

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN
SISTEMA DE GESTIÓN DE DONACIONES Y
VOLUNTARIADO PARA LA FUNDACIÓN JÓVENES
CONTRA EL CÁNCER”.**

Previa la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

PAÚL ALEJANDRO JARRÍN PEREIRA

NELSON ALBERTO JUMBO HIDALGO

SANGOLQUÍ, 06 de Junio del 2013

CERTIFICADO

En nuestra condición de Director y Co-Director CERTIFICAMOS que hemos revisado y evaluado el desarrollo de la Monografía Técnica “**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE DONACIONES Y VOLUNTARIADO PARA LA FUNDACIÓN JÓVENES CONTRA EL CÁNCER**”, presentado por los señores PAÚL ALEJANDRO JARRÍN PEREIRA y NELSON ALBERTO JUMBO HIDALGO en el cual se han observado las disposiciones de la Institución, metodología y técnicas que regulan esta actividad académica.

Por ello autorizamos a los mencionados alumnos a reproducir el documento definitivo, presentarlo a la Dirección de la Tecnología en Computación y proceder a la exposición pública del contenido.

Sangolquí, _____

ING. LORENA DUQUE

ING. JENNY RUIZ

CERTIFICADO

En calidad de alumnos de la CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA, de la Escuela Politécnica del Ejército, CERTIFICAMOS que la presente Monografía Técnica, previa a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA, es un trabajo inédito el cual no puede ser copiado, ni reproducido parcial ni totalmente sin la autorización escrita del (los) autor (es).

Es todo cuanto podemos CERTIFICAR en honor a la verdad.

Sangolquí, _____

PAÚL ALEJANDRO JARRÍN PEREIRA

NELSON ALBERTO JUMBO HIDALGO

DECLARACIÓN

Nosotros, Paúl Alejandro Jarrín Pereira y Nelson Alberto Jumbo Hidalgo, declaramos que el presente trabajo es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación personal y que hemos consultado las referencias bibliográficas que se incluyen en el documento.

La Escuela Politécnica del Ejército puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normativa institucional vigente.

Sangolquí, 06 de Junio de 2013

Paúl Alejandro Jarrín Pereira

Nelson Alberto Jumbo Hidalgo

AUTORIZACIÓN

Nosotros, Paúl Alejandro Jarrín Pereira y Nelson Alberto Jumbo Hidalgo, autorizamos a la Escuela Politécnica del Ejército a que publique en el repositorio digital de la biblioteca Alejandro Segovia el presente proyecto de tesis, así como también los materiales y documentos relacionados a la misma.

Sangolquí, 06 de Junio de 2013

Paúl Alejandro Jarrín Pereira

Nelson Alberto Jumbo Hidalgo

DEDICATORIA

A mis padres y hermanos quienes han sido un peldaño fundamental en mi vida, que han forjado la persona que soy hoy en día y nunca me han abandonado a lo largo de esta etapa.

PAÚL ALEJANDRO JARRÍN PEREIRA

DEDICATORIA

A mi familia porque cada uno de sus miembros me ha apoyado para lograr este objetivo en mi vida y nunca han permitido que me deje vencer por los distintos obstáculos que se me han presentado, a mi novia porque ha sido mi soporte y parte fundamental para el cumplimiento de este objetivo.

NELSON ALBERTO JUMBO HIDALGO

AGRADECIMIENTO

A mis padres y hermanos por ser un gran ejemplo a seguir y haberme otorgado la oportunidad de estudiar, a mis compañeros de carrera y actuales amigos por ser parte de mi vida, a la Escuela Politécnica del Ejército por todas las enseñanzas impartidas, a las Ingenieras Lorena Duque y Jenny Ruiz por la ayuda brindada en la elaboración de este proyecto, a la Fundación Jóvenes Contra el Cáncer por el patrocinio.

PAÚL ALEJANDRO JARRÍN PEREIRA

AGRADECIMIENTO

A mis padres por el gran esfuerzo que realizaron para brindarme la oportunidad de estudiar, a mis hermanos por brindarme su consejo oportuno, a mi novia por tener siempre fe en mí y ayudarme a vencer las dificultades que se presentaron, a la Escuela Politécnica del Ejército por todas las enseñanzas impartidas, a las Ingenieras Lorena Duque y Jenny Ruiz por la ayuda brindada en la elaboración de este proyecto, a la Fundación Jóvenes Contra el Cáncer por el patrocinio.

NELSON ALBERTO JUMBO HIDALGO

TABLA DE CONTENIDO

CAPÍTULO 1- INTRODUCCIÓN	2
1.1 TEMA.....	2
1.2 ANTECEDENTES.....	2
1.2.1 Misión de la Fundación	3
1.2.2 Visión de la Fundación	3
1.2.3 Valores de la Fundación:.....	3
1.2.4 Programas.....	4
1.3 PLANTEAMIENTO DEL PROBLEMA	4
1.4 FORMULACIÓN DEL PROBLEMA	7
1.5 OBJETIVOS.....	7
1.5.1 Objetivo general	7
1.5.2 Objetivos específicos	7
1.6 JUSTIFICACIÓN.....	8
1.7 ALCANCE	9
1.7.1 Módulo de seguridad y acceso.....	10
1.7.2 Módulo de Gestión de voluntarios y donantes.....	10
1.7.3 Módulo Gestión de donaciones.....	10
1.7.4 Módulo para Gestión de guerreros.....	11
1.7.5 Módulo para la administración y aprobación de eventos.....	11
1.7.6 Módulo para la Asignación de donaciones.....	11
1.7.7 Módulo de Reportes Dinámicos	12
1.7.8 Módulo de Administración de catálogos.....	12
1.8 FACTIBILIDAD.....	13
1.8.1 Factibilidad Técnica.....	13
1.8.2 Factibilidad Económica	13
1.8.3 Factibilidad Operativa.....	14
CAPÍTULO 2- MARCO TEÓRICO	16
2.1 LA ORGANIZACIÓN: “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”.....	16
2.1.1 Organigrama Institucional	16
2.1.2 Historia	16
2.1.3 La Fundación.....	17
2.1.4 Misión.....	17
2.1.5 Visión	17
2.1.6 Valores	18
2.1.7 Imagen	18
2.1.8 Programas.....	20
2.2 METODOLOGÍA.....	22
2.2.1 Extreme Programming	22
2.2.1.1 Planificación.....	24
2.3 TECNOLOGÍAS.....	31
2.3.1 PARA EL DESARROLLO DEL SISTEMA	31
2.4 HERRAMIENTAS	53
2.4.1 Herramientas para la construcción de proyectos	53
2.4.2 Servidor de aplicaciones	62
2.4.3 Herramientas para la capa WEB.....	67
2.4.4 Herramientas de Base de Datos	69
CAPÍTULO 3- PLANIFICACIÓN DISEÑO	71
3.1 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE.....	71
3.1.1 Historial de Revisiones.....	71
3.1.2 Introducción.....	71
3.1.3 Requerimientos Específicos.....	76
3.2 MODELO DE CASOS DE USO	132
3.2.1 Historial de Revisiones.....	132
3.2.2 Introducción.....	132

3.2.3	Descripción General de Actores.....	132
3.2.4	Especificación de caso de uso: Login	133
3.2.5	Especificación de caso de uso: Caso de Uso Crear Voluntario.....	134
3.2.6	Especificación de caso de uso: Caso de Uso Modificar Voluntario	135
3.2.7	Especificación de caso de uso: Eliminar Voluntario	135
3.2.8	Especificación de caso de uso: Buscar Voluntario	136
3.2.9	Especificación de caso de uso: Aprobar Voluntario	137
3.2.10	Especificación de caso de uso: Crear Tipo Inventario.....	138
3.2.11	Especificación de caso de uso: Caso de Uso Modificar Tipo Inventario	139
3.2.12	Especificación de caso de uso: Eliminar Tipo Inventario.....	139
3.2.13	Especificación de caso de uso: Crear Inventario.....	140
3.2.14	Especificación de caso de uso: Caso de Uso Modificar Inventario	141
3.2.15	Especificación de caso de uso: Eliminar Inventario.....	142
3.2.16	Especificación de caso de uso: Crear Donante.....	142
3.2.17	Caso de Uso Modificar Donante.....	143
3.2.18	Especificación de caso de uso: Eliminar Donante	144
3.2.19	Especificación de caso de uso: Crear Donación	145
3.2.20	Especificación de caso de uso: Caso de Uso Modificar Donación	146
3.2.21	Especificación de caso de uso: Crear Rol	146
3.2.22	Especificación de caso de uso: Caso de Uso Modificar Rol.....	147
3.2.23	Especificación de caso de uso: Crear Usuario.....	148
3.2.24	Especificación de caso de uso: Caso de Uso Modificar Usuario.....	149
3.2.25	Especificación de caso de uso: Crear Tipo Catálogo	149
3.2.26	Especificación de caso de uso: Buscar Tipo Catálogo.....	150
3.2.27	Especificación de caso de uso: Crear Catálogo	151
3.2.28	Especificación de caso de uso: Caso de Uso Modificar Catálogo.....	152
3.2.29	Especificación de caso de uso: Eliminar Catálogo	153
3.2.30	Especificación de caso de uso: Buscar Catálogo.....	153
3.2.31	Especificación de caso de uso: Crear Ciudad.....	154
3.2.32	Caso de Uso Modificar Ciudad.....	155
3.2.33	Especificación de caso de uso: Eliminar Ciudad.....	156
3.2.34	Especificación de caso de uso: Buscar Ciudad.....	157
3.2.35	Especificación de caso de uso: Crear Provincia.....	157
3.2.36	Especificación de caso de uso: Modificar Provincia	158
3.2.37	Especificación de caso de uso: Buscar Provincia	159
3.2.38	Especificación de caso de Uso: Crear Guerrero	160
3.2.39	Especificación de caso de Uso: Modificar Guerrero.....	161
3.2.40	Especificación de caso de Uso: Buscar Guerrero	161
3.2.41	Especificación de caso de Uso: Eliminar Guerrero	162
3.2.42	Especificación de caso de uso: Crear Evento	163
3.2.43	Especificación de caso de uso: Manejar invitaciones a eventos	164
3.2.44	Especificación de caso de uso: Aprobar evento.....	164
3.2.45	Especificación de caso de uso: Enviar invitaciones a eventos	165
3.2.46	Especificación de caso de uso: Registrar resultados de eventos.....	166
3.2.47	Especificación de caso de uso: Buscar Evento	167
3.2.48	Especificación de caso de uso: Visualizar Evento.....	167
3.2.49	Especificación de caso de uso: Eliminar Evento	168
3.2.50	Especificación de caso de Uso: Registrar asignaciones de donaciones	169
3.2.51	Especificación de caso de uso: Modificar Asignaciones	170
3.2.52	Especificación de caso de uso: Generar Reporte Asignaciones por fecha	171
3.2.53	Especificación de caso de Uso: Generar Reporte Guerreros por Prioridad	172
3.2.54	Especificación de caso de uso: Generar Reporte de Actividades de un voluntario	173
3.2.55	Especificación de caso de uso: Generar Reporte de Auditoría	174
3.2.56	Especificación de caso de uso: Generar Reporte de Horas de voluntariado	174
3.2.57	Especificación de caso de uso: Generar Reporte de Donaciones por Fecha	175
3.2.58	Especificación de caso de uso: Generar Reporte Guerreros Clasificación	176
3.2.59	Especificación de caso de uso: Generar Reporte Inventario.....	177
3.3	DIAGRAMA DE CASOS DE USO.....	179

3.3.1	Módulo de seguridad y acceso.....	179
3.3.2	Módulo de Gestión de voluntarios y donantes.....	180
3.3.3	Módulo Gestión de donaciones.....	181
3.3.4	Módulo para Gestión DE guerreros.....	182
3.3.5	Módulo para la administración y aprobación de eventos.....	183
3.3.6	Módulo para la Asignación de donaciones.....	184
3.3.7	Módulo de Reportes Dinámicos.....	185
3.3.8	Módulo de Administración de catálogos.....	186
3.4	DIAGRAMA DE DESPLIEGUE.....	187
3.5	DIAGRAMA ENTIDAD RELACIÓN.....	188
3.6	DIAGRAMA DE CLASES DEL MODELO DE PERSISTENCIA.....	188
CAPÍTULO 4- PRUEBAS.....		189
4.1	PLAN DE PRUEBAS.....	189
4.1.1	Historial de Versiones.....	189
4.1.2	Carta de Aprobación.....	189
4.1.3	Introducción.....	189
4.1.4	Propósito.....	189
4.1.5	Alcance.....	189
4.1.6	Definición de términos y acrónimos.....	190
4.1.7	Visión General.....	190
4.1.8	Ítems a probar.....	190
4.1.9	Estrategia.....	191
4.1.10	Condiciones Especiales.....	192
4.1.11	Recursos.....	192
4.1.12	Responsables.....	193
4.1.13	Casos de Prueba.....	193
CAPÍTULO 5- CONCLUSIONES Y RECOMENDACIONES.....		215
5.1	CONCLUSIONES.....	215
5.2	RECOMENDACIONES.....	216
REFERENCIAS BIBLIOGRÁFICAS.....		217
REFERENCIAS WEB.....		218
BIOGRAFÍA.....		219
BIOGRAFÍA.....		220
HOJA DE LEGALIZACIÓN DE FIRMAS.....		221

LISTADO DE TABLAS

Tabla 1: Factibilidad Económica	13
Tabla 2: Anotaciones definidas en JSR-303	32
Tabla 3: Conceptos asociados a un Bean en CDI.....	35
Tabla 4: BeanScopes en CDI.....	40
Tabla 5: Detalle de la Estructura de Directorios en JBOSS AS 7.....	67
Tabla 6: Límites establecidos para POSTGRESQL	70
Tabla 7: Historial de Revisiones Especificación de Requerimientos	71
Tabla 8: Historial de revisiones: Modelo de Casos de Uso	132
Tabla 9: Historial de Versiones de Pruebas	189

LISTADO DE FIGURAS

Figura 1: Organigrama Institucional	16
Figura 2 Alex Dávila	16
Figura 3: LOGO Fundación Jóvenes Contra el Cáncer.....	18
Figura 4: Ciclo de planeamiento y retroalimentación en Extreme Programming.....	23
Figura 5: Desarrollo de un proyecto con Extreme Programming.....	24
Figura 6: Comunicación en Extreme Programming.....	28
Figura 7: Stand Up Meeting	29
Figura 8 Ciclo de vida JSF	51
Figura 9: Servicios proporcionados por Jboss AS7.....	63
Figura 10: Ejemplo de la topología Managed Domain.....	65
Figura 11: Distribución de Directorios en JBOSS AS7	66
Figura 12 Pantalla de ingreso al sistema	73
Figura 13: Plantilla para las pantallas del sistema.....	74
Figura 14: Diagrama de casos de uso del Módulo de seguridad y acceso	179
Figura 15: Diagrama de casos de uso del Módulo de Gestión de voluntarios y donantes	180
Figura 16: Diagrama de casos de uso del Módulo de Gestión de Donaciones	181
Figura 17: Diagrama de casos de uso del Módulo de Gestión de Guerreros.....	182
Figura 18: Diagrama de casos de uso del Módulo para la administración de eventos.....	183
Figura 19: Diagrama de casos de uso del Módulo para la Asignación de Donaciones.....	184
Figura 20: Diagrama de casos de uso del Módulo de Reportes Dinámicos	185
Figura 21 Diagrama de casos de uso del Módulo de Administración de Catálogos.....	186
Figura 22: Diagrama de Despliegue.....	187

LISTADO DE ANEXOS

ANEXO A	222
ANEXO B	223

GLOSARIO DE TÉRMINOS

- **Guerrero:** Persona enferma de cáncer
- **Ángel:** Persona que ha fallecido a causa del cáncer.
- **Wrapper:** clase que permite abstraer datos
- **Classpath:** parámetro que indica a la JVM, donde se encuentran las clases y paquetes definidos por el usuario.
- **Estado:** El estado de un objeto está dado por los valores que cada uno de sus atributos tiene en un determinado instante de tiempo.
- **JMS:** Java Message Service, Tecnología que permite el manejo de colas de mensajes.
- **Backing Bean:** clase Java que sirve para almacenar los valores que cada uno de los components de JSF posee.
- **Transacción:** Interacción con una estructura de datos compuesta por varios procesos.
- **Testeable:** Propiedad de un sistema de software que indica que una aplicación de software puede ser fácilmente probada para la detección de posibles errores.
- **Bug:** Es un error o fallo en un programa de computador o sistema de software que desencadena un resultado indeseado.
- **Repositorio:** Un repositorio, depósito o archivo es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.
- **CVS (Concurrent Version System):** Es una aplicación informática que permite llevar un registro y control de versiones de cada uno de los archivos de un proyecto de software en el cual trabajan varios desarrolladores.
- **SVN (Subversion):** Es el sucesor de CVS, pero este se orienta a proyectos en lugar de archivos, es decir, maneja versiones de un proyecto de software completo en lugar de hacerlo por cada archivo.

RESUMEN

El presente proyecto trata sobre el sistema de Gestión de Donaciones y Voluntariado para la “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”, implementado una arquitectura JEE6 sobre un servidor Jboss AS 7 aplicando la metodología Extreme Programming.

Se usó una metodología ágil ya que la fundación no tenía una idea completamente clara sobre las necesidades que debía satisfacer el sistema, y este tipo de metodologías permite que los cambios tengan un impacto leve sobre el desarrollo de la aplicación porque el cliente puede ver los avances del proyecto continuamente y verificar que sus requerimientos se estén cumpliendo.

Para el desarrollo del proyecto se hizo uso de JEE6, que es una arquitectura robusta que permite el desarrollo de aplicaciones empresariales seguras y óptimas orientadas a la WEB, para soportar esta tecnología se necesita un servidor de aplicaciones certificado en la misma, razón por la cual se escogió JBOSS AS 7.

Las necesidades de la fundación fueron cubiertas mediante el desarrollo de los siguientes módulos:

- Módulo de seguridad y acceso
- Módulo de Gestión de voluntarios y donantes
- Módulo Gestión de donaciones
- Módulo para Gestión de guerreros
- Módulo para la administración y aprobación de eventos
- Módulo para la Asignación de donaciones
- Módulo de Reportes Dinámicos
- Módulo de Administración de catálogos

CAPÍTULO 1

INTRODUCCIÓN

1.1 TEMA

Análisis, diseño e implementación de un sistema de Gestión de Donaciones y Voluntariado para la “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”, utilizando la arquitectura JEE6 sobre un servidor Jboss AS 7 aplicando la metodología Extreme Programming.

1.2 ANTECEDENTES

“FUNDACIÓN JÓVENES CONTRA EL CÁNCER”, es una organización sin fines de lucro, que lucha en beneficio de los adolescentes con cáncer.

Se suscribe bajo el Acuerdo Ministerial 0009 del Ministerio de Inclusión Económica y Social (MIES) el 12 de septiembre del año 2007¹, y se encuentra ubicada en la calle Yugoslavia N35-170 y Azuay (Quito-Ecuador).

Las instalaciones cuentan con una oficina con Internet, música, biblioteca sobre todos los tipos de cáncer, sus tratamientos, efectos..., una sala de juegos en la que pueden distraerse, una sala en la que los pacientes reciben charlas o talleres y un comedor en el que almuerzan a diario.

Cabe destacar que como un Principio de la Fundación, el Director no recibe remuneración por su gestión.

¹<https://www.facebook.com/jovenescontraelcancer/info>

1.2.1 MISIÓN DE LA FUNDACIÓN²

Apoyar el crecimiento y formación académica, profesional y emocional de los jóvenes con cáncer de todo el Ecuador.

1.2.2 VISIÓN DE LA FUNDACIÓN

Estar presente en todas las provincias beneficiando a más de 200 jóvenes con cáncer con óptimos tratamientos y excelente calidad de vida.

1.2.3 VALORES DE LA FUNDACIÓN:

- **Integridad:** Esfuerzo por un comportamiento coherente, responsable y leal a la Fundación Jóvenes Contra el Cáncer.
- **Respeto por la Persona:** Creer en su dignidad, creatividad y capacidad de participación activa en la solución de los problemas. Por esta razón promovemos espacios para la reflexión y evaluación de nuestras actividades, generando así cambios necesarios para ser cada vez más eficientes.
- **Transparencia:** Compromiso de manejar y usar responsablemente los medios de los que disponemos para el cumplimiento de nuestra misión
- **Profesionalismo:** Búsqueda de mejora continua de las capacidades de gestión de nuestro personal y voluntarios, para asegurar la consecución de objetivos. La Fundación trabaja orientada a conseguir resultados.

²<http://www.jovenescontraelcancer.org/about.html>

1.2.4 PROGRAMAS

La fundación cuenta con *Programas*, **los cuáles son financiados mediante las donaciones de personas o empresas, y se detallan a continuación:**

- **Ayuda Económica** - AYUDA ECONÓMICA PARA JÓVENES DE ESCASOS RECURSOS
- **Psicoterapia** – La fundación cuenta con un psicoterapeuta el cual da apoyo emocional tanto a los jóvenes con cáncer como a sus familias.
- **Terapias Complementarias** - Talleres de: Reiki, Yoga, Tai Chi, Desarrollo Personal. Las clases las imparte profesionales en estas áreas como voluntarios.
- **Gastos Mortuorios**
- **Alcanza tu Sueño** – Permite cumplir al menos 2 o 3 sueños de los jóvenes con cáncer en etapa terminal.
- **Entretenimiento** – Permite establecer un ambiente especial y de aceptación social mediante entradas al cine, a conciertos o viajes.
- **Oportunidades Laborales**

1.3 PLANTEAMIENTO DEL PROBLEMA

El principio fundamental de la fundación es el ayudar a las personas que padecen de cáncer, por tanto necesitan conocer todo sobre sus pacientes, la mejor manera de hacerlo es contar con un sistema de información que permita manejar dicha información.

El sistema deberá administrar los datos personales del guerrero, la situación económica de él y de su familia, el tipo de cáncer que tiene así como la etapa en que este se encuentra, también se deberá llevar un registro de los medicamentos

específicos que necesita, dicha información podrá ser almacenada de tal forma que el personal tenga acceso a ella de manera oportuna.

Para que una fundación pueda servir a la gente necesita de dinero, medicamentos, comida y personas que estén dispuestas a brindar su ayuda de manera desinteresada; lograr que se combinen de tal forma que la fundación logre cumplir sus objetivos y ayudar a sus guerreros es un proceso complejo que involucra no solo el manejo del voluntariado, sino también deberá permitir gestionar los recursos obtenidos de las donaciones de acuerdo a prioridades establecidas por la fundación.

La fundación maneja los siguientes tipos de donaciones:

- Celulares: permiten a los guerreros de la fundación estar comunicados con sus familiares en cualquier momento.
- Medicamentos
- Alimentos no perecibles
- Ropa
- Económica mediante:
 - Depósito bancario o en sus ánforas
 - Vendiendo los productos elaborados en la fundación
- Medicamentos

Dentro de la fundación el manejo de los medicamentos es algo crucial, se debe tener en cuenta datos específicos sobre cada uno de éstos, para luego poderlos asignar a cada uno de los guerreros de acuerdo a las historias clínicas de los mismos, sus recursos económicos y a la etapa de la enfermedad en la que se encuentran y de esta forma poder conocer a cada momento su disponibilidad para al momento de pedir donaciones tener en cuenta aquello que está por terminarse.

Teniendo esto en cuenta se necesita una herramienta que automatice la asignación de donaciones de acuerdo a los parámetros ya establecidos y por otro lado automatizar la gestión del inventario de los distintos tipos de donaciones, permitiendo aumentar la velocidad y efectividad con la que se realiza dicho proceso; permitirá conocer a cada instante la situación en que se encuentra la fundación basándose en la disponibilidad de las donaciones que maneja.

Las personas naturales, jurídicas, o anónimas que realizan donaciones a la fundación deben estar disponibles dentro del sistema para que el personal de la misma esté en la capacidad de conocer quiénes están ayudando a cumplir con su misión; si se trata de una persona, una institución o si se trata de una donación anónima, de tratarse de una persona natural se necesitará conocer sus nombres, apellidos, teléfonos y correo electrónico, de tratarse de una institución se necesitará además el nombre de la misma.

Para poder ayudar a las personas lo más importante es generar lazos de amistad; con este fin la fundación realiza eventos a los cuales asisten las personas a cargo de la fundación, guerreros y los voluntarios, al término de éstos eventos es necesario registrar los resultados que produjeron respecto a la fundación, a cada guerrero y voluntario.

La asistencia de los voluntarios depende de los horarios que cada uno dispone. La fundación necesita una herramienta que le permita gestionar la información del voluntariado incluyendo sus datos personales, formación académica, experiencia laboral y disponibilidad de tiempo, para lo cual la fundación establece horarios, los cuales el voluntario podrá escoger al momento de registrarse como tal en la fundación.

Los eventos necesitan ser aprobados, para esto es necesario consultar el inventario de donaciones y enviar notificaciones a los voluntarios registrados de acuerdo a su horario de disponibilidad, surge aquí la necesidad de contar con una herramienta que permita al personal enviar dichas notificaciones vía correo

electrónico; por el mismo medio se podrán entonces receptor las confirmaciones de asistencia al evento.

Se evidencia claramente la necesidad de contar con un sistema de información que permita al personal de la fundación manejar de forma más efectiva sus recursos, su personal y sus voluntarios para brindar a la comunidad una ayuda de calidad.

1.4 FORMULACIÓN DEL PROBLEMA

¿De qué manera el implementar un Sistema de Gestión de Donaciones y Voluntariado permitirá a la Fundación Jóvenes contra el Cáncer manejar eficiente y eficazmente sus recursos para brindar una ayuda de calidad a sus guerreros?

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Realizar el análisis, diseño e implementación de un sistema de Gestión de Donaciones y Voluntariado para la “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”, utilizando la arquitectura JEE6 sobre un servidor Jboss AS 7 aplicando la metodología Extreme Programming.

1.5.2 OBJETIVOS ESPECÍFICOS

- Analizar con el cliente cada uno de los requerimientos que necesita el sistema para luego clasificarlos según su importancia y prioridad de acuerdo a la metodología escogida para el desarrollo del proyecto.

- Diseñar con la ayuda de una herramienta CASE los diagramas útiles para el sistema y determinar estándares para un sistema testeable, entendible, explorable y explicable
- Implementar el sistema para la “FUNDACIÓN DE JÓVENES CONTRA EL CÁNCER” usando la metodología XP, las tecnologías de JEE6 descritas en el alcance del proyecto y un servidor de aplicaciones jboss as 7.1.
- Acoplar un módulo de auditoría al sistema de la “FUNDACIÓN DE JÓVENES CONTRA EL CÁNCER”, el cual permita al administrador conocer quien realizó las diferentes transacciones sobre el mismo.

1.6 JUSTIFICACIÓN

El desarrollo de un sistema para la gestión de donaciones y voluntarios, ayudará a la “FUNDACIÓN DE JÓVENES CONTRA EL CÁNCER” a administrar de una mejor manera quienes son los voluntarios de la fundación, los datos de cada uno de los voluntarios, cuales son los horarios en los cuales están disponibles, podrán llevar reportes acerca de los jóvenes que se encuentran en esta fundación y saber con cuales medicinas serán tratados cada uno, también pondrán llevar de una mejor manera toda clase de donaciones que reciba la fundación, la fundación podrá sistematizar los procesos de donaciones y registros de voluntarios.

La información será manejada de una manera correcta por una base de datos y por el sistema evitando así errores en la misma, los cuáles pueden causar pérdidas de dinero y tiempo a la fundación.

Los procesos antes mencionados serán totalmente legibles y entendibles para las personas que administren el sistema, en este caso será la persona a cargo delegada por la fundación jóvenes contra el cáncer.

Para el desarrollo del proyecto se usara la metodología XP, debido a que la fundación necesita tener entregables funcionales cada cierto tiempo, por otro lado al fundamentarse en pruebas y en la optimización de código ayuda a que el sistema tenga bases sólidas y un alto rendimiento.

Permitirá, por otro lado tener un claro panorama de los requerimientos que el cliente desea que sean implementados, además gracias a la comunicación infaltable con el cliente, se podrá cumplir con dichas especificaciones a tiempo y acorde a lo que la fundación desea.

Del lado de los desarrolladores permitirá una actualización eficiente del código, ya que cualquiera de los dos desarrolladores puede corregir el código fuente del otro para mejorarlo, y además el código a subirse al repositorio de código debe estar probado.

En cuanto a las tecnologías a usar, se utilizará una arquitectura JEE6, la cual es óptima para distribución, transaccionalidad y portabilidad de una aplicación, esto nos permitirá asegurar una aplicación de calidad y correctamente realizada. También la usaremos para implicar cierto grado de investigación es nuestro tema, ya que esta arquitectura no es muy común que sea usada en Ecuador, porque es relativamente nueva.

Como servidor de aplicaciones se utilizará JBOSS AS 7, el cual es capaz de soportar la arquitectura JEE6 y optimizar los recursos que la aplicación use.

1.7 ALCANCE

El presente proyecto permitirá el desarrollo del sistema para la gestión de donaciones y voluntarios para la fundación de jóvenes contra el cáncer.

De esta forma, al término del mismo, la fundación contará con una herramienta más para seguir llevando a cabo su labor social de una manera más

eficaz y efectiva, además al no tener ningún costo para la fundación, éste podrá ser destinado a seguir ayudando a sus guerreros.

Los siguientes módulos permitirán llevar de una manera más ordenada los procesos que lleva a cabo la fundación.

El sistema de la gestión de donaciones y voluntarios para la fundación de jóvenes contra el cáncer, dispondrá de los siguientes 9 módulos:

1.7.1 MÓDULO DE SEGURIDAD Y ACCESO

Este módulo permite la administración de usuarios, roles y menús, los usuarios tendrán acceso a diferentes menús dependiendo del rol que estos tengan, esto quiere decir que a un usuario se le puede limitar el acceso a uno u otro menú.

1.7.2 MÓDULO DE GESTIÓN DE VOLUNTARIOS Y DONANTES

En este módulo se administra a las personas voluntarias e interesadas en los proyectos de la fundación, en los mismos que se pedirá información acerca de los datos personales del voluntario, los horarios de disponibilidad del mismo, y una serie de preguntas para saber sus intereses, estas series de preguntas, serán administrables esto quiere decir que mientras más preguntas existan en la base de datos, se contendrá más información acerca de la persona voluntaria

1.7.3 MÓDULO GESTIÓN DE DONACIONES

En este módulo se administra las diferentes donaciones que se realice a la fundación de una manera detallada estos quiere decir la cantidad, tipo, marca, estado en el que se encuentra la donación, etc.

Se tiene que controlar que las personas que realizan las donaciones pueden permanecer en el anonimato o pueden revelar

ellos mismo quien es el que realizo la donación.

1.7.4 MÓDULO PARA GESTIÓN DE GUERREROS

Guerreros se les conoce a las personas que tienen cáncer y luchan contra el mismo, mediante este módulo se administra a las diferentes personas que se encuentran en esta fundación, llevando así un tipo de historia clínica del mismo, todas sus preferencias, sus datos personales, la medicación que el enfermo debe de tomar o se le debe de administrar.

El sistema puede clasificar a los guerreros por la etapa de enfermedad, situación económica, tipo de hospital, tipo de cáncer, por familia.

1.7.5 MÓDULO PARA LA ADMINISTRACIÓN Y APROBACIÓN DE EVENTOS

Este módulo permite la administración de eventos que la fundación desee realizar, entre los datos que se ingresa del evento tendrá que ingresarse un horario en el cual se realizara el mismo, mediante este horario el sistema notificara a los voluntarios antes registrados cuyo horario de disponibilidad este dentro del horario del evento “Que un evento se encuentra dentro de su horario de disponibilidad esta por realizarse”.

1.7.6 MÓDULO PARA LA ASIGNACIÓN DE DONACIONES

En este módulo se administra la asignación de las donaciones a los diferentes guerreros, familias, fundaciones aliadas, varios.

