

**UNIVERSIDAD MAYOR DE SAN ANDRES**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMATICA**



**PROYECTO DE GRADO**

**SISTEMA WEB DE GESTION Y DIFUSION DE BECAS**

**Caso: EGPP (Escuela de Gestión Pública plurinacional)**

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA MENCION: INGENIERIA  
DE SISTEMAS INFORMATICOS

Postulante: Richar Angel Marquez Zacari

Tutor Metodológico: MSc. Fatima Consuelo Dolz de Moreno

Asesor: Lic. Carmen Rosa Huanca Quisbert

LA PAZ – BOLIVIA

2014



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## DEDICATORIA

A mi señor Padre Pastor Marquez por su apoyo incondicional, a mi Señora Madre Damiana que desde el cielo me sigue apoyando, a mi hermana Rosse Mary Naty por ayudarme en las buenas y malas, y mi familia entera por su cariño.

Richar Angel Marquez Zacari

## AGRADECIMIENTOS

A Dios por estar presente en todo momento y permitirme compartir con mis seres queridos buenos momentos.

A mi docente tutor MSc. Fatima Consuelo Dolz de Moreno por su valiosa guía en el presente trabajo.

A mi docente Asesor Lic. Carmen Rosa Huanca Quisbert por su confianza, orientación y sus continuas observaciones y correcciones para la buena elaboración del presente trabajo.

Al Ing. Sergio Sanchez Navas por la confianza depositada en mi persona para el desarrollo del presente proyecto.

Al Lic. Guillermo Delgado por su colaboración y comprensión en las diferentes etapas del proyecto.

## RESUMEN

El presente proyecto fue desarrollado para la Escuela de Gestión Pública Plurinacional (EGPP) en atención al requerimiento de su Unidad de Becas la cual está encargada de administrar las ofertas de becas de la cooperación internacional que llega bajo convenio con el gobierno Boliviano.

En la actualidad de EGPP cuenta con un sistema web en funcionamiento el cual fue elaborado con la herramienta de software propietario Genexus, el mismo que entre sus diferentes módulos se tiene al módulo de becas, la Unidad de Becas ha estado trabajando específicamente con el módulo de becas alimentando a este la información referente a la oferta de becas.

El proyecto de grado responde a la necesidad de la Unidad de Becas de contar con un Sistema Web elaborado bajo software libre que cubra las falencias que se ha encontrado en el módulo de becas a lo largo del tiempo.

La metodología de desarrollo que se empleó para el presente proyecto fue a la Programación Extrema (XP) conjuntamente con el Lenguaje de Modelado Web (WebML). Para la implementación se utilizó herramientas de software libre.

# INDICE

## CAPITULO I

### INTRODUCCION

|   |    |
|---|----|
| 1.1. INTRODUCCION.....                                  | 1  |
| 1.2. ANTECEDENTES.....                                  | 2  |
| 1.2.1. ANTECEDENTES INSTITUCIONALES.....                | 2  |
| 1.2.2. GENEXUS.....                                     | 4  |
| 1.2.3. SISTEMA WEB DE LA EGPP.....                      | 6  |
| 1.2.4. TRABAJOS SIMILARES.....                          | 8  |
| 1.3. PLANTEAMIENTO DE PROBLEMA.....                     | 8  |
| 1.4. OBJETO DE ESTUDIO.....                             | 9  |
| 1.5. OBJETIVOS.....                                     | 9  |
| 1.5.1 OBJETIVO GENERAL.....                             | 9  |
| 1.5.2. OBJETIVOS ESPECIFICOS.....                       | 9  |
| 1.6. JUSTIFICACION.....                                 | 10 |
| 1.6.1. JUSTIFICACION ECONOMICA.....                     | 10 |
| 1.6.2. JUSTIFICACION TECNICA.....                       | 10 |
| 1.6.3. JUSTIFICACION SOCIAL.....                        | 10 |
| 1.7. LIMITES Y ALCANCES.....                            | 11 |
| 1.8. METODOLOGIA, METODOS, TECNICAS Y HERRAMIENTAS..... | 11 |

## CAPITULO II

### MARCO TEORICO

|  |    |
|--|----|
| 2.1. INTRODUCCION.....                           | 13 |
| 2.2. INGENIERIA DE SOFTWARE.....                 | 13 |
| 2.3. METODOLOGIAS DE DESARROLLO DE SOFTWARE..... | 14 |
| 2.3.1. METODOLOGIAS TRADICIONALES.....           | 14 |
| 2.3.2. METODOLOGIAS AGILES.....                  | 14 |

|   |    |
|---|----|
| 2.4. PROGRAMACION EXTREMA (XP).....         | 15 |
| 2.4.1. PLANEACION.....                      | 16 |
| 2.4.2. DISEÑO.....                          | 18 |
| 2.4.3. CODIFICACION.....                    | 20 |
| 2.4.4. PRUEBAS.....                         | 22 |
| 2.5. BASES DE DATOS.....                    | 22 |
| 2.5.1. NORMALIZACION DE BASES DE DATOS..... | 22 |
| 2.6. WEBML.....                             | 24 |
| 2.6.1. MODELO ESTRUCTURAL.....              | 24 |
| 2.6.2. MODELO DE HIPERTEXTO.....            | 25 |
| 2.6.2.1. MODELO DE COMPOSICION.....         | 25 |
| 2.6.2.2. MODELO DE NAVEGACION.....          | 28 |
| 2.6.3. MODELO DE PRESENTACION.....          | 28 |
| 2.6.4. MODELO DE PERSONALIZACION.....       | 28 |
| 2.7. SOFTWARE SEGURO.....                   | 29 |
| 2.7.1. CRIPTOGRAFIA.....                    | 30 |
| 2.7.1.1. ENCRIPACION CON MD5.....           | 30 |
| 2.8. CALIDAD DEL SOFTWARE.....              | 31 |
| 2.8.1. METRICAS DE CALIDAD.....             | 31 |

### CAPITULO III

#### MARCO APLICATIVO

|  |    |
|--|----|
| 3.1. INTRODUCCION.....                   | 36 |
| 3.2. ESTUDIO DE LA SITUACION ACTUAL..... | 36 |
| 3.2.1. ORGANIGRAMA INSTITUCIONAL.....    | 37 |
| 3.3. DESARROLLO DEL SISTEMA WEB.....     | 38 |
| 3.3.1. PLANEACION.....                   | 39 |
| 3.3.1.1. EQUIPO DE TRABAJO.....          | 39 |
| 3.3.1.2. HISTORIAS DE USUARIO.....       | 39 |
| 3.3.1.3. PLAN DE ENTREGAS.....           | 56 |
| 3.3.2. DISEÑO Y CODIFICACION.....        | 57 |

|  |    |
|--|----|
| 3.3.2.1. MODULO DE CONVOCATORIAS.....      | 57 |
| 3.3.2.2. MODULO DE INGRESO DE USUARIO..... | 66 |
| 3.3.2.3 MODULO DE POSTULANTES.....         | 70 |
| 3.3.3. PRUEBAS.....                        | 81 |

## CAPITULO IV

### CALIDAD DEL SOFTWARE

|                          |    |
|--------------------------|----|
| 4.1. INTRODUCCION.....   | 87 |
| 4.2. FUNCIONALIDAD.....  | 87 |
| 4.3. USABILIDAD.....     | 89 |
| 4.4. CONFIABILIDAD.....  | 90 |
| 4.5. MANTENIBILIDAD..... | 92 |
| 4.6. PORTABILIDAD.....   | 92 |

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

|                           |    |
|---------------------------|----|
| 5.1. CONCLUSIONES.....    | 94 |
| 5.2. RECOMENDACIONES..... | 95 |

## BIBLIOGRAFIA

### ANEXOS

ANEXO A – ARBOL DE PROBLEMAS

ANEXO B – ARBOL DE OBJETIVOS

ANEXO C – MARCO LOGICO



## CAPITULO I

### INTRODUCCION

#### 1.1. INTRODUCCION

La información que se produce y maneja en las instituciones públicas y privadas hoy en día, es considerada como uno de los recursos más importantes. La correcta administración de esta trae consigo beneficios como: la entrega oportuna de información, apoyo en la toma de decisiones, menos recursos económicos en personal y material de escritorio, etc.

En toda institución se puede observar que con el transcurrir del tiempo el incremento de la información crece considerablemente, y que el medio más usado para el almacenamiento y procesamiento de esta creciente información es un sistema de información.

Un sistema de información en informática es cualquier sistema computacional que se utilice para obtener, almacenar, manipular, administrar, controlar, procesar, transmitir, o recibir datos para satisfacer una necesidad de información. [Sistema de información, 2014]

Esta herramienta basada en la web pasa a ser universalmente accesible con lo cual nos ahorra tiempo y dinero en la búsqueda de información y difusión de la misma. Considerando estos beneficios la Escuela de Gestión Pública Plurinacional la cual es institución pública, utiliza un sistema de información web (sistema web) en sus actividades con el objetivo de automatizar los procesos de gestión y difusión de información.

Respecto a las herramientas software el Estado Plurinacional de Bolivia realizó la promulgación del Decreto Supremo N°1874, el cual insta a las instituciones públicas a migrar al uso del software libre<sup>1</sup> y estándares abiertos<sup>2</sup>. [Decreto Supremo 1874, 2014]

En el presente proyecto de grado se realizará el desarrollo del sistema web de gestión y difusión de becas, este reemplazará al módulo de becas el cual se encuentra integrado en el sistema web de la EGPP, también se contemplará al decreto supremo N° 1874.

---

1 son programas de ordenador que pueden ser utilizados, copiados, modificados y redistribuidos libremente por los usuarios.  
2 son especificación de formato de archivo abierto disponible públicamente para lograr una tarea específica. Por ejemplo HTML, TXT, JGEP o PDF son ejemplo de formatos de archivos abierto.

## 1.2. ANTECEDENTES

### 1.2.1. ANTECEDENTES INSTITUCIONALES

La Escuela de Gestión Pública Plurinacional (EGPP), fue creada el 15 de julio de 2009 mediante Decreto Supremo Nº 212, Presidencia del Estado Plurinacional de Bolivia, como una entidad descentralizada, dependiente del Ministerio de Educación. Tiene el propósito de contribuir a la transformación del Estado Plurinacional, a través de la capacitación, investigación y asistencia técnica, a las y los servidores públicos, como a las entidades del Estado en los diferentes niveles de gobierno (Central, departamental, municipal y autonomías indígenas). [Decreto Supremo 0212, 2014]

Entre sus atribuciones establecidas en el Decreto Supremo, la Escuela de Gestión Pública Plurinacional desarrollara tres ámbitos de formación:

- Programas de postgrado especializados en diferentes áreas de la gestión pública (diplomados, maestrías y doctorados).
- Programas de formación continua especializada en diferentes áreas de la gestión pública (cursos, talleres, seminarios y otros).
- Programa de formación, capacitación y certificación de competencias a autoridades y líderes de los pueblos indígena originario campesinos.

En la actualidad la administración de la oferta de becas es realizada por la Escuela de Gestión Pública Plurinacional en cumplimiento a la Resolución Ministerial Nº 836/09, por medio de la cual el Ministerio de Educación delega esta función a esta entidad. Los procedimientos se encuentran normados a través de un Reglamento de Becas, el mismo que se encuentra aprobado mediante la Resolución Suprema Nº 218051 del año 1997, actualizado a través de la R.A. 036/2002, emitido por el Servicio Nacional de Administración de Personal (SNAP).

La Escuela de Gestión Pública Plurinacional (EGPP), está encargada de la canalización de la oferta de becas de aproximadamente 30 países u organismos de cooperación. Dichas becas están orientadas a todo ciudadano boliviano que radica en Bolivia y que cumple con los

requisitos establecidos en la respectiva convocatoria. Las becas ofertadas pueden ser completas o parciales, para cursos cortos, cursos de postgrado (diplomado, pasantía, maestría, doctorado, investigación, etc.) o pregrado (licenciatura, técnico superior y técnico medio), y que abarcan prácticamente todas las áreas del conocimiento impartidas en diversos idiomas, generalmente en inglés y en forma creciente, en español.

Existen becas orientadas al público tanto al sector público como al sector privado (esto se encuentra determinado en la convocatoria). Con la finalidad de que la información pueda llegar a los todos los sectores, especialmente aquellos que históricamente han sido excluidos, la EGPP difunde la oferta de becas actualizada a través de los siguientes canales:

- Página web de la EGPP <http://www.egpp.gob.bo> (actualizada semanalmente)
- Folleto Informativo (publicado mensualmente)
- Correos electrónico; La EGPP cuenta con el servicio de mailing a través del cual, la oferta de becas actualizada en forma semanal es remitida a quienes se encuentran suscritos a este servicio.
- Cartas de difusión directa (dirigidas a entidades públicas, educativas, organizaciones sociales, etc.)
- Notas de prensa (semanales)
- Espacios en periódicos de distribución nacional.
- Entrevistas en medios de comunicación oral.
- Exposiciones organizadas y coordinadas a solicitud de entidades y organizaciones sociales.
- Atención al público a través de una oficina especializada encargada de aclarar las dudas (relacionadas a los requisitos establecidos y a los términos y condiciones de las becas ofertadas) y, de guiar a los postulantes en la preparación de sus documentos.

La EGPP también realiza la difusión de becas que son canalizadas en forma directa, es decir, aquellas convocatorias para las cuales las postulaciones deben ser presentadas en forma directa a las representaciones diplomáticas o los organismos de cooperación que auspician los cursos ofrecidos. En este caso, los postulantes no necesitan cumplir con los requisitos establecidos por el Gobierno Boliviano.

## 1.2.2. GENEXUS

GeneXus es una herramienta de desarrollo de software ágil, multiplataforma, basada en conocimiento, orientada principalmente a aplicaciones web empresariales, plataformas Windows y dispositivos móviles o inteligentes. Es el producto principal de la compañía Uruguay Artech. El desarrollador describe sus aplicaciones en alto nivel, de manera mayormente declarativa, a partir de lo cual GeneXus genera código para múltiples plataformas (Windows, iSeries, Web, dispositivos móviles). [GeneXus, 2014]

GeneXus es una herramienta que parte de las “visiones de los usuarios”; captura su conocimiento y lo sistematiza en una base de conocimiento. A partir de su base de conocimiento, GeneXus es capaz de diseñar, generar y mantener de manera totalmente automática la estructura de la base de datos y los programas de la aplicación (los programas necesarios para que los usuarios puedan operar con sus visiones). [GeneXus International, 2012]

De forma genérica una aplicación comienza con un diseño, luego se prototipa, luego se implementa o pone en producción y en cualquiera de los pasos anteriores se puede regresar al diseño para realizar modificaciones. [GeneXus International, 2012]

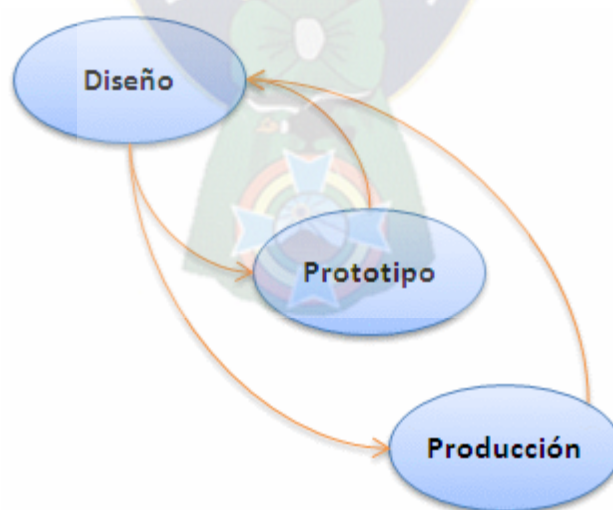


Figura 1.1.: Ciclos Diseño – Prototipación y diseño – Producción  
Fuente: [GeneXus International, 2012]

- **Diseño:** consiste en identificar y describir las visiones de datos de los usuarios. GeneXus captura el conocimiento por medio de visiones de objetos de la realidad del usuario. La tarea de diseño consiste, fundamentalmente, en identificar y describir estos objetos. A partir de estas descripciones, y automáticamente, GeneXus sistematiza el conocimiento capturado y va construyendo, en forma incremental, la Base de Conocimiento. [GeneXus International, 2012] Esta Base de Conocimiento es un repositorio único de toda la información del diseño, a partir de la cual GeneXus crea el modelo de datos físico (tablas, atributos, índices, redundancias, reglas de integridad referencial, etc.), y los programas de aplicación.
- **Prototipado:** Un prototipo GeneXus es una aplicación completa, funcionalmente equivalente a la aplicación de producción. La diferencia entre prototipación y producción consiste en que la primera se hace en un ambiente de microcomputador, mientras que la producción se realiza en el ambiente objeto del usuario (IBM iSeries, servidor Linux, Cliente / Servidor, JAVA, .NET, iOS, Android, BlackBerry etc. El prototipo permite que la aplicación sea totalmente probada antes de pasar a producción. [GeneXus International, 2012]
- **Implementación:** GeneXus genera automáticamente el código necesario para:
  - o Crear y mantener la base de datos;
  - o Generar y mantener los programas para manejar los objetos descritos por el usuario.
- **Mantenimiento:** El mantenimiento es a través de la misma herramienta tanto de la base de datos (estructura y contenido) como de los programas, es totalmente automático.
- **Documentación:** La documentación incluye la descripción de objetos específicos e información sobre la base de conocimiento resultante y sobre la base de datos diseñada.

### 1.2.3. SISTEMA WEB DE LA EGPP

El sistema web de la Escuela de Gestión Pública Plurinacional (EGPP) está elaborado bajo la herramienta de desarrollo Genexus. Este sistema está vigente desde el año 2011 (ver figura 1.2.), el cual está constituido por los siguientes módulos:

- Módulo de becas
- Módulo de contrataciones
- Módulo de tele educación
- Modulo institucional

← → ↻

**EGPP**  
Escuela de Gestión Pública Plurinacional  
MINISTERIO DE EDUCACIÓN

ESTADO PLURINACIONAL DE BOLIVIA

INICIO BECAS CONTRATACIONES TELE-EDUCACION INSTITUCIONAL FORMULARIO

**BECAS**

**BECAS**

- [Oferta de Becas](#)
- [Convocatorias Directas](#)

**AVISOS**

- [Contactos Unila](#)
- [UNILA Reinicio de Actividades](#)
- [UNILA Calendario Acad](#)
- [VENEZUELA Cambio de Carrera URGENTE](#)
- [VENEZUELA Suspensión proceso 2012](#)
- [UNILA SELECCIONADOS 2013](#)
- UNILA
  - [Preselección](#)
  - [Evaluación de habilidades específicas](#)
- MEXICO (Convocatoria)
  - [AFICHE MEXICO 2014](#)

**CONVOCATORIAS**

- EGIPTO (Cursos Cortos)
  - [Afiche Egipto](#)
  - [Formulario Centro Egipto](#)
- UNESCO
  - [Afiche UNESCO-ISED 2014](#)

**COMUNICADO**

**BOLETIN DEL MES**

**TELE EDUCACION**

**Enlaces Externos**

ABI MINISTERIO DE CULTURA SIC DES Bolívar MINISTERIO DE EDUCACIÓN Cambio

Calle Bolívar N°724 Esq. Indaburo - Casilla N° 12240  
Teléfonos 220-0131 - 220-0353 - 220-0322  
Fax: (591-2)220-0141  
La Paz - Bolivia

Figura 1.2.: Pantalla Principal del Sistema web EGPP  
Fuente: [Sistema web EGPP, 2014]

En la figura 1.3 podemos ver la pantalla por cual los postulantes pueden obtener información sobre las becas canalizadas por la EGPP.

www.egpp.gob.bo/Becas.html

**EGPP**  
Escuela de Gestión Pública Plurinacional  
MINISTERIO DE EDUCACIÓN

ESTADO PLURINACIONAL DE BOLIVIA

INICIO BECAS CONTRATACIONES TELE-EDUCACION INSTITUCIONAL FORMULARIO

**BECAS INFORMACION**

- [Como preparar su Beca Posterado](#)
- [Información general de Becas Posterado](#)
- [Como preparar su Beca Pregrado](#)
- [Información general de Becas Pregrado](#)
- [Reclamo de Revalidación v Homologación de Títulos Profesionales Extranjeros](#)

**BECAS DOCUMENTACION**

- [Informe Evento](#)
- [Propuesta Réplica](#)
- [Material Bibliográfico](#)
- [Formato de carta](#)
- [REQUISITOS PARA EL COMPROMISO](#)

UNIVERSITAS MAJOR PRINCENSIS DIVI ANDRE AE

Calle Bolívar N°724 Esq. Indaburo - Casilla N° 12240  
Teléfonos 220-0131 - 220-0353 - 220-0322  
Fax: (591-2)220-0141  
La Paz - Bolivia

Figura 1.3.: Pantalla de información sobre becas del módulo de becas  
Fuente: [Sistema web EGPP, 2014]

Los funcionarios de la EGPP del área de Becas a través del módulo de becas, realizan las siguientes actividades:

- Registro de información de convocatorias de becas: en esta actividad los funcionarios trabajan con información sobre correspondencia de becas, cooperaciones internacionales sus programas y cursos.

- Registro de información de interesados en postular a las becas: en esta actividad los funcionarios trabajan con información sobre unidades educativas, universidades, instituciones patrocinantes y postulantes.
- Los funcionarios elaboran informes de becas por medio de los reportes que genera el módulo de becas.

#### 1.2.4. TRABAJOS SIMILARES

Los siguientes trabajos que se presentan están relacionados con el presente proyecto de grado, estos son:

- [Copacondo R. J., 2009]. “Portal Web para el Arzobispado de Nuestra Señora de La Paz” en este trabajo se desarrolla un portal web aplicando la metodología Microsoft Solution Framework (MSF) para reemplazar el existente.
- [Yali N. F., 2011], ” Sistema de Control y Evaluación para el Proyecto C. D. I. – BO – 139” en este trabajo se desarrolla un sistema de información aplicando la metodología Programación extrema y el Lenguaje de Modelado Unificado, logrando automatizar los procesos manuales de dicha entidad.

#### 1.3. PLANTEAMIENTO DE PROBLEMA

En vista de lo anterior y en el ámbito del proyecto se identificaron los siguientes problemas:

El problema general:

¿De qué manera podemos mejorar el proceso de gestión y difusión de becas en la EGPP, que se realiza por medio del módulo de becas del sistema web EGPP?

Los problemas del módulo de becas del sistema web EGPP son los siguientes:

- Lentitud en el proceso de registro de convocatoria de becas y postulantes en el módulo de becas.



- Duplicidad en el ingreso de datos de personales del postulante al módulo de becas.
- Duplicidad de información de convocatorias e instituciones patrocinantes en la base de datos del módulo de becas.
- Altos tiempos de respuesta del módulo de becas.
- Personal no autorizado tiene acceso al módulo de becas.
- La información de becas puede ser fácilmente alterada.
- Duplicidad de archivos de información de becas en el repositorio del sistema web EGPP.
- Archivos de información de becas que se descargan del sistema web EGPP se guardan como historial.
- El sistema web EGPP está elaborado bajo un software propietario (Genexus) lo cual va en contra del decreto supremo 1874.
- El mantenimiento del sistema web EGPP no puede llevarse a cabo por la institución pues se desconoce el manejo de la herramienta Genexus.

#### 1.4. OBJETO DE ESTUDIO

El objeto de estudio de presente trabajo se centra en el proceso de gestión y difusión de la información de becas de la EGPP.

#### 1.5. OBJETIVOS

##### 1.5.1. OBJETIVO GENERAL

Mejorar el proceso de gestión y difusión de información de becas en la EGPP a través del desarrollo de un sistema web de gestión y difusión de becas (SwGDB), para reemplazar al módulo de becas del sistema web EGPP.

##### 1.5.2. OBJETIVOS ESPECIFICOS

- Desarrollar el sistema web de gestión y difusión de becas (SwGDB) de acuerdo al decreto supremo 1874.
- Permitir un único registro de datos personales de postulantes al SwGDB.

- Almacenamiento de información única de cooperaciones e instituciones patrocinantes en la base de datos del SwGDB.
- Ofrecer óptimos tiempos de respuesta del SwGDB.
- Realizar la normalización de la base de datos del Módulo de becas.
- Brindar un nivel de acceso al SwGDB para el personal autorizado.
- Controlar los archivos de información de becas en el repositorio del SwGDB.
- Crear un proceso de historial de descarga de archivos de información de becas en el SwGDB.
- Usar herramientas de software libre para la elaboración del SwGDB.
- Brindar información sobre el manejo de las herramientas software para el mantenimiento del SwGDB.

## 1.6. JUSTIFICACION

### 1.6.1. JUSTIFICACION ECONOMICA

El presente trabajo por tener como característica principal al mejoramiento del proceso de gestión y difusión de información de becas en la EGPP, nos ayudara a reducir el tiempo en estas actividades, logrando una solución en beneficios que en costos.

### 1.6.2. JUSTIFICACION TECNICA

La institución cuenta con ambientes propios, equipos computacionales y de comunicación adecuados para el desarrollo del sistema web. Además que se dará cumplimiento al Decreto Supremo N°1874 y a la disposición transitoria quinta del Decreto Supremo N° 1793 que dicta un plazo de 7 años para la migración de los sistemas en las entidades públicas a software libre y estándares abiertos. [Decreto Supremo 1793, 2014]

### 1.6.3. JUSTIFICACION SOCIAL

La implementación del presente proyecto mejorara las condiciones de trabajo de los funcionarios, como la calidad de servicio, el entorno de trabajo y la imagen de la institución frente a la sociedad.

