



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons  
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite  
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

### ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL DE BECARIOS Y EXBECARIOS DE UNA ASOCIACIÓN EDUCATIVA

Tesis para optar por el Título de Ingeniero Informático, que presenta el bachiller:

**Jessica Gutiérrez Zúñiga**

**Marllessi Marilyn Chanca De La Cruz**

**ASESOR: Ing. Carla Shirley Basurto Figueroa**

Lima, 23 de octubre de 2009

## RESUMEN

El presente proyecto de tesis tiene por objetivo realizar el análisis, diseño e implementación de un sistema de información web que automatice los procesos de negocio de una asociación educativa que brinda préstamos universitarios, en su sede de Perú.

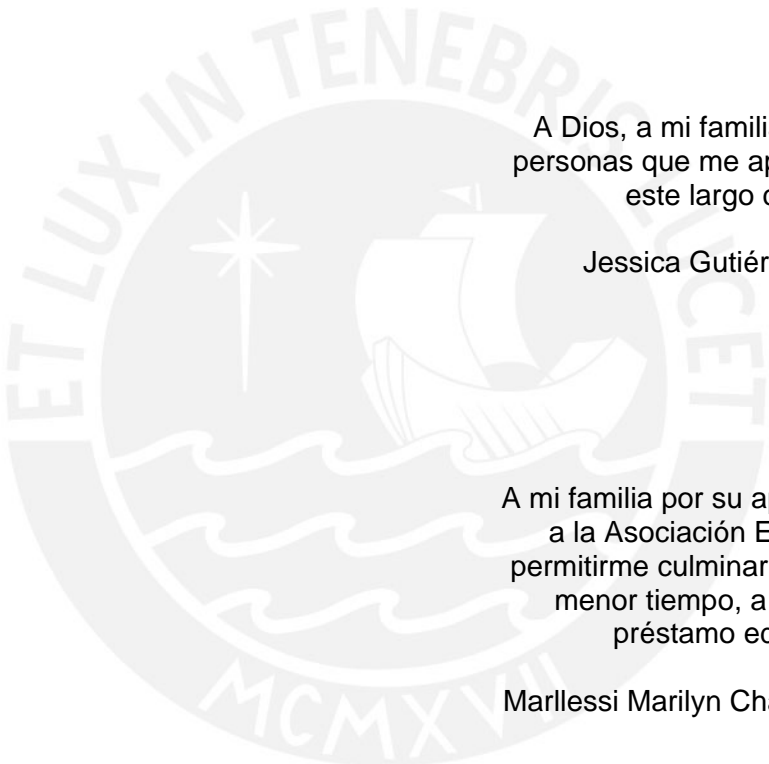
La referida asociación educativa es una entidad internacional privada que recibe donaciones de un directorio suizo y que tiene como objetivo prestar dicho dinero a alumnos universitarios de buen rendimiento académico pero de recursos limitados que impiden culminar sus estudios de pregrado satisfactoriamente.

Actualmente, los procesos que realiza dicha entidad en su sede de Perú son manuales, teniendo como consecuencia: lentitud de atención a los postulantes, poca coordinación con las instituciones educativas con las que trabaja, deficiente gestión de la información de los becarios y ex becarios, así como errores de registro de deudas y pagos realizados, entre otros.

Por ello, tras constantes reuniones con los usuarios finales y un arduo análisis de los procesos a automatizar, la solución planteada permitirá la buena interacción y comunicación entre la asociación educativa, las instituciones de educación superior con las que trabaja, postulantes, becarios, ex-becarios y el directorio suizo, de forma organizada, transparente y sin tener que realizarse tantas reuniones para el envío de la información.

Dado que el presente proyecto tuvo como una de sus metas poner en marcha el sistema de información planteado en el menor tiempo posible, el equipo de desarrollo decidió aplicar metodologías ágiles, tanto para la Gestión del Proyecto como para la Gestión de Desarrollo de la Solución. Esto permitió que el proyecto finalice exitosamente con la aprobación de los representantes de la asociación educativa, tal es así que el sistema desarrollado ya se encuentra en fase a producción y permitirá la interacción entre los diferentes usuarios finales, incluyendo a 150 becarios y 400 ex-becarios, aproximadamente.

Por otro lado, se definió el uso de herramientas de *software libre* en la medida de lo posible. Así, se reducen los costos de adquisición para el desarrollo del proyecto y sólo se consideran los costos por horas trabajadas por el equipo de desarrollo.



A Dios, a mi familia y a todas las personas que me apoyaron durante este largo camino.

Jessica Gutiérrez Zúñiga

A mi familia por su apoyo constante y a la Asociación Educación por permitirme culminar mis estudios en menor tiempo, a través de un préstamo educativo.

Marlessi Marilyn Chanca De La Cruz

## Tabla de Contenido

Introducción .....	1
1. Generalidades .....	2
1.1. Definición del problema.....	2
1.2. Marco conceptual .....	3
1.3. Plan de proyecto .....	5
1.3.1. Diagrama Work Breakdown Structure .....	6
1.3.2. Planificación inicial .....	7
1.4. Estado del arte .....	8
1.5. Descripción y sustentación de la solución .....	10
2. Análisis .....	12
2.1. Metodología aplicada para el desarrollo de la solución .....	12
2.1.1. Aplicación de Extreme programming.....	12
2.2. Identificación de requerimientos .....	14
2.3. Análisis de la solución .....	23
2.3.1. Viabilidad del sistema.....	23
2.3.2. Análisis técnico – económico .....	23
2.3.3. Asignación de funciones.....	24
2.3.4. Restricciones de costo y tiempo.....	24
2.3.5. Definición del sistema.....	24
3. Diseño .....	30
3.1. Arquitectura de la solución.....	30
3.1.1. Definición de la arquitectura .....	30
3.1.2. Definición de componentes .....	32
3.1.3. Beneficios de la arquitectura .....	33
3.2. Diseño de interfaz gráfica .....	34
3.2.1. Criterios para el diseño de la interfaz gráfica .....	34
3.2.2. Estructura general del sitio .....	34
3.2.3. Modelo de ventanas .....	35
3.3. Diseño de la solución .....	37
3.3.1. Diagrama de clases de base de datos .....	37
3.3.2. Diagrama de clases de diseño .....	39
3.3.3. Diagrama de secuencia.....	42
4. Construcción .....	51
4.1. Construcción .....	51
4.1.1. Tecnologías.....	51
4.1.2. Frameworks.....	61
4.1.3. Patrones de diseño.....	64
4.2. Pruebas .....	67
4.2.1. Estrategia de pruebas .....	67
4.2.2. Tipos de prueba.....	67

4.2.3.	Catálogo de pruebas .....	67
4.2.4.	Reporte de pruebas.....	78
5.	Observaciones, conclusiones y recomendaciones .....	79
5.1.	Observaciones .....	79
5.2.	Conclusiones.....	80
5.3.	Recomendaciones y trabajos futuros.....	80
	Bibliografía .....	82

## Índice de Ilustraciones

Figura 1.1.	Diagrama general del proceso de admisión de nuevos becarios. ....	3
Figura 1.2.	WBS del proyecto de Tesis. ....	6
Figura 1.3.	Planificación inicial del proyecto de Tesis.....	7
Figura 1.4.	Diagrama de Gantt del proyecto de Tesis. ....	8
Figura 1.5.	Sistema de Administración de Becas Escolares - SABE.....	9
Figura 1.6.	Sistema de Becas - SISBE.....	9
Figura 2.1.	Actores del sistema. ....	26
Figura 2.2.	Diagrama de actividades del proceso de Postulación. ....	28
Figura 3.1.	Diseño de la arquitectura del sistema.....	31
Figura 3.2.	Diagrama de componentes de la arquitectura del sistema. ....	32
Figura 3.3.	Estructura general del sitio. ....	34
Figura 3.4.	Ventana de bienvenida.....	35
Figura 3.5.	Ventana de login.....	35
Figura 3.6.	Ventana de Menú de opciones. ....	36
Figura 3.7.	Ventana de tipo Mantenimiento. ....	36
Figura 3.8.	Ventana de tipo Operación. ....	37
Figura 3.9.	Diagrama de clases de base de datos. ....	38
Figura 3.10.	Diagrama de clases de diseño – Procesos de inicio, mitad y fin de ciclo. ....	39
Figura 3.11.	Diagrama de clases de diseño – Proceso de totalización de deuda.....	40
Figura 3.12.	Diagrama de clases de diseño – Proceso de postulación. ....	41
Figura 3.13.	Diagrama de clases de diseño – Proceso de ingresos por pagos de ex-becarios.....	42
Figura 3.14.	Diagrama de secuencias – Proceso de Postulación: Publicar Información. ....	43
Figura 3.15.	Diagrama de secuencias – Proceso de Postulación: Ingresar datos - Institución.....	44
Figura 3.16.	Diagrama de secuencias – Proceso de Postulación: Ingresar datos - Postulante.....	45
Figura 3.17.	Diagrama de secuencias – Proceso de Postulación: Modificar datos - Institución. ....	46
Figura 3.18.	Diagrama de secuencias – Proceso de Postulación: Seleccionar entrevistados. ....	47
Figura 3.19.	Diagrama de secuencias – Proceso de Postulación: Asignar entrevista. ....	48
Figura 3.20.	Diagrama de secuencias – Proceso de Postulación: Consultar entrevista. ....	48
Figura 3.21.	Diagrama de secuencias – Proceso de Postulación: Seleccionar postulante.....	49
Figura 3.22.	Diagrama de secuencias – Proceso de Postulación: Admitir postulante. ....	50
Figura 4.1.	Funcionamiento de JSP.....	60
Figura 4.2.	Componentes del patrón MVC. ....	65
Figura 4.3.	Componentes del patrón MVP.....	66

## Índice de Tablas

Tabla 1.1.	Comparativo de funcionalidades de algunas soluciones existentes. ....	10
Tabla 2.1.	Requerimientos funcionales.....	21
Tabla 2.2.	Requerimientos no funcionales.....	22
Tabla 2.3.	Estimación de costos de construcción del proyecto. ....	23
Tabla 2.4.	Asignación de funciones. ....	24
Tabla 3.1.	Descripción de componentes de la arquitectura. ....	32
Tabla 3.2.	Beneficios de la arquitectura.....	33
Tabla 4.1.	Comparación de lenguajes de programación. ....	53
Tabla 4.2.	Comparación de entornos de desarrollo. ....	55
Tabla 4.3.	Comparación de servidores. ....	58

Tabla 4.4. Comparación de gestores de base de datos. ....	59
Tabla 4.5. Comparación de lenguajes para creación de páginas web.....	61
Tabla 4.6. Comparación de frameworks de capa de presentación.....	63
Tabla 4.8. Comparación de patrones de diseño. ....	66
Tabla 4.9. Clases de equivalencia del caso de prueba Mantener Postulante.....	70
Tabla 4.10. Pruebas unitarias para el caso de pruebas Mantener Postulante.....	72
Tabla 4.11. Pruebas de integración del proceso de Postulación. ....	77
Tabla 4.12. Formato del reporte de pruebas. ....	78





## Introducción

La educación superior es muy importante tanto a nivel personal como en beneficio de la sociedad. Algunas de sus ventajas son [AFR 2002]:

- Es una de las mejores herramientas para el ingreso económico de una persona ya que por su trabajo calificado, técnico o profesional, recibirá un salario superior al que reciben los que no tienen una carrera profesional, disminuyendo así: La diferencia entre las clases sociales y el inconformismo que sienten las clases más bajas, los cuales incluso podrían generar la aparición de grupos rebeldes en el país.
- Disminuye la tasa de delincuencia.
- Libera a las personas de la ignorancia, el engaño y la opresión de clases dirigentes corruptas que no buscan satisfacer los intereses de la comunidad sino de los suyos propios.
- Permite a las personas definir con mayor claridad sus intereses y por tanto elegir la opción que más les convenga, esto debido a que tienen mayores conocimientos que les permiten discernir de forma más acertada. Luego, si cada uno tiene claro sus intereses individuales la política que elijan será aquella que favorezca a la mayoría por lo que se fortalece la democracia.
- Permite a las personas establecer una buena comunicación por lo que aumentan las probabilidades de ser escuchados.
- Permite a las personas educar mejor a sus hijos, sobretodo en cuanto a la entrega de valores morales y el buen ejemplo, los cuales aseguran la correcta inserción del niño en la sociedad y disminuye la corrupción y la delincuencia.
- Permite a las personas no sólo conseguir un mejor empleo sino generarlo con mayores probabilidades de éxito.

Sin embargo, en nuestro país la calidad de enseñanza de instituciones estatales es cuestionable debido al poco presupuesto que cuentan para invertir en investigación e innovación. Por otro lado, las buenas instituciones privadas no están al alcance de muchos alumnos que tienen capacidad y ganas de estudiar pero que sus recursos económicos se lo impiden [CNE 2005]. Por ello, algunas entidades gubernamentales y privadas -nacionales e internacionales- apoyan mediante préstamos a algunos alumnos que cumplen ciertos requisitos para que puedan culminar sus estudios de pregrado satisfactoriamente. Entre algunas de estas entidades benéficas tenemos: La Agencia Interamericana para la Cooperación y el Desarrollo (AICD) de la Organización de los Estados Americanos (OEA) [BCP 2004], Asociación Educación, Ministerio de Educación [MDE 2008] e INABEC.

Por otro lado, las entidades que apoyan a los alumnos derivan todo su presupuesto en la educación de los mismos. Así, o cuentan con sistemas informáticos de bajo presupuesto (que no garantizan su buen funcionamiento) o no cuentan con sistemas informáticos que automaticen sus procesos y trabajan todo de forma manual por lo que el acceso a un expediente, por ejemplo, puede demorar más de lo debido. En cambio, si se automatizaran los procesos podría reducirse tiempo y hacer proyecciones del presupuesto con el que se contará a futuro para así tener un número de vacantes más real con el que se pueda aceptar mayor o menor cantidad de alumnos como becarios.



## 1. Generalidades

En el presente capítulo se describirá el problema identificado, un marco conceptual para el mejor entendimiento del resto del documento, el plan de proyecto definido para el equipo de desarrollo, así como el estado del arte respecto a proyectos similares al propuesto y la sustentación de la solución planteada.

### 1.1. Definición del problema

Con el presente proyecto se desea resolver el problema que actualmente tiene una asociación educativa con la administración de la información de sus postulantes, ingresos y egresos económicos, becarios y ex-becarios, así como la gestión e intervención de las instituciones educativas con las que trabaja. Este problema se debe a que no se cuenta con un sistema informático que automatice sus procesos: convocatorias de postulación, seguimiento y control de becarios y ex-becarios, pagos realizados por los alumnos y generación de reportes que puedan ser enviados inmediatamente al directorio que es quien envía dinero, semestralmente, para cubrir las pensiones de los alumnos becarios peruanos en sus distintas instituciones de estudio.

El problema antes mencionado ocasiona:

- Procesos lentos y desorganizados.
- Múltiples archivos en formato Excel y, por ende, información redundante y hasta incoherente.
- Deficiente gestión de la información de los postulantes durante las convocatorias.

- Inexactitud en las fechas de inicio y fin de los períodos académicos (ciclos semestrales) de cada institución educativa, lo cual influye en el retraso del inicio del proceso de postulación.
- Poca o casi nula información sobre el desempeño académico de los becarios a lo largo del período académico (ciclo semestral).
- Poca o casi nula información sobre el desempeño laboral de los becarios y ex-becarios.
- Poca o casi nula información sobre los datos de contacto de becarios y ex-becarios, tales como dirección, teléfono y correo electrónico, esenciales para la adecuada comunicación con la organización. Esto es realmente crítico para la confirmación del depósito de reembolsos.
- Información inexacta sobre qué alumnos terminaron sus estudios o cuándo lo harán, incluyendo el monto total adeudado realmente. Estos datos sirven para realizar una proyección de vacantes disponibles para los próximos semestres y empezar a realizar los cobros correspondientes con el propósito de obtener mayores fondos.
- Envío tardío de la información al directorio cuando éste lo requiere.
- Deficiente realización de proyecciones de cuánto será el gasto total de pensiones de los becarios en sus instituciones y, por ende, el número de becas disponibles para nuevos alumnos.

## 1.2. Marco conceptual

La **asociación educativa** que se considera para una personalización en el presente proyecto de fin de carrera, es una entidad suiza (con sede en Perú) sin fines de lucro cuyo objetivo es promover el progreso socio-económico mediante la formación de jóvenes valores empresariales en países de Latinoamérica. Dicha asociación apoya, mediante un préstamo educativo sin intereses, a jóvenes universitarios que tengan condiciones de liderazgo, de gran integridad y provenientes de familias de modestos recursos, cuya educación superior no podría ser posible sin una ayuda económica externa. Para ello, recibe donaciones de su directorio suizo dos veces al año, con la finalidad de realizar el pago de pensiones respectivo para cada uno de sus alumnos.

Se llama **becario** de la asociación educativa a aquel alumno que recibe un préstamo, para ello debió ser partícipe del siguiente proceso (ver figura 1.1):

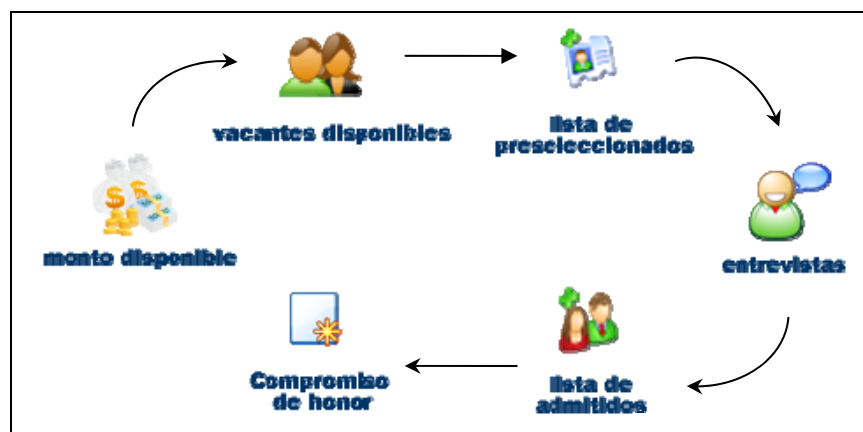


Figura 1.1. Diagrama general del proceso de admisión de nuevos becarios.