1.7.7 MÓDULO DE REPORTES DINÁMICOS

Mediante este módulo se puede obtener reportes de los diferentes módulos nombrados con anterioridad, los principales reportes serán:

- Actividades de un voluntario
- Donaciones por fecha
- Guerreros por clasificación
- Guerreros por prioridad
- Asignaciones por fecha
- Movimientos de Inventario
- Reporte del módulo de Auditoría

Es importante señalar que los reportes se van a realizar de manera dinámica (Esto quiere decir que ninguno de los reportes va a ser fijo, más el usuario tendrá un gran control sobre los campos a ser visualizados más no en los valores ya que estos dependerán de la base de datos), dando de esta forma una facilidad para adquirir los datos que desee al usuario.

1.7.8 MÓDULO DE ADMINISTRACIÓN DE CATÁLOGOS

Este módulo permite administrar los diferentes medicamentos, requerimientos, los tipos de clasificación de los guerreros, los tipos de donaciones que se puede realizar a la fundación.

Los módulos que permiten la gestión de donaciones y la asignación de los mismos permitirán a la fundación llevar un control más adecuado de su inventario,

y de la gestión que ha realizado con las donaciones recibidas, además al tener la información organizada, gracias al sistema, la toma de decisiones será más fácil, por lo tanto la asignación de recursos será más adecuada y se podrá ayudar a más personas.

1.8 FACTIBILIDAD

1.8.1 FACTIBILIDAD TÉCNICA

El desarrollador del presente sistema cuenta con los recursos necesarios tanto económicos, de hardware y de software para llevar a cabo y culminar el presente proyecto, además el conocimiento que posee en el área de desarrollo le es suficiente para llevar a cabo los requerimientos del proyecto.

1.8.2 FACTIBILIDAD ECONÓMICA

Tabla 1: Factibilidad Económica

Software	
Material	Valor en dólares
JDK 1.6	0
Jboss	0
Apache Maven	0
Eclipse Helios	0
PostgreSQL 9.0	0
Subtotal	\$ 0,00

Recursos Materiales

Material	Valor en dólares
Internet	240
Hojas de papel bond	10
Impresiones	200
Computador	0
Subtotal	\$ 450,00

Recursos Humanos

Desarrollador por 4 meses	Valor
Nelson Jumbo	\$2400 por 4 mes
Paúl Jarrín	\$2400 por 4 mes
Subtotal	\$4800.00

Valor Total

Recurso	Valor
Software	\$ 0,00
Recursos Materiales	\$ 450,00
Recursos Humanos	\$4800,00
TOTAL	\$ 5250,00

1.8.3 FACTIBILIDAD OPERATIVA

El tema es totalmente factible ya que cuenta con el auspicio de la "FUNDACIÓN DE JÓVENES CONTRA EL CÁNCER"

El presente proyecto, aparte de ser un aporte para la sociedad ayudando a la fundación de jóvenes contra el cáncer, será un proyecto el cual estará compuesto de una tecnología de punta, en cuanto a frameworks de aplicaciones web, servidores web y herramientas usadas.

En el proyecto se utilizara las siguientes herramientas y tecnologías:

- Para el servidor de aplicaciones necesitaremos un servidor robusto para soportar toda la transaccionalidad de la aplicación, por ello se usara jboss 7 el cual está en una versión estable.
- La base de datos ya que puede manejar una gran cantidad de datos, y sus transacciones pueden ser simultaneas, usaremos Postgres, la cual es una de las bases de datos robustas.
- En cuanto a la tecnología para el desarrollo de la aplicación se usara, JSF 2.0 en combinación con richFaces 4.0, son tecnologías nuevas y totalmente probadas y aprobadas. Para la compilación del proyecto se usara Maven que se trata de una potentísima herramienta para la gestión del ciclo de vida y dependencias de proyectos escritos en Java, permite gestionar todas las fases de un proyecto java.

CAPÍTULO 2 MARCO TEÓRICO

2.1 LA ORGANIZACIÓN: “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”

2.1.1 ORGANIGRAMA INSTITUCIONAL



Figura 1: Organigrama Institucional

2.1.2 HISTORIA

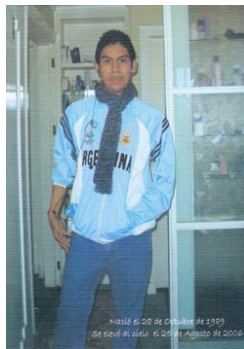


Figura 2 Alex Dávila

“Alex Dávila a sus 15 años tuvo cáncer a los huesos. Manos y corazones bondadosos del MoffittCÁNCER Center ayudaron a la fundación durante sus tratamientos.

El Gladiador Alex resistió la amputación de su hombro y la cirugía de sus dos pulmones.

Fue el artífice de crear este hermoso grupo de jóvenes guerreros con

diferentes tipos de cáncer. El 25 de Agosto 2006, Alex se marchó al cielo, pero su espíritu de lucha, amor y solidaridad se mantiene con nosotros.”³

2.1.3 LA FUNDACIÓN

“La Fundación Jóvenes Contra el Cáncer, es una organización sin fines de lucro, que lucha en beneficio de los adolescentes con cáncer.

La Fundación cuenta con una oficina con Internet, música, biblioteca sobre todos los tipos de cáncer, su tratamiento, efectos. En este lugar los chicos reciben charlas o talleres. Una pequeña sala de juegos, donde los chicos pueden compartir y distraerse. Además una cafetería. Diariamente los chicos almuerzan en la fundación.

Como un Principio de la Fundación, el Director no recibe remuneración por su gestión.”³

2.1.4 MISIÓN

“Apoyar el crecimiento y formación académica, profesional y emocional de los jóvenes con cáncer de todo el Ecuador.”³

2.1.5 VISIÓN

“Estar presente en todas las provincias beneficiando a más de 200 jóvenes con cáncer con óptimos tratamientos y excelente calidad de vida.”³

³ www.jovenescontraelcancer.org

2.1.6 VALORES

“Integridad: Esfuerzo por un comportamiento coherente, responsable y leal a la Fundación Jóvenes Contra el Cáncer.

Respeto por la Persona: Creer en su dignidad, creatividad y capacidad de participación activa en la solución de los problemas. Por esta razón se promueven espacios para la reflexión y evaluación de las actividades que se realizan, generando así cambios necesarios para ser cada vez más eficientes.

Transparencia: Compromiso de manejar y usar responsablemente los medios de los que disponemos para el cumplimiento de nuestra misión.

Profesionalismo: Búsqueda de mejora continua de las capacidades de gestión de del personal y voluntarios, para asegurar la consecución de objetivos. La Fundación trabaja orientada a conseguir resultados.”⁴

2.1.7 IMAGEN



Figura 3: LOGO Fundación Jóvenes Contra el Cáncer

“La imagen de la fundación utiliza 2 colores principales, el verde y el azul.

⁴ www.jovenescontraelcancer.org

- El VERDE, por la Esperanza, reflejada en cada acto. Cada día se renueva un chico y como las plantas surge nuevamente.
- El AZUL como el cielo indica que no hay límites. Elevamos nuestras miradas al cielo...A nuestro Padre, al Poder Espiritual.

El corazón está formado por 2 líneas: una gruesa que representa a los jóvenes guerreros hombres y la línea delgada por su delicadeza en su forma simboliza a las jóvenes guerreras mujeres.

En el centro del Corazón, dando forma a la J, un gran tronco de roble, simbolizando la fortaleza de cada guerrero.

Junto al tronco la letra O con una lagrima hacia arriba. Frecuentemente tendremos que secarnos nuestras lágrimas para continuar nuestro camino.

El muñeco se llama Alex en honor al creador de la Fundación, con una gran sonrisa, porque los chicos están alegres con una sonrisa a flor de pie.

Las quimioterapias causan entre otros efectos la caída del cabello, por esa razón Alex utiliza un gorrito.

Su ojo izquierdo bien abierto, porque se encuentra en estado de vigilia. Bajo su ojo un cubo de hielo que representan las lágrimas de rabia e impotencia pero también de esperanza. Su otro ojito esta semicerrado simbolizando el tiempo que duerme y descansa, no importa si tan solo son unos pocos minutos.”⁵

⁵ <http://www.jovenescontraelcancer.org/>

2.1.8 PROGRAMAS

- **Ayuda económica para jóvenes de escasos recursos**

“Un tratamiento de cáncer conlleva distintos gastos extra (dieta especial, transporte, medicamentos, exámenes médicos, hospitalizaciones, etc.) y no todas las familias los pueden asumir. Varios jóvenes viajan desde lugares lejanos en distintas provincias, al hospital donde reciben su tratamiento o se realizan exámenes y a la Fundación, otros tienen dificultad para movilizarse y deben viajar acompañados. Además por la situación de salud de sus hijos, algunas madres/padres deben dedicar el 100% de su tiempo al cuidado de sus hijos y deben abandonar sus trabajos.

Por esta razón, la Fundación gestiona ayudas económicas, para las familias más necesitadas de los jóvenes con cáncer. Este programa se ejecuta para evitar la deserción de sus tratamientos, facilitando dinero a los jóvenes y en algunos casos a su familiar acompañante para su transporte. Se da prioridad a las familias de jóvenes en etapa terminal y que se encuentran en tratamiento.”⁶

- **Psicoterapia**

“Tanto los jóvenes con cáncer como sus familias, enfrentan una etapa difícil y la Fundación puede apoyarlos emocionalmente con varias herramientas. Una de ellas es la psicoterapia. La Fundación Jóvenes contra el Cáncer cuenta con una Psicoterapeuta que atiende a los jóvenes en terapias individuales. Basados en la situación familiar y la condición de salud del joven, así como de la etapa que enfrenta, se fijan prioridades de atención. Actualmente muy pocos jóvenes son atendidos bajo este programa, debido a nuestras limitaciones económicas.

Nuestra meta es lograr apoyar a todos los jóvenes, con este programa e incluir a su núcleo familiar.”⁶

⁶ www.jovenescontraelcancer.org

- **Terapias complementarias**

“Las terapias complementarias, pueden apoyar a nuestros jóvenes y sus familias. Por esta razón, la Fundación cuenta con talleres de: Reiki, Yoga, Tai Chi, Desarrollo Personal y se seguirán sumando más alternativas.

El objetivo es mantener un equipo atendiendo estas áreas de manera permanente y con realizar los talleres con una frecuencia regular. “**Error! Marcador no definido.**

- **Gastos mortuorios**

“Que doloroso es para las familias enfrentar la pérdida de un ser querido. Adicional a este drama, se suman los gastos funerarios para las familias de escasos recursos. Nuestros jóvenes guerreros deben ser despedidos con todos los honores, luego de haber enfrentado con gallardía esta dura batalla. Garantizar un servicio funerario para los jóvenes con cáncer de escasos recursos es nuestro deber moral. Nuevamente esto solo es posible gracias a las personas que colaboran con sus donaciones de dinero.” **¡Error! Marcador no definido.**

- **Alcanza tu sueño**

“Consiste en cumplir al menos 2 de los 3 deseos de los jóvenes con cáncer en etapa terminal, como por ejemplo: un peluche, viaje a la playa, viaje en avión, ayuda familiar, visita de un artista, etc. En el año 2008 la FAE apoyo esta iniciativa llevando a un grupo de jóvenes a Salinas.

La Fundación quiere extender este proyecto a todos los chicos con cáncer del Ecuador en cualquier etapa de la enfermedad.”⁷

⁷ www.jovenescontraelcancer.org

- **Entretención**

“La adolescencia es una etapa de transición, de la niñez a la edad adulta y supone muchos cambios biológicos, psicológicos y sociales. El impacto psicológico puede ser muy intenso a esta edad. Los guerreros requieren un ambiente especial y actividades que los apoyen socialmente.

Tickets o pases de cortesía para cine, conciertos, recitales, obras de teatro, etc. son una buena idea. Un programa recreacional con viajes dentro y fuera de la ciudad, solo es posible gracias a los aportes de personas generosas.”⁷

- **Oportunidades laborales**

“En ciertas ocasiones los jóvenes de la fundación son discriminados, por sus tratamientos. No desean contratarlos por temor a que sufran una recaída o que se ausente de su puesto de trabajo. La fundación confía en sus guerreros y está consciente de que conseguir un trabajo, significa un apoyo emocional al joven, su familia y también una contribución a su economía.

Los guerreros poseen el carnet del CONADIS y tienen entre 13 y 22 años de edad y se pueden desenvolver en áreas administrativas, archivo, call center, digitación...”⁷

2.2 METODOLOGÍA

2.2.1 EXTREME PROGRAMMING

Para llevar a cabo el desarrollo del presente proyecto se hará uso de una metodología ágil llamada Extreme Programming debido a que los requerimientos del proyecto estarán en constante cambio ya que los

requerimientos del cliente siempre tienden a cambiar en el desarrollo, esto evitara que a la final de la entrega del proyecto el producto no satisfaga al cliente, por otra parte al ser una metodología iterativa permitirá ir entregando funcionalidades completas en cada iteración permitiendo al cliente contar con los procesos más críticos en primera instancia, es decir sin necesidad de esperar al final del proyecto ya podrá empezar a emplear el producto.

Extreme Programming también se basa en la programación en parejas para optimizar la productividad, ya que entre dos personas se pueden evitar errores de codificación en el instante en que se producen.

Sin embargo, la programación se realizara individualmente, debido a que los recursos a cargo del proyecto son únicamente 2 desarrolladores y el cliente.

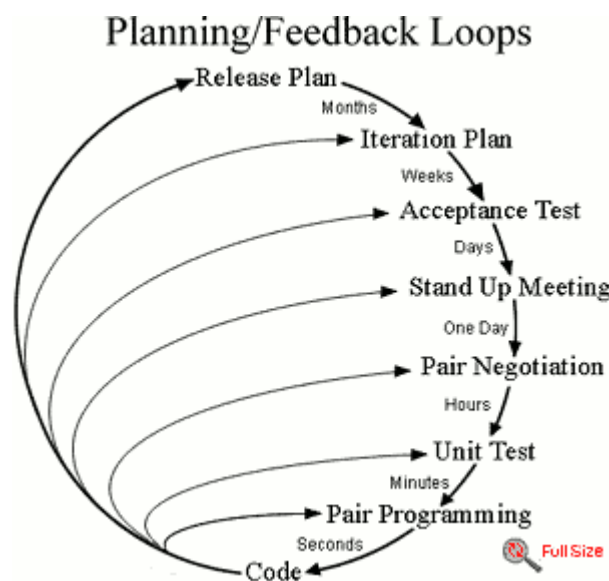


Figura 4: Ciclo de planeamiento y retroalimentación en Extreme Programming

2.2.1.1 PLANIFICACIÓN

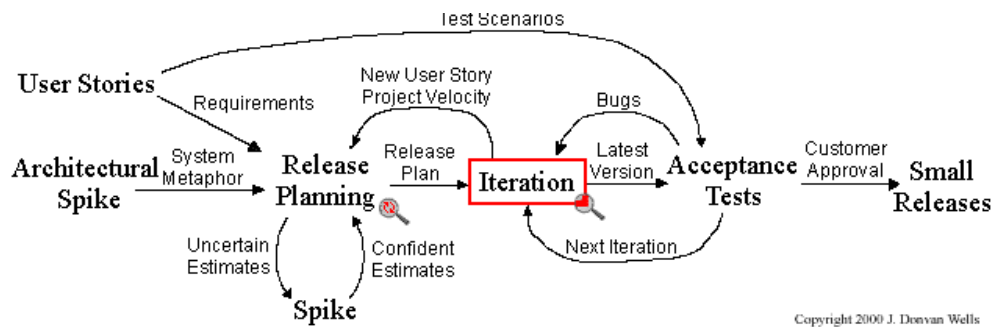


Figura 5: Desarrollo de un proyecto con Extreme Programming

En la etapa de planificación existen 5 tareas a cumplir que son:

- **Se escriben las “Historias de Usuario”:**

Las historias de usuario tienen el mismo propósito que los casos de uso, pero no son lo mismo, se utilizan para estimar el tiempo para la “Planificación de Reuniones”. También se utilizan en lugar de un extenso documento de requerimientos. Las historias de usuarios son escritas por los clientes como las cosas que el sistema necesita que hagan por ellos. Son similares a los escenarios de uso, salvo que no se limitan a la descripción de una interfaz de usuario. Historias de usuarios también impulsan a la creación de las pruebas de aceptación. Una o más pruebas de aceptación automatizadas se deben crear para verificar que las historias de usuario se han aplicado correctamente.

Uno de los mayores malentendidos con las historias de usuario es la forma en que difieren de las especificaciones de los requisitos tradicionales, la mayor diferencia es en el nivel de detalle. Las historias de usuarios sólo deben proporcionar detalles suficientes para hacer una estimación de en cuánto tiempo la historia se llevará a poner en práctica. Cuando llegue el momento de poner en práctica los desarrolladores de la

historia irán al cliente para recibir una descripción detallada de los requisitos cara a cara.

- **Creación del “Release Planning Meeting”:**

Luego de que las historias de usuarios han sido escritas se tiene que crear el “Release Plan” el mismo que es usado en las “ReleasePlanning Meeting”, el “Release Plan” especifica cuales historias de usuario se van a implementar por cada “Release” y las fechas de los mismos. Las historias de usuarios seleccionadas son transformadas en tareas individuales de programación e implementadas durante la iteración para completar las historias de usuarios, es importante señalar que una historia de usuario puede contener varias tareas de programación. Las historias de usuarios son transformadas en “Pruebas de aceptación” para ser probadas durante la iteración y también son verificadas en las siguiente iteración para finalizar la historia de usuario correctamente y continuar con el trabajo.

Retomando el tema de “ReleasePlanning Meeting”, su objetivo primordial es la estimación de tiempo por cada una de las historias de usuario por parte del equipo de trabajo y se indicara cuáles son las historias de usuario con prioridad estos es definido por el cliente.

- **Hacer frecuentes “Pequeñas Reuniones”:**El equipo de trabajo debe de ir proporcionando las historias de usuario desarrolladas al cliente para que el mismo pruebe, estas reuniones son usadas para indicar cuales son las historias de trabajo que están listas y pueden ser entregadas al cliente.
- **El proyecto es dividido en iteraciones:** Dividir el proyecto en iteraciones añade agilidad al proceso de desarrollo. Dependiendo de qué tan grande es el proyecto se realizara las iteraciones de 1 –

3 semana. Una semana es la mejor opción a pesar de que parece muy poco tiempo. Mantener la longitud de la iteración constante a lo largo del proyecto es importante, es esta constancia la que hace que la medición del progreso y la planificación sean sencilla y fiable en XP. No programe sus tareas de programación por adelantado, en lugar de esto tenemos una “Iteration Planning” al principio de cada iteración para planear lo que se hará durante ella.

- **“Iteration Planning” empieza con cada iteración:** En cada iteración se establece un “Iteration Planning” para asignar las tareas de programación, las historias de usuario que han fallado para el cliente son asignadas como tareas de programación en el siguiente “Iteration Planning”, las historias de usuario que no se han logrado concluir también son asignadas al siguiente “Iteration Planning. Al finalizar cada historia de usuario se entregará un historial de versiones de la misma.

2.2.1.2 Diseño

En la etapa de diseño existen 6 tareas a cumplir que son:

- **Simplicidad:** Un diseño simple siempre toma menos tiempo que uno complejo, si tu encuentras algo que es complejo reemplázalo por algo simple, siempre es más rápido y barato reemplazar código simple, el hecho de medir la simplicidad es muy subjetivo por eso se recomienda 4 cosas para el código, que sea “Probable”, “Entendible”, “Explorable” y “Explicable”.
Probable significa que tú puedes escribir pruebas de unidad para poder revisar los problemas.
Entendible significa que sea fácil mostrar a una persona nueva como funciona todo.
Explorable es la cualidad de ser capaz de encontrar lo que deseas cuando lo deseas, tener estándares en los nombres,

poner nombres a las clases y métodos que relacionen con lo que hacen.

Explicable significa saber para qué sirve cada parte de código que has escrito.

- **Escoger un sistema de metáfora:** Tiene dos características fundamentales, la primera es permitir que el sistema pueda ser explicado a los nuevos miembros que se unen al equipo, sin la necesidad de analizar cantidades inmensas de documentos y la segunda es tener un diseño que permita nombrar las clases y los métodos de una manera consistente, uniendo estas dos características se ahorrará tiempo ya que el código se explicará a sí mismo.
- **Usar “CRC cards” para el diseño de las sesiones:** Tarjetas de Clase, Responsabilidades y Colaboración (CRC) para diseñar el sistema como un equipo. El mayor valor de las tarjetas es permitir a las personas romper un procedimiento de pensamiento establecido y apreciar plenamente la tecnología de los objetos, cuantas más personas puedan ayudar a diseñar el sistema mayor es el número de buenas ideas incorporadas.
- **Crear “SpikeSolutions” para reducir el riesgo:** Cuando el equipo de trabajo se encuentra con un problema de riesgo el cual por decirlo tomaría 1 o 2 semanas a algún desarrollador resolverlo, se efectúa una “SpikeSolution” que es explorar las potenciales soluciones con todo el grupo de trabajo para saber cuál es la solución más factible y más fácil de

implementar, de esta manera los desarrolladores evitan perder tiempo desarrollándolo.

- **No añadir funcionalidad extra al sistema:** Es decir tiene que realizar lo que especifica la tarea, no debe dar funcionalidad que él crea será usada en el futuro, ya que con esto no se logrará cumplir las tareas asignadas para la iteración actual.
- **Refactorizar el código:** Ya que nuestro código va a ser reusado en la mayoría de partes posibles es muy importante refactorizar el código esto es significa crear getters y setters los atributos de las clases.

2.2.1.3 Gestión del proyecto



Figura 6: Comunicación en Extreme Programming⁸

La parte fundamental en la metodología XP es la comunicación, por esta razón se debe proveer al equipo de trabajo un espacio abierto, esto quiere decir eliminar las barreras físicas (paredes, modulares) entre los miembros del equipo.

⁸<http://www.extremeprogramming.org/map/development.html>

Una vez establecidos correctamente los canales de comunicación los miembros del equipo deben tener claro el fin de la iteración para avanzar acorde al tiempo que tienen para terminar sus tareas, ya que al final de cada una se debe presentar un producto lo más probado, completo, integrado y listo para producción posible. De esta manera se evita el esfuerzo futuro (no predecible) debido a un producto lleno de errores o incompleto.

Si algún miembro sabe que no terminará a tiempo lo que le ha sido encargado se deberá llevar a cabo una planeación de iteración para redimensionar el tiempo que tomara realizar cada una de estas tareas; es más importante acabar por completo una tarea, que tener varias incompletas, esto ayuda además a evitar la frustración del equipo debido a la sobrecarga de trabajo.

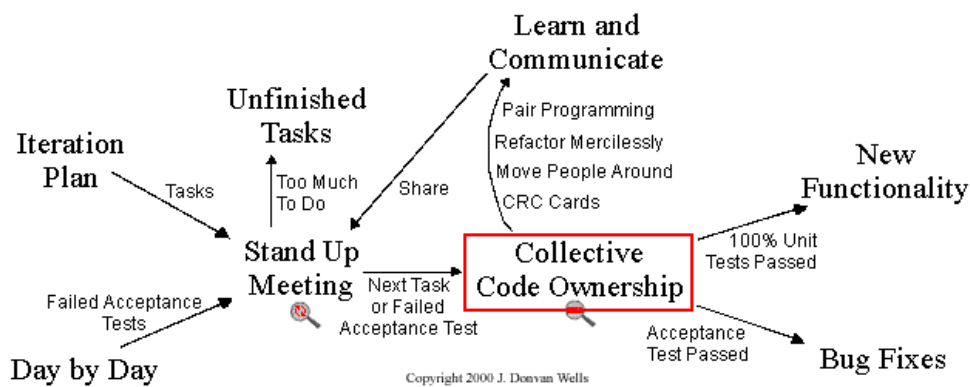


Figura 7: Stand Up Meeting⁹

Cada día se debe realizar el “*daily stand up meeting*”, su principal objetivo es establecer una comunicación entre todo el equipo, de esta manera se logran conocer los problemas que cada miembro ha tenido, las posibles soluciones y promover el trabajo en equipo.

Cada persona en esta reunión deberá exponer a las demás:

⁹<http://www.extremeprogramming.org/map/development.html>

- *Los problemas con los que se encontró ayer*
- *Las tareas a cumplir el día de hoy*
- *Que problemas se le podrían presentar en el transcurso del día y podrían quitarle tiempo*

Cada miembro debe conocer todo el proyecto; no se debe tener una persona en un área específica del proyecto ya que esto causa un cuello de botella debido a que es la única que la entenderá por completo y también es la única capaz de detectar errores, por el contrario todos deben conocer todo el proyecto, por esta razón los miembros del equipo pueden ir rotando en diferentes áreas, sin tener que por esto perder la productividad; es aquí donde entra la programación en parejas, ya que mientras una persona es asignada a un área diferente la otra puede seguir con la realización de las tareas habituales.

Algo importante a tener en cuenta es que ciertas veces la metodología puede conllevar problemas. Se deben seguir las reglas de XP, pero habrá veces en que puedan fallar y es el deber del equipo comunicar estas falencias para llevar adelante el proyecto y por otro lado mejorar la metodología como tal.

2.2.1.4 Codificación

En Extreme Programming el cliente se convierte en parte del equipo, permitiendo tener una comunicación clara con él para determinar si el proyecto avanza de acuerdo a lo que fue solicitado.

Es una pieza fundamental, él está presente durante todo el desarrollo del proyecto:

- Se encarga de formular las historias de usuario ayudado del desarrollador
- Posteriormente ayuda a la selección de las más importantes para el ReleasePlanning.
- Ayuda en la planeación de tiempos, para así asegurar que los requerimientos más importantes serán visibles primero.

- Complementa las historias de usuario para que las tareas de programación sean más claras.
- Y por último está presente en las pruebas de funcionalidad, para verificar que sus necesidades hayan sido satisfechas.

Gracias a la intervención del cliente el desarrollador puede avanzar con pasos firmes en sus tareas, pero debe tener en cuenta ciertos aspectos para que el producto resultante sea de calidad.

En primer lugar se deben establecer estándares, y éstos deben ser cumplidos por todos, esto ayuda a que el código sea comprensible, de esta forma el código puede ser compartido y entendido por cualquiera, ayudando al momento de refactorarlo.

Una buena práctica es realizar pruebas de unidad antes de empezar a codificar, de esta manera se puede tener un panorama más claro del desarrollo de la tarea completa. Se debe también ir incrementando la complejidad de las pruebas conforme se va complicando el código, así si se realizan cambios, se podrá verificar fácilmente los resultados de los mismos.

Por último el código debe ser integrado cada cierto tiempo, y de preferencia de una forma centralizada, es decir, que en un solo computador se tendrá el proyecto integrado, para poder realizar pruebas a este nivel, y poder mostrar al cliente todo el producto en funcionamiento.

2.3 TECNOLOGÍAS

2.3.1 PARA EL DESARROLLO DEL SISTEMA

- **JEE 6**

“Java Platform, Enterprise Edition, es el estándar de Java para el desarrollo de aplicaciones empresariales, se basa en la versión estándar de Java (Java SE), y añade librerías y servicios para el soporte de

escalabilidad, accesibilidad, seguridad, integridad y otros requerimientos de aplicaciones empresariales."¹⁰

- **BeanValidation (JSR -303)**

Es un framework estándar para la realización de validaciones en una aplicación, ayuda a que los desarrolladores realicen su lógica de validación en un solo lugar, reduciendo de esta forma la duplicación de código y la cantidad de errores.

Permite además crear nuevas anotaciones para poder validar cualquier tipo de objeto en tiempo real dentro de nuestra aplicación, esto implica que el servidor realizara un esfuerzo mínimo para validar un objeto, ya que antes de realizar cualquier acción o transacción contra una base de datos, se comprobará que estas validaciones sean correctas, evitando de esta manera la realización de tareas innecesarias.

Tabla 2: Anotaciones definidas en JSR-303

Anotación	Descripción	Ejemplo
@AssertFalse	El valor anotado del campo o propiedad debe ser "false".	@AssertFalse booleanisUnsupported;
@AssertTrue	El valor anotado del campo o propiedad debe ser "true".	@AssertTrue booleanisActive;
@DecimalMax	El valor del campo o propiedad debe ser un decimal menor o igual al definido en la anotación.	@DecimalMax("30.00") BigDecimaldiscount;
@DecimalMin	El valor del campo o	@DecimalMin("5.00")

¹⁰ <http://www.oracle.com/technetwork/articles/javaee/javaee6overview-141808.html>

	propiedad debe ser un decimal mayor o igual al definido en la anotación.	BigDecimal discount;
@Digits	El valor del campo o propiedad debe estar en el rango especificado. El número entero representa la cantidad de dígitos enteros que puede tener el campo, y el valor de "fraction" representa el máximo valor de cifras decimales.	@Digits(integer=6, fraction=2) BigDecimal price;
@Future	El valor del campo o propiedad debe ser una fecha en el futuro.	@Future Date eventDate;
@Max	El valor del campo o propiedad debe ser un entero menor o igual al definido en la anotación	@Max(10) int quantity;
@Min	El valor del campo o propiedad debe ser un entero mayor o igual al definido en la anotación.	@Min(5) int quantity;
@NotNull	El valor del campo o propiedad no debe ser nulo.	@NotNull String username;
@Null	El valor del campo o propiedad debe ser nulo.	@Null String unusedString;
@Past	El valor del campo o propiedad debe ser una fecha en el pasado.	@Past Date birthday;
@Pattern	El valor del campo o	@Pattern(regex="\\(\\d{3}\\)\\d{3}")

	propiedad debe corresponder con el regex expresado en la propiedad "regex".	-\\d{4}") StringphoneNumber;
@Size	El tamaño del campo o propiedad debe estar dentro de los límites especificados. Si es una cadena de texto se evalúa el largo de la cadena. Si es de tipo Collection, se evalúa su tamaño. Si es de tipo Map, se evalúa su tamaño. Si es de tipo Array, se evalúa su tamaño. Se pueden usar el elemento max o min por separado para definir límites.	@Size(min=2, max=240) String briefMessage;

Fuente: <http://docs.oracle.com/javaee/6/tutorial/doc/gircz.html>

- **Contexts and Dependency Injection (CDI)**

Esta característica de JEE6 está dada por la especificación JSR 299, para este proyecto se usará la implementación WELD, distribuida por JBOSS.

Define un conjunto de servicios, proporcionados por contenedores Java EE, que hacen que sea fácil para los desarrolladores utilizar EJB's, junto con la tecnología JavaServer Faces en aplicaciones web, su principal característica es que permite reemplazar los "managedbeans de JSF" por EJB's, permitiendo a la capa Web tener una mejor comunicación con la capa de negocio (EJB, JPA).

CDI también tiene muchos usos más amplios, permitiendo a los desarrolladores una gran cantidad de flexibilidad para integrar diferentes tipos de componentes de una manera precisa y con seguridades.

En este proyecto se usará WELD, que es la implementación de JBOSS para CDI.

- **Beans en CDI**

Cualquier componente que esté definido como tal en la especificación JEE es un bean, y su ciclo de vida es manejado por CDI acorde a su “scope”, por ejemplo tenemos: sesión-beans, message-driven-beans, servlets, etc.

Así no es necesario que una clase JAVA tenga una anotación para convertirse en un bean.

Tabla 3: Conceptos asociados a un Bean en CDI

BeanType	Incluye todas las superclases e interfaces en la jerarquía de tipos del bean.
Qualifier	Permite distinguir entre las múltiples implementaciones que un bean puede tener. Son definidos por el usuario.
Scope	Determina el ciclo de vida y visibilidad de cada una de las instancias de un bean. Los scopes que son parte de la especificación son: request, conversation, session y application y pueden ser extendidos; si un bean depende de otro heredará el scope del bean en el cual fue inyectado.
EL name	Es opcional pero puede ser usado para definir un nombre a ser usado desde components JSF.
Interceptors	Permiten extender el comportamiento de un bean.
Implementation	Todos los beans deben proveer una implementación, una clase JAVA es la encargada de hacerlo.

- **Injection**

Solo las instancias de un bean que son creadas mediante **@Inject** (declara un “injectionpoint”), son manejadas por **CDI**, aquellas creadas simplemente usando el operador **new** no son manejadas.

Hay que tener en cuenta que la inyección se lleva a cabo en el instante en que los campos o métodos son creados.