## 1.7. LIMITES Y ALCANCES

Los límites y alcances que tendrá el sistema web de gestión y difusión de becas (SGDB) son los siguientes:

- El SGDB solo trabajara con información de convocatorias de becas canalizadas por la EGPP e información de los postulantes a las convocatorias de becas.
- El sistema SGDB no tendrá una conexión directa con el sistema web EGPP.
- El sistema SGDB cubrirá la función del módulo de becas del sistema web EGPP en la institución.
- El sistema SGDB tendrá una pantalla de acceso (usuario, clave).

## 1.8. METODOLOGIA, METODOS, TECNICAS Y HERRAMIENTAS

Los métodos para la elaboración del perfil del proyecto son:

- Árbol de problemas
- Árbol de objetivos
- Marco lógico

La metodología para el desarrollo del proyecto:

- Se utilizara la metodología Programación Extrema (XP) que define un conjunto de buenas prácticas para el desarrollo de software y nos ayuda a la entrega de software operativo en intervalos cortos de tiempo, adicionalmente haremos el uso del Lenguaje de Modelado Web (WEBML) para la documentación del sistema.
- Normalización de bases de datos: este método nos ayudara a evitar la duplicidad de información de becas en la base de datos del SwGDB y mejorar su diseño.

Las fuentes y técnicas de recolección de información:

- Fuentes primarias: existen muchas fuentes primarias para la recolección de datos, entre estas técnicas se tiene a la observación y las entrevistas.

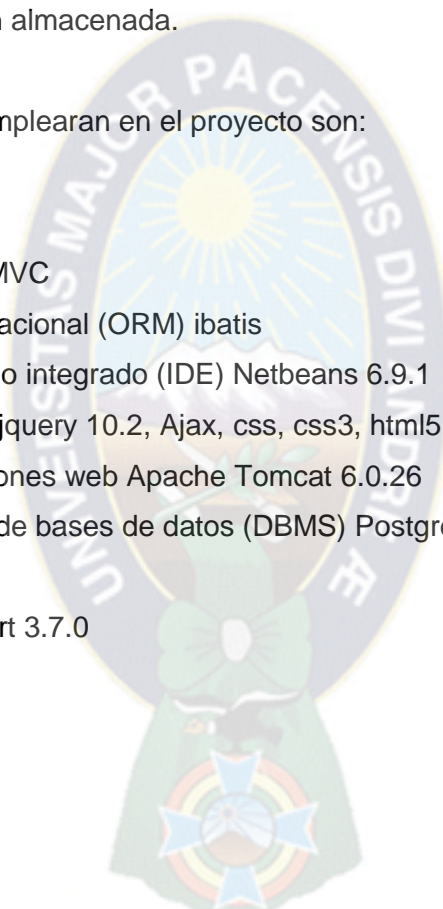
La observación: las personas lo usan todo el tiempo, pero para que esta tenga validez es necesaria que sea intencionada e ilustrada.

Las entrevistas: esta técnica se basa principalmente en la comunicación entre dos o más personas sobre ciertos aspectos de un determinado tema.

- Fuentes secundarias: entre este tipo de fuentes tenemos a los siguientes: los documentos, las revistas, artículos, libros, páginas web, estas nos brindan en su mayoría información almacenada.

Las herramientas que se emplearan en el proyecto son:

- Navicat Premium
- Framework Spring MVC
- Mapeo de objeto relacional (ORM) ibatis
- Entorno de desarrollo integrado (IDE) Netbeans 6.9.1
- Tecnologías cliente jquery 10.2, Ajax, css, css3, html5.
- Servidor de aplicaciones web Apache Tomcat 6.0.26
- Sistema de gestión de bases de datos (DBMS) Postgresql 9.3
- Ireport 3.7.0
- Librerías jasperreport 3.7.0



## CAPITULO II

### MARCO TEORICO

#### 2.1. INTRODUCCION

Los sistemas y aplicaciones basados en la web hacen posible que una población extensa de usuarios finales dispongan de una gran variedad de contenido y funcionalidad.

Para la buena comprensión del presente trabajo, en este apartado se realizara la descripción de las técnicas y herramientas utilizadas en la construcción de aplicaciones web, en nuestro caso en particular la metodología programación extrema, lenguaje de modelado web (WebML), conceptos de calidad de software, métricas de calidad, etc.

#### 2.2. INGENIERIA DE SOFTWARE

Entre las definiciones que tiene la Ingeniería de Software se puede mencionar las siguientes:

- ▲ Según Fritz Bauer: “La Ingeniería de Software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y trabaje con eficiencia en máquinas reales” [Pressman Roger, 2010, p.11]
- ▲ Una definición más completa dada por la IEEE: “La Ingeniería de Software es la aplicación de un enfoque sistemático disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la Ingeniería de Software” [Pressman Roger, 2010, p.11]

La ingeniería de software es el área de conocimiento relacionada con la producción de software, las formas prácticas para desarrollar y entregar un software útil.

Por otro lado, la ingeniería de software es una tecnología con varias capas. El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permiten el desarrollo racional y oportuno del software de cómputo.

El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

## 2.3. METODOLOGIAS DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas software.

El marco de trabajo de una metodología de desarrollo consiste en:

- ▲ Una filosofía de desarrollo de software, con el enfoque o enfoques del proceso de desarrollo de software.
- ▲ Múltiples herramientas, modelos y métodos para ayudar en el proceso de desarrollo de software.

Las metodologías de desarrollo se dividen en dos grupos como son: metodologías tradicionales y metodologías ágiles.

### 2.3.1. METODOLOGIAS TRADICIONALES

Las metodologías tradicionales están orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usaran. Estas consideran la importancia de documentar exhaustivamente el proyecto software.

### 2.3.2. METODOLOGIAS AGILES

Las Metodologías reconocidas como ágiles actualmente se sustentan (en menor o mayor proporción) en el manifiesto por el desarrollo ágil de software que consta de 4 valores de los cuales se desprenden 12 principios, [Pressman Roger, 2010], este se traduce en una filosofía que incluye formas de pensar y trabajar.

El Manifiesto valora:

- ⤴ Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- ⤴ Desarrollar software que funciona más que conseguir una buena documentación.
- ⤴ La colaboración con el cliente más que la negociación de un contrato.
- ⤴ Responder a los cambios más que seguir estrictamente un plan.

Bajo esta filosofía el desarrollo ágil se traduce en un proceso de desarrollo iterativo e incremental que minimiza los riesgos por medio de iteraciones cortas. Se disminuyen considerablemente los trámites burocráticos, creando de esta forma, una rapidez en la realización y entrega del software que se usa.

#### 2.4. PROGRAMACION EXTREMA (XP)

La Programación Extrema (XP), se distingue de otras por incluir prácticas y recomendaciones técnicas, también pone un fuerte énfasis en la adaptabilidad que en la previsibilidad; considera que los cambios en los requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable con lo cual nos permite lograr mejores productos. XP toma en cuenta los siguientes valores: simplicidad, comunicación, retroalimentación, coraje y respeto. [Letelier P. y Penadés C., 2004]

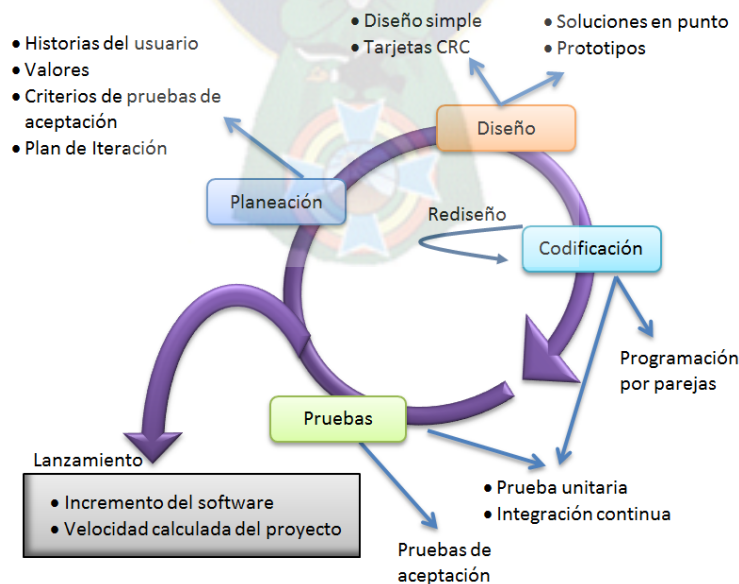


Figura 2.1.: Proceso XP  
Fuente: [Pressman, 2010]

XP usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. La figura 2.1 ilustra el proceso de XP con cuatro actividades estructurales. [Pressman, 2010]

#### 2.4.1. PLANEACION

La planeación comienza escuchando al cliente, actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieran. Escuchar lleva a la creación de algunas historias de usuario que describen la salida necesaria, características y funcionalidad del software que se va a elaborar.

##### a) HISTORIAS DE USUARIO

Las historias de usuario son la técnica utilizada en XP para describir la salida necesaria, características y funcionalidad que será útil para el usuario o comprador, de un sistema software. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Es aconsejable usar el formato: “como (rol) quiero (funcionalidad) para (beneficio)”, figura 2.2.

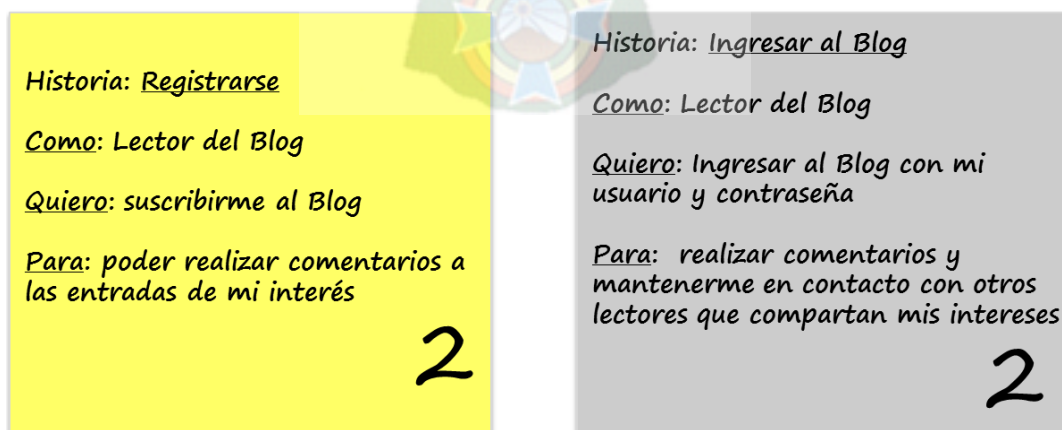


Figura 2.2.: Historias de usuario  
Fuente: [Salazar Lucho, 2013]



## b) PLAN DE ENTREGAS

El cronograma de entregas establece que historias de usuario serán agrupadas para conformar una entrega y el orden de las mismas.

## c) TAREAS

Cada historia de usuario se traduce en tareas específicas de programación.

## d) VELOCIDAD DEL PROYECTO

Después de la primera entrega del proyecto (también llamada incremento de software), el equipo XP calcula la velocidad de este. En pocas palabras, la velocidad del proyecto es el número de historias de los clientes implementadas durante la primera entrega.

La velocidad del proyecto se usa para: ayuda a estimar las fechas de entrega y programar las actividades para las entregas posteriores y determinar si se ha hecho un gran compromiso para todas las historias durante todo el desarrollo del proyecto. Si esto ocurre, se modifica el contenido de las entregas o se cambian las fechas al final.

## e) ENTREGAS PEQUEÑAS

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero sí que constituyan un resultado de valor para el negocio.

## f) REUNIONES

El objetivo de tener reuniones es mantener la comunicación entre el equipo, y compartir problemas y soluciones.

## g) ROLES

Se utiliza el concepto de roles para organizar quienes se encargaran de cada una de las actividades que deben realizarse durante el transcurso del proyecto. Cada uno de estos papeles es desempeñado por uno o varios integrantes del grupo.

- ▲ Programador: el programador escribe las pruebas unitarias y produce el código del sistema, debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- ▲ Cliente: el cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- ▲ Encargado de pruebas (*Tester*): el encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- ▲ Encargado de seguimiento (*Tracker*): el encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- ▲ Entrenador (*Coach*): es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- ▲ Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- ▲ Gestor (*Big boss*): es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### 2.4.2. DISEÑO

El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la

implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después.

#### a) DISEÑO SIMPLE

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.

#### b) METAFORA

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema, consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

#### c) TARJETAS CRC

XP estimula el uso de tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Las tarjetas CRC son el único producto del trabajo de diseño que se genera como parte del proceso XP.

|                     |               |
|---------------------|---------------|
| Nombre de la clase. |               |
| Responsabilidades   | Colaboradores |

Figura 2.3.: Tarjeta CRC  
Fuente: [Ablett Erick, 2013]

- ⤴ Clase: cualquier persona, cosa, evento, concepto, pantalla o reporte
- ⤴ Responsabilidades: son las cosas que conocen y las que realizan, sus atributos y métodos.
- ⤴ Colaboradores: son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

#### d) SOLUCIONES EN PUNTO

Si en diseño de una historia se encuentra un problema de diseño difícil, XP recomienda la creación inmediata de un prototipo operativo de esa porción del diseño. Entonces, se implementa y evalúa el prototipo del diseño, llamado solución en punto. El objetivo es disminuir el riesgo cuando comience la implementación verdadera y validar las estimaciones originales para la historia que contiene el problema de diseño.

#### e) REFACTORIZACION

La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo.

### 2.5.3. CODIFICACION

Después de que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas a cada una de las historias que se van a incluir en la entrega en curso (incremento software).

#### a) PRIMERO LA PRUEBA

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

## b) PROGRAMACIÓN EN PAREJAS

XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. Esto da un mecanismo para una solución de problemas en tiempo real (es frecuente que dos cabezas piensen más que una) y para el aseguramiento de calidad también en tiempo real (el código se revisa conforme se crea). También mantiene a los desarrolladores centrados en el problema de que se trate. En la práctica cada persona toma un papel un poco diferente.

## c) PROPIEDAD COLECTIVA DEL CODIGO

Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

## d) ESTÁNDAR DE PROGRAMACIÓN

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

## e) INTEGRACION CONTINUA

Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.

#### 2.4.4. PRUEBAS

XP enfatiza mucho en los aspectos relacionados con las pruebas, clasificándolas en diferentes tipos y funcionalidades específicas, indicando quien, cuando y como deben ser implementadas y ejecutadas.

##### a) PRUEBAS UNITARIAS

Estas pruebas se aplican a todos los métodos no triviales de todas las clases del proyecto con la condición de que no se liberara ninguna clase que no tenga asociada su correspondiente paquete de pruebas.

##### b) PRUEBAS DE ACEPTACION

Las pruebas de aceptación, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales de los sistemas que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software.

#### 2.5. BASES DE DATOS

Es un conjunto de datos estructurados almacenados en un soporte legible por ordenador y organizadas internamente por registros (formado por todos los campos referidos a una entidad u objeto almacenado) y campos (cada uno de los elementos que componen un registro). Permite recuperar cualquier clase de información: referencias, documentos textuales, imágenes, datos estadísticos, etc.

##### 2.5.1. NORMALIZACION DE BASES DE DATOS

En los modelos de base de datos relacionales, la normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener.

La normalización de una base de datos es un proceso con tres objetivos básicos:

- ▲ Asegurar la integridad de datos.
- ▲ Eliminar los datos redundantes
- ▲ Hacer los sistemas más eficientes.

También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. Entre las ventajas de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco. El proceso de normalización tiene un nombre y una serie de reglas para cada fase.

Grados de normalización: tenemos los siguientes niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización.

En la tabla siguiente se describe brevemente en que consiste cada una de las reglas, y posteriormente se explican con más detalle.

| Regla                      | Descripción  |
|----------------------------|--|
| Primera Forma Normal (1FN) | Incluye la eliminación de todos los grupos repetidos.  |
| Segunda Forma Normal (2FN) | Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).   |
| Tercera Forma Normal (3FN) | Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. |

Tabla 2.1.: Formas normales  
Fuente: [Normalización de bases de datos, 2003]

## 2.6. WEBML

WebML es un lenguaje de modelado gráfico utilizado para apoyar las actividades de diseño de sitios web a partir de su descripción desde distintos puntos de vista como son el conceptual, el navegacional y el de presentación entre otros [Escalona M J., 2004]. WebML está orientado para aplicaciones que tienen una alta interacción de datos, las diferentes fases de su ciclo de vida se muestran en la figura 2.4.

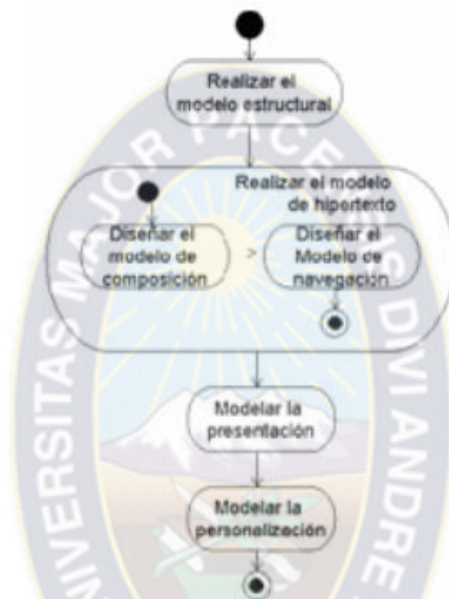


Figura 2.4.: Ciclo de vida de WebML  
Fuente: [Escalona M J., 2004]

### 2.6.1. MODELO ESTRUCTURAL

Cuando se trabaja con WebML el proceso de desarrollo comienza con la descripción conceptual del sistema, en la cual, utilizando herramientas CASE para modelado, como UML, DIA, Enterprise Architect, se representa la estructura estática del sistema, mediante la definición de entidades o contenedores de datos y sus relaciones. Una característica a destacar de WebML es que no exige ninguna herramienta específica para hacer este modelo.

Se puede utilizar:

- ♣ Diagramas de Entidad-Relación (E-R) que muestran todas las tablas, los diferentes campos de cada tabla, y las relaciones entre ellas.



- ⤴ Diagramas UML de clases que pueden representar la misma información que un diagrama de Entidad -Relación (por lo que se puede usar de manera equivalente), e incluso información adicional sobre el modelo de datos.

## 2.6.2. MODELO DE HIPERTEXTO

En este modelo se describen los diferentes hipertextos que van a ser publicados en el sitio Web. Cada uno de estos define una vista del sitio y su descripción se realiza mediante los modelos de composición y de navegación, como se puede apreciar en la figura 2.5.

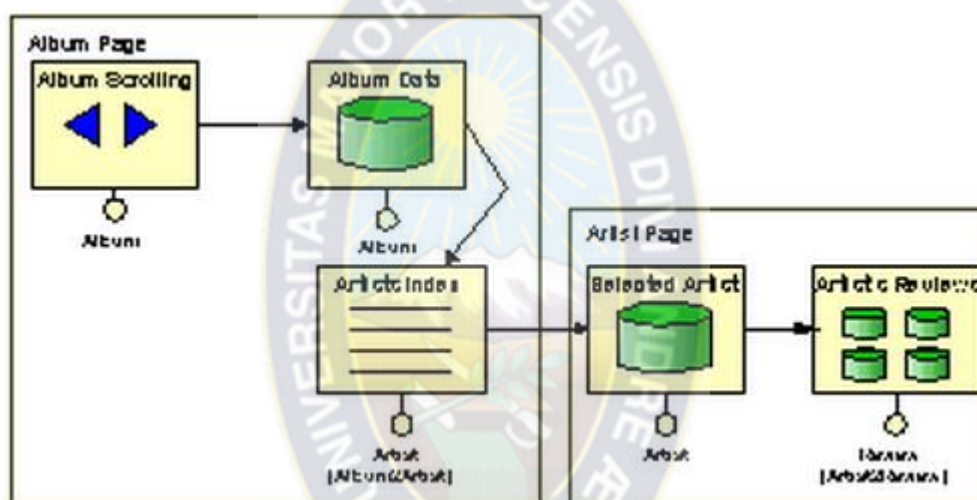


Figura 2.5.: Modelado de hipertexto  
Fuente: [WebML staff, 2004]

### 2.6.2.1. MODELO DE COMPOSICION

Muestra las unidades de contenido que pueden aparecer en el sitio web y las páginas mediante las cuales se agrupa la información para ser entregada al usuario.

WebML soporta siete tipos de unidades básicas:

- ⤴ Unidades de datos: muestran información de un solo objeto, son definidas para seleccionar una mezcla de información, su representación gráfica se puede apreciar en la figura 2.6.

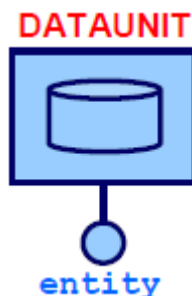


Figura 2.6.: Notación grafica WebML para Unidad de Datos  
Fuente: [WebML staff, 2004]

- Unidad multidatos: muestra información sobre un conjunto de objetos, presenta múltiples instancias de una entidad o componente. Tiene dos partes: el contenedor que incluye las instancias que se desea mostrar y la unidad de datos usada para la presentación de cada instancia, su representación gráfica se puede apreciar en la figura 2.7.

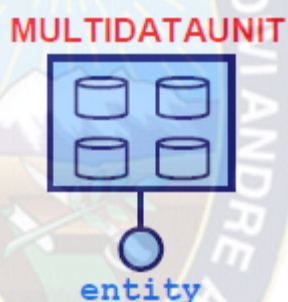


Figura 2.7.: Notación grafica WebML para Unidad Multidatos  
Fuente: [WebML staff, 2004]

- Unidades Índice (y sus variantes de múltiple opción y jerárquicos): presenta múltiples instancias de una unidad o componente como una lista, esta unidad tiene dos partes principales: el contenedor que se incluye las instancias que se desea mostrar y los atributos usados como clave del índice, su representación gráfica en la figura 2.8.



Figura 2.8.: Notación grafica WebML para Unidades Índice  
Fuente: [WebML staff, 2004]

- Unidad de desplazamiento: provee comandos para desplazarse a través de los objetos en un contenedor, su representación gráfica se puede apreciar en la figura 2.9.

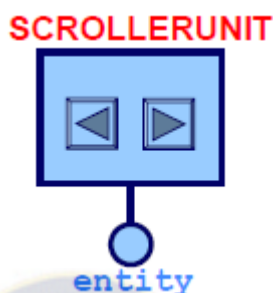


Figura 2.9.: Notación grafica WebML para Unidad de Desplazamiento  
Fuente: [WebML staff, 2004]

- Unidades de Entrada: representa un formulario y la entrada de datos a los campos, su representación gráfica se puede apreciar en la figura 2.10.



Figura 2.10.: Notación grafica WebML para Unidades de Entrada  
Fuente: [WebML staff, 2004]

- Unidad de Múltiple Opción: es una variante de la unidad de índice, donde cada elemento de la lista está asociada a una casilla de verificación que permite al usuario seleccionar varios objetos, su representación gráfica se puede ver en la figura 2.11.



Figura 2.11.: Notación grafica WebML para Unidad de Múltiple Opción  
Fuente: [WebML staff, 2004]

- ▲ Unidad Jerárquica: es una variante de la unidad de índice, las entradas del índice están organizadas en un árbol multinivel, su representación gráfica se puede apreciar en la figura 2.12.



Figura 2.12.: Notación gráfica WebML para Unidad Jerárquica  
Fuente: [WebML staff, 2004]

#### 2.6.2.2. MODELO DE NAVEGACION

Tiene como propósito especificar la forma como las unidades y páginas estarán conectadas para formar una estructura de hipertexto con significado y para definir el flujo de información necesaria para procesar las unidades y operaciones. Para alcanzar este propósito WebML cuenta con la noción de Link el cual consiste en una conexión directa entre unidades o páginas.

En WebML se definen dos tipos de links: no-contextuales y contextuales. Los primeros se establecen entre páginas y en ellos no se transmite ningún tipo de información de contexto, mientras que los links contextuales permiten la transmisión de información de contexto entre unidades fuente y destino.

#### 2.6.3. MODELO DE PRESENTACION

En esta fase se define claramente la apariencia gráfica de cada una de las páginas que conformarán el proyecto. WebML no incluye un modelo específico para establecer la presentación a nivel conceptual.

#### 2.6.4. MODELO DE PERSONALIZACION

Se encarga de definir cómo debe adaptarse el sistema a los diferentes roles de usuario. Este modelo tiene tres facetas:

- ⤴ Control de acceso: login/logout operaciones para el reconocimiento de usuario.
- ⤴ Asignación de Vista de Sitio: basado en el grupo al que pertenece el usuario.
- ⤴ Personalización de la Página: usuario y grupos dependientes del contenido.

En principio un grupo de usuarios es definido para todos (default) y estos no necesitan login. Para el ingreso al sitio se tiene grupos de usuarios con login, cada grupo está asociado a vistas de sitio.

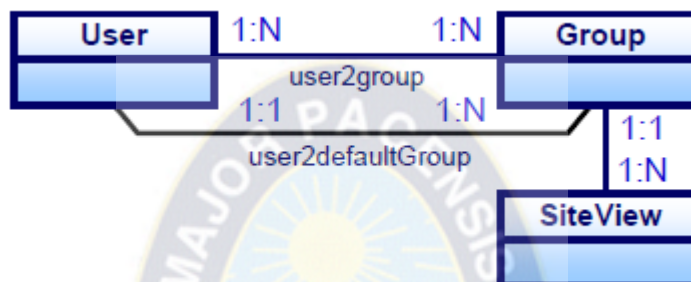


Figura 2.13.: Modelo Usuario/Grupo  
Fuente: [WebML staff, 2004]

## 2.7. SOFTWARE SEGURO

La criticidad de los sistemas software en diferentes dominios de la sociedad determina que si bien es importante a que el software se desarrolle de acuerdo a las necesidades del cliente, también lo es garantizar que el mismo lo sea seguro [Castellaro M, et al., 2009].