1. De acuerdo al dinero disponible (el enviado por el directorio suizo, lo que hay en la cuenta bancaria y lo que se proyecta cobrar a los ex-becarios) la secretaria ejecutiva de la asociación calcula el número de alumnos que podrán ser apoyados el próximo ciclo (semestre). Dichos alumnos deben estudiar en alguna de las **instituciones** con las que trabaja la asociación educativa.
2. La secretaria ejecutiva de la asociación envía a las instituciones, respectivamente:
  - El número de alumnos que éstas deben preseleccionar, de acuerdo a su situación económica, historial académico y otros requisitos que considere necesario cada institución.
  - Información que se debe otorgar a los alumnos acerca de la asociación. Esto es: descripción de la organización, requisitos de préstamo, carreras o especialidades a las que se hacen préstamos, cobertura (matrícula, seguro de renta educacional y máximo de créditos a prestar), porcentajes de préstamos, condiciones que siempre debe mantener un becario y forma de pago posterior al término de la carrera.
  - Los documentos en formato digital que los postulantes deben completar acerca de su información personal, económica y académica.
  - Las fechas, horas y la dirección del lugar en el que la secretaria ejecutiva junto con el directorio peruano, realizarán las entrevistas personales a los preseleccionados (alumnos elegidos por las instituciones).
3. Las instituciones envían a la organización la relación de alumnos preseleccionados junto con la información completada por estos.
4. Los alumnos preseleccionados son entrevistados.
5. La asociación selecciona a los alumnos que serán admitidos y el porcentaje de préstamo que se les otorgará a cada uno.
6. La secretaria ejecutiva de la organización envía la relación de alumnos admitidos a las instituciones para que estas les comuniquen a sus alumnos el porcentaje de préstamo otorgado y el Compromiso de Honor que deben firmar.
7. Las instituciones se encargan de comunicarle a los alumnos su admisión en la entidad, así como de hacerlos firmar el **Compromiso de Honor** (documento en el que el alumno se compromete a pagar a la asociación todo el dinero que esta invierta en sus estudios, al terminar su carrera).

Una vez firmado el Compromiso de Honor, oficialmente el alumno toma el nombre de **becario** para la asociación educativa. A este becario se le pagará:

- Matrícula,
- Seguro educacional y
- Derechos académicos correspondientes al porcentaje de préstamo aprobado.

Esto es por ciclo y hasta el término de su carrera, siempre que mantenga las condiciones de un becario.

Por otro lado, se llamará **ex-becario** de la asociación educativa a aquel alumno que culmine sus estudios o que, por algún caso especificado en el reglamento de la organización (constantes notas finales desaprobatorias o a solicitud del alumno por una mejor situación económica), pierda su préstamo antes de finalizar su carrera. Si

un alumno becario pasa al estado ex-becario no tiene más de seis meses para totalizar su deuda o préstamo otorgado y empezar a efectuar el pago del mismo.

### 1.3. Plan de proyecto

La Gestión de Proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del mismo.

Para la realización del presente proyecto se considera la conformación de un pequeño equipo de desarrollo (dos personas en total) y la continua colaboración del usuario final durante el ciclo de vida del proyecto, lo que conlleva a que existan diversos cambios durante el desarrollo. Por ello, se ha planteado la realización de las tareas y procesos de acuerdo a la metodología de gestión de proyectos **Scrum**, ya que se trata de una metodología ágil especialmente aplicada a pequeños equipos de desarrollo, se orienta a una entrega rápida de resultados y es altamente flexible [ANS 2008].

Las metodologías ágiles se centran en aspectos como la flexibilidad en la introducción de cambios y nuevos requisitos durante el proyecto, el factor humano, el producto final, la colaboración con el cliente y el desarrollo incremental como formas de asegurar los buenos resultados en proyectos con requisitos muy cambiantes o cuando se exige reducir los tiempos de desarrollo manteniendo una alta calidad.

Scrum se basa en un enfoque iterativo, donde cada iteración se denomina *Sprint*. A diferencia de las iteraciones en cascada, al final de cada Sprint se obtiene un producto que se va incrementando en sucesivos Sprints, priorizándose aquellos aspectos que aportan mayor funcionalidad. El principio básico que maneja es que es muy difícil contar con un catálogo completo de funcionalidades desde el inicio del proyecto, ya que los requisitos van surgiendo conforme el jefe de proyecto (*Product Owner*) y los usuarios del mismo van haciendo sucesivas aportaciones.

Esta metodología se compone básicamente de cinco etapas:

- **Revisión de planes de versión:** Se lleva a cabo por el equipo a fin de evaluar las diferentes factibilidades de los requerimientos y estimaciones.
- **Distribución, revisión y ajustes de estándares de producto:** Los desarrolladores realizan los ajustes de los estándares y dejan todo listo para comenzar con la etapa fundamental de Scrum: los *Sprints*.
- **Iteraciones (Sprints):** El desarrollo propiamente dicho se efectiviza durante el sprint. Engloba diferentes tareas como: codificación, test, documentación y revisiones. No existe una secuencia definida de cómo encarar las subtarefas de cada sprint.
- **Revisión de Sprint:** En este punto se pueden hacer modificaciones a la lista de requerimientos según los resultados de sprints, en caso de detectar algún requerimiento no haya sido tenido en cuenta desde un primer momento (de ocurrir esto debe tratarse de un requerimiento no funcional). Por último se planifica el próximo Sprint.
- **Cierre:** Es la última oportunidad para efectuar depuración antes de construir el entregable y dar el proceso por finalizado.

Adicionalmente, se debe complementar con alguna metodología ágil de desarrollo de software, como *Extreme Programming*, para su mejor aplicación [CHR 2007].

Considerando todo lo anterior y aplicando adicionalmente *Extreme Programming*, se han definido las siguientes grandes etapas para la realización del presente proyecto:

- Exploración.
- Planificación de la entrega.
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del proyecto.

Con ello, se ha determinado que el proyecto tendrá una duración de trescientos treinta y cuatro días, considerando ocho horas diarias de trabajo y el esfuerzo inicial de un solo tesista y el esfuerzo final de dos tesistas. Así, se calcula que el proyecto tendrá una vida de 4080 horas.

$$(158 \text{ días}) * (8 \text{ horas}) * (1 \text{ tesista}) + (176 \text{ días}) * (8 \text{ horas}) * (2 \text{ tesistas}) = 4080 \text{ horas}$$

### 1.3.1. Diagrama Work Breakdown Structure

En la figura 1.2 se muestra el árbol jerárquico de las tareas que se llevarán a cabo durante el ciclo de vida del proyecto.

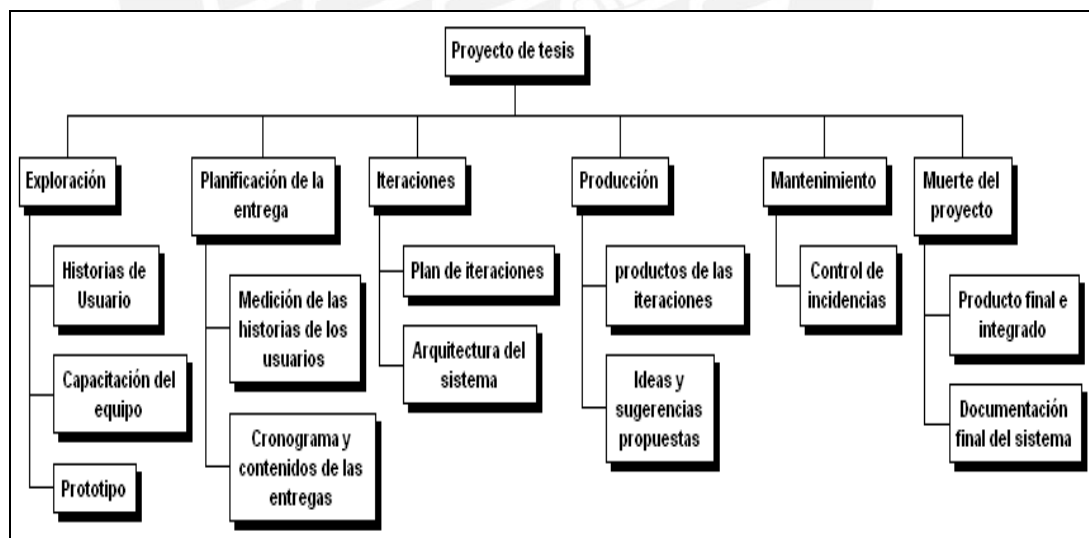


Figura 1.2. WBS del proyecto de Tesis.



### 1.3.2. Planificación inicial

En la figura 1.3 se muestra la lista de tareas planificadas, indicando las fechas de comienzo y fin de cada una de ellas. La columna “Predecesoras” indica la(s) tarea(s) que se debe(n) cumplir primero para que determinada tarea pueda llevarse a cabo.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1 <input type="checkbox"/> Proyecto de tesis	334 días	mar 15/04/08	vie 24/07/09		MC,JG
2 <input type="checkbox"/> Exploración	144 días	mar 15/04/08	sáb 01/11/08		MC
3 <input type="checkbox"/> Historias de los usuarios	8 días	mar 15/04/08	jue 24/04/08		MC
4 Se preparan algunas preguntas	1 día	mar 15/04/08	mar 15/04/08		MC
5 Se hace levantamiento de información	5 días	mié 16/04/08	mar 22/04/08	4	MC
6 Se ordena la información recogida	2 días	mié 23/04/08	jue 24/04/08	5	MC
7 <input type="checkbox"/> Capacitación del equipo	90 días	lun 28/04/08	sáb 30/08/08		MC,JG
8 Investigación de las herramientas a usar	25 días	lun 28/04/08	vie 30/05/08	6	MC
9 Familiarización con las herramientas, tecnología y prácticas a usar	68 días	jue 22/05/08	sáb 30/08/08		MC,JG
10 <input type="checkbox"/> Prototipo	72 días	jue 24/07/08	sáb 01/11/08		MC,JG
11 Se prueba la tecnología y las posibilidades de la arquitectura	15 días	lun 01/09/08	vie 19/09/08	9	MC,JG
12 Se muestra el prototipo al cliente hasta que quede conforme	68 días	jue 24/07/08	sáb 01/11/08		MC,JG[50%]
13 <input type="checkbox"/> Planificación de la entrega	6 días	mar 01/07/08	mar 08/07/08		MC
14 <input type="checkbox"/> Medición de las historias de usuarios	4 días	mar 01/07/08	vie 04/07/08		MC
15 Se priorizan las historias	1 día	mar 01/07/08	mar 01/07/08	3	MC
16 Se estima el esfuerzo de implementación de las historias	3 días	mié 02/07/08	vie 04/07/08		MC
17 <input type="checkbox"/> Cronograma y contenidos de las entregas	4 días	jue 03/07/08	mar 08/07/08		MC
18 Se determina la velocidad de las historias y del proyecto	2 días	jue 03/07/08	vie 04/07/08		MC
19 Se hace un sumado de puntos de las historias	2 días	vie 04/07/08	lun 07/07/08		MC
20 Se obtiene el número de iteraciones necesarias	2 días	lun 07/07/08	mar 08/07/08		MC
21 <input type="checkbox"/> Iteraciones	62 días	vie 04/07/08	lun 29/09/08		MC
22 <input type="checkbox"/> Plan de iteraciones	11 días	vie 04/07/08	vie 18/07/08		MC
23 Se tienen en cuenta las iteraciones no abordadas	5 días	vie 04/07/08	jue 10/07/08		MC
24 Se tiene en cuenta la velocidad del proyecto	5 días	lun 07/07/08	sáb 12/07/08		MC
25 Se tienen en cuenta las pruebas de aceptación no superadas en las iteraciones anteriores	5 días	jue 10/07/08	mié 16/07/08		MC
26 Se tienen en cuenta las tareas no terminadas en las iteraciones anteriores	5 días	lun 14/07/08	vie 18/07/08		MC
27 <input type="checkbox"/> Arquitectura del sistema	27 días	vie 22/08/08	lun 29/09/08		MC,JG
28 Se eligen las historias que fuercen la creación de la arquitectura	2 días	vie 22/08/08	lun 25/08/08	16	MC
29 Se decide el diseño de la arquitectura del sistema	5 días	mar 26/08/08	lun 01/09/08	28	JG,MC
30 Se crea un proyecto inicial con la arquitectura diseñada	20 días	mar 02/09/08	lun 29/09/08	29	JG,MC[50%]
31 <input type="checkbox"/> Producción	192 días	mar 30/09/08	mié 24/06/09		MC,JG
32 <input type="checkbox"/> Productos de las iteraciones	177 días	mar 30/09/08	mié 03/06/09		MC,JG
33 Se codifica lo acordado en las iteraciones	174 días	mar 30/09/08	mié 03/06/09	21	MC,JG
34 <input type="checkbox"/> Ideas y sugerencias propuestas	168 días	lun 03/11/08	mié 24/06/09		MC,JG
35 Se hacen pruebas adicionales	165 días	lun 03/11/08	mié 24/06/09		MC,JG
36 Se revisa el rendimiento del sistema en ambiente previo al del cliente	165 días	lun 03/11/08	mié 24/06/09		MC,JG
37 Se decide la inclusión de nuevas características a la versión actual	165 días	lun 03/11/08	mié 24/06/09		MC,JG
38 Se actualiza el plan de iteraciones	165 días	lun 03/11/08	mié 24/06/09		MC,JG
39 <input type="checkbox"/> Mantenimiento	168 días	jue 20/11/08	lun 13/07/09		MC,JG
40 <input type="checkbox"/> Control de incidencias	168 días	jue 20/11/08	lun 13/07/09		MC,JG
41 Se hace mantenimiento al sistema que está en producción	165 días	jue 20/11/08	lun 13/07/09		MC,JG
42 Se siguen desarrollando las nuevas iteraciones	165 días	jue 20/11/08	lun 13/07/09		MC,JG
43 Se realizan tareas de soporte para el cliente	165 días	jue 20/11/08	lun 13/07/09		MC,JG
44 Se actualiza la velocidad del proyecto	165 días	jue 20/11/08	lun 13/07/09		MC,JG
45 <input type="checkbox"/> Muerte del proyecto	27 días	jue 18/06/09	vie 24/07/09		MC,JG
46 <input type="checkbox"/> Producto final e integrado	3 días	jue 16/07/09	lun 20/07/09		MC,JG
47 Se integran los productos obtenidos en las iteraciones	3 días	jue 16/07/09	lun 20/07/09		MC,JG
48 <input type="checkbox"/> Documentación final del sistema	27 días	jue 18/06/09	vie 24/07/09		MC,JG
49 Se satisfacen las necesidades de rendimiento del sistema que tenga aún el cliente	20 días	jue 18/06/09	mié 15/07/09		MC,JG
50 Se satisfacen las necesidades de confiabilidad del sistema que tenga aún el cliente	15 días	lun 06/07/09	vie 24/07/09		MC,JG

Figura 1.3. Planificación inicial del proyecto de Tesis.



En la figura 1.4 se muestra el diagrama de Gantt de las tareas definidas en la planificación inicial.

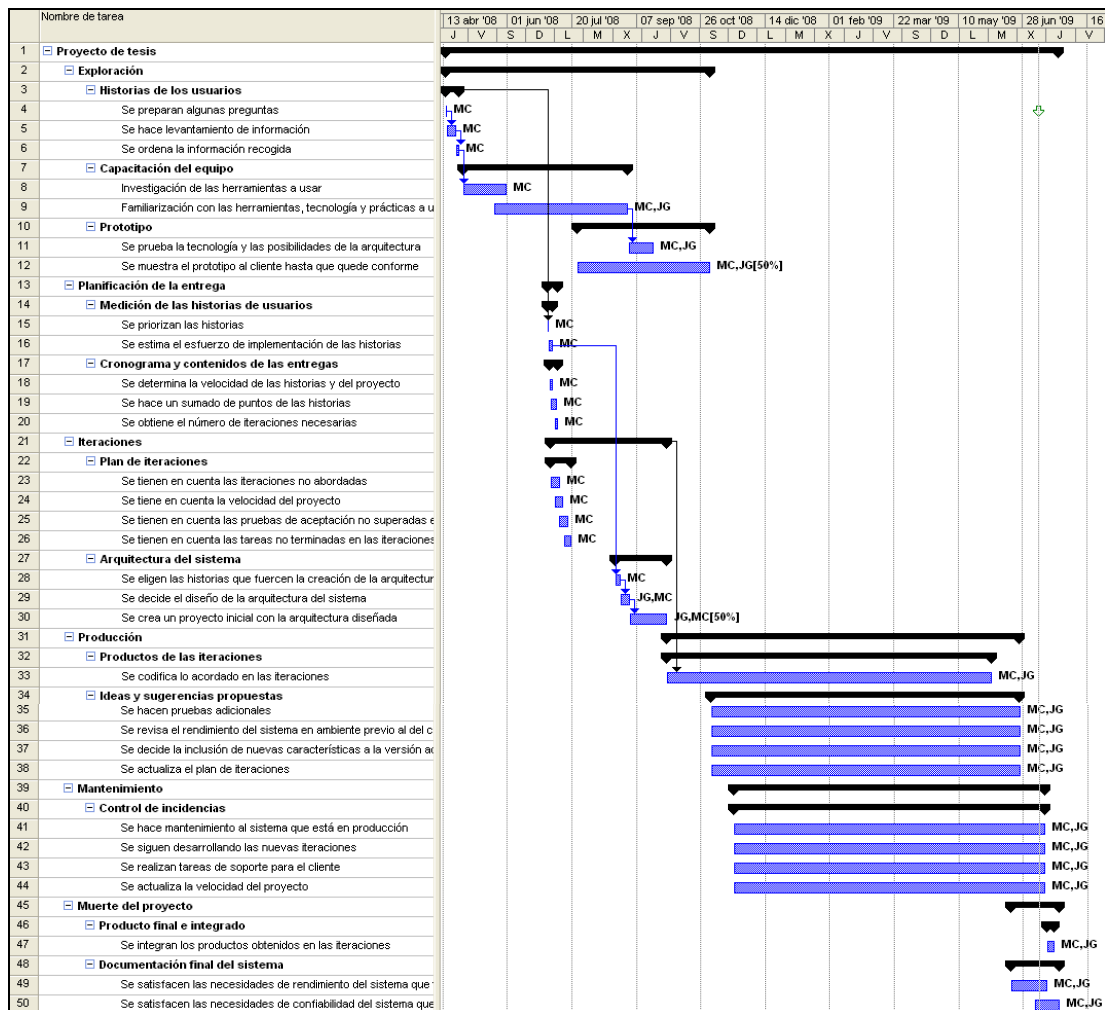


Figura 1.4. Diagrama de Gantt del proyecto de Tesis.

En el Anexo A: Diagramas de Planificación del Proyecto, se puede encontrar mayor detalle.

### 1.4. Estado del arte

Actualmente, existen algunos sistemas de préstamos universitarios, entre los que se puede mencionar:

- **SABE** (Sistema de administración de becas escolares): Sistema de información web de becas del Instituto Politécnico Nacional, en México. Este es un sistema que contempla diferentes tipos de becas y para diferentes instituciones de estudios. Además, permite a los alumnos becados, alumnos aspirantes y padres de familia hacer un seguimiento de su condición a través de una contraseña [IPN 2008]. En la figura 1.5 se muestra su página principal.



Figura 1.5. Sistema de Administración de Becas Escolares - SABE.