Con CDI, se pueden inyectar los siguientes tipos de objetos:

- SessionBeans.
- Java EE Resources: data sources, Java Message Service topics, queues, connection factories, etc.
- Persistence contexts (JPA EntityManager objects).
- Producer Fields.
- Objetos retornados por “producermethods”.
- Referencias a Web Services.
- Referencias a EJB remotos.

A continuación se detallan algunas formas de inyección:

Inyección a nivel de atributo

```
public class Estudiante {  
  
    @Inject  
    public Computer computer;  
  
}
```

Inyección a nivel de método

```
public class Estudiante {  
  
    @Inject  
    public setComputer (Computer computer){  
  
    }  
}
```

Inyección a múltiple a nivel de método

```
public class Estudiante {  
  
    @Inject  
    public init (Computer computer, Adress  
adress){  
  
    }  
}
```

Inyección a múltiple a nivel de constructor

```
public class Estudiante {  
  
    @Inject  
    public Estudiante (Computer computer, Adress  
adress){  
  
    }  
}
```

- **Producers**

CDI, usa el constructor de una clase para crear una nueva instancia de la misma; el desarrollador puede tener control de esto mediante “producermethods” los cuales deben estar anotados con @javax.enterprise.inject.Produces. Se usa principalmente para poder inyectar objetos que no son considerados beans.

Se debe tener en cuenta que los argumentos del “producermethod” son inyectados automáticamente por CDI.

Producer Method

```
public class EstudianteProducer {  
  
    @Produces  
    public Estudiante  
    obtenerNuevoEstudiante(Adress address){  
        return new Estudiante(address);  
    }  
}
```

Otra alternativa son los llamados “producerfields”, en este caso el valor del campo que está siendo producido es el valor a ser inyectado.

Producer Field

```
public class EstudianteProducer {  
  
    @Produces Adress address;  
  
    @Inject  
    public void init(Adress address){  
        estudiante = new Estudiante(address);  
    }  
}
```

- **Nombres para los beans**

CDI permite nombrar a los beans, principalmente para poderlos ver mediante EL(ExpressionLanguage) en la capa WEB en las páginas JSF.

A continuación un ejemplo:

@Named

```
@Named('estudiante')  
public class Estudiante {  
  
}  
  
@Named  
public class Estudiante {  
  
}
```

El valor entre comillas será el nombre del bean, en caso de no declararlo, CDI proveerá uno por defecto, el cual será igual al nombre de la clase pero con todas las letras de su nombre en minúsculas.

- **Visibilidad de los beans (BeanScopes)**

CDI maneja objetos contextuales, es decir objetos que tienen un ciclo de vida bien definido, mismo que está dado por su “scope”.

Cuando un objeto es requerido, CDI se encarga de buscar una instancia en el contexto apropiado para que pueda ser inyectada, en caso de no existir, una nueva instancia es creada y puesta en el contexto.

Es importante saber que si el scope es destruido, todos los beans dentro de éste dejarán también de existir.

La tabla a continuación muestra los distintos scopes definidos en CDI:

Tabla 4: BeanScopes en CDI

BeanScope	Descripción
@javax.enterprise.context. RequestScoped	Los beans anotados con <code>@RequestScoped</code> son compartidos en el contexto por el tiempo que dure una petición (request), esta puede ser una petición HTTP, una invocación a un EJB. Estos beans son destruidos una vez que la petición ha terminado.
@javax.enterprise.context. ConversationScoped	Los beans anotados con <code>@ConversationScoped</code> son compartidos a lo largo de múltiples peticiones HTTP, en la misma sesión pero solo si hay una conversación activa.
@javax.enterprise.context. SessionScoped	Los beans anotados con <code>@SessionScoped</code> son compartidos a lo largo de todas las peticiones HTTP llevadas a cabo dentro de una sesión. Los beans son destruidos una vez que la sesión ha finalizado.
@javax.enterprise.context. ApplicationScoped	Un bean anotado con <code>@ApplicationScoped</code> está con vida mientras la aplicación se encuentre corriendo, si ésta es apagada, los beans serán destruidos.
@javax.enterprise.context. Dependent	Los beans anotados con

@Dependent nunca son compartidos, siempre que se inyecta uno de éstos beans, una nueva instancia es creada y su ciclo de vida es manejado por el bean en el cual fue inyectado.

- **Java Persistence API (JPA) 2.0**

Es el estándar de JAVA para mapear objetos (entidades) a tablas de una base de datos relacional.

- **Entidad**

Es una clase que permite representar una tabla de una base de datos relacional, cada instancia se corresponde con una fila en la tabla.¹¹

Para ser considerada una entidad, la clase debe cumplir con las siguientes características:

- Debe estar anotada con `javax.persistence.Entity`
- Debe tener un constructor sin argumentos
- Ni la clase, ni ninguno de sus métodos puede estar declarado como `final`.

- **Mapeando una Entidad**

Una Entidad es declarada mediante la anotación `@ Entity`, y debe tener un atributo que sirva como identificador.

JPA asigna por automáticamente los nombres de tablas y columnas a las cuáles se hace referencia mediante el mapeo de nuestra Entidad, pero si se necesita establecer otros nombres se deben usar las anotaciones `@Table` y `@Column`.

¹¹<http://docs.oracle.com/javaee/6/tutorial/doc/bnbqa.html>

Entidad-Estudiante

```
@Entity
@Table(name="ESTUDIANTE")
public class Estudiante {
    @Id
    @Column(name="CODIGO_ESTUDIANTE")
    private Long codigoEstudiante;
}
```

- **Entity Manager**

Es el principal API en JPA, ya que es el encargado de manejar la creación-eliminación-actualización-consulta de entidades, crear queries para extraer registros de la base de datos y convertirlos a objetos.

- **Entity Manager manejado:** Este tipo de entity manager es manejado por un contenedor que soporte JPA SPI.¹² El contenedor que soporte esta interfaz puede ser un servidor de aplicaciones como Jboss.
- **Entity Manager no manejado:** Puede ser obtenido siempre y cuando la implementación de JPA se encuentre en el classpath del proyecto.

Obteniendo un Entity Manager Manejado

```
@Stateless
public class EstudianteEJB {
    @PersistenceContext(unitName="Colegio")
    EntityManagerem;
    ...
}
```

¹² JPA ServiceProvider Interface.

Obteniendo un Entity Manager no Manejado

```
...
    EntityManagerFactory emf = Persistence
        .createEntityManagerFactory("Colegio");
    EntityManager em =
        emf.createEntityManager();
    ...
    em.close();
```

- **Cómo usar el Entity Manager**

El entity manager tiene métodos que permiten manejar las operaciones CRUD, son los siguientes:

- `entityManager.persist`: persiste(inserta) un nuevo registro en la base de datos.
- `entityManager.merge`: actualiza un registro en la base de datos de acuerdo al estado de la entidad.
- `entityManager.find`: encuentra un registro en la base de datos.
- `entityManager.remove`: elimina un registro en la base de datos.

- **Estableciendo relaciones entre entidades**

Para representar las relaciones existentes en una base de datos relacional se usan las siguientes anotaciones: @OneToMany, @OneToOne, @ManyToOne, y su uso se describe a continuación:

Relación bidireccional uno a varios

Entidad-Estudiante

```
@Entity  
@Table(name="ESTUDIANTE")  
public class Estudiante {  
    @Id  
    @Column(name="CODIGO_ESTUDIANTE")  
    private Long codigoEstudiante;  
  
    @OneToMany(mappedBy="estudiante")  
    List<Asignatura>asignaturas;  
  
    //...getters and setters  
}
```

Entidad-Computador

```
@Entity  
@Table(name="ASIGNATURA")  
public class Asignatura {  
    @Id  
    @Column(name="CODIGO_ASIGNATURA")  
    private Long codigoAsignatura;  
  
    @ManyToOne  
    @JoinColumn("FK_CODIGO_ESTUDIANTE")  
    private Estudiante estudiante;  
    //...getters and setters  
}
```

Relación bidireccional uno a uno

Entidad-Estudiante

```
@Entity  
@Table(name="ESTUDIANTE")  
public class Estudiante {  
    @Id  
    @Column(name="CODIGO_ESTUDIANTE")  
    private Long codigoEstudiante;  
  
    @OneToOne(mappedBy="estudiante")  
    private Computador computadores;  
  
    //...getters and setters  
  
}
```

Entidad-Computador

```
@Entity  
@Table(name="COMPUTADOR")  
public class Computador {  
    @Id  
    @Column(name="CODIGO_COMPUTADOR")  
    private Long codigoComputador;  
  
    @OneToOne  
    @JoinColumn("FK_CODIGO_ESTUDIANTE")  
    private Estudiante estudiante;  
  
    //...getters and setters  
  
}
```

- **Enterprise Java Bean (EJB) 3.1**

Un EJB es un componente del lado del servidor que encapsula la lógica de negocio de una aplicación, por tanto dispone de las siguientes características: escalabilidad, transaccionalidad y seguridad.

Con JEE6 se han añadido las siguientes características:

- No es necesario crear una interfaz para el EJB.
- Permite crear EJB's de tipo Singleton, es decir, componentes que serán instanciados una sola vez.
- Se puede invocar métodos de forma asincrónica, usando una anotación.
- Ahora se pueden empaquetar EJB's dentro de un war.

- **Tipos de EJB en la especificación JEE6**

- **SessionBean**

Un SessionBean encapsula la lógica de negocio en servicios, los cuales podrán ser consumidos por varios clientes, ya sean estos locales o remotos.

- **StatelessSessionBean**

Este tipo de EJB, no mantienen una conversación con el cliente, es decir, la información del cliente solo es conservada mientras dure la invocación de algún método del EJB. Este tipo de componente sí puede ser usado para la implementación de WEB Services.

StatelessSessionBean

```
@javax.ejb.Stateless  
public class EstudianteEJB {  
    ...  
}
```

- **StatefulSessionBean**

Permite tener una conversación ente el cliente y el EJB, es decir, el estado¹³ es retenido hasta que la sesión termine, cada instancia de StatefulSessionBean puede estar asociada a un único cliente, y dicha asociación termina cuando el cliente termina la sesión. Hay que tener en cuenta que este componente no puede ser usado para la implementación de Web Services.

StatefulSessionBean

```
@javax.ejb.Stateful  
public class EstudianteEJB {  
    ...  
}
```

¹³ El estado de un objeto está dado por los valores que cada uno de sus atributos tiene en un determinado instante de tiempo.

- **SingletonSessionBeans**

Este tipo de EJB es instanciado una sola vez en toda la aplicación, es útil cuando se necesita ejecutar operaciones durante la inicialización o terminación de la aplicación o cuando se necesita compartir el estado de ciertos objetos en toda la aplicación.

SingletonSessionBean

```
@javax.ejb.Singleton
public class EstudianteEJB {
    ...
}
```

- **MessageDrivenBean**

Permite a las aplicaciones JEE procesar mensajes de forma asincrónica, estos mensajes son generados por cualquier aplicación de mensajes JMS¹⁴, al igual que los EJB de tipo Stateless, no guardan el estado de la conversación con el cliente, por último pueden procesar las peticiones de varios clientes.

MessageDrivenbean

```
@javax.ejb.MessageDriven
public class EstudianteEJB {
    ...
}
```

¹⁴JMS: Java Message Service

- **Beneficios del uso de EJB's**

El desarrollador puede preocuparse por la codificación de la lógica de negocio, dejando al contenedor EJB encargarse de la transaccionalidad y la seguridad, de esta forma se pueden consumir éstos servicios por varios clientes, como aplicaciones móviles, aplicaciones WEB y aplicaciones de escritorio.

Por último los componentes EJB son portables, y pueden ser usados en una nueva aplicación siempre y cuando el contenedor los soporte.

- JSF 2.0

JSF es un framework que facilita el desarrollo de interfaces de usuario para aplicaciones JEE. En su versión 2 permite la creación de páginas mediante un lenguaje conocido como **Facelets**, el cual permite el uso de HTML para la creación de dichas páginas, incorpora también el uso de **templates**, con lo cual se reduce la duplicación de código, ya que se pueden crear páginas que puedan ser reutilizadas por otras.

Tiene soporte nativo para AJAX, con lo cual se pueden realizar peticiones asincrónicas al servidor, y actualizar solo ciertas secciones de la página.

Por último ahora trae consigo "Compositecomponents", mediante esta característica se pueden crear componentes personalizados usando el lenguaje de marcado de JSF, así como validadores y convertidores, y se los puede almacenar en librerías para su reutilización.

- **ExpressionLanguage**

Es un lenguaje que permite a las páginas JSF interactuar con objetos JAVA.

- **Sintaxis**

Las expresiones EL son encerradas entre `#{...}`, y se accede a los atributos de un Bean con la sintaxis: `#{nombre_EL.propiedad}`.

La propiedad al cual se está haciendo referencia es un atributo privado del bean, que cuenta con métodos de acceso.

Expresión EL

```
#{estudiante.nombre}
```

Clase Java

```
@Named("estudiante")
public class Estudiante {
    private String nombre;

    public getNombre(){
        return nombre;
    }

    public setNombre(String nombre){
        this.nombre = nombre;
    }
}
```


- **Ciclo de vida de una página JSF**

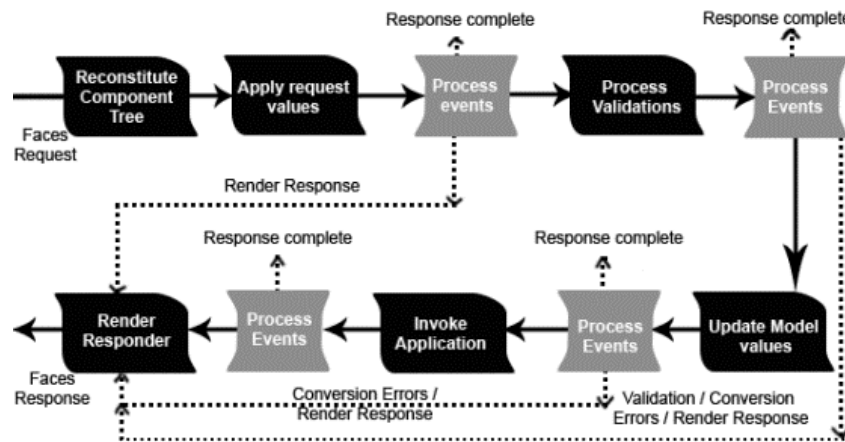


Figura 8 Ciclo de vida JSF¹⁵

- **Fase “Restore View”**

Empieza cuando un usuario realiza un request, ya sea dando click en un link, en un botón, etc.

El contenido de la página, es guardado en el objeto FacesContext, el cual contiene toda la información que JSF necesita para manejar los elementos de la interfaz gráfica para la petición actual y para la sesión actual.

El objeto FacesContext tiene un atributo conocido como “viewRoot” el cual almacena todos los componentes JSF de la página.

Luego el controlador FacesServlet envía una petición, y se obtiene un ID que identifica a la vista actual (view ID), y permite tener acceso a todos sus componentes, de no existir este código, el controlador de JSF crea una nueva vista.

Por tanto en esta fase el árbol de componentes es creado o restaurado.

¹⁵ <http://www.roseindia.net/jsf/jsflifecycle.shtml>

- **Fase “Applyrequestvalues”**

El objetivo de esta fase es obtener el estado actual de cada componente, este estado se encuentra en los parámetros enviados en el request.

Cada componente tiene un atributo conocido como “immediate”, si su valor se ha establecido en “true”, los procesos de validación, conversión y eventos asociados con este componente son procesados en esta fase.

De existir errores a nivel de validación o conversión estos son encolados en el objeto FacesContext.

Al final de esta fase cada componente tendrá asignados sus nuevos valores y los mensajes y eventos estarán en cola.

- **Fase “ProcessValidations”**

En esta fase los valores que cada componente tiene almacenados son comparados con las reglas de validaciones que se han establecido para cada uno, de no cumplirse, se añaden los respectivos mensajes a la cola del objeto Faces Context, entonces JSF se encarga de mostrar estos errores de validación al momento en la vista actual.

Si han existido errores de conversión en la fase “Applyrequestvalues”, también son mostrados.

Si no se han presentado errores se avanza a la fase “UpdateModelValues”.

- **Fase “UpdateModelValues”**

Luego de que la información ha sido verificada en las fases anteriores, los valores de los componentes pueden ser almacenados en los componentes del lado del servidor, por ejemplo en “backingbeans”¹⁶. De existir errores de conversión se procede a la fase “Render Response”, y se muestra los errores presentados.

- **Fase “InvokeApplication”**

En esta fase los valores en cada uno de los beans son los deseados, ya que se han realizado ya los procesos de validación y conversión respectivos, por lo tanto ahora pueden ser usados en los procesos de negocio de la aplicación.

La lógica de negocio es ejecutada al momento de hacer un submit de un formulario o al ir a otra página, una vez terminado esto se pasa a la fase final de JSF.

- **Fase “Render Response”**

En esta fase el contenedor JSF muestra de nuevo la página al usuario, esta página tendrá todos los componentes con sus nuevos valores.

2.4 HERRAMIENTAS

2.4.1 HERRAMIENTAS PARA LA CONSTRUCCIÓN DE PROYECTOS

- **Maven**

Es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl,

¹⁶Backing Bean: clase Java que sirve para almacenar los valores que cada uno de los components de JSF posee.

tiene un modelo de configuración de construcción más simple, basado en un formato XML. Maven utiliza un Project ObjectModel (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

La característica principal de Maven es que está listo para ser usado en red, el motor incluido en su núcleo puede descargar dinámicamente las librerías de un repositorio.

- **Conceptos de Maven**

- **Dependencias Transitivas (Transitive Dependencies)**

Esto es que las dependencias que declara una dependencia son utilizadas como dependencias propias del POM que se está utilizando. Ejemplo: si A declara dependencias sobre B y B declara dependencias sobre X e Y, cuando se ejecuten las fases del ciclo de vida de Maven se van a utilizar como dependencias B, X e Y.

- **POM (Project Object Model)**

Es un archivo xml llamado pom.xml y ubicado en la raíz de un proyecto (o un módulo) que declara datos sobre el mismo, dependencias y plugins a utilizar durante el ciclo de vida de la administración del proyecto.

- **Herencia y POM Real (Effective POM)**

Los POMs pueden heredar de otro POM que se defina como parent. La raíz implícita desde donde heredan todos los POMs es el Super POM definido por Maven. El POM resultante luego de incorporar todos los ancestros se

llama Effective POM. Se puede inspeccionar corriendo el goal effective-pom del plugin help parados sobre el directorio donde se encuentra el pom en cuestión.

```
mvn help:effective-pom
```

- **Goal**

Los Goals de Maven son las unidades mínimas de ejecución de las que disponemos durante su uso. Los goals pueden programarse en diferentes lenguajes pero en la práctica están la mayoría programados en Java. No es común que como usuario se piense en programar un goal. Un grupo de goals conforman un plugin. La ejecución de un goal se dispara desde línea de comandos invocando Maven con el nombre del plugin que lo contiene:

```
mvn plugin:goal
```

Otra forma de invocar un goal es indirectamente declarando por ejemplo el uso de un plugin en un POM. Maven lo invocará dependiendo de la fase del ciclo de vida cuando por ejemplo se ejecute

```
mvn compile
```

- **Dependencia (Dependency)**

Maven maneja todas las necesidades de conseguir artefactos (libs, jars, ears, zips, etc.) mediante las dependencias ubicando cada dependencia mediante las coordenadas.

- **Grupo (Group)**

El grupo es simplemente una forma de agrupar artefactos bajo una denominación común. La identificación del grupo es groupId. El nombre suele ser el nombre de la organización, el proyecto o el equipo de trabajo o bien la raíz del package de java que contendrá las clases del

artefacto, ej: com.xyz.foo. Lo importante es recordar que el groupId forma parte de las coordenadas de Maven con la que se identifican en forma única los artefactos en las dependencias y los repositorios.

- **Artefacto (Artifact)**

Maven utiliza el término artifact para denominar a la unidad mínima que maneja en su repositorio. Puede ser por ejemplo un jar, un ear, un zip, etc. Cuando Maven compila y empaqueta código, produce también artifacts que instala en el repositorio. Los artifacts están agrupados bajo el id de grupo (groupId) y tienen un id propio (artifactId), una versión, un clasificador y una extensión. Para administrar los artefactos en el repositorio Maven los acompaña con un respectivo pom.xml conteniendo los datos anteriores.

- **Repositorio (Repository)**

El funcionamiento de Maven se basa en el uso de un repositorio a donde ir a buscar las dependencias. La identificación de los artefactos que necesita se hace a través del sistema de coordenadas groupId:artifactId:version que asegura que cada dependencia sea única y proveer una ruta para conseguirla.

El repositorio por default es el repositorio central de Maven. A este repositorio se suman otros repositorios, un repositorio local y la posibilidad de utilizar un repositorio/proxy.

Cuando Maven sale a buscar y consigue una dependencia la guarda en el repositorio local que es un directorio en la máquina del usuario (~/.m2/repository). Las siguientes veces que necesite esta dependencia irá

la conseguirá en el directorio local haciéndolo mucho más rápido que la primera vez.

El directorio central y los directorios de terceros en lo único que se diferencian es en que se va a encontrar en cada uno. En el repositorio central de Maven podemos esperar encontrar la mayoría de las dependencias que necesitemos como JUnit, Maven mismo, Java, Spring, Hibernate, etc.

El repositorio/proxy es un servicio que podemos instalar en la LAN para cumplir una función similar a la del repositorio local pero compartido por un grupo de usuarios. Esto, además de mejorar la performance, permite tener un mayor control de las versiones de dependencias que se utilizarán.

- **Coordenadas (Coordinates)**

Las coordenadas de Maven es la forma de identificar a un proyecto, dependencia, plugin o artefacto en forma única en un pom y en un repositorio. Una coordenada se compone de los siguientes identificadores: groupId:artifactId:version.

A veces es necesario generar dos empaquetados distintos de una misma versión por cuestiones técnicas, por ejemplo un mismo artefacto para correr sobre dos entornos diferentes. En esta situación se puede utilizar un identificador más: groupId:artifactId:version:classifier.

Al momento de producir artefactos, estos cuatro identificadores funcionan en conjunto con uno más, pero que no forma parte de la coordenada: packaging y que se refiere al tipo de empaquetado del artefacto (ejs: zip, jar, ear). De hecho no es posible tener en el repositorio dos artefactos de la misma versión pero con diferente empaquetado.

groupId nombre bajo el que se agrupan una serie de artefactos (ej: junit, spring, org.xyx, com.tm.project(x)').

artifactId identificación de la unidad mínima administrada por el repositorio (ej: junit, moduloEjb, ejb-api, servlet-api).

version la versión del artefacto con la siguiente nomenclatura <major version>.<minor version>.<incremental version>-<qualifier>.

classifier clasificador que permite diferenciar iguales versiones pero preparadas o compiladas para diferentes entornos.

packaging no forma parte de la coordenada y sólo se utiliza al momento de generar un artefacto. Define el formato del empaquetado (ejs: jar, war, zip, pom).

- **MOJO (Maven Plain Old Java Object)**

MOJO es el término que utiliza Maven para referirse a un artefacto que representa un Goal. El término parte de un juego de palabras entre POJO (Plain Old Java Object, clases simples sin imposiciones de un framework) y Mojo (un hechizo mágico que se hace típicamente envolviendo ciertos elementos en un paño rojo). Maven hacen al término MOJO extensivo también a goals programados en otros lenguajes.

Para programar un Goal utilizando Java como lenguaje, basta con escribir una clase que haga lo que se necesita hacer con la única imposición de implementar una interfase. El resto de la configuración va en un descriptor en xml. Luego solo resta empaquetarlo con la ayuda del propio Maven y publicarlo en algún repositorio.

- **Plugin**

Un Plugin es un conjunto de Mojos (Goals) junto con una descripción y configuración en un xml. Los plugins pueden instalarse y distribuirse mediante los repositorios. Los goals de los plugins pueden utilizarse desde línea de comando: `mvn plugin:goal` o declarándolo en el pom en la sección `<plugins/>` con el tag `<plugin/>`.

- **Ciclo de Vida (Lifecycle)**

Maven utiliza el concepto de ciclos de vida para ejecutar una secuencia ordenada de pasos denominados fases. El ciclo de vida más común y el usado por defecto por Maven es el de build.

Este ciclo de vida define la secuencia de fases que va desde validar la integridad hasta el despliegue en su instalación final de un proyecto. Cuando se solicita la ejecución de una fase Maven ejecuta primero todas las fases anteriores siguiendo la secuencia y termina en la fase solicitada.

Cuando la ejecución llega a una determinada fase, Maven busca los plugins que estén adjuntados a la misma y ejecuta los goals correspondientes. Pueden existir ninguno o más plugins asociados a una fase.

Las fases en sí sólo definen un concepto en forma vaga y no ejecutan código sino que son anclajes para adjuntar los goals de los plugins para que se ejecuten en el momento apropiado.

Para pedirle a Maven que ejecute los goals asociados hasta un determinado punto en el ciclo de vida se ejecuta `mvn phase` donde `phase` es alguna de las fases del ciclo de vida en cuestión, por ejemplo `mvn`

package ejecutará la secuencia de fases desde testing, pasando por compilación y terminando en el empaquetado de los artefactos. En esta secuencia pasa por una veintena de fases.

- **Arquetipo (Archetype)**

La descripción y administración de un proyecto con Maven requiere que el proyecto tenga una estructura determinada y que tenga los pom.xml correspondientes. Existe un plugin en Maven que permite generar la estructura y configuración inicial de un proyecto a partir de una plantilla. Estas plantillas con llamadas archetypes (arquetipos) y el plugin que permite la generación a partir de los arquetipos se llama también archetype.

Un arquetipo consta de un directorio resources que contiene todos los archivos y directorios que contendrá el proyecto generado y un descriptor en xml.

Los arquetipos pueden publicarse en un repositorio al igual que cualquier otro artefacto.

- **Perfiles (Profiles)**

Maven está diseñado para maximizar la portabilidad de los artefactos. Es posible redefinir prácticamente cualquier parte de un POM definiendo un perfil. Pueden definirse varios perfiles. La activación de un perfil puede dispararse por uno o una combinación de las siguientes evaluaciones:

- Existencia o inexistencia de un archivo
- Sistema operativo
- Variables de entorno del sistema operativo

- Inexistencia de variables de entorno del sistema operativo
- Propiedades (Properties)

Las propiedades son valores que pueden ser accedidos en cualquier lugar del pom donde se espera un valor envolviendo la propiedad en `${}`, ej: `${project.groupId}`.

Existen algunas propiedades preestablecidas:

- `${project.*}` para acceder a cualquier valor definido en el POM
- `${parent.*}` para acceder a los valores de un POM ancestro
- `${settings.*}` para acceder a valores definidos en `~/.m2/settings.xml`
- `${env.*}` variables de entorno
- `${x.y.z}` para acceder a valores que desde Java se acceden con `System.getProperty("x.y.z")`
- `${x.y.z}` para acceder a propiedades declaradas en el POM por el usuario en forma `<properties><x.y.z>value</x.y.z></properties>`

○ **Documentación**

Maven puede generar un sitio con toda la documentación del proyecto, en la que se incluyen la información básica del proyecto y las dependencias del proyecto, además permite la integración con herramientas de Integración Continua, Gestión de problemas, listas de correo así como la utilización de reportes personalizados mediante la configuración de plugins. Entre esos reportes personalizados se puede incluir: ¹⁷

¹⁷ <http://www.dosideas.com>

- Resultado de pruebas unitarias
- Reportes de Checkstyle
- Reportes de PMD
- Reportes de Cobertura
- Reportes de Macker
- Para crear dicho site hay que utilizar la siguiente sentencia
- mvn site
- mvn site:stage -DstagingDirectory=<directorio destino>

2.4.2 SERVIDOR DE APLICACIONES

- **Jboss AS 7**

Es un servidor de aplicaciones pionero en la arquitectura JEE6 de código abierto implementado en Java . Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en abril del 2006, fundada por Marc Fleury. El proyecto está apoyado por una red mundial de colaboradores al ser código abierto.

Los cambios en el kernel de JBoss 7 estan basados principalmente en los siguientes proyectos:

JBoss Modules: Éste se encarga de cargar las clases de los recursos contenidos en el contenedor. Se puede pensar en JBoss Modules como una envoltura delgada de arranque, para la ejecución de una aplicación en un entorno modular.

Modular Service Container (MSC): Éste proporciona una manera de instalar, desinstalar y administrar los servicios utilizados por un contenedor. MSC permite además la inyección de recursos en los servicios y la gestión de dependencia entre servicios.

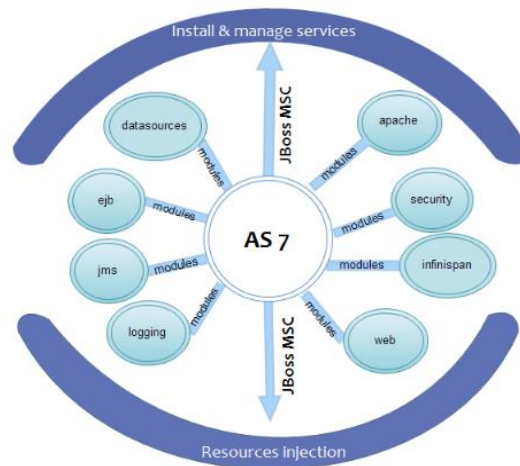


Figura 9: Servicios proporcionados por Jboss AS7

- **Modos de Operación**

Hasta ahora estuvimos hablando de JBoss como una instancia independiente que se ejecuta con el comando standalone.sh, pero en JBoss AS 7 existen dos formas de operación, las cuales se explican a continuación:

- **Standalone**

Para muchos casos, la capacidad de manejo centralizado no es necesario. Para estos casos, una instancia de JBoss Application Server 7 se puede ejecutar como un "standalone server". Una instancia de standalone server es un proceso independiente, más parecido a lo que es una instancia de JBoss Application Server 3, 4, 5, o 6. Estas instancias se

ejecutan usando el script de inicio standalone.sh (standalone.bat para windows).

Si se necesita ejecutar más de una instancia es posible con el modo standalone pero la responsabilidad de la administración de los nodos queda del lado del administrador. Por ejemplo, para deployar una aplicación en un grupo de servidores standalone, se tendrá que hacer el deploy en cada uno de los nodos.

- **Managed Domain**

Una de las nuevas funcionalidades principales de JBoss AS 7 es poder manejar múltiples instancias de JBoss Application Server 7 desde un mismo punto de control. En un conjunto de servidores cada uno es referido como un miembro de un "dominio" (domain) con un único proceso controlador de dominio (Domain Controller) actuando como punto de administración central. Todas las instancias de JBoss Application Server 7 en el mismo dominio comparten una política de administración común, con el Domain Controller el cual se asegura de que cada servidor esté configurado acorde a esta política.

Cuando se ejecuta JBoss Application Server 7 en el modo "managed domain" (via el script de inicio domain.sh), se intenta ejecutar un Host Controller y al menos una instancia JBoss Application Server 7. Uno de estos Host Controller tiene que ser configurado para actuar como el Domain Controller.

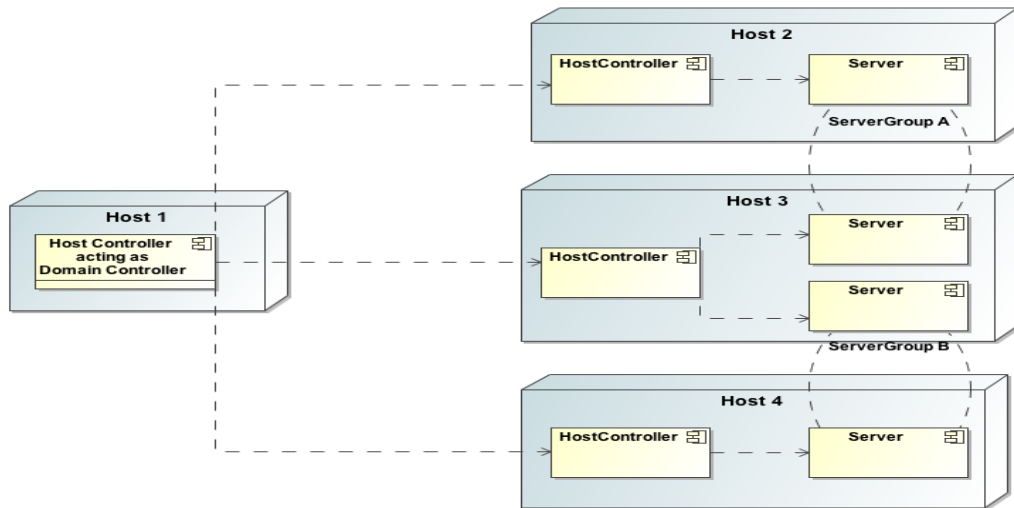


Figura 10: Ejemplo de la topología Managed Domain

- **Estructura de directorios**

Apenas se ve la estructura de directorios se nota que hay un cambio importante a diferencia de versiones anteriores. Básicamente se encuentra dividido en dos partes principales que muestran la diferencia entre standalone y domain.

