas, se tuvo problemas al momento de requerir información a la base datos, para solucionarlo se configuraron los permisos para acceder a postgres (ver anexo D3).

Una vez finalizado, ya se puede realizar la entrega de los sistemas al cliente.

CAPITULO VI

PRUEBAS DEL USO DEL SISTEMA



6. PRUEBAS DE USO DEL SISTEMA

6.1. Objetivos

Las pruebas del software se elaboran con el fin de especificar que elemento o componentes serán probados, para que el grupo de trabajo pueda realizar el proceso de validación y verificación de los requerimientos funcionales y no funcionales.

Desarrollando las pruebas del sistema podemos obtener información sobre los errores, defectos o fallas que tienes el prototipo para que posteriormente se realicen las correcciones según sea el caso y poder garantizar un software de calidad

6.2. Estrategias de Prueba

Tomar información de la misma universidad para garantizar calidad del sistema a entregar. Además le permite al responsable de las pruebas saber exactamente los criterios que se deben tener en cuenta para probar cada elemento del sistema.

6.3. Actividades a realizar

Las principales actividades de prueba a realizar están asociadas con las tareas de:

6.3.1. Pruebas Unitarias

Se probaron y verificaron las definiciones de las clases desarrolladas en lenguaje java y se llegó a la conclusión de que los campos empleados están conforme Para esta prueba se utilizó la herramienta Netbeans 8.

Estas pruebas se realizaron al finalizar cada iteración.

6.3.2. Pruebas de Integración

Las pruebas de integración entre las capas Bean, Dao y Lógica se encuentran enlazadas y conectadas dando como

resultado una buena integración entre sus componentes
Para esta prueba se utilizó la herramienta Netbeans 8.
Esta prueba se realizó durante la fase de producción.

6.3.3. Pruebas de Implantación

Las pruebas de Implantación corresponden a las verificaciones necesarias para asegurar que el sistema software funciona correctamente en el entorno de trabajo, las validaciones en las cajas de texto, registros, actualizaciones y asignaciones son correctas y apropiadas para el uso del software.

6.3.4. Pruebas de Aceptación

Al momento de ser entregado, se realizaron estas pruebas, las cuales fueron realizadas por el encargado de la oficina de admisión, encargado de la unidad de soporte Informático y Centro Pre Universitario de la Universidad Nacional Autónoma De chota.

Los documentos a entregar con los sistemas son:

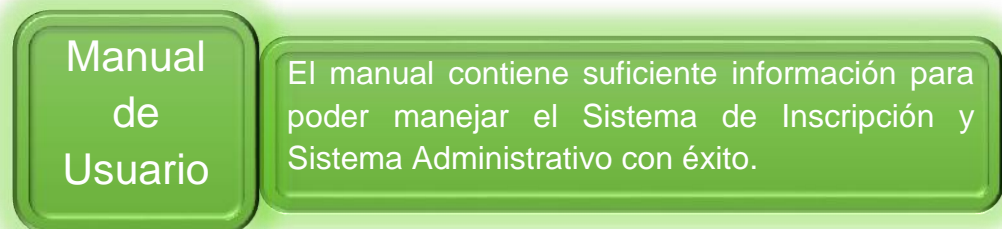


Figura 3.1: Manual de usuario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 1	N° Historia de Usuario: 1
Nombre Caso de Prueba: Registro de Postulante con datos erróneos.	
Nombre Historia de Usuario: Inscripción de Postulante.	
Descripción: El postulante ingresa a la página web de inscripción e ingresará los datos de su número de voucher, su dni y la modalidad a la que postula, el sistema verifica esta información en la base de datos y si no son correctos aparecerá un mensaje de error indicando la razón por la cual no se permite el registro.	
Condiciones de ejecución: Ninguno.	
Entradas: <ul style="list-style-type: none"> - El postulante ingresa al sistema de inscripción y en la página de bienvenida presiona el botón “Inscríbete”, esto nos dirigirá a otra página con un formulario para ingresar datos. - En este formulario el postulante ingresará en los cuadros de texto correspondientes, el número de su voucher (del pago realizado en el banco), el número de su dni u otro documento de identidad y la modalidad a la que postula. - El sistema verifica los datos ingresados y si son incorrectos muestra un mensaje de error. 	
Resultado esperado: El sistema no permite el ingreso a los formularios de registro si los datos ingresados son incorrectos o el postulante ya se ha registrado, mostrando un mensaje con el error.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.1: Prueba de Aceptación 1. Registro de Postulante con datos erróneos.
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 2	N° Historia de Usuario: 1
Nombre Caso de Prueba: Registro de Postulante	
Nombre Historia de Usuario: Inscripción de Postulante	
Descripción: El postulante ingresa a la página web de inscripción e ingresará los datos de su número de voucher, su dni y la modalidad a la que postula, el sistema verifica que los datos sean correctos y le permite el acceso a unos formularios para que registre sus datos personales, el colegio o universidad donde estudió y la carrera a la que postula, al finalizar aparecerá un resumen de los datos ingresados, presiona el botón de confirmación y se registran los datos.	
Condiciones de ejecución: Ninguno.	
Entradas: <ul style="list-style-type: none"> - El postulante ingresa al sistema de inscripción y en la página de bienvenida presiona el botón “Inscríbete”, esto nos dirigirá a otra página con un formulario para ingresar datos. - En este formulario el postulante ingresará en los cuadros de texto correspondientes, el número de su voucher (del pago realizado en el banco), el número de su dni u otro documento de identidad y la modalidad a la que postula. - El sistema verifica que los datos ingresados son correctos y permite el acceso al formulario de inscripción. - En el primer formulario de registro se ingresan los datos personales como el apellido paterno, apellido materno, nombres, fecha de nacimiento, tipo de documento de identidad, sexo, teléfono, correo electrónico, lugar de nacimiento, datos de su residencia actual y el teléfono de su apoderado, en el número de su documento de identidad el sistema lo toma del formulario anterior y no permite cambiarlo. Cuando se han llenado estos campos, se presiona el botón “Siguiente”. - En el segundo formulario se ingresan los datos de su formación académica, dependiendo de la modalidad a la que postula se mostrará los campos correspondientes para que selecciona el colegio o la universidad de la que proviene y la escuela a la que postula. Luego se presiona el botón “Siguiente”. - En el tercer formulario aparecerá un resumen de los datos ingresados anteriormente, se presiona el botón “Finalizar inscripción” y se direcciona a la página de bienvenida para comenzar un nuevo registro. 	

Resultado esperado: La información ingresada por el postulante se registra en la base de datos.
Evaluación de la prueba: Prueba satisfactoria.

*Tabla 6.2: Prueba de Aceptación 2. Registro de Postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 3	N° Historia de Usuario: 2
Nombre Caso de Prueba: Registrar foto de postulante	
Nombre Historia de Usuario: Registro de Fotos	
Descripción: El usuario con rol de Fotos ingresa al formulario y realiza la búsqueda de un postulante por su nombre o dni, se toma la foto de este postulante y el sistema captura la imagen, de ser necesario se agrega una observación y se registra la información en la base de datos, excepto la foto, que será guardada en una carpeta del sistema y una copia en el disco local del servidor.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Fotos.	
Entradas: <ul style="list-style-type: none"> - El personal encargado de tomar las fotos ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Registro de Fotos”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán sus datos en un recuadro. - Se toma la foto del postulante y se presiona el botón “Mostrar Foto” para que aparezca en el sistema. - Si se considera necesario se agrega una observación y se presiona el botón “Registrar”. 	
Resultado esperado: La información es registrada en la base de datos y la foto es guardada en una carpeta del servidor.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.3: Prueba de Aceptación 3. Registrar foto de postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 4	N° Historia de Usuario: 3
Nombre Caso de Prueba: Registrar huella de postulante	
Nombre Historia de Usuario: Registro de Huella	
Descripción: El usuario con rol de Huellas ingresa al formulario y realiza la búsqueda de un postulante por su nombre o dni, se captura la huella de este postulante y el sistema captura la imagen, de ser necesario se agrega una observación y se registra la información en la base de datos, excepto la huella, que será guardada en una carpeta del sistema y una copia en el disco local del servidor.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Huella.	
Entradas: <ul style="list-style-type: none"> - El personal encargado de capturar las huellas ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Registro de Huellas”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán sus datos en un recuadro. - El postulante coloca su dedo índice en el lector de huella digital y el sistema automáticamente mostrará la huella capturada. - Si se considera necesario se agrega una observación y se presiona el botón “Registrar”. 	
Resultado esperado: La información es registrada en la base de datos y la huella es guardada en una carpeta del servidor.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.4: Prueba de Aceptación 4. Registrar huella de postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 5	N° Historia de Usuario: 4
Nombre Caso de Prueba: Datos correctos del postulante	
Nombre Historia de Usuario: Verificar datos de postulante	
Descripción: El usuario con rol de Carnet ingresa al formulario para verificar datos de postulante, realiza la búsqueda de un postulante por su nombre o dni, lo selecciona de la lista que aparece y verifica que los datos registrados sean correctos.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Carnet.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Carnet ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Generación de Carnet”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán todos sus datos en la parte inferior, incluyendo su foto y huella. - El postulante verifica que todos sus datos sean correctos. 	
Resultado esperado: Si todos los datos del postulante son correctos y su foto y huella están registrados, el sistema habilitará la opción para imprimir su carnet.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.5: Prueba de Aceptación 5. Verificar datos de postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 6	N° Historia de Usuario: 4
Nombre Caso de Prueba: Modificar datos incorrectos del postulante	
Nombre Historia de Usuario: Verificar datos de postulante	
Descripción: El usuario con rol de Carnet ingresa al formulario para verificar datos de postulante, realiza la búsqueda de un postulante por su nombre o dni, lo selecciona de la lista que aparece y verifica los datos, si son incorrectos selecciona la opción Modificar y aparece un modal con todos los datos del postulante para que sean actualizados, se guardan los cambios, se cierra el modal y los datos correctos aparecen en el formulario.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Carnet.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Carnet ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Generación de Carnet”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán todos sus datos en la parte inferior, incluyendo su foto y huella. - El postulante verifica todos sus datos y si son incorrectos el usuario presiona el botón “Modificar” y aparece una ventana con toda la información del postulante, cambia los datos erróneos y presiona el botón “Modificar”, cierra esta ventana y en el formulario aparecen los datos corregidos. 	
Resultado esperado: La información del postulante es actualizada en la base de datos y en el formulario se muestran los nuevos datos.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.6: Prueba de Aceptación 6. Modificar datos incorrectos del postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 7	N° Historia de Usuario: 5
Nombre Caso de Prueba: Imprimir carnet de postulante	
Nombre Historia de Usuario: Impresión de carnet	
Descripción: El usuario con rol de Carnet ingresa al formulario para verificar datos de postulante, realiza la búsqueda de un postulante por su nombre o dni, lo selecciona de la lista que aparece y verifica que los datos sean correctos y que esté registrada su foto y huella. Con estos datos completos, el sistema habilitará la opción de Imprimir Carnet, se presiona este botón y aparecerá una vista previa del carnet y la opción de imprimirlo.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Carnet.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Carnet ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Generación de Carnet”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán todos sus datos en la parte inferior, incluyendo su foto y huella. - El usuario verifica que todos los datos sean correctos y esté registrada su foto y huella. - Presiona el botón “Imprimir Carnet” y aparecerá una vista previa del mismo, se presiona el botón “Imprimir”. 	
Resultado esperado: El sistema envía la orden para imprimir el carnet con los datos del postulante.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.7: Prueba de Aceptación 7. Imprimir carnet de postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 8	N° Historia de Usuario: 5
Nombre Caso de Prueba: Imprimir carnet de postulante	
Nombre Historia de Usuario: Impresión de carnet	
Descripción: El usuario con rol de Carnet ingresa al formulario para verificar datos de postulante, realiza la búsqueda de un postulante por su nombre o dni, lo selecciona de la lista que aparece y verifica que los datos sean correctos y que esté registrada su foto y huella. Con estos datos completos, el sistema habilitará la opción de Imprimir Carnet, se presiona este botón y aparecerá una vista previa del carnet y la opción de imprimirlo.	
Condiciones de ejecución: Debe existir un proceso de admisión en transcurso y un usuario con rol de Carnet.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Carnet ingresará al sistema con su usuario y contraseña. - En el menú “Proceso” selecciona “Generación de Carnet”. - En el formulario se realiza la búsqueda de un postulante por su nombre o dni; mientras se ingresan estos datos en las cajas de texto correspondiente, en la lista inferior van apareciendo las coincidencias. - En la lista se selecciona al postulante y aparecerán todos sus datos en la parte inferior, incluyendo su foto y huella. - El usuario verifica que todos los datos sean correctos y esté registrada su foto y huella. - Presiona el botón “Imprimir Carnet” y aparecerá una vista previa del mismo, se presiona el botón “Imprimir”. 	
Resultado esperado: El sistema envía la orden para imprimir el carnet con los datos del postulante.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.8: Prueba de Aceptación 8. Imprimir carnet de postulante
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 9	N° Historia de Usuario: 6
Nombre Caso de Prueba: Cambiar modalidad de postulante a ordinario	
Nombre Historia de Usuario: Cambio de modalidad (De extraordinario a ordinario)	
Descripción: El usuario con rol de Administrador ingresa al formulario de Cambio de modalidad y en la caja de texto correspondiente ingresa el código del postulante, presiona en el botón Buscar y aparecerán sus datos en el recuadro inferior, se verifica que la modalidad sea extraordinaria, sino el sistema al momento de registrar enviará un mensaje de error, se presiona el botón “Cambiar Modalidad e Imprimir Carnet”, en base de datos se actualiza la modalidad y se imprime el carnet con este nuevo dato.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresará al sistema con su usuario y contraseña. - En el menú “Administración” selecciona “Cambio de Modalidad”. - En el formulario selecciona la pestaña “Extraordinario”. - En la caja de texto correspondiente ingresa el código del postulante y presiona Enter o sino el botón “Buscar”. - Aparecerán los datos del postulante en los recuadros inferiores y el usuario verifica que la modalidad sea de tipo extraordinaria. - Si es así, presiona el botón “Cambiar Modalidad e Imprimir Carnet”. - Se imprime el carnet del postulante con modalidad de postulante “Ordinario”. 	
Resultado esperado: El sistema actualiza la modalidad del postulante a “Ordinario” y se imprime un nuevo carnet con este dato actualizado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.9: Prueba de Aceptación 9. Cambiar modalidad de postulante a ordinario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 10	N° Historia de Usuario: 7
Nombre Caso de Prueba: Cambiar modalidad de postulante a ingresante CPU	
Nombre Historia de Usuario: Cambio de modalidad (De CPU a Ingresante CPU)	
Descripción: El usuario con rol de Administrador ingresa al formulario de Cambio de modalidad y en la caja de texto correspondiente ingresa el código del postulante, presiona en el botón Buscar y aparecerán sus datos en el recuadro inferior, se verifica que la modalidad sea CPU, se ingresa el número del voucher (del pago realizado al banco) se presiona el botón “Cambiar Modalidad e Imprimir Carnet”, en base de datos se verifica que el número de voucher y el pago realizado sea correcto, y si el postulante ha ingresado por el examen CPU sino se enviará un mensaje de error, se actualiza la modalidad a Ingresante CPU y se imprime el carnet con este nuevo dato.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresará al sistema con su usuario y contraseña. - En el menú “Administración” selecciona “Cambio de Modalidad”. - En el formulario selecciona la pestaña “Centro Pre”. - En la caja de texto correspondiente ingresa el código del postulante y presiona Enter o sino el botón “Buscar”. - Aparecerán los datos del postulante en los recuadros inferiores y el usuario verifica que la modalidad sea de tipo CPU y aparezca como Ingresante. - En la caja de texto correspondiente se ingresa el número de su voucher (del pago realizado en el banco). - Se presiona el botón “Cambiar Modalidad e Imprimir Carnet”. - Se imprime el carnet del postulante con modalidad de postulante “Ingresante CPU”. 	
Resultado esperado: El sistema registra de nuevo al postulante con la modalidad de “Ingresante CPU” y se imprime un nuevo carnet.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.10: Prueba de Aceptación 10. Cambiar modalidad de postulante a ingresante CPU
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 11	N° Historia de Usuario: 8
Nombre Caso de Prueba: Listado total de postulantes	
Nombre Historia de Usuario: Lista de postulantes	
Descripción: El usuario con rol de Administrador ingresa al formulario del reporte Lista de postulantes y selecciona los parámetros de búsqueda: proceso de admisión, modalidad y escuela, presiona el botón “Buscar” y aparecerá una lista de todos los postulantes de acuerdo a los parámetros elegidos.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Reportes” selecciona “Lista de postulantes”. - En el formulario aparecen unas cajas de selección, en ellas se elige el proceso de admisión, la escuela y la modalidad. - Presionar Enter o sino el botón “Buscar”. - Aparecerá una lista con todos los postulantes que cumplan con los parámetros de búsqueda. 	
Resultado esperado: El sistema muestra una lista de postulantes con sus datos personales, de acuerdo a los parámetros de búsqueda seleccionados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.11: Prueba de Aceptación 11. Listado total de postulantes
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 12	N° Historia de Usuario: 9
Nombre Caso de Prueba: Listar postulantes por modalidad y escuela	
Nombre Historia de Usuario: Total de postulantes por modalidad y escuela	
Descripción: El usuario con rol de Administrador ingresa al formulario del reporte Total de postulantes y se mostrará tres pestañas para listar el total de postulantes por modalidad, escuela o por ambos datos, se incluye también unos cuadros estadísticos.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Reportes” selecciona “Total de postulantes por modalidad y escuela”. - En el formulario aparecen tres pestañas donde se mostrarán los resultados agrupados por modalidad, escuela y ambos. - Se elige una de las pestañas, y en el combo se selecciona un proceso de admisión. - Presionar Enter o sino el botón “Buscar”. - Aparecerá una lista mostrando el total de postulantes inscritos y agrupados por modalidad, escuela o por ambos (dependiendo de la pestaña seleccionado) y en la parte inferior un cuadro estadístico con estos resultados. 	
Resultado esperado: El sistema muestra una lista y un cuadro estadístico con el total de postulantes por modalidad y escuela.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.12: Prueba de Aceptación 12. Listar postulantes por modalidad y escuela
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 13	N° Historia de Usuario: 10
Nombre Caso de Prueba: Registrar Personal	
Nombre Historia de Usuario: Gestión de personal	
Descripción: Se ingresa al formulario de Gestión de Personal y se ingresan los datos requeridos, solo nombre, apellidos, dni y usuario son datos obligatorios, los demás campos pueden quedar en blanco o llenarse posteriormente. Se presiona el botón “Registrar” y la información se guarda en la bd. La contraseña de usuario no se ingresa porque el sistema toma como clave predeterminada el número de su dni.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En los campos correspondientes se ingresa el nombre, apellido paterno, apellido materno, sexo, fecha de nacimiento, dni, teléfono, correo, dirección, nombre de usuario y se asigna un rol. - Presionar el botón “Registrar”. - Aparece un mensaje indicando que se ha registrado correctamente el personal, presionamos en el botón “Aceptar”. - En la lista de personales aparece el nuevo dato registrado. 	
Resultado esperado: El nuevo personal es registrado en la base de datos y el sistema lo muestra en la lista de personal.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.13: Prueba de Aceptación 13. Registrar Personal
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 14	N° Historia de Usuario: 10
Nombre Caso de Prueba: Modificar Personal	
Nombre Historia de Usuario: Gestión de personal	
Descripción: Se ingresa al formulario de Gestión de Personal y de la lista que aparece en la parte inferior se selecciona al personal que serán modificados sus datos y se presiona el botón “Editar”, aparece toda la información en los campos correspondientes y el botón “Registrar” cambia por el botón “Actualizar”. Una vez que se han cambiado los datos necesarios, se presiona el botón “Actualizar”, la información se guarda en la base de datos y en la lista de personal aparecen los datos modificados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En la lista se busca al personal que se modificarán sus datos y se presiona el botón “Editar”. - Los datos del personal aparecen en los campos correspondientes, se modifica la información y se presiona el botón “Actualizar”. - Aparece un mensaje indicando que se han actualizado correctamente los datos del personal, se presiona el botón “Aceptar”. - En la lista de personal aparecen los datos modificados. 	
Resultado esperado: La información del personal es actualizada en la base de datos y el sistema lo muestra en la lista de personal.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.14: Prueba de Aceptación 14. Modificar Personal
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 15	N° Historia de Usuario: 10
Nombre Caso de Prueba: Resetear contraseña de personal	
Nombre Historia de Usuario: Gestión de personal	
Descripción: <p>Si un usuario ha olvidado su contraseña, un administrador puede ingresar al sistema y resetearla, la contraseña cambia al número de dni del usuario.</p> <p>Se ingresa al formulario de Gestión de Personal y de la lista que aparece en la parte inferior se selecciona al personal que va restablecer su contraseña, se presiona el botón “Resetear” y en base de datos se actualiza esta información. El usuario ya puede acceder al sistema con su nueva contraseña.</p>	
Condiciones de ejecución: <p>Debe existir un usuario con rol de Administrador.</p>	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En la lista se busca al personal que se restablecerá su contraseña. - Se presiona el botón “Resetear” y aparece un mensaje preguntando si realmente deseamos cambiar la contraseña del usuario. Presionamos el botón “Aceptar” y otro mensaje aparece confirmando que se han realizado los cambios. 	
Resultado esperado: <p>El personal que ha pedido el restablecimiento de su contraseña ya puede ingresar al sistema con su usuario y su nueva contraseña.</p>	
Evaluación de la prueba: <p>Prueba satisfactoria.</p>	