- SISBE** (Sistema de becas): Sistema de información web de becas de la Universidad Nacional de Río Cuarto (UNRC), en Argentina. Este es un sistema que forma parte de una gran red de sistemas informáticos integrales de la UNRC, por lo que fue diseñado bajo una metodología de desarrollo de sistemas integrados y es soportado automáticamente por la herramienta CASE Oracle Designer/2000. Además, es sencillo, organizado, de fácil acceso, seguro con el tráfico en la red y está relacionado con el sistema integral de alumnos (SIAL), del cual obtiene datos de los alumnos UNRC. Respecto a su funcionalidad, administra datos de los que aspiran una beca y genera informes/consultas del estado de los alumnos becados, efectúa liquidación de becas y genera diferentes tipos de estadísticas. [NRC 2004]. En la figura 1.6 se muestra su página principal.

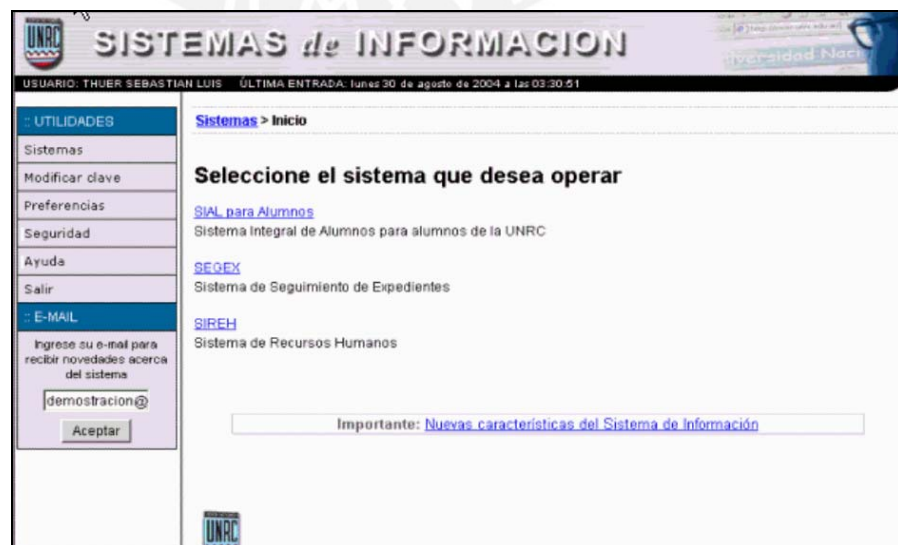


Figura 1.6. Sistema de Becas - SISBE.

- **Scholarship System Scoping Study:** Sistema de información web de becas de la Universidad de Washington. Este es un proyecto que busca: facilitar el sistema de los becarios, integrarse a la base de datos de los estudiantes, contemplar un módulo de cuentas de los alumnos, mejorar la eficiencia operacional, producir mayor satisfacción en los clientes, y procesar rápido las solicitudes de los alumnos [DKL 2004].

En la tabla 1.1 se muestra la comparación entre los sistemas mencionados anteriormente y el sistema propuesto en el presente proyecto.

Característica	SABE	SISBE	SSSS	PROPUESTO
Sistema de información web.	SI	SI	SI	SI
Maneja diferentes tipos de becas.	SI	NO	NO	SI
Manejo de diferentes instituciones.	SI	NO	NO	SI
Permite a los alumnos ingresar al sistema.	SI	NO	NO	SI
Se integra a otros sistemas.	-	SI	SI	NO
Es soportado por la herramienta Case Oracle Designer/2000.	NO	SI	NO	NO
Es sencillo de manejar, organizado, de fácil acceso y seguro.	NO	SI	NO	SI
Tiene módulo: Cuentas de los becarios y ex-becarios.	NO	NO	SI	SI
Procesa rápido las solicitudes de los usuarios.	NO	NO	SI	SI
Facilidad de mantenimiento debido a una programación modular.	NO	NO	NO	SI
Desarrollado con software libre.	NO	NO	NO	SI

**Tabla 1.1.** Comparativo de funcionalidades de algunas soluciones existentes.

## 1.5. Descripción y sustentación de la solución

El sistema de información propuesto contemplará lo siguiente:

- Será desarrollado bajo plataforma web, haciendo uso de software libre.
- Será modular y de tipo escalable.
- Permitirá el manejo de información de alumnos con diferentes tipos de becas y de diferentes instituciones.
- Permitirá llevar el control de pagos de los becarios y ex-becarios.
- Permitirá realizar seguimiento del desempeño académico y laboral de los becarios y ex-becarios.
- Permitirá el manejo de usuarios, según distintos privilegios establecidos.
- Permitirá a las diferentes instituciones ingresar información de sus alumnos, facultades, carreras, ciclos y representantes.
- Permitirá generar reportes como ayuda a la toma de decisiones.

- Proporcionará información consistente y confiable.
- Poseerá una interfaz gráfica de usuario intuitiva y de rápida respuesta (debido a que un buen diseño minimizará el tiempo de los procesos).
- Permitirá el envío de correos electrónicos con información únicamente enfocada a los procesos de negocio, tales como: postulación, totalización de deudas, entre otros.

Cabe mencionar que estas características son similares a las ofrecidas por los sistemas extranjeros investigados (no se encontraron resultados para la búsqueda de sistemas de información que manejen becas de pregrado en el Perú) y también se basan en las necesidades de los usuarios que interactúan en los procesos, dichas necesidades fueron identificadas en el levantamiento de información.

Así, el producto final del desarrollo del presente proyecto traerá consigo los siguientes beneficios:

- **Para la asociación educativa:** Podrán publicar la información pertinente para las instituciones (préstamos disponibles, carreras recibidas, máximo de créditos por cubrir, porcentajes de préstamo), encontrar el expediente de un alumno fácilmente, visualizar el promedio de notas de los becarios por ciclo, visualizar un historial de datos de contacto y de desempeño laboral de los becarios y ex-becarios, realizar cálculos de proyecciones de presupuesto anual y de flujo de caja en base a los ingresos y egresos económicos anteriores, contar con la información actualizada de las cuentas bancarias, visualizar y exportar los reportes que crean convenientes.  
Además, tendrá una imagen tecnológica más sólida por contar con un sistema que representa actualidad, renovación y que ve en la informática una herramienta útil que le ayuda a mejorar sus procesos.
- **Para el directorio:** Podrán visualizar y exportar los reportes (expedientes de los postulantes, actuales alumnos que cuentan con el préstamo, notas de los mismos, gasto mensual y por ciclo para cubrir las pensiones de sus alumnos y para sus distintas instituciones, estado de pagos de los ex-becarios, historial de desempeño laboral de becarios y ex-becarios, entre otras cosas) que necesiten en cualquier momento.
- **Para los alumnos:** Podrán completar sus datos directamente en el sistema cuando se encuentren en el proceso de postulación así como conocer su estado mientras dura ese proceso, podrán visualizar el monto adeudado hasta el momento de la consulta, podrán actualizar sus datos personales (la dirección y teléfonos, por ejemplo, son importantes para la asociación pues muchas veces solo a través de ellos pueden ubicar a los alumnos para que realicen el pago del préstamo), podrán recibir notificaciones en sus correos cada vez que sea necesario (cuando deben registrar sus datos, cuando se acerque una fecha en la que deben cancelar, entre otros), y armar su cronograma de pagos (el cual debe ser aprobado por la secretaria ejecutiva).
- **Para las instituciones:** Podrán actualizar en el sistema las facultades y carreras con las que cuenta, así como el valor del crédito o pensión por cada una de ellas; registrar las fechas de inicio y fin de cada ciclo (importante para todo lo relacionado a pagos de los alumnos), registrar directamente el monto total por ciclo que la organización debe cancelar por cada uno de sus becarios, ingresar las notas de los alumnos a mitad y fin de ciclo, reafirmar el estado real del alumno al finalizar un ciclo (becario o ex-becario) y encontrar rápidamente el historial de sus alumnos.



## 2. Análisis

En el presente capítulo se detallará la aplicación de la metodología utilizada, los requerimientos identificados, así como el análisis y definición del sistema propuesto.

### 2.1. Metodología aplicada para el desarrollo de la solución

Dadas las características del presente proyecto de fin de carrera:

- Equipo de desarrollo pequeño (dos tesistas).
- Tiempo limitado para el período de desarrollo.
- Múltiples usuarios involucrados (postulantes, becarios, ex-becarios, representantes de instituciones, secretaria ejecutiva de la asociación y directorio) por lo que se tienen requerimientos que cambian en el tiempo.
- Se requieren prácticas adaptativas en lugar de predictivas.
- Se tiene como miembros del equipo a los usuarios finales.
- El código será probado (pruebas unitarias y de integración) para asegurar calidad y confianza de todos los usuarios.

Se ha optado por seguir una metodología ágil para el desarrollo de la solución en lugar de las protocolares [WIG 2007], esta es **Extreme Programming**, conocida también como **XP**.

#### 2.1.1. Aplicación de Extreme programming

Se trata de una de las metodologías más exitosas entre las ágiles, tiene como objetivo la satisfacción del cliente, asegura obtener un software de calidad, diseñada para equipos de desarrollo de software pequeños, de bajo riesgo, flexible (para requerimientos que cambien en el tiempo), que promueve prácticas

adaptativas en lugar de predictivas, centrada en la gente o el equipo más que en los procesos y herramientas, que se enfoca más en el resultado que en la documentación exhaustiva, que tiene como miembro de equipo al usuario final (cerca y constantemente involucrado con el proyecto, de intensa comunicación) y que ofrece retroalimentación continua. [JWI 2008]

A continuación se muestra algunos aspectos de dicha metodología que motivaron su elección para el desarrollo de la solución propuesta.

*Fundamentos:*

- Mejorar la comunicación en todo nivel.
- Simplicidad en el desarrollo.
- Búsqueda continua de la retroalimentación.
- Actitud valiente de todo miembro del equipo para la realización de cambios.

*Prácticas o forma de trabajo:*

- Pruebas unitarias.
- Refabricación.
- Programación en pares.
- Iteraciones cortas.
- Documentación principalmente en el código desarrollado.

*Derechos del cliente:*

- Decidir qué se implementa.
- Conocer el estado del proyecto en cualquier momento.
- Añadir, modificar o eliminar requerimientos en cualquier momento.

*Derechos del desarrollador:*

- Decidir cómo se implementarán los procesos.
- Pedir al cliente aclaraciones de sus requerimientos en cualquier momento.
- Modificar los requerimientos en base a nuevos descubrimientos.
- Trabajar no más de 40 horas semanales.

*Elementos que aseguran el éxito del producto:*

- Comunicación permanente cliente – desarrollador.
- Los módulos codificados deben ser simples y bastante documentados.
- La retroalimentación debe ser constante.

A continuación se muestra las etapas de Extreme Programming [CSG 2004, DOW 2006] que serán consideradas para el desarrollo del sistema propuesto, indicando en cada una de ellas las tareas a realizarse.

- **Exploración:**
  - Obtención de historias de los usuarios: Se hace un levantamiento de información.
  - Capacitación: El equipo se familiariza con las herramientas, tecnología y prácticas a usar en el proyecto.
  - Realización de prototipo: Se prueba la tecnología y se exploran las posibilidades de la arquitectura.
- **Planificación de la entrega:**
  - Medición de las historias de usuario: Se da una prioridad de historias de usuario y una estimación del esfuerzo necesario en la implementación de cada una (se usan puntajes que van de 1 a 3).
  - Se realiza un cronograma y contenidos de las entregas: Realizado junto con el cliente. Cada entrega debería hacerse en no más de 3



meses. Para ello se determina la velocidad con que podrán realizarse las historias, suma de puntos de las historias y finalmente el número de iteraciones necesarias.

- **Iteraciones:**
  - Elaboración de plan de iteraciones: Iteraciones de no más de 3 semanas, se tienen en cuenta historias no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.
  - Elaboración de la arquitectura del sistema: Se escoge a las historias de usuario que fueren la creación de la arquitectura.
- **Producción:**
  - Se proponen ideas y sugerencias, las cuales serán implementadas posteriormente. Para ello se hacen pruebas adicionales, se revisa el rendimiento antes de trasladar el sistema al entorno del cliente, se decide sobre la inclusión o no de nuevas características a la versión actual y se actualiza el tiempo que tomarán las iteraciones.
- **Mantenimiento:**
  - Se hace un control de Incidencias: Mantenimiento del sistema que está en producción mientras se sigue desarrollando las nuevas iteraciones. Además, se realizan tareas de soporte para el cliente y se actualiza la velocidad del proyecto.
- **Muerte del proyecto:**
  - Documentación final del sistema: Se pregunta al cliente hasta que se hayan satisfecho sus necesidades de rendimiento y confiabilidad del sistema.

## 2.2. Identificación de requerimientos

Para la identificación de los requerimientos de la solución propuesta, se considera el levantamiento de información realizado para la asociación educativa suiza llamada 'Asociación Educación', en su sede de Perú. La información recabada es producto de las constantes reuniones entre los miembros del equipo de desarrollo del presente proyecto y las diferentes personas que interactúan con la asociación.

Cabe señalar que dicha entidad cuenta con un directorio suizo, ente que lo representa y que envía donaciones para los préstamos universitarios. Además, trabaja con tres instituciones educativas reconocidas: Pontificia Universidad Católica del Perú, Universidad de Lima y TECSUP.

El listado de requerimientos funcionales y no funcionales se obtuvo de la siguiente manera:

- Algunos como resultado de la investigación de sistemas parecidos al que se desarrollará con el presente proyecto [IPN 2008].
- Los demás como resultado de las entrevistas con los diferentes tipos de usuarios que contará el sistema: administradora de la asociación, representantes de las instituciones, ex-becarios y becarios. En el Anexo J: Historias de Usuario, se pueden encontrar algunos ejemplos de historias de usuario realizadas en el levantamiento de información.

En las tablas 2.1 y 2.2 se muestra el listado de requerimientos funcionales y no funcionales, respectivamente, con las explicaciones pertinentes de cómo resuelven el problema.

Recordar que:

- Se medirá en una escala del 1 al 5, donde 1 es el mínimo valor de importancia y 5 el máximo.
- E = Exigible y D = Deseable

N°	Requerimientos Funcionales			
	Generales:	Dif.	Pri.	Ex.
1	Permitirá manejar diferentes tipos de becas. Esto permitirá que los postulantes puedan acceder a diferentes porcentajes de beca de acuerdo a su situación económica.	3	5	E
2	Permitirá manejar información de alumnos de diferentes instituciones. Esto permitirá a la asociación trabajar con diferentes instituciones y, por ende, manejar diferentes lógicas de pago de pensiones según la institución.	3	5	E
3	Permitirá realizar el mantenimiento de los datos de las instituciones, tales como: nombres, apellidos y correos electrónicos de cada uno de los representantes. Esto permitirá a la asociación tener información actualizada sobre los representantes de las instituciones y sus datos de contacto, útiles para la comunicación.	2	5	E
4	Permitirá realizar el mantenimiento de los datos de las facultades de cada institución, en caso existan, tales como: nombre. Esto permitirá que los representantes de las instituciones puedan agregar, modificar o eliminar facultades, directamente al sistema.	3	4	D
5	Permitirá realizar el mantenimiento de los datos de las carreras de cada institución, tales como: código asignado por la institución, nombre, número de créditos totales de la carrera, número de ciclos totales de la carrera. Esto permitirá que los representantes de las instituciones puedan agregar, modificar o eliminar carreras, según la facultad, directamente al sistema.	3	4	D
6	Permitirá el manejo de diferentes estados para los postulantes. Esto permitirá conocer en qué etapa del proceso de postulación se encuentra cada alumno, restringiendo la posibilidad de admisión en caso no se haya cumplido con todos los estados previos del proceso completo.	4	5	E
7	Permitirá a los becarios y ex-becarios actualizar sus datos personales, tales como: número de DNI, estado civil, correo electrónico, teléfono, dirección, número de ciclos y número de créditos que faltan para culminar sus estudios. Esto permitirá a la asociación tener actualizada la información de datos personales de sus becarios.	3	4	E

<b>Generales:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
8	<p>Permitirá a los becarios y ex-becarios actualizar sus datos laborales, tales como: nombre y teléfono de la empresa en la que laboran, así como el tipo de trabajo que desempeñan.</p> <p>Esto permitirá a la asociación tener actualizada la información de datos laborales de sus becarios, para un posterior seguimiento de su desempeño profesional.</p>	3	4	D
9	<p>Permitirá a la asociación realizar el mantenimiento de los tipos de trabajo que se utilizarán en el sistema.</p> <p>Esto permitirá a la asociación categorizar los tipos de trabajo en los que se desempeñan sus alumnos, según su criterio.</p>	3	3	D
10	<p>Permitirá a la asociación realizar el mantenimiento del contenido de los correos predeterminados que se utilizarán en el sistema.</p> <p>Esto permitirá a la asociación personalizar los correos que se enviarán a los diferentes actores del sistema.</p>	3	3	D
11	<p>Permitirá a la asociación realizar el mantenimiento de los conceptos de ingresos y egresos que se utilizarán en el sistema.</p> <p>Esto permitirá a la asociación personalizar los conceptos que se usarán para movimientos de ingresos y egresos.</p>	3	3	D
12	<p>Permitirá enviar e-mails a los actores interesados en cada etapa de un proceso.</p> <p>Esto permitirá la fluida comunicación entre los usuarios, más aún cuando les interesa el proceso.</p>	3	5	E
13	<p>Permitirá exportar documentos en formato PDF.</p> <p>Esto permitirá a los usuarios archivar los documentos que les interesen e imprimirlos cuando los necesiten.</p>	4	3	D
<b>Postulación:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
14	<p>Permitirá a la asociación publicar los datos necesarios del proceso de postulación: número de vacantes por institución, fecha límite en la que las instituciones revisarán los datos de los postulantes y las fechas y horas de las entrevistas personales para cada institución.</p> <p>Esto permitirá a las instituciones tener en cuenta datos importantes del proceso de postulación.</p>	3	5	E
15	<p>Permitirá a las instituciones ingresar los datos básicos de sus postulantes, tales como: nombres, apellidos, correo electrónico y login de usuario.</p> <p>Esto permitirá a las instituciones invitar postulantes (a través de e-mails) a participar del proceso selectivo para acceder a un préstamo educativo.</p>	4	5	E
16	<p>Generará automáticamente una contraseña aleatoria para cada postulante al momento de enviar e-mails con las invitaciones de postulación.</p> <p>Esto permitirá que, por seguridad, los postulantes accedan personalmente a sus cuentas de usuario, sin intervención de las instituciones.</p>	2	5	E