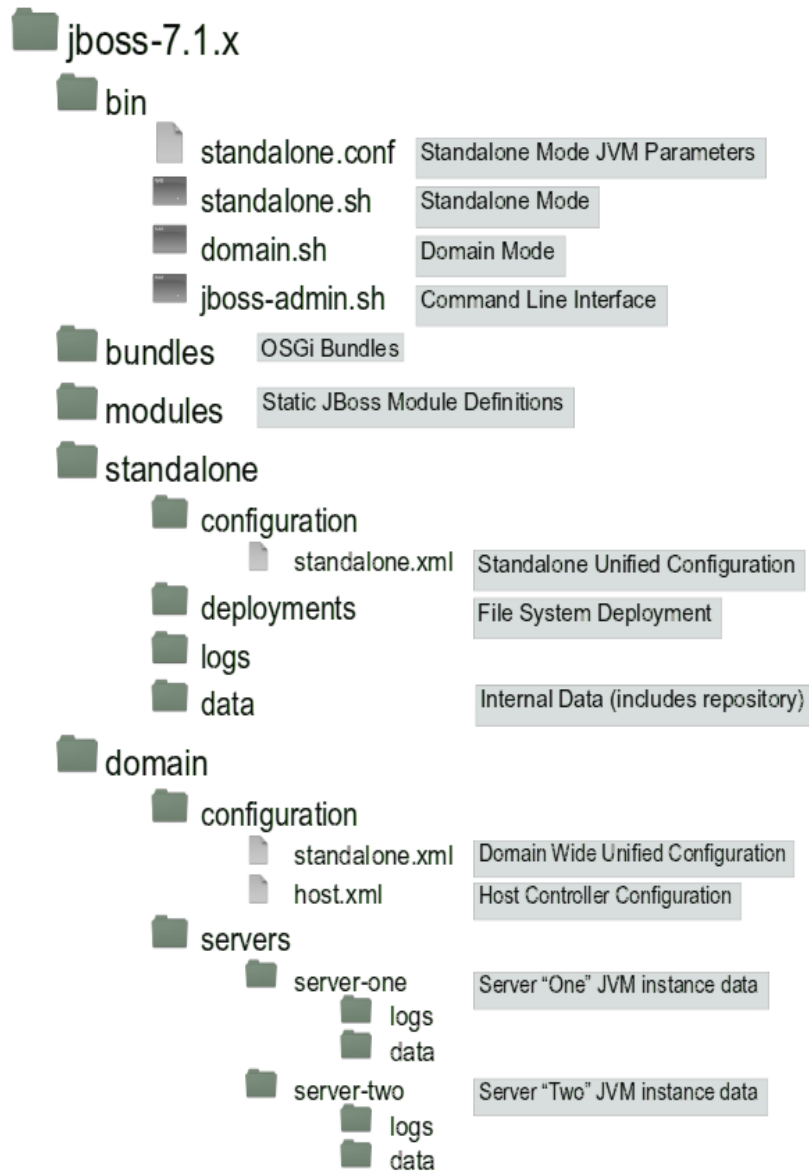


Figura 11: Distribución de Directorios en JBOSS AS7

Tabla 5: Detalle de la Estructura de Directorios en JBOSS AS 7

Directorio	Descripción
bin	Scripts de arranque con sus archivos de configuración.
bundles	OSGi bundles
docs/schema	Esquemas XML y archivos de definición
domain	Archivos de configuración, contenido desplegado, y áreas escribibles por procesos del modo domain.
modules	Como JBoss AS 7 está basado en una arquitectura de classloading modular, varios módulos usados en el server se encuentran aquí.
standalone	Archivos de configuración, contenido desplegado, y áreas escribibles por server en modo standalone.
welcome-content	Página de bienvenida por defecto

2.4.3 HERRAMIENTAS PARA LA CAPA WEB

- **Rich Faces 4**

Es una biblioteca de código abierto basada en Java y desarrollada por JBoss que permite crear aplicaciones web con Ajax. Trabaja en conjunto con el framework de Java Server Faces(JSF). Se implementa unos filtros para permitir peticiones Ajax en la página.

La ventaja de Ajax es que provoca una ejecución con el servidor para finalmente realizar una renderización parcial o total de una página web, de esta forma el control de lo que sucede está en el servidor.

- **Requerimientos Técnicos:**

Los requerimientos mínimos para poder trabajar con RichFaces son los siguientes:

- Java Development Kit (JDK) 6 o mayor
- Un servidor de aplicaciones web como puede ser JBoss Application Server 7
- Un browser, como Firefox 7, Chrome 14, o Internet Explorer 9

- **Librerías y Dependencias**

- RichFaces libraries
- richfaces-core-api.jar
- richfaces-core-impl.jar
- richfaces-components-api.jar
- richfaces-components-ui.jar

El framework tiene dependencias con terceros que pueden ser clasificadas en mandatorias y opcionales como se muestra a continuación:

- **Dependencias de terceros mandatorias.**

- **Java Server Faces 2.x implementación**

- javax.faces.jar (version 2.1.5 o mayor)
- myfaces-impl.jar (version 2.1.5 o mayor)

- **Google Guava**

- guava.jar (version 10.0.1)
- CSS Parser
- cssparser.jar (version 0.9.5)

- **Simple API for CSS**

- sac.jar (version 1.3)

- **Dependencias de terceros opcionales**
 - **Bean validation (JSR-303) integración del lado del cliente (JSR-303 API e implementación)**
 - validation-api.jar (version 1.0.0.GA)
 - hibernate-validator.jar (version 4.2.0.Final o mayor)
 - **Transport library - Atmosphere (sin dependencias)**
 - atmosphere-runtime.jar (version 0.8.4)
 - **JMS integration (JMS API e implementación)**
 - jms.jar (version 1.1)
 - hornetq-jms.jar (version 2.2.7.Final o mayor)
 - **CDI integration (CDI API e implementación)**
 - cdi-api.jar (version 1.0-SP4)
 - javax.inject.jar (version 1)
 - jsr-250-api.jar (version 1.0)
 - weld-servlet.jar (version 1.1.4.Final)
 - **Extended caching (EhCache)**
 - ehcache.jar (version 1.6.0)

2.4.4 HERRAMIENTAS DE BASE DE DATOS

- **PostgreSQL**

Es una base de datos objeto relacional (ORDBMS) que lleva aproximadamente 15 años en el mercado, ha ganado una amplia fama y está disponible para múltiples sistemas operativos, entre ellos: Windows y UNIX.

Este tipo de manejadores de base de datos añaden características al modelo relacional clásico, entre ellas el soporte de herencia, con lo cual se acerca más al paradigma de la orientación a objetos.¹⁸

Tabla 6: Límites establecidos para POSTGRESQL

Límite	Valor
Tamaño máximo de la base de datos	Ilimitado
Máximo tamaño para una tabla	32 TB
Máximo tamaño para una fila	1.6 TB
Máximo tamaño para un campo	1 GB
Cantidad máxima de filas por tabla	Ilimitado
Cantidad máxima de columnas por tabla	250 - 1600 dependiendo del tipo de dato de las columnas
Cantidad máxima de Índices en una tabla	Ilimitado

Fuente: <http://www.postgresql.org/about/>

¹⁸<http://www.slideshare.net/armandorob/objeto-relacional-bases-datos-2>
<http://www.openecuator.org/modules/news/article.php?storyid=31>
http://docs.jboss.org/richfaces/latest_4_2_X/Developer_Guide/en-US/html/

CAPÍTULO 3

PLANIFICACIÓN DISEÑO

3.1 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

3.1.1 HISTORIAL DE REVISIONES

Tabla 7: Historial de Revisiones Especificación de Requerimientos

Fecha	Versión	Descripción	Autor
04/03/2013	1.0	Especificación de requerimientos de software	Paúl Jarrín Nelson Jumbo
23/03/2013	2.0	Especificación de requerimientos de software	Paúl Jarrín Nelson Jumbo

3.1.2 INTRODUCCIÓN

3.1.2.1 Propósito

El propósito es definir los requerimientos funcionales para el sistema de Gestión de Donaciones y Voluntariado para la “FUNDACIÓN JÓVENES CONTRA EL CÁNCER”.

3.1.2.2 Alcance

El presente proyecto permitirá el desarrollo del sistema para la gestión de donaciones y voluntarios para la fundación de jóvenes contra el cáncer.

De esta forma, al término del mismo, la fundación contará con una herramienta más para seguir llevando a cabo su labor social de una manera más eficaz y efectiva, además al no tener ningún costo para la fundación, éste podrá ser destinado a seguir ayudando a sus guerreros.

Los siguientes módulos permitirán llevar de una manera más ordenada los procesos que lleva a cabo la fundación.

El sistema de la gestión de donaciones y voluntarios para la fundación de jóvenes contra el cáncer, dispondrá de los siguientes 9 módulos:

- Módulo de Seguridad y Acceso
- Módulo de Gestión de Voluntarios y Donantes
- Módulo Gestión de Donaciones
- Módulo para Gestión de Guerreros
- Módulo para la Administración y Aprobación de Eventos
- Módulo para la Asignación de Donaciones
- Módulo de Reportes Dinámicos
- Módulo de Administración de Catálogos

3.1.2.3 Descripción General

Dado que los sistemas de información actualmente son una herramienta de apoyo en las organizaciones es muy importante sistematizar los procesos que abarcan el negocio de la organización, es por esta razón que la “FUNDACIÓN JÓVENES CONTRA EL CÁNCER” ha decidido que se desarrolle un sistema para automatizar los procesos definidos en el alcance de este documento.

3.1.2.4 Perspectiva del producto

El aplicativo web de gestión de donaciones y voluntariado para la fundación de JÓVENES CONTRA EL CÁNCER permitirá a los miembros de la misma, gestionar la información de los guerreros,

voluntarios, eventos, donaciones de una manera centralizada y ordenad, para que los usuarios de la fundación puedan tener un rápido acceso a la información generada por estos procesos, dejando atrás el almacenamiento de los datos en papel.

3.1.2.5 Interfaces del sistema

El sistema corre sobre un servidor de aplicaciones con una base de datos postgresql y todo esto está sobre un sistema operativo centos, para poder hacer uso del sistema se necesita de un explorador web: como Firefox Google Chrome.

3.1.2.6 Interfaces del Usuario

Pantalla de acceso al sistema

The image shows a login interface with a green header containing the text 'Ingreso al sistema'. Below the header, there are two text input fields. The first is labeled 'Usuario' and the second is labeled 'Contraseña'. To the right of the 'Contraseña' field is a button labeled 'Login'. At the bottom of the interface, the text 'JCCE-Sistema de Gestion de Donaciones' is displayed.

Figura 12 Pantalla de ingreso al sistema

3.1.2.7 Plantilla para los demás formularios del sistema



Figura 13: Plantilla para las pantallas del sistema

3.1.2.8 Interfaces de Hardware

Requerimientos Mínimos

- Procesador Intel Core2Duo a 2.0GHz o equivalente.
- 2GB de RAM
- Tarjeta de red Ethernet 10/100.

Requerimientos óptimos

- Procesador Intel Core i7 a 2.0GHz o equivalente.
- 8GB de RAM
- Tarjeta de red Ethernet 10/100/1000, o tarjeta de red inalámbrica 802.11a-b-g-n.

3.1.2.9 Interfaces de Software

Para poner en funcionamiento el sistema se necesita de lo siguiente:

- Base de datos postgresql 8 o superior
- JDK (Java Development Kit) versión 1.6
- Servidor Jboss EAP 6.0
- Explorador Web

3.1.2.10 Funciones del producto

El sistema de gestión de donaciones y voluntariado para la FUNDACIÓN JÓVENES CONTRA EL CÁNCER, debe estar en la capacidad de realizar los siguientes procesos:

- Administración y búsqueda de Guerreros
- Administración, búsqueda y aprobación de Voluntarios.
- Administración de donantes
- Administración de donaciones
- Administración de Catálogos
- Manejo de asignaciones de las donaciones
- Auditoria del sistema
- Manejo de Inventarios
- Generación de reportes
 - Asignaciones por Fecha
 - Guerreros por prioridad
 - Actividades de los voluntarios
 - Auditoria
 - Horas de voluntariado
 - Donaciones por fecha
 - Guerreros por clasificación
 - Movimientos de inventario
- Administración de eventos y de invitaciones a los mismos
- Manejo de usuario y roles

3.1.2.11 Características del Usuario

- El usuario deberá tener conocimientos básicos de computación
- El administrador será el encargado de asignar los diferentes roles a cada uno de los usuarios que manejaran el sistema.
- El administrador deberá tener conocimientos básicos de un sistema operativo Linux.

3.1.2.12 Restricciones

- El sistema no registrara las donaciones monetarias que se hacen a la fundación.
- El sistema no permitirá desactivar el módulo de auditoria
- El sistema no permitirá un borrado físico de la información
- El sistema no tendrá un funcionamiento correcto en versiones antiguas de Internet Explorer.

3.1.2.13 Asunciones y dependencias

- El sistema operativo deberá contar con JDK versión 1.6
- La base de datos será postgresql.
- Servidor de aplicaciones jboss EAP 6

3.1.3 REQUERIMIENTOS ESPECÍFICOS

3.1.3.1 Requisitos de Interfaces externas

3.1.3.1.1 Interfaz con el usuario.

Debe ser amigable para que permita al usuario interactuar de manera fácil con el sistema.

3.1.3.1.2 Interfaz con el hardware.

El equipo debe contar con una tarjeta de red, mouse, teclado y un monitor con una resolución mínima de 1024 x 768.

3.1.3.1.3 Interfaz de comunicaciones.

El sistema podrá ser operable en red con la infraestructura que cuente la empresa del estilo cliente-servidor., en este caso, se debe basar en el protocolo TCP/IP.

3.1.3.2 Requerimientos Funcionales

Id. Requerimiento	3.1 Crear Voluntario
Descripción	Permitir ingresar un nuevo voluntario
Entradas	Datos Personales: Tipo, Nombres, Apellidos, Dirección del domicilio, Teléfono Domicilio, Teléfono Celular, Email, Provincia, Ciudad, Fecha de registro, Edad, Horas de voluntariado, Cedula Datos Académicos: Título, Institución, Habilidades y Competencias, Idiomas, Otros Experiencia Laboral: Cargo, Empresa, Cargo actual o no. Preguntas: Respuestas Horarios: Horario temprano, horario tarde.
Salidas	Confirmación de voluntarios registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos voluntarios, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de voluntario.
Postcondiciones	El voluntario queda registrado en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del

	voluntario.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.2 Modificar Voluntario
Requerimiento	
Descripción	Permitir modificar un voluntario ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de voluntario modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de voluntario y haber ingresado el registro del voluntario.
Postcondiciones	El voluntario modificado en el sistema.
Efectos Colaterales	Se actualiza el voluntario en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.3 Eliminar Voluntario
Requerimiento	
Descripción	Permitir eliminar un voluntario ya ingresado
Entradas	Acción del usuario sobre el botón eliminar
Salidas	Confirmación de voluntario eliminado exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un estado del registro, con lo cual el voluntario queda desactivado.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de voluntario y haber ingresado el registro del voluntario.
Postcondiciones	El alumno queda en estado desactivado por lo cual no va a aparecer en el sistema.
Efectos Colaterales	Se actualiza el voluntario en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.4 Buscar Voluntario
Requerimiento	
Descripción	Permite encontrar un voluntario por cedula, nombre o apellido
Entradas	Cedula o nombre o usuario y Acción del usuario sobre el botón buscar.
Salidas	Voluntarios correspondientes a los criterios

	de búsqueda.
Proceso	El sistema busca y muestra los voluntarios con los criterios de búsqueda ingresados, si no se especifica ningún criterio el sistema encontrara todos los voluntarios activos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de voluntario y haber ingresado el registro del voluntario.
Postcondiciones	Ninguna
Efectos Colaterales	Ninguno
Prioridad	Media
Rol que lo ejecuta	Usuario

Id.	3.5 Aprobar Voluntario
Requerimiento	
Descripción	Permite aprobar un voluntario previamente ingresado.
Entradas	Acción del usuario sobre el botón Aprobar.
Salidas	Confirmación de usuario Aprobado en el sistema por un mensaje
Proceso	El sistema busca el registro que se selecciona para aprobar, y lo modifica en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de voluntario y voluntario en estado no aprobado.

Postcondiciones	Voluntario queda aprobado en el sistema.
Efectos	Se modifica un registro en la base de datos del sistema.
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario.

Id.	3.6 Crear Donante
Requerimiento	
Descripción	Permitir ingresar un nuevo donante
Entradas	<p>Si el tipo de donante es: Persona Nombre, apellido, número de identificación, dirección y teléfono.</p> <p>Si el tipo de donante es: Empresa Nombre, Número de Identificación, Dirección, Teléfono y Responsable.</p>
Salidas	Confirmación de donante registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos donantes, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de donantes.
Postcondiciones	El donante queda registrado en el sistema
Efectos	Se ingresa un nuevo registro en la base de

Colaterales	datos del sistema con los datos del donante.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.7 Modificar Donante
Requerimiento	
Descripción	Permitir modificar un donante ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de donante modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de donante y haber ingresado el registro del donante.
Postcondiciones	Donante modificado en el sistema.
Efectos	Se actualiza el donante en el cliente de base de datos
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.9 Eliminar Donante
Requerimiento	
Descripción	Permitir eliminar un donante ya ingresado
Entradas	Acción del usuario sobre el botón eliminar
Salidas	Confirmación de donante eliminado exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un estado del registro, con lo cual el donante queda desactivado.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de donante y haber ingresado el registro del donante.
Postcondiciones	El donante queda en estado desactivado por lo cual no se va a visualizar en el sistema.
Efectos Colaterales	Se actualiza el donante en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.10 Crear Donación
Requerimiento	
Descripción	Permitir ingresar una nueva donación
Entradas	Donante, Tipo inventario, Inventario y Cantidad.
Salidas	Confirmación de donación registrada exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el

registro de nuevas donaciones, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.

Precondiciones Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de donaciones.

Postcondiciones La donación queda registrado en el sistema

Efectos Colaterales Se ingresa un nuevo registro en la base de datos del sistema con los datos de la donación.

Prioridad Alta

Rol que lo ejecuta Usuario

Id. 3.11 Modificar Donación

Requerimiento

Descripción Permitir modificar una donación ya ingresado

Entradas Acción del usuario sobre el botón actualizar

Salidas Confirmación de donación modificada exitosamente con un mensaje

Proceso El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.

Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de donación y haber ingresado el registro de la donación.
Postcondiciones	Donación modificada en el sistema.
Efectos Colaterales	Se actualiza la donación en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.12 Crear Tipo Inventario
Requerimiento	
Descripción	Permitir ingresar un nuevo tipo de inventario
Entradas	Nombre y Descripción
Salidas	Confirmación de tipo de inventario registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos tipos de inventario, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de tipo de inventario.
Postcondiciones	El tipo de inventario queda registrado en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del tipo de inventario.

Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.13 Modificar Tipo Inventario
Requerimiento	
Descripción	Permitir modificar un tipo de inventario ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de tipo de inventario modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de tipo de inventario y haber ingresado el registro del tipo de inventario.
Postcondiciones	Tipo de inventario modificado en el sistema.
Efectos Colaterales	Se actualiza el tipo de inventario en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.14 Eliminar Tipo Inventario
Requerimiento	
Descripción	Permitir eliminar un tipo de inventario ya ingresado
Entradas	Acción del usuario sobre el botón eliminar

Salidas	Confirmación del tipo de inventario eliminado exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un estado del registro, con lo cual el tipo de inventario queda desactivado.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de tipo de inventario y haber ingresado el registro del tipo de inventario.
Postcondiciones	El tipo de inventario queda en estado desactivado por lo cual no se va a visualizar en el sistema.
Efectos Colaterales	Se actualiza el tipo de inventario en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.15 Crear Inventario
Requerimiento	
Descripción	Permitir ingresar un nuevo inventario
Entradas	Nombre, Descripción, Unidades Disponibles y Tipo de inventario
Salidas	Confirmación de inventario registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos inventarios, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el

	sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de inventario.
Postcondiciones	El inventario queda registrado en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del inventario.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.16 Modificar Inventario
Requerimiento	
Descripción	Permitir modificar un inventario ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de inventario modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de inventario y haber ingresado el registro del inventario.
Postcondiciones	Inventario modificado en el sistema.
Efectos	Se actualiza el inventario en el cliente de

Colaterales	base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.17 Eliminar Inventario
Requerimiento	
Descripción	Permitir eliminar un inventario ya ingresado
Entradas	Acción del usuario sobre el botón eliminar
Salidas	Confirmación del inventario eliminado exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un estado del registro, con lo cual el inventario queda desactivado.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de inventario y haber ingresado el registro del inventario.
Postcondiciones	El inventario queda en estado desactivado por lo cual no se va a visualizar en el sistema.
Efectos	Se actualiza el inventario en el cliente de base de datos
Colaterales	base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.18 Crear Rol
Requerimiento	
Descripción	Permitir ingresar un nuevo rol
Entradas	Nombre, Descripción y Accesos (Menús que puede visualizar)
Salidas	Confirmación de rol registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos roles, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de rol.
Postcondiciones	El rol queda registrado en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del rol.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.19 Modificar Rol
Requerimiento	
Descripción	Permitir modificar un rol ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de rol modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se

	selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de rol y haber ingresado el registro del rol.
Postcondiciones	Rol modificado en el sistema.
Efectos Colaterales	Se actualiza el rol en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id. 3.20 Crear Usuario

Requerimiento

Descripción	Permitir ingresar un nuevo usuario
Entradas	Cedula, Nick, Nombre, Correo Electrónico, Correo Opcional, Clave
Salidas	Confirmación de usuario registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos usuarios, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de usuario.
Postcondiciones	El usuario queda registrado en el sistema

Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del usuario.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id. 3.21 Modificar Usuario

Requerimiento

Descripción	Permitir modificar un usuario ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de usuario modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de usuario y haber ingresado el registro del usuario.
Postcondiciones	Usuario modificado en el sistema.
Efectos Colaterales	Se actualiza el usuario en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id. 3.22 Crear Tipo Catálogo

Requerimiento

Descripción	Permitir ingresar un nuevo tipo catálogo
Entradas	Nombre y Nemónico

Salidas	Confirmación de tipo catálogo registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevos tipos de catálogo, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de tipo de catálogos.
Postcondiciones	El tipo catálogo queda registrado en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos del tipo de catálogo.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.23 Crear Catálogo
Requerimiento	
Descripción	Permitir ingresar un nuevo catálogo
Entradas	Nombre ,Nemónico y Tipo de catálogo
Salidas	Confirmación de catálogo registrado exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de nuevo catálogo, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el

	sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de catálogos.
Postcondiciones	El catálogo queda registrado en el sistema
Efectos	Se ingresa un nuevo registro en la base de datos del sistema con los datos del catálogo.
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.24 Modificar Catálogo
Requerimiento	
Descripción	Permitir modificar un catálogo ya ingresado
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de catálogo modificado exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de catálogo y haber ingresado el registro del catálogo.
Postcondiciones	Catálogo modificado en el sistema.
Efectos	Se actualiza el catálogo en el cliente de base de datos
Colaterales	

Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.25 Eliminar Catálogo
Requerimiento	
Descripción	Permitir eliminar un catálogo ya ingresado
Entradas	Acción del usuario sobre el botón eliminar
Salidas	Confirmación del catálogo eliminado exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un estado del registro, con lo cual el catálogo queda desactivado.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de catálogo y haber ingresado el registro del catálogo.
Postcondiciones	El catálogo queda en estado desactivado por lo cual no se va a visualizar en el sistema.
Efectos Colaterales	Se actualiza el catálogo en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario
Id.	3.26 Crear Provincia
Requerimiento	
Descripción	Permitir ingresar una nueva provincia
Entradas	Nombre

Salidas	Confirmación de provincia registrada exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de una nueva provincia, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de provincia.
Postcondiciones	La provincia queda registrada en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos de la provincia.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.27 Modificar Provincia
Requerimiento	
Descripción	Permitir modificar una provincia ya ingresada
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de la provincia modificada exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos

	que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de provincia y haber ingresado el registro de la provincia.
Postcondiciones	Provincia modificada en el sistema.
Efectos	Se actualiza la provincia en el cliente de base de datos
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.28 Buscar Provincia
Requerimiento	
Descripción	Permitir buscar una provincia ingresada
Entradas	Nombre y Acción del usuario sobre el botón buscar
Salidas	Provincias correspondientes a los criterios de búsqueda.
Proceso	El sistema busca y muestra las provincias con los criterios de búsqueda ingresados, si no se especifica ningún criterio el sistema encontrara todas las provincias activas.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de provincia y haber ingresado el registro de una provincia.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	

Prioridad	Media
Rol que lo ejecuta	Usuario

Id.	3.29 Crear Ciudad
Requerimiento	
Descripción	Permitir ingresar una nueva ciudad
Entradas	Nombre y Provincia
Salidas	Confirmación de ciudad registrada exitosamente con un mensaje
Proceso	El sistema despliega un formulario para el registro de una nueva ciudad, en el cual se debe ingresar los datos descritos anteriormente en entradas, los mismos datos que se validan y se registran en el sistema.
Precondiciones	Estar logueado en el sistema y tener el rol adecuado para visualizar la página de ficha de ciudad.
Postcondiciones	La ciudad queda registrada en el sistema
Efectos Colaterales	Se ingresa un nuevo registro en la base de datos del sistema con los datos de la ciudad.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.30 Modificar Ciudad
Requerimiento	
Descripción	Permitir modificar una ciudad ya ingresada
Entradas	Acción del usuario sobre el botón actualizar
Salidas	Confirmación de la ciudad modificada exitosamente con un mensaje
Proceso	El sistema busca el registro que se selecciona para modificar, lo muestra, el usuario ingresa los cambios, los mismos que se guardan en la base de datos.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de ciudad y haber ingresado el registro de la ciudad.
Postcondiciones	Ciudad modificada en el sistema.
Efectos	Se actualiza la ciudad en el cliente de base de datos
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.31 Eliminar Ciudad
Requerimiento	
Descripción	Permitir eliminar un ciudad ya ingresado
Entradas	Acción del usuario sobre el botón eliminar
Salidas	Confirmación de ciudad eliminada exitosamente con un mensaje
Proceso	El sistema busca el registro seleccionado para eliminar, pregunta si verdaderamente se lo desea eliminar y se actualiza un

	estado del registro, con lo cual la ciudad queda desactivada.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de ciudad y haber ingresado el registro de la ciudad.
Postcondiciones	La ciudad queda en estado desactivado por lo cual no se va a visualizar en el sistema.
Efectos Colaterales	Se actualiza la ciudad en el cliente de base de datos
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.32 Buscar Ciudad
Requerimiento	
Descripción	Permitir buscar una ciudad ingresada
Entradas	Nombre y Acción del usuario sobre el botón buscar
Salidas	Ciudades correspondientes a los criterios de búsqueda.
Proceso	El sistema busca y muestra las ciudades con los criterios de búsqueda ingresados, si no se especifica ningún criterio el sistema encontrara todas las ciudades activas.
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de ficha de ciudad y haber ingresado el registro de una ciudad.
Postcondiciones	Ninguna

Efectos Colaterales	Ninguno
Prioridad	Media
Rol que lo ejecuta	Usuario
Id.	3.33 Crear Guerrero
Requerimiento	
Descripción	Permitir registrar un guerrero en el sistema
Entradas	Requerimientos: 3.34, 3.38,3.39,3.40,3.41
Salidas	Confirmación de registro ingresado exitosamente
Proceso	El sistema despliega el formulario para registro de nuevos guerreros, en el cual se registran los datos de Entrada.
Precondiciones	El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de ingreso de guerreros.
Postcondiciones	El guerrero queda ingresado en el sistema
Efectos Colaterales	Se ingresan nuevos registros en la base de datos correspondientes a la información que se ingresó del mismo.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.34 Registrar datos personales del guerrero
Requerimiento	
Descripción	Permitir registrar los datos personales de un guerrero
Entradas	Nombres, Apellidos, Fecha de Nacimiento,

Edad, Provincia de Nacimiento, Ciudad de Nacimiento, Email, Cédula de identidad, Requerimiento 3.35, Dirección de Domicilio, Provincia de Domicilio, Ciudad de Domicilio, Teléfono domicilio, Teléfono Celular, Costea o no sus gastos, Recibe o no el Bono Joaquín Gallegos Lara, Creencia o religión, Deporte que practica, Hobby, Comida favorita, Requerimiento 3.36, Requerimiento 3.37, Fecha de registro, Clasificación.

Salidas Mensajes indicando los campos que son obligatorios.

Proceso El sistema despliega el formulario que permite el ingreso de las entradas que se han detallado.

Precondiciones Ninguna.

Postcondiciones El usuario puede proceder a ingresar los Contactos del Guerrero.

Efectos Ninguno.

Colaterales

Prioridad Alta

Rol que lo ejecuta Psicólogo

Id. Requerimiento 3.35 Registrar Carnet del CONADIS del guerrero.

Descripción Permitir registrar los datos del carnet del CONADIS del guerrero.

Entradas Número de carnet, Tipo de discapacidad,

Porcentaje de discapacidad.

Salidas Confirmación de que el registro se ha ingresado correctamente.

Proceso El sistema muestra una ventana emergente que permite ingresar la información del carnet del CONADIS.

Precondiciones El usuario debe encontrarse en la pantalla de Datos Generales del Guerrero.

Postcondiciones El guerrero tiene un carnet del CONADIS registrado.

Efectos Colaterales Se ingresa en la base de datos la información del carnet del CONADIS asociado al guerrero.

Prioridad Alta

Rol que lo ejecuta psicólogo

Id. **3.36 Registrar Lugares del Ecuador que desea conocer el guerrero.**

Descripción Permitir registrar los lugares dentro del Ecuador que desea conocer el guerrero.

Entradas Nombre del lugar que desea conocer.

Salidas Confirmación de que el registro se ha ingresado correctamente.

Proceso El sistema muestra una ventana emergente que permite ingresar la información de los lugares que desea conocer el guerrero.