Tabla 6.15: Prueba de Aceptación 15. Resetear contraseña de personal
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 16	N° Historia de Usuario: 10
Nombre Caso de Prueba: Registrar rol	
Nombre Historia de Usuario: Gestión de personal	
Descripción: Se ingresa al formulario de Gestión de Personal y al lado del campo Rol aparece un botón “+ Agregar”, se da click y mostrará un modal con una caja de texto para ingresar el nombre del nuevo rol, se presiona el botón “Registrar” y el nuevo rol se guarda en la base de datos, también aparece en la lista de roles en el modal y en la lista desplegable del formulario	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En el campo Rol, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana se encuentra una caja de texto para ingresar el nombre de un nuevo rol. - Se presiona el botón “Registrar” y aparece un mensaje indicando que la información se ha registrado correctamente, se presiona el botón “Aceptar”. - El nuevo dato aparece en la lista de roles, en la parte inferior. - Se cierra la ventana y comprobamos que el nuevo dato aparece en la lista desplegable del campo Rol. 	
Resultado esperado: El nuevo rol es registrado en la base de datos y se muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.16: Prueba de Aceptación 16. Registrar rol
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 17	N° Historia de Usuario: 10
Nombre Caso de Prueba: Modificar rol	
Nombre Historia de Usuario: Gestión de personal	
Descripción: Se ingresa al formulario de Gestión de Personal y al lado del campo Rol aparece un botón “+ Agregar”, se da click y mostrará un modal con una lista de los roles registrados, en el rol que se va a modificar se presiona el botón “Editar”, el nombre del rol aparece en una caja de texto para ser modificado y el botón Registrar cambia al botón “Actualizar”, se presiona este botón y los cambios se guardan en la base de datos.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En el campo Rol, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los roles registrados. - En el rol que se desea modificar, se presiona el botón “Editar” y en la caja de texto correspondiente aparece el nombre para editarlo. - Realizados los cambios se presiona el botón “Actualizar” y aparece un mensaje indicando que la información se ha actualizado correctamente, se presiona el botón “Aceptar”. - En la lista de roles aparece el dato modificado. - Se cierra la ventana y comprobamos que el dato modificado aparece en la lista desplegable del campo Rol. 	
Resultado esperado: Los cambios en el rol son guardados en la base de datos.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.17: Prueba de Aceptación 17. Modificar rol
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 18	N° Historia de Usuario: 10
Nombre Caso de Prueba: Eliminar rol	
Nombre Historia de Usuario: Gestión de personal	
Descripción: Se ingresa al formulario de Gestión de Personal y al lado del campo Rol aparece un botón “+ Agregar”, se da click y mostrará un modal con una lista de los roles registrados, en el rol que se va a eliminar se presiona el botón “Eliminar” aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Personal”. - En el campo Rol, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los roles registrados. - En el rol que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - El rol ya no se muestra en la lista de roles. 	
Resultado esperado: En base de datos se cambia la vigencia del rol a <i>false</i> y en el sistema ya no se muestra este dato.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.18: Prueba de Aceptación 18. Eliminar rol
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 19	N° Historia de Usuario: 11
Nombre Caso de Prueba: Ingreso al sistema con usuario o contraseña incorrectos	
Nombre Historia de Usuario: Iniciar sesión	
Descripción: Cuando el usuario ingresa al sistema se muestra una página de inicio y en los cuadros correspondientes ingresará su nombre de usuario y contraseña, en base de datos se verifica esta información y si es incorrecta se mostrará un mensaje de error y no se podrá acceder al sistema.	
Condiciones de ejecución: Ninguna	
Entradas: <ul style="list-style-type: none"> - El personal ingresa al sistema. - En la página principal aparecen unos cuadros de texto donde se ingresa el nombre de usuario y la contraseña. - Se presiona Enter o sino el botón “Ingresar”. - El sistema verifica los datos y si son incorrectos muestra un mensaje con el error encontrado 	
Resultado esperado: Si el usuario o contraseña son incorrectos o si se ha deshabilitado al usuario, se le denegará el acceso al sistema. En base de datos se guarda la información sobre este intento de acceso.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.19: Prueba de Aceptación 19. Ingreso al sistema con usuario o contraseña incorrectos
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 20	N° Historia de Usuario: 11
Nombre Caso de Prueba: Ingreso al sistema con usuario y contraseña correctos	
Nombre Historia de Usuario: Iniciar sesión	
Descripción: Cuando el usuario ingresa al sistema se muestra una página de inicio y en los cuadros correspondientes ingresará su nombre de usuario y contraseña, en base de datos se verifica esta información y si es correcta se permitirá el acceso al sistema.	
Condiciones de ejecución: Ninguna	
Entradas: <ul style="list-style-type: none"> - El personal ingresa al sistema. - En la página principal aparecen unos cuadros de texto donde se ingresa el nombre de usuario y la contraseña. - Se presiona Enter o sino el botón “Ingresar”. - El sistema verifica los datos y si son correctos se mostrará la página de su perfil y un menú con las opciones del sistema. En el menú sólo aparecen las opciones que han sido permitidas en su rol de usuario. 	
Resultado esperado: Si el usuario y contraseña son correctos se permite el acceso al menú con las opciones correspondientes a su rol de usuario. En base de datos se guarda la información sobre este acceso al sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.20: Prueba de Aceptación 20. Ingreso al sistema con usuario y contraseña correctos
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 21	N° Historia de Usuario: 12
Nombre Caso de Prueba: Modificar contraseña de usuario	
Nombre Historia de Usuario: Perfil de Usuario	
Descripción: Cuando el usuario ingresa a su perfil de usuario aparece una opción para modificar su contraseña, al seleccionar esta opción se muestra un modal en donde ingresará su contraseña y una contraseña nueva. La contraseña nueva se guarda en base de datos.	
Condiciones de ejecución: Ninguno.	
Entradas: <ul style="list-style-type: none"> - El personal ingresa al sistema con su usuario y contraseña. - La primera página que aparece es el perfil de usuario, se muestra un botón “Cambiar Contraseña”, se presiona y aparece una ventana. - En la ventana se pide ingresar la contraseña antigua, y dos veces la contraseña nueva, se llenan todos los campos y se presiona el botón “Guardar contraseña”. - Aparece un mensaje indicando que la información se ha guardado correctamente. - Presionamos el botón “Aceptar”. 	
Resultado esperado: En base de datos se actualiza la contraseña y el usuario podrá utilizarla en el siguiente acceso al sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.21: Prueba de Aceptación 21. Modificar contraseña de usuario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 22	N° Historia de Usuario: 13
Nombre Caso de Prueba: Registrar permisos de usuario	
Nombre Historia de Usuario: Asignar permisos a usuarios	
Descripción: El usuario ingresa al formulario Asignar permisos de usuario donde se muestra una lista con todas las opciones del sistema, se escoge un rol de usuario y en la lista se seleccionan los formularios a los que se permitirá el acceso. Se guardan los cambios.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Administración” selecciona “Asignar permisos”. - En este formulario se selecciona un rol de usuario y en la lista de opciones del sistema aparecen con un check los formularios a los que se permite ingresar. Con un clic en ellas se selecciona o deselecciona las opciones. - Presionamos el botón “Registrar”. - Aparece un mensaje indicando que la información se ha guardado correctamente. - Presionamos el botón “Aceptar”. 	
Resultado esperado: La información se guarda en base de datos. En el menú del sistema solo se mostrarán los formularios a los que se ha permitido el acceso dependiendo del rol de usuario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.22: Prueba de Aceptación 22. Registrar permisos de usuario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 23	N° Historia de Usuario: 14
Nombre Caso de Prueba: Registrar modalidad	
Nombre Historia de Usuario: Gestión de modalidad	
Descripción: Se ingresa al formulario de Gestión de modalidad y en las cajas de texto se ingresan los datos de la nueva modalidad. Se presiona el botón “Registrar” y la información se guarda en la base de datos y aparece en el formulario, en la lista de modalidades.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En los campos correspondientes se ingresa el nombre, se selecciona el tipo de modalidad y en “Datos requeridos del postulante” se selecciona que información se va a pedir al postulante cuando ingrese al formulario de inscripción. - Presionar el botón “Registrar”. - Aparece un mensaje indicando que se ha registrado correctamente la nueva modalidad, se presiona en el botón “Aceptar”. - En la lista de modalidades aparece el nuevo dato registrado. 	
Resultado esperado: La nueva modalidad es registrada en la base de datos y el sistema lo muestra en la lista de modalidades.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.23: Prueba de Aceptación 23. Registrar modalidad
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 24	N° Historia de Usuario: 14
Nombre Caso de Prueba: Modificar modalidad	
Nombre Historia de Usuario: Gestión de modalidad	
Descripción: Se ingresa al formulario de Gestión de Modalidad y de la lista que aparece en la parte inferior se selecciona la modalidad que será modificada y se presiona el botón “Editar”, aparece toda la información en los campos correspondientes y el botón “Registrar” cambia por el botón “Actualizar”. Una vez que se han cambiado los datos necesarios, se presiona el botón “Actualizar”, la información se guarda en la base de datos y en la lista de modalidades aparecen los datos modificados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En la lista se busca la modalidad que se modificará y se presiona el botón “Editar”. - Los datos de la modalidad aparecen en los campos correspondientes, se modifica la información y se presiona el botón “Actualizar”. - Aparece un mensaje indicando que se han actualizado correctamente los datos de la modalidad, se presiona el botón “Aceptar”. - En la lista de modalidad aparecen los datos modificados. 	
Resultado esperado: La información de la modalidad es actualizada en la base de datos y el sistema lo muestra en la lista de modalidades.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.24: Prueba de Aceptación 24. Modificar modalidad
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 25	N° Historia de Usuario: 14
Nombre Caso de Prueba: Eliminar modalidad	
Nombre Historia de Usuario: Gestión de modalidad	
Descripción: Se ingresa al formulario de Gestión de Modalidad, en la parte inferior se encuentra la lista de las modalidades, en la modalidad que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En la parte inferior aparece una lista con todas las modalidades registradas. - En la modalidad que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - La modalidad es eliminada y ya no se muestra en la lista. 	
Resultado esperado: En base de datos se cambia la vigencia de la modalidad a <i>false</i> y en el sistema ya no se muestra este dato.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.25: Prueba de Aceptación 25. Eliminar modalidad
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 26	N° Historia de Usuario: 14
Nombre Caso de Prueba: Registrar tipo de modalidad	
Nombre Historia de Usuario: Gestión de Modalidad	
Descripción: Se ingresa al formulario de Gestión de Modalidad y al lado del campo Tipo de Modalidad aparece un botón “+ Agregar”, se da click y mostrará un modal con una caja de texto para ingresar el nombre del nuevo tipo de modalidad, se presiona el botón “Registrar” y se guarda en la base de datos, el nuevo tipo de modalidad aparece en la lista del modal y en la lista desplegable del formulario	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En el campo Tipo de modalidad, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana se encuentra una caja de texto para ingresar el nombre de un nuevo tipo de modalidad. - Se presiona el botón “Registrar” y aparece un mensaje indicando que la información se ha registrado correctamente, se presiona el botón “Aceptar”. - El nuevo dato aparece en la lista de la parte inferior. - Se cierra la ventana y comprobamos que el nuevo dato aparece en la lista desplegable del campo Tipo de modalidad. 	
Resultado esperado: El nuevo tipo de modalidad es registrado en la base de datos y se muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.26: Prueba de Aceptación 26. Registrar tipo de modalidad
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 27	N° Historia de Usuario: 14
Nombre Caso de Prueba: Modificar tipo de modalidad	
Nombre Historia de Usuario: Gestión de Modalidad	
Descripción: Se ingresa al formulario de Gestión de Modalidad y al lado del campo Tipo de Modalidad aparece un botón “+ Agregar”, se da click y mostrará un modal con una lista de los tipo de modalidad registrados, en el dato que se va a modificar se presiona el botón “Editar”, el nombre del tipo de modalidad aparece en una caja de texto para ser modificado y el botón “Registrar” cambia al botón “Actualizar”, se presiona este botón y los cambios se guardan en la base de datos.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En el campo Tipo de Modalidad, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los tipos de modalidad registrados. - En el tipo de modalidad que se desea modificar, se presiona el botón “Editar” y en la caja de texto correspondiente aparece el nombre para editarlo. - Realizados los cambios se presiona el botón “Actualizar” y aparece un mensaje indicando que la información se ha actualizado correctamente, se presiona el botón “Aceptar”. - En la lista de tipos de modalidad aparece el dato modificado. - Se cierra la ventana y comprobamos que el dato modificado aparece en la lista desplegable del campo Tipo de Modalidad. 	
Resultado esperado: Los cambios realizados son guardados en la base de datos y mostrados en el formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.27: Prueba de Aceptación 27. Modificar tipo de modalidad
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 28	N° Historia de Usuario: 14
Nombre Caso de Prueba: Eliminar tipo de modalidad	
Nombre Historia de Usuario: Gestión de Modalidad	
Descripción: Se ingresa al formulario de Gestión de Modalidad y al lado del campo Tipo de Modalidad aparece un botón “+ Agregar”, se da click y mostrará un modal con una lista de los tipos de modalidad registrados, en el dato que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Modalidad”. - En el campo Tipo de Modalidad, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los tipos de modalidad registrados. - En el tipo de modalidad que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - El tipo de modalidad eliminado ya no se muestra en la lista. 	
Resultado esperado: En base de datos se cambia la vigencia del tipo de modalidad a <i>false</i> y en el sistema ya no se muestra este dato.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.28: Prueba de Aceptación 28. Eliminar tipo de modalidad
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 29	N° Historia de Usuario: 15
Nombre Caso de Prueba: Registrar escuela	
Nombre Historia de Usuario: Gestión de escuela	
Descripción: Se ingresa al formulario de Gestión de escuela y en las cajas de texto se ingresan los datos de la nueva escuela. Se presiona el botón “Registrar”, la información se guarda en la base de datos y aparece en el formulario, en la lista de datos registrados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Escuela”. - En los campos correspondientes se ingresa el nombre y se selecciona el grupo al que pertenece. - Presionar el botón “Registrar”. - Aparece un mensaje indicando que se ha registrado correctamente la nueva escuela, se presiona en el botón “Aceptar”. - En la lista de escuelas aparece el nuevo dato registrado. 	
Resultado esperado: La nueva escuela es registrada en la base de datos y el sistema lo muestra en la lista de escuelas.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.29: Prueba de Aceptación 29. Registrar escuela
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 30	N° Historia de Usuario: 15
Nombre Caso de Prueba: Modificar escuela	
Nombre Historia de Usuario: Gestión de escuela	
Descripción: Se ingresa al formulario de Gestión de Escuela y de la lista que aparece en la parte inferior se selecciona la escuela que será modificada, se presiona el botón “Editar” y aparece toda la información en los campos correspondientes y el botón “Registrar” cambia por el botón “Actualizar”. Una vez que se han cambiado los datos necesarios, se presiona el botón “Actualizar”, la información se guarda en la base de datos y en la lista de escuelas aparecen los datos modificados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Escuela”. - En la lista se busca la escuela que se modificará y se presiona el botón “Editar”. - Los datos de la escuela aparecen en los campos correspondientes, se modifica la información y se presiona el botón “Actualizar”. - Aparece un mensaje indicando que se han actualizado correctamente los datos de la escuela, se presiona el botón “Aceptar”. - En la lista de escuela aparecen los datos modificados. 	
Resultado esperado: La información de la escuela es actualizada en la base de datos y el sistema lo muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.30: Prueba de Aceptación 30. Gestión de escuela
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 31	N° Historia de Usuario: 15
Nombre Caso de Prueba: Eliminar escuela	
Nombre Historia de Usuario: Gestión de escuela	
Descripción: Se ingresa al formulario de Gestión de Escuela, en la parte inferior se encuentra la lista de los datos registrados, en la escuela que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema. No debe existir ningún postulante inscrito en la escuela a eliminar, sino el sistema no permitirá realizar esta acción.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Escuela”. - En la parte inferior aparece una lista con todas las escuelas registradas. - En la escuela que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - La escuela es eliminada y ya no se muestra en la lista. 	
Resultado esperado: En base de datos se verifica que no haya ningún postulante registrado en esta escuela si es así se cambia la vigencia de la escuela a <i>false</i> y en el sistema ya no se muestra este dato, sino se muestra un mensaje de error indicando que no se puede eliminar esta escuela porque tiene un dato asociado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.31: Prueba de Aceptación 31. Eliminar escuela
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 32	N° Historia de Usuario: 15
Nombre Caso de Prueba: Registrar grupo	
Nombre Historia de Usuario: Gestión de Escuela	
Descripción: Se ingresa al formulario de Gestión de Escuela y al lado del campo Grupo aparece un botón “+ Agregar”, se da click y mostrará un modal con una caja de texto para ingresar el nombre del nuevo grupo, se presiona el botón “Registrar” y se guarda en la base de datos, el nuevo grupo aparece en la lista del modal y en la lista desplegable del formulario	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Escuela”. - En el campo Grupo, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana se encuentra una caja de texto para ingresar el nombre de un nuevo grupo. - Se presiona el botón “Registrar” y aparece un mensaje indicando que la información se ha registrado correctamente, se presiona el botón “Aceptar”. - El nuevo dato aparece en la lista de la parte inferior. - Se cierra la ventana y comprobamos que el nuevo dato aparece en la lista desplegable del campo Grupo. 	
Resultado esperado: El nuevo grupo es registrado en la base de datos y se muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.32: Prueba de Aceptación 32. Registrar grupo
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 33	N° Historia de Usuario: 15
Nombre Caso de Prueba: Modificar grupo	
Nombre Historia de Usuario: Gestión de Escuela	
Descripción: Se ingresa al formulario de Gestión de Escuela y al lado del campo Grupo aparece un botón "+ Agregar", se da click y mostrará un modal con una lista de los grupos registrados, en el dato que se va a modificar se presiona el botón "Editar", el nombre del grupo aparece en una caja de texto para ser modificado y el botón "Registrar" cambia al botón "Actualizar", se presiona este botón y los cambios se guardan en la base de datos.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú "Registros" selecciona "Gestión de Escuela". - En el campo Grupo, al lado derecho aparece un botón "+ Agregar", se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los grupos registrados. - En el grupo que se desea modificar, se presiona el botón "Editar" y en la caja de texto correspondiente aparece el nombre para editarlo. - Realizados los cambios se presiona el botón "Actualizar" y aparece un mensaje indicando que la información se ha actualizado correctamente, se presiona el botón "Aceptar". - En la lista de grupos aparece el dato modificado. - Se cierra la ventana y comprobamos que el dato modificado aparece en la lista desplegable del campo Grupo. 	
Resultado esperado: Los cambios realizados son guardados en la base de datos y mostrados en el formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.33: Prueba de Aceptación 33. Modificar grupo
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 34	N° Historia de Usuario: 15
Nombre Caso de Prueba: Eliminar grupo	
Nombre Historia de Usuario: Gestión de Escuela	
Descripción: Se ingresa al formulario de Gestión de Escuela y al lado del campo Grupo aparece un botón “+ Agregar”, se da click y mostrará un modal con una lista de los grupos registrados, en el dato que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema. No debe existir ninguna escuela registrada en este grupo sino el sistema no permitirá eliminarlo.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Escuela”. - En el campo Grupo, al lado derecho aparece un botón “+ Agregar”, se presiona y aparece una nueva ventana. - En esta ventana aparece una lista de todos los grupos registrados. - En el grupo que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - El grupo eliminado ya no se muestra en la lista. 	
Resultado esperado: En base de datos se verifica que el grupo no esté asociado a una escuela, si cumple con esto se cambia la vigencia del grupo a <i>false</i> y en el sistema ya no se muestra este dato, sino se enviará un mensaje de error al sistema indicando que ese grupo no se puede eliminar.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.34: Prueba de Aceptación 34. Eliminar grupo
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 35	N° Historia de Usuario: 16
Nombre Caso de Prueba: Registrar tarifario	
Nombre Historia de Usuario: Gestión de Tarifario	
Descripción: Se ingresa al formulario de Gestión de Tarifario y en las cajas de texto se ingresan los datos del nuevo tarifario. Se presiona el botón “Registrar”, la información se guarda en la base de datos y aparece en el formulario, en la lista de datos registrados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Tarifario”. - En los campos correspondientes se selecciona la modalidad, la escuela y se ingresa el monto total correspondiente. Si esta tarifa puede ser cancelada en cuotas, se activa la casilla “Pago por cuotas” y aparecen unas cajas de texto para ingresar el número de cuotas y el monto de la primera cuota. - Presionar el botón “Registrar”. - Aparece un mensaje indicando que se ha registrado correctamente el nuevo tarifario, se presiona en el botón “Aceptar”. - En la lista de tarifarios aparece el nuevo dato registrado. 	
Resultado esperado: En la base de datos se verifica que no exista un tarifario vigente con la misma escuela y modalidad, si es así el nuevo tarifario es registrado en la base de datos y el sistema lo muestra en la lista del formulario, sino se enviará un mensaje al sistema indicando que los datos ya existen.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.35: prueba de Aceptación 35. Registrar tarifario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 36	N° Historia de Usuario: 16
Nombre Caso de Prueba: Modificar tarifario	
Nombre Historia de Usuario: Gestión de Tarifario	
Descripción: Se ingresa al formulario de Gestión de Tarifario y de la lista que aparece en la parte inferior se selecciona el tarifario que será modificado, se presiona el botón “Editar” y aparece toda la información en los campos correspondientes y el botón “Registrar” cambia por el botón “Actualizar”. Una vez que se han cambiado los datos necesarios, se presiona el botón “Actualizar”, la información se guarda en la base de datos y en la lista de tarifarios aparecen los datos modificados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Tarifario”. - En la lista se busca el tarifario que se modificará y se presiona el botón “Editar”. - Los datos del tarifario aparecen en los campos correspondientes, se modifica la información y se presiona el botón “Actualizar”. - Aparece un mensaje indicando que se han actualizado correctamente los datos del tarifario, se presiona el botón “Aceptar”. - En la lista de tarifarios aparecen los datos modificados. 	
Resultado esperado: La información del tarifario es actualizada en la base de datos y el sistema lo muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.36: Prueba de Aceptación 36. Modificar tarifario
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 37	N° Historia de Usuario: 16
Nombre Caso de Prueba: Eliminar tarifario	
Nombre Historia de Usuario: Gestión de Tarifario	
Descripción: Se ingresa al formulario de Gestión de Tarifario, en la parte inferior se encuentra la lista de los datos registrados, en el tarifario que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema. No debe existir ningún postulante relacionado con este tarifario, sino el sistema no permitirá realizar esta acción, en este caso, si ya no se requiere este tarifario, se accede a la opción “Editar” y se desactiva la casilla de vigencia de esta tarifa.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Tarifario”. - En la parte inferior aparece una lista con todos los tarifarios registrados. - En el tarifario que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - El tarifario es eliminado y ya no se muestra en la lista. 	
Resultado esperado: En base de datos se verifica que no haya ningún postulante asociado con este tarifario si es así se elimina este dato y ya no se muestra en el sistema, sino se envía un mensaje de error indicando que no se puede eliminar este tarifario porque tiene un dato asociado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.37: Prueba de Aceptación 37. Eliminar tarifario
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 38	N° Historia de Usuario: 17
Nombre Caso de Prueba: Registrar proceso de admisión	
Nombre Historia de Usuario: Gestión de Proceso de Admisión	
Descripción: Se ingresa al formulario de Gestión de Proceso de Admisión y en las cajas de texto se ingresan los datos del nuevo proceso de admisión. Se presiona el botón “Registrar”, la información se guarda en la base de datos y aparece en el formulario, en la lista de datos registrados.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Proceso de Admisión”. - En los campos correspondientes se ingresa el nombre y las fechas de inicio y fin del proceso. En la siguiente sección “Detalle del Proceso de Admisión” se irá agregando los tipos de modalidad disponibles en este proceso junto con la fecha límite en que se puede realizar la inscripción durante el proceso, presionar el botón “Agregar Detalle” cada vez que se incluya un dato. - Cuando ya esté toda la información lista presionar el botón “Registrar”. - Aparece un mensaje indicando que se ha registrado correctamente el nuevo proceso de admisión, se presiona en el botón “Aceptar”. - En la lista de proceso de admisión aparece el nuevo dato registrado. 	
Resultado esperado: En la base de datos se verifica que las fechas del nuevo proceso de admisión no se cruce con otro proceso y que la fecha límite del detalle se encuentre dentro del rango de fechas del proceso, si todo es conforme el nuevo proceso de admisión es registrado en la base de datos y el sistema lo muestra en la lista del formulario, sino se enviará un mensaje al sistema indicando que los datos ya son incorrectos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.38: Prueba de Aceptación 38. Registrar proceso de admisión
Fuente: Elaboración propia

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 39	N° Historia de Usuario: 17
Nombre Caso de Prueba: Modificar proceso de admisión	
Nombre Historia de Usuario: Gestión de Proceso de Admisión	
Descripción: Se ingresa al formulario de Gestión de Proceso de Admisión y de la lista que aparece en la parte inferior se selecciona el tarifario que será modificado, se presiona el botón “Editar” y aparece toda la información en los campos correspondientes y el botón “Registrar” cambia por el botón “Actualizar”. Una vez que se han cambiado los datos necesarios, se presiona el botón “Actualizar”, la información se guarda en la base de datos y en la lista de procesos de admisión aparecen los datos modificados. Si un proceso de admisión ya ha finalizado o se encuentra en trancurso no se podrán modificar sus datos y el botón “Editar” será deshabilitado.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Proceso de Admisión”. - En la lista se busca el proceso de admisión que se modificará y se presiona el botón “Editar”. - Los datos del proceso de admisión aparecen en los campos correspondientes, se modifica la información y se presiona el botón “Actualizar”. - Aparece un mensaje indicando que se han actualizado correctamente los datos del proceso de admisión, se presiona el botón “Aceptar”. - En la lista de procesos de admisión aparecen los datos modificados. 	
Resultado esperado: La información del proceso de admisión es actualizada en la base de datos y el sistema lo muestra en la lista del formulario.	
Evaluación de la prueba: Prueba satisfactoria.	