	<b>Postulación:</b>	<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
17	<p>Permitirá a los postulantes completar sus datos personales, académicos y socioeconómicos.</p> <p>Esto permitirá a los postulantes completar fácilmente sus datos y que luego las instituciones puedan verificarlas.</p>	4	5	E
18	<p>Permitirá a las instituciones completar y agregar información correspondiente a sus postulantes, tal como un informe de la asistente social y/o decano.</p> <p>Esto permitirá a las instituciones verificar la información completada por los postulantes y adicionar el informe de la situación económica de cada uno de ellos, requerido por la asociación.</p>	4	4	E
19	<p>Permitirá a las instituciones preseleccionar a los alumnos invitados y que hayan completado sus datos, para formar parte del proceso de postulación.</p> <p>Esto permitirá a las instituciones filtrar a los estudiantes que, según su criterio, cumplen los requisitos exigidos para el proceso de postulación de la asociación.</p>	3	5	E
20	<p>Permitirá a la asociación elegir a los postulantes preseleccionados por las instituciones que, según su criterio, deberían realizar una entrevista personal.</p> <p>Esto permitirá a la asociación seleccionar solo aquellos postulantes que pasaron el filtro de las instituciones, para que se les asigne una hora de entrevista.</p>	3	5	E
21	<p>Permitirá a las instituciones asignar a cada postulante seleccionado para entrevista, una hora adecuada para la realización de la misma, de acuerdo al rango de horas disponibles para entrevistar dispuestas por la asociación.</p> <p>Esto permitirá a las instituciones administrar las horas de la entrevista de acuerdo a la disponibilidad de los postulantes.</p>	3	4	E
22	<p>Permitirá a la secretaria ejecutiva de la asociación completar y agregar información de los postulantes, tal como un informe propio de la situación económica.</p> <p>Esto permitirá a la secretaria ejecutiva dar un formato adecuado a la información de los postulantes que será visualizada por el directorio suizo, filtrando lo que debe no debe colocarse.</p>	3	5	E
23	<p>Permitirá a la secretaria ejecutiva de la asociación elegir a los postulantes que podrían ser admitidos, previamente a la decisión del directorio suizo.</p> <p>Esto permitirá a la secretaria ejecutiva seleccionar solo a aquellos postulantes que pasaron satisfactoriamente las etapas previas del proceso de postulación.</p>	3	5	E
24	<p>Permitirá al directorio suizo seleccionar a los postulantes que serán admitidos.</p> <p>Esto permitirá al directorio suizo seleccionar solamente a los postulantes elegidos por la secretaria ejecutiva.</p>	3	5	E

<b>Postulación:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
25	<p>Permitirá al directorio suizo la posibilidad de asignar, a cada postulante admitido, un porcentaje de préstamo igual o menor al solicitado por el alumno.</p> <p>Esto permitirá al directorio suizo aprobar un porcentaje de préstamo conveniente, según su criterio, para cada alumno.</p>	3	5	D
26	<p>Permitirá a los postulantes consultar la información correspondiente a las entrevistas.</p> <p>Esto permitirá a los postulantes tener a su disposición la información de la entrevista personal que le haya sido asignada, en caso la tuviera y haya ocurrido algún percance con el mensaje e-mail que se le envía con los datos de la misma.</p>	2	3	E
<b>Operaciones de ciclo:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
27	<p>Permitirá realizar el mantenimiento de la información de los ciclos por cada institución, tales como: nombre, fechas de inicio y fin, y factor de ajuste, si se maneja.</p> <p>Esto permitirá que los representantes de las instituciones puedan configurar los datos de cada ciclo, esenciales para la mayoría de procesos tales como pagos de alumnos.</p>	3	5	E
28	<p>Permitirá a las instituciones realizar el mantenimiento de las escalas de pago y el valor del crédito por cada carrera o del curso en el ciclo, según sea.</p> <p>Esto permitirá que las instituciones puedan configurar los datos de escalas de pago, esenciales para el registro de montos totales a pagar por cada becario.</p>	3	5	E
29	<p>Permitirá a las instituciones ingresar, cada inicio de ciclo, los montos subtotales y totales que la asociación debe pagar por cada uno de sus becarios, considerando: la cantidad de cursos matriculados, la cantidad de créditos matriculados, el pago por derechos académicos, el pago por el seguro estudiantil y el factor de ajuste vigente, según sea el caso.</p> <p>Esto permitirá a la asociación tener actualizada la información económica de cada uno de sus becarios.</p>	4	5	E
30	<p>Asignará automáticamente como monto total a pagar en el ciclo, para cada institución, la suma total de los montos a pagar por cada uno de los becarios.</p> <p>Esto permitirá a la asociación tener información exacta sobre los pagos a realizar a cada institución en el ciclo, sin errores de cálculo puesto que ya no sería un cálculo manual.</p>	3	5	E
31	<p>Permitirá a las instituciones ingresar a mediados de cada ciclo y para cada becario: cantidad de cursos llevados, cantidad de cursos reprobados y promedio de notas de dichos cursos.</p> <p>Esto permitirá a la asociación tener conocimiento del avance académico de cada uno de sus becarios por ciclo.</p>	3	3	D
32	<p>Permitirá a las instituciones ingresar, cada fin de ciclo: el total de créditos matriculados, el total de créditos reprobados y la nota promedio final obtenida en el ciclo, por cada uno de sus becarios.</p> <p>Esto permitirá a la asociación tener actualizada la información académica de cada uno de sus becarios.</p>	3	4	D



<b>Operaciones de Ciclo:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
33	<p>Permitirá a las instituciones seleccionar, cada inicio de ciclo, a los becarios del ciclo anterior como becarios o ex-becarios del ciclo actual, según sea si a los alumnos les falta completar estudios, serán expulsados o serán condonadas sus deudas por motivos de fuerza mayor.</p> <p>Esto permitirá a las instituciones brindar información actualizada a la asociación sobre la culminación de estudios de sus alumnos.</p>	4	5	E
34	<p>Convertirá automáticamente a un ex-becario del ciclo anterior en ex-becario del ciclo vigente si se cumple con tener deuda pendiente que no se encuentre condonada.</p> <p>Esto permitirá a la asociación tener actualizada la lista de ex-becarios que seguirán realizando pagos.</p>	4	5	E
<b>Ingresos y egresos - Reembolsos</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
35	<p>Permite a la asociación registrar un egreso por pago de pensiones u otro concepto. Así mismo, le permitirá también realizar una transacción entre las cuentas bancarias con las que se cuenta en el sistema.</p> <p>Esto le permitirá a la asociación tener actualizada la información de sus cuentas bancarias.</p>	2	3	E
36	<p>Permitirá a una institución totalizar la deuda de un ex-becario.</p> <p>Esto permitirá a un ex-becario actualizar la deuda que tiene con la asociación.</p>	2	5	E
37	<p>Permitirá a la asociación realizar un calendario de pagos de la deuda totalizada de un ex-becario.</p> <p>Esto permitirá a la asociación realizar una mejor proyección del dinero que dispondrá a futuro.</p>	2	2	E
38	<p>Permitirá a un alumno (becario o ex-becario) que no pertenezca a TECSUP, registrar un pago a cuenta de su préstamo. Dicho pago solo será válido cuando la asociación lo verifique y acepte.</p> <p>Esto permitirá a la asociación verificar de forma acertada los pagos depositados en la cuenta bancaria.</p>	3	2	E
39	<p>Permitirá a la asociación ingresar un pago o verificar lo registrado por un alumno (becario o ex-becario).</p> <p>Esto permitirá a la asociación tener actualizados los ingresos percibidos.</p>	2	2	E
40	<p>Permitirá al representante de Tecsup registrar los ingresos realizados por sus alumnos en determinado ciclo.</p> <p>Esto permitirá a Tecsup manejar una cuenta bancaria independiente de las otras instituciones.</p>	3	3	E
41	<p>Permitirá a un ex-becario consultar la información correspondiente al calendario de pagos definido para él.</p> <p>Esto permitirá a los ex-becarios tener a su disposición la información correspondiente, en caso suceda algún percance con la información que le proporcionó la asociación en el momento de diseñar y aprobar el calendario de pagos.</p>	3	3	D



<b>Ingresos y egresos - Reembolsos</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
42	Permitirá a los alumnos consultar la información correspondiente a su deuda. Esto permitirá a los alumnos tener a su disposición la información correspondiente para su comprobación.	3	3	D
43	Permitirá a la asociación enviar e-mails masivos o particulares, con recordatorios de pagos a sus alumnos. Esto permitirá a la asociación notificar a sus alumnos si un pago acordado está próximo a cumplirse.	3	4	E
<b>Cálculos y reportes:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
44	Permitirá generar reportes con información de los postulantes, tales como: datos personales, ficha del postulante, información socio-económica e informes. Esto solo para las instituciones y la asociación. Esto permitirá, tanto a las instituciones como a la asociación, imprimir dicha información y tenerla a la mano para la mejor evaluación de los postulantes.	4	2	D
45	Permitirá generar reportes con las notas de los becarios por ciclo. Esto permitirá a la asociación realizar un seguimiento académico a sus becarios, así como para tomar acciones preventivas ante la posibilidad de pérdida de préstamos.	2	4	E
46	Permitirá generar reportes con el historial de empresas en las cuales ha trabajado cada alumno, desde su admisión a la asociación. Esto permitirá a la asociación realizar un seguimiento del desarrollo profesional de sus becarios y ex-becarios.	2	4	D
47	Permitirá generar reportes con los últimos datos de contacto registrados, de los becarios y ex-becarios, tales como: teléfono, correo electrónico y dirección de su vivienda. Esto permitirá a la asociación tener a su disposición la información de contacto de sus alumnos para alguna posible comunicación con ellos.	2	4	D
48	Permitirá a la secretaria ejecutiva generar reportes con el historial de datos de contacto de cada becario y/o ex-becario. Esto permitirá a la secretaria ejecutiva de la asociación realizar un seguimiento de los datos de contacto de sus alumnos, ante una eventual comunicación fallida con ellos.	2	4	D
49	Permitirá generar reportes con la lista de becarios en un ciclo y los montos registrados para pago por cada uno de ellos. Esto permitirá a la asociación tener a su disposición la información de montos a pagar por concepto de pensiones de sus alumnos.	3	4	D
50	Permitirá generar reportes de Student Information, el cual es un documento en inglés visto por el directorio suizo. Esto permitirá al directorio suizo tener a su disposición la información de los becarios para un posible seguimiento.	2	3	D

<b>Cálculos y reportes:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
51	<p>Permitirá a la secretaria ejecutiva hacer cálculos y generar reportes con <b>proyecciones</b> del flujo de caja, en base a ingresos y egresos anteriores.</p> <p>Esto permitirá a la secretaria ejecutiva de la asociación tener a su disposición información crítica para una posible proyección de los gastos a realizar el próximo ciclo.</p>	3	3	D
52	<p>Permitirá a la secretaria ejecutiva generar reportes con <b>proyecciones</b> del presupuesto anual, en base a ingresos y egresos anteriores.</p> <p>Esto permitirá a la secretaria ejecutiva de la asociación tener a su disposición información crítica para una posible proyección del presupuesto requerido para el próximo año.</p>	3	3	D
<b>Seguridad:</b>		<b>Dif.</b>	<b>Pri.</b>	<b>Ex.</b>
53	<p>Permitirá acceder al sistema a postulantes, becarios, ex-becarios, instituciones y a los miembros de la asociación, manejando diferentes permisos.</p> <p>Esto permitirá que los usuarios vean solo lo que les corresponda, además de asegurar la seguridad de la información que maneja el sistema.</p>	2	5	E
54	<p>Validará la autenticación de todo usuario que intente ingresar al sistema.</p> <p>Esto permitirá que solo ingresen al sistema los usuarios registrados.</p>	2	5	E
55	<p>Permitirá a todos los usuarios del sistema modificar su clave de acceso (contraseña).</p> <p>Esto permitirá a los usuarios colocar la contraseña que les sea más fácil de recordar.</p>	3	4	E
56	<p>Permitirá a todos los usuarios del sistema recuperar su contraseña con tan solo ingresar su nombre de usuario. Tras lo cual se enviará un mensaje a su cuenta de correo principal, con los pasos a seguir.</p> <p>Esto permitirá a los usuarios restablecer su contraseña en caso de olvido.</p>	3	5	D
57	<p>Permitirá el registro, modificación y eliminación de la información de usuarios.</p> <p>Esto permitirá al administrador del sistema gestionar la información concerniente a los usuarios que tendrán acceso al sistema.</p>	2	5	E

**Tabla 2.1.** Requerimientos funcionales.

N°	Requerimientos No Funcionales	Dif.	Pri.	Ex.
1	El sistema será desarrollado en lenguaje Java, utilizando el IDE Netbeans v6.5. Esto permitirá minimizar los costos de la etapa de construcción del sistema.	3	5	E
2	El sistema será accesado a través de Internet. Esto permitirá a los usuarios acceder al sistema desde cualquier parte del mundo y en cualquier momento a través de una computadora con conexión a Internet.	5	5	E
3	Se utilizará un manejador de base de datos MySql. Esto permitirá a la asociación no pagar la licencia correspondiente a base de datos.	2	4	E
4	El sistema será intuitivo y de fácil aprendizaje. Para ello se considerarán las opiniones de diseño por parte de los usuarios finales. Esto permitirá a los usuarios familiarizarse y adaptarse rápidamente al sistema.	3	5	E
5	Se presentará documentación con la información técnica adecuada para el uso del sistema. Esto permitirá a los usuarios tener una visión más clara sobre las funcionalidades que abarca el sistema.	3	3	D

**Tabla 2.2.** Requerimientos no funcionales.

Dichos requerimientos han sido agrupados en módulos, los cuales fueron distribuidos entre las tesis de la siguiente manera:

Marllessi Chanca:

- Ingresos y Egresos
- Reembolsos
- Proyecciones
- Seguridad
- Mantenimientos básicos (Datos del personal administrativo y Conceptos)

Jessica Gutiérrez:

- Postulación
- Ajustes de ciclo
- Actualización de datos personales
- Reportes
- Mantenimientos básicos (Contenidos de correos electrónicos, Tipos de trabajo)

### 2.3. Análisis de la solución

El análisis del presente proyecto se realizará considerando cinco puntos de vista: la viabilidad del sistema como solución a un nivel técnico, el análisis técnico - económico del proyecto, la asignación de funciones a los elementos del sistema, las restricciones de costo y tiempo dentro del cual el proyecto se realiza, y la definición del sistema base para un trabajo posterior de ingeniería.

#### 2.3.1. Viabilidad del sistema

El sistema planteado no sólo ha recogido los procedimientos que maneja la asociación educativa, sino que existen diversos productos y propuestas en el mundo que corroboran la facilidad y orden que proporciona realizar un sistema de préstamos educativos (ver la sección “Estado del arte” del Capítulo 1). Además, se han consolidado múltiples reuniones con los diferentes tipos de usuarios finales del sistema para así recoger todas las perspectivas y necesidades que cada uno tiene.

En conclusión, con la realización del presente proyecto se pretende solucionar la problemática específica peruana teniendo en cuenta las mejores características de sistemas extranjeros. Así, se puede anticipar que el sistema final será un éxito.

#### 2.3.2. Análisis técnico – económico

En cuanto al análisis a nivel económico, se puede mencionar que el usar software de desarrollo y *frameworks* libres, minimiza notablemente el costo del sistema pues ya no supone el costo por adquisición de licencias. Por ello, sólo habría que considerar el costo que produce el tiempo de la construcción del proyecto. En la tabla 2.3 se muestra la estimación de los costos.

Concepto	Horas	Costo / Hora	Costo
Levantamiento de la información	80	S/. 10	S/. 800.00
Capacitación del equipo	300	S/. 10	S/. 3000.00
Implementación del prototipo	300	S/. 15	S/. 4500.00
Implementación del sistema	4080	S/. 20	S/. 81600.00
Capacitación de usuarios finales	80	S/. 15	S/. 1200.00
Gasto en útiles de oficina, impresiones y anillados			S/. 600.00
Gasto en pasajes			S/. 800.00
<b>Inversión Total</b>			<b>S/. 92500.00</b>

**Tabla 2.3.** Estimación de costos de construcción del proyecto.

En cuanto al análisis a nivel técnico, se puede mencionar que actualmente muchas o todas las plataformas existentes pueden soportar la tecnología necesaria para desarrollar el sistema planteado, además que se puede configurar la base de datos localmente. En la fase de producción, será necesario un servidor web o servicio de *hosting* para la base de datos y que los usuarios tengan computadoras con acceso a Internet.

### 2.3.3. Asignación de funciones

En la tabla 2.4 se listan los elementos involucrados en el sistema y las funciones asignadas a cada uno de ellos.

Elemento	Función asignada
Software	Para el desarrollo y pruebas del sistema, se necesitará el IDE Netbeans, MySql como sistema administrador de base de datos y Glassfish para la administración de un servidor de aplicaciones. Para el usuario final, bastará con tener sistema operativo Windows 2000 en adelante o alguna distribución Linux.
Hardware	Para el desarrollo y pruebas del sistema, se necesitará trabajar como mínimo con procesador Pentium IV, 512 de RAM y 100Mb de disco duro libre, para un rápido procesamiento.
Recursos	Se cuenta con dos tesisistas a cargo de todas las etapas implicadas en la construcción y pruebas del sistema.
Base de datos	Se trabajará con la base de datos a través de MySql, la cual será alojada en un servidor que corra en Linux.

Tabla 2.4. Asignación de funciones.

### 2.3.4. Restricciones de costo y tiempo

En cuanto al costo, como ya se ha mencionado anteriormente, sólo se tendrá en consideración el tiempo invertido en la construcción (aprendizaje y elaboración).

En cuanto al tiempo, se ha estimado que la duración del proyecto es de trescientos treinta y cuatro días, considerando ocho horas diarias de trabajo. Así, se calcula que el proyecto tendrá una vida de cuatro mil ochenta horas totales.

$$\begin{array}{l}
 \text{Vida del proyecto} = 334 \text{ días} \\
 \text{Horas de Trabajo diario} = 8 \text{ horas} \\
 \text{Miembros del equipo} = 2 \text{ personas}
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Vida del proyecto} \\ \text{Horas de Trabajo diario} \\ \text{Miembros del equipo} \end{array}} \right\} \text{Esfuerzo referencial dedicado} = (1264 + 2816) \text{ horas} \\
 = 4080 \text{ horas}$$

### 2.3.5. Definición del sistema

La solución planteada en el presente proyecto permitirá el acceso al directorio suizo, representantes de Asociación Educación en Perú, instituciones, postulantes, becarios y ex-becarios, por lo que las funcionalidades del sistema serán divididas entre este tipo de personas. En la figura 2.1 se ilustra el catálogo de actores del sistema.