Las siguientes son propiedades fundamentales de un software seguro:

- ⤴ Confidencialidad: la propiedad de que cierta información no esté disponible para ciertas entidades.
- ⤴ Integridad: la información no debe poder ser alterada o manipulada.
- ⤴ Auditoria: un registro cronológico de los eventos relevantes a la seguridad de un sistema. este registro luego examinarse para reconstruir un escenario en particular.
- ⤴ Autenticación: el proceso de verificar la identidad de una entidad.
- ⤴ Control de acceso: el proceso de regular las clases de acceso que una entidad tiene sobre sobre los recursos.
- ⤴ Disponibilidad: la propiedad de que el sistema sea accesible a las entidades autorizadas.

- ▲ No repudio: la habilidad de prevenir que una entidad desmienta o niegue la responsabilidad de acciones que han sido ejecutadas. Asegura que no pueda eludir la propiedad de auditoria.

Los efectos de vulnerar la seguridad del software se pueden describir en términos de los efectos sobre estas propiedades fundamentales.

### 2.7.1. CRIPTOGRAFIA

La criptografía responde a los problemas básicos de seguridad en la transmisión de información por un canal inseguro. El esquema fundamental de un proceso criptográfico (cifrado/descifrado) puede resumirse en la figura 2.14.



Figura 2.14.: Proceso criptográfico  
Fuente: [Jimenez S. Jose C., 2001]

#### 2.7.1.1. ENCRIPCIÓN CON MD5

En criptografía, MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo que proporciona un código asociado a un archivo o texto concretos, MD5 es ampliamente usado en la encriptación de contraseñas restringiendo de esta manera el ingreso a los usuarios no autorizados [MD5, 2014].

## 2.8. CALIDAD DEL SOFTWARE

La calidad de software se resume en su totalidad en la satisfacción del producto software en su conjunto por parte del cliente y usuarios finales, pues si el cliente y usuarios finales no están satisfechos nada de los demás importa. Por otro lado el software puede ser evaluado bajo la norma ISO – 9126 la cual identifica seis atributos clave del software como son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos software.

A continuación se detalla cada una de las características que se establecen en el estándar ISO – 9126.

- ▲ Funcionalidad: grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- ▲ Confiabilidad: cantidad de tiempo que es software se encuentra disponible para su uso según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
- ▲ Usabilidad: grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.
- ▲ Eficiencia: grado en el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento en el tiempo y comportamientos de los recursos.
- ▲ Mantenibilidad: facilidad con la que pueden efectuarse reparaciones al software según lo indican los atributos siguientes: analizable, cambiable, estable y susceptible de someterse a pruebas.
- ▲ Portabilidad: facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

### 2.8.1. METRICAS DE CALIDAD

Una métrica en el contexto de desarrollo de software es una medida cuantitativa de un determinado atributo que posee un sistema software.

A continuación mencionaremos las diferentes métricas bajo las características identificadas por la ISO – 9126.

## FUNCIONALIDAD

La métrica de punto de función (PF) se usa de manera efectiva como medio para medir la funcionalidad que entra a un sistema software. Para efectivizar esta medida consideramos lo siguiente:

- ▲ Número de entradas de usuario
- ▲ Número de salidas de usuario
- ▲ Número de consultas/peticiones de usuario
- ▲ Numero de archivos lógicos internos
- ▲ Numero de interfaces externas

| Parámetros de medición          | Cuenta               | Factor de ponderación |       |          | = | Cuenta               |
|---------------------------------|----------------------|-----------------------|-------|----------|---|----------------------|
|                                 |                      | Simple                | Medio | Complejo |   |                      |
| Número de entradas de usuario   | <input type="text"/> | x 3                   | 4     | 6        | = | <input type="text"/> |
| Número de salidas de usuario    | <input type="text"/> | x 4                   | 5     | 7        | = | <input type="text"/> |
| Número de peticiones de usuario | <input type="text"/> | x 3                   | 4     | 6        | = | <input type="text"/> |
| Número de archivos              | <input type="text"/> | x 7                   | 10    | 15       | = | <input type="text"/> |
| Número de interfaces externas   | <input type="text"/> | x 5                   | 7     | 10       | = | <input type="text"/> |
| Cuenta total                    | →                    |                       |       |          | = | <input type="text"/> |

Figura 2.15.: Calculo de puntos de función

Fuente: [Pressman, 2010]

La cuenta total debe ajustarse utilizando la siguiente ecuación:

$$PF = \text{cuenta total} \times (0.65 + 0.01 \times \sum (Fi))$$

Donde cuenta total es la suma de todas las entradas PF obtenidas en la figura 2.2 y los  $F_i$  ( $i=1$  a 14) son factores de ajuste de valor (FAV) con base en las respuestas a las siguientes preguntas (tabla 2.1):



| #  | FACTORES DE AJUSTE DE VALOR  |
|----|--|
| 1  | ¿El sistema requiere respaldo y recuperación confiables?   |
| 2  | ¿Se requiere comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?               |
| 3  | ¿Existen funciones de procesamiento distribuidas?  |
| 4  | ¿El desempeño es crucial?  |
| 5  | ¿El sistema correrá en un entorno operativo existente y enormemente utilizado?   |
| 6  | ¿El sistema requiere entrada de datos en línea?  |
| 7  | ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones? |
| 8  | ¿Los archivos lógicos internos se actualizan en línea?   |
| 9  | ¿Las entradas, salidas, los archivos o consultas son complejos?  |
| 10 | ¿El procesamiento interno es complejo?   |
| 11 | ¿El código se diseña para ser reutilizable?  |
| 12 | ¿La conversión y la instalación se incluyen en el diseño?  |
| 13 | ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?   |
| 14 | ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?  |

Tabla 2.2.: Factor de ajuste de valor  
Fuente: [Pressman, 2010]

## CONFIABILIDAD

Considerando un sistema software, una medida sencilla de su confiabilidad es el tiempo medio entre fallas (TMEF) [Pressman, 2010]:

$$\text{TMEF} = \text{TMPF} + \text{TMPR}$$

Donde:

TMPF es el tiempo medio para la falla

TMPR es el tiempo medio para la reparación

Además de la medida de confiabilidad, también se tiene otra para la disponibilidad del software la cual se define como:

$$\text{Disponibilidad} = \text{TMPF} / (\text{TMPF} + \text{TMPR}) \times 100\%$$

## USABILIDAD

La forma de determinar si existe usabilidad en un sistema software es evaluarla o probarla. Los usuarios interactúan con el sistema y responden a las siguientes preguntas.

| PREGUNTAS SOBRE LA PERCEPCION DE USABILIDAD   |
|---|
| ¿El sistema es utilizable sin ayuda o enseñanza continúa?                                     |
| ¿Las reglas de interacción ayudan a un usuario preparado a trabajar con eficiencia?           |
| ¿Los mecanismos de interacción se hacen más flexibles a medida que los usuarios conocen más?  |
| ¿Se ha adaptado el sistema al ambiente físico y social en el que se usara?                    |
| ¿El usuario está al tanto del estado del sistema? ¿Sabe en todo momento en donde esta?        |
| ¿La interfaz está estructurada de manera de manera lógica y consistente?                      |
| ¿Los mecanismos, iconos y procedimientos de interacción son consistentes en toda la interfaz? |
| ¿La interacción prevé errores y ayuda el usuario a corregirlos?                               |
| ¿La interfaz es tolerante a los errores que se cometen?                                       |
| ¿Es sencilla la interacción?  |

Tabla 2.3.: Preguntas de usabilidad  
Figura: [Pressman, 2010]

## MANTENIBILIDAD

Como una medida de mantenimiento se considera la facilidad que tiene el software para recibir mantenimiento extraído del TMPR vista en la confiabilidad.

Se sugiere un índice de madurez de software (IMS) que proporcione un indicio de estabilidad de un producto software (con base a los cambios que ocurra con cada liberación del producto) para ello se determina la siguiente información [Pressman, 2010]:

Mt: número de módulos en la liberación actual

Fc: número de módulos en la liberación actual que cambiaron

Fa: número de módulos en la liberación actual que se agregaron

Fd: número de módulos en la liberación anterior que se borraron en la liberación actual

El índice de madurez del software se calcula de la siguiente forma:

$$\text{IMS} = \text{Mt} - (\text{Fa} + \text{Fc} + \text{Fd}) / \text{Mt}$$

Conforme el IMS tiende a 1.0 el producto comienza a estabilizarse. El IMS también puede usarse como una métrica para planificar la actividad de mantenimiento de software.



## CAPITULO III

### MARCO APLICATIVO

#### 3.1. INTRODUCCION

En este apartado veremos el desarrollo del presente trabajo, para este propósito se utiliza a la Programación Extrema (XP) como metodología de desarrollo y al Lenguaje de Modelado Web (WebML) para el modelado del sistema, este último nos ayudara a ampliar la descripción del proyecto para su comprensión.

#### 3.2. ESTUDIO DE LA SITUACION ACTUAL

La Escuela de Gestión Pública Plurinacional (EGPP) por medio de la Unidad de Becas está encargada en representación del Gobierno Boliviano de administrar las ofertas de becas de la cooperación internacional (país u organismo ofertante). Estas ofertas de becas se traducen en convocatorias de becas las cuales se dividen en:

- ▲ Convocatorias canalizadas: estas convocatorias están dirigidas a los ciudadanos bolivianos que radican en Bolivia, la Unidad de Becas administra la información referente a estas convocatorias y a sus postulantes.
- ▲ Convocatoria directa: la Unidad de Becas administra estas convocatorias solo para su difusión.

La EGPP cuenta con un sistema web en el cual se encuentra integrado un módulo para la gestión y difusión de becas (módulo de becas). El módulo de becas procesa la información de:

- ▲ cooperaciones internacionales (canalizadas y directas)
- ▲ programas de cooperaciones canalizadas
- ▲ correspondencias de convocatorias
- ▲ convocatorias (canalizadas y directas)
- ▲ instituciones (Unidades Educativas, Universidades, instituciones Públicas, instituciones Privadas) patrocinadoras de postulantes.
- ▲ postulaciones.

Los funcionarios que trabajan en la Unidad de Becas y directamente con el módulo de becas realizan actividades específicas (consulta, registro, modificación y eliminación sobre los elementos de la tabla 3.1), por otro lado también tenemos al postulante que usa el sistema web EGPP como medio de consulta de información sobre las convocatorias de becas.

En la tabla 3.1 se tiene la relación de información entorno a las convocatorias.

|  | Convocatoria canalizada | Convocatoria directa |
|--|-------------------------|----------------------|
| Cooperación canalizada                   | X                       |                      |
| Cooperación directa                      |                         | X                    |
| Programa de cooperación                  | X                       |                      |
| Correspondencia                          | X                       | X                    |
| Institución patrocinadora del postulante | X                       |                      |
| Postulante                               | X                       |                      |

Tabla 3.1: Relación de información entorno a las convocatorias

El análisis y evaluación al módulo de becas, ha podido constatar las limitaciones que traen consigo la insatisfacción de las personas que trabajan con el módulo de becas.

### 3.2.1. ORGANIGRAMA INSTITUCIONAL

En la figura 3.1 se muestra el Organigrama Institucional general correspondiente a la Escuela de Gestión Pública Plurinacional (EGPP).

En la actualidad la Unidad de Becas de la EGPP es la encargada de administrar la información referente a la oferta de becas de las cooperaciones internacionales, para esto se tienen procesos establecidos para realizar dicha administración. El flujo de información comienza con la recepción de documentación física, específicamente cuando esta se trata de convocatorias y postulaciones de becas en la EGPP la cual es derivada a la Unidad de Becas, el equipo técnico de dicha unidad comienza con el registro de esta documentación en secretaria y luego pasa por el encargado de difusión.

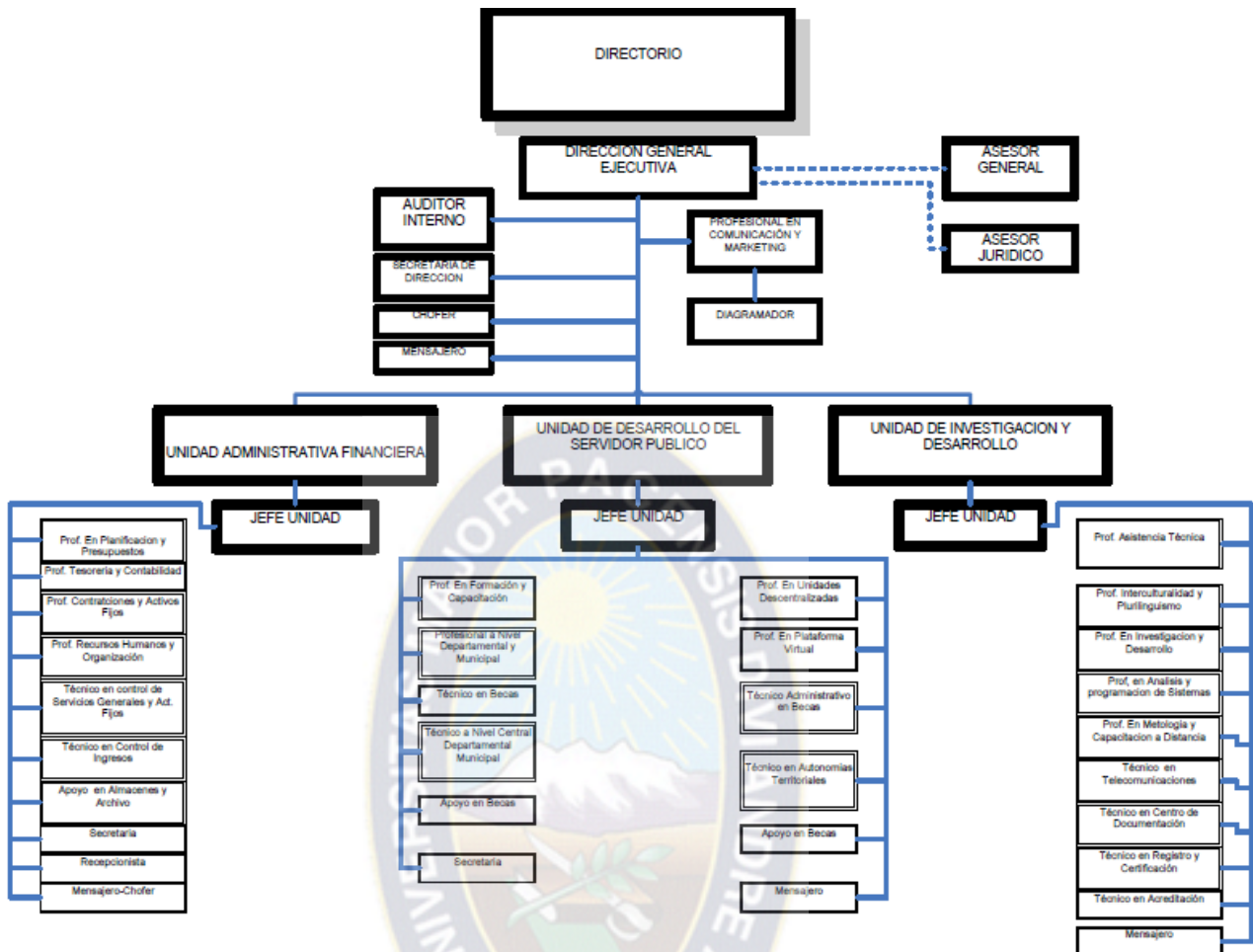


Figura 3.1: Organigrama EGPP  
Fuente: [Sistema web EGPP, 2014]

### 3.3. DESARROLLO DEL SISTEMA WEB

Es esta sección veremos la aplicación de la Programación Extrema (XP) la cual consta de actividades estructurales como ser: planeación, diseño, codificación y pruebas.

En las actividades de diseño y codificación se incluirán a los modelos WebML, coadyuvando de esta manera a una mejor descripción del proyecto.

### 3.3.1. PLANEACION

#### 3.3.1.1. EQUIPO DE TRABAJO

La estructura organizacional de la Unidad de Becas se encuentra representada por el Rol cliente y sus responsables (tabla 3.1). A continuación en la tabla 3.1 se muestra los integrantes del equipo de trabajo en base a los roles para el proyecto:

| ROL                  | RESPONSABLE            | DESCRIPCION  |
|----------------------|------------------------|--|
| Jefe de proyecto     | Ing. Sergio Sanchez    | Responsable del proyecto   |
| Cliente              | Lic. Guillermo Delgado | Jefe de la Unidad de Becas   |
|                      | Lic. Karina Villarroel | Difusión de becas  |
|                      | Mery Mamani            | Información de Becas   |
|                      | Emiliana Ilaya         | Seguimiento exbecarios   |
|                      | Mabel Pañagua          | Secretaria   |
| Equipo de desarrollo | Richar Marquez         | Coordinar el desarrollo del proyecto (análisis, diseño e implementación de las historias de usuario) |

Tabla 3.2: Equipo de trabajo

#### 3.3.1.2. HISTORIAS DE USUARIO

Considerando que nuestro objetivo general es el de reemplazar y mejorar la funcionalidad del módulo de becas, para la obtención de la historias de usuario se realizó un trabajo de recuperación (abstracción) de información de la base de datos, interfaces y procesamiento del módulo de becas conjuntamente con el equipo de trabajo, a continuación describimos este proceso:

- a) Para la recuperación de información de la base de datos existente se utilizó a la herramienta Navicat Premium que es un administrador de bases de datos gráfico, a través de este se estableció una conexión local con la base de datos existente. Con esta herramienta logramos la comprensión de esta base de datos (tablas, atributos, relaciones, llaves primarias, llaves foráneas, tipos de datos, etc.) y se procedió a su normalización.

b) En las interfaces existentes se identificó la estructura y comportamiento del módulo de becas, lo cual nos llevó a analizar lo siguiente:

- ✦ Acciones básicas que procesa las diferentes interfaces (registro, modificación, eliminación, presentación de información).
- ✦ respuesta del comportamiento del sistema a dichas acciones (validación de datos, mensajes de alerta)

c) Para la recuperación del procesamiento del módulo de becas se interactuó de manera directa con el módulo de becas observando y analizando los datos procesados y recuperados a través de la interfaz y base de datos existentes. Los procesos visibles que se observó con los siguientes:

- ✦ Cargado de datos en los formularios (campos select dependientes)
- ✦ Validación de datos en el registro de formularios
- ✦ Generación de código de convocatorias
- ✦ Inferencia de datos (resultados almacenados)
- ✦ Registro de descargas.
- ✦ Emisión de reportes.

En la figura 3.2 se muestra el proceso dinámico en la recuperación de información que se describió. Estableciendo de esta manera las historias de usuario la cuales rescatan la funcionalidad del módulo de becas y su mejoramiento.

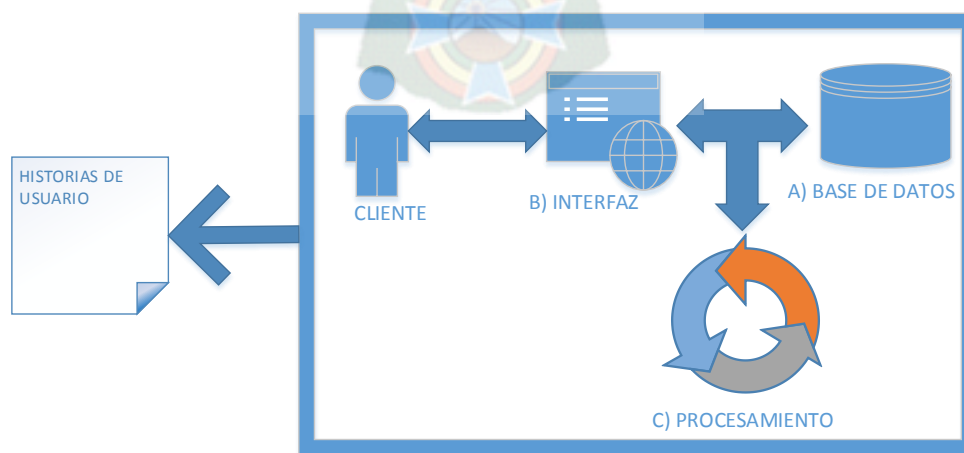


Figura 3.2: Recuperación de Información



Tras el trabajo de recuperación de información del módulo de becas se llegan a establecer las siguientes historias de usuario las cuales se muestra de manera general en la tabla 3.3.

| NRO | HISTORIA DE USUARIO                  | TAREAS   |
|-----|--------------------------------------|--|
| 1   | Registro de cooperación              | 1.1: registro de cooperación                       |
|     |                                      | 1.2: información de cooperación                    |
|     |                                      | 1.3: registro de contacto de cooperación           |
|     |                                      | 1.4: información de contacto de cooperación        |
| 2   | Registro de programa cooperación     | 2.1: registro de programa de cooperación           |
|     |                                      | 2.2: información de programa de cooperación        |
| 3   | Registro de correspondencia          | 3.1: registro de correspondencia                   |
|     |                                      | 3.2: información de correspondencia                |
| 4   | Registro de convocatoria             | 4.1: registro de convocatoria directa              |
|     |                                      | 4.2: información de convocatoria directa           |
|     |                                      | 4.3: registro de convocatoria canalizada           |
|     |                                      | 4.4: información de convocatoria canalizada        |
| 5   | Ingreso de usuario                   | 5.1: interfaz principal                            |
|     |                                      | 5.2: registro de usuario                           |
|     |                                      | 5.3: administración de usuario                     |
|     |                                      | 5.4: ingreso de usuario                            |
|     |                                      | 5.5: historial de usuario                          |
| 6   | Registro de patrocinador             | 6.1: registro de unidad educativa patrocinadora    |
|     |                                      | 6.2: Información de unidad educativa patrocinadora |
|     |                                      | 6.3: registro de contacto de unidad educativa      |
|     |                                      | 6.4: información de contacto de unidad educativa   |
|     |                                      | 6.5: registro de universidad patrocinadora         |
|     |                                      | 6.6: información de universidad patrocinadora      |
|     |                                      | 6.7: registro de contacto de universidad           |
|     |                                      | 6.8: información de contacto de universidad        |
|     |                                      | 6.9: registro de institución patrocinadora         |
|     |                                      | 6.10: información de institución patrocinadora     |
|     |                                      | 6.11: registro de contacto de institución          |
|     |                                      | 6.12: información de contacto de institución       |
| 7   | Registro de postulante               | 7.1: registro de postulante                        |
|     |                                      | 7.2: Información de postulante                     |
|     |                                      | 7.3: registro de contactos de emergencia           |
|     |                                      | 7.4: información de contactos emergencia           |
| 8   | Registro de formación del postulante | 8.1: registro de trabajo                           |
|     |                                      | 8.2: información de trabajo                        |
|     |                                      | 8.3: registro de trabajo anterior                  |
|     |                                      | 8.4: información de trabajo anterior               |
|     |                                      | 8.5: registro de bachillerato                      |
|     |                                      | 8.6: información de bachillerato                   |
|     |                                      | 8.7: registro de estudios superiores               |
|     |                                      | 8.8: información de estudios superiores            |
|     |                                      | 8.9: registro de conocimiento de idiomas           |
|     |                                      | 8.10: información de conocimientos de idiomas      |

|    |                         |   |
|----|-------------------------|---|
|    |                         | 8.11: registro de conocimiento del idioma ingles    |
|    |                         | 8.12: información de conocimiento del idioma ingles |
|    |                         | 8.13: registro de cursos                            |
|    |                         | 8.14: información de cursos                         |
| 9  | Registro de postulación | 9.1: registro de postulación                        |
|    |                         | 9.2: información de postulación                     |
| 10 | Reportes                | 10.1: emisión de reportes                           |

Tabla 3.3: Historias de Usuario

A continuación se detalla las historias de usuario obtenidas, las cuales se muestran en las tablas: 3.4, 3.9, 3.12, 3.15, 3.20, 3.26, 3.39, 3.44, 3.59 y 3.62, y las tareas asociadas a cada historia de usuario.