Precondiciones El usuario debe encontrarse en la pantalla

	de Datos Generales del Guerrero.
Postcondiciones	El guerrero tiene registrados los lugares del Ecuador que desea conocer.
Efectos Colaterales	Se ingresa en la base de datos la información de los lugares del Ecuador que desea conocer un guerrero.
Prioridad	Alta
Rol que lo ejecuta	psicólogo

Id.	3.37 Registrar los deseos del guerrero
Requerimiento	
Descripción	Permitir registrar los deseos que un guerrero quiere cumplir.
Entradas	Descripción del deseo del guerrero.
Salidas	Confirmación de que el registro se ha ingresado correctamente.
Proceso	El sistema muestra una ventana emergente que permite ingresar la información de los deseos del guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Datos Generales del Guerrero.
Postcondiciones	El guerrero tiene registrados sus deseos.
Efectos Colaterales	Se ingresa en la base de datos la información de los deseos del guerrero.
Prioridad	Alta
Rol que lo ejecuta	psicólogo

Id.	3.38 Registrar los contactos del guerrero
Requerimiento	
Descripción	Permitir registrar los contactos de un guerrero.
Entradas	Cédula, Nombres, Apellidos, Edad, Trabaja en, Teléfono domicilio, Teléfono Celular, Tipo de Contacto, Parentezco, Vive o no con el guerrero.
Salidas	Confirmación de que el registro se ha ingresado correctamente.
Proceso	El sistema muestra una ventana emergente que permite ingresar información de un contacto del guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Contactos del guerrero
Postcondiciones	El guerrero tiene registrados sus contactos.
Efectos Colaterales	Se ingresa en la base de datos la información de los contactos del guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.39 Registrar los Datos Académicos del Guerrero
Requerimiento	
Descripción	Permitir registrar los Datos Académicos del Guerrero
Entradas	Estudia o no actualmente, Actual o última institución, Nivel o Curso, Especialidad/Carrera, Tipo de institución, Recibe o no beca.
Salidas	Ninguna

Proceso	El sistema muestra un formulario para ingresar los datos académicos del guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Datos Académicos
Postcondiciones	El guerrero tiene registrados sus datos académicos Se puede pasar a la sección para ingreso de datos laborales del guerrero
Efectos Colaterales	Se ingresa en la base de datos la información de los datos académicos del guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo
Id.	3.40 Registrar los Datos Laborales del Guerrero
Requerimiento	Guerrero
Descripción	Permitir registrar los Datos Laborales del Guerrero
Entradas	Trabaja o no Actualmente, Empresa, Cargo, Área, Desde, Tiene o no IESS
Salidas	Ninguna
Proceso	El sistema muestra un formulario para ingresar los datos laborales del guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Datos Laborales
Postcondiciones	El guerrero tiene registrados sus datos laborales Se puede pasar a la sección de Condición Médica del guerrero
Efectos Colaterales	Se ingresa en la base de datos la información de los datos laborales del

	guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id. Requerimiento	3.41 Registrar la Condición Médica del Guerrero
Descripción	Permite registrar la información correspondiente a la Condición Médica del Guerrero.
Entradas	Requerimientos: 3.42,3.43,3.44
Salidas	Ninguna
Proceso	El sistema muestra un formulario para ingresar la información de la Condición Médica del Guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Condición Médica
Postcondiciones	El guerrero tiene registrados los datos sobre Condición Médica
Efectos Colaterales	Se ingresa en la base de datos la información de la condición médica del guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id. Requerimiento	3.42 Registrar Datos generales de Condición Médica del Guerrero
Descripción	Permite registrar la información general sobre la Condición Médica del Guerrero

Entradas	Etapa, Diagnóstico, Se encuentra o no en Tratamiento, Condición Física, Medicamentos, Observaciones, Ha abandonado o no su tratamiento, Razón por la cual ha abandonado su tratamiento.
Salidas	Ninguna
Proceso	El sistema muestra un formulario para ingresar la información de la Condición Médica del Guerrero.
Precondiciones	El usuario debe encontrarse en la pantalla de Condición Médica
Postcondiciones	El guerrero tiene registrados los datos sobre Condición Médica
Efectos Colaterales	Se ingresa en la base de datos la información de la condición médica del guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.43 Registrar Quimioterapias que ha recibido el guerrero
Requerimiento	
Descripción	Permite registrar la información sobre las quimioterapias que ha recibido el guerrero
Entradas	Fecha Inicio, Fecha Fin, Hospital
Salidas	Ninguna
Proceso	El sistema muestra un formulario para ingresar la información sobre las quimioterapias que ha recibido el guerrero
Precondiciones	El usuario debe encontrarse en la pantalla de Condición Médica

Postcondiciones	El guerrero tiene registrados los datos sobre las quimioterapias que ha recibido.
Efectos Colaterales	Se ingresa en la base de datos la información de las quimioterapias que ha recibido el guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.44 Registrar Deudas por tratamiento que ha adquirido el guerrero
Requerimiento	
Descripción	Permite registrar la información sobre las deudas que el guerrero ha adquirido para poder llevar a cabo su tratamiento
Entradas	Total de la deuda, Valor mensual que abona, Forma de pago, Institución a la que adeuda.
Salidas	Ninguna
Proceso	El sistema muestra un formulario para ingresar la información sobre las deudas que ha adquirido el guerrero
Precondiciones	El usuario debe encontrarse en la pantalla de Condición Médica
Postcondiciones	El guerrero tiene registrados los datos sobre las deudas que ha adquirido.
Efectos Colaterales	Se ingresa en la base de datos la información de las deudas que ha adquirido el guerrero.
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.45 Buscar Guerrero
Requerimiento	
Descripción	Permite buscar los guerreros registrados en el sistema.
Entradas	Cédula, Nombres, Apellidos
Salidas	Los guerreros que correspondan con los criterios del búsqueda
Proceso	El sistema muestra un formulario en el cual se ingresa información para buscar guerreros.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla de guerreros.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.46 Eliminar Guerreros
Requerimiento	
Descripción	Permite hacer un borrado lógico de los guerreros registrados en el sistema.
Entradas	Acción sobre el botón eliminar
Salidas	Mensaje indicando que se ha eliminado exitosamente el registro
Proceso	El sistema muestra una ventana emergente para confirmar la eliminación

	del guerrero.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla de guerreros.
Postcondiciones	El estado del guerrero pasa a INACTIVO
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Psicólogo

Id.	3.47 Crear Evento
Requerimiento	
Descripción	Permite registrar la información sobre los distintos eventos que realiza la fundación.
Entradas	Nombre, Lugar, Descripción, Fecha Inicio, Fecha Fin, Hora Inicio, Hora Fin, Requerimiento 3.48
Salidas	Mensaje indicando que se ha guardado correctamente el evento.
Proceso	El sistema muestra un formulario para ingresar información sobre el evento que desea realizar.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a esta pantalla
Postcondiciones	El evento queda registrado con estado REPROBADO

Efectos Colaterales	Se ingresa en la base de datos la información del evento a realizarse.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id. 3.48 Manejar invitaciones a eventos

Requerimiento

Descripción	Permite llevar un control de las invitaciones que se hacen tanto a guerreros como a voluntarios a los eventos que realiza la fundación, se invita a todos los guerreros registrados y a aquellos Voluntario APROBADOS por la fundación cuyo horario corresponda con el del evento.
Entradas	Ninguna
Salidas	Ninguna
Proceso	El sistema muestra un formulario en el cual puede observar los voluntarios y guerreros que van a ser invitados al evento, también puede establecer si un guerrero asiste o no al mismo.
Precondiciones	El usuario debe haber llenado los datos generales del evento.
Postcondiciones	Ninguna
Efectos Colaterales	Se ingresa en la base de datos la información de las invitaciones al evento.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.49 Aprobar evento
Requerimiento	
Descripción	Permite aprobar un evento que ha sido registrado con anterioridad
Entradas	Acción del usuario sobre el botón aprobar.
Salidas	Mensaje indicando que se ha aprobado correctamente el evento
Proceso	El sistema muestra un formulario en el cual puede observar todos los eventos que se han registrado en el sistema.
Precondiciones	El usuario debe tener acceso a la pantalla de eventos.
Postcondiciones	Se envían notificaciones a los guerreros y voluntarios que participan en el evento de acuerdo a requerimiento 3.50
Efectos Colaterales	Se ingresa en la base de datos la información de las invitaciones al evento.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.50 Enviar invitaciones a eventos
Requerimiento	
Descripción	Permite enviar invitaciones mediante correo electrónico tanto a los voluntarios como a los guerreros a un evento.
Entradas	Ninguna
Salidas	Ninguna
Proceso	Una vez que se aprueba un evento el

	sistema envía notificaciones de correo electrónico a todos los guerreros que están registrados en el sistema y a aquellos voluntarios ACTIVOS y APROBADOS cuyo horario corresponda al del evento.
Precondiciones	El evento debe estar guardado
Postcondiciones	Los datos generales del evento no pueden ser cambiados Se pueden editar las asistencias de los guerreros al evento
Efectos Colaterales	Cuando el voluntario o el guerrero confirman sus asistencia desde el correo electrónico que recibieron, en la base de datos queda guardada su asistencia al evento.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.51 Registrar resultados de eventos
Requerimiento	
Descripción	Permite registrar los resultados de un evento, para cada voluntario y para cada Guerrero que asistió al mismo.
Entradas	Acción sobre el botón Registrar resultados, Resultado del evento para cada guerrero/voluntario
Salidas	Mensaje indicando que los resultados se han guardado con éxito.
Proceso	El sistema muestra un formulario en el

cual se pueden ingresar los resultados para el evento, para cada voluntario y para cada guerrero que asistió.

Precondiciones El evento debe estar APROBADO

Postcondiciones Ninguna

Efectos Los resultados quedan registrados en la base de datos.

Colaterales

Prioridad Alta

Rol que lo ejecuta Usuario

Id. 3.52 Visualizar Evento

Requerimiento

Descripción Permite visualizar los datos de un evento, una vez que éste ha sido aprobado.

Entradas Acción sobre el botón visualizar

Salidas Mensaje indicando que se han guardado los cambios con éxito.

Proceso El sistema muestra la información del evento, pero únicamente se puede hacer cambios en la parte de invitaciones, en la cual se puede manejar la asistencia de los guerreros al evento.

Precondiciones El evento debe estar APROBADO

Postcondiciones Ninguna

Efectos Las asistencias quedan guardadas en la base de datos.

Colaterales

Prioridad Alta

Rol que lo ejecuta Usuario

ejecuta

Id. 3.53 Buscar Evento

Requerimiento

Descripción Permite buscar los eventos registrados en el sistema.

Entradas Nombre, Descripción, Fecha de Inicio.

Salidas Los eventos que correspondan con los criterios del búsqueda

Proceso El sistema muestra un formulario en el cual se ingresa información para buscar eventos.

Precondiciones El usuario debe haber ingresado al sistema
El usuario debe tener acceso a la pantalla de eventos.

Postcondiciones Ninguna

Efectos Ninguno

Colaterales

Prioridad Alta

Rol que lo ejecuta Usuario

Id. 3.54 Eliminar Evento

Requerimiento

Descripción Permite hacer un borrado lógico de los eventos registrados en el sistema.

Entradas Acción sobre el botón eliminar

Salidas Mensaje indicando que se ha eliminado exitosamente el registro

Proceso El sistema muestra una ventana

	emergente para confirmar la eliminación del evento.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla de eventos.
Postcondiciones	El estado del evento pasa a INACTIVO
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.55 Registrar asignaciones de donaciones
Requerimiento	
Descripción	Permite registrar las asignaciones de cada una de las donaciones que se han registrado en la fundación.
Entradas	Requerimiento 3.56 Requerimiento 3.57
Salidas	Mensaje indicando que se ha guardado exitosamente el registro
Proceso	El sistema muestra un formulario en el cual se debe escoger el ingreso de “Asignaciones Varias” o de “Asignaciones a Guerreros”
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla de asignaciones.
Postcondiciones	La asignación queda registrada en el

sistema

Efectos	Se crean registros en la base de datos
Colaterales	para cada asignación registrada.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id. 3.56 Registrar Asignaciones Varias

Requerimiento

Descripción	Permite registrar aquellas asignaciones que se hacen a FUNDACIONES Aliadas, a Familias y a Otros.
Entradas	Tipo de asignación, Tipo de inventario, Ítem del inventario, Cantidad a asignar
Salidas	Mensaje indicando que se ha guardado exitosamente el registro
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información de la asignación.
Precondiciones	El usuario debe haber ingresado al sistema Acción sobre el botón Crear asignaciones varias en la pantalla principal de asignaciones.
Postcondiciones	La asignación queda registrada en el sistema El stock del ítem del inventario del cual se hizo la asignación debe ser actualizado.
Efectos	Se crean registros en la base de datos

Colaterales	para cada asignación registrada.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.57 Registrar Asignaciones a Guerreros
Requerimiento	
Descripción	Permite registrar aquellas asignaciones que se hacen a Guerreros.
Entradas	Tipo de inventario, Ítem del inventario, Guerrero para asignar, Cantidad a asignar
Salidas	Mensaje indicando que se ha guardado exitosamente el registro
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información de la asignación.
Precondiciones	El usuario debe haber ingresado al sistema
Postcondiciones	La asignación queda registrada en el sistema El stock del ítem del inventario del cual se hizo la asignación debe ser actualizado.
Efectos	Se crean registros en la base de datos
Colaterales	para cada asignación registrada.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.58 Modificar Asignaciones
Requerimiento	
Descripción	Permite editar la información de las asignaciones que se han hecho en el sistema
Entradas	Acción sobre el botón editar
Salidas	Mensaje indicando que se ha guardado exitosamente el registro
Proceso	El sistema muestra un formulario en el cual se procede a editar la información de las asignaciones
Precondiciones	El usuario debe haber ingresado al sistema
Postcondiciones	La asignación se actualiza en el sistema El stock del ítem del inventario del cual se hizo la asignación debe ser actualizado.
Efectos Colaterales	Se actualizan los registros en la base de datos de acuerdo a la información que ha sido modificada.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.59 Generar Reporte Asignaciones por fecha
Requerimiento	
Descripción	Permite generar un reporte que muestra las asignaciones que se han hecho en un rango de fechas.
Entradas	Se muestra o no el campo cédula, Fecha

	inicio, Fecha Fin
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF, con los siguientes campos: Tipo asignación, Fundación aliada, Nombres, Apellidos, Cédula, Tipo inventario, Ítem, Cantidad, Fecha.
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.60 Generar Reporte Guerreros por
Requerimiento	Prioridad
Descripción	Permite generar un reporte que muestra los guerreros registrados en la fundación y los clasifica de acuerdo a la etapa en la que se encuentra su enfermedad
Entradas	Se muestra o no el campo medicamentos, se muestra o no el campo Cédula, Etapa

	de la enfermedad
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Apellidos, Nombres, Cédula, Medicamentos, Etapa enfermedad
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.61 Generar Reporte de Actividades de un voluntario
Requerimiento	
Descripción	Permite generar un reporte que muestra los distintos eventos a los que ha asistido un voluntario
Entradas	Para la búsqueda de voluntarios: Nombres, Apellidos, Actividades desde(Indica desde que fecha se necesitan obtener las actividades del voluntario), Actividades hasta(Indica hasta qué fecha se necesitan obtener las actividades del voluntario)

Se puede escoger si los siguientes campos, aparecen o no en el reporte: Nombre Evento, Descripción Evento, Fecha Inicio, Día, Hora Inicio, Hora Fin, Resultado del evento.

Salidas Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos:
Datos personales del voluntario: Nombres, Apellidos
Actividades del voluntario: Nombre Evento, Descripción Evento, Fecha Inicio, Día, Hora Inicio, Hora Fin, Resultados

Proceso El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.

Precondiciones El usuario debe haber ingresado al sistema
El usuario debe tener acceso a la pantalla para la generación de este reporte.

Postcondiciones Ninguna

Efectos Ninguno

Colaterales

Prioridad Alta

Rol que lo ejecuta Usuario

Id.	3.62 Generar Reporte de Auditoría
Requerimiento	
Descripción	Permite generar un reporte que muestra los cambios que se han hecho sobre la base de datos en un rango de fechas.
Entradas	Fecha de inicio, Fecha de fin.
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Fecha, Usuario, Acción, Tabla, Campo, Valor actual, Valor anterior
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.63 Generar Reporte de Horas de
Requerimiento	voluntariado
Descripción	Permite generar un reporte que muestra las horas de voluntariado que cada

	voluntario ha cumplido en la fundación.
Entradas	Campos que pueden o no aparecer en el reporte: Cédula, Provincia de Residencia, Lugar de Residencia
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Nombres, Apellidos, Cédula, Provincia Residencia, Ciudad residencia, Horas
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos Colaterales	Ninguno
Prioridad	Alta
Rol que lo ejecuta	Usuario
Id.	3.64 Generar Reporte de Donaciones por Fecha
Requerimiento	
Descripción	Permite generar un reporte que muestra las donaciones que ha recibido la fundación dentro de un rango de fecha.
Entradas	Campos que pueden o no aparecer en el reporte: Tipo Donante, Email, Número Identificación, Teléfono, Dirección ,

	Responsable
	Fecha Desde, Fecha Hasta.
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Tipo donante, Nombres, Apellidos, Número identificación, Email Teléfono, Dirección, Responsable, Tipo Inventario, Ítem, Cantidad, Fecha
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos Colaterales	Ninguno
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.65 Generar Reporte de Guerreros por Clasificación
Requerimiento	Clasificación
Descripción	Permite generar un reporte que muestra los guerreros registrados en la fundación y los clasifica de acuerdo a la Clasificación ingresada en los datos generales de los

	mismos.
Entradas	Campos que pueden o no aparecer en el reporte: Cédula, Provincia de Residencia, Ciudad de Residencia, Diagnóstico
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Apellidos, Nombres, Cédula, Provincia, Residencia, Ciudad, Residencia Diagnóstico, Clasificación
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos Colaterales	Ninguno
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.66 Generar Reporte de Movimientos de Inventario
Requerimiento	
Descripción	Permite generar un reporte que muestra los diferentes movimientos de entrada y salida que se han hecho con cada ítem registrado en el inventario en un rango dado de fechas.
Entradas	Tipo de inventario, Fecha desde, Fecha hasta.
Salidas	Archivo conteniendo la información del reporte en EXCEL o PDF con los siguientes campos: Ítem inventario, Stock actual, Movimientos: Fecha, Cantidad Asignado a, Tipo, Ítem
Proceso	El sistema muestra un formulario en el cual se procede a ingresar la información para la generación del reporte.
Precondiciones	El usuario debe haber ingresado al sistema El usuario debe tener acceso a la pantalla para la generación de este reporte.
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.67 Buscar Tipo Catálogo
Requerimiento	
Descripción	Permitir buscar un Tipo Catálogo ingresado
Entradas	Nombre y Acción del usuario sobre el botón buscar
Salidas	Tipos Catálogo correspondientes a los criterios de búsqueda.
Proceso	El sistema busca y muestra los Tipos Catálogo con los criterios de búsqueda ingresados, si no se especifica ningún criterio el sistema encontrará todos los Tipos Catálogo
Precondiciones	Estar logueado en el sistema, tener el rol adecuado para visualizar la página de Tipos de Catálogo
Postcondiciones	Ninguna
Efectos	Ninguno
Colaterales	
Prioridad	Media
Rol que lo ejecuta	Usuario

Id.	3.68 Buscar Catálogo
Requerimiento	
Descripción	Permitir buscar un Catálogo ingresado
Entradas	Nombre y Acción del usuario sobre el botón buscar
Salidas	Catálogos correspondientes a los criterios de búsqueda.
Proceso	El sistema busca y muestra los Catálogo con los criterios de búsqueda ingresados,

si no se especifica ningún criterio el sistema encontrará todos los Catálogo

Precondiciones Estar logueado en el sistema, tener el rol adecuado para visualizar la página de Catálogo

Postcondiciones Ninguna

Efectos Ninguno

Colaterales

Prioridad Media

Rol que lo ejecuta Usuario

Id. 3.69 Modificar Guerrero

Requerimiento

Descripción Permitir modificar un guerrero registrado en el sistema

Entradas Requerimientos: 3.34, 3.38,3.39,3.40,3.41

Salidas Confirmación de registro actualizado exitosamente

Proceso El sistema despliega el formulario de la ficha de guerrero con los datos del mismo en modo de edición.

Pre condiciones El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de ingreso de guerreros, haber ingresado el registro del guerrero previamente.

Post condiciones La información del Guerrero queda actualizada.

Efectos Se actualiza el Guerrero en la base de datos.

Colaterales

Prioridad	Alta
Rol que lo ejecuta	Usuario

Id.	3.70 Modificar Evento
Requerimiento	
Descripción	Permitir modificar un evento registrado en el sistema
Entradas	Nombre, Lugar, Descripción, Fecha Inicio, Fecha Fin, Hora Inicio, Hora Fin, Requerimiento 3.48
Salidas	Confirmación de registro actualizado exitosamente
Proceso	El sistema muestra un formulario para ingresar información sobre el evento que desea realizar.
Precondiciones	El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de modificación de evento. Existir un evento
Postcondiciones	La información del evento queda actualizada.
Efectos Colaterales	Se actualiza el evento en la base de datos.
Prioridad	Alta
Rol que lo ejecuta	Usuario

3.2 MODELO DE CASOS DE USO

3.2.1 HISTORIAL DE REVISIONES

Tabla 8: Historial de revisiones: Modelo de Casos de Uso

Fecha	Versión	Descripción	Autor
04/03/2013	1.0	Especificación de casos de uso	Paúl Jarrín Nelson Jumbo
23/03/2013	2.0	Especificación de casos de uso	Paúl Jarrín Nelson Jumbo

3.2.2 INTRODUCCIÓN

El presente documento permitirá conocer cuál debe ser la funcionalidad del sistema de acuerdo a los requerimientos que se establecieron y de esta manera poder comprobar que dichos requerimientos han sido implementados.

3.2.3 DESCRIPCIÓN GENERAL DE ACTORES

Para el sistema de gestión de donaciones y voluntariado de la FUNDACIÓN JÓVENES CONTRA EL CÁNCER se necesita los siguientes actores.

- Administrador: Es el encargado de la creación de usuario y la asignación de roles a los mismos.
- Usuario: De acuerdo al rol que le ha sido asignado por el administrador tendrá acceso a las diferentes pantallas del sistema

3.2.4 ESPECIFICACIÓN DE CASO DE USO: LOGIN

3.2.4.1 Descripción

Este caso comienza cuando el administrador o el usuario quieren acceder al sistema ingresando nombre del usuario y la contraseña.

3.2.4.2 Meta

Autenticar al usuario contra la base de datos y usando el algoritmo de des encriptación MD5.

3.2.4.3 Actores

Administrador, Operador, base de datos

3.2.4.4 Flujo de Eventos

Flujo Básico

Acceder al sistema

El usuario ingresa el nombre de usuario y contraseña

Flujos Alternativos

No acceso al sistema

El usuario ingresa el nombre de usuario o contraseña incorrecta

3.2.4.5 Precondiciones

Usuario debe estar registrado en el sistema

3.2.4.6 Condición de Éxito

Usuario ingresa al sistema y puede visualizar el menú que su perfil le permita.

3.2.4.7 Condición de Fallo

No puede acceder al sistema.

3.2.5 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO CREAR VOLUNTARIO

3.2.5.1 Descripción

El usuario ingresa la información necesaria de un voluntario con la finalidad de obtener un registro del mismo.

3.2.5.2 Meta

Ingresar un voluntario en el sistema

3.2.5.3 Actores

Usuario

3.2.5.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de nuevos voluntarios, en el cual se debe ingresar los datos del voluntario, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el voluntario no se registra.

3.2.5.5 Requisitos especiales

Ninguno

3.2.5.6 Precondiciones

El usuario debe tener permisos para visualizar la ficha de voluntario y estar logueado en el sistema.

3.2.5.7 Condición de Éxito

Se ingresa un nuevo voluntario con éxito.

3.2.5.8 Condición de Fallo

No se guarda la información.

3.2.6 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR VOLUNTARIO

3.2.6.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón actualizar en Ficha de voluntario.

3.2.6.2 Meta

Actualizar el voluntario.

3.2.6.3 Actores

Usuario

3.2.6.4 Flujo de Eventos

Flujo Básico

El usuario actualiza la información del voluntario seleccionado y guarda.

3.2.6.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de voluntario, estar logueado en el sistema y debe de existir al menos un registro de voluntario.

3.2.6.6 Condición de Éxito

Voluntario actualizado

3.2.6.7 Condición de Fallo

Voluntario no actualizado

3.2.7 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR VOLUNTARIO

3.2.7.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón eliminar en Ficha de voluntario.

3.2.7.2 Meta

Borrado lógico del sistema de un voluntario

3.2.7.3 Actores

Usuario

3.2.7.4 Flujo de Eventos

Flujo Básico

El usuario selecciona un voluntario presionando el botón eliminar y después de confirmar pasa a estar inactivo.

3.2.7.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de voluntario, estar logueado en el sistema y debe de existir al menos un registro de voluntario.

3.2.7.6 Condición de Éxito

Se cambia el estado del voluntario

3.2.7.7 Condición de Fallo

No se cambia el estado del voluntario

3.2.8 ESPECIFICACIÓN DE CASO DE USO: BUSCAR VOLUNTARIO

3.2.8.1 Descripción

El caso de uso comienza cuando el usuario llena los criterios de búsqueda y realiza la acción de presionar el botón buscar.

3.2.8.2 Meta

El sistema busca los voluntarios por los criterios ingresados.

3.2.8.3 Actores

Usuario

3.2.8.4 Flujo de Eventos

Flujo Básico

El usuario ingresa los criterios de búsqueda y presiona en el botón buscar, el sistema encuentra los voluntarios según los criterios ingresados, si no se especifica ningún criterio se encuentran todos los voluntarios.

Flujos Alternativos

El sistema no encuentra voluntarios con los criterios dados.

3.2.8.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de voluntario, estar logueado en el sistema y debe de existir al menos un registro de voluntario.

3.2.8.6 Condición de Éxito

Se encuentran los voluntarios por los criterios de búsqueda dados.

3.2.9 ESPECIFICACIÓN DE CASO DE USO: APROBAR VOLUNTARIO

3.2.9.1 Descripción

El caso de uso comienza cuando el usuario llena realiza la acción de presionar el botón Aprobar.

3.2.9.2 Meta

El sistema cambia el estado del voluntario ha aprobado.

3.2.9.3 Actores

Usuario

3.2.9.4 Flujo de Eventos

Flujo Básico

El usuario presiona el botón aprobar en la lista de voluntarios, el sistema actualiza el estado del voluntario ha aprobado

Flujos Alternativos

El sistema no actualiza el estado del voluntario.

3.2.9.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de voluntario, estar logueado en el sistema y debe de existir al menos un registro de voluntario.

3.2.9.6 Condición de Éxito

El voluntario queda en estado aprobado

3.2.9.7 Condición de Fallo

El voluntario no cambia de estado

3.2.10 ESPECIFICACIÓN DE CASO DE USO: CREAR TIPO INVENTARIO

3.2.10.1 Descripción

El usuario ingresa la información necesaria de un tipo de inventario con la finalidad de obtener un registro del mismo.

3.2.10.2 Meta

Ingresar un tipo inventario en el sistema

3.2.10.3 Actores

Usuario

3.2.10.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de nuevo tipo de inventario, en el cual se debe ingresar los datos del tipo de inventario, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el tipo de inventario no se registra.

3.2.10.5 Requisitos especiales

Ninguno

3.2.10.6 Precondiciones

El usuario debe tener permisos para visualizar la ficha de tipo de inventario y estar logueado en el sistema.

3.2.10.7 Condición de Éxito

Se ingresa un nuevo tipo inventario con éxito.

3.2.10.8 Condición de Fallo

No se guarda la información.

3.2.11 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR TIPO INVENTARIO

3.2.11.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón actualizar en Pantalla de tipo inventario.

3.2.11.2 Meta

Actualizar el tipo de inventario.

3.2.11.3 Actores

Usuario

3.2.11.4 Flujo de Eventos

Flujo Básico

El usuario actualiza la información del tipo de inventario seleccionado y guarda.

3.2.11.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de tipo de inventario, estar logueado en el sistema y debe de existir al menos un registro de tipo de inventario.

3.2.11.6 Condición de Éxito

Tipo inventario actualizado

3.2.11.7 Condición de Fallo

Tipo inventario no actualizado

3.2.12 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR TIPO INVENTARIO

3.2.12.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón eliminar en la pantalla de tipo de inventario.

3.2.12.2 Meta

Borrado lógico del sistema de un tipo inventario

3.2.12.3 Actores

Usuario

Flujo de Eventos

Flujo Básico

El usuario selecciona un tipo de inventario presionando el botón eliminar y después de confirmar pasa a estado inactivo.

3.2.12.4 Precondiciones

El usuario debe tener permisos para visualizar la ficha de tipo inventario, estar logueado en el sistema y debe de existir al menos un registro de tipo de inventario.

3.2.12.5 Condición de Éxito

Se cambia el estado del tipo de inventario

3.2.12.6 Condición de Fallo

No se cambia el estado del tipo de inventario

3.2.13 ESPECIFICACIÓN DE CASO DE USO: CREAR INVENTARIO

3.2.13.1 Descripción

El usuario ingresa la información necesaria de un inventario con la finalidad de obtener un registro del mismo.

3.2.13.2 Meta

Ingresar un inventario en el sistema

3.2.13.3 Actores

Usuario

3.2.13.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de nuevo inventario, en el cual se debe ingresar los datos del inventario, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el inventario no se registra.

3.2.13.5 Requisitos especiales

Ninguno

3.2.13.6 Precondiciones

El usuario debe tener permisos para visualizar la ficha de inventario y estar logueado en el sistema.

3.2.13.7 Condición de Éxito

Se ingresa un nuevo inventario con éxito.

3.2.13.8 Condición de Fallo

No se guarda la información.

3.2.14 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR INVENTARIO

3.2.14.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón actualizar en Pantalla de inventario.

3.2.14.2 Meta

Actualizar el inventario.

3.2.14.3 Actores

Usuario

3.2.14.4 Flujo de Eventos

Flujo Básico

El usuario actualiza la información del inventario seleccionado y guarda.

3.2.14.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de inventario, estar logueado en el sistema y debe de existir al menos un registro de inventario.

3.2.14.6 Condición de Éxito

Inventario actualizado

3.2.14.7 Condición de Fallo

Inventario no actualizado

3.2.15 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR INVENTARIO

3.2.15.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón eliminar en la pantalla de inventario.

3.2.15.2 Meta

Borrado lógico del sistema de un inventario

3.2.15.3 Actores

Usuario

3.2.15.4 Flujo de Eventos

Flujo Básico

El usuario selecciona un inventario presionando el botón eliminar y después de confirmar pasa a estado inactivo.

3.2.15.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de inventario, estar logueado en el sistema y debe de existir al menos un registro de inventario.

3.2.15.6 Condición de Éxito

Se cambia el estado del inventario

3.2.15.7 Condición de Fallo

No se cambia el estado del inventario

3.2.16 ESPECIFICACIÓN DE CASO DE USO: CREAR DONANTE

3.2.16.1 Descripción

El usuario ingresa la información necesaria de un donante con la finalidad de obtener un registro del mismo.

3.2.16.2 Meta

Ingresar un donante en el sistema

3.2.16.3 Actores

Usuario

3.2.16.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de nuevo donante, en el cual se debe ingresar los datos del donante, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el donante no se registra.

3.2.16.5 Requisitos especiales

Ninguno

3.2.16.6 Precondiciones

El usuario debe tener permisos para visualizar la ficha de donante y estar logueado en el sistema.

3.2.16.7 Condición de Éxito

Se ingresa un nuevo donante con éxito.

3.2.16.8 Condición de Fallo

No se guarda la información.

3.2.17 CASO DE USO MODIFICAR DONANTE

3.2.17.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón actualizar en la pantalla de donante.

3.2.17.2 Meta

Actualizar el donante.

3.2.17.3 Actores

Usuario

3.2.17.4 Flujo de Eventos

Flujo Básico

El usuario actualiza la información del donante seleccionado y guarda.

3.2.17.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de donante, estar logueado en el sistema y debe de existir al menos un registro de donante.

3.2.17.6 Condición de Éxito

Donante actualizado

3.2.17.7 Condición de Fallo

Donante no actualizado

3.2.18 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR DONANTE

3.2.18.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón eliminar en la pantalla de donante.

3.2.18.2 Meta

Borrado lógico del sistema de un donante

3.2.18.3 Actores

Usuario

3.2.18.4 Flujo de Eventos

Flujo Básico

El usuario selecciona un donante presionando el botón eliminar y después de confirmar pasa a estado inactivo.

3.2.18.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de donante, estar logueado en el sistema y debe de existir al menos un registro de donante.

3.2.18.6 Condición de Éxito

Se cambia el estado del donante

3.2.18.7 Condición de Fallo

No se cambia el estado del donante

3.2.19 ESPECIFICACIÓN DE CASO DE USO: CREAR DONACIÓN

3.2.19.1 Descripción

El usuario ingresa la información necesaria de una donación con la finalidad de obtener un registro del mismo.

3.2.19.2 Meta

Ingresar una donación en el sistema

3.2.19.3 Actores

Usuario

3.2.19.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de una nueva donación, en el cual se debe ingresar los datos de la donación, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y la donación no se registra.