Realizando un análisis de los requerimientos obtenidos, se definieron los siguientes módulos que contemplará el sistema propuesto:

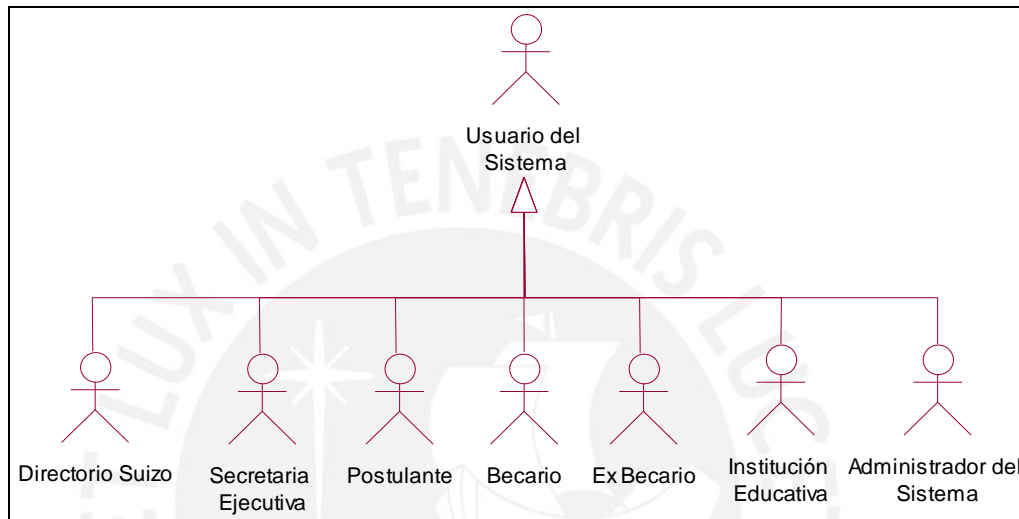
- **Postulación:** Incluye todos los procesos relacionados a la postulación, tales como: publicación de información de vacantes y entrevistas (convocatoria), registro y modificación de información de los postulantes, registro y modificación de informes sobre situación económica de los postulantes, asignación de fechas y horarios de entrevistas, y admisión de postulantes.



- **Ajustes de ciclo:** Incluye todos los procesos relacionados al mantenimiento de información relevante a inicio, mitad y fin de cada ciclo de estudio en las instituciones educativas.
  - Para **Inicio de Ciclo**, se tiene: actualización del valor del crédito por escala de pago y carrera en el ciclo vigente, actualización del factor de ajuste (en caso aplique), selección de becarios que continuarán con el préstamo, registro y modificación de total de créditos matriculados por becario, registro y modificación de montos totales de pensiones por cada becario, y registro de montos totales que la asociación debe pagar por concepto de pensiones a cada institución. Por su parte, la asociación puede consultar dichos montos.
  - Para **Mitad de Ciclo**, se tiene: registro y modificación de promedio de notas parciales y cantidad de créditos reprobados de cada becario, considerando solo las notas registradas en las instituciones en el ciclo vigente. Por su parte, la asociación puede consultar dichos promedios de notas parciales por becario.
  - Para **Fin de Ciclo**, se tiene: registro y modificación de promedio de notas finales y el total de créditos reprobados de cada becario. Por su parte, la asociación puede consultar dichos promedios de notas finales por becario y, según los resultados, podría expulsar a algún becario posteriormente.
- **Actualización de datos personales:** Incluye el mantenimiento de la información de becarios, ex-becarios e instituciones educativas, como parte de un seguimiento continuo que desea realizarse. En el caso de instituciones educativas, se incluye también el mantenimiento de las facultades (en caso aplique) y las carreras respectivas.
- **Reportes:** Incluye la exportación a archivos en formato PDF de: Datos personales, Ficha del Postulante, Información Socio-económica e Informes de los postulantes; Historial de desarrollo profesional de los becarios y ex-becarios; Historial de datos de contacto por alumno; Historial de datos generales y de contacto de todos los alumnos; Liquidación de pagos de pensiones de los becarios por cada institución; Notas de los becarios por ciclo; Flujo de Caja Semestral y Presupuesto Anual.
- **Ingresos y egresos:** Incluye el registro y modificación de ingresos (todo aquello que la asociación perciba, a excepción de los reembolsos), el registro y modificación de egresos (todo aquello que la asociación paga a terceros, incluyendo el pago de pensiones a las instituciones) y el registro de transferencias.
- **Reembolsos:** Incluye el registro de pagos de un becario o ex-becario, según sea la institución educativa a la que pertenece (si pertenece a una institución diferente de la lógica Tecsup el registro es directo en el sistema pero si pertenece a una institución de lógica Tecsup será el representante de dicha institución quien haga el registro luego de verificar la conformidad en la cuenta bancaria que administran). A su vez, comprende el registro y modificación de la información de la totalización de una deuda y del calendario de pagos de cada alumno.
- **Proyecciones:** Incluye la visualización del flujo de caja estimado para un semestre determinado, considerando datos registrados en semestres anteriores. También incluye la visualización del presupuesto anual estimado, considerando datos registrados en años anteriores.



- **Seguridad:** Incluye todo aquello relacionado a la seguridad del sistema, de modo que pueda lograrse confidencialidad durante la transferencia de información crítica de la organización.
- **Mantenimientos básicos:** Incluye todos aquellos mantenimientos de información básica con las cuales el sistema trabajará, tales como: datos del personal administrativo (comprende asignación de roles), contenidos predeterminados de los correos electrónicos que se enviarán a través del sistema, tipos de trabajo de las personas y conceptos por los cuales se realizarán los ingresos o egresos.



**Figura 2.1.** Actores del sistema.

Realizando un análisis exhaustivo de los procesos que manejaría el sistema, y considerando las entrevistas y reuniones con diversos tipos de usuarios, se han identificado los siguientes procesos principales: Postulación, Inicio de ciclo, Mitad de ciclo, Fin de ciclo, Totalización de una deuda e Ingresos por pagos de ex-becarios. En la figura 2.2 se ilustra el proceso de Postulación.

En el Anexo C: Diagrama de actividades de procesos principales, se encontrarán los diagramas de los demás procesos mencionados. Mientras que en el Anexo B: Descripción de los Procesos Principales, se puede encontrar una descripción detallada del flujo realizado en cada proceso.

A partir de la información descrita sobre dichos procesos, se ha elaborado un diagrama de clases de análisis que se considerarán para el desarrollo del sistema planteado, ilustrándose en la figura 2.3.

En dicho diagrama de clases de análisis, se puede encontrar a la clase BEPersona, la cual alberga información resumida de la persona, representa todo tipo de usuario que ingresa al sistema y de la que heredan las tablas BEPostulante, BEBecario, BEExbecario. Estas tres clases representan la clasificación de los tipos de alumno que existen en el sistema.

- La primera clase (BEPostulante), se relaciona con BEFichaPostulante (para poder guardar sus datos generales), BEInforme (para poder guardar los informes y observaciones que hacen los representantes de la institución a la

que pertenece el postulante y posteriormente también la secretaria académica) y BEInfSocioEconom (para mayor detalle de la situación económica del postulante).

- La segunda clase (BEBecario), además de guardar información actualizada del becario, se relaciona directamente con BEBecarioxCiclo, la cual contiene el detalle del ciclo en el que el alumno se matriculó. Asimismo, esta clase es indispensable para el proceso de ingresos y egresos (conociendo los montos prestados por ciclo).
- La tercera clase (BEExbecario), contiene información actualizada del exbecario y su deuda (si fue expulsado y si su deuda fue condonada). Asimismo, da soporte a la clase BEPosiblePago cuando el alumno solicita un cronograma de pagos. Por otro lado, la clase BEExbecarioxCiclo es importante solamente para alumnos que pertenezcan a una institución cuya lógica sea parecida a la de la institución Tecsup, pues en esos casos su deuda está sujeta a un factor de ajuste -determinado por la institución- que cambia cada ciclo.

Adicionalmente, se cuenta con clases como BEInstitucion, BEFacultad, BECarrera que manejan información actualizada de las carreras que existen por facultad en cada institución.

Finalmente, las clases BEEgreso, BEIngreso y BECuentaBancaria básicamente se usan en el registro y administración económica de la asociación educativa.

En el Anexo D: Diagrama de Clases de Análisis de la Solución, se puede encontrar el mismo diagrama de clases de análisis del sistema de forma ampliada para una mejor visualización de las clases y tipos de atributos.

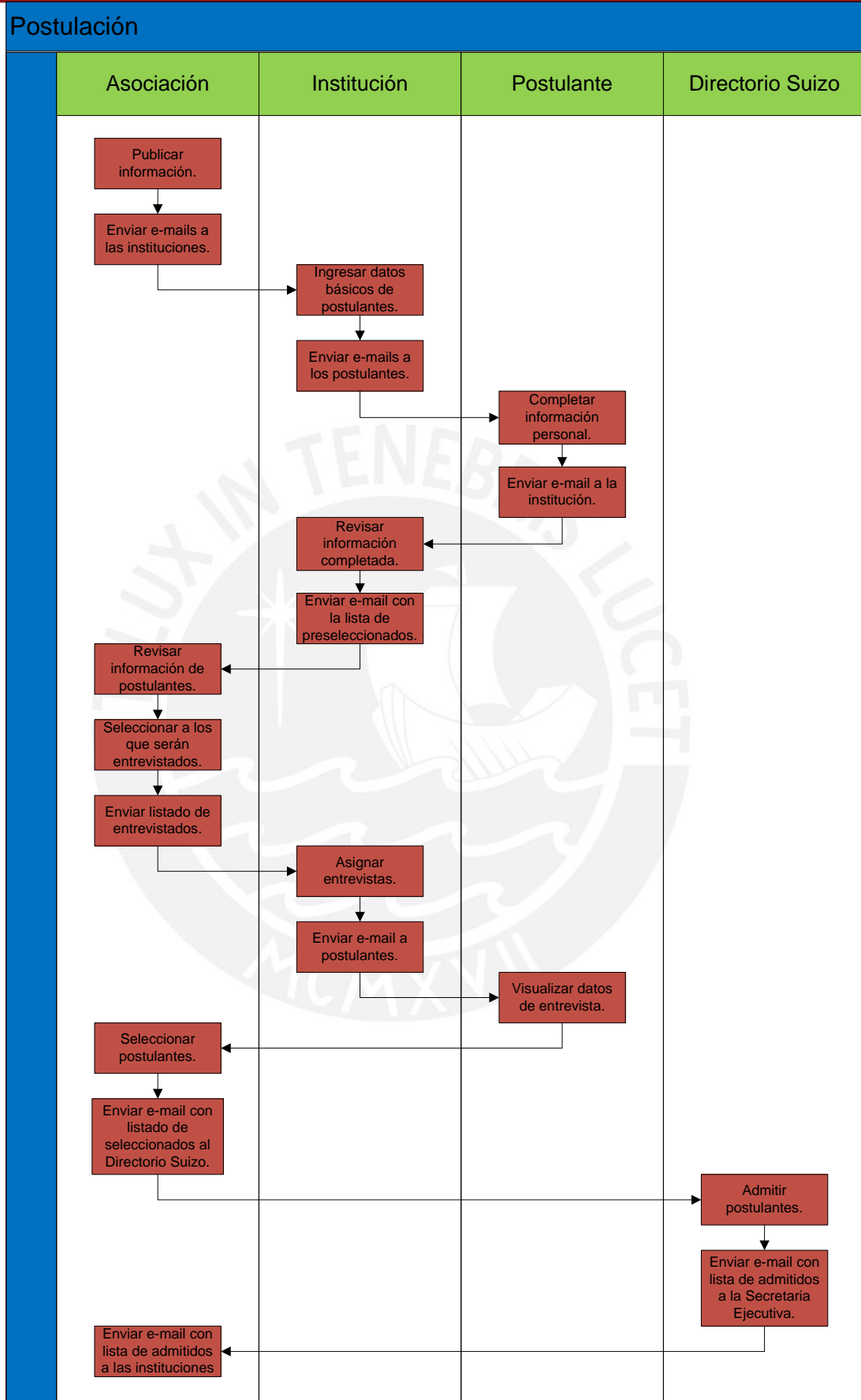


Figura 2.2. Diagrama de actividades del proceso de Postulación.

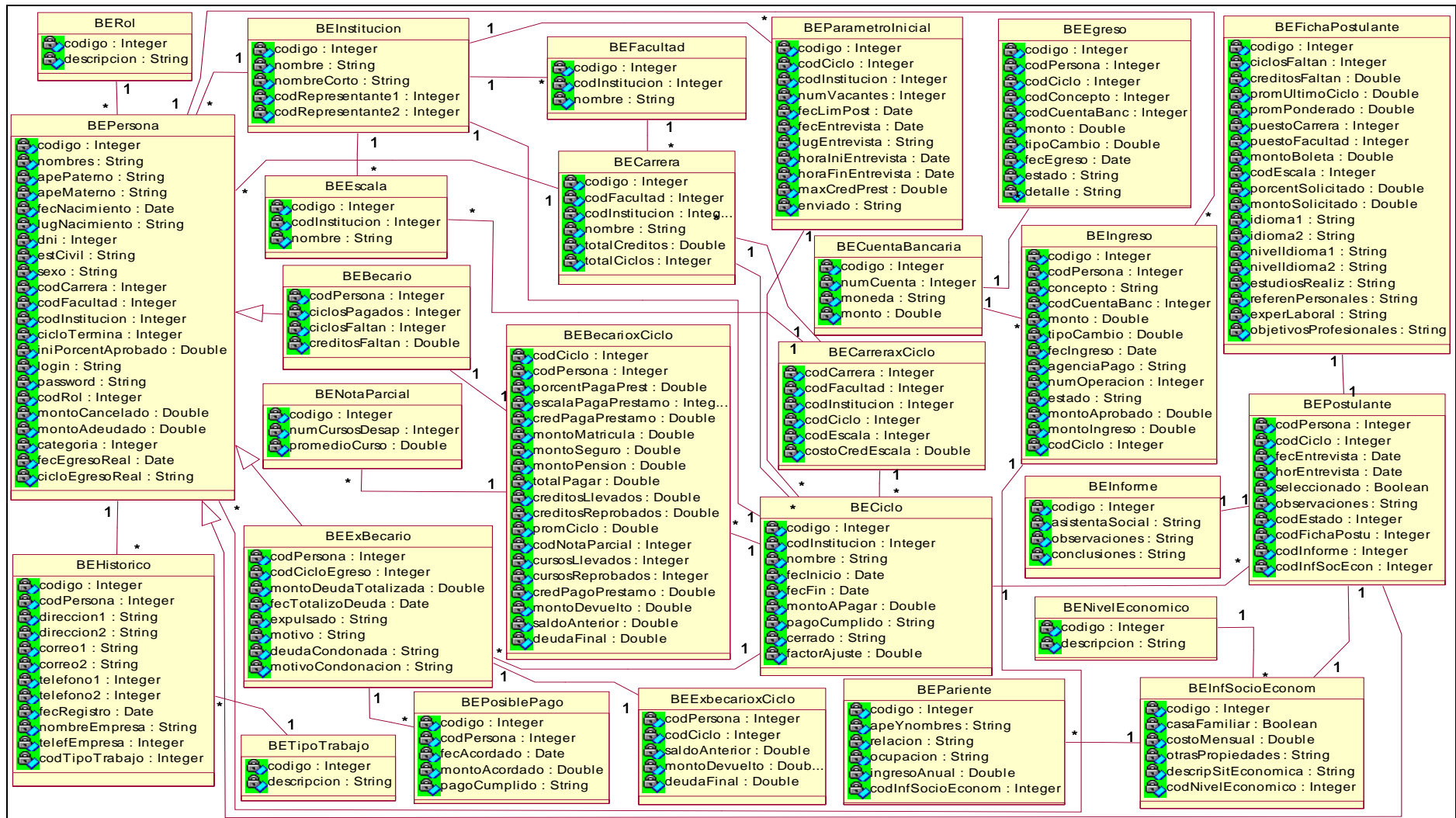


Figura 2.3. Diagrama de clases de análisis.

### 3. Diseño

En el presente capítulo se definirá la arquitectura y el diseño de la interfaz gráfica de la solución propuesta.

#### 3.1. Arquitectura de la solución

Dentro de esta sección se discutirá la definición de los componentes, las aplicaciones consideradas para la integración, así como la infraestructura tecnológica que dará soporte a la solución propuesta.

Cabe resaltar que la justificación de las tecnologías a utilizar se describe en el siguiente capítulo.

##### 3.1.1. Definición de la arquitectura

Considerando los requerimientos planteados, se escogió desarrollar la aplicación en lenguaje Java, ya que permitirá brindarle portabilidad al sistema, utilizando el IDE NetBeans.

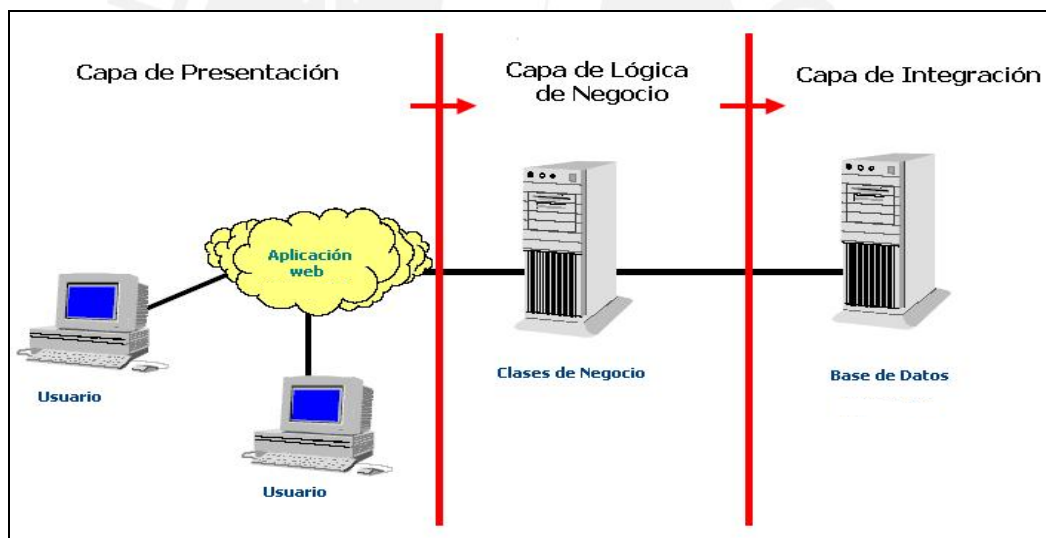
Asimismo, se eligió el uso del gestor de base de datos relacional *MySQL*, aunque no ofrece tanta robustez como *PostgreSQL* [DPE 2007], ya que permitirá un buen rendimiento de acceso a los datos para el modelo de base de datos planteado. Además, es de mencionar que no existirá un gran número de conexiones que accedan continuamente al sistema web a realizar.



Por otro lado, se escogió una arquitectura web de 3 capas a fin de separar los componentes de arquitectura, de modo que sea más flexible y fácil de realizar mantenimiento al sistema. Las capas a considerar son las siguientes:

- **Capa de Presentación:** Compuesta por la interfaz gráfica de usuario que permitirá el acceso a la lógica de negocio a través de formularios web, pudiendo utilizar un marco de trabajo que facilite el desarrollo gráfico.
- **Capa de Lógica de Negocio:** Contiene todas las clases que definen la complejidad del negocio, el flujo y las reglas del proceso de negocio, así como las validaciones pertinentes de los datos que van a ser almacenados en la base de datos a través de un puente llamado Capa de Acceso a Datos. Estas clases comprenden principalmente los métodos que se reducen a operaciones elementales como crear, actualizar, eliminar o listar registros, pudiendo utilizar un marco de trabajo que permita una mejor y más sencilla programación.
- **Capa de Integración:** Maneja el acceso a la base de datos relacional planteada, pudiendo utilizar un marco de trabajo que mantenga la persistencia de los mismos. Estará compuesta por los gestores de base de datos que resolverán las peticiones de recuperar y almacenar información.