Historia de Usuario N° 1, esta se encuentra relacionada con el tratamiento de la información de cooperaciones canalizadas y directas las cuales son esenciales para el establecimiento de las diferentes convocatorias. El registro de cooperaciones contiene cuatro grupos de datos como son: cooperación, máxima autoridad de cooperación, representación de cooperación en Bolivia y contactos de cooperación, este último tiene un registro por separado al de cooperaciones.

| HISTORIA DE USUARIO   |                                 |
|---|---------------------------------|
| Numero: 1   | Nombre: registro de cooperación |
| Permitir al usuario realizar el registro de cooperaciones (directa y canalizada) y a sus contactos. La cooperación registrada será accedida mediante una lista de cooperaciones (directa o canalizada) para luego realizar su búsqueda, con opción de modificar o eliminar. |                                 |

Tabla 3.4: Historia de usuario: registro de cooperación

Las tareas de programación que corresponden a la historia de usuario registro de cooperación se muestran en las tablas: 3.5, 3.6, 3.7 y 3.8.

|  |
|--|
| Tarea 1.1: registro de cooperación   |
| Elaborar un formulario para el registro de cooperaciones (directa y canalizada) con verificación y validaciones de datos de entrada y el procesamiento para su almacenamiento en la base de datos. |

Tabla 3.5: Tarea de programación: registro de cooperación

|   |
|---|
| Tarea 1.2: información de cooperación   |
| Elaborar una interfaz que muestre la lista de cooperaciones registradas con la opción realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- buscar cooperación</li> <li>- eliminar cooperación</li> <li>- modificar cooperación</li> <li>- mostrar datos de cooperación</li> </ul> |

Tabla 3.6: Tarea de programación: información de cooperación

|   |
|---|
| Tarea 1.3: registro de contacto de cooperación  |
| Elaborar un formulario para el registro de contactos de cooperación (directa y canalizada) en el cual se despliegue una lista de cooperaciones canalizadas para su selección, teniendo en cuenta la validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.7: Tarea de programación: registro de contacto de cooperación

|  |
|--|
| Tarea 1.4: información de contacto de cooperación  |
| Elaborar una interfaz que muestre la lista de contactos de una determinada cooperación con la opción de modificar o eliminar los contactos de dicha lista. |

Tabla 3.8: Tarea de programación: información de contacto de cooperación

Historia de Usuario N° 2, trata de los programas que tienen las diferentes cooperaciones internacionales, específicamente a las cooperaciones que son tratadas como canalizadas por la unidad de becas.

| HISTORIA DE USUARIO   |   |
|---|---|
| Numero: 2   | Nombre: registro de programa de cooperación |
| Permitir al usuario realizar el registro de programas de cooperaciones canalizadas. El programa registrado será accedido mediante una lista de programas de cooperación con opción de modificar o eliminar. |   |

Tabla 3.9: Historia de usuario: registro de programa de cooperación

Las tareas de programación que corresponden a la historia de usuario registro de programa de cooperación se muestran en las tablas: 3.10 y 3.11.

|   |
|---|
| Tarea 2.1: registro de programa de cooperación  |
| Elaborar un formulario para el registro de programa de cooperación en el cual se despliegue las cooperaciones canalizadas para su selección, con la verificación y validación del nombre de programa y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.10: Tarea de programación: registro de programa de cooperación

|  |
|--|
| Tarea 2.2: información de programa de cooperación  |
| Elaborar una interfaz que muestre la lista de programas de una determinada cooperación canalizada, con opciones de eliminación o modificación de los programas de la lista |

Tabla 3.11: Tarea de programación: información de programa de cooperación

Historia de Usuario N° 3, se encuentra relacionada con los documentos que llegan a la EGPP respecto a las convocatorias de becas que son derivadas a la unidad de becas, en esta se procede a su registro tomando los datos de identificación y referencia de estos como ser: número de hoja de ruta, gestión, cantidad de hojas del documento, fecha de ingreso a la EGPP y fecha de ingreso a la unidad de becas.

| HISTORIA DE USUARIO   |                                     |
|---|-------------------------------------|
| Numero: 3   | Nombre: registro de correspondencia |
| El usuario realiza el registro de correspondencia que llega a la institución referente a las convocatorias de becas. La correspondencia registrada será accedida mediante una lista de correspondencias para luego realizar su búsqueda con opción de modificar o eliminar. |                                     |

Tabla 3.12: Historia de usuario: registro de correspondencia

Las tareas de programación que corresponden a la historia de usuario registro de correspondencia se muestran en las tablas: 3.13 y 3.14.

|  |
|--|
| Tarea 3.1: registro de correspondencia   |
| Elaborar un formulario para el registro de correspondencia con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.13: Tarea de programación: registro de correspondencia

|   |
|---|
| Tarea 3.2: información de correspondencia   |
| Elaborar una interfaz que muestre la lista de correspondencias registradas con opciones para realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- buscar correspondencia</li> <li>- modificar correspondencia</li> <li>- eliminar correspondencia.</li> </ul> |

Tabla 3.14: Tarea de programación: información de correspondencia

Historia de Usuario N° 4, esta contempla a las convocatorias directas y canalizadas que son extractadas de la documentación de correspondencia que llega a la unidad de becas que posteriormente se realiza su registro.

|  |                                  |
|--|----------------------------------|
| HISTORIA DE USUARIO  |                                  |
| Numero: 4  | Nombre: registro de convocatoria |
| El usuario realiza el registro de convocatoria de beca (directa o canalizada) asociando a este una correspondencia, una cooperación, un programa (programa en el caso de una convocatoria canalizada y la opción de subir archivos adjuntos). La convocatoria registrada será accedida mediante una lista de convocatorias para luego realizar su búsqueda con opción de modificar o eliminar. |                                  |

Tabla 3.15: Historia de Usuario: registro de convocatoria

Las tareas de programación que corresponden a la historia de usuario registro de convocatoria, se muestran en las tablas: 3.16, 3.17, 3.18 y 3.19.

|   |
|---|
| Tarea 4.1: registro de convocatoria directa   |
| Elaborar un formulario para el registro de convocatoria de beca directa desplegando en esta las cooperaciones directas para su selección, con la verificación y validación de correspondencia registrada. Generar código de convocatoria directa en base a parámetros establecidos en el anterior sistema y el procesamiento del formulario para el almacenamiento a la base de datos |

Tabla 3.16: Tarea de programación: registro de convocatoria directa

|  |
|--|
| Tarea 4.2: información de convocatoria directa   |
| <p>Elaborar una interfaz que muestre la lista de convocatorias directas registradas con la opción de realizar las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- búsqueda de convocatoria directa.</li> <li>- modificación de convocatoria directa.</li> <li>- eliminación de convocatorias directa.</li> <li>- Mostrar datos de la convocatoria directa</li> </ul> |

Tabla 3.17: Tarea de programación: información de convocatoria canalizada

|  |
|--|
| Tarea 4.3: registro de convocatoria canalizada   |
| <p>Elaborar un formulario para el registro de convocatoria canalizada desplegando en esta las cooperaciones canalizadas y sus programas para su selección, la verificación y validación de correspondencia registrada y campos para de subir 3 archivos adjuntos a este. Generar código de convocatoria canalizada en base a parámetros establecidos en el anterior sistema y el procesamiento del formulario para el almacenamiento a la base de datos.</p> |

Tabla 3.18: Tarea de programación: registro de convocatoria canalizada

|   |
|---|
| Tarea 4.4: información de convocatoria canalizada   |
| <p>Elaborar una interfaz que muestre una lista de convocatorias canalizadas registradas con la opción de realizar las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- Buscar convocatoria canalizada</li> <li>- Modificar convocatoria canalizada</li> <li>- eliminar convocatoria canalizada.</li> <li>- Mostrar datos de convocatoria canalizada</li> </ul> |

Tabla 3.19: Tarea de programación: información de convocatoria canalizada

Historia de Usuario N° 5, esta contempla a la administración de usuarios (funcionarios de la unidad de becas de la EGPP) que trabajan con la información de convocatorias en la unidad de becas por medio del módulo de becas.

| HISTORIA DE USUARIO  |                            |
|--|----------------------------|
| Numero: 5  | Nombre: ingreso de usuario |
| Permitir al usuario autorizado ingresar al sistema mediante la introducción de su usuario y contraseña, para luego este realice las actividades establecidas por el administrador del sistema. |                            |

Tabla 3.20: Historia de usuario: ingreso de usuario

Las tareas de programación que corresponden a la historia de usuario: ingreso de usuario, se muestran en las tablas: 3.21, 3.22, 3.23, 3.24 y 3.25.

| Tarea 5.1: interfaz principal  |
|--|
| Desarrollar la interfaz principal del sistema considerando los colores institucionales, logotipo, banner publicitario, menú de navegación, enlaces a descargar de documentos y enlaces a la lista de convocatorias directas y canalizadas. |

Tabla 3.21: Tarea de programación: interfaz principal

| Tarea 5.2: registro de usuario   |
|--|
| Elaborar un formulario para el registro de usuarios realizando el encriptado MD5 de la contraseña, con la verificación y validación de los datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.22: Tarea de programación: registro de usuario

| Tarea 5.3: administración de usuarios   |
|---|
| Desarrollar la interfaz que muestre la lista de usuarios registrados y el procesamiento que permita al administrador del sistema a:   |
| <ul style="list-style-type: none"> <li>- habilitar a un usuario para el ingreso al sistema.</li> <li>- inhabilitar a un usuario para el ingreso al sistema.</li> <li>- habilitar o inhabilitar permisos de registro, eliminación o modificación de registros a un usuario.</li> </ul> |

Tabla 3.23: Tarea de programación: administración de usuarios

|   |
|---|
| Tarea 5.4: ingreso de usuario   |
| Elaborar un formulario para el ingreso de usuarios con el procesamiento de verificación de usuario y contraseña, llegando a realizar encriptación de la contraseña para la verificación. Tras el ingreso del usuario al sistema presentar a este las interfaces definidas por el administrador. |

Tabla 3.24: Tarea de programación: ingreso de usuario

|   |
|---|
| Tarea 5.5: historial de usuario   |
| Desarrollar el procesamiento de registro a la base de datos de acciones de usuario como ser: registro, eliminación y modificación sobre los diferentes formularios y listas de registros. |

Tabla 3.25: Tarea de programación: historial de usuario

Historia de Usuario Nº 6, consta de las instituciones patrocinadoras las cuales son el medio de certificación del nivel académico o experiencia laboral de los postulantes. Cada institución patrocinadora contiene datos de contactos hacia estas.

| HISTORIA DE USUARIO   |                                  |
|---|----------------------------------|
| Numero: 6   | Nombre: registro de patrocinador |
| Permitir al usuario realizar el registro de instituciones patrocinadoras (Unidad educativa, Universidad o Institución) y sus contactos. Las instituciones patrocinadoras registradas serán accedidas por medio de una lista para luego realizar su búsqueda, con la opción de eliminar o modificar. |                                  |

Tabla 3.26: Historia de usuario: registro de patrocinador

Las tareas de programación que corresponden a la historia de usuario registro de patrocinador, se muestran en las tablas: 3.27, 3.28, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.37 y 3.38.

|   |
|---|
| Tarea 6.1: registro de unidad educativa patrocinadora   |
| Elaborar un formulario para el registro de unidad educativa patrocinadora con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.27: Tarea de programación: registro de unidad educativa patrocinadora



|  |
|--|
| Tarea 6.2: información de unidad educativa patrocinadora   |
| Elaborar una interfaz que muestre la lista de unidades educativas patrocinadoras registradas con la opción de realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- Buscar unidad educativa patrocinadora</li> <li>- Modificar unidad educativa patrocinadora</li> <li>- eliminar unidad educativa patrocinadora.</li> <li>- Mostrar datos de unidad educativa patrocinadora</li> </ul> |

Tabla 3.28: Tarea de programación: información de unidad educativa patrocinadora

|  |
|--|
| Tarea 6.3: registro de contacto de unidad educativa  |
| Elaborar un formulario para el registro de contactos de unidad educativa, con la validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.29: Tarea de programación: registro de contacto de unidad educativa

|  |
|--|
| Tarea 6.4: información de contacto de unidad educativa   |
| Elaborar una interfaz que muestre la lista de contactos de una unidad educativa con la opción de modificar o eliminar los registros de la lista. |

Tabla 3.30: Tarea de programación: información de contacto de unidad educativa

|  |
|--|
| Tarea 6.5: registro de universidad patrocinadora   |
| Elaborar un formulario para el registro de Universidad patrocinadora con verificación y validación de datos y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.31: Tarea de programación: registro de universidad patrocinadora

|   |
|---|
| Tarea 6.6: información de universidad patrocinadora   |
| Elaborar una interfaz que muestre la lista de Universidades patrocinadoras registradas con la opción de realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- Buscar unidad Universidad patrocinadora</li> <li>- Modificar unidad Universidad patrocinadora</li> <li>- eliminar unidad Universidad patrocinadora.</li> <li>- Mostrar datos de Universidad patrocinadora</li> </ul> |

Tabla 3.32: Tarea de programación: información de universidad patrocinadora

|  |
|--|
| Tarea 6.7: registro de contacto de universidad |
|--|

|   |
|---|
| Elaborar un formulario para el registro de contactos de universidad, con la validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.33: Tarea de programación: registro de contacto de universidad

|   |
|---|
| Tarea 6.8: información de contacto de universidad |
|---|

|   |
|---|
| Elaborar una interfaz que muestre la lista de contactos de una universidad con la opción de modificar o eliminar los registros de la lista. |
|---|

Tabla 3.34: Tarea de programación: información de contacto de universidad

|  |
|--|
| Tarea 6.9: registro de institución patrocinadora |
|--|

|  |
|--|
| Elaborar un formulario para el registro de institución patrocinadora con verificación y validación de datos y el procesamiento del formulario para el almacenamiento a la base de datos. |
|--|

Tabla 3.35: Tarea de programación: registro de institución patrocinador

|  |
|--|
| Tarea 6.10: información de institución patrocinadora |
|--|

|   |
|---|
| Elaborar una interfaz que muestre la lista de instituciones patrocinadoras registradas con la opción de realizar las siguientes acciones: |
|---|

- Buscar institución patrocinadora
- Modificar institución patrocinadora
- eliminar institución patrocinadora.
- Mostrar datos de institución patrocinadora

Tabla 3.36: Tarea de programación: información de institución patrocinadora

|   |
|---|
| Tarea 6.11: registro de contacto de institución |
|---|

|   |
|---|
| Elaborar un formulario para el registro de contactos de institución patrocinadora, con la validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.37: Tarea de programación: registro de contacto de institución

|   |
|---|
| Tarea 6.12: información de contacto de institución  |
| Elaborar una interfaz que muestre la lista de contactos de una institución patrocinadora con la opción de modificar o eliminar los registros de la lista. |

Tabla 3.38: Tarea de programación: información de contacto de institución

Historia de Usuario N° 7, se refiere a los documentos que llegan a la EGPP referentes a postulantes de las convocatorias de becas los cuales son derivados a la unidad de becas para su correspondiente registro.

| HISTORIA DE USUARIO  |                                |
|--|--------------------------------|
| Numero: 7  | Nombre: registro de postulante |
| Permitir al usuario realizar el registro de postulante y sus contactos de emergencia. Los postulantes registrados serán accedidos por medio de una lista para su búsqueda con la opción de modificar o eliminar. |                                |

Tabla 3.39: Historia de usuario: registro de postulante

Las tareas de programación que corresponden a la historia de usuario registro de postulante se muestran en las tablas: 3.40, 3.41, 3.42 y 3.43.

|  |
|--|
| Tarea 7.1: registro de postulante  |
| Elaborar un formulario para el registro de postulantes con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.40: Tarea de programación: registro de postulante

|   |
|---|
| Tarea 7.2: información de postulante  |
| Elaborar una interfaz que muestre la lista de postulantes registrados con la opción de realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- Buscar postulante.</li> <li>- Modificar postulante</li> <li>- Eliminación postulante.</li> <li>- Mostrar postulante.</li> </ul> |

Tabla 3.41: Tarea de programación: información de postulante

|  |
|--|
| Tarea 7.3: registro de contacto de emergencia  |
| Elaborar un formulario para el registro de contactos de emergencia del postulante, con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.42: Tarea de programación: registro de contacto de emergencia

|  |
|--|
| Tarea 7.4: información de contacto de emergencia   |
| Elaborar una interfaz que muestre la lista de contactos de emergencia de un postulante registrado con la opción de modificar o eliminar los registros de la lista. |

Tabla 3.43: Tarea de programación: información de contacto de emergencia

Historia de usuario N° 8, esta se deriva del registro de postulantes considerando en este a la formación académica y laboral del postulante como ser: trabajo, trabajo anterior, bachillerato, estudios superiores, conocimiento de idiomas, conocimiento del idioma inglés y cursos relacionados con el área de la beca.

| HISTORIA DE USUARIO  |  |
|--|--|
| Numero: 8  | Nombre: registro de formación del postulante |
| Permitir al usuario realizar el registro de la formación académica y laboral del postulante. La formación registrada referente a un postulante será accedida por medio de interfaces con la opción de modificar o eliminar dichos registros. |  |

Tabla 3.44: Historia de usuario: registro de postulante

Las tareas de programación que corresponden a la historia de usuario: registro de formación del postulante se muestran en las tablas: 3.45, 3.46, 3.47, 3.48, 3.49, 3.50, 3.51, 3.52, 3.53, 3.54, 3.55, 3.56, 3.57 y 3.58

|   |
|---|
| Tarea 8.1: registro de trabajo  |
| Elaborar un formulario para el registro de trabajo del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.45: Tarea de programación: registro de trabajo

|                                   |
|-----------------------------------|
| Tarea 8.2: información de trabajo |
|-----------------------------------|

|   |
|---|
| Elaborar una interfaz que muestre el trabajo registrado del postulante con la opción de realizar la modificación o eliminación de dicho registro. |
|---|

Tabla 3.46: Tarea de programación: información de trabajo

|   |
|---|
| Tarea 8.3: registro de trabajo anterior |
|---|

|  |
|--|
| Elaborar un formulario para el registro de trabajo anterior del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|--|

Tabla 3.47: Tarea de programación: registro de trabajo anterior

|  |
|--|
| Tarea 8.4: información de trabajo anterior |
|--|

|  |
|--|
| Elaborar una interfaz que muestre una lista de los trabajos anteriores registrados del postulante, con la opción de realizar la modificación o eliminación de los registros de la lista. |
|--|

Tabla 3.48: Tarea de programación: información de trabajo anterior

|                                     |
|-------------------------------------|
| Tarea 8.5: registro de bachillerato |
|-------------------------------------|

|   |
|---|
| Elaborar un formulario para el registro de la información referente al bachillerato del postulante con verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.49: Tarea de programación: registro de bachillerato

|  |
|--|
| Tarea 8.6: información de bachillerato |
|--|

|   |
|---|
| Elaborar una interfaz que muestre la información de bachillerato registrado del postulante, con la opción de modificar o eliminar dicho registro. |
|---|

Tabla 3.50: Tarea de programación: información de bachillerato

|  |
|--|
| Tarea 8.7: registro de estudios superiores |
|--|

|   |
|---|
| Elaborar un formulario para el registro de estudios superiores del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.51: Tarea de programación: registro de estudios superiores

|   |
|---|
| Tarea 8.8: información de estudios superiores |
|---|

|   |
|---|
| Elaborar una interfaz que muestre una lista de estudios superiores registrados del postulante, con la opción de modificar o eliminar los registros de la lista. |
|---|

Tabla 3.52: Tarea de programación: información de estudios superiores

|  |
|--|
| Tarea 8.9: registro de conocimiento de idiomas |
|--|

|   |
|---|
| Elaborar un formulario para el registro de conocimiento de idiomas del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.53: Tarea de programación: registro de conocimiento de idiomas

|  |
|--|
| Tarea 8.10: información de conocimiento de idiomas |
|--|

|  |
|--|
| Elaborar una interfaz que muestre conocimiento de idiomas registrados del postulante, con la opción de modificar dicho registro. |
|--|

Tabla 3.54: Tarea de programación: información de conocimiento de idiomas

|  |
|--|
| Tarea 8.11: registro de conocimiento del idioma ingles |
|--|

|  |
|--|
| Elaborar un formulario para el registro de conocimiento del idioma ingles del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|--|

Tabla 3.55: Tarea de programación: registro de conocimiento del idioma ingles

|   |
|---|
| Tarea 8.12: información de conocimiento del idioma ingles |
|---|

|  |
|--|
| Elaborar una interfaz que muestre la información referente al conocimiento del idioma ingles registrada del postulante, con la opción de modificar dicho registro. |
|--|

Tabla 3.56: Tarea de programación: información de conocimiento del idioma ingles

|                                |
|--------------------------------|
| Tarea 8.13: registro de cursos |
|--------------------------------|

|   |
|---|
| Elaborar un formulario para el registro cursos del postulante con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |
|---|

Tabla 3.57: Tarea de programación: registro de cursos

|  |
|--|
| Tarea 8.14: información de cursos  |
| Elaborar una interfaz que muestre una lista de los cursos registrados del postulante, con la opción de modificar o eliminar los registros de la lista. |

Tabla 3.58: Tarea de programación: información de cursos

Historia de Usuario N° 9, esta trata de la relación de los registros: convocatoria canalizada, institución patrocinadora y postulantes, llegando a asociarlas para una postulación.

| HISTORIA DE USUARIO   |                     |
|---|---------------------|
| Numero: 9   | Nombre: postulación |
| Permitir al usuario realizar el registro de postulación mediante la relación convocatoria canalizada, institución patrocinadora y postulante. Los registros de postulación serán accedidos por medio de una lista para su búsqueda con la opción de modificar o eliminar. |                     |

Tabla 3.59: Historia de usuario: postulación

Las tareas de programación que corresponden a la historia de usuario postulación se muestran en las tablas: 3.60 y 3.61.

|  |
|--|
| Tarea 9.1: registro de postulación   |
| Elaborar un formulario para el registro de postulación con la verificación y validación de datos de entrada y el procesamiento del formulario para el almacenamiento a la base de datos. |

Tabla 3.60: Tarea de programación: registro de inscripción de postulante

|  |
|--|
| Tarea 9.2: información de postulación  |
| Elaborar una interfaz que muestre la lista de postulaciones con la opción de realizar las siguientes acciones: <ul style="list-style-type: none"> <li>- Buscar postulación</li> <li>- Modificar postulación</li> <li>- Eliminar postulación</li> <li>- Mostrar postulación.</li> </ul> |

Tabla 3.61: Tarea de programación: información de postulación

Historia de Usuario N° 10, se trata de la emisión de reportes referentes a las convocatorias directas y canalizadas.

| HISTORIA DE USUARIO  |                  |
|--|------------------|
| Numero: 10   | Nombre: reportes |
| Permitir al usuario emitir reportes en base a parámetros definidos por la unidad de becas. |                  |

Tabla 3.62: Historia de Usuario: reportes

La tarea de programación que corresponde a la historia de usuario reportes se muestra en la tabla 3.63.

|  |
|--|
| Tarea 10.1: emisión de reportes  |
| Desarrollar el procesamiento para la generación de reportes de: oferta de becas directas y canalizadas en formato pdf. |

Tabla 3.63: Tarea de programación: emisión de reportes

### 3.3.1.3. PLAN DE ENTREGAS

En el plan de entregas se establecen los módulos del presente trabajo, al mismo tiempo tenemos a las historias de usuario que conforman a estos módulos. Los incrementos se presentan en base a la relación que se tiene entre historias de usuario y tiempos estimados de entrega, estos tiempos se definen en base a la complejidad de las tareas de programación que tienen las historias de usuario. El plan de entregas se muestra en la tabla 3.64.

| MODULO             | HISTORIAS DE USUARIO   | INCREMENTO | DURACION  | JULIO |    |    | AGOSTO |    |    | SEPTIEMBRE |    |    | OCTUBRE |     |     | NOVIEMBRE |     |     |     |     |     |     |     |     |     |     |
|--------------------|------------------------|------------|-----------|-------|----|----|--------|----|----|------------|----|----|---------|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                    |                        |            |           | S1    | S2 | S3 | S4     | S5 | S6 | S7         | S8 | S9 | S10     | S11 | S12 | S13       | S14 | S15 | S16 | S17 | S18 | S19 | S20 | S21 | S22 | S23 |
| Convocatorias      | Historia de usuario 1  | primer     | 3 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 2  |            |           |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 3  |            |           |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 4  | segundo    | 3 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
| Ingreso al sistema | Historia de usuario 5  | tercer     | 2 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
| Postulantes        | Historia de usuario 6  | cuarto     | 3 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 7  |            |           |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 8  | quinto     | 3 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
|                    | Historia de usuario 9  | sexto      | 2 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |
| Reportes           | Historia de usuario 10 | septimo    | 2 semanas |       |    |    |        |    |    |            |    |    |         |     |     |           |     |     |     |     |     |     |     |     |     |     |

Tabla 3.64: plan de entregas



### 3.3.2. DISEÑO Y CODIFICACION

En esta sección se presenta a los módulos definidas en el plan de entregas (tabla 3.64) los cuales son: módulo de convocatorias, módulo de ingreso al sistema, módulo de postulantes y el módulo de reportes. La presentación de los módulos que conforman el proyecto será bajo el siguiente orden: a) Tarjetas CRC, b) Modelo Estructural y c) Modelo de Hipertexto.

#### 3.3.2.1. MODULO DE CONVOCATORIAS

##### a) TARJETAS CRC

Las tarjetas CRC nos ayudaran a la implementación de las clases que serán utilizadas en el desarrollo del módulo de convocatorias. También nos ayuda a identificar que clases interactúan directa o indirectamente dentro de este módulo. El presente modulo consta de las siguientes tarjetas CRC, las cuales se muestran en las tablas 3.65, 3.66, 3.67, 3.68, 3.69, 3.70, 3.71 y 3.72.

| COOPERACION   |   |
|---|---|
| Se encarga de administrar las cooperaciones directas y canalizadas. También realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Grupo</li> <li>▪ Tipo</li> <li>▪ País</li> </ul> |

Tabla 3.65: Tarjeta CRC Cooperación

| CONTACTOS  |   |
|--|---|
| Se encarga de administrar los contactos de cooperaciones directas y canalizadas. También realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Cooperación</li> </ul> |

Tabla 3.66: Tarjeta CRC Contactos

| REPRESENTACION  |   |
|---|---|
| Forma parte del registro de cooperaciones el cual es manipulado conjuntamente con este. | <ul style="list-style-type: none"> <li>▪ Cooperación</li> </ul> |

Tabla 3.67: Tarjeta CRC Representación

| PROGRAMA   |   |
|--|---|
| Se encarga de la administración de programas de cooperaciones. También realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Cooperación</li> </ul> |

Tabla 3.68: Tarjeta CRC Programa

| BECA   |   |
|--|---|
| Se encarga de la administración de convocatorias directas y canalizadas. También realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Cooperación</li> <li>▪ Correspondencia</li> <li>▪ Idiomas</li> <li>▪ Modalidad país duración</li> <li>▪ Clasificación</li> <li>▪ Estudio área</li> <li>▪ Estudio área detalle</li> <li>▪ Estudio nivel</li> <li>▪ Estudio nivel detalle</li> <li>▪ Tipo presentación</li> <li>▪ Modalidad</li> </ul> |

Tabla 3.69: Tarjeta CRC Beca

| ARCHIVO ADJUNTO  |   |
|--|---|
| Se encarga de administrar los archivos adjuntos de las convocatorias para su descarga. | <ul style="list-style-type: none"> <li>▪ beca</li> <li>▪ Archivo Adjunto Ubicación</li> </ul> |

Tabla 3.70: Tarjeta CRC Cooperación

| BECA CANALIZADA   |  |
|---|--|
| Forma parte del registro de convocatorias canalizadas, y su manipulación es conjuntamente a este. | <ul style="list-style-type: none"> <li>▪ Beca</li> <li>▪ Programa</li> </ul> |

Tabla 3.71: Tarjeta CRC Beca canalizada

| ARCHIVO ADJUNTO DOCUMENTACION   |   |
|---|---|
| Se encarga de administrar los archivos referentes a documentos que deben presentar los postulantes y los pasos para la postulación. | <ul style="list-style-type: none"> <li>▪ Archivo Adjunto Ubicación</li> </ul> |

Tabla 3.72: Tarjeta CRC Cooperación

## b) MODELO ESTRUCTURAL

A continuación se muestra en la figura 3.3 y 3.4 la estructura de datos obtenida a través de las tarjetas CRC, la cual soportara el procesamiento de las operaciones requeridas por el Modulo de Convocatorias.