3.2.19.5 Requisitos especiales

Ninguno

3.2.19.6 Precondiciones

El usuario debe tener permisos para visualizar la ficha de donación y estar logueado en el sistema.

3.2.19.7 Condición de Éxito

Se ingresa una nueva donación con éxito.

3.2.19.8 Condición de Fallo

No se guarda la información.

3.2.20 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR DONACIÓN

3.2.20.1 Descripción

El caso de uso comienza cuando el usuario realiza la acción de presionar el botón actualizar en la pantalla de donación.

3.2.20.2 Meta

Actualizar la donación.

3.2.20.3 Actores

Usuario

3.2.20.4 Flujo de Eventos

Flujo Básico

El usuario actualiza la información de la donación seleccionada y guarda.

3.2.20.5 Precondiciones

El usuario debe tener permisos para visualizar la ficha de la donación, estar logueado en el sistema y debe de existir al menos un registro de donación.

3.2.20.6 Condición de Éxito

Donación actualizada

3.2.20.7 Condición de Fallo

Donación no actualizada

3.2.21 ESPECIFICACIÓN DE CASO DE USO: CREAR ROL

3.2.21.1 Descripción

El administrador ingresa la información necesaria de un rol con la finalidad de obtener un registro del mismo.

3.2.21.2 Meta

Ingresar un rol en el sistema

3.2.21.3 Actores

Administrador

3.2.21.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de un nuevo rol, en el cual se debe ingresar los datos del rol, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el rol no se registra.

3.2.21.5 Requisitos especiales

Ninguno

3.2.21.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de rol y estar logueado en el sistema.

3.2.21.7 Condición de Éxito

Se ingresa un nuevo rol con éxito.

3.2.21.8 Condición de Fallo

No se guarda la información.

3.2.22 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR ROL

3.2.22.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón actualizar en la pantalla de rol.

3.2.22.2 Meta

Actualizar el rol.

3.2.22.3 Actores

Administrador

3.2.22.4 Flujo de Eventos

Flujo Básico

El administrador actualiza la información del rol seleccionado y guarda.

3.2.22.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de rol, estar logueado en el sistema y debe de existir al menos un registro de rol.

3.2.22.6 Condición de Éxito

Rol actualizada

3.2.22.7 Condición de Fallo

Rol no actualizada

3.2.23 ESPECIFICACIÓN DE CASO DE USO: CREAR USUARIO

3.2.23.1 Descripción

El administrador ingresa la información necesaria de un usuario con la finalidad de obtener un registro del mismo.

3.2.23.2 Meta

Ingresar un usuario en el sistema con un rol asignado

3.2.23.3 Actores

Administrador

3.2.23.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de un nuevo usuario, en el cual se debe ingresar los datos del usuario, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarias y el usuario no se registra.

3.2.23.5 Requisitos especiales

Ninguno

3.2.23.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de usuario y estar logueado en el sistema.

3.2.23.7 Condición de Éxito

Se ingresa un nuevo usuario con éxito.

3.2.23.8 Condición de Fallo

No se guarda el usuario.

3.2.24 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR USUARIO

3.2.24.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón actualizar en la pantalla de usuario.

3.2.24.2 Meta

Actualizar el usuario.

3.2.24.3 Actores

Administrador

3.2.24.4 Flujo de Eventos

Flujo Básico

El administrador actualiza la información del usuario seleccionado y guarda.

3.2.24.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de usuario, estar logueado en el sistema y debe de existir al menos un registro de usuario.

3.2.24.6 Condición de Éxito

Usuario actualizado

3.2.24.7 Condición de Fallo

Usuario no actualizado

3.2.25 ESPECIFICACIÓN DE CASO DE USO: CREAR TIPO CATÁLOGO

3.2.25.1 Descripción

El usuario ingresa la información necesaria de un tipo catálogo con la finalidad de obtener un registro del mismo.

3.2.25.2 Meta

Ingresar un tipo catálogo en el sistema

3.2.25.3 Actores

Administrador

3.2.25.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de un nuevo tipo catálogo, en el cual se debe ingresar los datos del tipo catálogo, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el tipo catálogo no se registra.

3.2.25.5 Requisitos especiales

Ninguno

3.2.25.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de tipo catálogo y estar logueado en el sistema.

3.2.25.7 Condición de Éxito

Se ingresa un nuevo tipo catálogo con éxito.

3.2.25.8 Condición de Fallo

No se guarda la información

3.2.26 ESPECIFICACIÓN DE CASO DE USO: BUSCAR TIPO CATÁLOGO

3.2.26.1 Descripción

El caso de uso comienza cuando el administrador llena los criterios de búsqueda y realiza la acción de presionar el botón buscar en la pantalla tipo catálogo.

3.2.26.2 Meta

El sistema busca los tipos de catálogo por los criterios ingresados.

3.2.26.3 Actores

Administrador

3.2.26.4 Flujo de Eventos

Flujo Básico

El administrador ingresa los criterios de búsqueda y presiona en el botón buscar, el sistema encuentra los tipos de catálogo según los criterios ingresados, si no se especifica ningún criterio se encuentran todos los tipos de catálogo.

3.2.26.5 Flujos Alternativos

El sistema no encuentra tipos de catálogo con los criterios dados.

3.2.26.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de tipo de catálogo, estar logueado en el sistema y debe de existir al menos un registro de tipo de catálogo.

3.2.26.7 Condición de Éxito

Se encuentran los tipos de catálogos por los criterios de búsqueda dados.

3.2.27 ESPECIFICACIÓN DE CASO DE USO: CREAR CATÁLOGO

3.2.27.1 Descripción

El administrador ingresa la información necesaria de un catálogo con la finalidad de obtener un registro del mismo.

3.2.27.2 Meta

Ingresar un catálogo en el sistema

3.2.27.3 Actores

Administrador

3.2.27.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de un nuevo catálogo, en el cual se debe ingresar los datos del catálogo, los mismos datos que se validan y se registran en el sistema.

3.2.27.5 Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y el catálogo no se registra.

3.2.27.6 Requisitos especiales

Ninguno

3.2.27.7 Precondiciones

El administrador debe tener permisos para visualizar la ficha de catálogo y estar logueado en el sistema.

3.2.27.8 Condición de Éxito

Se ingresa un nuevo catálogo con éxito.

3.2.27.9 Condición de Fallo

No se guarda la información.

3.2.28 ESPECIFICACIÓN DE CASO DE USO: CASO DE USO MODIFICAR CATÁLOGO

3.2.28.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón actualizar en la pantalla de catálogo.

3.2.28.2 Meta

Actualizar el catálogo.

3.2.28.3 Actores

Administrador

3.2.28.4 Flujo de Eventos

Flujo Básico

El administrador actualiza la información del catálogo seleccionado y guarda.

3.2.28.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de catálogo, estar logueado en el sistema y debe de existir al menos un registro de catálogo.

3.2.28.6 Condición de Éxito

Catálogo actualizado

3.2.28.7 Condición de Fallo

Catálogo no actualizado

3.2.29 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR CATÁLOGO

3.2.29.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón eliminar en la pantalla de catálogo.

3.2.29.2 Meta

Borrado lógico del sistema de un catálogo

3.2.29.3 Actores

Administrador

3.2.29.4 Flujo de Eventos

Flujo Básico

El administrador selecciona un catálogo presionando el botón eliminar y después de confirmar pasa a estado inactivo.

Precondiciones

El administrador debe tener permisos para visualizar la ficha de catálogo, estar logueado en el sistema y debe de existir al menos un registro de catálogo.

3.2.29.5 Condición de Éxito

Se cambia el estado del catálogo

3.2.29.6 Condición de Fallo

No se cambia el estado del catálogo

3.2.30 ESPECIFICACIÓN DE CASO DE USO: BUSCAR CATÁLOGO

3.2.30.1 Descripción

El caso de uso comienza cuando el usuario llena los criterios de búsqueda y realiza la acción de presionar el botón buscar en la pantalla catálogo.

3.2.30.2 Meta

El sistema busca los catálogos por los criterios ingresados.

3.2.30.3 Actores

Administrador

3.2.30.4 Flujo de Eventos

Flujo Básico

El administrador ingresa los criterios de búsqueda y presiona en el botón buscar, el sistema encuentra los catálogos según los criterios ingresados, si no se especifica ningún criterio se encuentran todos los catálogos.

3.2.30.5 Flujos Alternativos

El sistema no encuentra catálogos con los criterios dados.

3.2.30.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de catálogo, estar logueado en el sistema y debe de existir al menos un registro de catálogo.

3.2.30.7 Condición de Éxito

Se encuentran los catálogos por los criterios de búsqueda dados.

3.2.31 ESPECIFICACIÓN DE CASO DE USO: CREAR CIUDAD

3.2.31.1 Descripción

El administrador ingresa la información necesaria de una ciudad con la finalidad de obtener un registro del mismo.

3.2.31.2 Meta

Ingresar una ciudad en el sistema

3.2.31.3 Actores

Administrador

3.2.31.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de una nueva ciudad, en el cual se debe ingresar los datos de la

ciudad, los mismos datos que se validan y se registran en el sistema.

3.2.31.5 Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y la ciudad no se registra.

3.2.31.6 Requisitos especiales

Ninguno

3.2.31.7 Precondiciones

El administrador debe tener permisos para visualizar la ficha de ciudad y estar logueado en el sistema.

3.2.31.8 Condición de Éxito

Se ingresa una nueva ciudad con éxito.

3.2.31.9 Condición de Fallo

No se guarda la información.

3.2.32 CASO DE USO MODIFICAR CIUDAD

3.2.32.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón actualizar en la pantalla de ciudad.

3.2.32.2 Meta

Actualizar la ciudad.

3.2.32.3 Actores

Administrador

3.2.32.4 Flujo de Eventos

Flujo Básico

El administrador actualiza la información de la ciudad seleccionada y guarda.

3.2.32.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de la ciudad, estar logueado en el sistema y debe de existir al menos un registro de ciudad.

3.2.32.6 Condición de Éxito

Ciudad actualizada

3.2.32.7 Condición de Fallo

Ciudad no actualizada

3.2.33 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR CIUDAD

3.2.33.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón eliminar en la pantalla de ciudad.

3.2.33.2 Meta

Borrado lógico del sistema de una ciudad

3.2.33.3 Actores

Administrador

3.2.33.4 Flujo de Eventos

Flujo Básico

El administrador selecciona una ciudad presionando el botón eliminar y después de confirmar pasa a estado inactivo.

3.2.33.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de ciudad, estar logueado en el sistema y debe de existir al menos un registro de ciudad.

3.2.33.6 Condición de Éxito

Se cambia el estado de la ciudad

3.2.33.7 Condición de Fallo

No se cambia el estado de la ciudad

3.2.34 ESPECIFICACIÓN DE CASO DE USO: BUSCAR CIUDAD

3.2.34.1 Descripción

El caso de uso comienza cuando el administrador llena los criterios de búsqueda y realiza la acción de presionar el botón buscar en la pantalla ciudad.

3.2.34.2 Meta

El sistema busca las ciudades por los criterios ingresados.

3.2.34.3 Actores

Administrador

3.2.34.4 Flujo de Eventos

Flujo Básico

El administrador ingresa los criterios de búsqueda y presiona en el botón buscar, el sistema encuentra las ciudades según los criterios ingresados, si no se especifica ningún criterio se encuentran todas las ciudades.

3.2.34.5 Flujos Alternativos

El sistema no encuentra ciudades con los criterios dados.

3.2.34.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de ciudades, estar logueado en el sistema y debe de existir al menos un registro de ciudad.

3.2.34.7 Condición de Éxito

Se encuentran las ciudades por los criterios de búsqueda dados.

3.2.35 ESPECIFICACIÓN DE CASO DE USO: CREAR PROVINCIA

3.2.35.1 Descripción

El administrador ingresa la información necesaria de una provincia con la finalidad de obtener un registro del mismo.

3.2.35.2 Meta

Ingresar una provincia en el sistema

3.2.35.3 Actores

Administrador

3.2.35.4 Flujo de Eventos

Flujo Básico

El sistema despliega un formulario para el registro de una nueva provincia, en el cual se debe ingresar los datos de la provincia, los mismos datos que se validan y se registran en el sistema.

Flujos Alternativos

Los datos ingresados no pasan las validaciones necesarios y la provincia no se registra.

3.2.35.5 Requisitos especiales

Ninguno

3.2.35.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de la provincia y estar logueado en el sistema.

3.2.35.7 Condición de Éxito

Se ingresa una nueva provincia con éxito.

3.2.35.8 Condición de Fallo

No se guarda la información

3.2.36 ESPECIFICACIÓN DE CASO DE USO: MODIFICAR PROVINCIA

3.2.36.1 Descripción

El caso de uso comienza cuando el administrador realiza la acción de presionar el botón actualizar en la pantalla de provincia.

3.2.36.2 Meta

Actualizar la provincia.

3.2.36.3 Actores

Administrador

3.2.36.4 Flujo de Eventos

Flujo Básico

El administrador actualiza la información de la provincia seleccionada y guarda.

3.2.36.5 Precondiciones

El administrador debe tener permisos para visualizar la ficha de la provincia, estar logueado en el sistema y debe de existir al menos un registro de provincia.

3.2.36.6 Condición de Éxito

Provincia actualizada

3.2.36.7 Condición de Fallo

Provincia no actualizada

3.2.37 ESPECIFICACIÓN DE CASO DE USO: BUSCAR PROVINCIA

3.2.37.1 Descripción

El caso de uso comienza cuando el administrador llena los criterios de búsqueda y realiza la acción de presionar el botón buscar en la pantalla provincia.

3.2.37.2 Meta

El sistema busca las provincias por los criterios ingresados.

3.2.37.3 Actores

Administrador

3.2.37.4 Flujo de Eventos

Flujo Básico

El administrador ingresa los criterios de búsqueda y presiona en el botón buscar, el sistema encuentra las provincias según los criterios ingresados, si no se especifica ningún criterio se encuentran todas las provincias.

3.2.37.5 Flujos Alternativos

El sistema no encuentra ciudades con los criterios dados.

3.2.37.6 Precondiciones

El administrador debe tener permisos para visualizar la ficha de provincias, estar logueado en el sistema y debe de existir al menos un registro de provincia.

3.2.37.7 Condición de Éxito

Se encuentran las provincias por los criterios de búsqueda dados.

3.2.38 ESPECIFICACIÓN DE CASO DE USO: CREAR GUERRERO

3.2.38.1 Descripción

El usuario ingresa la información de los guerreros al sistema.

3.2.38.2 Meta

Ingresar guerrero en el sistema

3.2.38.3 Actores

Usuario

3.2.38.4 Flujo de Eventos

Flujo Básico

El sistema despliega el formulario para registro de nuevos guerreros, en el cual se registran los datos del mismo.

Flujos Alternativos

- No se ingresaron datos obligatorios
- La cédula ingresada no es correcta
- La cédula ingresada ya fue registrada para otro guerrero
- No se ingresó la información del carnet del CONADIS
- No se ingresaron al menos 1 deseo y 1 lugar
- No se ingresó al menos un Contacto de tipo FAMILIAR
- No se ingresaron datos académicos
- No se ingresó la etapa de la enfermedad del guerrero
- No se ingresó la razón por la cual el guerrero abandonó su tratamiento
- Teléfono celular ingresado no tiene 10 dígitos
- Teléfono convencional no tiene 9 dígitos

3.2.38.5 Requisitos especiales

Ninguno

3.2.38.6 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de ingreso de guerreros.

3.2.38.7 Condición de Éxito

Se ingresa un nuevo guerrero con éxito.

3.2.38.8 Condición de Fallo

El registro no fue guardado

3.2.39 ESPECIFICACIÓN DE CASO DE USO: MODIFICAR GUERRERO

3.2.39.1 Descripción

El caso de uso comienza cuando el usuario selecciona un guerrero para actualizar.

3.2.39.2 Meta

Actualizar el guerrero seleccionado.

3.2.39.3 Actores

Usuario

3.2.39.4 Flujo de Eventos

Flujo Básico

El usuario actualiza y guarda la información del guerrero.

3.2.39.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de guerreros.

3.2.39.6 Condición de Éxito

La información del guerrero es actualizada

3.2.39.7 Condición de Fallo

No se actualiza la información del guerrero en el sistema

3.2.40 ESPECIFICACIÓN DE CASO DE USO: BUSCAR GUERRERO

3.2.40.1 Descripción

El caso de uso comienza cuando el usuario selecciona la acción de búsqueda de guerreros.

3.2.40.2 Meta

Encontrar un guerrero de acuerdo a: Cédula, Nombres, Apellidos

3.2.40.3 Actores

Usuario

3.2.40.4 Flujo de Eventos

Flujo Básico

- El usuario ingresa los filtros de búsqueda.
- El usuario encuentra la información deseada
- Si no se especifican criterios de búsqueda el sistema encuentra todos los guerreros registrados

3.2.40.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de guerreros.

3.2.40.6 Condición de Éxito

El usuario encuentra la información deseada

3.2.40.7 Condición de Fallo

Ninguna

3.2.41 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR GUERRERO

3.2.41.1 Descripción

El caso de uso comienza cuando el usuario selecciona un guerrero para eliminar

3.2.41.2 Meta

Borrado lógico de un guerrero

3.2.41.3 Actores

Usuario

3.2.41.4 Flujo de Eventos

Flujo Básico

El usuario selecciona un guerrero y escoge la opción eliminar.

3.2.41.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de guerreros.

3.2.41.6 Condición de Éxito

El guerrero ha cambiado su estado a INACTIVO

3.2.41.7 Condición de Fallo

El guerrero no cambia de estado

3.2.42 ESPECIFICACIÓN DE CASO DE USO: CREAR EVENTO

3.2.42.1 Descripción

El caso de uso comienza cuando el usuario requiere registrar un nuevo evento para la fundación

3.2.42.2 Meta

Registrar un evento en el sistema.

3.2.42.3 Actores

Usuario

3.2.42.4 Flujo de Eventos

Flujo Básico

El usuario escoge la opción para crear un nuevo evento e ingresa la información correspondiente y lo guarda.

Flujo Alternativo

No se ingresaron datos obligatorios del evento

3.2.42.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

3.2.42.6 Condición de Éxito

El evento queda registrado en el sistema

3.2.42.7 Condición de Fallo

El evento no se registra en el sistema

3.2.43 ESPECIFICACIÓN DE CASO DE USO: MANEJAR INVITACIONES A EVENTOS

3.2.43.1 Descripción

Este caso de uso empieza cuando el usuario desea ver las invitaciones que serán enviadas para el evento.

3.2.43.2 Meta

Controlar la asistencia de voluntarios y guerreros a un evento

3.2.43.3 Actores

Usuario

3.2.43.4 Flujo de Eventos

Flujo Básico

El sistema muestra al usuario un formulario con los voluntarios y guerreros que van a ser invitados una vez que este sea aprobado.

El usuario puede marcar a un guerrero como asistente o inasistente al evento.

3.2.43.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

3.2.43.6 Condición de Éxito

Se tiene información de la asistencia de guerreros y voluntarios al evento

3.2.44 ESPECIFICACIÓN DE CASO DE USO: APROBAR EVENTO

3.2.44.1 Descripción

El caso de uso comienza cuando el usuario requiere aprobar un evento ya registrado anteriormente.

3.2.44.2 Meta

Aprobar un evento

3.2.44.3 Actores

Usuario

3.2.44.4 Flujo de Eventos

Flujo Básico

El usuario escoge la opción aprobar para un evento en particular.

3.2.44.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

3.2.44.6 Condición de Éxito

El evento cambia su estado ha APROBADO

3.2.44.7 Condición de Fallo

Ninguna

3.2.45 ESPECIFICACIÓN DE CASO DE USO: ENVIAR INVITACIONES A EVENTOS

3.2.45.1 Descripción

El caso de uso comienza cuando el usuario aprueba un evento

3.2.45.2 Meta

Enviar invitaciones a guerreros y voluntarios sobre un evento aprobado

3.2.45.3 Actores

Usuario

3.2.45.4 Flujo de Eventos

Flujo Básico

Cuando el usuario escoge la opción aprobar para un evento en particular, se envían invitaciones al correo electrónico de todos los guerreros registrados y a todos los voluntarios cuyo horario corresponda con el del evento.

3.2.45.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

El evento debe ser aprobado

3.2.45.6 Condición de Éxito

Todos los guerreros y voluntarios reciben las invitaciones en sus correos

3.2.45.7 Condición de Fallo

Ninguna

3.2.46 ESPECIFICACIÓN DE CASO DE USO: REGISTRAR RESULTADOS DE EVENTOS

3.2.46.1 Descripción

El caso de uso comienza cuando el usuario selecciona la opción de registro de resultados para un evento aprobado.

3.2.46.2 Meta

Registrar el resultado de cada voluntario y guerrero asistente al evento.

3.2.46.3 Actores

Usuario

3.2.46.4 Flujo de Eventos

Flujo Básico

El usuario escoge la opción registrar resultados para un evento aprobado

El usuario puede registrar el resultado para cada voluntario y guerrero que asistió al evento

3.2.46.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

El evento debe estar aprobado

3.2.46.6 Condición de Éxito

Todos los guerreros y voluntarios asistentes al evento tienen registrado un resultado

3.2.46.7 Condición de Fallo

Ninguna

3.2.47 ESPECIFICACIÓN DE CASO DE USO: BUSCAR EVENTO

3.2.47.1 Descripción

El caso de uso comienza cuando el usuario selecciona la acción de búsqueda de eventos.

3.2.47.2 Meta

Encontrar un evento de acuerdo a: Nombre, Descripción, Fecha de inicio

3.2.47.3 Actores

Usuario

3.2.47.4 Flujo de Eventos

Flujo Básico

- El usuario ingresa los filtros de búsqueda.
- El usuario encuentra la información deseada
- Si no se especifican criterios de búsqueda el sistema encuentra todos los eventos registrados

3.2.47.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

3.2.47.6 Condición de Éxito

El usuario encuentra la información deseada

3.2.47.7 Condición de Fallo

Ninguna

3.2.48 ESPECIFICACIÓN DE CASO DE USO: VISUALIZAR EVENTO

3.2.48.1 Descripción

El caso de uso comienza cuando el usuario selecciona la acción de visualización de eventos.

3.2.48.2 Meta

Visualizar los datos del evento

3.2.48.3 Actores

Usuario

3.2.48.4 Flujo de Eventos

Flujo Básico

El sistema muestra la información del evento solo como lectura únicamente se puede hacer cambios en la parte de invitaciones, en la cual se puede manejar la asistencia de los guerreros al evento.

3.2.48.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

El evento debe estar aprobado

3.2.48.6 Condición de Éxito

El usuario encuentra la información deseada

3.2.48.7 Condición de Fallo

Ninguna

3.2.49 ESPECIFICACIÓN DE CASO DE USO: ELIMINAR EVENTO

3.2.49.1 Descripción

El caso de uso comienza cuando el usuario selecciona un evento para eliminar

3.2.49.2 Meta

Borrado lógico de un evento

3.2.49.3 Actores

Usuario

3.2.49.4 Flujo de Eventos

Flujo Básico

El usuario selecciona un evento y escoge la opción eliminar.

3.2.49.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de eventos.

3.2.49.6 Condición de Éxito

El evento ha cambiado su estado a INACTIVO

3.2.49.7 Condición de Fallo

El evento no cambia de estado

3.2.50 ESPECIFICACIÓN DE CASO DE USO: REGISTRAR ASIGNACIONES DE DONACIONES

3.2.50.1 Descripción

Permite registrar las asignaciones de cada una de las donaciones que se han registrado en la fundación.

3.2.50.2 Meta

Registrar las asignaciones a FUNDACIONES aliadas, familias, guerreros y otros

3.2.50.3 Actores

Usuario

3.2.50.4 Flujo de Eventos

Flujo Básico

El usuario selecciona que tipo de asignación desea realizar

En caso de Asignaciones varias el usuario llena los datos: Tipo de asignación, Tipo de inventario, Ítem del inventario, Cantidad a asignar

En caso de Asignaciones a Guerreros el usuario llena los datos: Tipo de inventario, Ítem del inventario, Guerrero para asignar, Cantidad a asignar

El usuario guarda la asignación.

3.2.50.5 Requisitos especiales

El ítem de inventario seleccionado debe contar con stock suficiente.

3.2.50.6 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de asignaciones.

3.2.50.7 Condición de Éxito

La asignación es realizada

3.2.50.8 Condición de Fallo

No se han ingresado los datos obligatorios

La cantidad a asignar supera el stock disponible de un ítem del inventario.

3.2.51 ESPECIFICACIÓN DE CASO DE USO: MODIFICAR ASIGNACIONES

3.2.51.1 Descripción

Permite editar la información de cada una de las donaciones que se han registrado en la fundación.

3.2.51.2 Meta

Editar la información de una asignación.

3.2.51.3 Actores

Usuario

3.2.51.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona una asignación para editar
- El usuario Cambia la información deseada
- El usuario Guarda los cambios

3.2.51.5 Requisitos especiales

El ítem de inventario seleccionado debe contar con stock suficiente.

3.2.51.6 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página principal de asignaciones.

3.2.51.7 Condición de Éxito

La asignación es realizada

3.2.51.8 Condición de Fallo

No se han ingresado los datos obligatorios

La cantidad a asignar supera el stock disponible de un ítem del inventario.

3.2.52 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE ASIGNACIONES POR FECHA

3.2.52.1 Descripción

Permite generar un reporte que muestra las asignaciones que se han hecho en un rango de fechas.

3.2.52.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.52.3 Actores

Usuario

3.2.52.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.
- El usuario selecciona el rango de fechas
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.52.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de asignaciones por fecha.

3.2.52.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.52.7 Condición de Fallo

Ninguna

3.2.53 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE GUERREROS POR PRIORIDAD

3.2.53.1 Descripción

Permite generar un reporte que muestra los guerreros registrados en la fundación y los clasifica de acuerdo a la etapa en la que se encuentra su enfermedad

3.2.53.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.53.3 Actores

Usuario

3.2.53.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.
- El usuario selecciona la etapa de la enfermedad de los guerreros que deben aparecer en el reporte.
- En caso de no seleccionar una etapa, el reporte contendrá a todos los guerreros registrados en el sistema.
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.53.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de guerreros por prioridad

3.2.53.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.53.7 Condición de Fallo

Ninguna

3.2.54 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE DE ACTIVIDADES DE UN VOLUNTARIO

3.2.54.1 Descripción

Permite generar un reporte que muestra los distintos eventos a los que ha asistido un voluntario

3.2.54.2 Meta

Obtener un reporte en formato EXCEL o PDF

3.2.54.3 Actores

Usuario

3.2.54.4 Flujo de Eventos

Flujo Básico

- El usuario busca el voluntario del cual necesita obtener el reporte
- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte
- El usuario selecciona el reporte PDF o el EXCEL para un voluntario en particular
- El reporte es generado

3.2.54.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de Actividades de un voluntario

3.2.54.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.54.7 Condición de Fallo

Ninguna

3.2.55 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE DE AUDITORÍA

3.2.55.1 Descripción

Permite generar un reporte que muestra los cambios que se han hecho sobre la base de datos en un rango de fechas.

3.2.55.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.55.3 Actores

Administrador

3.2.55.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona el rango de fechas
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.55.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de auditoría

3.2.55.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.55.7 Condición de Fallo

Ninguna

3.2.56 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE DE HORAS DE VOLUNTARIADO

3.2.56.1 Descripción

Permite generar un reporte que muestra las horas de voluntariado que cada voluntario ha cumplido en la fundación.

3.2.56.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.56.3 Actores

Usuario

3.2.56.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.56.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de horas de voluntariado.

3.2.56.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.56.7 Condición de Fallo

Ninguna

3.2.57 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE DE DONACIONES POR FECHA

3.2.57.1 Descripción

Permite generar un reporte que muestra las donaciones que ha recibido la fundación dentro de un rango de fecha.

3.2.57.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.57.3 Actores

Usuario

3.2.57.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.

- El usuario selecciona el rango de fechas
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.57.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de guerreros por prioridad

3.2.57.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.57.7 Condición de Fallo

Ninguna

3.2.58 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE GUERREROS POR CLASIFICACIÓN

3.2.58.1 Descripción

Permite generar un reporte que muestra los guerreros registrados en la fundación y los clasifica de acuerdo a la Clasificación ingresada en los datos generales de los mismos.

3.2.58.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.58.3 Actores

Usuario

3.2.58.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.
- El usuario selecciona la clasificación de los guerreros que deben aparecer en el reporte.
- En caso de no seleccionar una clasificación, el reporte contendrá a todos los guerreros registrados en el sistema.

- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.58.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de guerreros por clasificación

3.2.58.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.58.7 Condición de Fallo

Ninguna

3.2.59 ESPECIFICACIÓN DE CASO DE USO: GENERAR REPORTE DE MOVIMIENTOS DE INVENTARIO

3.2.59.1 Descripción

Permite generar un reporte que muestra los diferentes movimientos de entrada y salida que se han hecho con cada ítem registrado en el inventario en un rango dado de fechas.

3.2.59.2 Meta

Obtener un reporte en formato EXCEL o PDF.