*Tabla 6.39: Prueba de Aceptación 39. Modificar proceso de admisión
Fuente: Elaboración propia*

CASO DE PRUEBA DE ACEPTACIÓN	
N° Caso de Prueba: 40	N° Historia de Usuario: 17
Nombre Caso de Prueba: Eliminar proceso de admisión	
Nombre Historia de Usuario: Gestión de Proceso de Admisión	
Descripción: Se ingresa al formulario de Gestión de Proceso de Admisión, en la parte inferior se encuentra la lista de los datos registrados, en el proceso de admisión que se va a eliminar se presiona el botón “Eliminar”, aparece un mensaje de confirmación, lo aceptamos y este dato ya no se muestra en el sistema. Si un proceso de admisión ya ha finalizado o se encuentra en transcurso no se podrá eliminar, incluso el botón “Eliminar” será deshabilitado.	
Condiciones de ejecución: Debe existir un usuario con rol de Administrador.	
Entradas: <ul style="list-style-type: none"> - El personal con rol de Administrador ingresa al sistema con su usuario y contraseña. - En el menú “Registros” selecciona “Gestión de Proceso de Admisión”. - En la parte inferior aparece una lista con todos los procesos de admisión registrados. - En el proceso de admisión que se va a eliminar, se presiona el botón “Eliminar”, aparece un mensaje preguntando si se está seguro de la acción, presionamos el botón “Aceptar”. - El tarifario es eliminado y ya no se muestra en la lista. 	
Resultado esperado: En base de datos se cambia la vigencia del proceso de admisión a <i>false</i> y en el sistema ya no se muestra este dato	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.40: Prueba de Aceptación 40. Eliminar proceso de admisión
Fuente: Elaboración propia

6.4. Características que no serán probadas

Fue excluido del plan de pruebas:

Aplicaciones y herramientas no desarrolladas (por ejemplo, Sistema Operativo.

6.5. Personal administrativo a quienes fue aplicado la capacitación y pruebas de funcionalidad del sistema

	Apellidos y Nombres	Cargo
1	Mg. Richart Hernández Fiesta	Jefe de la Oficina de Admisión – Unach.
2	Tc. Johan Jeampier Briones Delgado	Unidad de Soporte Informático – Unach.
3	Lic. Gilder Cieza Altamirano	Encargado del Centro Pre – Unach.
4	Celia Idrogo Barboza	Secretaria del Centro Pre – Unach.

Tabla 6.41: Personal administrativo que realizó las pruebas de los sistemas
Fuente: Elaboración propia

CAPITULO VII

SEGURIDAD



7. SEGURIDAD

7.1. Sistema Administrativo

- A. Para ingresar al sistema se necesita un usuario y contraseña que han sido asignados por un administrador.

La primera vez que se ingresa al sistema la contraseña predeterminada es el dni del usuario.



Figura 1.1: Iniciar sesión

Fuente: Elaboración propia

- B. La contraseña del usuario puede ser modificada por una de su preferencia cuando ingrese al sistema en el menú de su perfil.

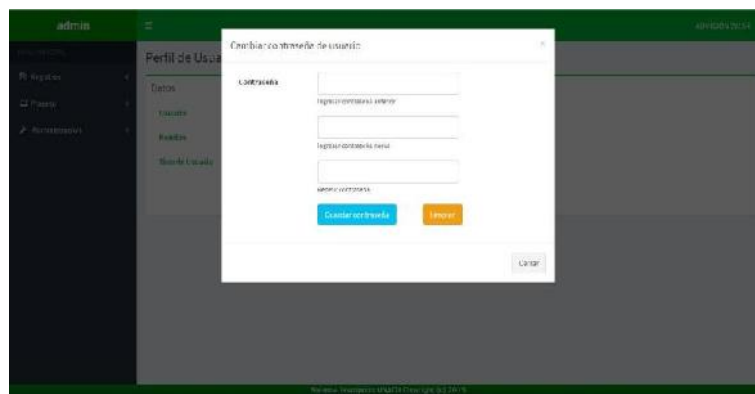


Figura 1.2: Modificar contraseña de usuario

Fuente: Elaboración propia

- C. Solo un administrador puede asignar permisos de usuarios, dependiendo del rol que cumplan, se permitirá el acceso a determinadas páginas.

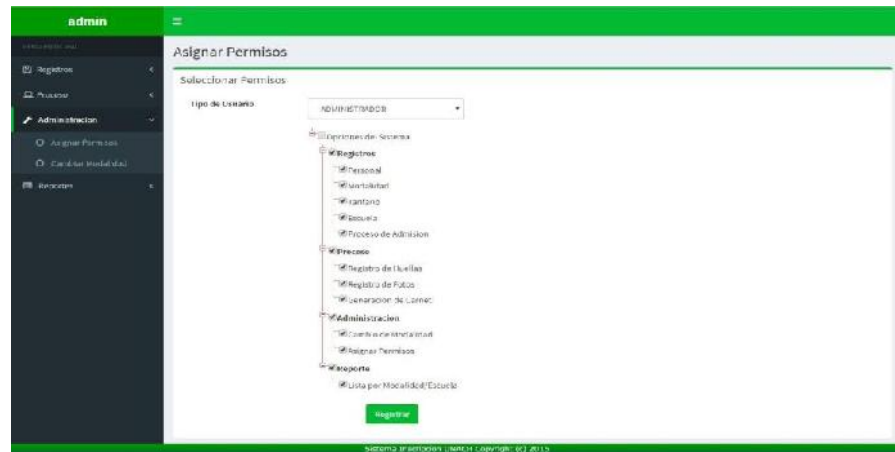


Figura 1.3: Asignar permisos
Fuente: Elaboración propia

- D. Si un usuario intenta ingresar a una página a la que no tiene acceso, por ejemplo, mediante el link, el sistema mostrará la página de "Acceso Denegado".



Figura 1.4: Acceso denegado
Fuente: Elaboración propia

- E. El usuario solo puede visualizar las opciones de menú que han sido asignadas a su rol.



Figura 1.5: Menú para el usuario con rol de Fotos
Fuente: Elaboración propia

7.2. Sistema de Inscripción



Figura 2.1: Sistema de inscripción de postulantes
Fuente: Elaboración propia

- A. Para ingresar al sistema de inscripción el postulante deberá ingresar un usuario, contraseña y la modalidad a que postula.

Figura 2.2: Ingreso a la inscripción de postulantes
Fuente: Elaboración propia

- B. El postulante deberá llenar cada uno de los campos solicitados en forma ordenada

The screenshot shows the 'ADMISSION 2016 I' form for 'Solo para Triángulo de Oro'. The form is titled 'Datos del Postulante Unach'. It contains three input fields: 'Documento de Identidad', 'Documento de Pago', and 'Modalidad a que postula'. The 'Modalidad a que postula' field has a dropdown menu with the option 'Selección Opción'. There is a 'Consultar' button at the bottom. A green curved line highlights the form area.

Figura 2.3: Paso1 - Campos Validados I
Fuente: Elaboración propia

- C. El postulante una vez registrado no podrá realizar nuevamente el proceso de inscripción.

The screenshot shows the 'ADMISSION 2016 I' form for 'Solo para Triángulo de Oro'. A red error message box is displayed over the form, stating 'ERROR: Datos ingresados son incorrectos'. The form fields are filled with the following data: 'Documento de Identidad' (22222222), 'Documento de Pago' (222222), and 'Modalidad a que postula' (GRADUADOS Y TITULADOS). The 'Consultar' button is visible at the bottom. A green curved line highlights the form area.

Figura 2.4: Paso 1 - Campos Validados II – Mensaje de error
Fuente: Elaboración propia

- D. Dependiendo de la modalidad ingresada en el paso 1 el sistema mostrará los campos información académica o información extraordinaria.

UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA

OFICINA CENTRAL DE ADMISION

Ficha de Inscripción UNACH

Información Académica :

*) Datos necesarios de la Institución Educativa.

Departamento:

Provincia:

Distrito:

Tipo de Institución Educativa:

Nombre de la Institución Educativa:

Año de Egreso de la Institución Educativa:

Información ExtraOrdinaria :

*) Esta información solo será llenada para los que postulan bajo la modalidad de Cambio de universidad y Graduados y titulados.

Tipo de Universidad:

Departamento:

Universidad:

Universidad :

Escuela a que postula:

Universidad Nacional Autónoma de Chota — Copyright © 2016 — Todos los Derechos Reservados. Oficina Administrativa: Jr. Gregorio Makra N° 875 — Oficina Central: Jr. José Osorio N° 418 — Ciudad Universitaria: Colpamatará

Figura 2.5: Campos Validados III
Fuente: Elaboración propia

D.1. Modalidad Graduado y Titulado – Cambio de Universidad

UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA OFICINA CENTRAL DE ADMISION

Ficha de Inscripción UNACH

Modo de Ingreso: GRADUADOS Y TITULADOS

Información Extraordinaria:

Tipo de Universidad: INICIAL

Departamento: CAJAMARCA

Provincia: UNIV. NACIONAL PEDRO RUIZ GALLO

Universidad:

Escuela o que postula: CIVIL COMERCIAL

Figura 2.6: Paso 3 - Graduados y titulados - Cambio de Universidad
Fuente: Elaboración propia

D.2. Modalidad Primeros puestos, Deportista Calificado, Ordinario, Persona Discapacitado.

UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA OFICINA CENTRAL DE ADMISION

Ficha de Inscripción UNACH

Información Académica:

Departamento:

Provincia:

Distrito:

Tipo de Institución Educativa:

Nombre de la Institución Educativa:

Año de ingreso de la institución educativa:

Universidad:

Escuela o que postula:

Figura 2.7: Paso 3 - Campos Validados
Fuente: Elaboración propia

- E. En caso el Postulante quisiera ingresar mediante “url” sin haber iniciado sesión este lo redireccionará a la página de inicio denegando su acceso y pidiendo su usuario y contraseña.

CAPITULO VIII

MANTENIMIENTO Y ACTUALIZACIÓN



8. MANTENIMIENTO Y ACTUALIZACIÓN

El administrador de un sistema gestor de bases de datos, tendrá la función de realizar periódicamente el mantenimiento de los datos, actualizaciones, y limpieza de los identificadores internos y de las estadísticas de planificación de las consultas, a una reindexación periódica de las tablas, a continuación se describe en qué consiste las actividades a realizar por el administrador:

8.1. Vacuum:

Es el proceso en el cual se eliminan definitivamente tuplas marcadas para borrar y hay una reorganización de datos a nivel físico. Se justifica el uso de vacuum por lo siguiente:

- ✓ Recuperar el espacio de disco perdido en borrados y actualizaciones de datos.
- ✓ Actualizar las estadísticas de datos utilizados por el planificador de consultas SQL.
- ✓ Protegerse ante la pérdida de datos por reutilización de identificadores de transacción.

8.2. Reindexacion:

La reindexación completa de la base de datos no es una tarea muy habitual, pero puede mejorar sustancialmente la velocidad de las consultas complejas en tablas con mucha actividad.

CONCLUSIONES Y RECOMENDACIONES

A. CONCLUSIONES

- Automatizando el proceso de admisión se comprobó que hubo una mejora en la calidad del trabajo de los usuarios del sistema, siendo más rápido y confiable el registro de la información.
- Utilizando PostgreSQL como gestor de base de datos del sistema nos permite ahorrar en costos de licencia, es multiplataforma, nos permite crear usuarios, darle permisos y restringir acceso a la información.
- Utilizando la metodología de desarrollo " *eXtreme Programming - XP* ", nos ha permitido mejorar nuestros procesos de ejecución durante todas las etapas de desarrollo elaborando un buen producto.
- Utilizando el patrón de diseño modelo, vista, controlador; nos brindó un gran número de funcionalidades como la reutilización de código, un mejor diseño, y modularización, que facilitará el mantenimiento y actualización del sistema.
- Registrando las fotos y las huellas de los postulantes podemos verificar su identidad y evitamos suplantaciones, mejorando la seguridad en el proceso de admisión.

B. RECOMENDACIONES

- Mientras dure el proceso de admisión se recomienda crear copias de seguridad de la base de datos diariamente.
- Adquirir equipos de suministro ininterrumpido de fluido eléctrico (UPS) con la finalidad de evitar caídas eléctricas que podrían ocasionar pérdida de datos.
- Concientizar a los usuarios del sistema sobre la confidencialidad de las claves de acceso para evitar que personas ajenas a la institución accedan a la información.

DEFINICIÓN DE TÉRMINOS

- **Analista:** Persona que diseña el software necesario para realizar una aplicación.
- **Applet:** Es un componente de una aplicación que se ejecuta en el contexto de otro programa puede ejecutarse de manera independiente, y a veces interactúa con el usuario.
- **Bases de datos:** Es un conjunto de datos almacenados sistemáticamente para su posterior uso.
- **Hardware:** Son las partes físicas de un ordenador.
- **IDE:** Un **entorno de desarrollo integrado**, (sigla en inglés de *integrated development environment*). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, Gambas, etc.
- **JAVA:** Java es un lenguaje de programación orientado a objetos, el lenguaje tiene un modelo de objetos más simple y elimina herramientas de bajo nivel.
- **JDK (Java Development Kit):** Se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje Java.
- **SDK (Software Development Kit):** kit de desarrollo de software es un conjunto de herramientas de desarrollo de software que permite a un programador crear programas y aplicaciones para un sistema o plataforma concretos.

- **MVC (Modelo Vista Controlador):** Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **JDBC:** driver que permite conectar un programa Java con una base de datos.
- **JSP:** página HTML donde se puede añadir marcas con scripts en java.
- **NetBeans:** Es un entorno de desarrollo integrado, libre de código abierto, para desarrolladores de software.
- **Postgres:** Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.
- **Sistema:** Es una serie de elementos que funcionan en conjunto para alcanzar un objetivo.
- **Sistema operativo:** Es un conjunto de programas de computadora que permite la administración eficaz de sus recursos.
- **Software:** Es el conjunto de los componentes necesarios para hacer posible la realización de una tarea, la cual trabaja conjuntamente con el hardware.
- **Software Libre:** El software libre (free software) respeta la libertad de los usuarios sobre el producto que éste adquiera, por lo que, una vez obtenido, éste puede ser distribuido, modificado, redistribuido, copiado y usado libremente.

REFERENCIAS BIBLIOGRÁFICAS


- Aumaille, B. (2002). *J2EE: Desarrollo de aplicaciones Web*. Ediciones ENI.
- Bahit, E. (2012). *Scrum & eXtreme Programming para Programadores*. Buenos Aires: Safe Creative.
- Beck, K. (2000). *Una explicación de la Programación Extrema. Aceptar el cambio*. (F. J. Zapata, Trad.) Addison-Wesley: Pearson Education.
- Cantone, D. (2006). *La Biblia del Programador Implementación y Debugging*. Buenos Aires: MP Ediciones.
- Deitel, P., & Deitel, H. (2012). *Como programar en Java* (Novena ed.). Mexico: Pearson Educación.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: Marcombo.
- Groussard, T. (2012). *Java 7: Los fundamentos del lenguaje Java*. Barcelona: Ediciones ENI.
- Hernández, J. (2014). *Análisis y Desarrollo Web*.
- INTERNET YA – Soluciones Web. (s.f.). *INTERNET YA*. Recuperado el 7 de Abril de 2015, de <http://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/>
- Jaramillo, S., Cardona, S. A., & Villa, D. A. (2008). *Programación Avanzada en Java*. Quindio: Elizcom.
- Java. (s.f.). *¿Dónde obtener información técnica de Java?* Recuperado el 17 de Abril de 2015, de Java: <https://www.java.com/es/download/faq/techinfo.xml>
- Java. (s.f.). *¿Qué es Java y para qué es necesario?* Recuperado el 17 de Abril de 2015, de Java: http://www.java.com/es/download/faq/whatis_java.xml
- Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme Programming Installed*. Boston: Addison-Wesley.
- Kendall, K., & Kendall, J. (2005). *Análisis y Diseño de Sistemas* (Sexta ed.). (A. Nuñez, Trad.) Mexico: Pearson Educación,.

- Manifiesto Ágil. (2001). *Manifiesto por el Desarrollo Ágil de Software*. Recuperado el 10 de Abril de 2015, de Manifiesto Ágil: <http://agilemanifesto.org/iso/es/manifesto.html>
- Manifiesto Ágil. (2001). *Principios del Manifiesto Ágil*. Recuperado el 10 de Abril de 2015, de Manifiesto Ágil: <http://agilemanifesto.org/iso/es/principles.html>
- Martinez, R. (2 de Octubre de 2010). *Sobre PostgreSQL*. Recuperado el 13 de Abril de 2015, de PostgreSQL: http://www.postgresql.org.es/sobre_postgresql
- Ordax, J. M., & Ocaña, P. A. (2012). *Programación web en Java*. Barcelona: Ministerio de Educación de España.
- Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2015). *Curso de Ingeniería de Software*. IT Campus Academy.
- Raya, R. (24 de Julio de 2014). *¿Qué son las metodologías ágiles?* Recuperado el 8 de Abril de 2015, de LeanMonitor: <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>
- Sánchez, M. Á. (2001). *JavaScript*. Antequera: Innovación y Cualificación.
- Schwaber, K., Sutherland, J., & Beedle, M. (2001). *Agile Software Development with SCRUM*. Nueva Jersey: Prentice Hall.
- Universidad de Alicante. (s.f.). *AJAX*. Recuperado el 8 de Mayo de 2015, de Servicio de Informática: <http://si.ua.es/es/documentacion/mootools/ajax.html>
- Van Lancker, L. (2012). *HTML 5: Los fundamentos del lenguaje*. Barcelona: Ediciones ENI.


ANEXOS

A. Documentos

RESOLUCION N° 218-2016-C.O/UNACH



Universidad Nacional Autónoma de Chota
"Año de la Consolidación del Mar de Grau"



RESOLUCIÓN N° 218-2016-C.O/UNACH
Chota, 17 de mayo de 2016
-1-

El Fedatario de la Universidad Nacional Autónoma de Chota CERTIFICA: Que la presente Resolución corresponde a su original que se puso a la vista Chota **17 MAYO 2016**

VISTO:
Carta N° 110A-2016-UNACH/P, de fecha 17 de mayo de 2016, Acta de Sesión Extraordinaria número Catorce (014), de fecha 17 de mayo del año 2016, y,

CONSIDERANDO:

Que, de conformidad con el artículo 18° de la Constitución Política del Perú, La universidad es la comunidad de profesores, alumnos y graduados. (...). Cada universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico. Las universidades se rigen por sus propios estatutos en el marco de la Constitución y de las leyes.

Que, la Ley Universitaria, Ley N° 30220, en su artículo 8° establece que, El Estado reconoce la autonomía universitaria. La autonomía inherente a las universidades se ejerce de conformidad con lo establecido en la Constitución, la presente ley y demás normas aplicables. Esta autonomía se manifiesta en los siguientes regímenes 8.4) Normativo, implica la potestad autodeterminativa para la creación de normas internas (estatuto y reglamentos), destinadas a regular la institución universitaria.

Que, el artículo 29° de la Ley Universitaria, referido a la Comisión Organizadora establece que, La Comisión tiene a su cargo la aprobación del Estatuto, reglamentos y documentos de gestión académica y administrativa de la universidad, formulados en los instrumentos de planeamiento, así como en su conducción y dirección hasta que se constituyan los órganos de gobierno que, de acuerdo a la presente ley, le correspondan.

Que, mediante Ley N° 29531 se creó la Universidad Nacional Autónoma de Chota como persona jurídica de derecho público interno, con sede en la provincia de Chota, departamento de Cajamarca.

Que, a través de la Resolución Viceministerial N° 043-2015-MINEDU, de fecha 13 de agosto del año 2015, se resolvió constituir la Comisión Organizadora de la Universidad Nacional Autónoma de Chota, la misma que está integrada por: FLORENCIA ADELINA ARTEAGA TORRES, presidente; EDIN EDGARDO ALVA PLASENCIA, Vicepresidente Académico; y, EDGAR CARLOS QUISPE PEÑA, Vicepresidente de Investigación.