## c) MODELO DE HIPERTEXTO

### i) MODELO DE NAVEGACION

En la figura 3.5 podemos observar el modelo de navegación a través del cual se puede organizar el hipertexto referente al módulo de convocatorias, este comprende menú de convocatorias y los links hacia las diferentes áreas que componen al módulo de convocatorias. Considérese a un usuario que ingreso al sistema el cual por medio del menú principal elije la opción convocatorias, en seguida se despliega esta opción con los enlaces de registro de cooperación, lista de cooperaciones, registro de contactos de cooperación, registro de cooperaciones directas, registro de contactos de cooperaciones directas, registro de programas de cooperación, registro de correspondencias, lista de correspondencias, registro de convocatorias, lista de convocatorias, registro de convocatorias directas y las listas de convocatorias directas. Para el menú se utiliza una unidad índice la cual contiene los enlaces no contextuales

que se conectan hacia contenedores que conforman los nodos de navegación del módulo de convocatorias.

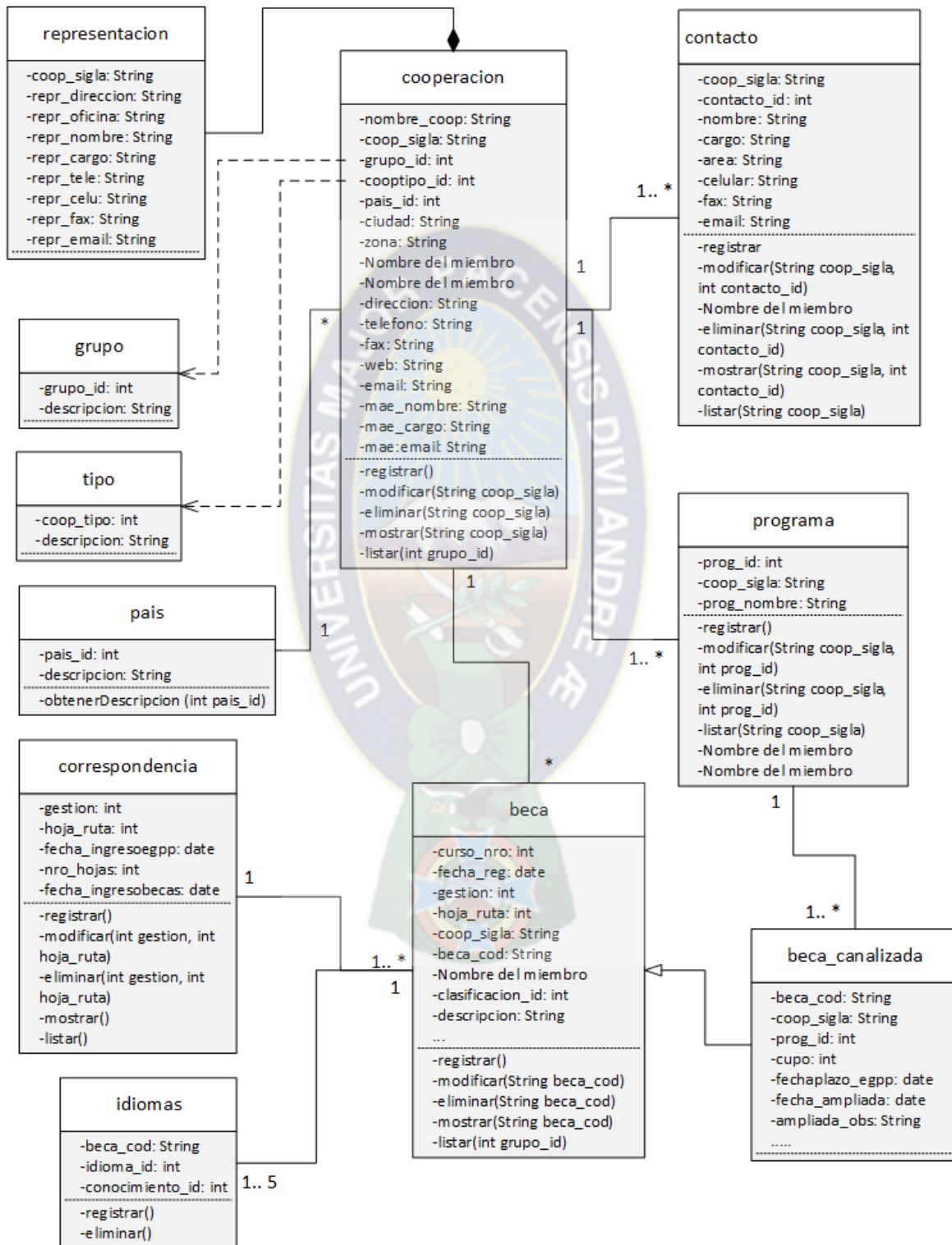


Figura 3.3: Modelo de Estructura – Módulo de convocatorias

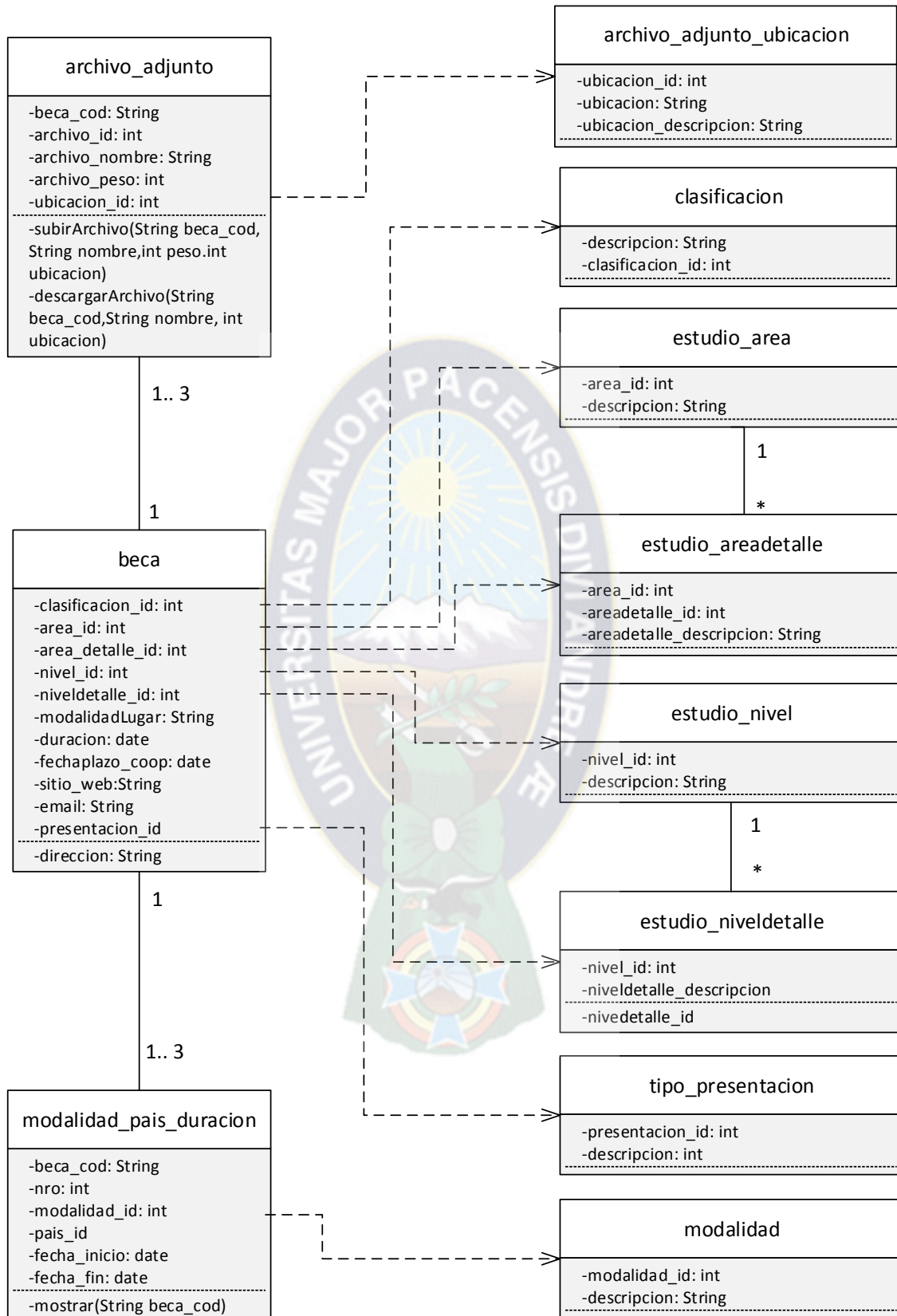


Figura 3.4: Modelo de estructura – Modulo de convocatorias (continuación)

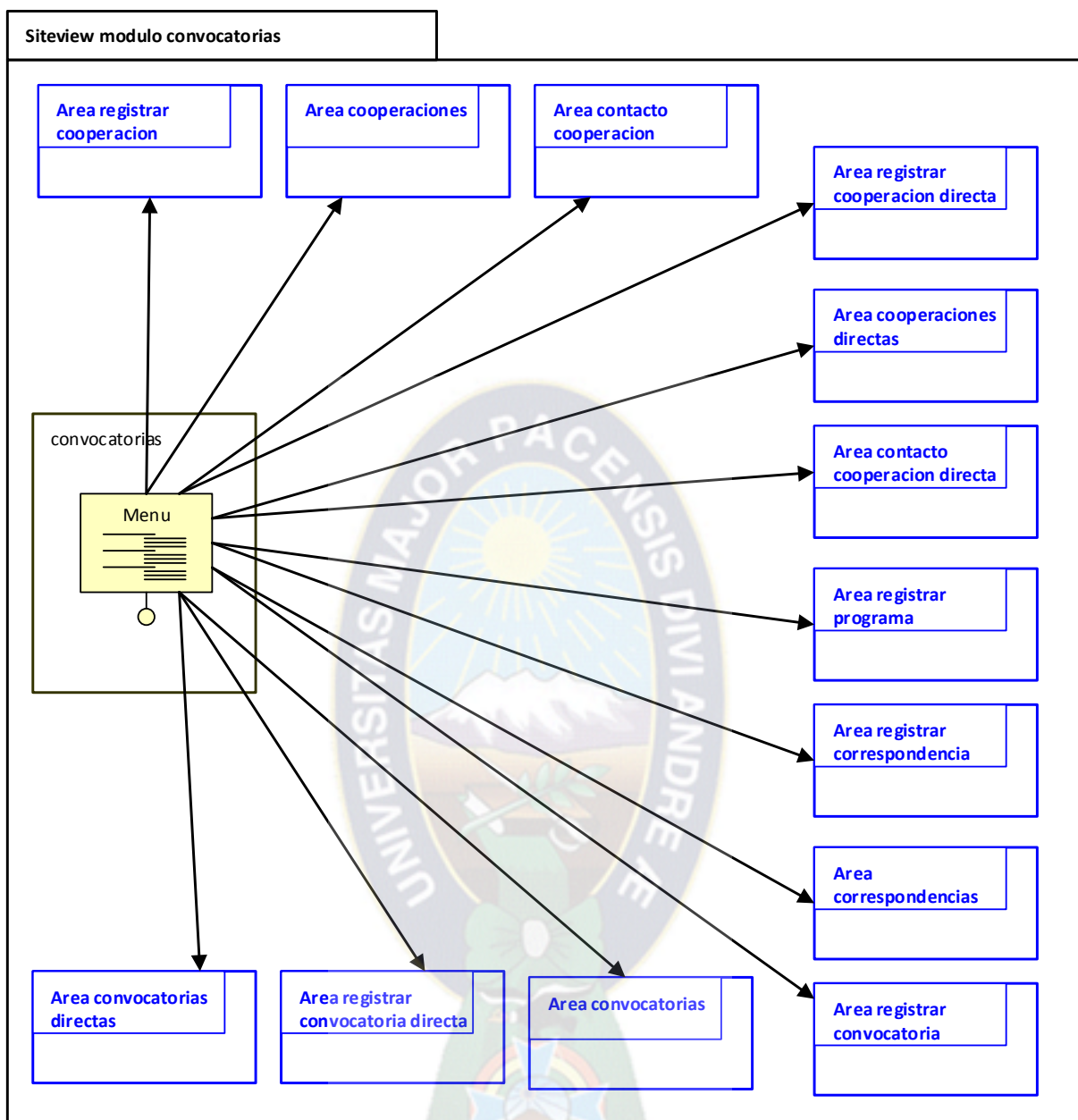


Figura 3.5: Modelo de Navegación – Módulo de Convocatorias

## ii) MODELOS DE COMPOSICION

A continuación describimos algunos de los nodos del modelo de navegación del módulo de convocatorias más representativos como ser: área registrar cooperación (registrar) y área cooperaciones (listar, mostrar, modificar, eliminar), los cuales comprenden el uso de operaciones que sirven para expresar como se manejan los datos.

- ▲ Modelo de composición del registro de cooperaciones (área registrar cooperación): este modelo se detalla en la figura 3.6, la cual especifica y detalla la realización desde principio a fin de esta operación a nivel de página y tecnología web. Considérese a un usuario con permisos de escritura en el sistema. El proceso comienza observándose la interfaz de registro de cooperación donde el usuario llena los campos del formulario y solicita el registro del mismo, acción la cual lleva al sistema a consultar al sí existe una cooperación que posea el nombre o sigla ingresado por el usuario. Esta búsqueda se realiza empleando una unidad de consulta simple la cual revisa en la tabla cooperación alguna ocurrencia del nombre o sigla ingresado en el formulario en cuestión. Si existe alguna cooperación que concuerde con esos datos se muestra un mensaje indicando la existencia de la misma. De lo contrario una unidad de registro o ingreso de datos se encarga de registrar los datos de la nueva cooperación ingresados por el usuario a la base de datos. Luego el sistema nos redirección a la página de bienvenida. En caso de que ocurra algún inconveniente durante todo el proceso la aplicación mostrara un mensaje al usuario indicando que corrija la información ingresada y la lista de los datos a corregir.

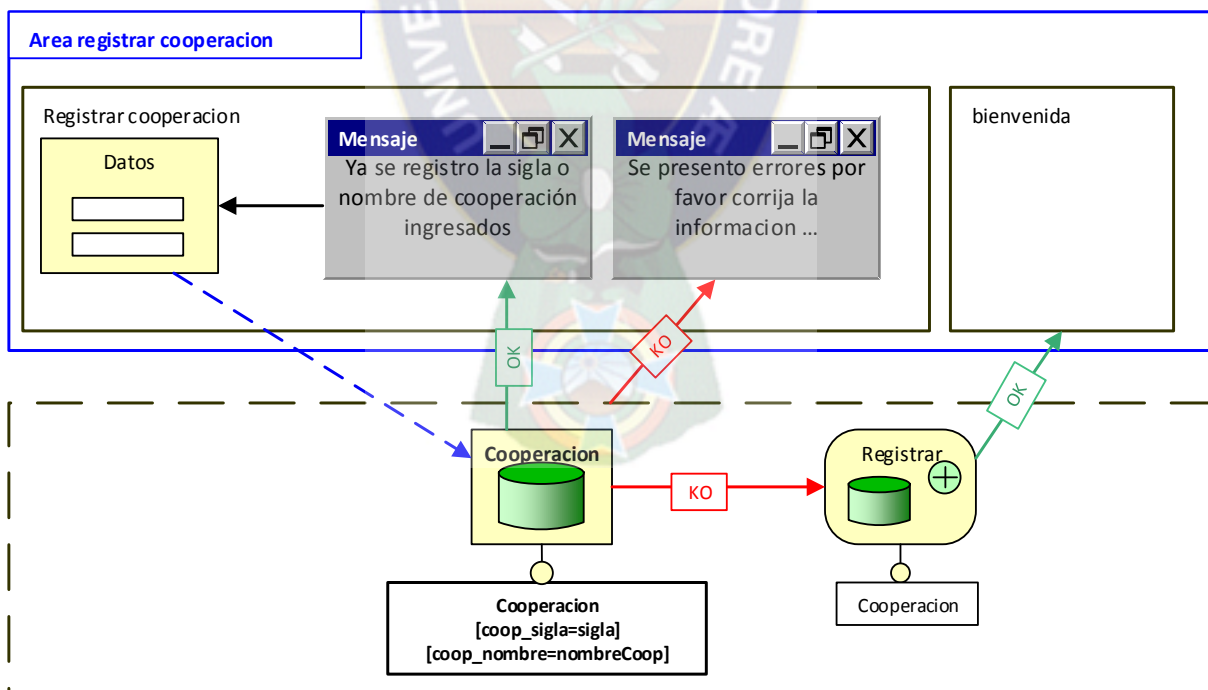


Figura 3.6: Modelo de Composición – Área Registrar Cooperación

- ✦ Modelo de composición de la lista de cooperaciones (área cooperaciones): considere que se encuentra en la interfaz de la lista de cooperaciones registradas, la cual presenta opciones de búsqueda, modificación, eliminación y mostrar cooperación, como se puede observar en la figura 3.7, la cual explica detalladamente el funcionamiento de estas opciones. Se presenta la lista de cooperaciones en una unidad de índice por medio de la cual podemos seleccionar alguna cooperación para realizar alguna operación, esta lista se puede desplazar a otra utilizando la unidad de scroller; opción de búsqueda: esta operación resulta por no encontrarse la cooperación deseada, para esto introducimos datos de búsqueda en un campo de entrada para esto empleamos una unidad de entrada buscar en la lista la cooperación deseada; opción de eliminación: considérese al usuario con permisos de eliminación, este proceso comienza seleccionando una cooperación para su eliminación esto lleva a la aplicación a verificar si existe algún programa o convocatoria asociado a la cooperación seleccionada esta operación se realiza empleando dos unidades de consulta simple las cuales verifican en las tablas de programa o convocatoria la sigla de la cooperación seleccionada de encontrar una o más coincidencias se muestra un mensaje que indica que no es posible eliminar la cooperación de lo contrario una unidad de eliminación procede a la eliminación de la cooperación seleccionada; opción de mostrar: la selección de una cooperación de la lista para ver sus datos nos lleva a una página en la cual nos muestra todos los datos registrados respecto a dicha cooperación, para esto se emplea una unidad de datos simple la cual contiene todos los datos relacionados con la sigla de cooperación que recibe como parámetro, la cual carga los datos de la cooperación a una unidad de entrada de datos.
  
- ✦ Modelo de composición modificar cooperación (área modificar cooperación): la operación de modificar es exclusivamente para usuarios con permisos de modificación, este proceso se detalla en la figura 3.8, la cual corresponde a la realización completa a nivel tecnología web de este modelo de composición. Considérese que la página donde se encuentra todos los campos de datos de la cooperación a modificar se está observando actualmente en la interfaz del sistema. El usuario hace las modificaciones necesarias correspondientes a dicha cooperación y procede a solicitar y confirmar la modificación de datos. Una unidad de consulta simple verifica en la base de datos, específicamente en la tabla cooperación, una cooperación con sigla distinta al de la cooperación recién modificada pero con igual nombre de cooperación. De ser así se



observa un mensaje de duplicación de cooperación (el nombre de cooperación ya existe). De lo contrario una unidad de modificación de datos recibe los datos sustitutos y dicha unidad se encarga de realizar el cambio de la información en la tabla cooperación de la base de datos, para luego devolvemos a la interfaz principal de la área de cooperaciones. En caso de ocurrir algún error durante la modificación se muestra un mensaje indicando los detalles del inconveniente.

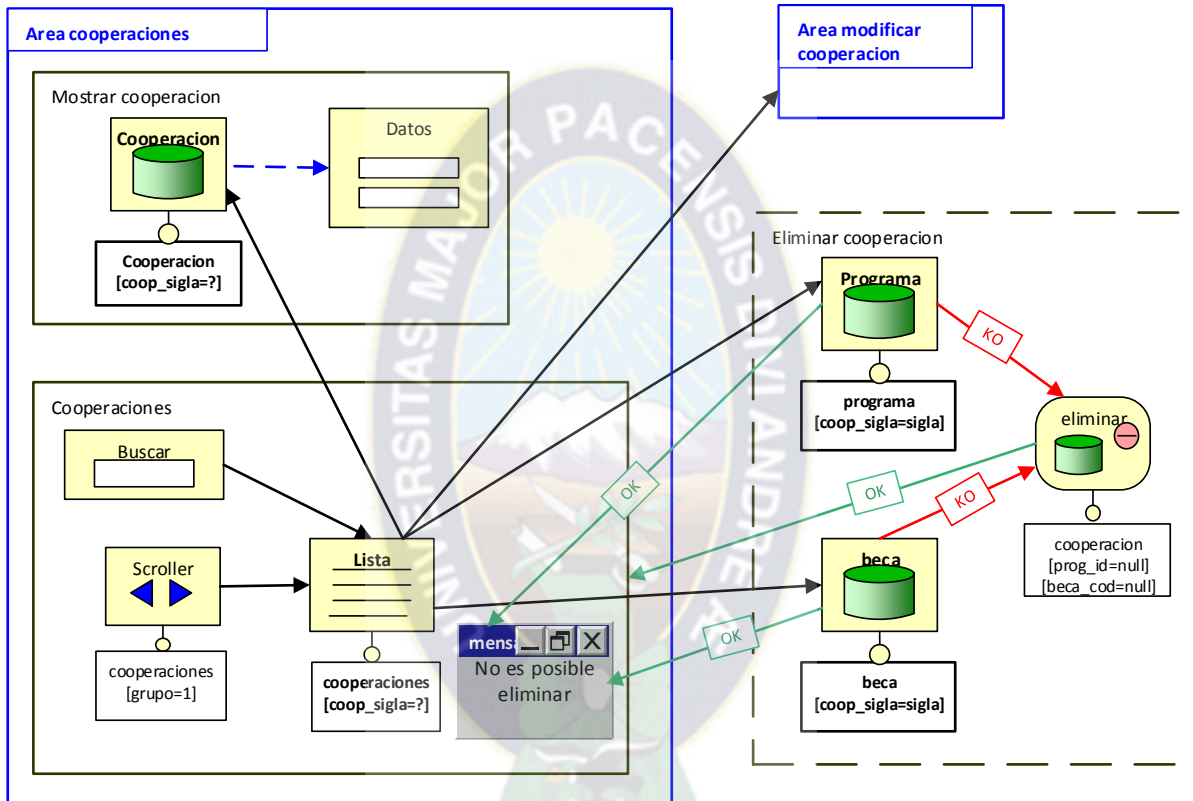


Figura 3.7: Modelo de Composición – Área cooperaciones

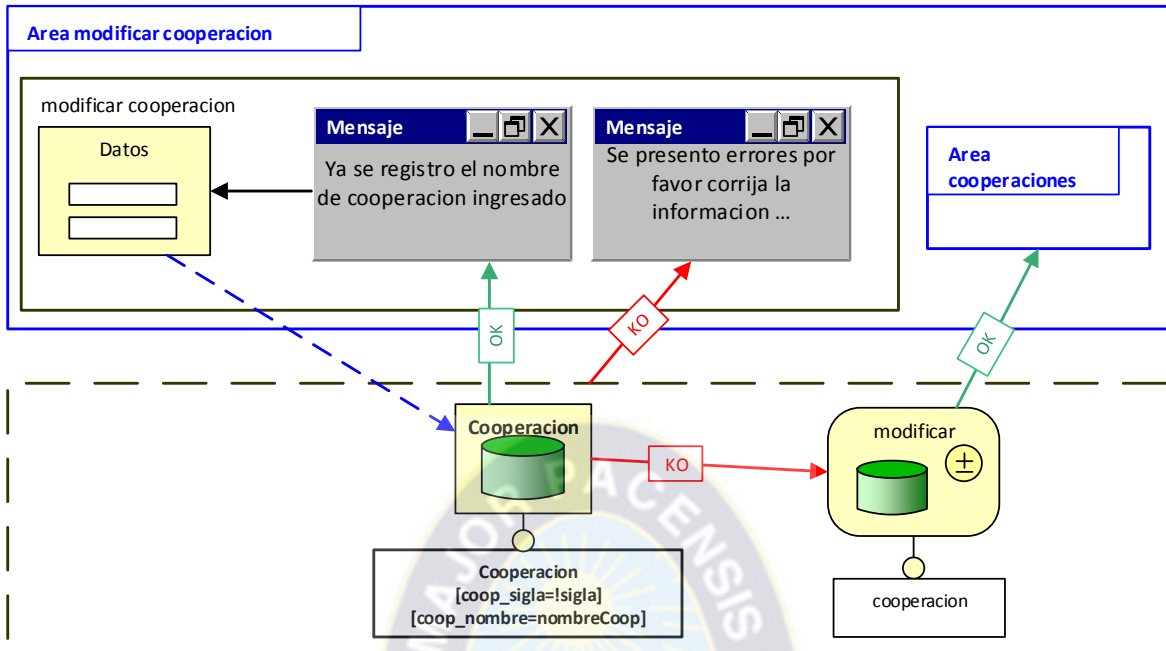


Figura 3.8: Modelo de Composición – Área Modificar Cooperación

### 3.3.2.2. MODULO DE INGRESO AL SISTEMA

#### a) TARJETAS CRC

Las tarjetas CRC correspondientes al Módulo de ingreso al sistema se muestran en las tablas 3.73 y 3.74.

| USUARIO  |  |
|--|--|
| Se encarga de administrar a los usuarios y sus permisos. | <ul style="list-style-type: none"> <li>▪ Usuario Estado</li> </ul> |

Tabla 3.73: Tarjeta CRC Usuario

| HISTORIAL   |  |
|---|--|
| Se encarga de administrar las acciones de los usuarios sobre los registros. | <ul style="list-style-type: none"> <li>▪ Historial acción</li> <li>▪ Historial formulario</li> </ul> |

Tabla 3.74: Tarjeta CRC Historial

## b) MODELO ESTRUCTURAL

A continuación se muestra en la figura 3.9 la estructura de datos obtenida a través de las tarjetas CRC, la cual soportara las operaciones requeridas por el Modulo de Ingreso al Sistema.