En la figura 3.1 se ilustra la interacción de las capas de la arquitectura planteada.



**Figura 3.1.** Diseño de la arquitectura del sistema.

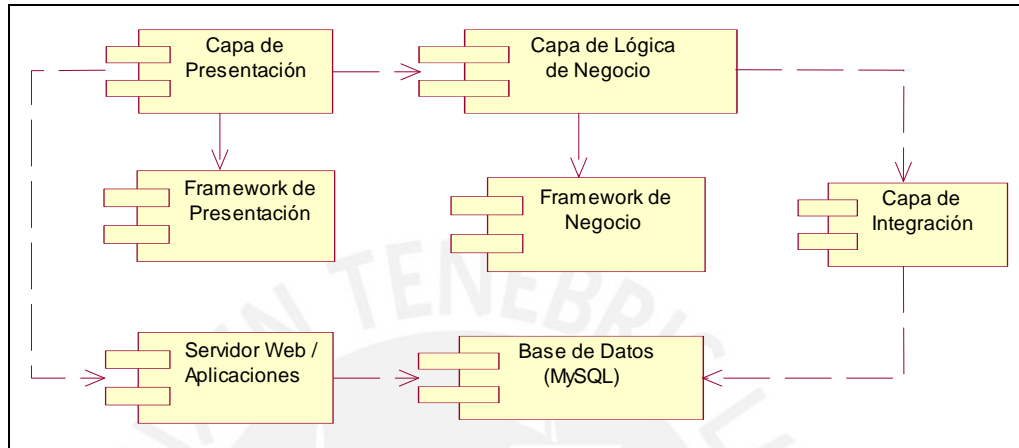
Cabe resaltar que para la elección de la arquitectura se consideró la persistencia de información, mecanismos de seguridad y reutilización de software, descritos a continuación.

- **Persistencia de información:** la arquitectura mantendrá la integridad de las estructuras asegurando que no estén incompletas o corruptas, por esto la información estará centralizada y actualizada.
- **Mecanismos de seguridad:** el usuario deberá iniciar sesión en el sistema, lo cual asegurará un acceso restringido a las funcionalidades del sistema dependiendo de los perfiles de usuario.

- **Reutilización de software:** la arquitectura será definida para poder usar encapsulamiento y herencia en la codificación.

### 3.1.2. Definición de componentes

En la figura 3.2 se presentan los componentes a utilizar en la arquitectura del sistema.



**Figura 3.2.** Diagrama de componentes de la arquitectura del sistema.

La descripción de dichos componentes se muestra en la tabla 3.1.

Componente	Descripción
Capa de Presentación	Componente que representa la interfaz gráfica de usuario.
Capa de Lógica de Negocio	Componente que define las clases y lógica de negocio que interactuarán con la base de datos definida.
Capa de Integración	Componente que define el acceso a los datos.
Base de Datos	Componente que representa a la base de datos definida.
Framework de Presentación	Componente que permite el mejor manejo y desarrollo de la interfaz gráfica de usuario.
Framework de Negocio	Componente que permite el mejor manejo de buenas prácticas de programación.
Servidor Web / Aplicaciones	Componente que almacena la aplicación web, pudiendo mostrarla a través de Internet para la interacción con los usuarios.

**Tabla 3.1.** Descripción de componentes de la arquitectura.

### 3.1.3. Beneficios de la arquitectura

En esta sección se describirán los beneficios de la arquitectura elegida, beneficios por los cuales se entiende que resulta propicia para los fines del presente proyecto de fin de carrera.

En la tabla 3.2 se muestra la relación entre los beneficios de la arquitectura elegida y cada uno de los siete requerimientos no funcionales (NF1 a NF7) descritos en la sección 2.2 “Identificación de requerimientos” del presente documento.

Característica	Descripción	Requerimientos no funcionales				
		NF1 Java / Netbeans	NF2 Internet	NF3 MySQL	NF4 Aprendiza.	NF5 Docum.
Flexibilidad	Los componentes pueden ser reemplazados o reorganizados ágilmente para crear nuevos procesos de negocio.  Utilización de modelo de capas.			X	X	
Reutilización	Permite el uso de encapsulamiento y herencia en la codificación.	X			X	
Escalabilidad	Permite que en un futuro se pueda ampliar el sistema sin perder calidad en el rendimiento.	X		X		X
Portabilidad	Permite que la aplicación sea utilizada en sistemas operativos Windows o Linux.  Utilización de lenguaje Java y tecnologías compatibles.	X				
Uso de software libre	Uso de software libre para reducir los costos de desarrollo e implementación.  Utilización de MySql y frameworks de software libre.	X		X		X
Disponibilidad de uso a cualquier hora	Uso de un modelo web, que permite el acceso a la aplicación en cualquier momento a través de Internet.	X	X			

**Tabla 3.2.** Beneficios de la arquitectura.

## 3.2. Diseño de interfaz gráfica

### 3.2.1. Criterios para el diseño de la interfaz gráfica

Con el objetivo de uniformizar la interfaz gráfica (diseño de páginas) y hacerlo más intuitivo o fácil de manejar, se mencionarán las consideraciones que se deben tomar al diseñar el software:

- Se identificará el contenido principal a mostrar para resaltarlo en una ubicación estratégica o colocarlo con un color más llamativo.
- Se usarán colores que no cansen la vista del usuario.
- Se hará uso de gráficos que reflejen acciones determinadas, en los botones y fondos principales.
- Se colocarán nombres adecuados a todas las páginas para que el usuario sepa dónde está, en todo momento.
- Se mostrarán enlaces de rápido acceso a otras operaciones.

### 3.2.2. Estructura general del sitio

En el sistema a elaborar se ha identificado el siguiente esquema general de trabajo:

- LOGIN  
Notificaciones de contraseña
- OPERACIONES  
Ingreso y egreso de dinero  
Postulación  
Operaciones de ciclo
- REPORTE  
Reportes de postulantes  
Reportes de becarios  
Reportes de ex-becarios
- CÁLCULOS  
Proyección de vacantes  
Flujo de caja
- MANTENIMIENTOS  
Mantenimiento de instituciones  
Mantenimiento de ciclos  
Mantenimiento de carreras  
Mantenimiento de facultades  
Mantenimiento de personal administrativo  
Mantenimiento de conceptos  
Mantenimiento de correos
- SEGURIDAD  
Cambio de contraseña  
Restablecimiento de contraseña

En la figura 3.3 se ilustra la estructura general del sitio web que se propone.

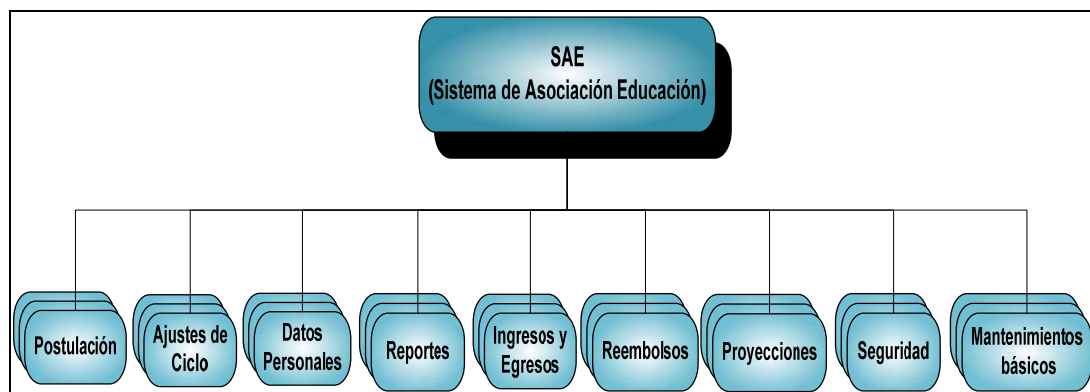


Figura 3.3. Estructura general del sitio.

### 3.2.3. Modelo de ventanas

En las figuras 3.4, 3.5, 3.6, 3.7 y 3.8 se muestran algunos ejemplos de los tipos de contenidos de las ventanas que siguen el esquema de trabajo mencionado.

**Ventana de bienvenida.** En esta ventana se muestra el mensaje de bienvenida del sistema.

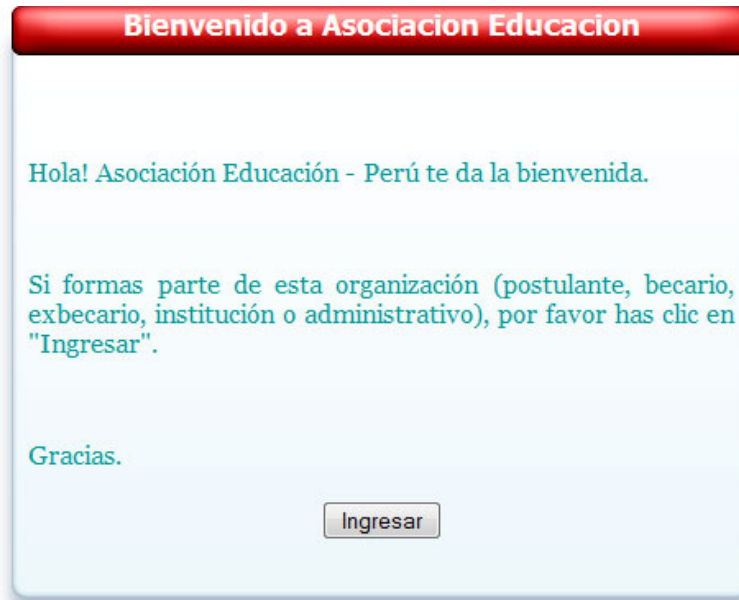


Figura 3.4. Ventana de bienvenida.

**Ventana de ingreso al sistema.** En esta ventana el usuario ingresa su nombre de usuario y contraseña de acceso.



Figura 3.5. Ventana de login.



**Ventana de menú de opciones.** En esta ventana el usuario tiene la posibilidad de elegir la funcionalidad que desea utilizar.



**Figura 3.6.** Ventana de Menú de opciones.

**Ventana de tipo mantenimiento.** En esta ventana se puede realizar una búsqueda utilizando ciertos parámetros o filtros, así como ingresar un nuevo registro, modificarlo o eliminarlo.



**Figura 3.7.** Ventana de tipo Mantenimiento.

**Ventana de tipo operación.** Esta ventana permite realizar operaciones propias del negocio.



**Nueva convocatoria**

Nombre Institución:  Ciclo:

**Datos de la Convocatoria:**

N° Vacantes:  Máximo de créditos a prestar:

Fecha límite para enviar preseleccionados:

**Datos de la Entrevista:**

Lugar de entrevista:  Fecha de entrevista:

Hora inicio de entrevista:  :  A.M. Hora fin de entrevista:  :  A.M.

Recuerde que si no aparece alguna institución, se debe a que esta ya tiene una convocatoria asociada para el presente ciclo.

**Figura 3.8.** Ventana de tipo Operación.

### 3.3. Diseño de la solución

En esta sección se presentará el diseño detallado de la solución planteada, proporcionando los diagramas que describen el diseño de la base de datos, las clases de diseño de la solución de acuerdo a los principales procesos definidos anteriormente y la interacción de los usuarios con los distintos componentes de las capas de la arquitectura definida.

#### 3.3.1. Diagrama de clases de base de datos

En la figura 3.9 se pueden observar todas las tablas que contendrá la base de datos del sistema planteado. Dicho diagrama soporta todos los procesos y requerimientos encontrados.

En el Anexo E: Diseño de Base de Datos, se puede encontrar el mismo diagrama de forma ampliada para una mejor visualización de las clases y atributos.

Observación:

- Se tiene la tabla padre PERSONA, de la cual heredan: POSTULANTE, BECARIO y EXBECARIO.

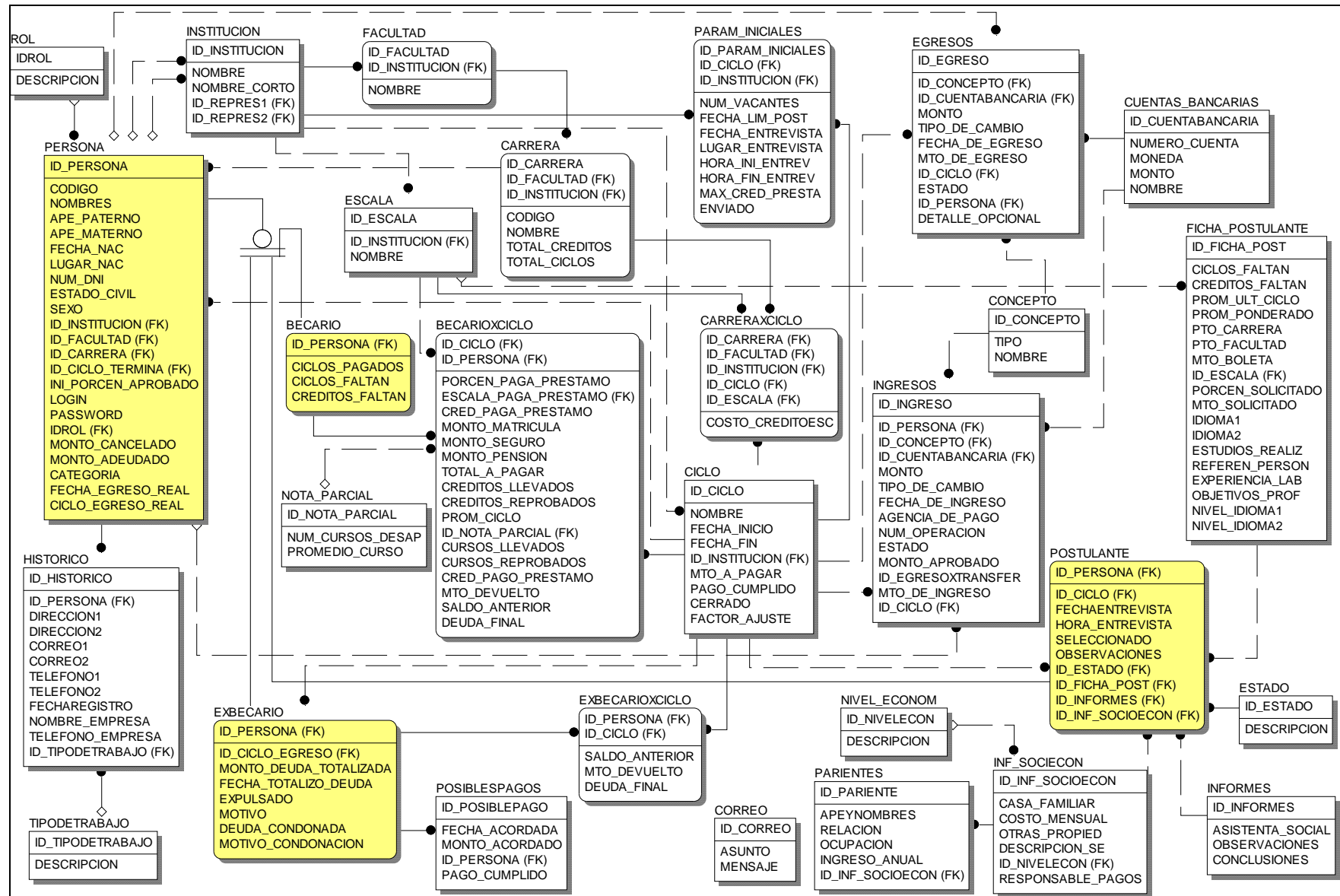


Figura 3.9. Diagrama de clases de base de datos.







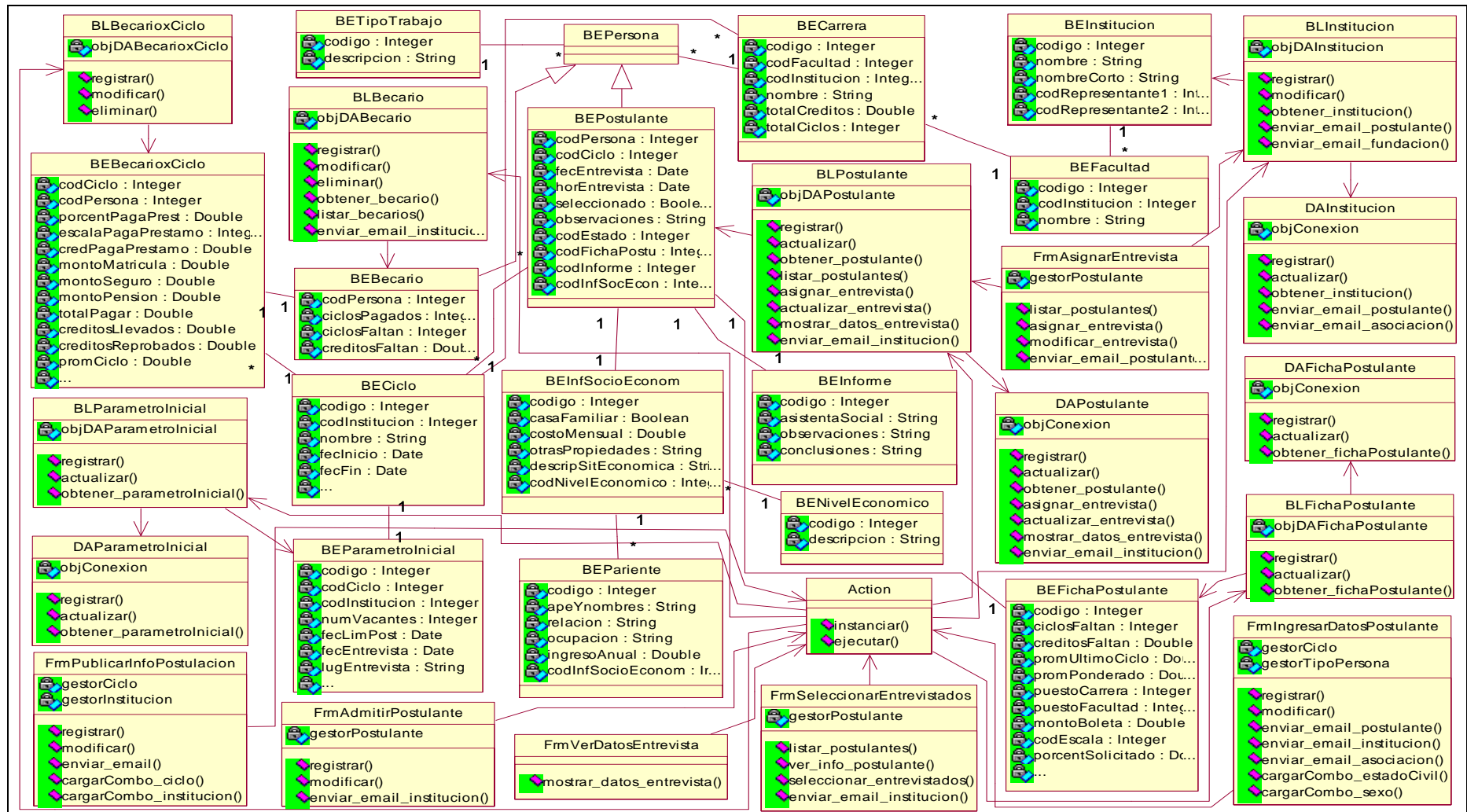
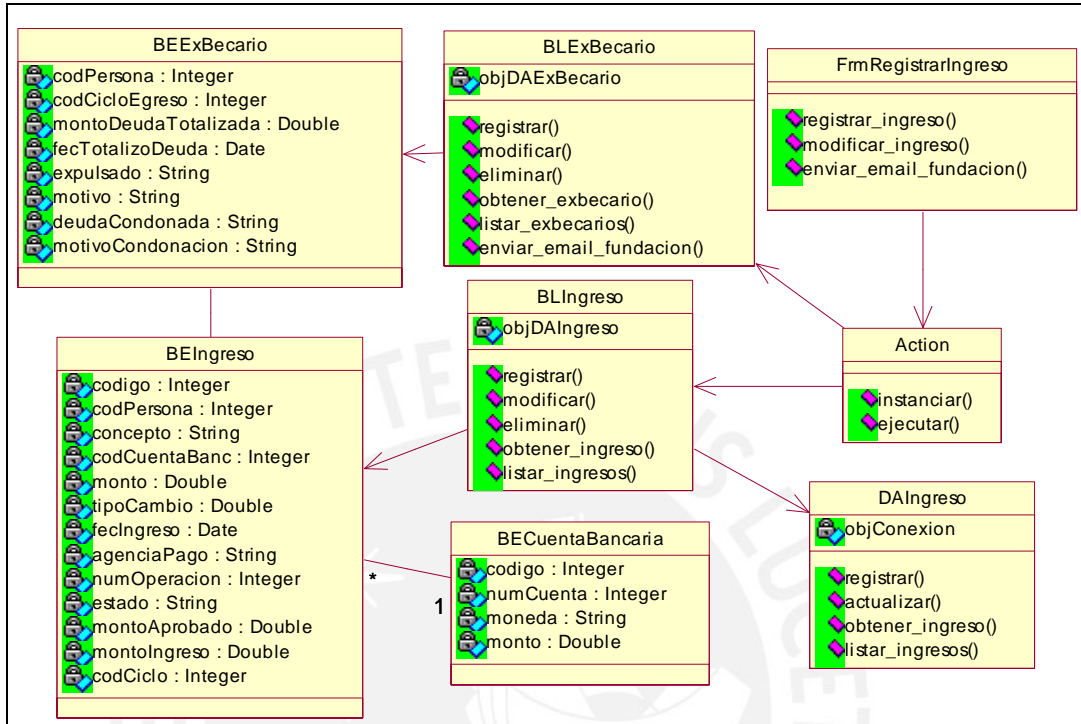