Que, mediante Carta N° 110A-2016-UNACH/P, de fecha 17 de mayo de 2016, emitido por la Presidenta de la Comisión Organizadora de la Universidad Nacional Autónoma de Chota, Dra. Florencia Adelina Arteaga Torres, hace llegar el Estatuto de la UNACH 2016, para su revisión y aprobación en Consejo Universitario.

Que, en Sesión Extraordinaria número Catorce (014), de fecha 17 de abril del año 2016, la Comisión Organizadora de la Universidad Nacional Autónoma de Chota, mediante acuerdo de consejo N° 291-C.O/UNACH, aprueba el Estatuto de la Universidad Nacional Autónoma de Chota, el mismo que consta de quince (15) capítulos, doscientos ochenta (280) artículos, ocho (8) disposiciones finales y transitorias y dos (2) disposiciones complementarias.

Que, de conformidad con los artículos 29° y 62° de la Ley Universitaria, Ley N° 30220 y el artículo 53° del Estatuto de la Universidad Nacional Autónoma de Chota.

Sede Central: Jirón José Osorio N° 418 – Sede Administrativa: Jirón Gregorio Malca N° 575 – Chota – Telef. 078-351164

Figura A.1: Resolución N° 218-2016-C.O/UNACH - Pág. 1



RESOLUCIÓN N° 218-2016-C.O./UNACH

Chota, 17 de mayo de 2016

-2-

SE RESUELVE:

ARTÍCULO PRIMERO: APROBAR el Estatuto de la Universidad Nacional Autónoma de Chota, el mismo que consta de quince (15) capítulos, doscientos ochenta (280) artículos, ocho (8) disposiciones finales y transitorias y dos (2) disposiciones complementarias.

ARTÍCULO SEGUNDO: DISPONER la publicación de la presente Resolución en el Diario Oficial "El Peruano" y el Estatuto de la Universidad Nacional Autónoma de Chota, en el portal institucional de la Universidad (www.unach.edu.pe).

ARTÍCULO TERCERO: NOTIFIQUESE a las autoridades, oficinas correspondientes de la Universidad Nacional Autónoma de Chota, al Ministerio de Educación (MINEDU) y a la Superintendencia Nacional de Educación Superior Universitaria (SUNEDU).

REGÍSTRESE, COMUNÍQUESE, CÚMPLASE Y ARCHÍVESE.


Dra. FLORENCIA ACUÑA ARTEAGA TORRES
PRESIDENTA
COMISIÓN ORGANIZADORA
UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA



Abog. Arnulfo Bustamante Mejía
SECRETARIO GENERAL - UNACH

El Fedatario de la Universidad Nacional Autónoma de Chota CERTIFICA: Que la presente Resolución corresponde a su original que se auto a la Chota Chota 17 MAYO 2016



Abog. Arnulfo Bustamante Mejía
SECRETARIO GENERAL - UNACH

C.c
Archivo

Figura A.2: Resolución N° 218-2016-C.O./UNACH Pág. 2



UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA

"Un sueño hecho realidad"

LA PRESIDENTA DE LA COMISIÓN ORGANIZADORA DE LA UNIVERSIDAD
NACIONAL AUTÓNOMA DE CHOTA, QUE SUSCRIBE:

CERTIFICA:

Que, los Bachilleres **KATHERINE VANESSA QUINTOS ESTRADA Y LUIS EDGARDO PÉREZ BONILLA**, egresados de la Escuela Profesional de Ingeniería en Computación e Informática de la Universidad Nacional Pedro Ruiz Gallo, han presentado y ejecutado el proyecto de tesis titulado "SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCIÓN DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA - 2015"; la Oficina General de Informática de la Universidad Nacional Autónoma de Chota, ha verificado y aprobado que cumple con el correcto funcionamiento del sistema web.

Se expide el presente documento a solicitud del interesado para los fines que estime conveniente.


Dra. FLORENCIA ADELINA ARTEAGA TORRES
PRESIDENTA
COMISION ORGANIZADORA
UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA

C.C.
Interesados
Archivo

Ciudad Universitaria: Colpamatará
Oficina Central: Jr. José Osorio N° 418
Oficina Administrativa: Jr. Gregorio Malca N° 875

Universidad Nacional Autónoma de Chota
Chota -- Perú

www.unach.edu.pe
informes@unach.edu.pe
☎ 076 - 351144

Figura A.3: Certificado de aprobación del sistema web otorgado por la Universidad Nacional Autónoma de Chota



UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA

"AÑO DE LA CONSOLIDACIÓN DEL MAR DE GRAU"

INFORME N° 043-2016-OIS-UNACH/.

A : Dra. Florencia Adelina Arteaga Torres.
Presidenta de la Comisión Organizadora - UNACH

De : Johan Jeampier Briones Delgado
Unidad de Soporte Informático -OGI - UNACH

ASUNTO : INFORMAR SOBRE SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE
INSCRIPCION DE POSTULANTES DE LA UNIVERSIDAD NACIONAL
AUTÓNOMA DE CHOTA - 2015

FECHA : 12 de julio de 2016.

El Redactor de la Universidad Nacional Autónoma de Chota CERTIFICA: Que la presente Fotocopia corresponde a su original que se puso a la vista Chota 25 JUL 2016



Es grato dirigirme al despacho de su digno cargo con la finalidad de expresarle un afectuoso saludo y deseos de éxito en su labor cotidiana, a la vez informarle sobre el Sistema Web presentado por la Bach. Katherine Vanessa Quintos Estrada identificada con DNI N° 46660168 y el Bach. Luis Edgardo Pérez Bonilla identificado con DNI N° 43744356, egresados de la Universidad Nacional Pedro Ruiz Gallo de la Escuela Profesional de Ingeniería en Computación e Informática, los que solicitaron la verificación y aprobación del Sistema Web denominado "SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCION DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA - 2015", el cual fue probado el día 01 de julio de 2016 en las computadoras designadas para la Oficina de Admisión.

Dicho sistema fue presentado y expuesto de forma detallada por los tesisistas, en cuanto a su funcionamiento, de igual forma indicando el correcto manejo del sistema ya que el mismo cuenta con los siguientes procedimientos:

1. **Registro de datos del postulante.** En este procedimiento se registra de forma detallada los datos del postulante, como nombres y apellidos, lugar de nacimiento, residencia actual, datos del apoderado, de igual forma se registra la Información Académica, y registro de información según la modalidad de postulación.
2. **Sistema Administrativo.** El sistema también cuenta con una interfaz en la que se iniciará realizando lo siguiente:
 - Registrando al personal que tendrá el trabajo de registrar información adicional e importante para el registro del postulante, en roles como personal para el registro de fotos, personal para el registro de Huellas, y personal para carnet.
 - Registrar los diferentes tipos de modalidades de postulación e indicando el tipo de examen, ya siendo Extraordinario u Ordinario.
 - Registro de las Escuelas Profesionales.
 - Registrar el tarifario en donde se ingresaran los montos en soles por derecho de inscripción indicando la modalidad y la procedencia académica del postulante, cabe recalcar que los montos varían de acuerdo al tipo de examen y modalidad.
 - Registrar el Proceso de Admisión, indicando su fecha de inicio del proceso y la fecha en la que finalizara, cabe recalcar que en este procedimiento el sistema será

Jr.: Gregorio Malca N° 857 - Chota - Tel: 076-351144

Página Web: www.unach.edu.pe

UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA
- UNACH -
SECRETARÍA PRESIDENCIAL

Fecha: 12-07-16 Anexos: 2-04-05

Firma: [Firma] Hora: 9:54 am

R. 17 16

Figura A.4: Informe sobre el sistema para el proceso de inscripción de postulantes enviado por la Unidad de Soporte Informático a la presidenta de la Comisión Organizadora - UNACH. Pág. 1.



UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA

"AÑO DE LA CONSOLIDACIÓN DEL MAR DE GRAU"

estricto, ya que al llegar a la fecha final el sistema desactivara las opciones del proceso,


- Registro de fotos, aquí el personal a cargo tendrá la facilidad de tomar la foto y el sistema reconocerá y ajustara la foto del postulante de acuerdo a las características indicadas por parte de la oficina de admisión.
- Registro de huellas, en esta interfaz el personal a cargo registrara las huellas de cada postulante ya que es importante para llevar a cabo el proceso.
- Impresión de carnet, aquí el personal a cargo tendrá la facilidad de imprimir los carnets de los postulantes.
- Modificación de información del postulante, en esta interfaz el personal a cargo tendrá la posibilidad de modificar alguna información del postulante mal ingresado.
- Asignación de permisos, en este apartado se le dará los permisos necesarios a cada personal a cargo y así evitar dificultades al momento de desarrollar el proceso.
- Cambio de modalidad, aquí el personal a cargo tendrá la posibilidad de cambiar de modalidad extraordinaria a ordinaria siempre y cuando el postulante no alla aprobado el examen extraordinario, cabe recalcar que es muy importante el canje de carnet de postulante extraordinario a ordinario.

Todo lo explicado anteriormente se ha verificado por parte de la Oficina General de Informática, comprobando así el correcto funcionamiento del Sistema Web, de igual forma en el proceso de prueba del sistema se ha indicado algunas sugerencias y actualizaciones que el sistema podría tener en el futuro. De igual forma viendo que el sistema cumple con la toma de información necesaria para el proceso de admisión y de igual manera presenta procedimientos novedosos como la toma y ajuste de la foto del postulante, la toma de huellas y la impresión de los carnets, la Oficina General de Informática propone su aplicación de dicho sistema en los futuros procesos de admisión realizados por la Universidad.

Por lo antes ya expuesto la Oficina General de Informática da la VERIFICACION Y APROBACION DEL FUNCIONAMIENTO DE SISTEMA WEB DENOMINADO "SISTEMA BAJO ENTORNO WEB PARA EL PROCESO DE INSCRIPCION DE POSTULANTES DE LA UNIVERSIDAD NACIONAL AUTÓNOMA DE CHOTA - 2015" presentado por los jóvenes egresados de la Universidad Nacional Pedro Ruiz Gallo.

Sin otro particular, me despido no sin antes reiterarle los sentimientos de mi más especial consideración y estima personal.

El Rector de la Universidad Nacional Autónoma de Chota CERTIFICA Que la presente Fotocopia corresponde a su original que se puso a la vista Chota 25 JUL 2016


Johan Jeampier Briones Delgado
Unidad De Soporte Informático - OGI - UNACH



Juan Carlos Bustamante Mejía
SECRETARIO GENERAL - UNACH

Jr.: Gregorio Malca N° 857 - Chota - Tel: 076-351144
Página Web: www.unach.edu.pe

Figura A.5: Informe sobre el sistema para el proceso de inscripción de postulantes enviado por la Unidad de Soporte Informático a la presidenta de la Comisión Organizadora - UNACH. Pág. 2.

B. Instalación y configuración de Apache Tomcat

Pasos Previos

Instalación y configuración del JDK

1. Descargamos el archivo en la siguiente dirección

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>. .

Aceptamos el Accept License Agreement, y dependiendo si nuestra maquina es de 32 o 64 bits, deberán optar por **jdk-7u79-windows-i586.exe** o **jdk-7u67-windows-x64.exe**.

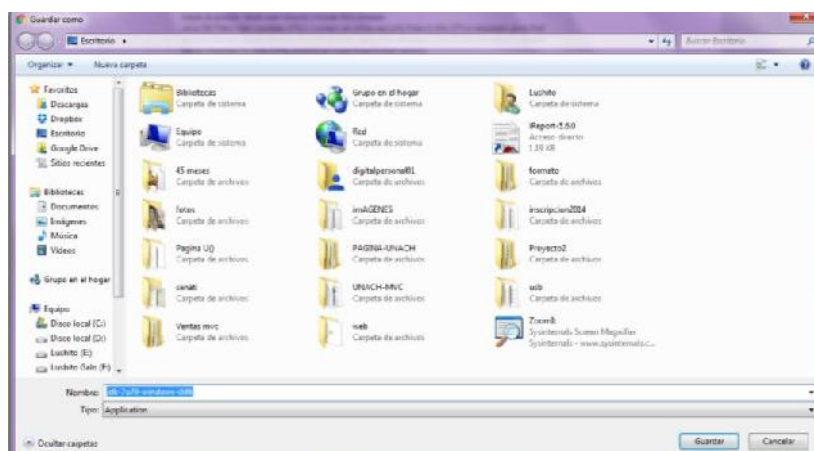
Looking for JDK on ARM?
JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

Java SE Development Kit 7u79
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	148.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	16.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

Java SE Development Kit 7u79 Demos and Samples Downloads
You must accept the Oracle BSD License to download this software.

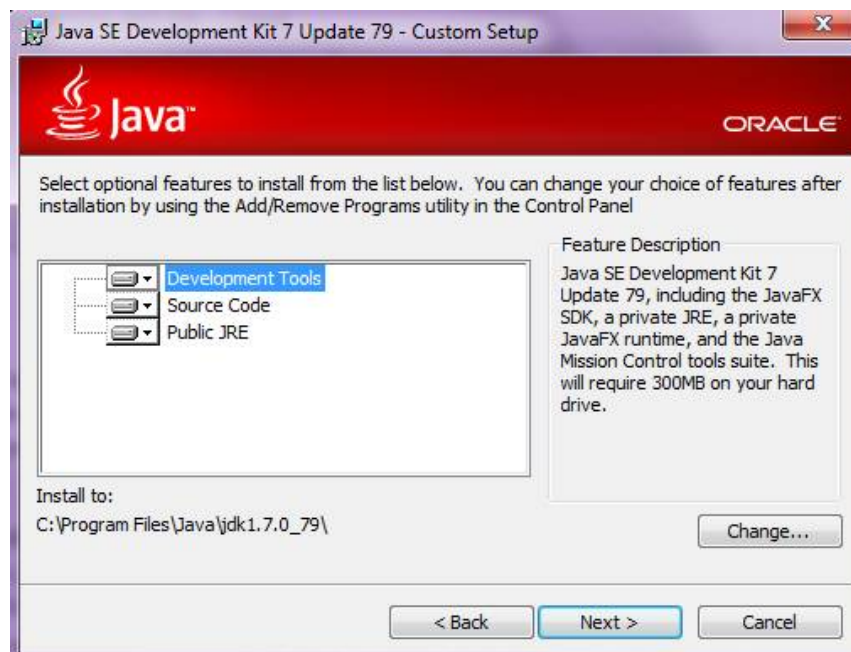
En nuestro caso descargamos la versión correspondiente a nuestro sistema e iniciamos la descarga.



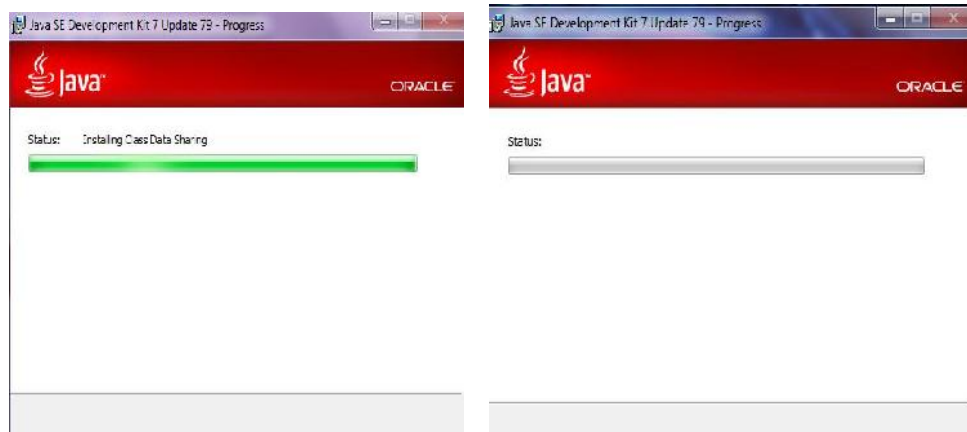
2. Ejecutamos el archivo descargado y seguimos la secuencia de instalación de java.



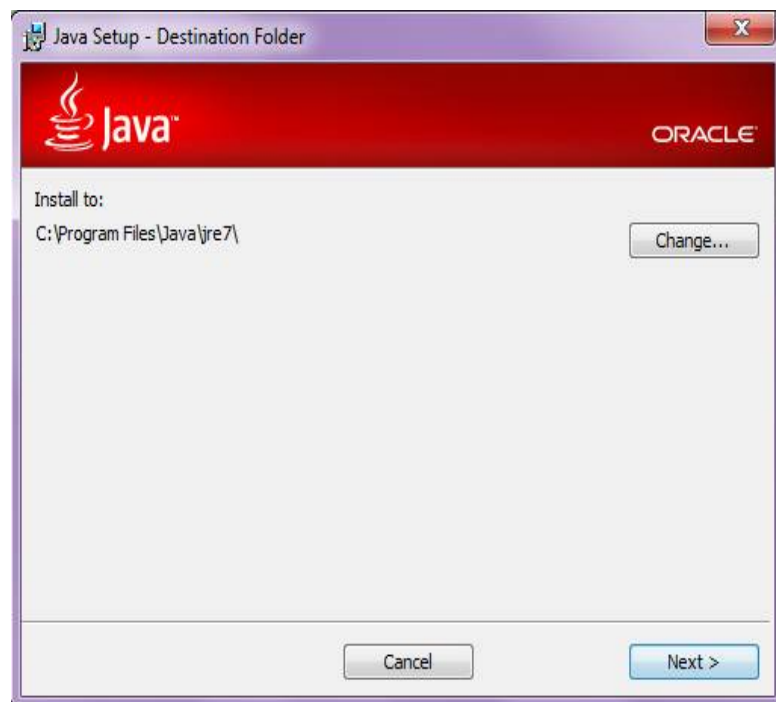
3. En esta ventana nos muestra las características que deseamos instalar, en nuestro caso no modificaremos nada y pulsaremos "next".



4. Esperamos unos momentos mientras termina el proceso de instalación.



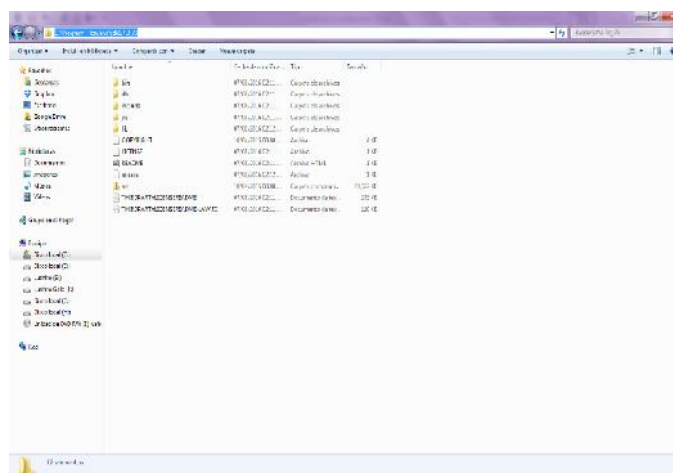
5. Se visualiza otra ventana llamada "Java Setup" donde se nos menciona la ruta que tendrá el JRE7, damos click en el botón Next.



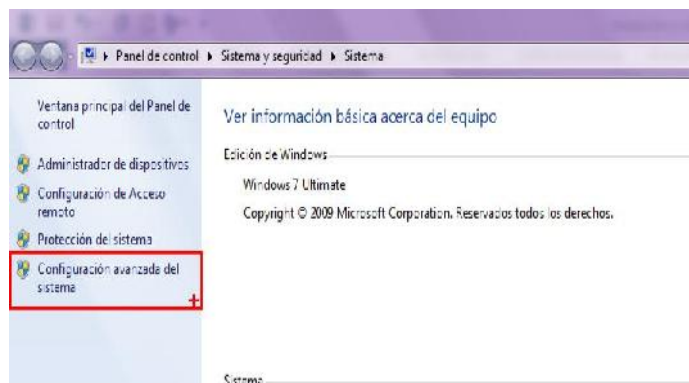
6. Se visualiza la ventana "Java Setup - Progress" donde vemos el progreso de la instalación del JRE7.



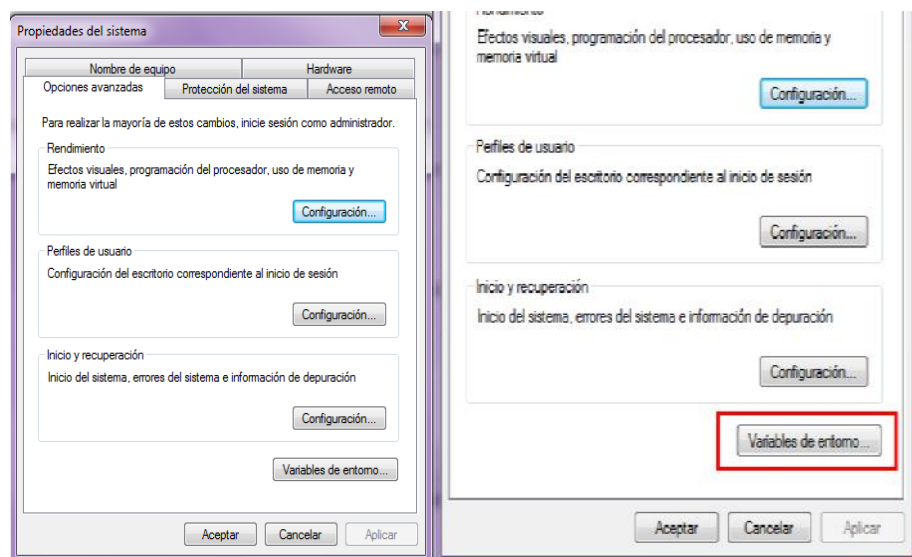
7. Nos aparece finalmente la ventana "Java SE Development Kit 7 Update 79 - Complete", donde nos indica que se ha instalado satisfactoriamente, damos click en el botón [Close].
8. Después de la instalación del JDK 7, verificamos la ruta donde se instaló el software y copiamos la ruta en nuestro caso se instaló en la siguiente dirección "C:\Program Files\Java\jdk1.7.0_79".