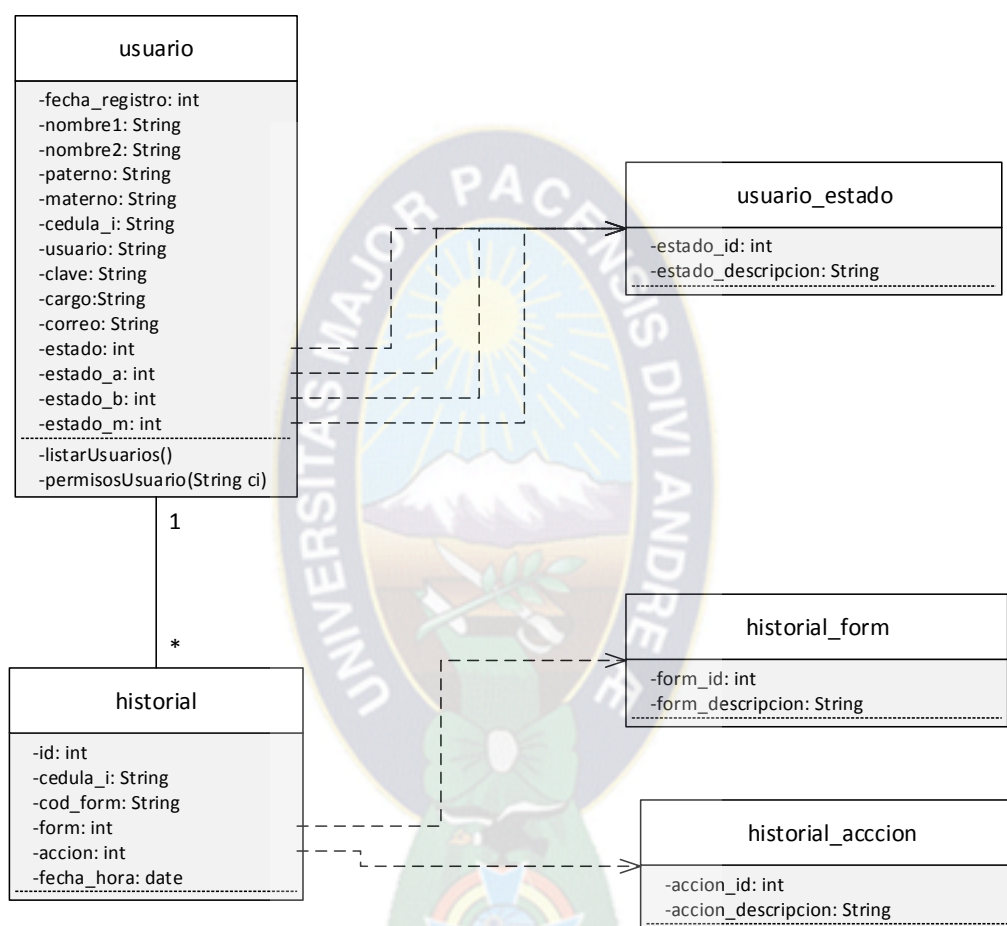


Figura 3.9: Modelo de estructura – Módulo de Ingreso al Sistema

## c) MODELO DE HIPERTEXTO

### i) MODELOS DE COMPOSICION

A continuación describimos los nodos del módulo de ingreso al sistema como son: ingreso de usuario y registro de usuario, los cuales comprenden el uso de operaciones que sirven para expresar como se manejan los datos.

- ✦ Modelo de composición de Ingreso de Usuario: este modelo se detalla en la figura 3.10, en este se puede observar el proceso que realizaría un funcionario del área de becas para poder ingresar al sistema. este proceso a nivel páginas web comienza con la interfaz exterior del sistema, donde se observa, entre otras cosas, un formulario con dos campos para que el usuario coloque su nombre de usuario y su contraseña. Al hacer la solicitud de ingreso a la aplicación el sistema consulta si existe un usuario registrado cuyo login y password sean los indicados, además que su estado para con el sistema sea de habilitado. Esto es llevado acabo utilizando una unidad de consulta simple encargada de buscar dichos datos en la tabla Usuario. En caso de que no cumpla alguna de estas condiciones se muestra un mensaje de rechazo y se vuelve a mostrar la interfaz de ingreso, de lo contrario si cumple todos los requisitos y es un usuario registrado y autenticado, automáticamente ingresara al sistema para iniciar su sesión.

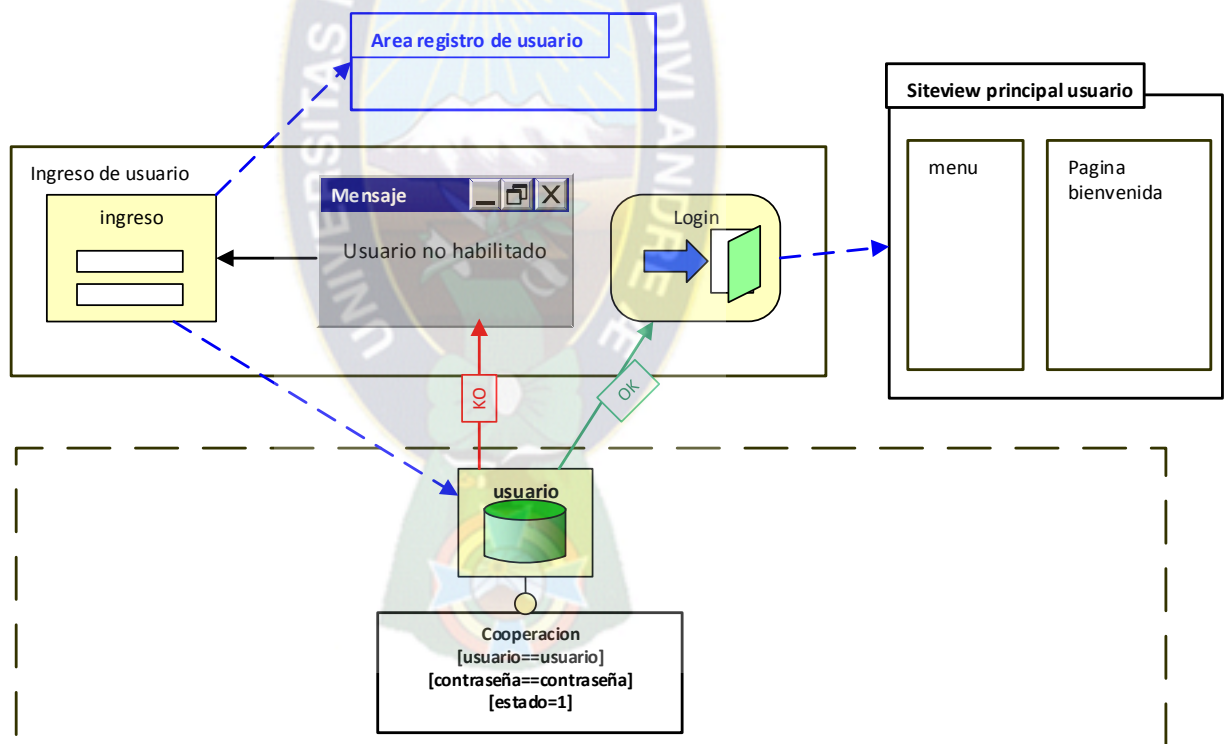


Figura 3.10: Modelo de Composición – Ingreso de Usuario

Posteriormente con el uso de una unidad de ingreso, el sistema lo llevara a la página principal con las opciones de acciones establecidas por el administrador como ser: registrar, eliminar o modificar registros. Al mismo tiempo se obtiene los datos relevantes

del usuario en variables globales conocidas como variables de sesión para el posterior registro de acciones sobre los registros.

- ▲ Modelo de composición registro de usuario (área registro de usuario) este modelo se muestra en la figura 3.11. Considérese a un usuario con permisos de escritura en el sistema. El proceso comienza observándose la interfaz de registro de usuario donde el usuario llena los campos del formulario y solicita el registro del mismo, acción la cual lleva al sistema a consultar al sí existe un usuario que posea la cedula de identidad o nick de usuario ingresados por el usuario. Esta búsqueda se realiza empleando una unidad de consulta simple la cual revisa en la tabla usuario alguna ocurrencia del cedula o Nick de usuario ingresado en el formulario en cuestión. Si existe algún usuario que concuerde con esos datos se muestra un mensaje indicando la existencia de la misma. De lo contrario una unidad de registro o ingreso de datos se encarga de registrar los datos de un nuevo usuario ingresados por el usuario a la base de datos.

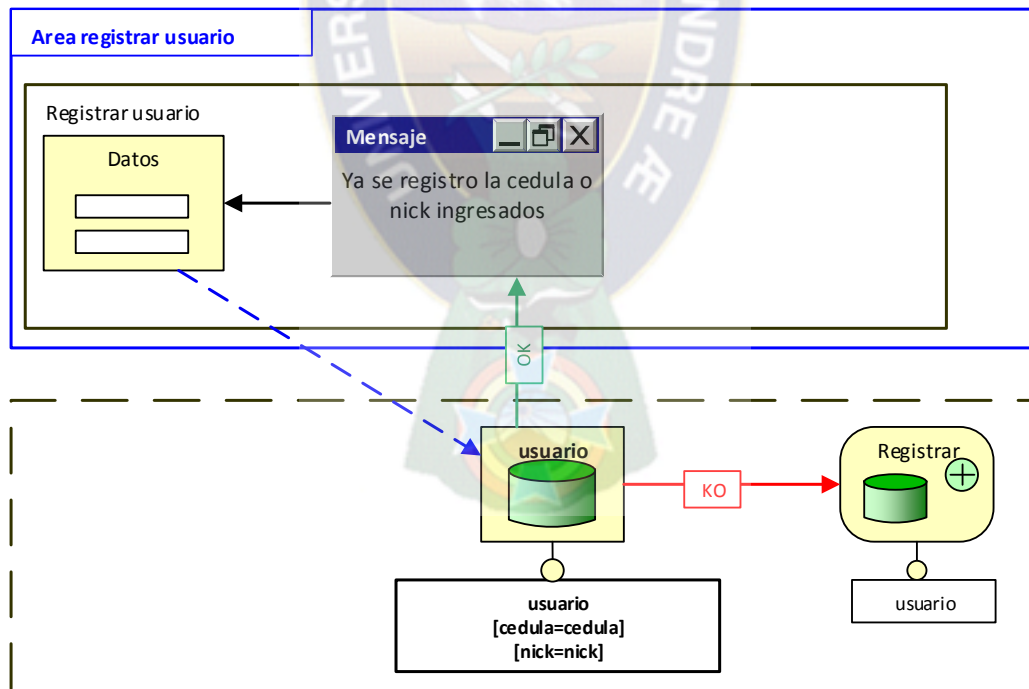


Figura 3.11: Modelo de Composición – Área Registrar usuario

## 3.3.2.3. MODULO DE POSTULANTES

## a) TARJETAS CRC

Las tarjetas CRC correspondientes al Módulo de postulantes se muestra en las tablas 3.75, 3.76, 3.77, 3.78, 3.79, 3.80, 3.81, 3.82, 3.83, 3.84, 3.85, 3.86, 3.87, 3.88 y 3.89.

| PATROCINADOR  |   |
|---|---|
| Se encarga de administrar a las instituciones patrocinadoras, Universidades y Unidades Educativas. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Patrocinador tipo</li> </ul> |

Tabla 3.75: Tarjeta CRC Patrocinador

| PATROCINADOR INSTITUCION  |  |
|---|--|
| Forma parte del registro Institución patrocinadora y su manipulación es conjuntamente a este. | <ul style="list-style-type: none"> <li>▪ Patrocinador sector</li> <li>▪ Patrocinador sector detalle</li> <li>▪ Patrocinador sector detalle tipo</li> </ul> |

Tabla 3.76: Tarjeta CRC Patrocinador Institucion

| PATROCINADOR UNIVERSIDAD  |   |
|---|---|
| Forma parte del registro Universidad y su manipulación es conjuntamente a este. | <ul style="list-style-type: none"> <li>▪ Pais</li> <li>▪ Departamento</li> <li>▪ Depto provincia</li> <li>▪ Depto prov Municipio</li> <li>▪ Depto prov munc Localidad</li> <li>▪ Patrocinador area</li> <li>▪ Patrocinador dependencia</li> </ul> |

Tabla 3.77: Tarjeta CRC Patrocinador Universidad

| PATROCINADOR UNIDAD EDUCATIVA  |   |
|--|---|
| Forma parte del registro Unidad Educativa y su manipulación es conjuntamente a este. | <ul style="list-style-type: none"> <li>▪ Pais</li> <li>▪ Departamento</li> <li>▪ Depto provincia</li> <li>▪ Depto prov Municipio</li> <li>▪ Depto prov munc Localidad</li> <li>▪ Patrocinador area</li> <li>▪ Patrocinador dependencia</li> <li>▪ Patrocinador turno</li> </ul> |

Tabla 3.78: Tarjeta CRC Patrocinador Unidad Educativa

| PATROCINADOR CONTACTO   |  |
|---|--|
| Se encarga de administrar los contactos de: Instituciones patrocinadoras, Universidades y Unidades educativas. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Patrocinador</li> </ul> |

Tabla 3.79: Tarjeta CRC Patrocinador Contacto

| POSTULANTE   |  |
|--|--|
| Se encarga de administrar a los Postulantes de las convocatorias canalizadas. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Pais</li> <li>▪ Departamento</li> <li>▪ Depto provincia</li> <li>▪ Depto prov Municipio</li> <li>▪ Depto prov munc Localidad</li> <li>▪ Genero</li> <li>▪ Estado civil</li> </ul> |

Tabla 3.80: Tarjeta CRC Postulante

| POSTULANTE CONTACTO  |  |
|--|--|
| Se encarga de administrar a los contactos de los postulantes. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Vinculo parentesco</li> </ul> |

Tabla 3.81: Tarjeta CRC Postulante Contacto

| POSTULANTE TRABAJO   |  |
|--|--|
| Se encarga de administrar la información referente al trabajo de los postulantes. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Patrocinador</li> </ul> |

Tabla 3.82: Tarjeta CRC Postulante Trabajo

| POSTULANTE TRABAJO ANTERIOR   |  |
|---|--|
| Se encarga de administrar la información referente a los trabajos anteriores de los postulantes. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> </ul> |

Tabla 3.83: Tarjeta CRC Postulante Trabajo Anterior



| POSTULANTE BACHILLERATO   |  |
|---|--|
| Se encarga de administrar la información referente a las materias y las notas obtenidas del postulante en su Unidad Educativa. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Patrocinador</li> </ul> |

Tabla 3.84: Tarjeta CRC Postulante Bachillerato

| POSTULANTE ESTUDIOS SUPERIORES   |   |
|--|---|
| Se encarga de administrar la información referente los estudios superiores obtenidos por el postulante. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Patrocinador</li> <li>▪ Estudio nivel</li> <li>▪ Estudio nivel detalle</li> <li>▪ Estudio titulación</li> <li>▪ Selección si no</li> </ul> |

Tabla 3.85: Tarjeta CRC Postulante Estudios Superiores

| POSTULANTE IDIOMA   |   |
|---|---|
| Se encarga de administrar la información referente los idiomas de conocimiento del postulante. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Idioma</li> <li>▪ Idioma conocimiento</li> </ul> |

Tabla 3.86: Tarjeta CRC Postulante idioma

| POSTULANTE IDIOMA INGLES   |   |
|--|---|
| Se encarga de administrar la información referente al nivel de conocimiento del postulante del idioma inglés. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Idioma conocimiento</li> </ul> |

Tabla 3.87: Tarjeta CRC Postulante Idioma Ingles

| POSTULANTE CURSOS   |  |
|---|--|
| Se encarga de administrar la información referente a los cursos del postulante relacionados con la beca que postula. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> </ul> |

Tabla 3.88: Tarjeta CRC Postulante Cursos

| POSTULACION   |   |
|---|---|
| Se encarga de administrar las postulaciones bajo la asociación convocatoria canalizada y postulante. También se realiza el listado, modificación, eliminación y acceso a la información de estos registros. | <ul style="list-style-type: none"> <li>▪ Postulante</li> <li>▪ Convocatoria canalizada</li> <li>▪ Patrocinador</li> <li>▪ Patrocinio tipo</li> <li>▪ Estado becado</li> </ul> |

Tabla 3.89: Tarjeta CRC Postulación

## b) MODELO ESTRUCTURAL

A continuación se muestra en la figura 3.12, 3.13 y 3.14 la estructura de datos obtenida a través de las tarjetas CRC, la cual soportara el procesamiento de las operaciones requeridas por el Modulo de Postulantes.