Figura 3.12. Diagrama de clases de diseño – Proceso de postulación.

- **Proceso de Ingresos por pagos de ex-becarios**

En la figura 3.13 se muestran las clases de diseño implicadas en el registro y verificación del lugar de pago, número de operación y el monto abonado en un banco por parte de un ex-becario.



**Figura 3.13.** Diagrama de clases de diseño – Proceso de ingresos por pagos de ex-becarios.

### 3.3.3. Diagrama de secuencia

A continuación se muestran los diagramas de secuencia de los procesos de negocio más complejos, basados en las clases de diseño presentadas anteriormente. En el Anexo F: Diagramas de Secuencia de los Procesos Principales, se pueden encontrar otros diagramas de secuencia del sistema.

Observación: La clase Action es necesaria para la interacción de las demás clases con la capa de presentación (páginas o *frames*, representados por el prefijo 'Frm'), según el marco de trabajo que se utilizará (Struts v2.0).

- **Proceso de Postulación**

#### **Publicar Información**

En la figura 3.14 se ilustra la secuencia de eventos que la Secretaria Ejecutiva de la asociación seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la publicación de la información de vacantes, fechas, horarios y lugar de entrevistas.

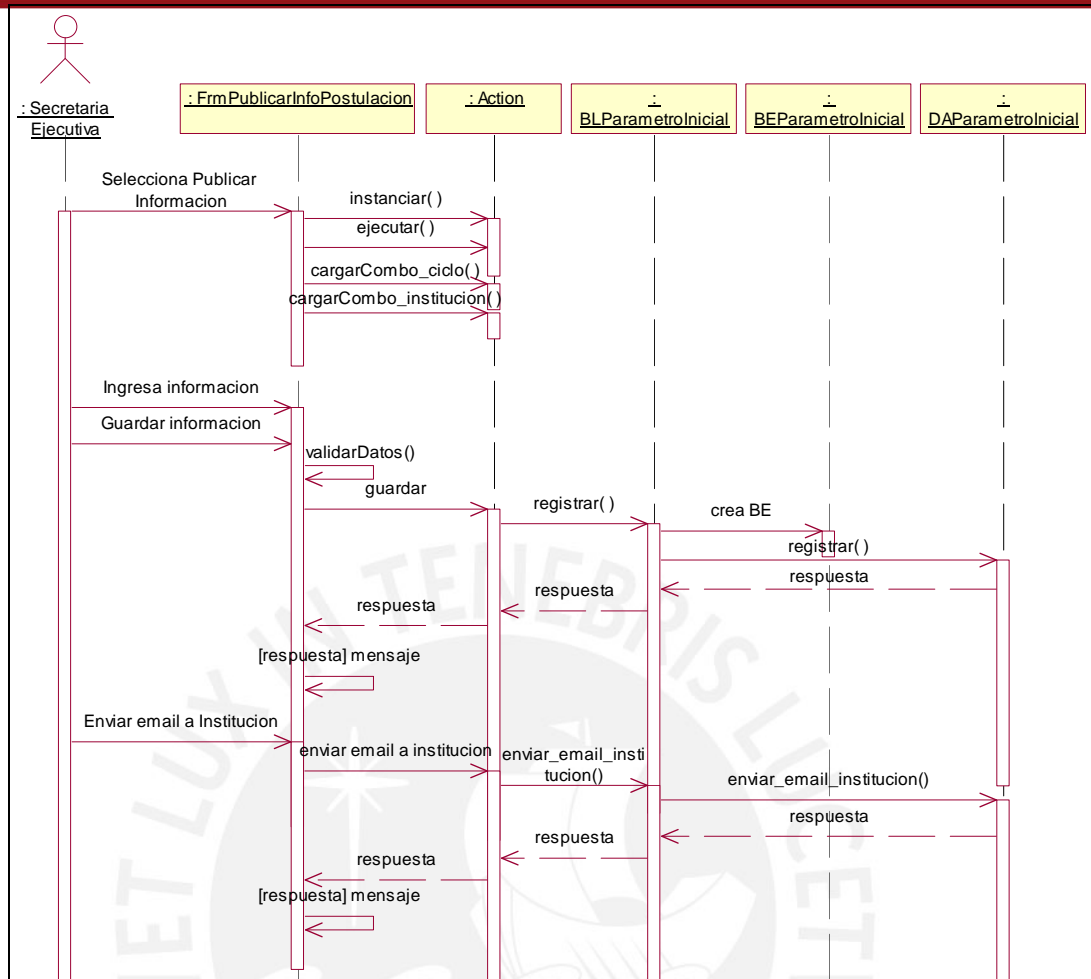
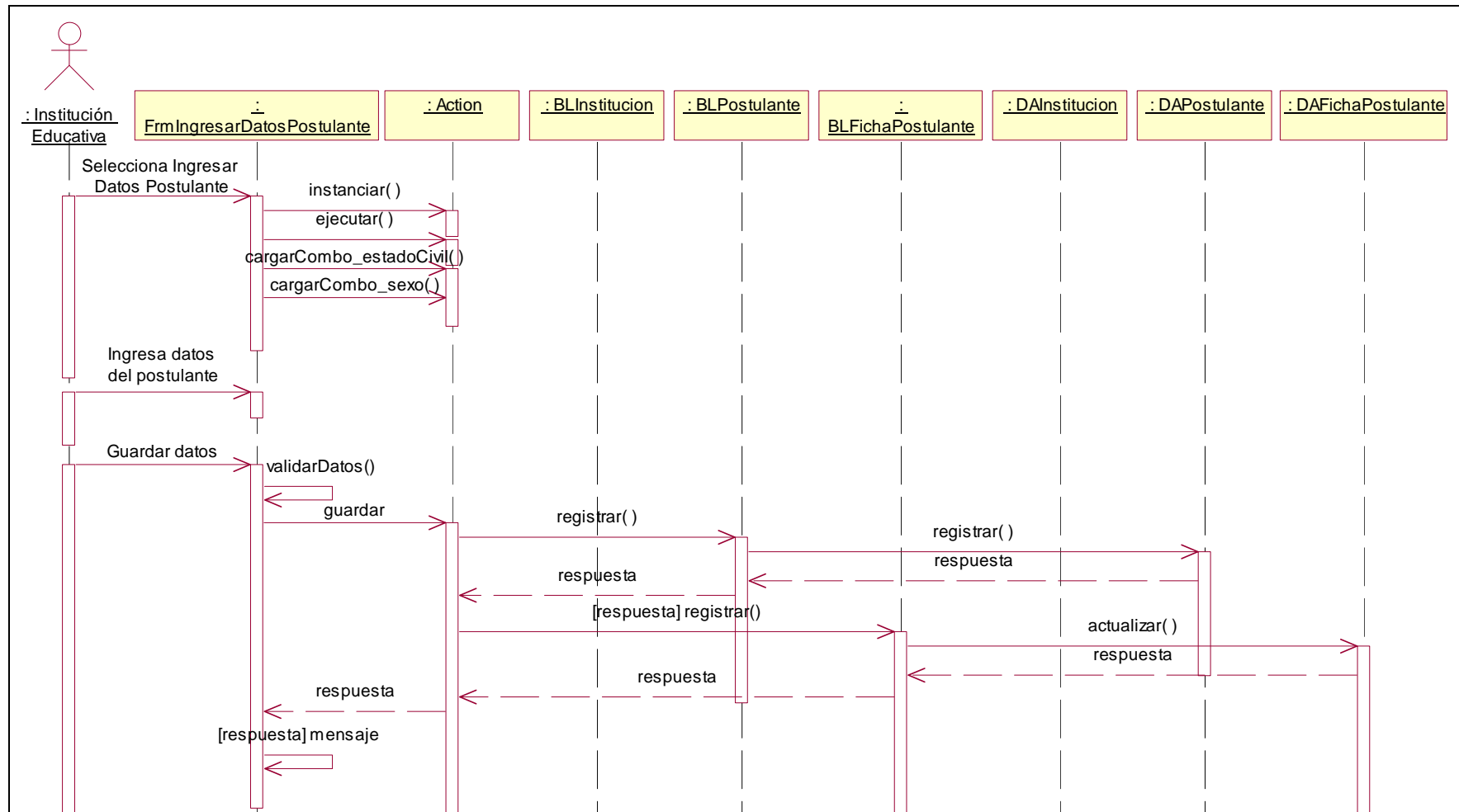


Figura 3.14. Diagrama de secuencias – Proceso de Postulación: Publicar Información.

### Ingresar Datos del Postulante - Institución

En la figura 3.15 se ilustra la secuencia de eventos que cada Institución Educativa seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo el registro de los datos básicos de los postulantes.



**Figura 3.15.** Diagrama de secuencias – Proceso de Postulación: Ingresar datos - Institución.

### Ingresar Datos del Postulante - Postulante

En la figura 3.16 se ilustra la secuencia de eventos que cada Postulante seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo el registro de su información personal y la ficha del postulante que le corresponde.

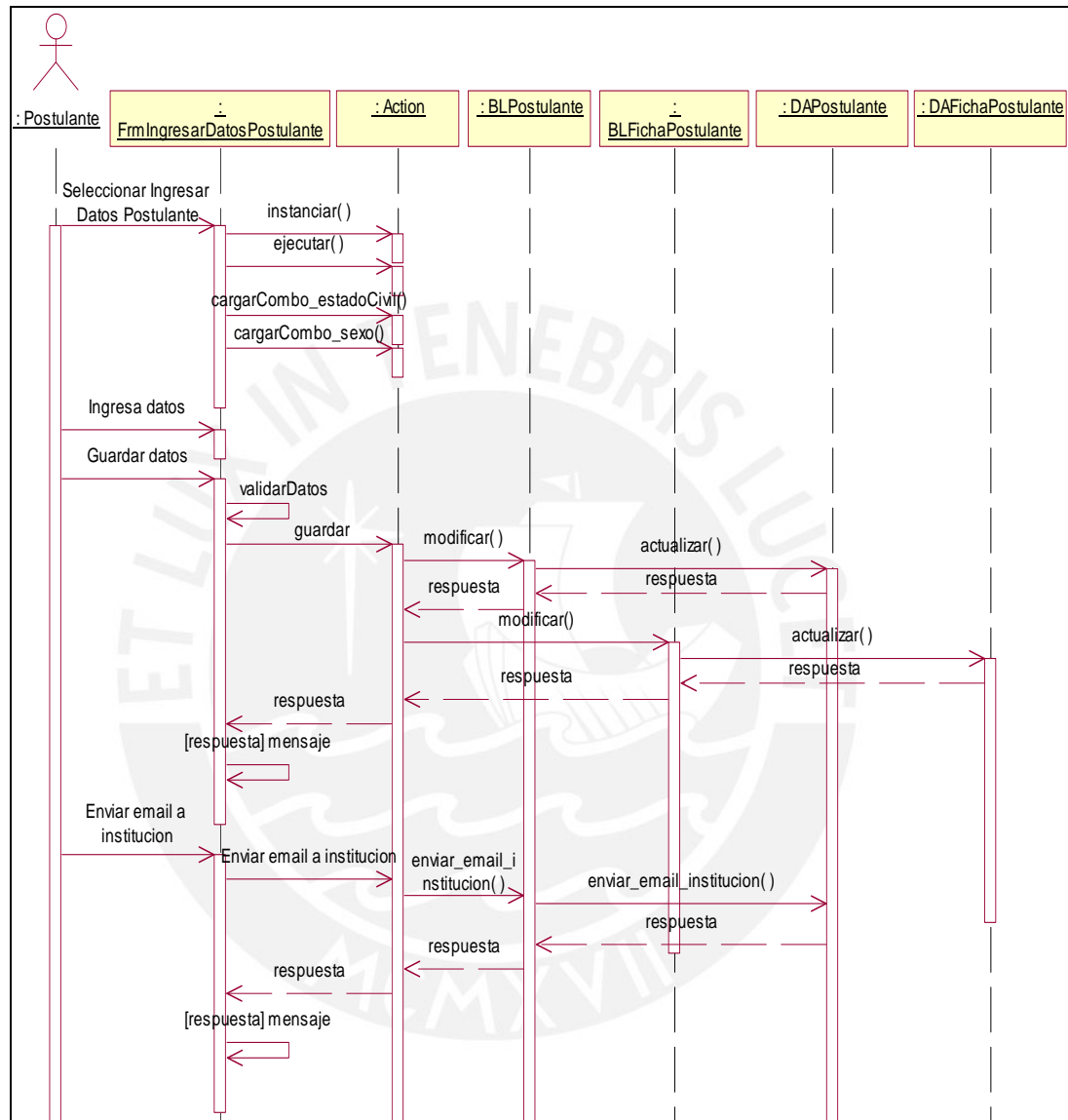
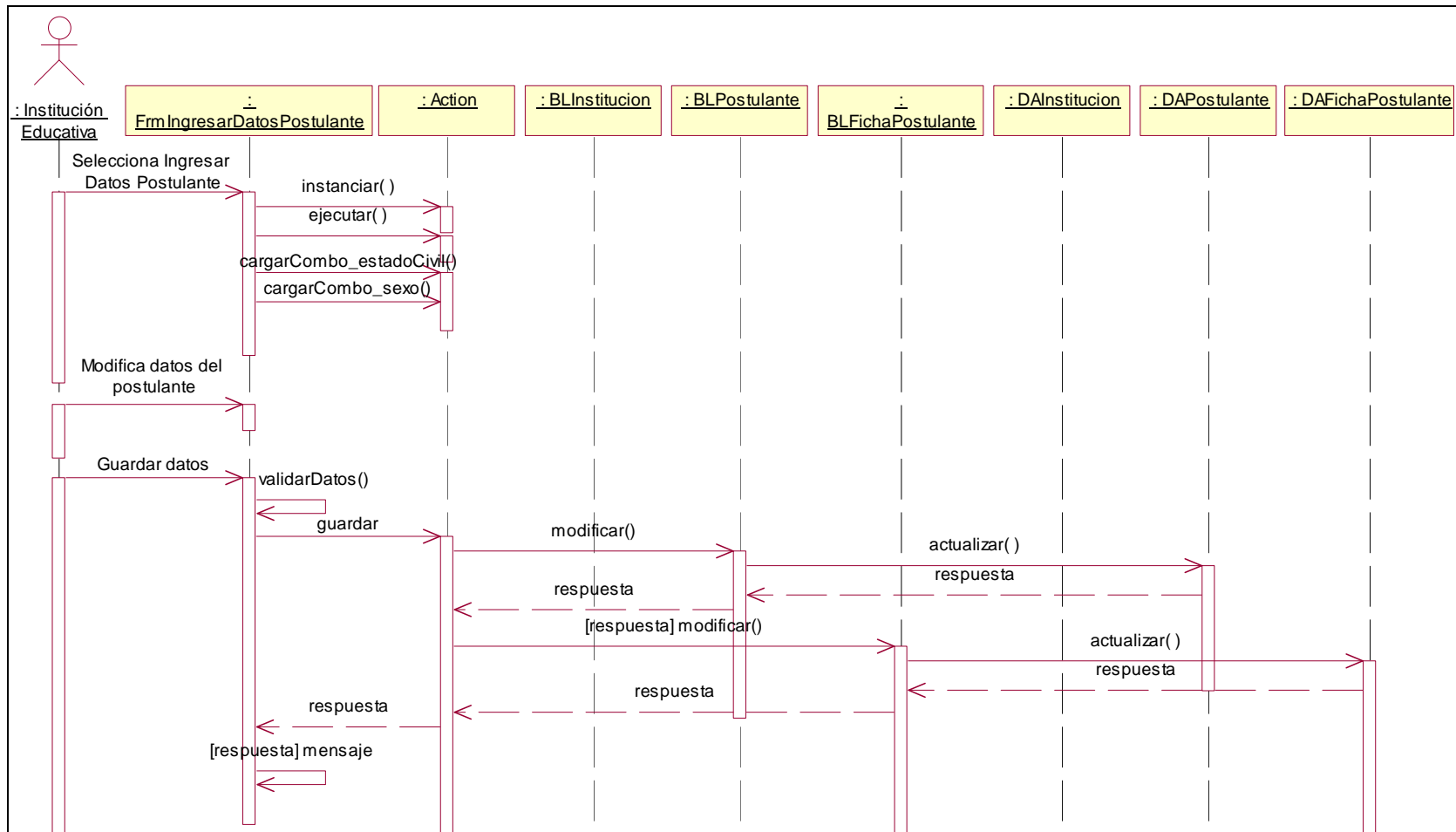


Figura 3.16. Diagrama de secuencias – Proceso de Postulación: Ingresar datos - Postulante.