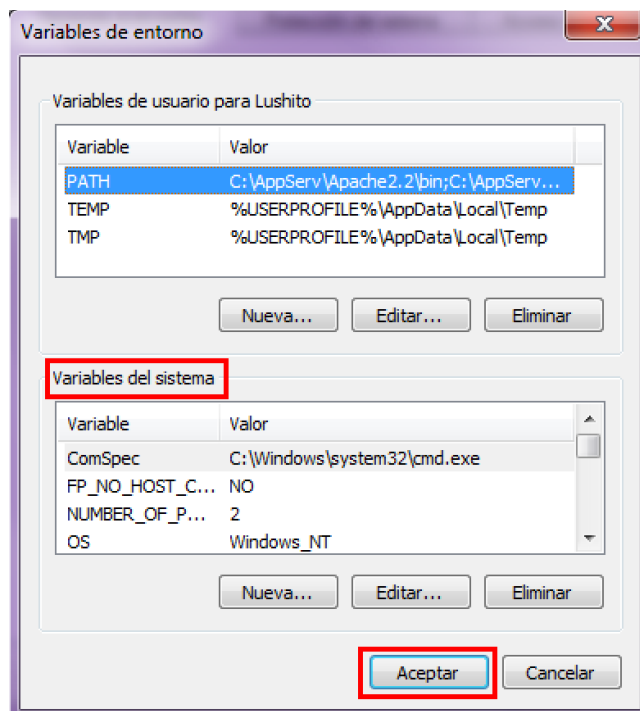
9. El siguiente paso será declarar las variables de entorno JAVA_HOME y modificar la variable Path, para esto hacemos click derecho sobre Equipo y luego click en Propiedades, nos muestra una ventana de información básica del equipo y en el lado izquierdo hacemos click en Configuración avanzada del sistema.



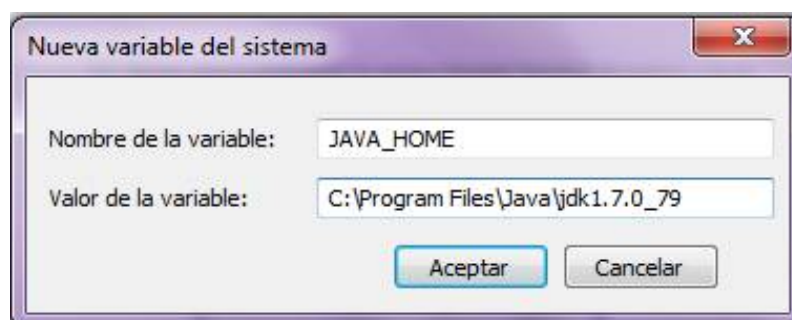
10. Se abre la ventana llamada Propiedades del sistema, nos dirigimos a la pestaña **Opciones avanzadas** y hacemos click en el botón - **Variables de entorno**.



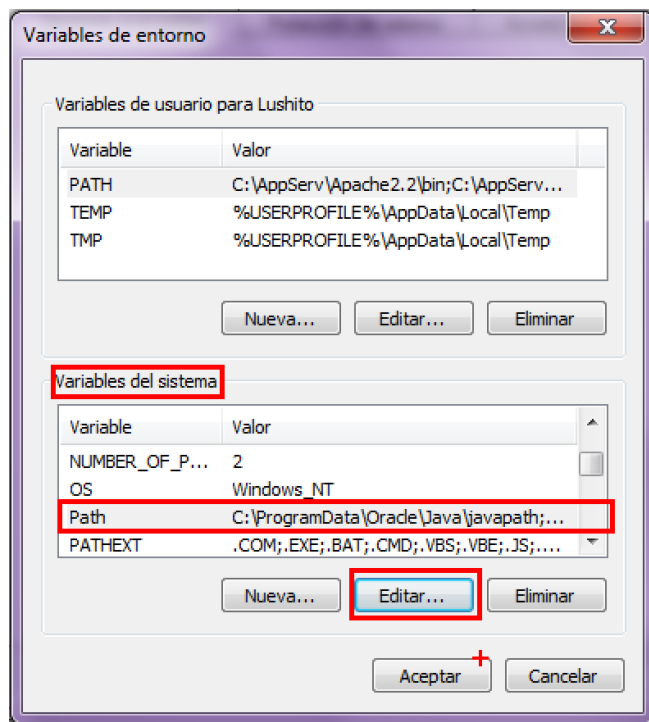
11. A continuación se abre la ventana llamada Variables de entorno, y en el apartado **Variables del sistema** hacemos click en el botón **Nueva**.



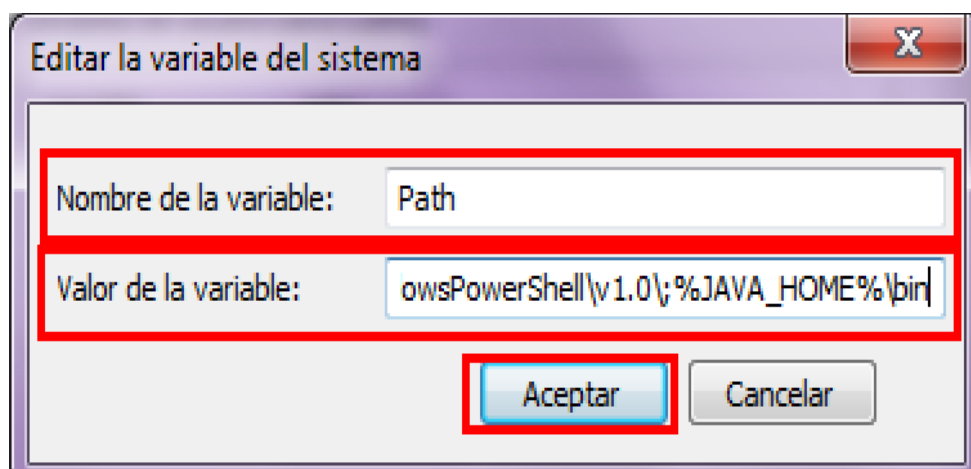
12. En la ventana llamada Nueva variable del sistema ponemos en el **campo Nombre de la variable** el texto "**JAVA_HOME**", y en el campo Valor de la variable ingresamos la ruta de nuestro JDK, que para este caso recordando el Paso 8, la ruta del JDK quedo en "C:\Program Files\Java\jdk1.7.0_79". Damos en el botón "Aceptar".



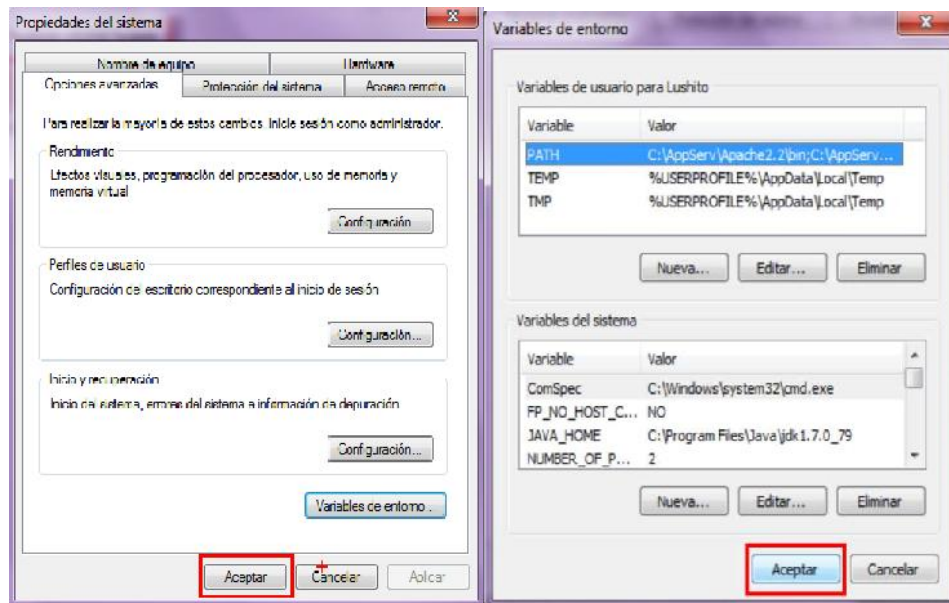
13. En esta ventana modificamos la Variable "Path"



14. Se abrirá la ventana llamada Editar la variable del sistema, y desde aquí el campo "**Nombre de la variable**" lo dejaremos tal como está, nos fijaremos en el campo "**Valor de la variable**", nos ubicamos al final del texto que contiene dicho campo y separado por punto y coma(;) ingresamos al final del texto "%JAVA_HOME%\bin". Luego click en el botón Aceptar.



15. Aceptamos los cambios



16. Ahora verificamos si las variables de entorno han sido configuradas correctamente, para esto, abrimos la ventana de “CMD”, e ingresamos “**java -version**” y damos ENTER.



17. Si todo ha sido configurado correctamente nos saldrá un mensaje indicándonos la versión de java e información adicional como se muestra en la imagen siguiente:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600.1]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Luchito>java -version
Error: no se ha encontrado o cargado la clase principal ?version

C:\Users\Luchito>java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (Build 1.0.0_45 b14)
Java HotSpot(TM) Client VM (build 25.45-b02, mixed mode, sharing)