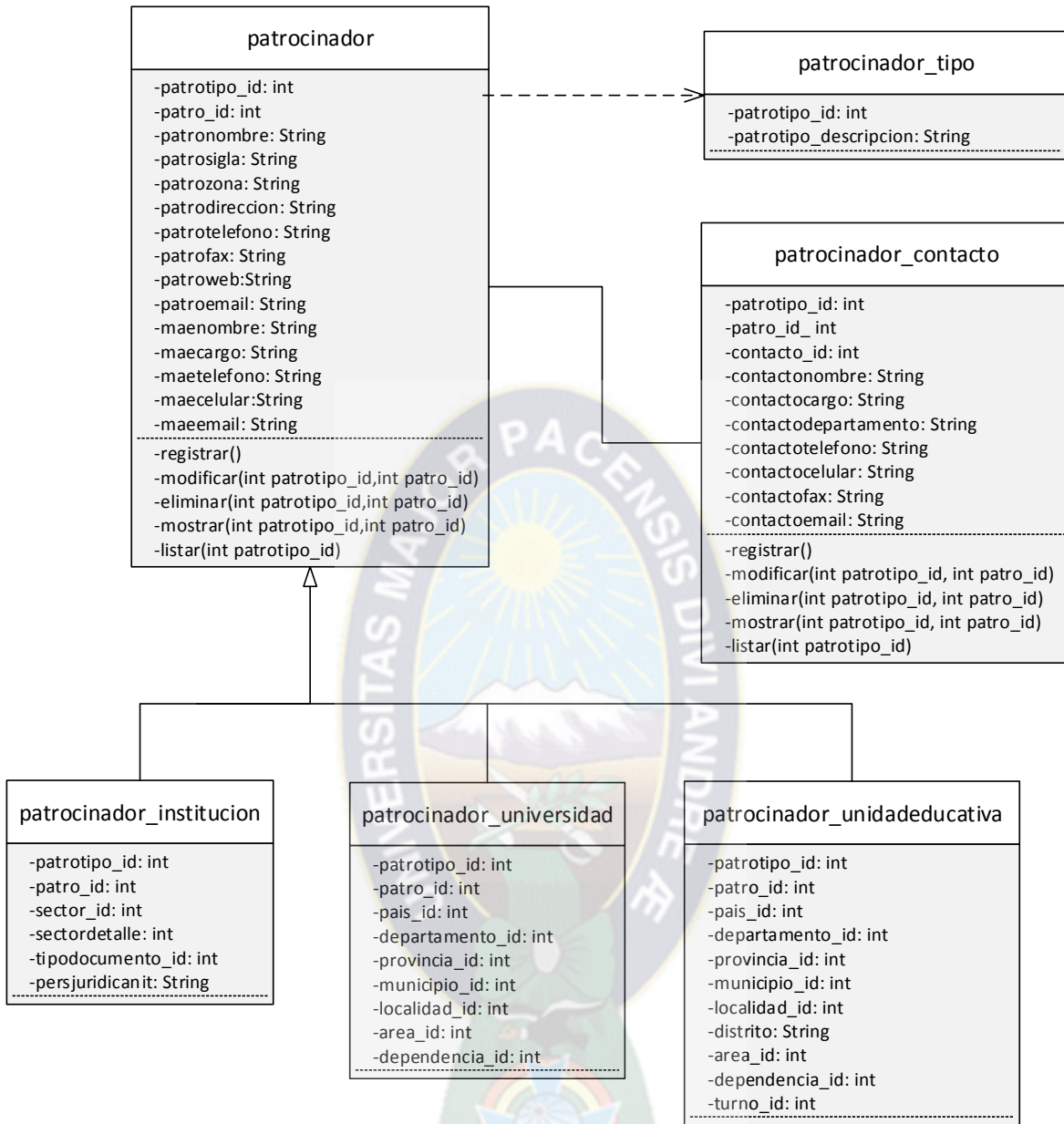


Figura 3.12: Modelo de estructura – Modulo de Postulantes

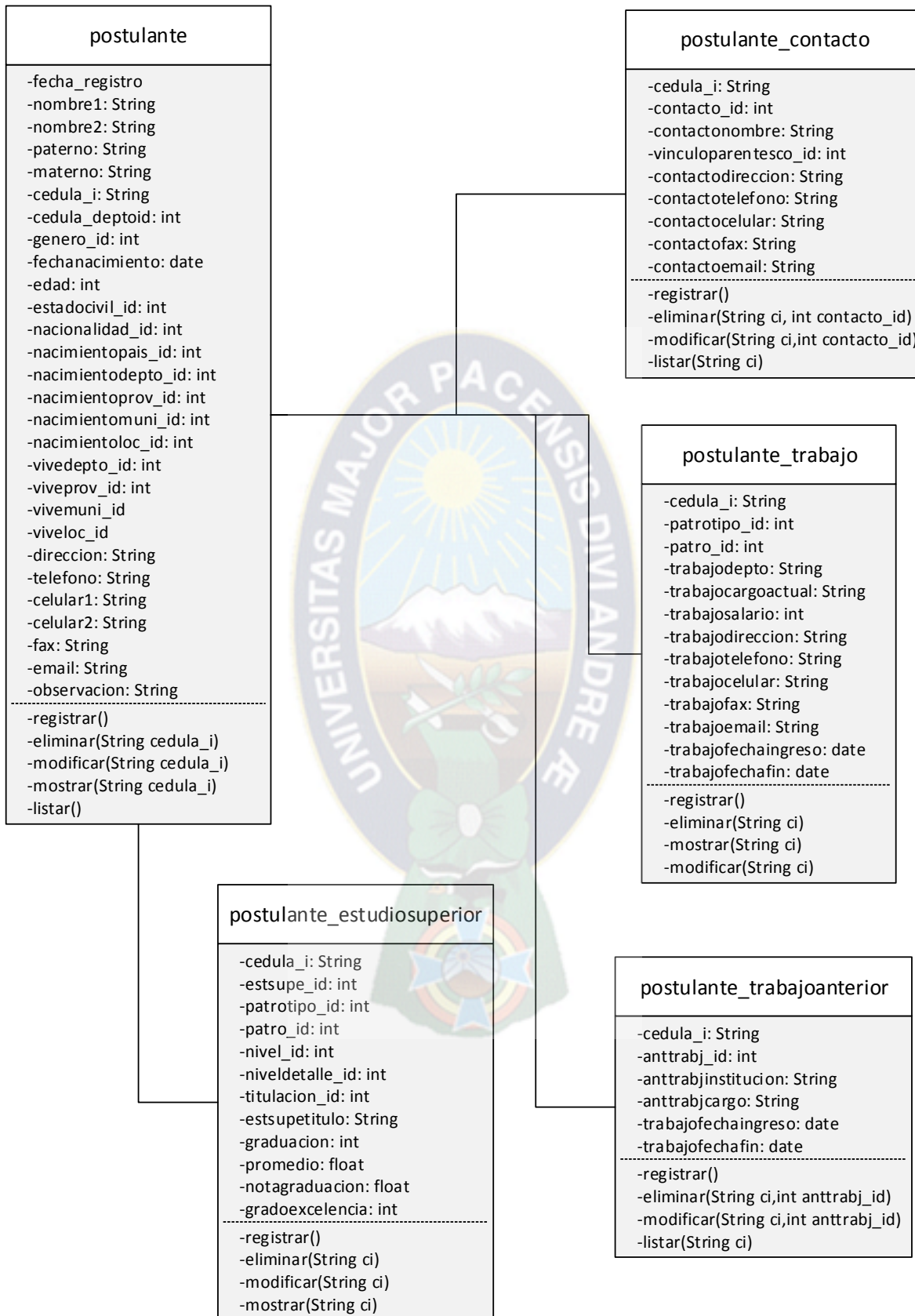


Figura 3.13: Modelo de estructura – Modulo de Postulantes

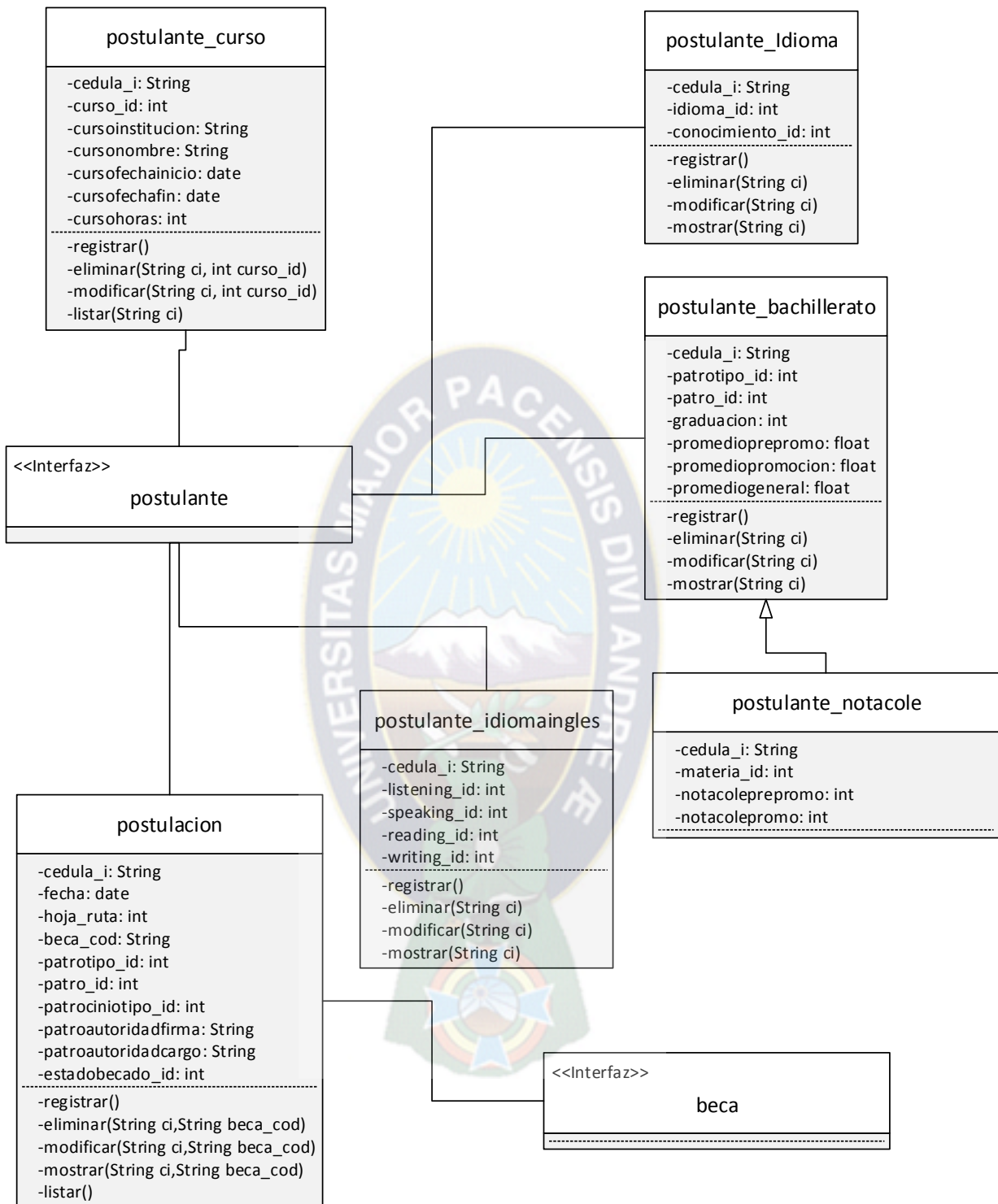


Figura 3.14: Modelo de estructura – Modulo de Postulantes

## c) MODELO DE HIPERTEXTO

## i) MODELO DE NAVEGACION

En la figura 3.15 podemos apreciar el modelo de navegación a través del cual se puede organizar el hipertexto referente al módulo de postulantes, este comprende al menú de postulante y los links hacia las diferentes áreas que componen al módulo de postulantes.

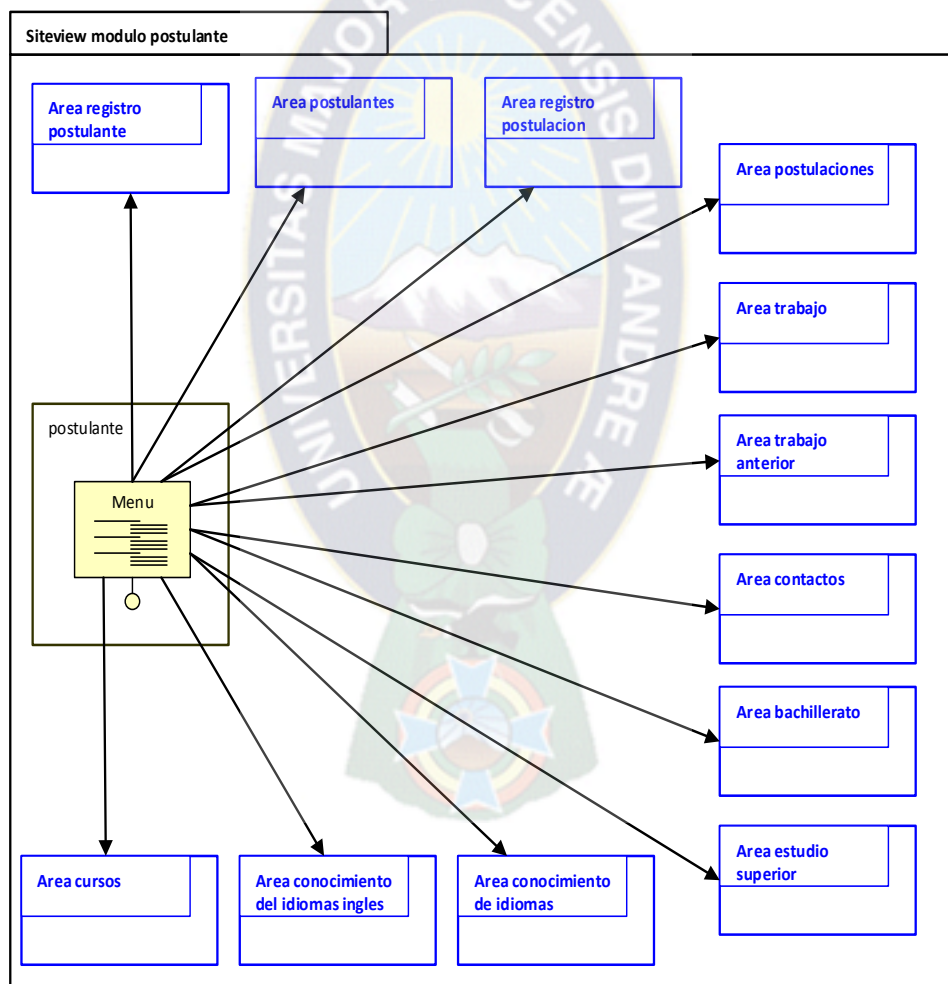


Figura 3.15: Modelo de Navegación – Modulo de Postulantes

Considérese a un usuario que ingreso al sistema el cual por medio del menú principal elige la opción postulante, en seguida se despliega esta opción con los enlaces de

registro de postulante, lista de postulantes, registro de postulación, lista de postulaciones, registro de trabajo, registro de trabajo anterior, registro de contactos de emergencia del postulante, registro de bachillerato, registro de estudios superiores, registro de conocimiento de idiomas, registro de nivel de conocimiento del idioma inglés y registro de curso. Para el menú se utiliza una unidad índice la cual contiene los enlaces no contextuales que se conectan hacia contenedores que conforman los nodos de navegación del módulo de postulantes.

## ii) MODELOS DE COMPOSICION

A continuación describimos algunos de los nodos del modelo de navegación del módulo de postulante más representativos como ser: área registrar postulante (registrar) y área postulante (listar, mostrar, modificar, eliminar), los cuales comprenden el uso de operaciones que sirven para expresar como se manejan los datos.

- ▲ Modelo de composición de registro de postulante (área registrar postulante) este modelo se detalla en la figura 3.16. Considérese a un usuario con permisos de escritura en el sistema. El proceso comienza observándose la interfaz de registro de postulante donde el usuario llena los campos del formulario y solicita el registro del mismo, acción la cual lleva al sistema a consultar al sí existe un postulante que posea la cedula de identidad ingresada. Esta búsqueda se realiza empleando una unidad de consulta simple la cual revisa en la tabla postulantes alguna ocurrencia de la cedula de identidad ingresada en el formulario en cuestión. Si existe algún postulante que concuerde con esos datos se muestra un mensaje indicando la existencia de la misma. De lo contrario una unidad de registro o ingreso de datos se encarga de registrar los datos de un nuevo postulante a la base de datos. Luego el sistema nos redirección a la página de bienvenida. En caso de que ocurra algún inconveniente durante todo el proceso la aplicación mostrara un mensaje al usuario indicando que corrija la información ingresada y la lista de los datos a corregir.
- ▲ Modelo de composición de lista de postulantes (área postulantes): considere que se encuentra en la interfaz de la lista de postulantes registrados, la cual presenta opciones de búsqueda, modificación, eliminación y mostrar postulante, como se puede observar en la figura 3.17, la cual explica detalladamente el funcionamiento de esta opciones.

Se presenta la lista de postulantes en una unidad de índice por medio de la cual podemos seleccionar algún postulante para realizar alguna operación, esta lista se puede desplazar a otra utilizando la unidad de scroller; opción de búsqueda: esta operación resulta por no encontrarse la cooperación deseada, para esto introducimos datos de búsqueda en un campo de entrada para esto empleamos una unidad de entrada buscar en la lista de postulantes deseado; opción de eliminación: considérese al usuario con permisos de eliminación, este proceso comienza seleccionando un postulante para su eliminación esto lleva a la aplicación a verificar si existe alguna postulación asociada a al postulante seleccionado este operación se realiza empleando dos unidades de consulta simple las cuales verifica en las tabla postulación la cedula de identidad seleccionada de encontrar una coincidencia se muestra un mensaje que indica que no es posible eliminar al postulante de los contrario una unidad de eliminación procede a la eliminación del postulante seleccionado; opción de mostrar: la selección de un postulante de la lista para ver sus datos nos lleva a una página en la cual nos muestra todos los datos registrados respecto a dicho postulante, para esto se emplea una unidad de datos simple la cual contiene todos los datos relacionados con la cedula de identidad del postulante que recibe como parámetro, la cual carga los datos de la cooperación a una unidad de entrada de datos.

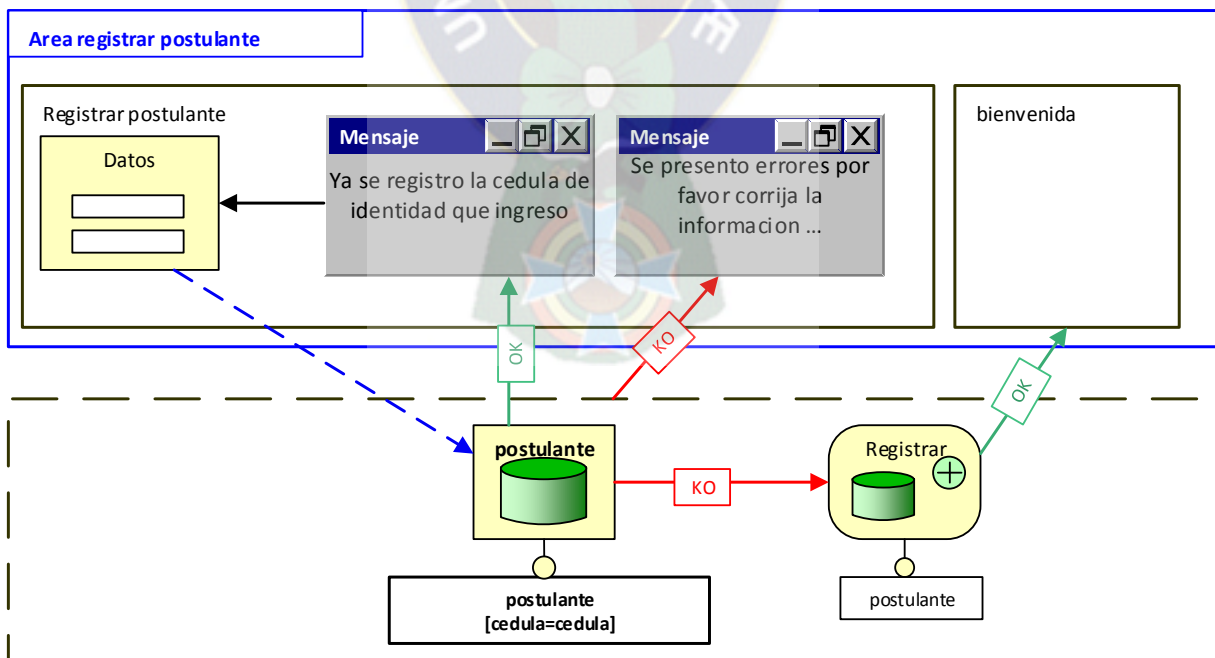


Figura 3.16: Modelo de Composición – Área Registrar Postulante



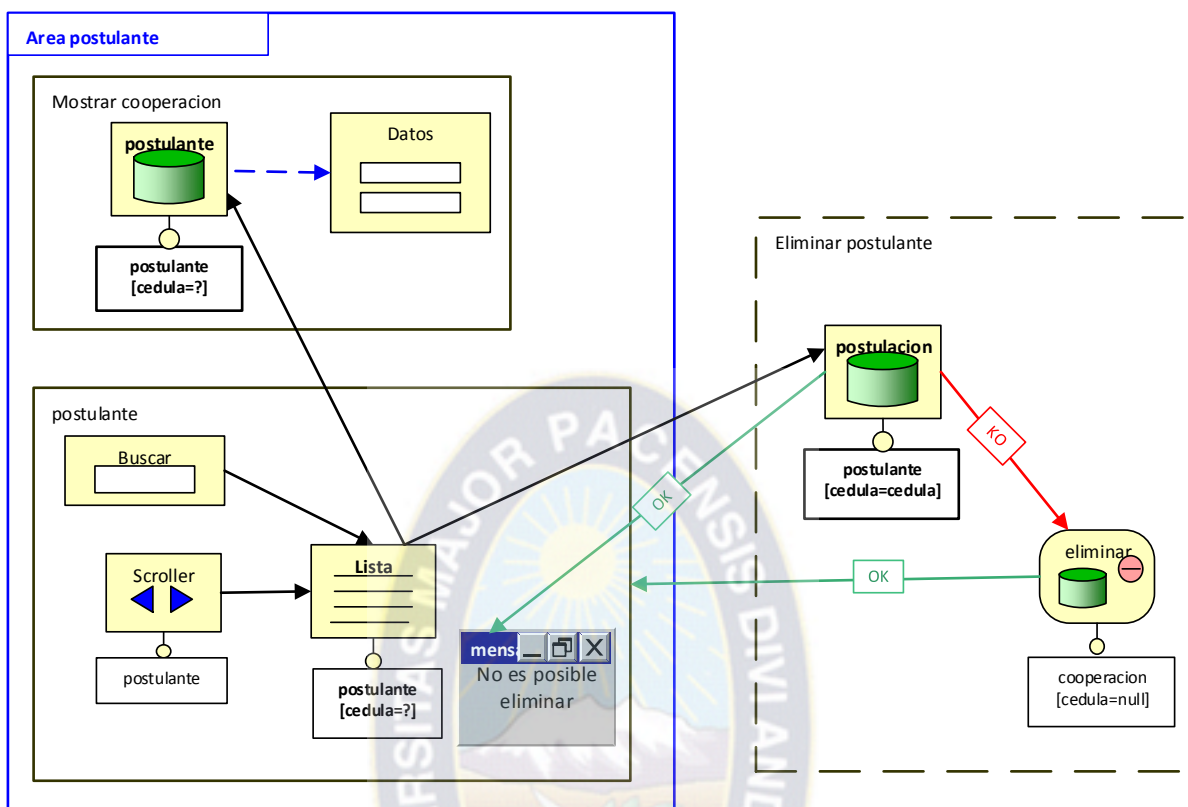


Figura 3.17: Modelo de Composición – Área Postulante

### 3.3.3. PRUEBAS

En esta fase haremos uso de las pruebas de aceptación las cuales nos permiten evaluar las historias de usuario creadas. Estas se las realizo conjuntamente con el equipo de trabajo para su posterior evaluación.

| CASO DE PRUEBA DE ACEPTACION   |                           |
|--|---------------------------|
| Código: 01   | Historia de usuario: N° 1 |
| Nombre: Registro de cooperación  |                           |
| Descripción: realizar la prueba de aceptación de la HU1  |                           |
| Condiciones de ejecución:  |                           |
| <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución.</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul> |                           |
| Entrada / pasos de ejecución:  |                           |

|  |
|--|
| <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de cooperación por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |
| Resultado esperado: el sistema verifica y valida todos los datos ingresados realizando consultas de la sigla de cooperación y el nombre para anular la duplicidad de información   |
| Evaluación de la prueba: aceptada  |

Tabla 3.90: Prueba de Aceptación: Registro de Cooperación

| CASO DE PRUEBA DE ACEPTACION   |                           |
|--|---------------------------|
| Código: 02   | Historia de usuario: N° 2 |
| Nombre: Registro de programa de cooperación  |                           |
| Descripción: realizar la prueba de aceptación de la HU2  |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>  |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de programa de cooperación por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |                           |
| Resultado esperado: el sistema verifica y valida todos los datos ingresados consultando el nombre de programa para tener nombre únicos de programas de cooperación.  |                           |
| Evaluación de la prueba: aceptada  |                           |

Tabla 3.91: Prueba de Aceptación: Registro de Programa de Cooperación

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 03  | Historia de usuario: N° 3 |
| Nombre: Registro de correspondencia   |                           |
| Descripción: realizar la prueba de aceptación de la HU3   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul> |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de correspondencia por medio del menú principal</li> </ul>                              |                           |

|   |
|---|
| - Debe hacer el llenado de todos los campos del formulario puesto que todos son obligatorios y por ultimo guardar   |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos la gestión y hoja ruta para evitar la duplicidad registros de correspondencia. |
| Evaluación de la prueba: aceptada   |

Tabla 3.92: Prueba de Aceptación: Registro de Correspondencia

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 04  | Historia de usuario: N° 4 |
| Nombre: Registro de convocatoria  |                           |
| Descripción: realizar la prueba de aceptación de la HU4   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de convocatoria por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |                           |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos la gestión y hoja ruta los cuales deben estar registrados. Responder a los campos de selección con eventos de datos en el formulario en los casos establecidos.  |                           |
| Evaluación de la prueba: aceptada   |                           |

Tabla 3.93: Prueba de Aceptación: Registro de Convocatoria

| CASO DE PRUEBA DE ACEPTACION   |                           |
|--|---------------------------|
| Código: 05   | Historia de usuario: N° 5 |
| Nombre: Ingreso de usuario   |                           |
| Descripción: realizar la prueba de aceptación de la HU5  |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe situarse en el formulario de autenticación del sistema</li> </ul>                               |                           |
| Resultado esperado: el sistema verifica y valida el usuario y contraseña ingresados, en el caso de no estar registrado o no estar habilitado el sistema muestra los mensajes correspondientes. |                           |
| Evaluación de la prueba: aceptada  |                           |

Tabla 3.94: Prueba de Aceptación: Ingreso de Usuario

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 06  | Historia de usuario: N° 6 |
| Nombre: Registro de patrocinador  |                           |
| Descripción: realizar la prueba de aceptación de la HU6   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de patrocinador por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |                           |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos el nombre de patrocinador para su respectiva validación en el caso de duplicidad. Responder a los campos de selección con eventos de datos en el formulario en los casos establecidos.                                 |                           |
| Evaluación de la prueba: aceptada   |                           |

Tabla 3.95: Prueba de Aceptación: Registro de patrocinador

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 07  | Historia de usuario: N° 7 |
| Nombre: Registro de postulante  |                           |
| Descripción: realizar la prueba de aceptación de la HU7   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de postulante por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |                           |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos la cedula de identidad del postulante para su respectiva validación en el caso de duplicidad. Responder a los campos con eventos de datos en el formulario en los casos establecidos.                                |                           |
| Evaluación de la prueba: aceptada   |                           |

Tabla 3.96: Prueba de Aceptación: Registro de postulante

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 08  | Historia de usuario: N° 8 |
| Nombre: Registro de formación del postulante  |                           |
| Descripción: realizar la prueba de aceptación de la HU8   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de formación del postulante por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul> |                           |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos la cedula de identidad del postulante ya este debe estar registrado. Responder a los campos con eventos de datos en el formulario en los casos establecidos.   |                           |
| Evaluación de la prueba: aceptada   |                           |

Tabla 3.97: Prueba de Aceptación: Registro de Formación del Postulante

| CASO DE PRUEBA DE ACEPTACION  |                           |
|---|---------------------------|
| Código: 09  | Historia de usuario: N° 9 |
| Nombre: Registro de postulación   |                           |
| Descripción: realizar la prueba de aceptación de la HU9   |                           |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul>   |                           |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder al registro de postulación por medio del menú principal</li> <li>- Debe hacer el llenado de todos los campos obligatorios del formulario y por ultimo guardar</li> </ul>                                    |                           |
| Resultado esperado: el sistema verifica y valida los datos ingresados consultando a la base de datos la cedula de identidad del postulante este debe estar registrado y el código de convocatoria e cual no debe estar asociado al postulante para un nuevo registro. Responder a los campos con eventos de datos en el formulario en los casos establecidos. |                           |
| Evaluación de la prueba: aceptada   |                           |

Tabla 3.98: Prueba de Aceptación: Registro de Postulación

| CASO DE PRUEBA DE ACEPTACION  |                            |
|---|----------------------------|
| Código: 10  | Historia de usuario: N° 10 |
| Nombre: Reportes  |                            |
| Descripción: realizar la prueba de aceptación de la HU10  |                            |
| Condiciones de ejecución: <ul style="list-style-type: none"> <li>- Servidor PostgreSQL Instalado</li> <li>- Aplicación en ejecución</li> <li>- El usuario debe estar registrado, tener como estado habilitado y tener permiso de registro.</li> </ul> |                            |
| Entrada / pasos de ejecución: <ul style="list-style-type: none"> <li>- Primero el usuario debe autenticarse en el sistema</li> <li>- Debe acceder a la lista de oferta de becas por medio del menú principal</li> </ul>                               |                            |
| Resultado esperado: el sistema emite los reportes de las ofertas de becas vigentes.   |                            |
| Evaluación de la prueba: aceptada   |                            |

Tabla 3.99: Prueba de Aceptación: Reportes



## CAPITULO IV

### CALIDAD DEL SOFTWARE

#### 4.1. INTRODUCCION

Para medir la calidad del Sistema Web de Gestión y Difusión de Becas, caso: EGPP (Escuela de Gestión Publica Plurinacional) se empleó métricas de calidad en función al estándar ISO-9126.

#### 4.2. FUNCIONALIDAD

La funcionalidad es medida a través del punto función (PF), como se puede apreciar en la tabla 4.1.