3.2.59.3 Actores

Usuario

3.2.59.4 Flujo de Eventos

Flujo Básico

- El usuario selecciona los campos opcionales que desea que aparezcan en el reporte.
- El usuario selecciona el rango de fechas para la generación del reporte
- El usuario selecciona el reporte PDF o el EXCEL
- El reporte es generado

3.2.59.5 Precondiciones

El usuario debe haber ingresado al sistema y tener el rol adecuado para visualizar la página de reporte de movimientos de inventario

3.2.59.6 Condición de Éxito

El usuario puede descargar el reporte generado

3.2.59.7 Condición de Fallo

Ninguna

3.3 DIAGRAMA DE CASOS DE USO

3.3.1 MÓDULO DE SEGURIDAD Y ACCESO

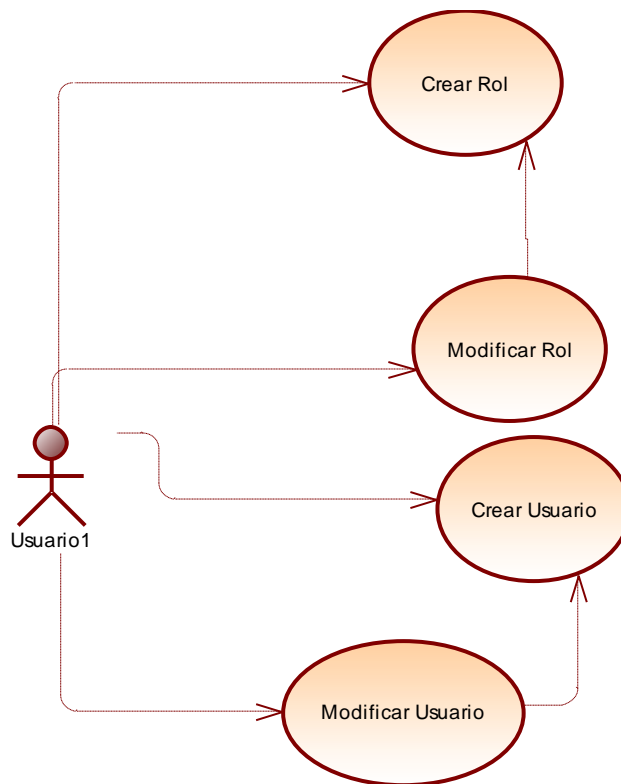


Figura 14: Diagrama de casos de uso del Módulo de seguridad y acceso

3.3.2 MÓDULO DE GESTIÓN DE VOLUNTARIOS Y DONANTES

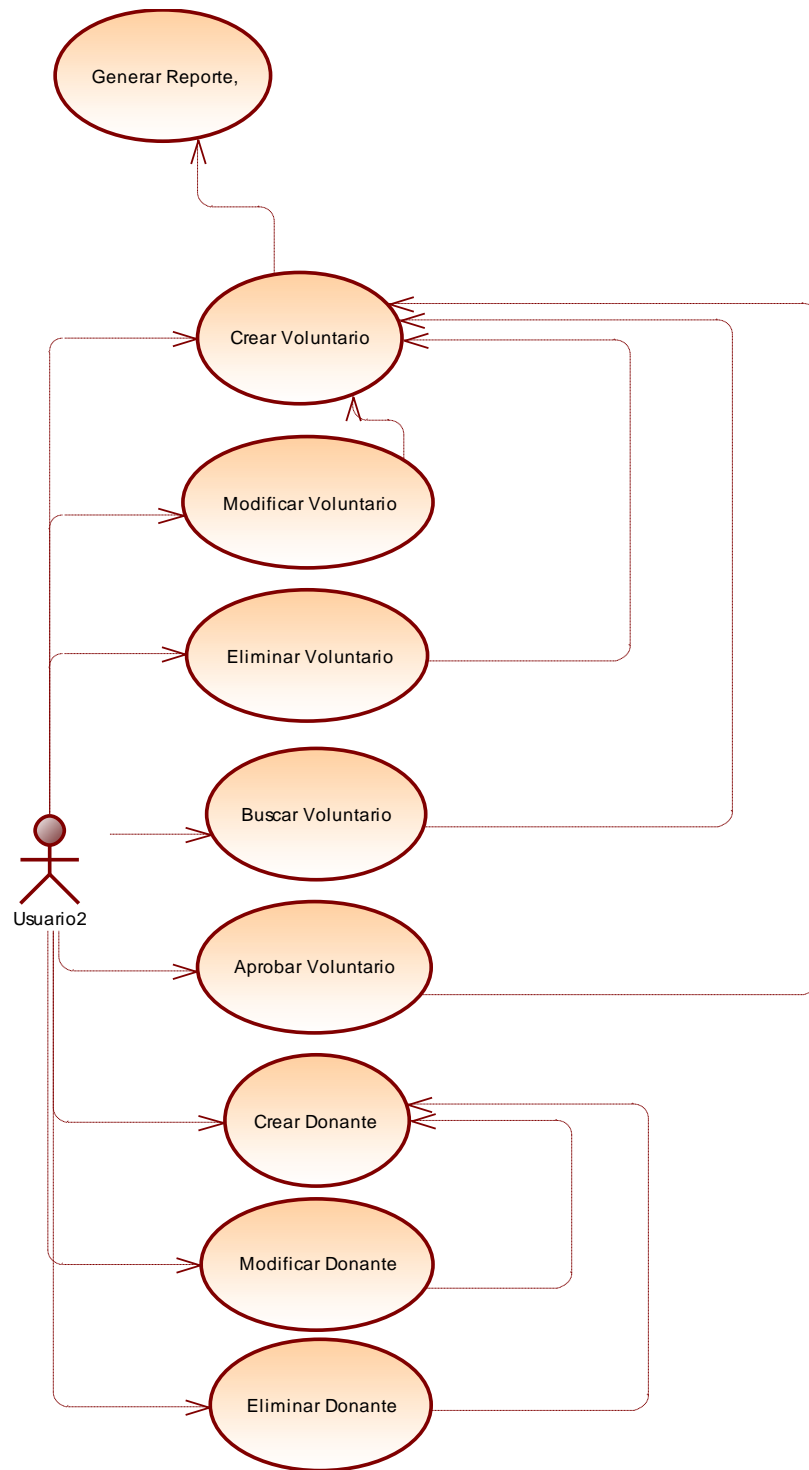


Figura 15: Diagrama de casos de uso del Módulo de Gestión de voluntarios y donantes

3.3.3 MÓDULO GESTIÓN DE DONACIONES

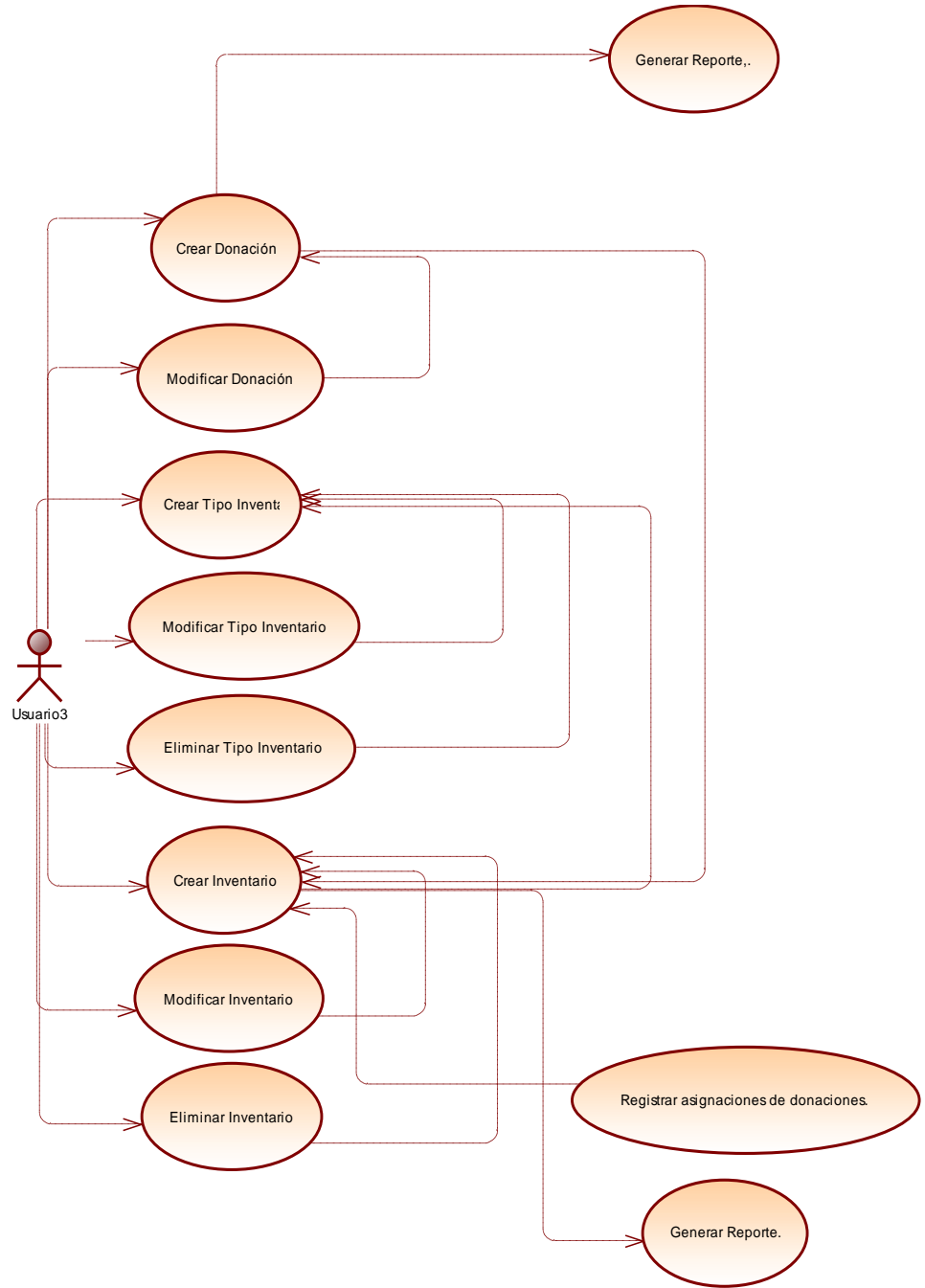


Figura 16:Diagrama de casos de uso del Módulo de Gestión de Donaciones

3.3.4 MÓDULO PARA GESTIÓN DE GUERREROS

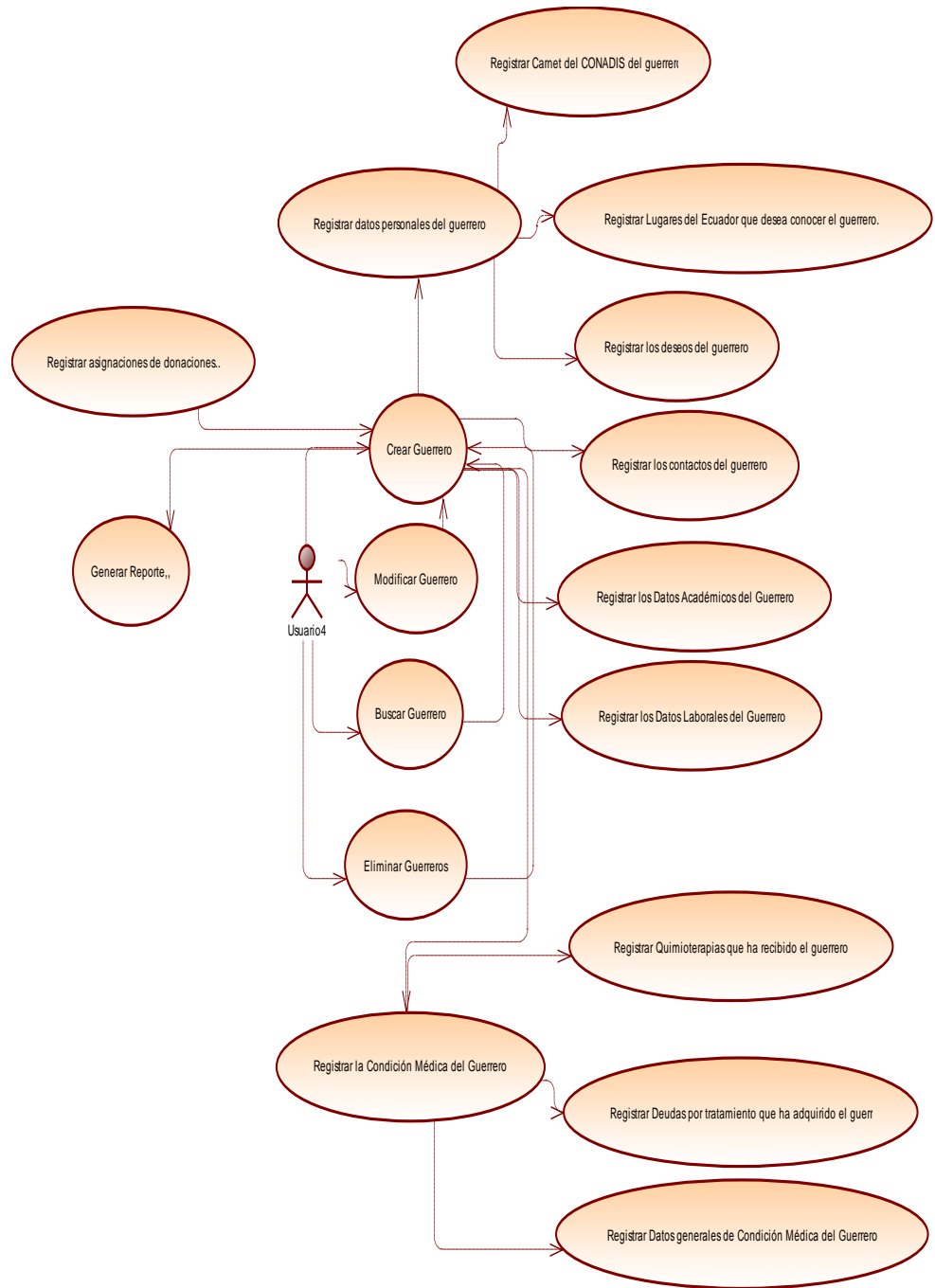


Figura 17: Diagrama de casos de uso del Módulo de Gestión de Guerreros

3.3.5 MÓDULO PARA LA ADMINISTRACIÓN Y APROBACIÓN DE EVENTOS

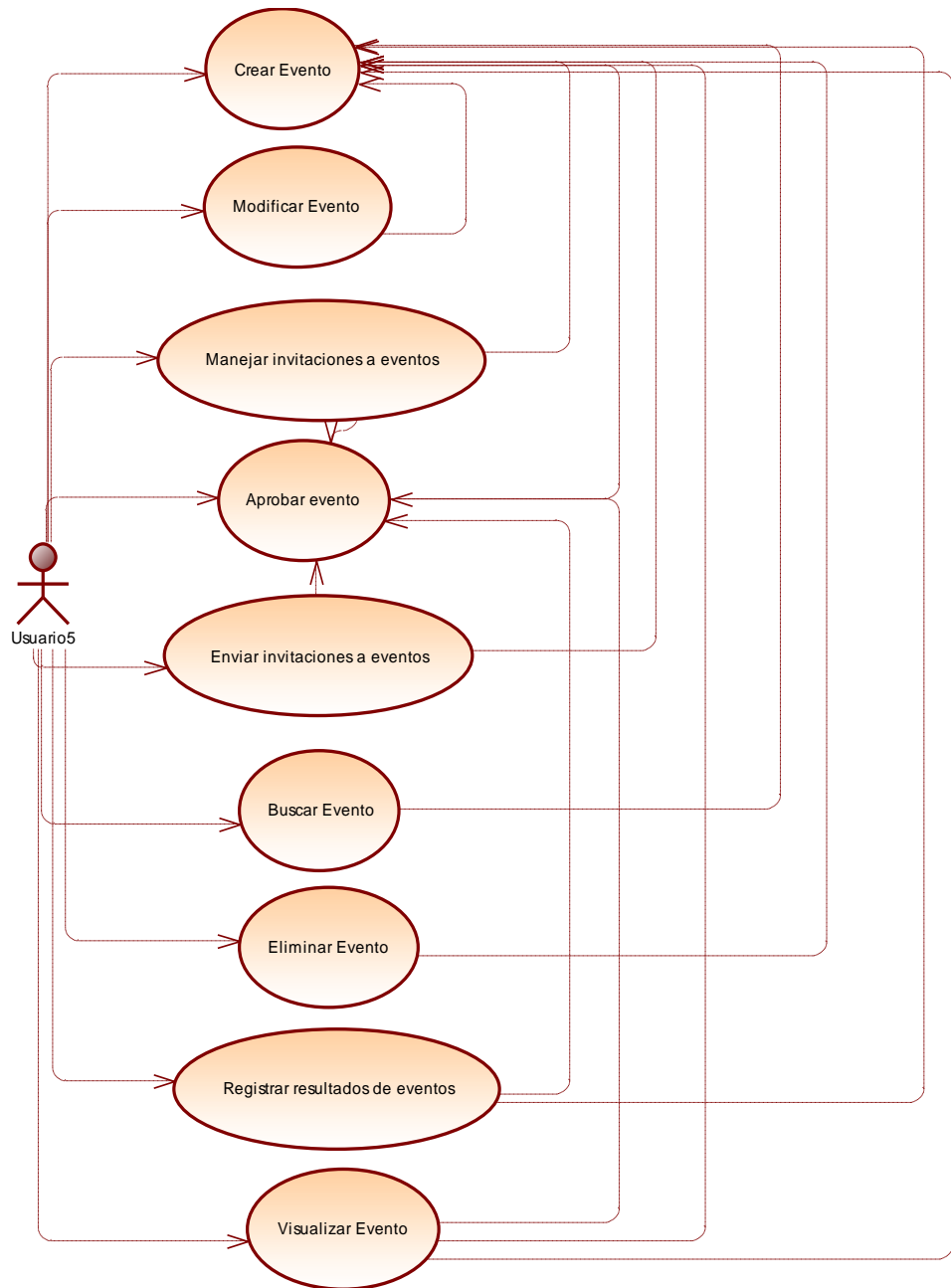


Figura 18: Diagrama de casos de uso del Módulo para la administración y aprobación de eventos

3.3.6 MÓDULO PARA LA ASIGNACIÓN DE DONACIONES

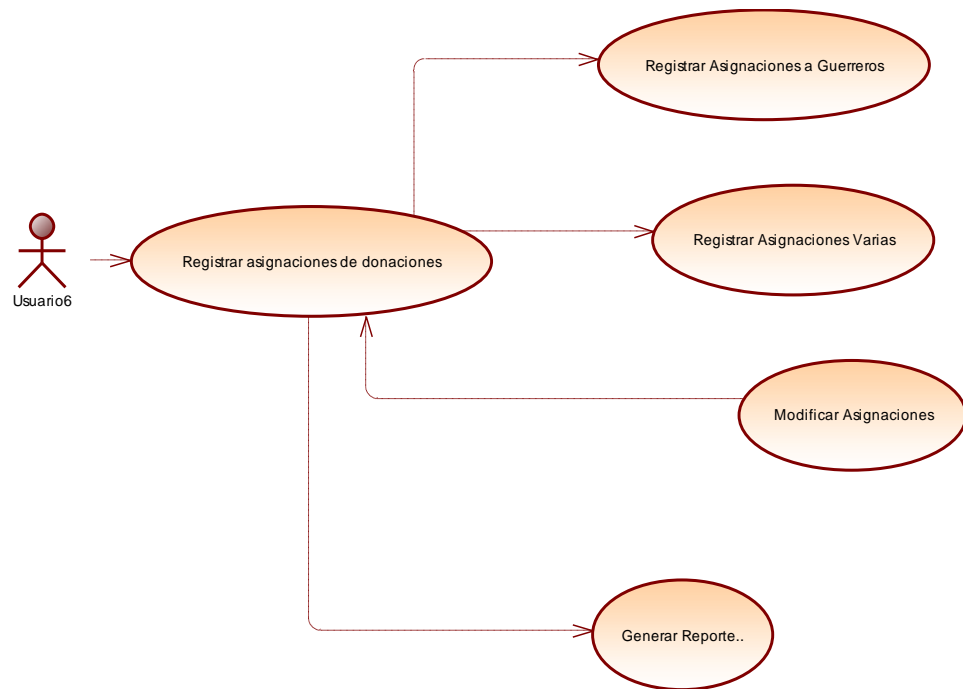


Figura 19: Diagrama de casos de uso del Módulo para la Asignación de Donaciones

3.3.7 MÓDULO DE REPORTES DINÁMICOS

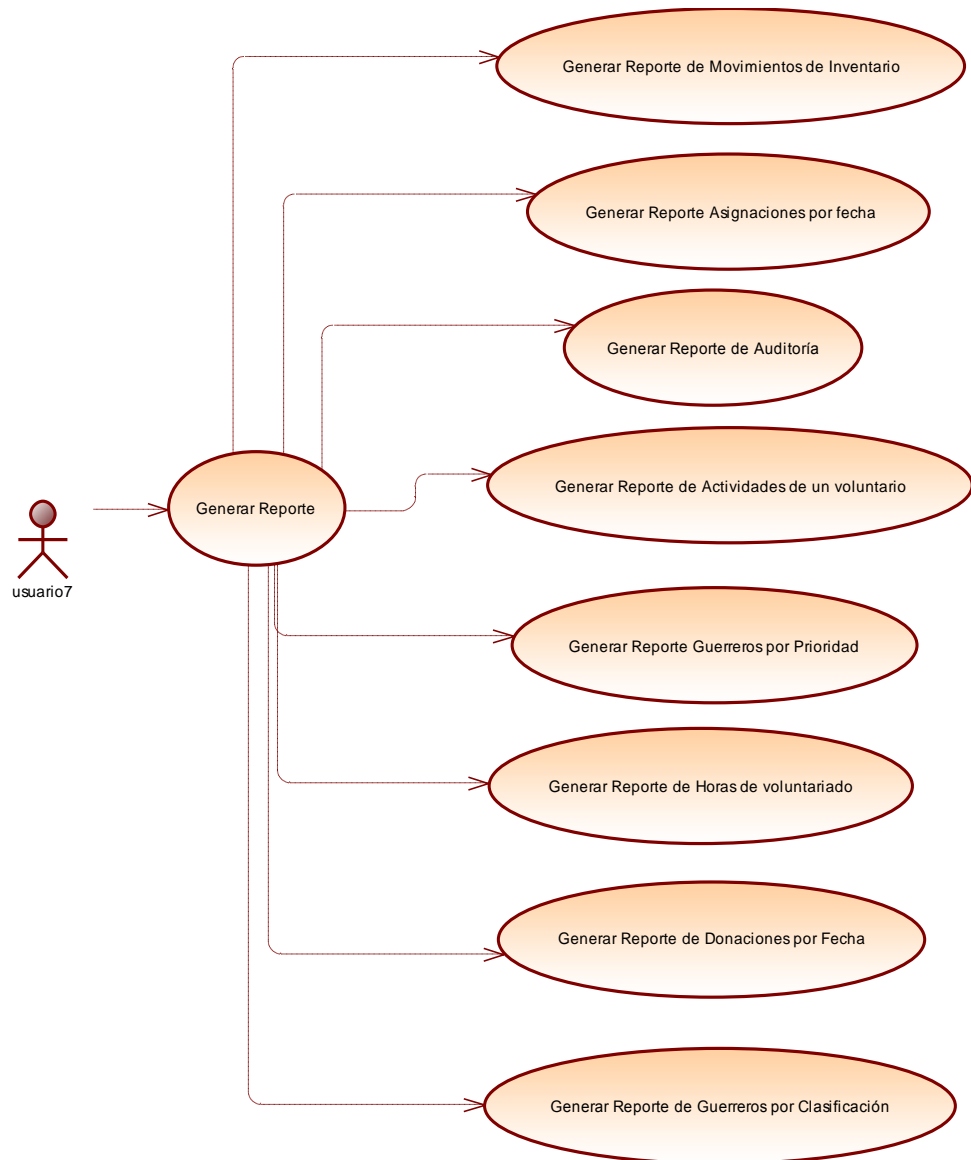


Figura 20: Diagrama de casos de uso del Módulo de Reportes Dinámicos

3.3.8 MÓDULO DE ADMINISTRACIÓN DE CATÁLOGOS

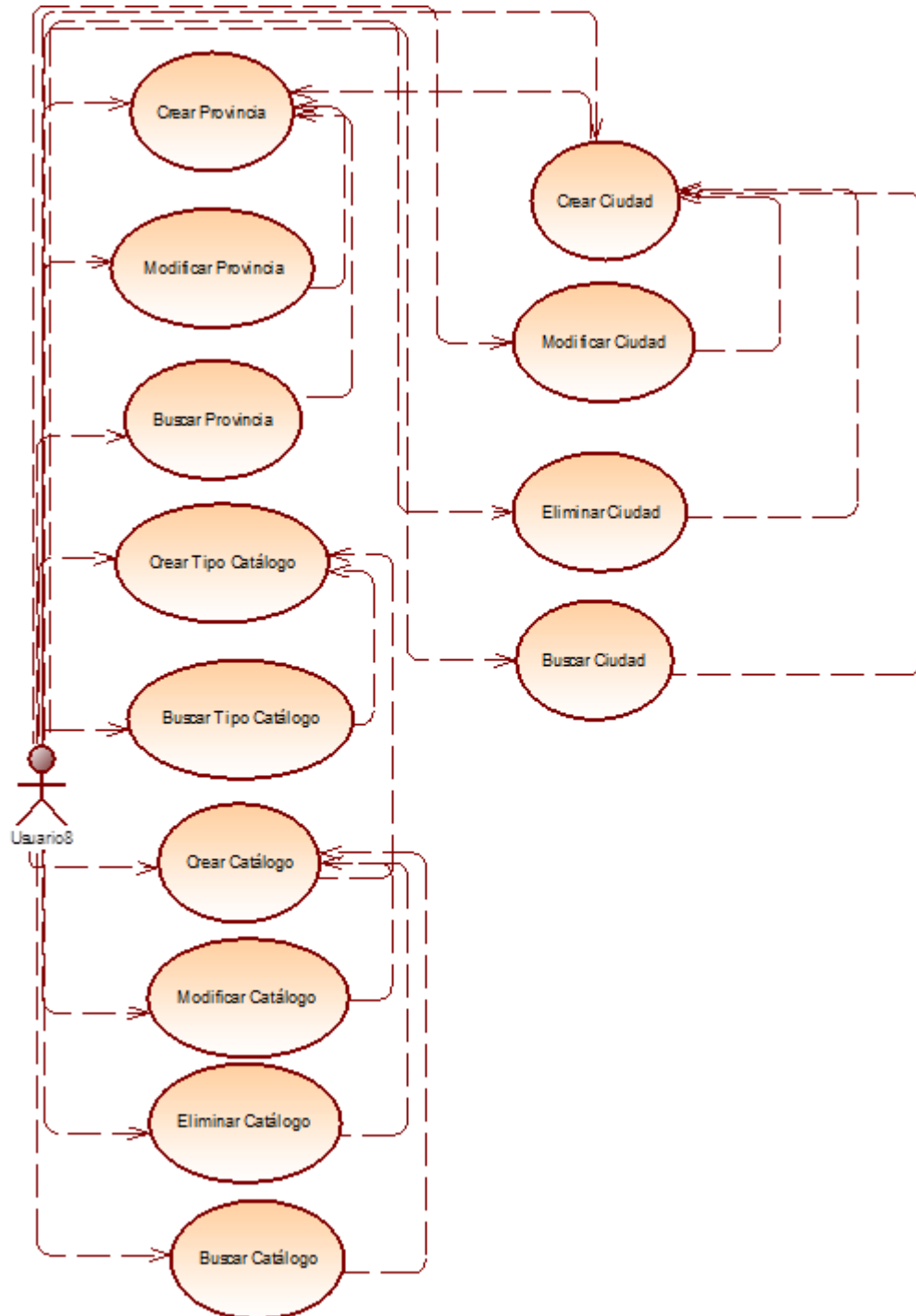


Figura 21 Diagrama de casos de uso del Módulo de Administración de Catálogos

3.4 DIAGRAMA DE DESPLIEGUE

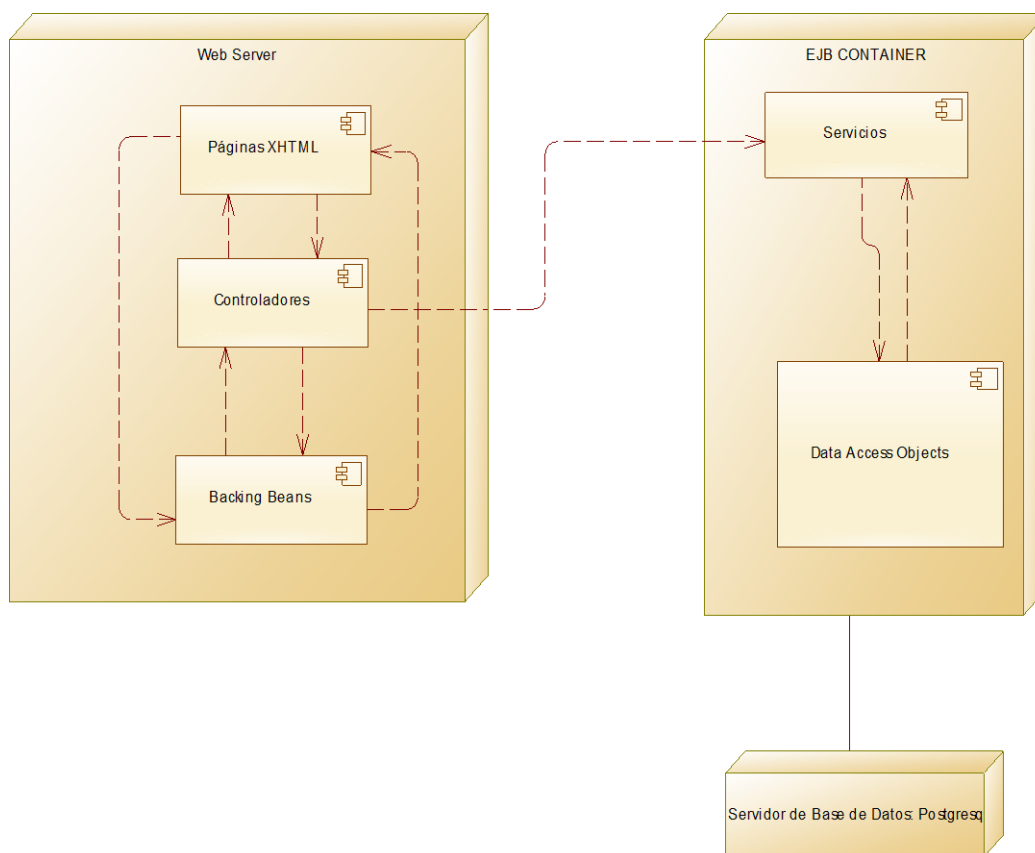


Figura 22: Diagrama de Despliegue

3.5 DIAGRAMA ENTIDAD RELACIÓN

Ver el Anexo A

3.6 DIAGRAMA DE CLASES DEL MODELO DE PERSISTENCIA

Ver el Anexo B

CAPÍTULO 4

PRUEBAS

4.1 PLAN DE PRUEBAS

4.1.1 HISTORIAL DE VERSIONES

Tabla 9: Historial de Versiones de Pruebas

Versión	Encargado	Descripción
1.0.0	Paúl Jarrín Nelson Jumbo	Creación de plantilla

4.1.2 CARTA DE APROBACIÓN

Ver el Anexo G

4.1.3 INTRODUCCIÓN

Se pretende describir las diferentes pruebas que se llevarán a cabo sobre los artefactos del software de “DONACIONES Y VOLUNTARIADO PARA LA FUNDACIÓN JÓVENES CONTRA EL CÁNCER” - JOVCAN generado como propósito de tesis, cuyos resultados serán entregados a la FUNDACIÓN JÓVENES CONTRA EL CÁNCER, por parte del cliente, a fin de validar su cumplimiento.

4.1.4 PROPÓSITO

Este documento pretende establecer un estándar de pruebas que se va a aplicar al proyecto JOVCAN. Este está dirigido a la Directora y Codirectora del proyecto de tesis, a fin de que se conozca la exactitud y el nivel de cumplimiento de los requerimientos planteados a inicio de este proyecto.

4.1.5 ALCANCE

Se validarán y verificarán los procesos críticos de cada uno de los módulos establecidos en el alcance del proyecto.

4.1.6 DEFINICIÓN DE TÉRMINOS Y ACRÓNIMOS

Usuario: Persona encargada de manejar el sistema

Administrador: Persona, natural o jurídica encargada de la gestión de perfiles de usuario

4.1.7 VISIÓN GENERAL

En las siguientes secciones se describirán las herramientas, documentos y plantillas que se utilizarán en la aplicación de pruebas para el sistema JOVCAN.

4.1.8 ÍTEMS A PROBAR

4.1.8.1 Funciones

Las funciones que se van a probar del sistema son las siguientes:

- Crear Rol
- Crear Usuario
- Crear Voluntario
- Crear Donante
- Crear Donación
- Crear Guerrero
- Crear Evento
- Aprobar Evento
- Registrar asignaciones de donaciones
- Generar el Reporte de Guerreros por prioridad
- Crear catálogo

4.1.8.2 Módulos

Los módulos que se van a probar son los siguientes:

- Módulo de Seguridad y Acceso
- Módulo de Gestión de voluntarios y donantes
- Módulo Gestión de donaciones
- Módulo para Gestión de guerreros

- Módulo para la administración y aprobación de eventos
- Módulo para la Asignación de donaciones
- Módulo de Reportes Dinámicos
- Módulo de Administración de catálogos

4.1.8.3 Interfaces

Las interfaces que se probarán son las siguientes:

- Interfaz de Administración de Roles y Usuarios
- Interfaz de Administración de Voluntarios
- Interfaz de Administración de Donantes
- Interfaz de Administración de Donaciones
- Interfaz de Administración de Guerreros
- Interfaz de Administración de Eventos
- Interfaz de Asignación de Donaciones
- Interfaz de Reporte de Guerreros por prioridad
- Interfaz de Administración de Catálogos

4.1.9 ESTRATEGIA

A fin de dar cumplimiento a este plan, se procederá a utilizar las diferentes funcionalidades del sistema, para poder generar la data necesaria para las pruebas propuestas. Se requerirá la participación del usuario a fin de que el mismo realice las diferentes pruebas planteadas en este documento, las pruebas se realizará en un servidor alternativo que tenga el mismo ambiente que el servidor de producción.

4.1.10 CONDICIONES ESPECIALES

Se describen las condiciones bajo las cuales el plan debe ser suspendido, repetido o culminado, en caso de que estas existan.