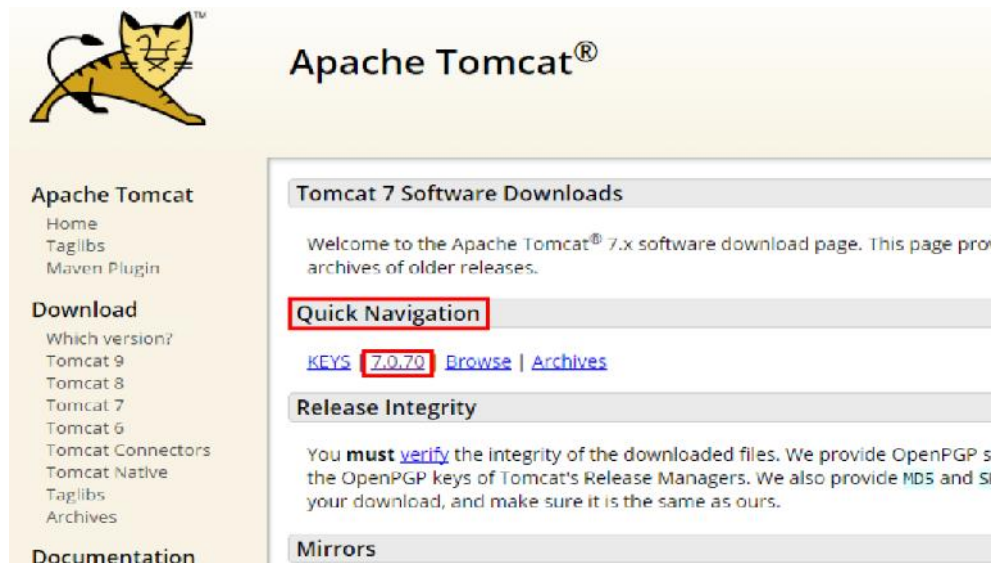
C:\Users\Luchito>
```

Instalación y configuración del Apache Tomcat

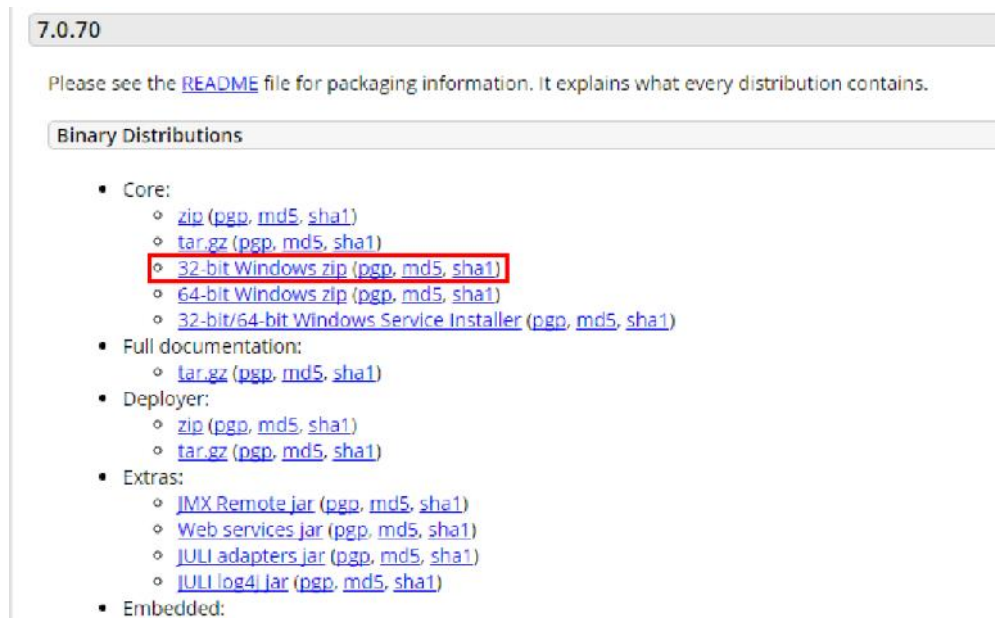
1. Lo primero que haremos es descargar Apache Tomcat 7, de la siguiente dirección: <http://tomcat.apache.org/download-70.cgi>.



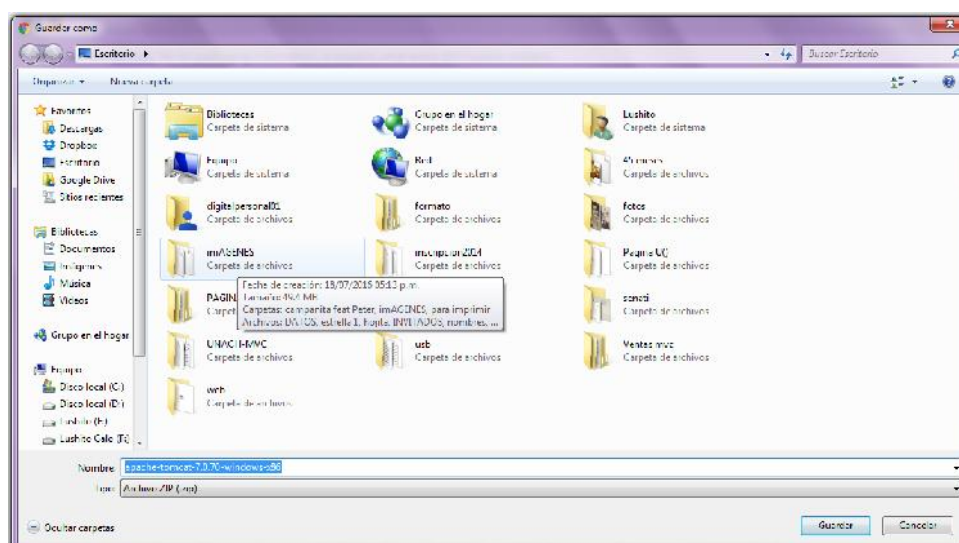
2. En el apartado Quick Navigation damos click en el link que dice 7.0.70, nos muestra un apartado dentro de la misma página.



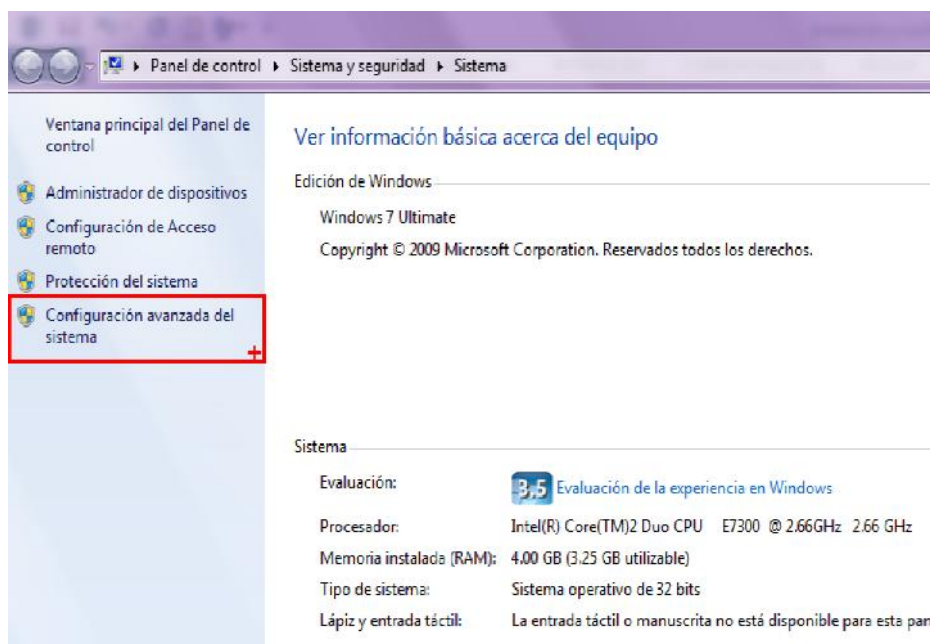
3. Dependiendo de la arquitectura del sistema operativo que tenemos (32 bits o 64 bits), descargamos el archivo correspondiente de la lista core 32-bit windows zip (pgp, md5, sha1).

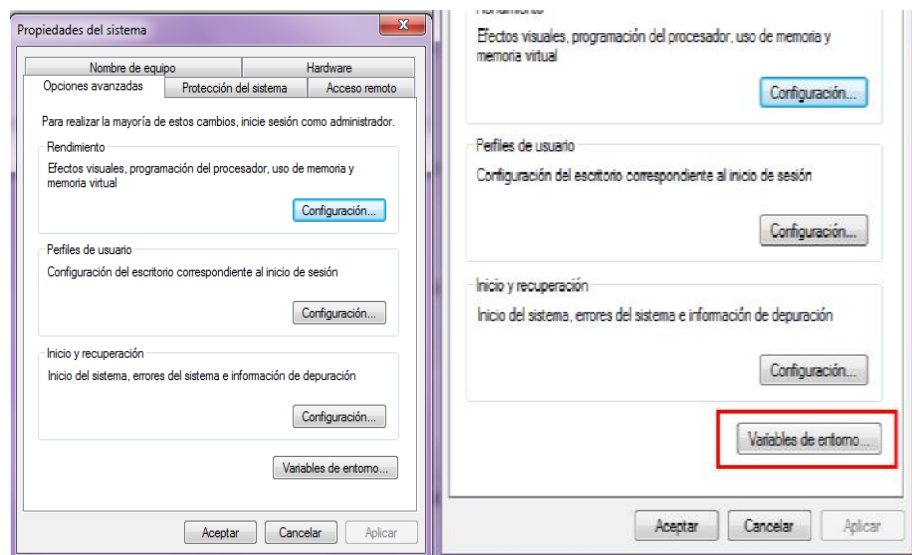


y guardaremos el archivo en una unidad de nuestro sistema.

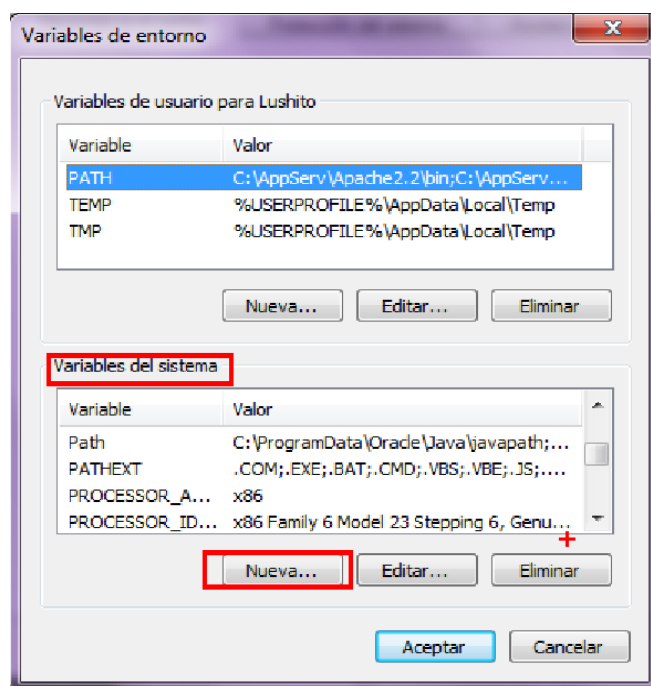


4. Antes de descomprimir el archivo creamos variable de entorno **CATALINA_HOME** y editamos las variables **CLASSPATH** y **Path**, para realizar esto hacemos click derecho sobre Equipo y luego sobre Propiedades.

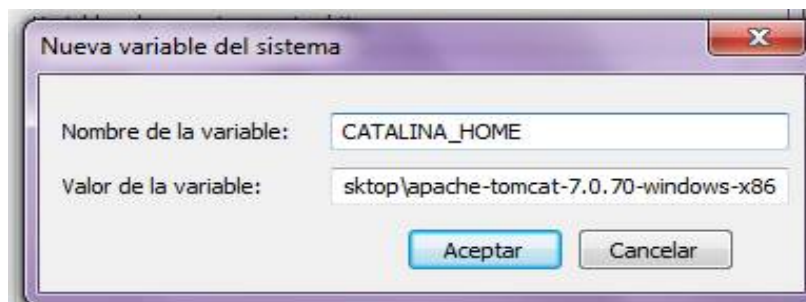




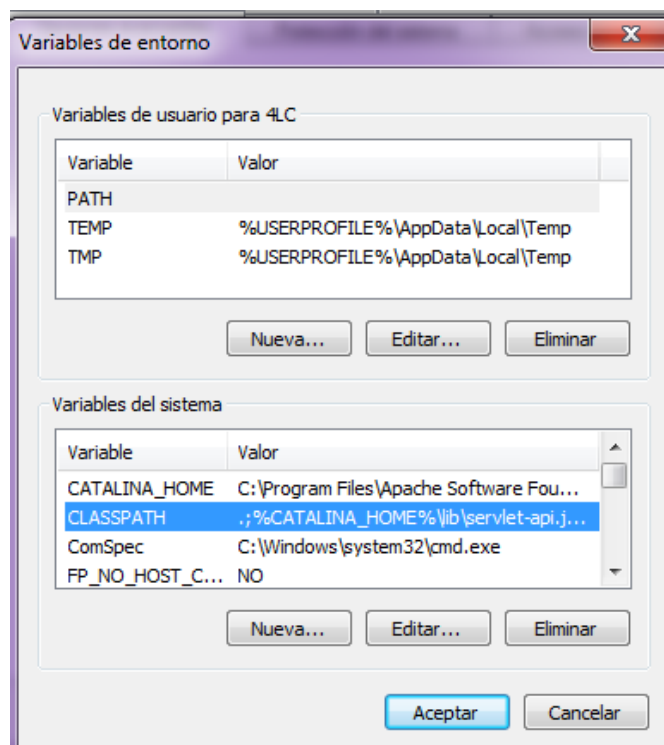
5. Se mostrará la ventana variables de entorno, desde el apartado de variables del sistema hacemos click en el botón Nueva.



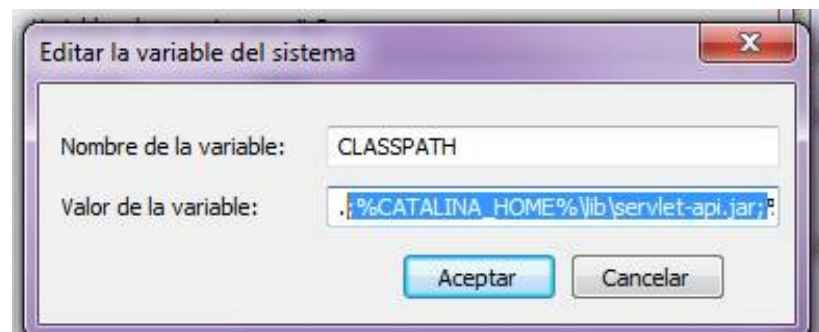
6. Se mostrará la ventana nueva variable del sistema, en el campo nombre de la variable ingresamos el texto **CATALINA_HOME**, y sobre el campo Valor de la variable ingresamos la ruta “C:\Users\Lushito\Desktop\apache-tomcat-7.0.70-windows-x86\apache-tomcat-7.0.70” y luego click en el botón Aceptar.



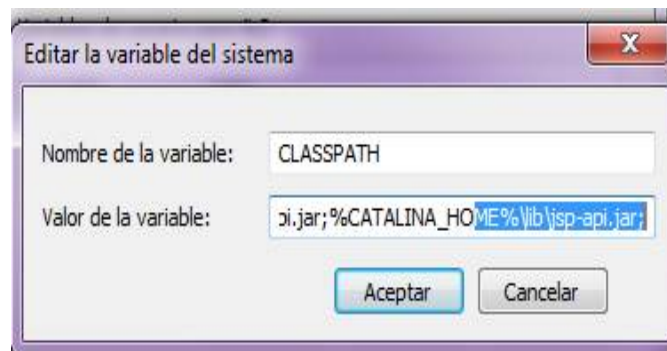
7. Editaremos la variable de entorno CLASSPATH , para realizar esto desde el apartado Variables del sistema, buscamos la variable CLASSPATH , la seleccionados y damos click en el botón Editar.



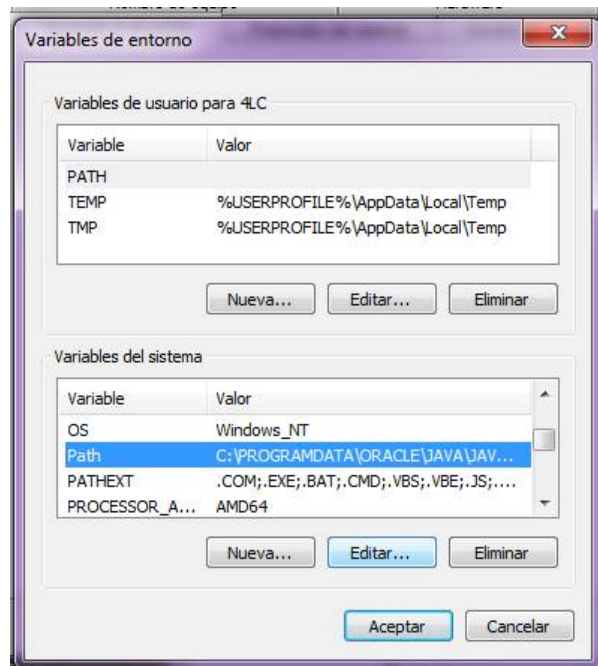
8. Se mostrará la ventana Editar la variable del sistema, y nos mostrara los valores actuales de la variable CLASSPATH, nos ubicamos en el campo Valor de la variable y nos desplazamos hasta el final, luego colocamos un punto y coma e ingresamos el texto **%CATALINA_HOME%\lib\servlet-api.jar;**



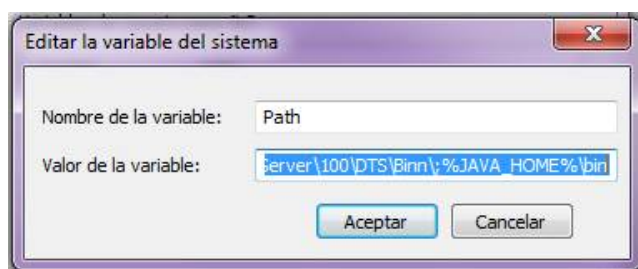
Ingresamos el texto **%CATALINA_HOME%\lib\jsp-api.jar;**. Después de haber agregado estos dos valores a la variable CLASSPATH, damos click en el botón Aceptar para que se guarden estos nuestros valores.



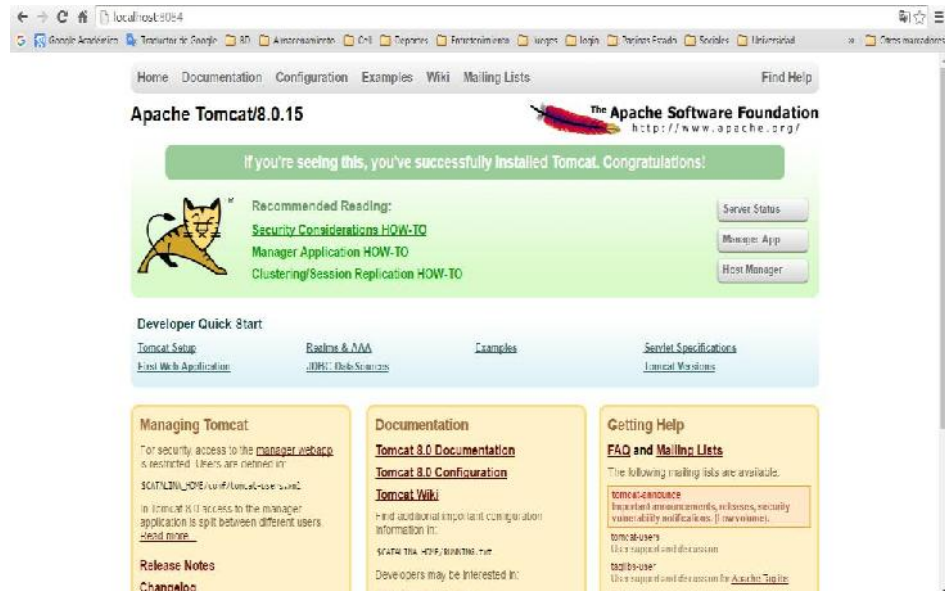
9. Ahora editaremos la variable de entorno Path, la seleccionamos y damos click en el botón Editar.



10. Aquí nos ubicaremos sobre el campo Valor de la variable, nos desplazaremos hasta el final del valor y colocaremos un punto y coma he ingresaremos el texto **%CATALINA_HOME%\bin** . Note que al final del valor de la variable Path quedo sin punto y coma. Damos click en el botón Aceptar.



11. Ahora el paso final consiste en abrir nuestro navegador y tecleamos en la barra de direcciones `http://localhost:8080` y se mostrara la ventana Apache Tomcat al cual nos muestra la pantalla de bienvenida de Apache Tomcat, esto quiere decir que todo se ha configurado exitosamente.



C. Instalación y Configuración del Lector de Huella Digital

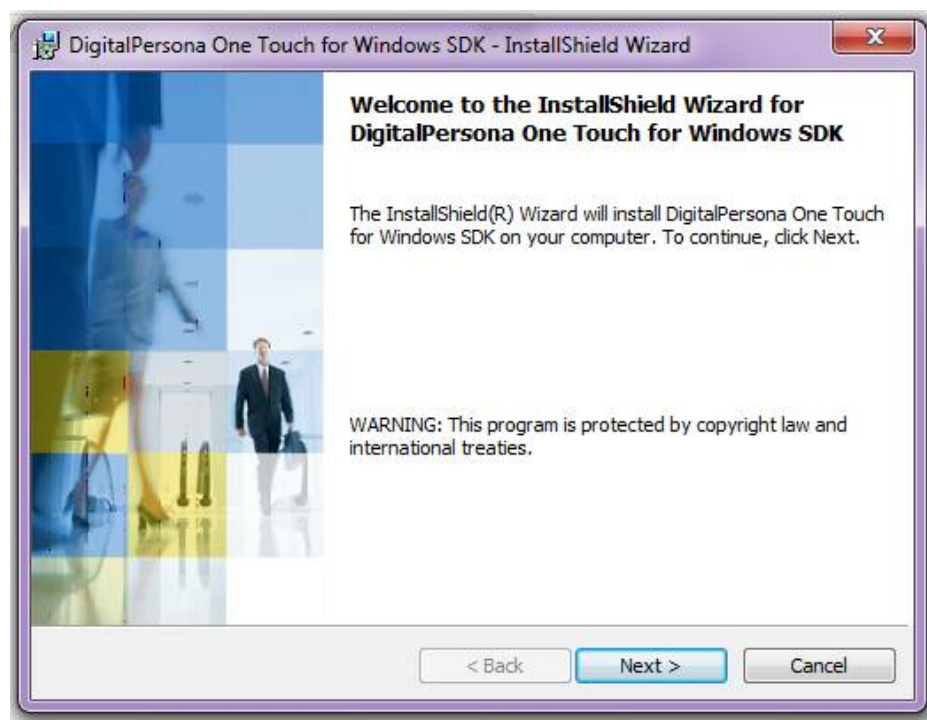
Instalación del Lector de Huella Digital

En el presente proyecto de tesis hemos utilizado un Lector de Huella Digital **DigitalPersona U.are.U 4000B**

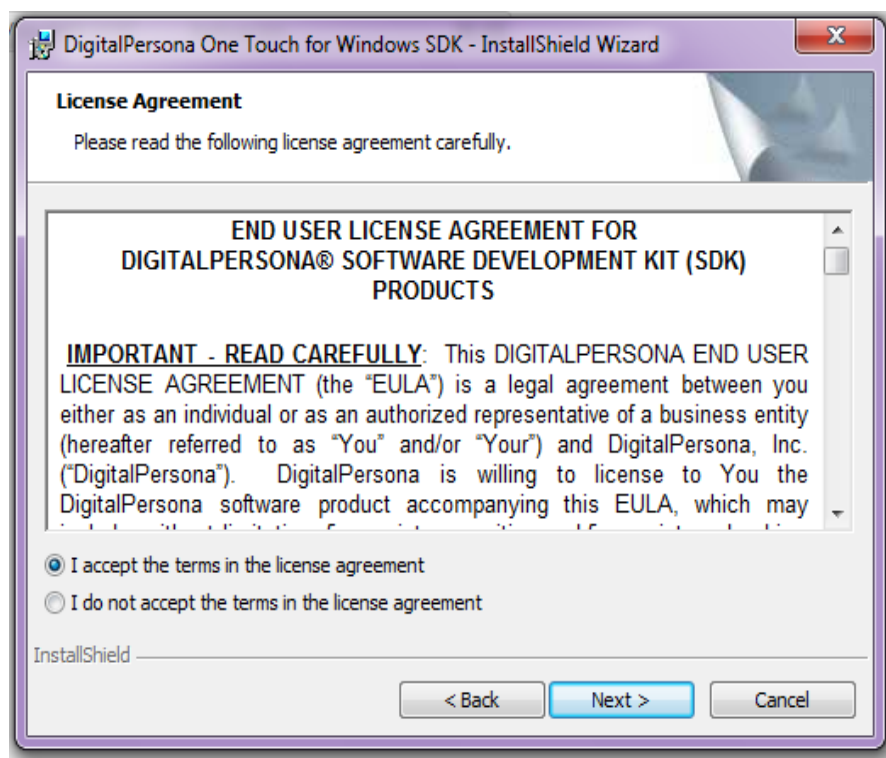


Para que el sistema pueda trabajar con este hardware debemos instalar el SDK con el que viene incluido.

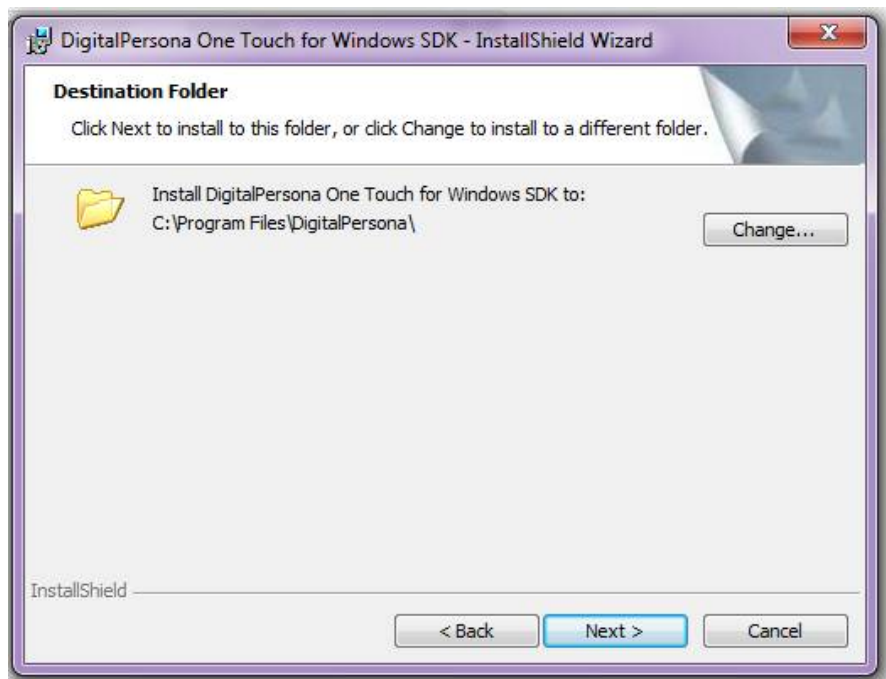
1. Ejecutamos el archivo setup.exe y nos aparecerá la siguiente ventana. Clic en el botón "Next".



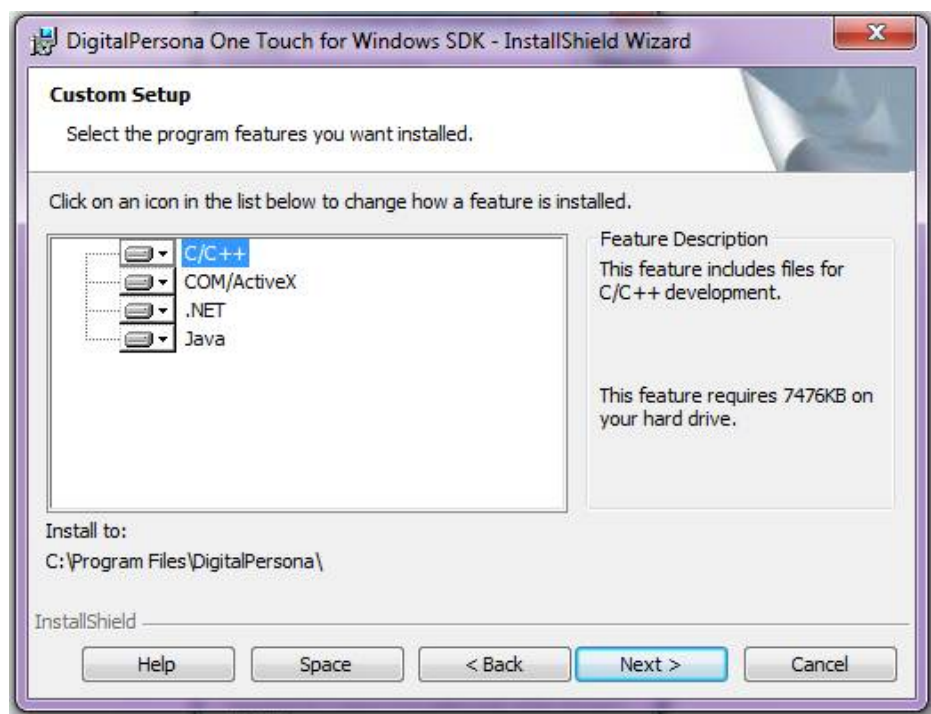
2. Seleccionamos la opción para aceptar los términos y condiciones. Clic en el botón “Next”.



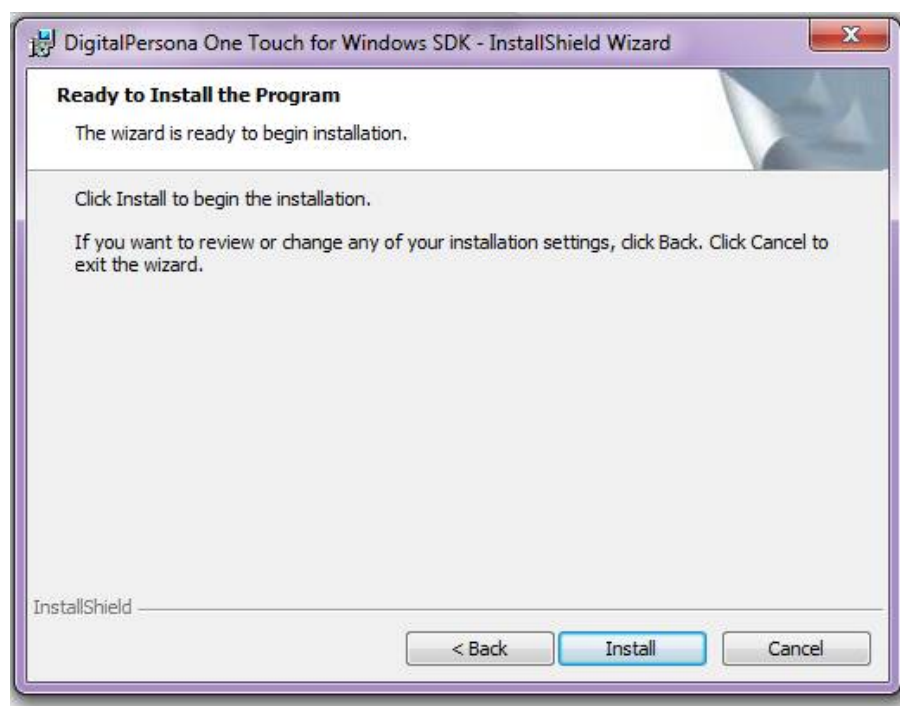
3. Seleccionamos la carpeta en donde deseamos que se instale el SDK o lo podemos dejar en la ruta predeterminada. Clic en el botón “Next”.



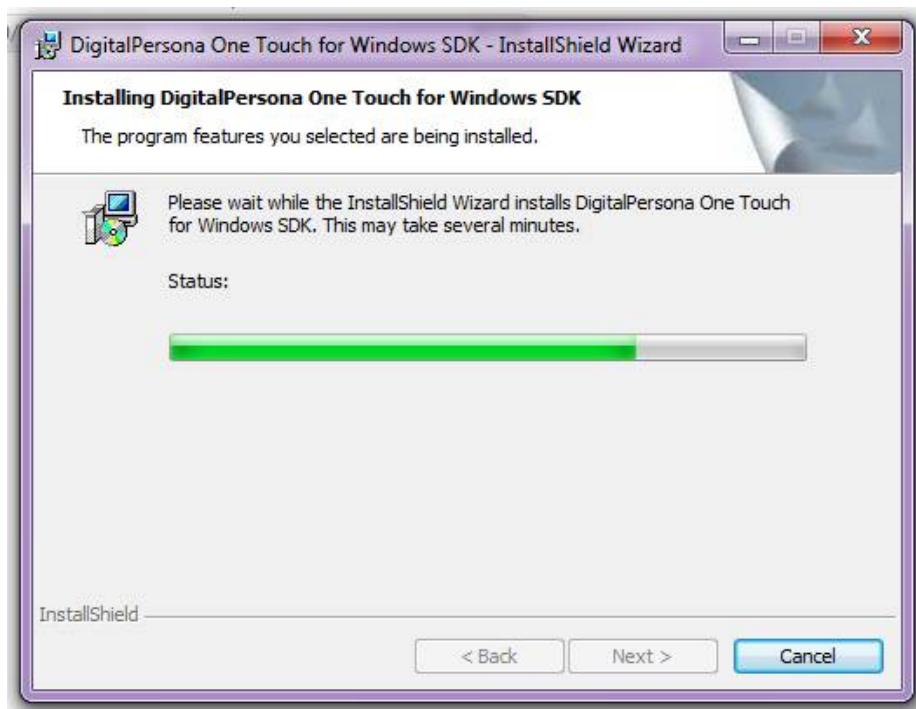
4. En la siguiente ventana aparecerá una lista con todos los componentes que se van a instalar. Clic en el botón “Next”.



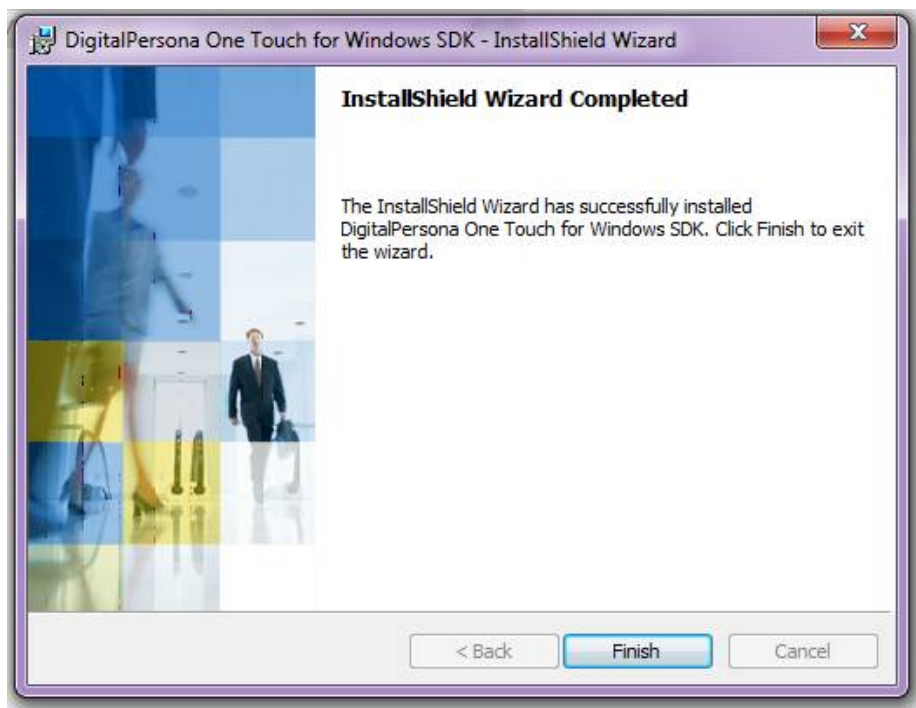
5. Nos aparecerá una ventana con la opción de instalar o retroceder para revisar las configuraciones ya realizadas. Clic en el botón “Install”.



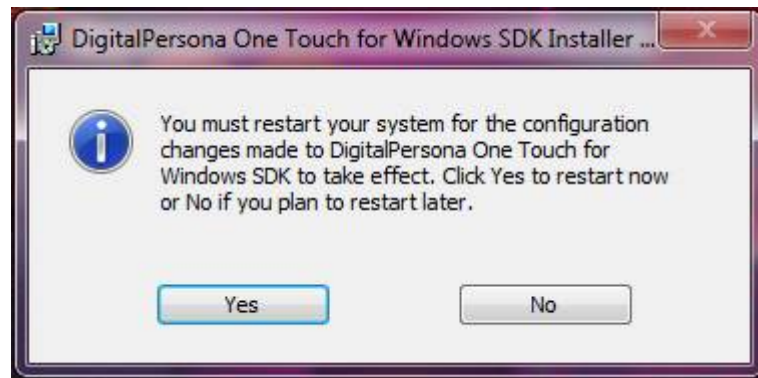
6. Esperamos a que termine de instalar el SDK en nuestra computadora



7. Una vez finalizado el proceso aparecerá una ventana avisándonos que la instalación se ha completado. Clic en el botón "Finish".



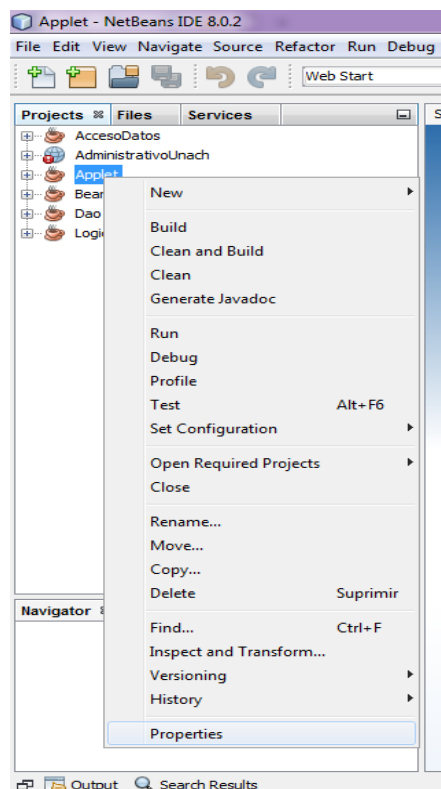
8. Aparecerá un mensaje indicándonos reiniciar nuestro sistema para que los cambios en la configuración se efectúen. Seleccionamos la opción “Yes”.



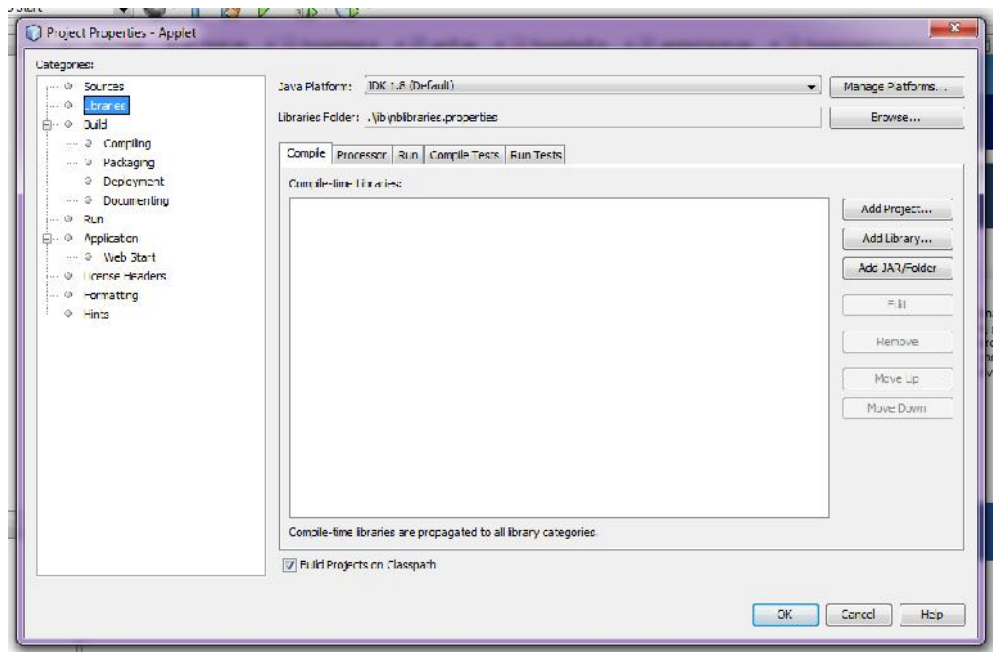
Agregar las librerías del Lector de Huella Digital

Una vez que hemos instalado el SDK, debemos agregar las librerías proporcionadas en nuestro applet.

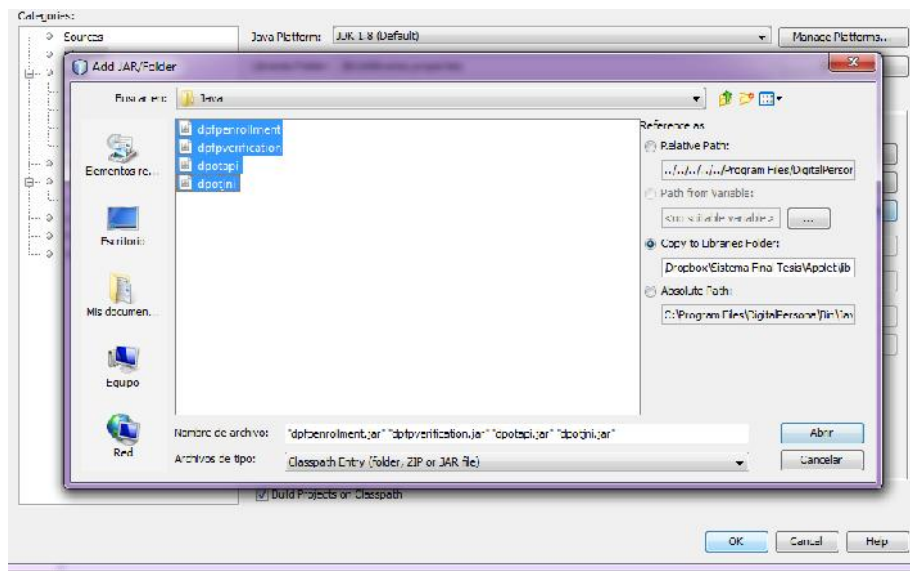
1. En Netbeans damos clic derecho en nuestro proyecto Applet y seleccionamos Properties (Propiedades).



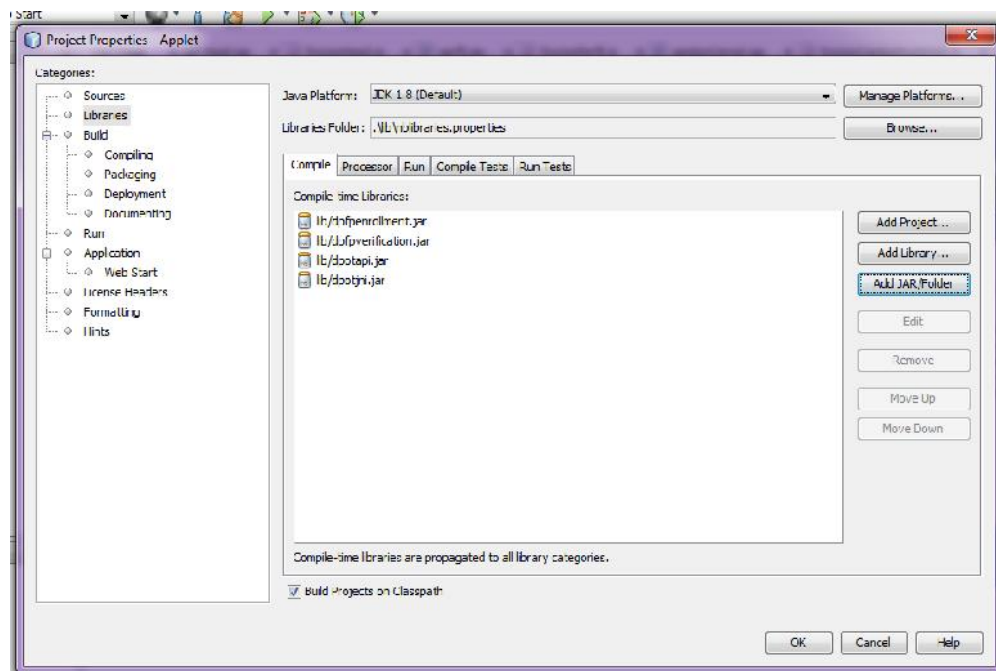
2. Aparecerá un cuadro, en la lista de la izquierda seleccionamos Libraries y luego Add JAR/Folder.



3. Nos dirigimos a la carpeta donde se encuentra instalado el SDK, en nuestro caso la ruta es: *C:\Program Files\DigitalPersona\Bin\Java*, ahí encontraremos cuatro archivos .jar, los seleccionamos todos y damos clic en el botón Abrir.

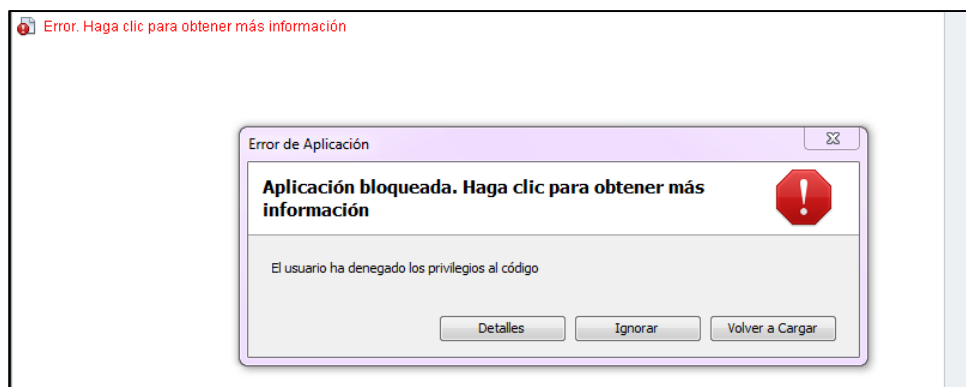
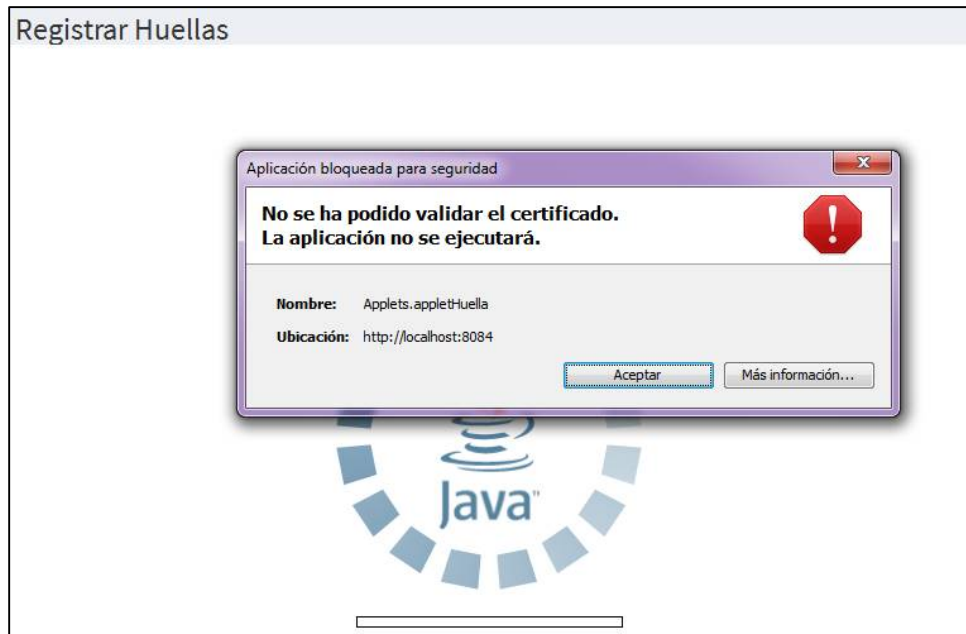


4. Aceptamos los cambios y ya podemos comenzar a programar en nuestro applet.



D. Permisos del Applet

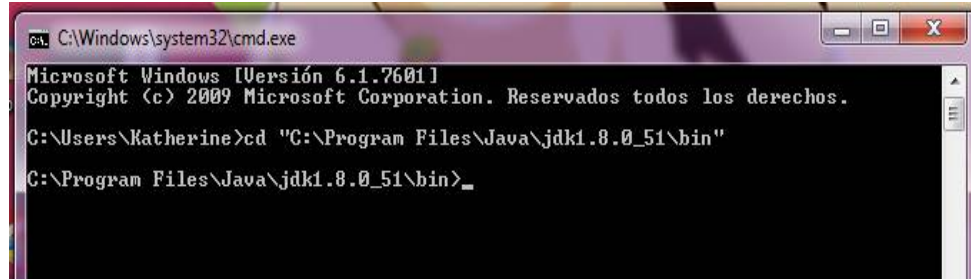
Un applet es una aplicación Java que se ejecuta localmente y por seguridad se le aplican muchas restricciones. En nuestro proyecto, cuando se ingresaba al applet de Captura de Huellas nos aparecían los siguientes mensajes de error:



Para evitar estos errores es necesario firmar el applet y agregarlo a la lista de sitios seguros de Java, y así cuando el navegador lo ejecute, lo reconocerá como una aplicación segura y nos permitirá trabajar con él.

Firmar un applet

1. Ingresamos a la consola de Windows (CMD) y ubicamos la ruta donde está instalado el jdk, y dentro de la carpeta bin, en nuestro caso: *C:\Program Files\Java\jdk1.8.0_51\bin*.

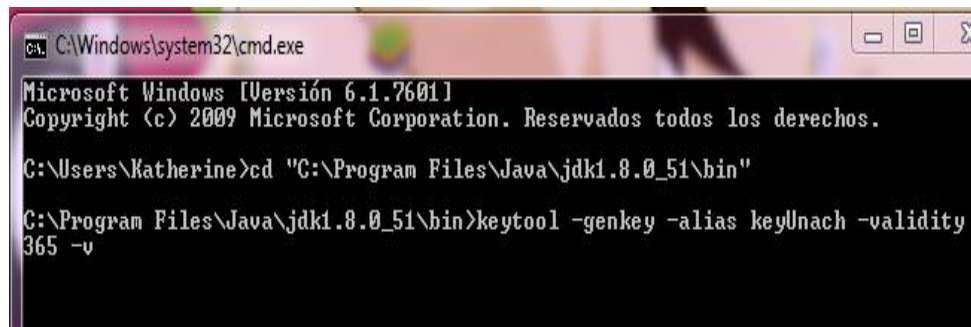


2. Ingresamos esta línea:

```
keytool -genkey -alias keyUnach -validity 365 -v
```

donde:

- keytool: es una herramienta proporcionada por Java para crear un almacén de claves
- genkey: es el comando para generar un certificado autofirmado.
- alias: es el nombre con que se identificará estas claves.
- validity: es el tiempo de validez, en días, que se asignará al certificado.



Luego ejecutamos el comando.

3. Nos pedirá que ingresemos una contraseña, la escribimos y damos Enter, y así con toda la información solicitada. Una vez finalizado se generará un archivo .keystore. En la última línea del proceso nos indicará la carpeta en donde se ha guardado.

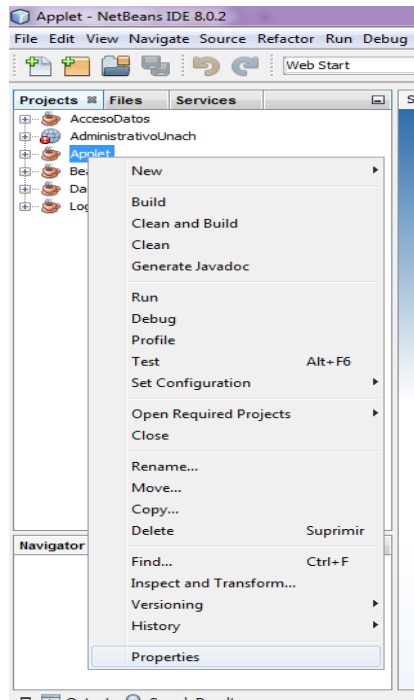
```
C:\Windows\system32\cmd.exe
C:\Users\Katherine>cd "C:\Program Files\Java\jdk1.8.0_51\bin"
C:\Program Files\Java\jdk1.8.0_51\bin>keytool -genkey -alias keyUnach -validity
365 -v
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: Universidad Nacional Autonoma de Chota
¿Cuál es el nombre de su unidad de organización?
[Unknown]: Oficina de Informatica
¿Cuál es el nombre de su organización?
[Unknown]: UNACH
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Cajamarca
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Chota
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: PE
¿Es correcto CN=Universidad Nacional Autonoma de Chota, OU=Oficina de Informatic
a, O=UNACH, L=Cajamarca, ST=Chota, C=PE?
[no]: si

Generando par de claves DSA de 1.024 bits para certificado autofirmado (SHA1with
DSA) con una validez de 365 días
para: CN=Universidad Nacional Autonoma de Chota, OU=Oficina de Informati
ca, O=UNACH, L=Cajamarca, ST=Chota, C=PE
Introduzca la contraseña de clave para <keyUnach>
<INTRO si es la misma contraseña que la del almacén de claves>:
[Almacenando C:\Users\Katherine\.keystore]