La cuenta total debe ajustarse utilizando la siguiente ecuación:

$$PF = \text{cuenta total} \times (0.65 + 0.01 \times \sum (Fi)) \quad (1)$$

| PARAMETRO DE MEDIDA          | CUENTA |   | FACTOR DE PONDERACION |       |          | = | TOTAL |      |
|------------------------------|--------|---|-----------------------|-------|----------|---|-------|------|
|                              |        |   | SIMPLE                | MEDIO | COMPLEJO |   |       |      |
| Nro de entradas de usuario   | 51     | X | 3                     | 4     | 6        | = | 204   |      |
| Nro de salidas de usuario    | 47     | X | 3                     | 5     | 7        | = | 235   |      |
| Nro de peticiones de usuario | 49     | X | 3                     | 4     | 6        | = | 196   |      |
| Nro de archivos              | 72     | X | 7                     | 10    | 15       | = | 720   |      |
| Nro de interfaces externas   | 4      | X | 5                     | 7     | 10       | = | 28    |      |
| Cuenta total                 |        |   |                       |       |          |   |       | 1383 |

Tabla 4.1: Calculo de Punto Función

A continuación para ajustar la complejidad de los puntos función según el factor de ajuste veremos el resultado de la tabla 4.2

| #                        | FACTORES DE AJUSTE DE VALOR  | valor |
|--------------------------|--|-------|
| 1                        | ¿El sistema requiere respaldo y recuperación confiables?   | 4     |
| 2                        | ¿Se requiere comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?               | 3     |
| 3                        | ¿Existen funciones de procesamiento distribuidas?  | 1     |
| 4                        | ¿El desempeño es crucial?  | 4     |
| 5                        | ¿El sistema correrá en un entorno operativo existente y enormemente utilizado?   | 5     |
| 6                        | ¿El sistema requiere entrada de datos en línea?  | 5     |
| 7                        | ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones? | 5     |
| 8                        | ¿Los archivos lógicos internos se actualizan en línea?   | 5     |
| 9                        | ¿Las entradas, salidas, los archivos o consultas son complejos?  | 2     |
| 10                       | ¿El procesamiento interno es complejo?   | 2     |
| 11                       | ¿El código se diseña para ser reutilizable?  | 3     |
| 12                       | ¿La conversión y la instalación se incluyen en el diseño?  | 3     |
| 13                       | ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?   | 4     |
| 14                       | ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?  | 5     |
| Total ( $\Sigma (F_i)$ ) |  | 51    |

Tabla 4.2: Factores de ajuste de valor

Calculo del punto función reemplazando los datos obtenidos en la ecuación 1, tenemos

$$PF=1383 \times (0.65 + 0.01 \times 51)$$

$$PF=1604.28$$

Ahora realizamos la comparación de funcionalidad de sistema con el PF máximo que se puede alcanzar es decir que  $\Sigma (F_i)=70$

$$PF=1383 \times (0.65 + 0.01 \times 70)$$

$$PF=1867.05$$



Con máximos valores de ajuste a la complejidad se tiene que la funcionalidad real es:

$$\text{FUNCIONALIDAD} = (1604.28 / 1867.05) * 100$$

$$\text{FUNCIONALIDAD} = 85.93\%$$

El Sistema Web de Gestión y Difusión de Becas, caso: EGPP tiene una funcionalidad del 85.93%

#### 4.3. USABILIDAD

La usabilidad es medida a través de las respuestas del cuestionario en la tabla 4.3, considerando la escala de evaluación del 1 a 5 (1=pésimo, 2=malo, 3=regular, 4=bueno, 5 muy bueno)

| Nro              | PREGUNTAS SOBRE LA PERCEPCION DE USABILIDAD   | Valor promedio |
|------------------|---|----------------|
| 1                | ¿El sistema es utilizable sin ayuda o enseñanza continúa?                                     | 4.6            |
| 2                | ¿Las reglas de interacción ayudan a un usuario preparado a trabajar con eficiencia?           | 3.8            |
| 3                | ¿Los mecanismos de interacción se hacen más flexibles a medida que los usuarios conocen más?  | 3.8            |
| 4                | ¿Se ha adaptado el sistema al ambiente físico y social en el que se usara?                    | 4.2            |
| 5                | ¿El usuario está al tanto del estado del sistema? ¿Sabe en todo momento en donde esta?        | 4              |
| 6                | ¿La interfaz está estructurada de manera de manera lógica y consistente?                      | 4              |
| 7                | ¿Los mecanismos, iconos y procedimientos de interacción son consistentes en toda la interfaz? | 4              |
| 8                | ¿La interacción prevé errores y ayuda el usuario a corregirlos?                               | 4.6            |
| 9                | ¿La interfaz es tolerante a los errores que se cometen?                                       | 3.8            |
| 10               | ¿Es sencilla la interacción?  | 4.2            |
| Total percepción |   | 41             |

Tabla 4.3: Percepción de Usabilidad

La encuesta se la realizó 5 funcionarios de la EGPP de área de becas los cuales interactúan con el sistema para realizar sus funciones.

Con el resultado obtenido y utilizando la siguiente ecuación tenemos:

$$\text{USABILIDAD} = \left[ \frac{\text{Total percepción}}{n} \cdot 100 / 10 \right]$$

Reemplazando con  $n=5$  por ser 5 las personas encuestadas se obtenemos:

$$\text{USABILIDAD} = \left[ \frac{41}{5} \cdot 100 / 10 \right]$$

$$\text{USABILIDAD} = 82\%$$

Por lo tanto concluimos que la facilidad de uso del sistema es del 82%.

#### 4.3. CONFIABILIDAD

La medida de confiabilidad es el tiempo medio entre fallas (TMEF), y usando la siguiente ecuación:

$$\text{TMEF} = \text{TMPF} + \text{TMPR} \quad (1)$$

Donde:

TMPF es el tiempo medio para la falla

TMPR es el tiempo medio para la reparación

Para el cálculo del TMPR se tomó la siguiente ecuación:

$$\text{TMPR} = \text{TMAC} + \text{TMIC} + \text{TMPC} + \text{TMDC} \quad (2)$$

Donde:

TMAC = tiempo medio de analizar los cambios

TMIC = tiempo medio de implementar los cambios

TMPC = tiempo medio de probar los cambios

TMDC = tiempo medio de distribuir los cambios

Tomando una muestra de 5 fallas las cuales se presentaron hasta fecha, y realizando una media aritmética de tiempos en horas de estas se tiene los siguientes resultados en la tabla 4.4.

|               | TMAC | TMIC | TMPC | TMDC |
|---------------|------|------|------|------|
| $\Sigma ti/5$ | 5/5  | 10/5 | 10/5 | 5/5  |
|               | 1    | 2    | 1    | 1    |

Tabla 4.4: Medición de parámetros del TMPR

Reemplazando los datos obtenido en la ecuación 2 tenemos:

$$\text{TMPR} = 1 \text{ Hrs} + 2 \text{ Hrs} + 1 \text{ Hrs} + 1 \text{ Hrs}$$

$$\text{TMPR} = 5 \text{ Hrs}$$

De las 5 muestras que se hicieron se estima una falla en aproximadamente 20 horas de uso continuas, esto se puede expresar de la siguiente manera:

$$\text{TMPF} = 20 \text{ Hrs}$$

Ahora reemplazando el TMPF y el TMPR en la ecuación 1, tenemos

$$\text{TMEF} = 20 \text{ Hrs} + 5 \text{ Hrs}$$

$$\text{TMEF} = 25 \text{ Hrs}$$

Ahora calculamos la disponibilidad usando la siguiente ecuación:

$$\text{DISPONIBILIDAD} = \text{TMPF} / \text{TMPF} + \text{TMPR} \times 100\% \quad (3)$$

Reemplazando los datos en la ecuación 3 tenemos:

$$\text{DISPONIBILIDAD} = [20 \text{ Hrs} / (20 \text{ Hrs} + 5 \text{ Hrs})] \times 100\%$$

$$\text{DISPONIBILIDAD} = 80 \%$$

Recordemos que la confiabilidad consta de la cantidad de tiempo que el software se encuentra disponible para su uso, entonces se tiene que en un lapso de 25 horas se tiene un valor de 80% de éxito de funcionamiento del sistema.

#### 4.4. MANTENIBILIDAD

Para la mantenibilidad se toma en cuenta al índice de madurez de software (IMS), el cual se calcula de la siguiente forma:

$$\text{IMS} = \text{Mt} - (\text{Fa} + \text{Fc} + \text{Fd}) / \text{Mt}$$

Donde:

Mt: número de módulos en la liberación actual.

Fc: número de módulos en la liberación actual que cambiaron.

Fa: número de módulos en la liberación actual que se agregaron.

Fd: número de módulos en la liberación anterior que se borraron en la liberación actual.

A partir del sistema se obtuvieron los siguientes valores necesarios para el cálculo del IMS:

$$\text{Mt} = 4, \text{Fc} = 1, \text{Fa} = 0, \text{Fd} = 0$$

Reemplazando los datos en la ecuación del IMS tenemos:

$$\text{IMS} = [4 - (0 + 1 + 0)] / 4$$

$$\text{IMS} = 0.75$$

Sabiendo que conforme el IMS tiende a 1.0 el producto comienza a estabilizarse, por consiguiente el IMS es de 0.75% lo que quiere decir que es mantenible y no requiere de grandes esfuerzos para restablecerlo en caso de producirse un error o caída del sistema.

#### 4.5. PORTABILIDAD

Para la puntuación de portabilidad, se desarrollaron las siguientes preguntas como se muestra en la tabla 4.6 y considerando que el software pueda llevarse de un ambiente a otro.

| FACTOR DE AJUSTE  | VALOR |
|---|-------|
| Es capaz de ser adaptado a otros ambientes con facilidad<br>(instituciones similares) | 90    |
| Es fácil de instalar  | 90    |
| Estando en otro ambiente conserva su funcionalidad                                    | 100   |
| Es capaz de reemplazar a un software similar  | 95    |
| Total   | 93.75 |

Tabla 4.6: Factor de Ajuste de la Portabilidad

Tras los resultados de la tabla 4.6, podemos decir que tenemos un 93.75% de portabilidad lo que significa que el producto software tiene la capacidad de poder ejecutarse en diferentes plataformas sin problemas.



## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

El Sistema Web de Gestión y Difusión de Becas, caso: EGPP (Escuela de Gestión Pública Plurinacional), de la Unidad de Becas de la EGPP cuenta con una herramienta personalizada que ofrece información completa, actualizada y confiable, alcanzándose los siguientes resultados:

- ▲ Se trabajó de acuerdo al decreto supremo 1874, empleando herramientas de software libre.
- ▲ Cuenta con implementación de la verificación y validación en los formularios anulándose el ingreso de duplicidad de información en los registros
- ▲ Se optimizó los campos de los formularios, manteniendo en estos los necesarios, logrando así optimizar la respuesta en el cargado y envío de los formularios.
- ▲ La normalización de la base de datos del módulo de becas se realizó con éxito, migrando los datos a la base de datos normalizada sin pérdida de información.
- ▲ Se cuenta con un formulario para el registro de nuevos usuarios (funcionarios de la EGPP). El ingreso al sistema para los usuarios registrados y autorizados es mediante su usuario y contraseña, una vez que se ingresa al sistema cada usuario tiene permisos sobre los registros que son establecidos por el administrador de sistema, quien tiene la facultad de dar la autorización del ingreso a los usuarios registrados, además el sistema registra automáticamente a la base de datos las acciones de los usuarios sobre los registros.
- ▲ Para la subida de archivos al sistema se realiza a la verificación y validación de la extensión, tamaño y nombre del archivo.
- ▲ El sistema realiza de manera automática el registro de las descargas de los archivos de información de las convocatorias.

## 5.2. RECOMENDACIONES

Las recomendaciones que se tiene son las siguientes:

- ▲ Se recomienda realizar la implementación de mecanismos de control en la base de datos para el seguimiento de acciones sobre las tablas.
- ▲ Para la utilización de alguna nueva librería estudiar la compatibilidad con el sistema.
- ▲ Para trabajos de reemplazo de sistemas se recomienda realizar un estudio minucioso de las interfaces, procesamiento, base datos y la funcionalidad que ofrece.



## BIBLIOGRAFIA

- Ablett Erick, et al. (2013). Programación Extrema (Wiki). Recuperado de:  
<http://programacion-extrema.wikispaces.com>
- Artech (2008). Primeros pasos con Genexus 9.0. Recuperado de:  
[http://www.exa.unicen.edu.ar/catedras/modysim/tutoriales/primeros\\_pasos\\_con\\_genexus\\_90.pdf](http://www.exa.unicen.edu.ar/catedras/modysim/tutoriales/primeros_pasos_con_genexus_90.pdf)
- Copacondo R. J. (2009). Portal Web para el Arzobispado de Nuestra Señora de La Paz (Proyecto de Grado). Universidad Mayor de San Andrés, Carrera de Informática. La Paz – Bolivia.
- Decreto Supremo 0212 (2014). Promulgación del Decreto Supremo 0212. Gaceta oficial del Estado Plurinacional de Bolivia. Recuperado de:  
<http://www.gacetaoficialdebolivia.gob.bo/normas/buscar/212>
- Decreto Supremo 1874 (2014). Promulgación del Decreto Supremo 1874. Gaceta oficial del Estado Plurinacional de Bolivia. Recuperado de:  
<http://www.gacetaoficialdebolivia.gob.bo/normas/listadonor/11/page:3>
- Decreto Supremo 1793 (2014). Promulgación del Decreto Supremo 1793, disposición transitoria quinta. Gaceta oficial del Estado Plurinacional de Bolivia. Recuperado de:  
<http://www.gacetaoficialdebolivia.gob.bo/normas/listadonor/11/page:4>
- Escalona M J. (2004). Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software (Tesis Doctoral). Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla. Recuperado de:  
<http://www.lsi.us.es/docs/doctorado/tesis/tesis.pdf>
- GeneXus. (2014). *Wikipedia, La enciclopedia libre*. Recuperado de:  
desde <http://es.wikipedia.org/w/index.php?title=GeneXus&oldid=72735318>.
- GeneXus International (2012). Visión General de Genexus. Recuperado de:  
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCoQ>



FjAA&url=http%3A%2F%2Fwww.genexus.com%2Ffiles%2Fwp-vision-general%3Fes&ei=ftlvU6GdN8qwsQSijlGYBg&usg=AFQjCNHnt42-5CS3Z7UTI663KPH1JEFkfQ&bvm=bv.66111022,d.cWc&cad=rja

- Jimenez S. Jose C. (2001). Tecnología y Sociedad: Comercio Electrónico Internet y su Seguridad. Recuperado de:  
<http://www.coit.es/publicac/publbit/bit126/sociedad4.htm>
- Letelier P. y Penadés C. (2004). Metodologías Ágiles en el Desarrollo de Software: eXtreme Programing (XP). Universidad Politécnica de Valencia. Recuperado de:  
<http://www.willydev.net/descargas/masyxp.pdf>
- Castellaro M, et al. (2009). Hacia la Ingeniería de Software Seguro. Facultad Regional Santa Fe - Universidad Tecnológica Nacional. Recuperado de:  
[http://sedici.unlp.edu.ar/bitstream/handle/10915/21332/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/21332/Documento_completo.pdf?sequence=1)
- MD5. (2014). *Wikipedia*, La enciclopedia libre. Recuperado de: <http://es.wikipedia.org/w/index.php?title=MD5&oldid=76607511>.
- Normalización de Bases de Datos. (2003). MySQL Hispano. Recuperado de:  
<http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf>
- Pressman, Roger S. (2010). Ingeniería de Software un Enfoque Práctico. Séptima Edición España: McGraw – Hill.
- Salazar Lucho (2013). Escribiendo Historias de Usuario Altamente Efectivas. Recuperado de: <http://www.gazafatonarioit.com/2013/08/escribiendo-historias-de-usuario.html>
- Silberschatz Abraham et al. (2002). Fundamentos de Bases de datos. Cuarta edición España: McGraw – Hill.

- Sistema de información. (2014). Wikipedia, La enciclopedia libre. Recuperado de:  
[http://es.wikipedia.org/w/index.php?title=Sistema\\_de\\_informaci%C3%B3n&oldid=74118459](http://es.wikipedia.org/w/index.php?title=Sistema_de_informaci%C3%B3n&oldid=74118459).
- Sistema web EGPP (2014). Sistema de información web EGPP. Recuperado de:  
<http://www.egpp.gob.bo/>
- WebML staff (2004). Model – Driven Design of Web Applications: WebML.  
Recuperado de: <http://www.webml.org/>
- Yali N. F. (2011). Sistema de Control y Evaluación para el Proyecto C. D. I. – BO – 139 (Proyecto de Grado). Universidad Mayor de San Andrés, Carrera de Informática. La Paz – Bolivia.

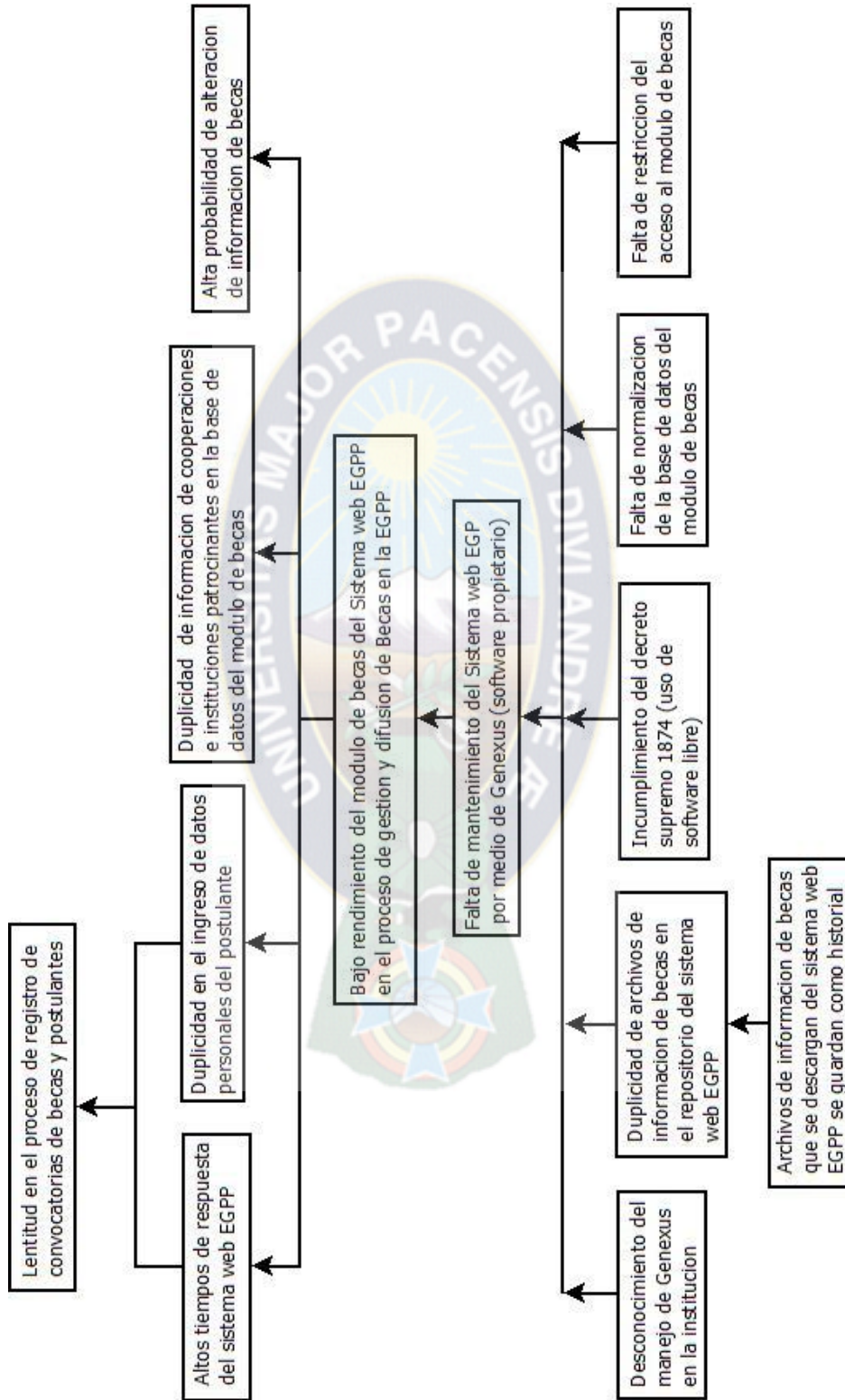


# ANEXOS



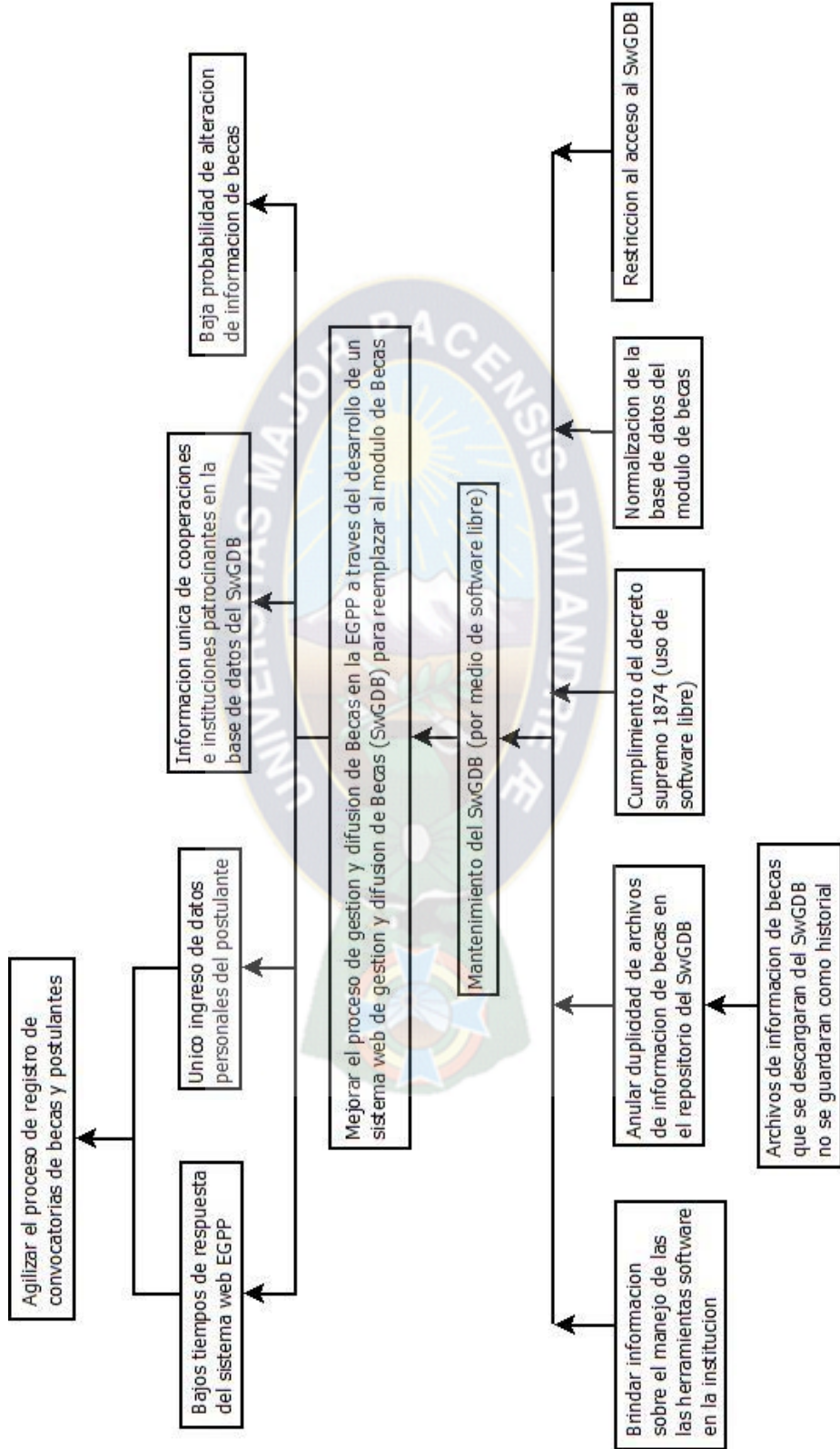
ANEXO A

ÁRBOL DE PROBLEMAS



ANEXO B

ARBOL DE OBJETIVOS



## ANEXO C: MARCO LOGICO

| <u>RESUMEN NARRATIVO</u>   | <u>INDICADORES OBJETIVAMENTE VERIFICABLES</u>   | <u>MEDIOS DE VERIFICACION</u>  | <u>SUPUESTOS</u>   |
|--|---|--|--|
| <b>FIN</b><br>Agilizar el proceso de gestión y difusión de información de becas en la EGPP y su coherencia.  | Reducción del tiempo de registro de información de convocatorias de becas y postulantes a partir del mes de diciembre 2014.       | evaluación de tiempos de respuesta entre el módulo de becas y el SwGDB.  | Afluencia de postulantes a las diferentes ofertas de becas para el exterior  |
| <b>PROPOSITO</b><br>Mejorar el proceso de gestión y difusión de información de becas en la EGPP a través del desarrollo de un sistema web de gestión y difusión de becas (SwGDB), para reemplazar al módulo de becas del sistema web EGPP.   | Mayor número de registros de convocatorias de becas y postulantes.  | Los funcionarios del área de becas de la EGPP hacen uso del SwGDB.<br>Certificación de la EGPP.<br>Aval de Tutor y Asesor (Docente Carrera de Informática UMSA).   | Se tiene el software y hardware adecuados.   |
| <b>PRODUCTOS</b><br>1. Diseño y Desarrollo del SwGDB.<br>2. Uso de herramientas de software libre para la elaboración del SwGDB.   | Nivel de satisfacción de los funcionarios del área de becas y de los postulantes.   | Revisión por el Tutor, Asesor y funcionarios responsables del área de becas y de la Unidad de investigación y desarrollo de la EGPP.   | Los equipos de computación cuentan con el hardware y software necesarios para el buen funcionamiento de SwGDB.                             |
| <b>ACTIVIDADES</b><br>1.1. Análisis de la situación actual del módulo de becas.<br>1.1.1. Identificación de las tareas funcionales del módulo de becas (estudio del procesamiento, interfaz y base de datos).<br>1.2. Normalización la base de datos del módulo de becas.<br>1.3. Diseño de SwGDB.<br>1.4. Codificación y pruebas del SwGDB.<br>1.4.1. Crear un login para el ingreso al SwGDB.<br>1.4.2. Crear un proceso para guardar datos de fecha y hora de descarga de archivos de información de becas.<br>2. documentar información de las herramientas software y elaboración del manual de usuario | 1.1. semanas 2<br>1.1.1. semanas 3<br>1.2. semanas 1<br>1.3. semanas 3<br>1.4. semanas 18<br>2. semanas 1<br><br>Total semanas 28 | 1.1. Entrevistas con funcionarios de la EGPP.<br>1.1.1. Interacción con el módulo de becas y análisis de la información de la base de datos del módulo de becas.<br>1.2. Documentación.<br>1.3. Incrementos software operativos.<br>2. Manual técnico y Manual de usuario del SwGDB. | Se cuenta con el apoyo de los funcionarios de becas de la EGPP y se tiene acceso a la información de la base de datos del módulo de becas. |