Tabla 5.2: Condiciones Especiales para pruebas

Estado	Acontecimientos
Suspendido	<ul style="list-style-type: none">• El usuario no ha asistido a las reuniones• El servidor de base de datos no está operativo
Repetido	<ul style="list-style-type: none">• El sistema no fue capaz de guardar la información generada por el usuario
Culminado	<ul style="list-style-type: none">• La información generada por el usuario fue registrada correctamente en el sistema

4.1.11 RECURSOS

Para la aplicación de pruebas se necesitará a la persona que responsable del sistema dentro de la fundación y se requiere que en el momento de realizar las pruebas, la máquina cuente con lo siguiente:

- Procesador Intel Core I7 2 GHz
- Memoria RAM 8 GB
- Navegador Internet Mozilla.
- Sistema Operativo Ubuntu.
- Postgresql 9
- JAVA 1.6
- Jboss eap 6

4.1.12 RESPONSABLES

Tabla5.3: Responsables del Plan de Pruebas

Responsable	Tareas o Actividades
Jeanneth Farinango	Supervisión del plan de pruebas Pruebas de funcionamiento de interfaces
Jeanneth Farinango	Pruebas de funcionamiento de módulos Entrega de documentación generada Cartas de aceptación de pruebas

4.1.13 CASOS DE PRUEBA

4.1.13.1 CP-01

Caso de Prueba [CP-001]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación: CP – 001

Nombre del Proyecto: JOVCAN – Seguridad y Acceso. Crear Rol

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad de la creación de un nuevo rol en el módulo Seguridad y Acceso.

Requisitos asociados RF. 3.18

Variables de Entrada (Inputs): Descritas en RF. 3.18

Flujo normal del evento Crear nuevo Rol:

1. Se presionará el botón Nuevo
2. Se ingresará el Nombre.
3. Se ingresará la Descripción
4. Se asignará por lo menos un acceso
5. Se presionará el botón Guardar

Resultado esperado: Desplegar un mensaje de registro insertado

Flujo alternativo Crear nuevo rol:

1. Los campos marcados con asterisco (*) no fueron ingresados.
2. El nombre del rol no es único
3. No se seleccionó ningún acceso

Resultado alternativo esperado: No insertará el nuevo registro
Para cada flujo alternativo, los mensajes esperados son los siguientes:

1. "Valor Obligatorio"
2. "El nombre de rol ya existe"
3. "No puede crear un rol sin accesos"

Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]

Fecha de Ejecución: 2013-03-23

Ejecutado por: Jeanneth Farinango

Lugar de ejecución Fundación de Jóvenes Contra el Cáncer

Resultado Culminado

4.1.13.2 CP-02

Caso de Prueba [CP-02]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación: CP – 02

Nombre del Proyecto: JOVCAN - Seguridad y Acceso .Crear Usuario

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad de la creación de un usuario en el módulo Seguridad y Acceso.

Requisitos asociados RF.3.20

Variables de Entrada Descritas RF.3.20

(Inputs):

Flujo normal del evento **Crear Usuario:**

1. Se presionará el botón Nuevo
2. Se ingresará el Número de Cédula
3. Se ingresará el Nick
4. Se ingresará el Nombre
5. Se ingresará el Correo Electrónico
6. Se ingresará el Correo electrónico opcional
7. Se ingresará la clave
8. Se asignará por lo menos un rol
9. Se presionará el botón Guardar

Resultado esperado: Desplegar un mensaje de registro creado con éxito

Flujo alterno **Crear Usuario:**

1. Los campos marcados con asterisco (*) no fueron ingresados.
2. La cédula ingresada no es

		válida.
		3. Correo electrónico ingresado no es válido
		4. No se asignó ningún rol
		5. El Nick del usuario ya existe
Resultado esperado:	alternativo	No se guardará ningún registro Para cada flujo alterno, los mensajes esperados son los siguientes: 1. "Valor Obligatorio" 2. "Cédula de Identidad incorrecta" 3. "Email no válido" 4. "Cada usuario debe tener al menos un rol" 5. "El nombre de usuario ya existe"
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]		
Fecha de Ejecución:	2013-03-23	
Ejecutado por:	Jeanneth Farinango	
Lugar de ejecución	Fundación de Jóvenes Contra el Cáncer	
Resultado	Culminado	

4.1.13.3 CP-03

Caso de Prueba [CP-03]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación:	CP – 03	
Nombre del Proyecto:	JOVCAN - Gestión de voluntarios y donantes .Crear Voluntario	
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad de la creación de un nuevo voluntario en el módulo Gestión de voluntarios y donantes.	
Requisitos asociados	RF 3.1	
Variables de Entrada (Inputs):	Descritas RF. 3.1	
Flujo normal del evento	Crear Voluntario:	
	1.	Se presionará el botón Nuevo
	2.	Se escogerá el tipo
	3.	Se ingresará el nombre
	4.	Se ingresará el apellido

	<ol style="list-style-type: none"> 5. Se ingresará la dirección del domicilio 6. Se ingresará el teléfono del domicilio 7. Se ingresará el teléfono celular 8. Se ingresará el email 9. Se escogerá la provincia 10. Se escogerá la ciudad 11. Se ingresará la fecha de registro 12. Se ingresará la edad 13. Se ingresará horas de voluntariado 14. Se ingresará el número de cédula 15. Dará click en la pestaña “Datos Académicos” 16. Se escogerá el título 17. Se ingresará la institución 18. Se ingresará habilidades y competencias 19. Se ingresará otros 20. Se ingresará Idiomas (Se presionará el botón nuevo y se ingresará el idioma) 21. Dará click en la pestaña “Experiencia laboral” 22. Se presionará el botón Nuevo 23. Se ingresará el cargo 24. Se ingresará la empresa 25. Se señalará si es cargo actual o no 26. Dará click en la pestaña “Preguntas” 27. Se ingresará las respuestas de las preguntas 28. Dará click en la pestaña “Horario” 29. Se presionará el botón Nuevo 30. Se ingresará un horario 31. Se presionará el botón Guardar
Resultado esperado:	Desplegar un mensaje de registro creado con éxito
Flujo alterno	Crear Voluntario: <ol style="list-style-type: none"> 1. Los campos marcados con asterisco (*) no fueron ingresados. 2. La cédula ingresada no es válida. 3. Correo electrónico ingresado

- no es válido
4. Teléfono convencional tiene menos de 9 dígitos
 5. Teléfono celular tiene menos de 10 dígitos
 6. Se ingresaron 2 horarios repetidos
 7. No pueden existir dos voluntarios con el mismo número de cédula
 8. No se puede registrar 2 veces el mismo idioma

Resultado esperado:	alternativo	<p>No se guardará ningún registro</p> <p>Para cada flujo alterno, los mensajes esperados son los siguientes:</p> <ol style="list-style-type: none"> 1. “Valor Obligatorio” 2. “Cédula de Identidad incorrecta” 3. “Email no válido” 4. “Teléfono no válido” 5. “Teléfono no válido” 6. “Se encontraron errores: Existen horarios temprano/tarde que se encuentran repetidos” 7. “La cédula de identidad ya ha sido registrada” 8. “Se encontraron errores: Existen idiomas que se encuentran repetidos”
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]		
Fecha de Ejecución:	2013-03-23	
Ejecutado por:	Jeanneth Farinango	
Lugar de ejecución	Fundación de Jóvenes Contra el Cáncer	
Resultado	Culminado	

4.1.13.4 CP-04

Caso de Prueba [CP-04]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación: CP – 04

Nombre del Proyecto: JOVCAN - Gestión de voluntarios y donantes. Crear Donante

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad de la creación de un nuevo donante en el módulo Gestión de voluntarios y donantes.

Requisitos asociados RF 3.6

Variables de Entrada Descritas RF 3.6
(Inputs):

Flujo normal del evento

Crear Donante:

1. Se presionará el botón Nuevo
2. Se escogerá el Tipo de Donante
 - Persona
 1. Se ingresará el nombre
 2. Se ingresará el apellido
 3. Se ingresará el número de identificación
 4. Se ingresará la dirección
 5. Se ingresará el teléfono.
 - Empresa
 1. Se ingresará el nombre
 2. Se ingresará el número de identificación
 3. Se ingresará la dirección
 4. Se ingresará el teléfono
 5. Se ingresará el responsable

3. Se presionará el botón Guardar

Resultado esperado: Desplegar un mensaje de registro creado con éxito

Flujo alternativo

Crear Donante:

1. Los campos marcados con asterisco (*) no fueron ingresados.

Resultado alternativo esperado: No se guardará ningún registro

Para cada flujo alternativo, los mensajes esperados son los siguientes:

1. “Valor Obligatorio”

Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]

Fecha de Ejecución: 2013-03-23

Ejecutado por: Jeanneth Farinango

Lugar de ejecución Fundación de Jóvenes Contra el Cáncer

Resultado Culminado

4.1.13.5 CP-05

Caso de Prueba [CP-05]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación:	CP – 05
Nombre del Proyecto:	JOVCAN - Gestión de donaciones. Crear Donación
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad de la creación de una donación en el módulo Gestión de donaciones.
Requisitos asociados	RF 3.10
Variables de Entrada (Inputs):	Descritas RF 3.10
Flujo normal del evento	Crear nueva donación: <ol style="list-style-type: none">1. Se presionará el botón Nuevo2. Se escogerá el Donante3. Se escogerá el Tipo de Inventario4. Se escogerá el Inventario5. Se ingresará la Cantidad.6. Se presionará el botón Guardar
Resultado esperado:	Desplegar un mensaje de registro insertado.
Flujo alterno	Crear nuevo donación: <ol style="list-style-type: none">1. Los campos marcados con asterisco (*) no fueron ingresados.
Resultado alternativo esperado:	No se guardará ningún registro Para cada flujo alterno, los mensajes esperados son los siguientes: <ol style="list-style-type: none">1. “Valor Obligatorio”
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]	
Fecha de Ejecución:	2013-03-23
Ejecutado por:	Jeanneth Farinango
Lugar de ejecución	Fundación de Jóvenes Contra el Cáncer
Resultado	Culminado

4.1.13.6 CP-06

Caso de Prueba [CP-06]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación: CP – 06

Nombre del Proyecto: JOVCAN - Administración de catálogos.
Crear catálogo

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad de Añadir Ítems a un catálogo en el módulo Administración de catálogos.

Requisitos asociados RF 3.23

Variables de Entrada (Inputs): Descritas RF 3.23

Flujo normal del evento Crear catálogo :

1. Se presionará el botón Nuevo
2. Se presionará el botón “...”
3. Se escogerá el tipo de catálogo
4. Se ingresará el Nemónico
5. Se ingresará el Nombre
6. Se presionará el botón Guardar

Resultado esperado: Desplegar un mensaje de registro insertado.

Flujo alternativo

1. Los campos marcados con asterisco (*) no fueron ingresados.
2. El nemónico ya existe

Resultado alternativo esperado: No se guardará ningún registro

Para cada flujo alternativo, los mensajes esperados son los siguientes:

1. “Valor Obligatorio”
2. “Nemónico ya existe”

Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]

Fecha de Ejecución: 2013-03-23

Ejecutado por: Jeanneth Farinango

Lugar de ejecución Fundación de Jóvenes Contra el Cáncer

Resultado Culminado

4.1.13.7 CP-07

Caso de Prueba [CP-07]

Elaborado por: Paúl Jarrín, Nelson Jumbo

Código de Identificación:	CP – 07
Nombre del Proyecto:	JOVCAN - Reportes Dinámicos. Generar el Reporte de Guerreros por prioridad
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad al generar el Reporte de Guerreros por prioridad en el módulo Reportes Dinámicos.
Requisitos asociados	RF.3.60
Variables de Entrada (Inputs):	Descritas RF. 3.60
Flujo normal del evento	Generar el Reporte de Guerreros : 1. Se indicará si la columna medicamentos aparece o no en el reporte 2. Se indicará si la columna cédula aparece o no en el reporte 3. Se escogerá la etapa de la enfermedad 4. Se presionará el botón PDF 5. Se presionará el botón EXCEL
Resultado esperado:	Se generarán los reportes en PDF y EXCEL
Flujo alterno	No existen guerreros registrados en la base de datos
Resultado alternativo esperado:	El reporte mostrará el mensaje: "No existen datos para mostrar"
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]	
Fecha de Ejecución:	2013-03-23
Ejecutado por:	Jeanneth Farinango
Lugar de ejecución	Fundación de Jóvenes Contra el Cáncer
Resultado	Culminado

Caso de Prueba [CP-08]

Elaborado por: Paúl Jarrín y Nelson Jumbo

Código de Identificación: CP – 08

Nombre del Proyecto: JOVCAN - Módulo para Gestión de guerreros. Crear Guerrero

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad de la creación de un nuevo usuario en el módulo Administrador.

Requisitos asociados RF 3.33

Variables de Entrada (Inputs): Las descritas en: RF 3.33

Flujo normal del evento Crear nuevo Guerrero:

1. Presionará el botón Nuevo
2. Se ingresará Nombres
3. Se ingresará Apellidos
4. Se ingresará Fecha de Nacimiento
5. Se ingresará Edad
6. Se ingresará Provincia de Nacimiento
7. Se ingresará Ciudad de Nacimiento
8. Se ingresará Email
9. Se ingresará Cédula de identidad
10. Presionará el botón Editar en el campo Carnet CONADIS
 - a. Presionará el botón Editar
 - b. Se ingresará Número de carnet
 - c. Se ingresará Tipo de discapacidad
 - d. Se ingresará Porcentaje de discapacidad.
 - e. Presionará el botón Guardar
 - f. Se desplegará un mensaje indicando que el registro se ha ingresado con éxito.
 - g. Presionará el botón Cerrar
11. Se ingresará Dirección

de Domicilio

12. Se ingresará Provincia de Domicilio

13. Se ingresará Ciudad de Domicilio

14. Se ingresará Teléfono domicilio

15. Se ingresará Teléfono Celular

16. Se ingresará Costea o no sus gastos

17. Se ingresará Recibe o no el Bono Joaquín Gallegos Lara

18. Se ingresará Creencia o religión

19. Se ingresará Deporte que practica

20. Se ingresará Hobby

21. Se ingresará Comida favorita

22. Presionar el botón Editar junto al campo Lugares que desea conocer

a. Presionará el botón Agregar

b. Ingresará el nombre del lugar que desea conocer

c. Dará clic en el botón Guardar

d. Se desplegará un mensaje indicando que el registro se ha ingresado con éxito.

e. Dará clic en el botón cerrar

23. Presionará el botón Editar junto al campo Deseos

a. Presionará el botón Agregar

b. Ingresará la descripción del deseo

c. Dará clic en el botón Guardar

d. Se desplegará un mensaje indicando que el registro se ha ingresado con éxito.

e. Dará clic en el botón

cerrar

24. Seleccionará la fecha de registro del guerrero
 25. Escogerá la clasificación del Guerrero
 26. Dará clic en la pestaña Contactos
 27. Presionará el botón Nuevo
 28. Aparecerá una venta emergente para registrar el contacto
 - a. Se ingresará Cédula
 - b. Se ingresará Nombres
 - c. Se ingresará Apellidos
 - d. Se ingresará Edad
 - e. Se ingresará Trabaja en
 - f. Se ingresará Teléfono domicilio
 - g. Se ingresará Teléfono Celular
 - h. Se ingresará Tipo de Contacto
 - i. Se ingresará Parentezco
 - j. Se ingresará Vive o no con el guerrero.
 - k. Presionará el botón Guardar
 - l. Se desplegará un mensaje indicando que el Contacto se ha añadido con éxito
 - m. Presionará el botón Cerrar
 29. Dará clic en la pestaña Datos Académicos
 - a. Se indicará Estudia o no actualmente
 - b. Se ingresará Actual o última institución
 - c. Se seleccionará Nivel o Curso Se ingresará
 - d. Se ingresará Especialidad/Carrera
 - e. Se escogerá Tipo de institución
 - f. Se indicará si Recibe o no beca.
 30. Dará clic en la pestaña
-

Datos Laborales

- a. Indicará si Trabaja o no Actualmente
- b. Se ingresará Empresa
- c. Se ingresará Cargo
- d. Se ingresará Área
- e. Se seleccionará la fecha Desde la cual trabaja
- f. Se indicará si Tiene o no IESS

31. Dará clic en la pestaña Condición Médica

a. Ingresará los Datos Generales de la Condición Médica

- i. Se Escogerá Etapa
- ii. Se ingresará Diagnóstico
- iii. Se indicará si Se encuentra o no en Tratamiento
- Se ingresará Condición Física
- iv. Se ingresará Medicamentos
- v. Se ingresará Observaciones
- vi. Se indicará si Ha abandonado o no su tratamiento

- 1. Se ingresará la Razón por la cual ha abandonado su tratamiento.

b. Ingresará la información sobre Quimioterapias

- i. Seleccionará la Fecha Inicio
- ii. Seleccionará la Fecha Fin
- iii. Escogerá el Hospital

c. Ingresará la información sobre las Deudas por tratamiento que ha adquirido el guerrero

- i. Se ingresará el Total de la deuda
-

- ii. Se ingresará el Valor mensual que abona
- iii. Se escogerá la Forma de pago
- iv. Se escogerá la Institución a la que adeuda.

Resultado esperado:	Desplegar un mensaje de registro insertado con éxito
Flujo alterno	<p>Crear nuevo Guerrero:</p> <ol style="list-style-type: none"> 1. No se ingresaron datos obligatorios (aquellos marcados con *) 2. La cédula ingresada no es correcta 3. La cédula ingresada ya fue registrada para otro guerrero 4. No se ingresó la información del carnet del conadis 5. No se ingresaron al menos 1 deseo y 1 lugar 6. No se ingresó al menos un Contacto de tipo FAMILIAR 7. No se ingresaron datos académicos 8. No se ingresó la etapa de la enfermedad del guerrero 9. No se ingresó la razón por la cual el guerrero abandonó su tratamiento 10. Teléfono celular ingresado no tiene 10 dígitos 11. Teléfono convencional no tiene 9 dígitos

Resultado alternativo esperado:	<p>No se guardará ningún registro</p> <p>Para cada flujo alterno, los mensajes esperados son los siguientes:</p> <ol style="list-style-type: none"> 1. “Valor Obligatorio” 2. “Cédula de Identidad incorrecta” 3. “Esta cédula de identidad ya ha sido registrada” 4. “Debe ingresar un carnet del CONADIS para el guerrero” 5. “Debe ingresar al menos
--	---

- un deseo/lugar”
- 6. “Debe ingresar al menos un contacto de tipo:FAMILIAR, para el guerrero”
- 7. “Por favor ingrese todos los campos obligatorios en la pestaña [Datos Académicos]”
- 8. “Por favor ingrese todos los campos obligatorios en la pestaña [Condición Médica]”
- 9. “Por favor ingrese la razón por la cual el guerrero abandonó el tratamiento [Condición Médica]”
- 10. “Teléfono no válido”
- 11. “Teléfono no válido”

Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]

Fecha de Ejecución:	2013-03-23
Ejecutado por:	Jeanneth Farinango
Lugar de ejecución	Fundación Jóvenes Contra el Cáncer
Resultado	Culminado

Caso de Prueba [CP-09]

Elaborado por: Paúl Jarrín y Nelson Jumbo

Código de Identificación: CP – 09

Nombre del Proyecto: JOVCAN – Módulo para la administración y Aprobación de Eventos. Crear Evento

Descripción (Alcance y Objetivos): Detectar los posibles defectos de la funcionalidad en la creación y aprobación de eventos

Requisitos asociados RF 3.47

Variables de Entrada (Inputs): Las descritas en RF 3.47

Flujo normal del evento **Crear Evento:**

1. Dará clic en el botón Nuevo
2. Se ingresará Nombre
3. Se ingresará Lugar
4. Se ingresará Descripción
5. Se seleccionará Fecha Inicio
6. Se seleccionará Fecha Fin
7. Se seleccionará Hora Inicio
8. Se seleccionará Hora Fin
9. Dará clic en el pestaña invitaciones
 - a. Aparecerán aquellos voluntarios cuyo horario coincida con el del evento
 - b. Aparecerán todos los guerreros
 - c. Se presionará el botón Confirmar Asistencia para añadir un Guerrero a la lista de asistentes
 - d. Se presionará el botón Cancelar Asistencia para quitar un Guerrero a la lista de asistentes
10. Se presionará el botón Guardar

Resultado esperado: El Evento queda guardado con estado REPROBADO
Se Desplegará un mensaje indicando que el registro se ha guardado con éxito

Flujo alternativo **Crear evento:**

1. No se ingresaron datos obligatorios (aquellos marcados con *)

Resultado esperado:	alternativo	No se creará ningún evento Para cada flujo alterno, los mensajes esperados son los siguientes: 1. “Valor Obligatorio”
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]		
Fecha de Ejecución:	2013-03-23	
Ejecutado por:	Jeanneth Farinango	
Lugar de ejecución	Fundación Jóvenes Contra el Cáncer	
Resultado	Culminado	

4.1.13.10CP-10

Caso de Prueba [CP-10]

Elaborado por: Paúl Jarrín y Nelson Jumbo

Código de Identificación:	CP – 10	
Nombre del Proyecto:	JOVCAN – Módulo para la administración y Aprobación de Eventos. Aprobar Evento	
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad mediante la cual se aprueban los eventos incluyendo el envío de invitaciones y la confirmación de asistencia mediante email.	
Requisitos asociados	RF 3.49	
Variables de Entrada (Inputs):	Las descritas en RF 3.49	
Flujo normal del evento	Aprobar Evento: 1. En la lista de eventos registrados en el sistema se presionará el botón Aprobar en el evento deseado 2. Se mostrará una alerta para confirmar la aprobación del evento. 3. Se presionará el botón Aceptar 4. Se enviarán las invitaciones a los emails de cada uno de los guerreros y voluntarios disponibles para el evento. 5. Se revisará el correo electrónico de algún guerrero y de	

		algún voluntario invitado
		6. Se confirmará la asistencia
		7. Se re direccionará al usuario a una pantalla agradeciéndole por su asistencia
		8. El estado de asistencia del guerrero y del voluntario han cambiado a ASISTE.
Resultado esperado:		<ol style="list-style-type: none"> 1. Las invitaciones han sido enviadas a los correos electrónicos respectivos 2. El estado de asistencia cambia a ASISTE cuando el guerrero o el voluntario han confirmado su asistencia desde el correo electrónico. 3. Se desplegará un mensaje indicando que se ha guardado el registro con éxito.
Flujo alterno		El servidor de correo no fue capaz de despachar los mensajes
Resultado alternativo esperado:		No se envían las invitaciones a los correos electrónicos
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]		
Fecha de Ejecución:		2013-03-23
Ejecutado por:		Jeanneth Farinango
Lugar de ejecución		Fundación Jóvenes Contra el Cáncer
Resultado		Culminado

4.1.13.11 CP-11

Caso de Prueba [CP-11]

Elaborado por: Paúl Jarrín y Nelson Jumbo

Código de Identificación:	CP – 11
Nombre del Proyecto:	JOVCAN – Módulo para la Asignación de Donaciones. Registrar Asignaciones de Donaciones (Varias)
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad de Asignar una donación a: FUNDACIONES Aliadas, Familias, Varios.
Requisitos asociados	RF 3.55
Variables de Entrada	Las Descritas en RF 3.55

(Inputs):	
Flujo normal del evento	Crear Asignaciones Varias: <ol style="list-style-type: none"> 1. Se presionará el botón Crear Asignaciones Varias 2. Se escogerá el Tipo de Asignación <ol style="list-style-type: none"> a. Si el Tipo de Asignación es Fundaciones Aliadas se escogerá una Fundación Aliada 3. Se escogerá el Tipo de Inventario 4. Se escogerá el Ítem del Inventario 5. Se ingresará la cantidad para asignar 6. Se presionará el botón Guardar
Resultado esperado:	Se desplegará un mensaje indicando que se la asignación se ha realizado con éxito.
Flujo alterno	Crear Asignaciones Varias: <ol style="list-style-type: none"> 1. No se ingresaron los campos obligatorios(aquellos marcados con *) 2. No se seleccionó un Ítem del Inventario 3. La cantidad especificada para la asignación excede el stock disponible 4. La cantidad establecida es CERO
Resultado esperado:	alternativo No se ingresa la asignación Para cada flujo alterno, los mensajes esperados son los siguientes: <ol style="list-style-type: none"> 1. “Valor Obligatorio” 2. “Debe seleccionar un ítem del inventario para realizar la asignación” 3. “No hay stock suficiente para la asignación” 4. “La cantidad para asignar no puede ser cero”
Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]	
Fecha de Ejecución:	2013-03-23
Ejecutado por:	Jeanneth Farinango
Lugar de ejecución	Fundación Jóvenes Contra el Cáncer
Resultado	Culminado

Caso de Prueba [CP-12]

Elaborado por: Paúl Jarrín y Nelson Jumbo

Código de Identificación:	CP – 12
Nombre del Proyecto:	JOVCAN – Módulo para la Asignación de Donaciones. Registrar Asignaciones de Donaciones (Guerrero)
Descripción (Alcance y Objetivos):	Detectar los posibles defectos de la funcionalidad de Asignar una donación a un Guerrero
Requisitos asociados	RF 3.55
Variables de Entrada (Inputs):	Las Descritas en RF 3.55
Flujo normal del evento	Crear Asignaciones Guerrero: <ol style="list-style-type: none"> 1. Se escogerá el Tipo de Inventario 2. Se escogerá el Ítem del Inventario 3. Se seleccionará el Guerrero para realizar la asignación 4. Se ingresará la cantidad para asignar 5. Se presionará el botón Guardar
Resultado esperado:	Se desplegará un mensaje indicando que se la asignación se ha realizado con éxito.
Flujo alterno	Crear Asignaciones Guerrero: <ol style="list-style-type: none"> 1. No se ingresaron los campos obligatorios(aquellos marcados con *) 2. No se seleccionó un Ítem del Inventario 3. La cantidad especificada para la asignación excede el stock disponible 4. La cantidad establecida es CERO
Resultado esperado:	alternativo No se ingresa la asignación <p>Para cada flujo alterno, los mensajes esperados son los siguientes:</p> <ol style="list-style-type: none"> 1. “Valor Obligatorio” 2. “Debe seleccionar un ítem del inventario para realizar la asignación” 3. “No hay stock suficiente para la asignación” 4. “La cantidad para asignar no puede

ser cero”

Evaluación de prueba [refiere a los aspectos que resultaron de aplicar el caso de prueba y las variables referentes a este]

Fecha de Ejecución: 2013-03-23

Ejecutado por: Jeanneth Farinango

Lugar de ejecución Fundación Jóvenes Contra el Cáncer

Resultado Culminado

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

Al culminar el presente proyecto se han cumplido con los objetivos propuestos y por lo tanto se exponen a continuación las siguientes conclusiones y recomendaciones:

5.1 CONCLUSIONES

- La metodología ágil XP se basa tener entregas parciales y funcionales del producto lo cual permitió que la fundación tenga las partes más críticas del sistema en funcionamiento lo antes posible, también ayudó a que se hagan correcciones tempranas causando un impacto leve al desarrollo del proyecto.
- Se desarrolló un módulo de auditoria el cual lleva un control de todas las operaciones que el usuario realiza en el sistema, permitiendo registrar la inserción, actualización y eliminación de registros de cada una de las tablas de la base de datos, lo cual permite a la fundación tener un control adecuado sobre el manejo de la información generada por el sistema.
- La instalación del sistema sobre el servidor JBOSS AS 7 fue óptima debido a que este servidor es un contenedor JEE6 certificado lo cual garantiza una implementación correcta de la especificación, evitando bugs y permitiendo al desarrollador crear código sin preocuparse del servidor de aplicaciones sobre el cual va a desplegar su aplicación.

5.2 RECOMENDACIONES

- En la ESPE se debería dedicar una materia para enseñar metodologías ágiles de desarrollo, entre ellas XP, ya que desde el año 2012 han tomado fuerza y se están volviendo tendencia debido a que el usuario desea ver un producto parcial con carácter de producción lo antes posible, permitiéndole detectar errores y posibles cambios, mismos que pueden ser visibles solo con el software en funcionamiento.
- En un ambiente donde los requerimientos cambian constantemente o cuando el usuario no tiene claro lo que desea que haga el sistema se recomienda utilizar la metodología Extreme Programming (XP), ya que permite que el impacto de dichos cambios sea bajo debido a las entregas parciales de software que se hacen al cliente.
- En proyectos que son desarrollados por un grupo de personas es vital el contar con un repositorio de código como por ejemplo CVS, SVN que permita a los desarrolladores unificar los avances hechos por cada uno, permitiéndoles a todos trabajar sobre la última actualizada del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- Cohn, M. (2004). *User stories applied: for agile software development*. Boston: Pearson Education.

REFERENCIAS WEB

- Chuto, A. (s.f.). *SlideShare*. Obtenido de <http://www.slideshare.net/armandorob/objeto-relacional-bases-datos-2>
- dosIdeas.com. (s.f.). *Maven*. Obtenido de <http://www.dosideas.com>
- Dzone. (s.f.). <http://refcardz.dzone.com>. Obtenido de <http://refcardz.dzone.com/refcardz/getting-started-with-jpa>
- Dzone. (s.f.). refcardz.dzone.com. Obtenido de <http://refcardz.dzone.com/refcardz/contexts-and-dependency>
- Fundación Jóvenes Contra el Cáncer. (s.f.). *Jóvenes Contra el Cáncer*. Obtenido de <http://www.jovenescontraelcancer.org/about.html>
- Fundación Jóvenes Contra el Cáncer. (s.f.). *Jóvenes Contra el Cáncer*. Obtenido de <https://www.facebook.com/jovenescontraelcancer/info>
- Jboss. (s.f.). *Richfaces 4.2 Developer Guide*. Obtenido de http://docs.jboss.org/richfaces/latest_4_2_X/Developer_Guide/en-US/html/
- OpenEcuador.org. (s.f.). *Postgresql*. Obtenido de <http://www.openecuador.org/modules/news/article.php?storyid=31>
- Oracle. (s.f.). *JEE6 Overview*. Obtenido de <http://www.oracle.com/technetwork/articles/javaee/javaee6overview-141808.html>
- Oracle. (s.f.). *Jee6 Tutorial*. Obtenido de <http://docs.oracle.com/javaee/6/tutorial/doc/gircz.html>
- Oracle. (s.f.). *JEE6 Tutorial-JPA*. Obtenido de <http://docs.oracle.com/javaee/6/tutorial/doc/bnbqa.html>
- Rose India Technologies PVT.LTD. (s.f.). www.roseindia.net. Obtenido de <http://www.roseindia.net/jsf/jsflifecycle.shtml>
- Xtreme Programming. (s.f.). *extremeprogramming.org*. Obtenido de <http://www.extremeprogramming.org/map/development.html>

BIOGRAFÍA

PAÚL ALEJANDRO JARRÍN PEREIRA

Nombres y Apellidos: Paúl Alejandro Jarrín Pereira

Lugar de Nacimiento: Santa Prisca, Quito, Pichincha

Fecha de Nacimiento: 30 de Abril de 1988

Educación Primaria: Escuela Marista "Borja 2"
1994-2000

Educación Secundaria: Colegio Marista "Borja 2"
2000-2006

Bachiller en Ciencias, especialización Físico Matemático

Educación Superior: Escuela Politécnica del Ejército ESPE
2006-2011

Ingeniería en Sistemas e Informática

Educación Idiomas: Escuela Politécnica del Ejército ESPE

Certificación Suficiencia en Inglés

BIOGRAFÍA

NELSON ALBERTO JUMBO HIDALGO

Nombres y Apellidos: Nelson Alberto Jumbo Hidalgo

Lugar de Nacimiento: Sangolquí, Rumiñahui, Pichincha

Fecha de Nacimiento: 16 de Septiembre de 1988

Educación Primaria: Escuela Fiscal Mixta Teodoro Wolf
1994-2000

Educación Secundaria: Colegio Experimental "Juan Pío Montúfar"
2000-2006

Bachiller en Ciencias, especialización Físico Matemático

Educación Superior: Escuela Politécnica del Ejército ESPE
2006-2011

Ingeniería en Sistemas e Informática

Educación Idiomas: Escuela Politécnica del Ejército ESPE
Certificación Suficiencia en Inglés

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

PAÚL ALEJANDRO JARRÍN PEREIRA

Y

NELSON ALBERTO JUMBO HIDALGO

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña MsC.

Lugar y fecha: _____

ANEXO A
DIAGRAMA ENTIDAD RELACIÓN

ANEXO B
DIAGRAMA DE CLASES DEL MODELO DE PERSISTENCIA