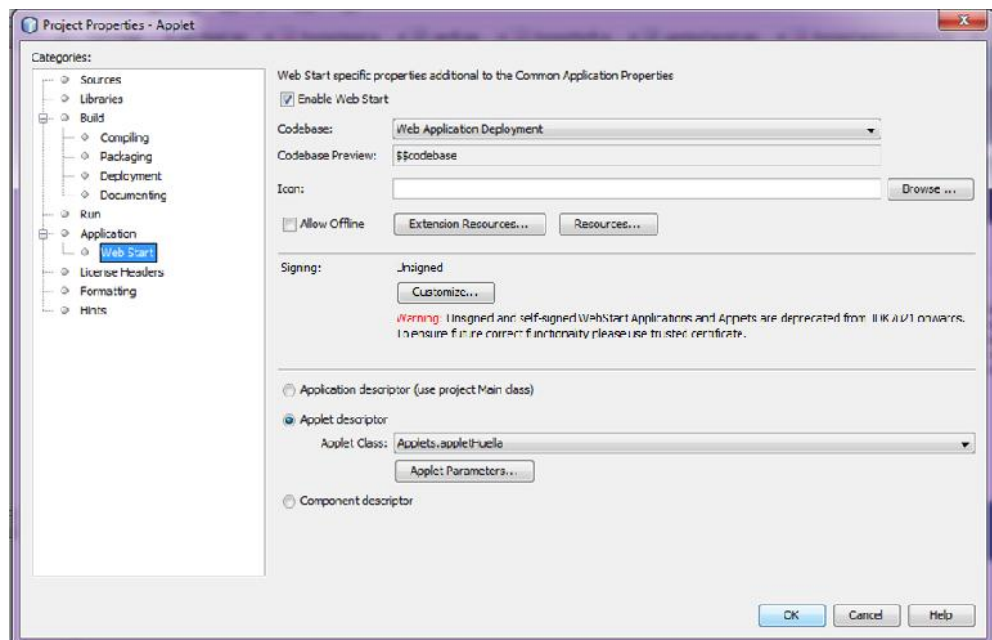
C:\Program Files\Java\jdk1.8.0_51\bin>
```

Ahora ya podemos firmar nuestro applet. Java posee una herramienta para realizarlo, su nombre es jarsigner, pero nosotros hemos optado por utilizar Netbeans, para que cada vez que el proyecto se construya, el programa se encargue de firmarlo.

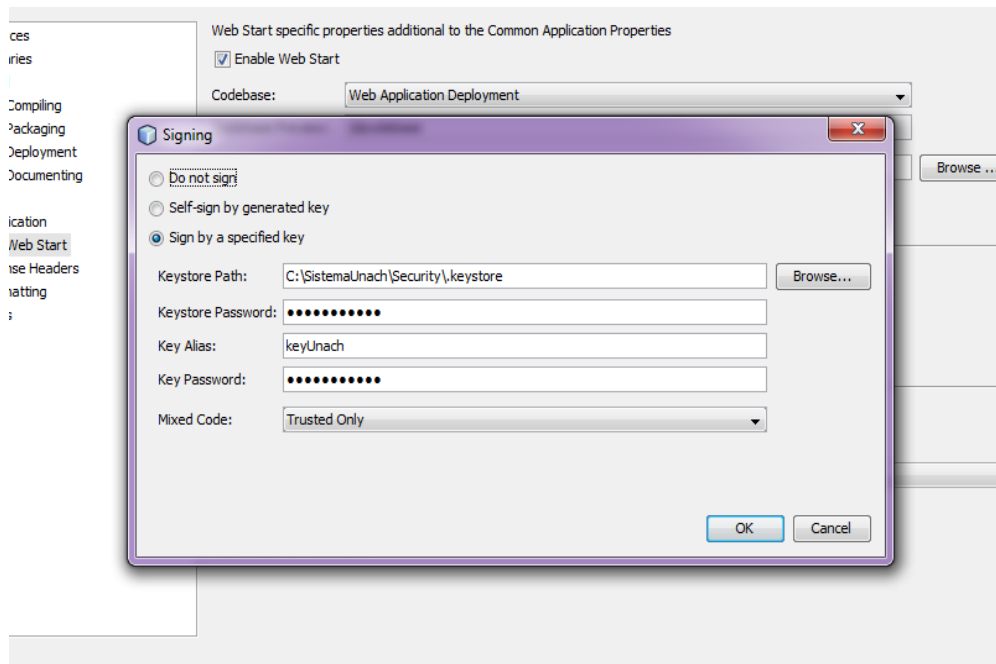
4. Ingresamos a netbeans y en el applet damos clic derecho y elegimos Properties.



5. Aparecerá un cuadro, en la lista de Categorías seleccionamos Application→Web Start



6. En la sección Signing seleccionamos Customize..., en el cuadro que aparece seleccionamos la tercera opción e ingresamos los datos que registramos en el keystore.



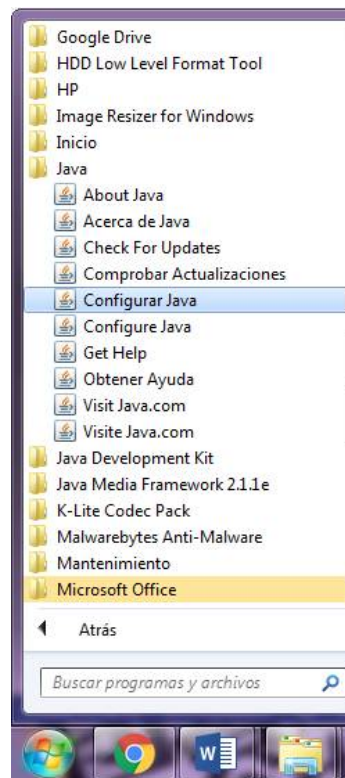
7. Aceptamos los cambios y el applet ya estará firmado.

Seguridad en Java

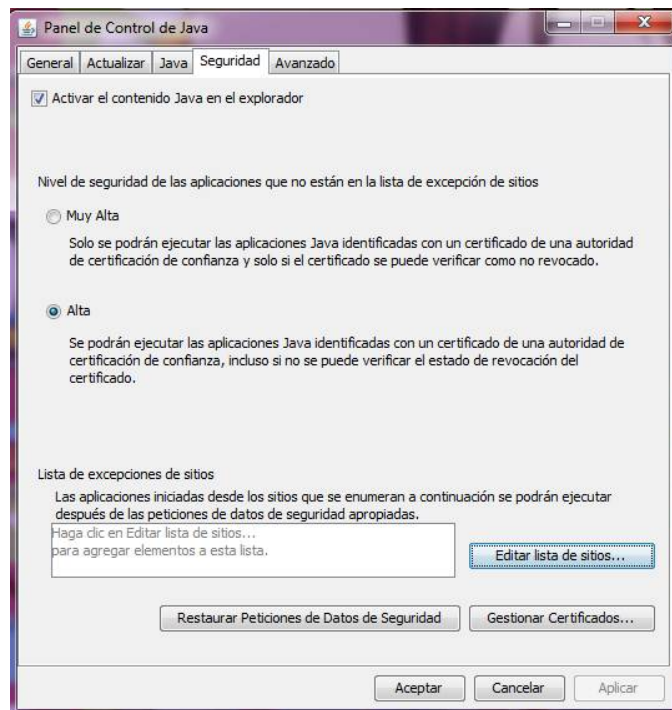
La dirección desde donde accedemos a nuestro applet debe ser ingresada en la lista de sitios seguros de Java para que nos permita el acceso.

Este procedimiento se realizará en cada computadora que va a trabajar con el applet.

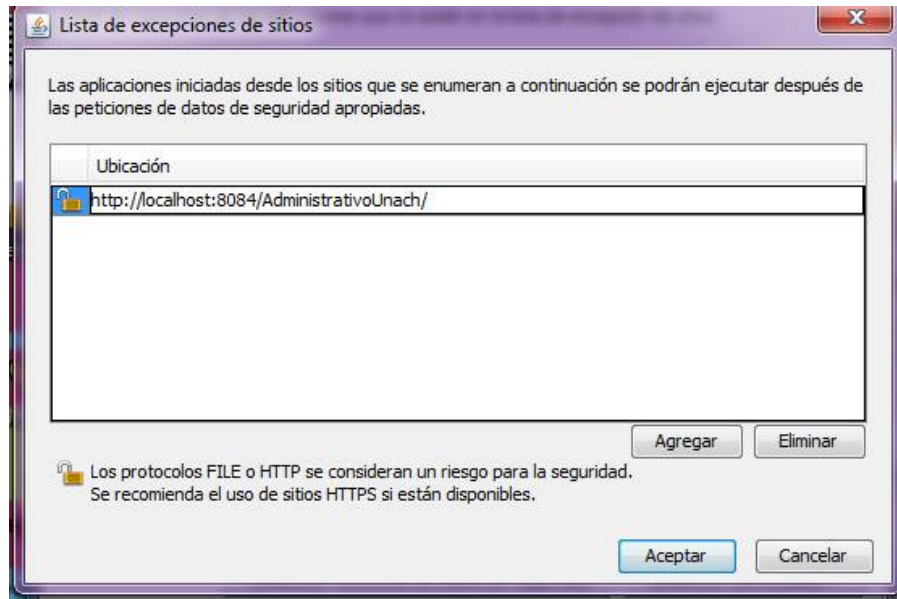
1. Seleccionamos Inicio/Todos los programas/Java/Configurar Java



2. En el cuadro que aparece nos dirigimos a la pestaña Seguridad



3. Seleccionamos nivel de seguridad Alta, damos clic en el botón “Editar lista de sitios” y agregamos la ruta del sistema.



4. Guardamos los cambios.

Permisos para acceder a la base de datos

El applet se conectará normalmente con la base de datos si se ejecuta como localhost (en el servidor), pero si accedemos desde otra computadora, el firewall del servidor bloqueará la conexión para solucionarlo, debemos configurar los archivos de postgresQL (la base de datos que estamos utilizando) y el firewall.

En PostgreSQL

1. Ubicamos el archivo *postgresql.conf* en la carpeta de instalación de PostgreSQL, en nuestro caso lo encontramos en la siguiente ruta: C:\Program Files (x86)\PostgreSQL\9.4\data\postgresql.conf, lo abrimos con un editor de texto y en la línea *listen_addresses* modificamos y escribimos “*” para que acepte todas las conexiones


```

C:\Program Files (x86)\PostgreSQL\9.4\data\postgresql.conf - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?

postgresql.conf
46 # (change requires restart)
47
48 # If external_pid_file is not explicitly set, no extra PID file is written.
49 #external_pid_file = '' # write an extra PID file
50 # (change requires restart)
51
52
53 # -----
54 # CONNECTIONS AND AUTHENTICATION
55 # -----
56
57 # - Connection Settings -
58
59 listen_addresses = '*' # what IP address(es) to listen on:
60 # comma-separated list of addresses;
61 # defaults to 'localhost'; use '*' for all
62 # (change requires restart)
63 port = 5432 # (change requires restart)
64 max_connections = 100 # (change requires restart)
65 # Note: Increasing max_connections costs ~800 bytes of shared memory per
66 # connection slot, plus lock space (see max_locks_per_transaction).
67 #superuser_reserved_connections = 3 # (change requires restart)
68 #unix_socket_directories = '' # comma-separated list of directories
69 # (change requires restart)
70 #unix_socket_group = '' # (change requires restart)
71 #unix_socket_permissions = 0777 # begin with 0 to use octal notation
72 # (change requires restart)
73 #bonjour = off # advertise server via Bonjour
74 # (change requires restart)
75 #bonjour_name = '' # defaults to the computer name
76 # (change requires restart)
77
78 # - Security and Authentication -
79
Normal text file length: 21908 lines: 616

```

2. En el archivo C:\Program Files (x86)\PostgreSQL\9.4\data\pg_hba.conf al final agregar la siguiente línea

host BD_Inscripcion postgres 192.168.1.0/24 trust

Bd_Inscripción es el nombre de la base de datos y con la dirección ip que hemos escrito estamos indicando que acepte todos los accesos de las computadoras conectadas a la red local.

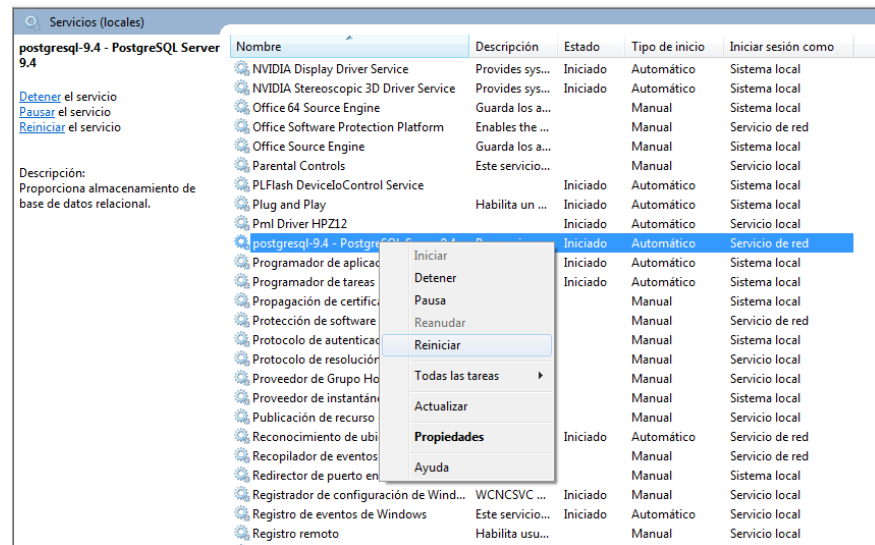
```

C:\Program Files (x86)\PostgreSQL\9.4\data\pg_hba.conf - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?

pg_hba.conf
56 # Database and user names containing spaces, commas, quotes and other
57 # special characters must be quoted. Quoting one of the keywords
58 # "all", "sameuser", "samerole" or "replication" makes the name lose
59 # its special character, and just match a database or username with
60 # that name.
61 #
62 # This file is read on server startup and when the postmaster receives
63 # a SIGHUP signal. If you edit the file on a running system, you have
64 # to SIGHUP the postmaster for the changes to take effect. You can
65 # use "pg_ctl reload" to do that.
66
67 # Put your actual configuration here
68 # -----
69 #
70 # If you want to allow non-local connections, you need to add more
71 # "host" records. In that case you will also need to make PostgreSQL
72 # listen on a non-local interface via the listen_addresses
73 # configuration parameter, or via the -i or -h command line switches.
74
75
76 # TYPE  DATABASE  USER  ADDRESS  METHOD
77
78 # IPv4 local connections:
79 host    all  all  127.0.0.1/32  md5
80 host    BD_Inscripcion postgres 192.168.1.0/24  trust
81 # IPv6 local connections:
82 host    all  all  ::1/128  md5
83 # Allow replication connections from localhost, by a user with the
84 # replication privilege.
85 #host    replication  postgres  127.0.0.1/32  md5
86 #host    replication  postgres  ::1/128  md5
87
88
Normal text file length: 418

```

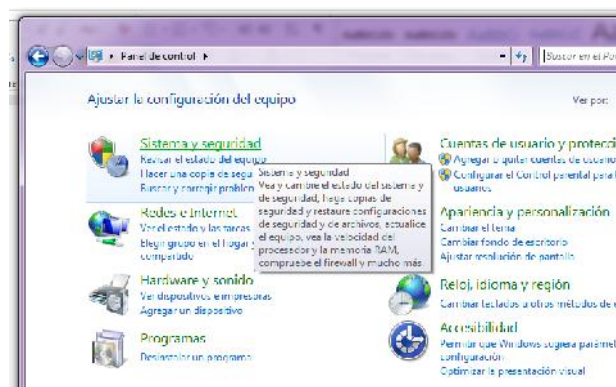
3. Reiniciar el servicio de postgresQL desde services.msc.

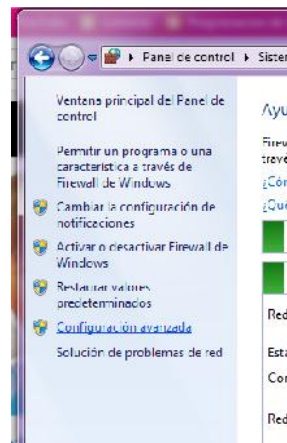


En el Firewall

Configuramos el firewall del servidor para que acepte conexión remota (del applet a la base de datos)

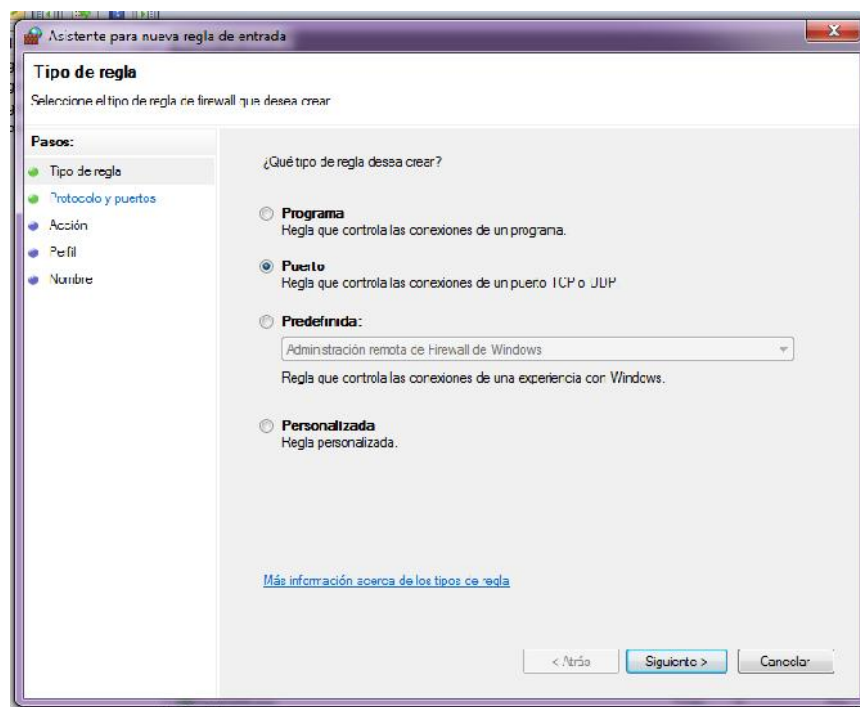
1. Ir al Firewall de Windows: Panel de Control → Sistema y Seguridad → Firewall de Windows → Configuración Avanzada



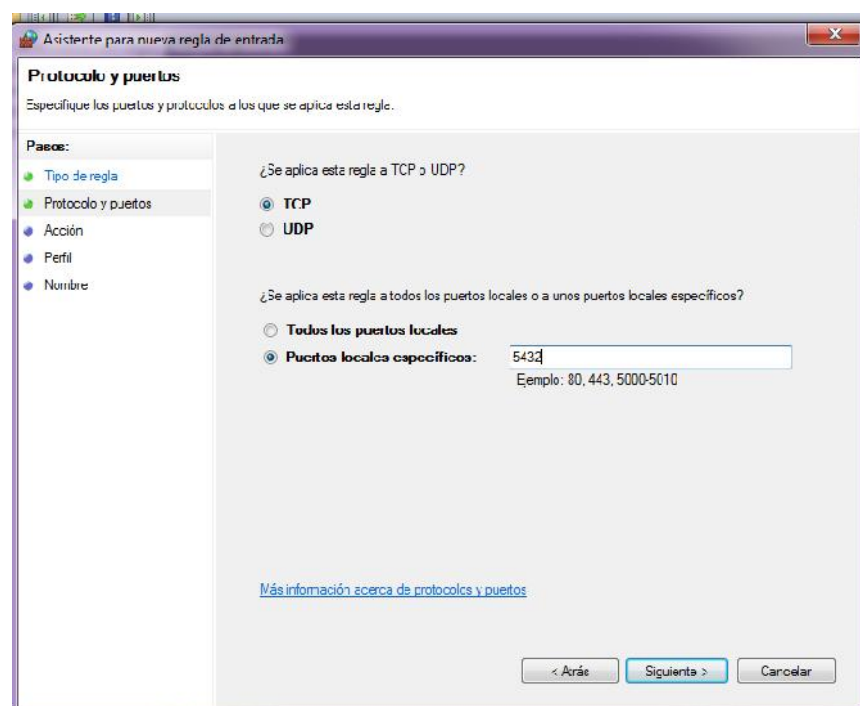


- [illegible]

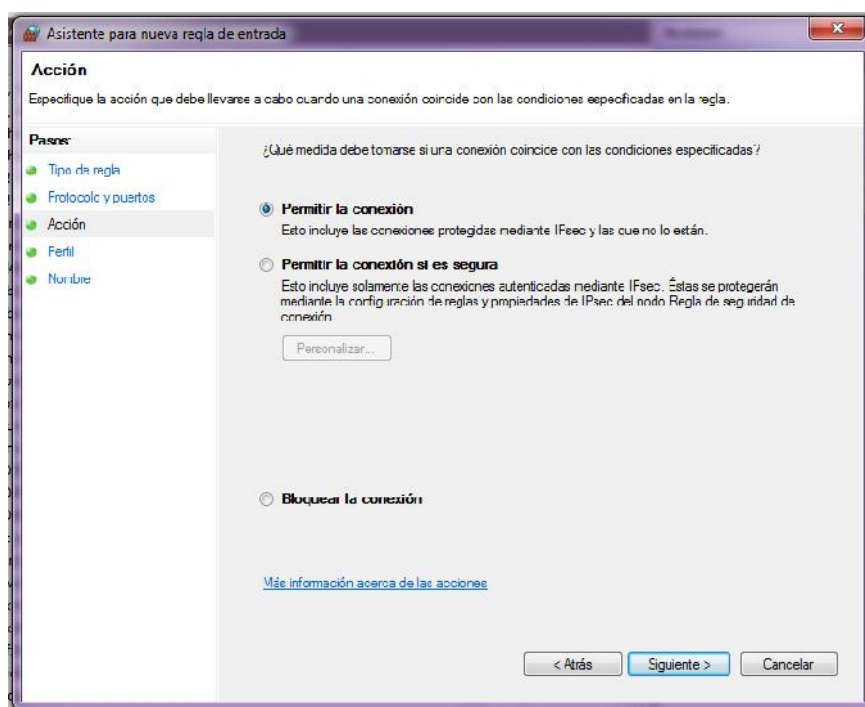
3. En el cuadro que aparece seleccionar Puerto y clic en el botón Siguiente.



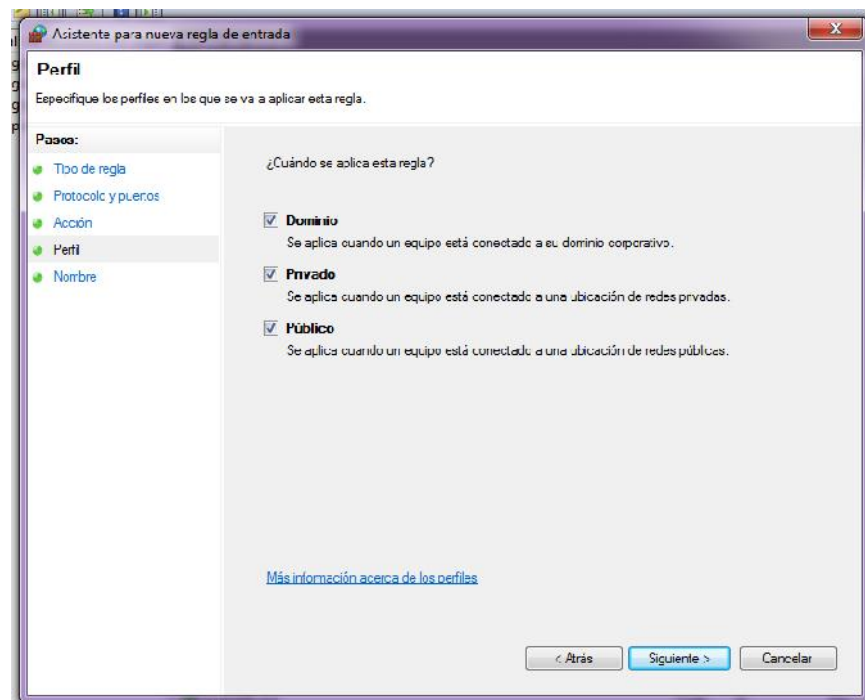
4. Luego seleccionar TCP y en Puertos Locales Específicos ingresar 5432 (puerto de postgres), clic en el botón Siguiente.



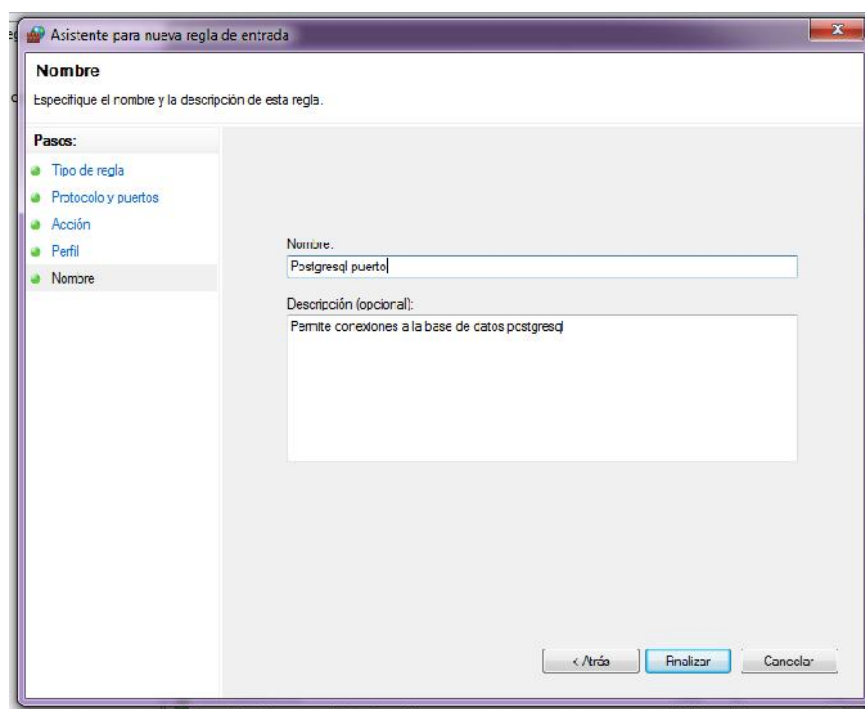
5. Seleccionar Permitir la conexión, clic en el botón Siguiente.



6. Seleccionar las tres opciones: Dominio, Privado y Público, clic en el botón Siguiente.



7. Ingresar un nombre y opcional una descripción y Finalizar



8. En el listado podemos observar que aparece la nueva regla

✓ Microsoft Office Outlook	Público	Si	Per	Expor
✓ minithunderplatform.exe	Público	Si	Per	Ayud
✓ MiniThunderPlatform.exe	Todo	Si	Per	Postgresq
✓ MultiMgr.exe	Todo	Si	Per	Desh
✓ mysqld	Público	Si	Per	Corta
✓ mysqld	Público	Si	Per	Copi
✓ NetBeans IDE	Público	Si	Per	Elimi
✓ NetBeans IDE	Público	Si	Per	Propi
✓ Postgresql puerto	Todo	Si	Per	Ayud
✓ Servicio Bonjour	Público	Si	Per	
✓ Servicio Bonjour	Público	Si	Per	
✓ Servicio Bonjour	Público	Si	Per	
✓ Servicio Bonjour	Público	Si	Per	