### Modificar Datos del Postulante – Institución

En la figura 3.17 se ilustra la secuencia de eventos que cada Institución Educativa seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la modificación de los datos ingresados por los postulantes, de modo que se ingrese información verdadera de la situación real de cada postulante.





**Figura 3.17.** Diagrama de secuencias – Proceso de Postulación: Modificar datos - Institución.

### Seleccionar Entrevistados

En la figura 3.18 se ilustra la secuencia de eventos que la Secretaria Ejecutiva seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la selección de los postulantes “para entrevista”.

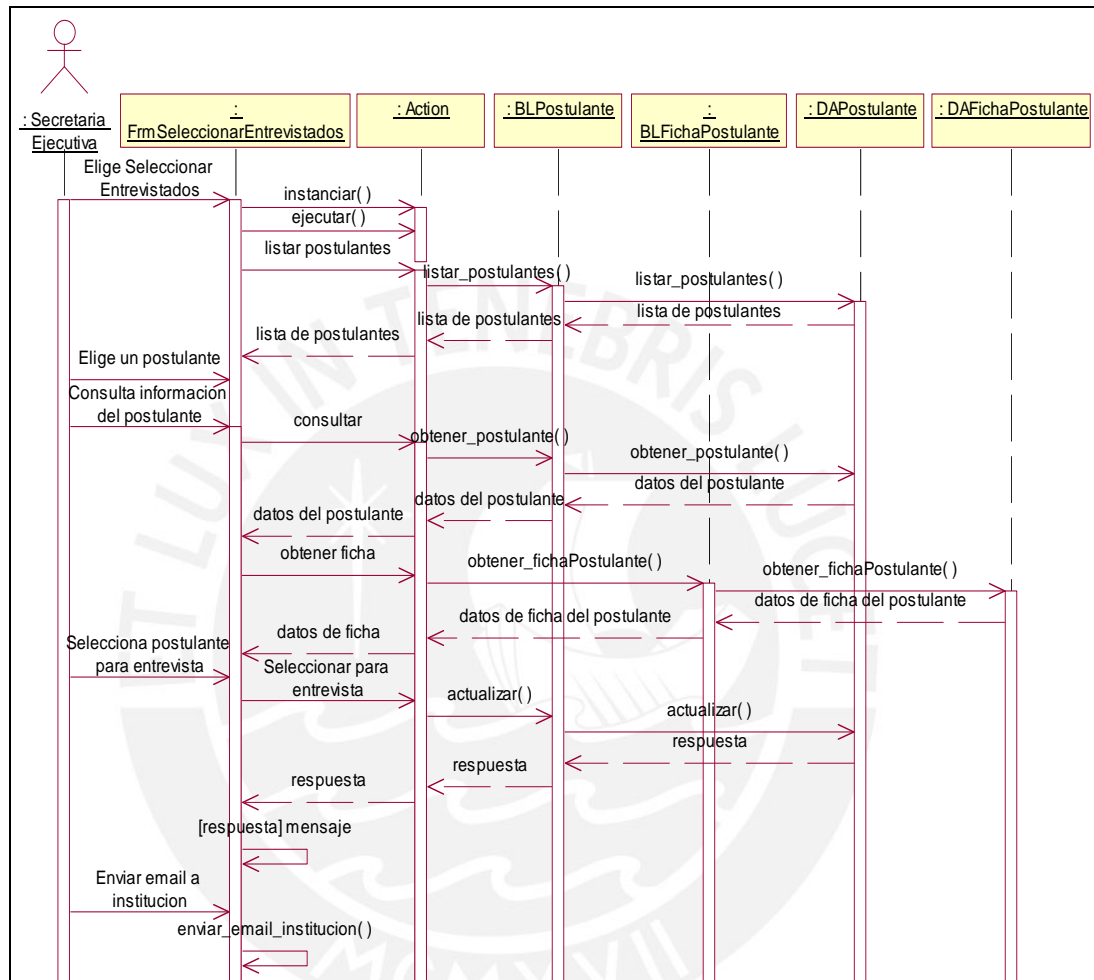


Figura 3.18. Diagrama de secuencias – Proceso de Postulación: Seleccionar entrevistados.

### Asignar Entrevista

En la figura 3.19 se ilustra la secuencia de eventos que cada Institución Educativa seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la asignación de las entrevistas a todos los postulantes que han sido seleccionados por la Secretaria Ejecutiva anteriormente.

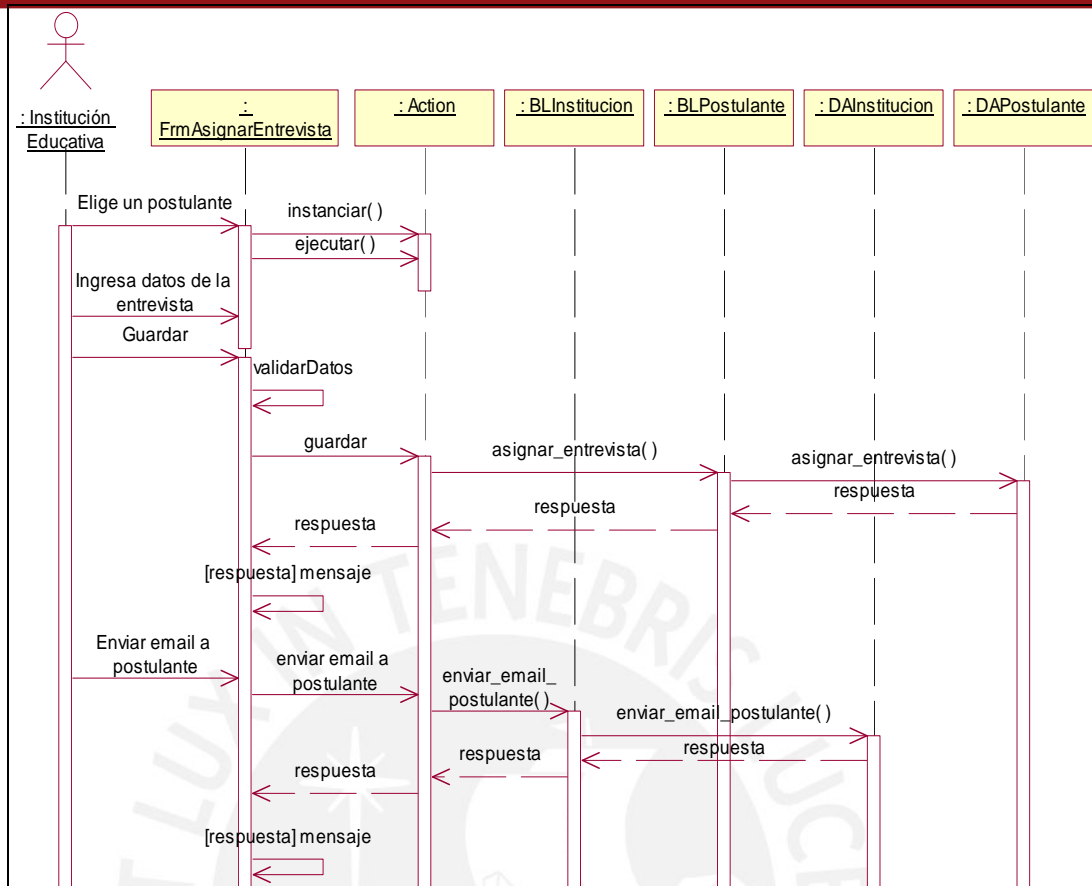


Figura 3.19. Diagrama de secuencias – Proceso de Postulación: Asignar entrevista.

### Consultar Datos de la Entrevista

En la figura 3.20 se ilustra la secuencia de eventos que cada Postulante seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la consulta de los datos de la entrevista asignada.

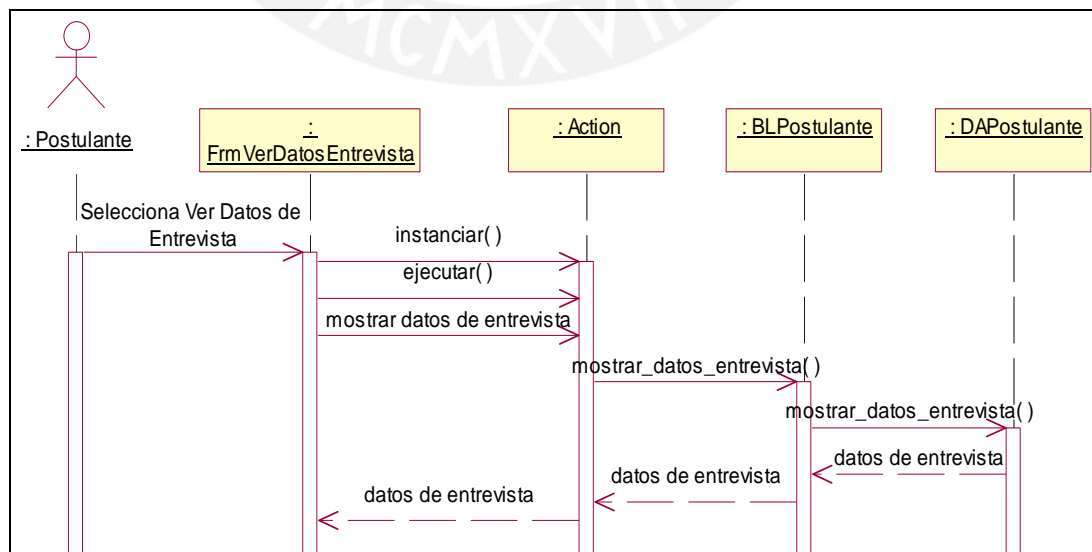


Figura 3.20. Diagrama de secuencias – Proceso de Postulación: Consultar entrevista.

### Seleccionar postulante – Secretaria ejecutiva

En la figura 3.21 se ilustra la secuencia de eventos que la Secretaria Ejecutiva de la asociación seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la selección de los postulantes que podrían ser admitidos por el directorio suizo como nuevos becarios.

Se considera que serán seleccionados aquellos postulantes que pasaron por la etapa de entrevistas satisfactoriamente y que, a criterio de la secretaria ejecutiva de la asociación, realmente requieren una beca o préstamo para culminar sus estudios.

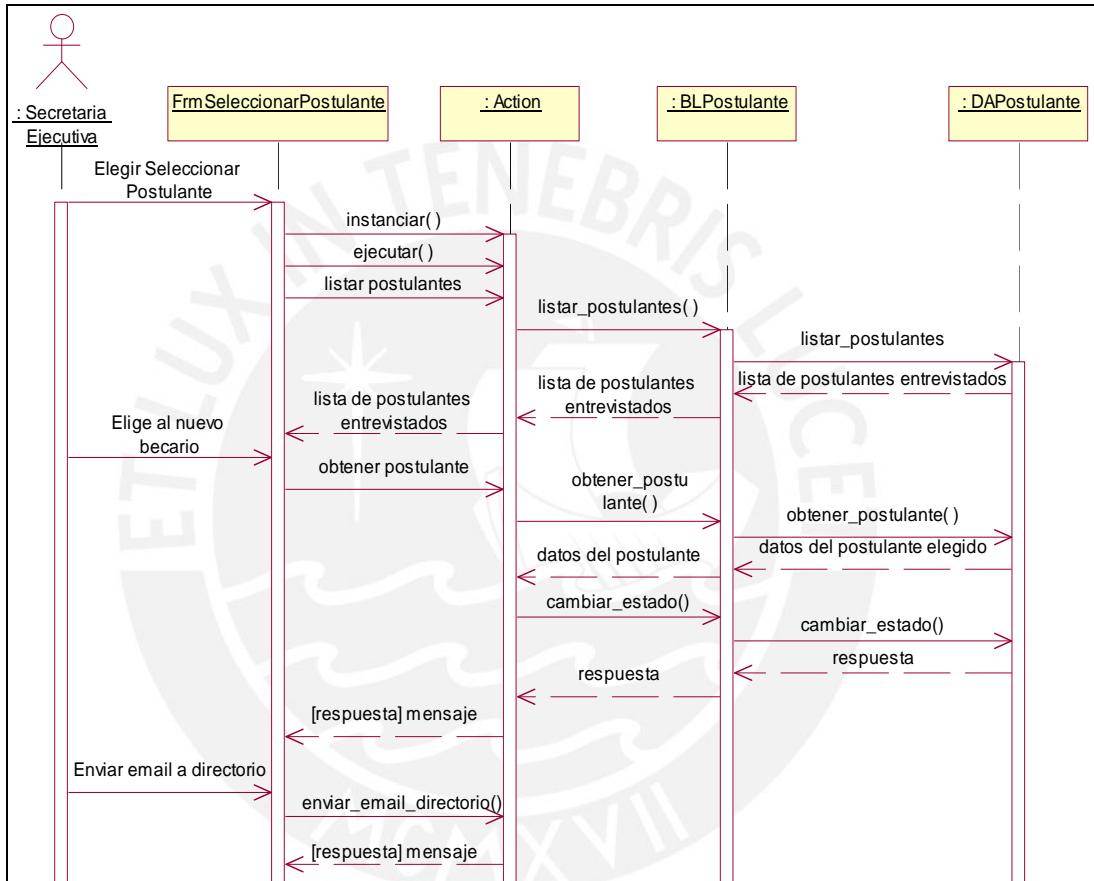


Figura 3.21. Diagrama de secuencias – Proceso de Postulación: Seleccionar postulante.

### Admitir a postulante

En la figura 3.22 se ilustra la secuencia de eventos que el Directorio Suizo seguirá en el sistema propuesto, así como la interacción interna de las clases de diseño implicadas, para llevar a cabo la admisión de postulantes como nuevos becarios.

Se considera que serán admitidos aquellos postulantes que fueron previamente seleccionados por la Secretaria Ejecutiva de la asociación.

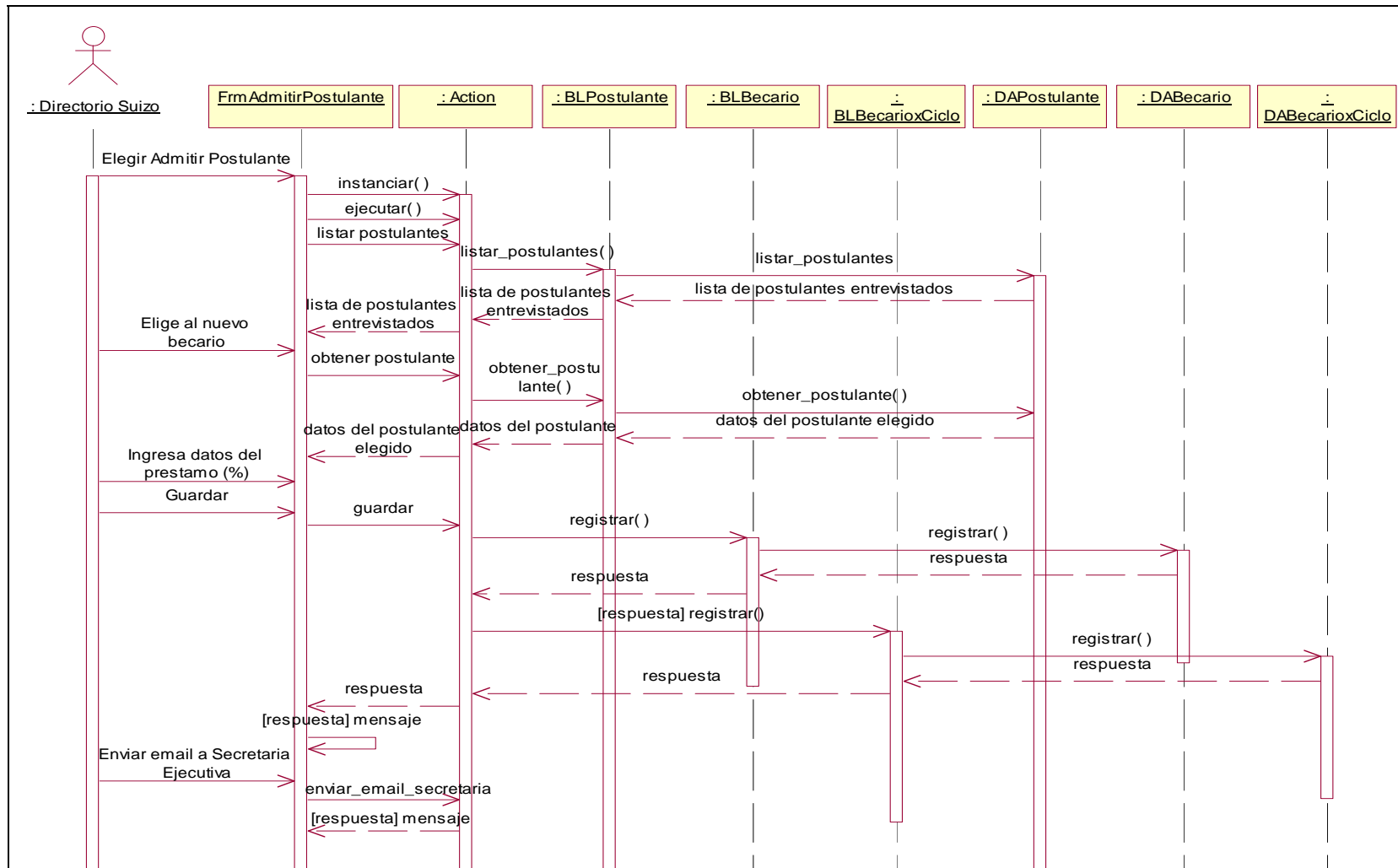


Figura 3.22. Diagrama de secuencias – Proceso de Postulación: Admitir postulante.



## 4. Construcción

En el presente capítulo se detallarán las tecnologías a utilizar para la construcción de la solución propuesta, haciendo una comparación de las ventajas y beneficios ofrecidos por cada una de ellas. Adicionalmente, se describirá la estrategia de pruebas a utilizar que permitirán obtener una mejor calidad del producto final.

### 4.1. Construcción

Para la construcción de la arquitectura propuesta se utilizarán los elementos tecnológicos, marcos de trabajo (*frameworks*) y patrones de diseño, descritos a continuación. Adicionalmente, se ha establecido un estándar de programación. En el Anexo K: Estándares de Programación, se puede encontrar al mencionado estándar seguido por las tesis durante el proceso de programación.

#### 4.1.1. Tecnologías

##### *Lenguaje de programación*

Debido a que se requiere desarrollar una solución que no implique costos por adquisición de licencias y sea fácil de manejar, se ha optado por utilizar un lenguaje de programación orientado a objetos y que, adicionalmente, sea compatible con otras tecnologías de software libre.

Entre los lenguajes analizados tenemos:

✓ **Ruby**

**Descripción:** Es un lenguaje interpretado de alto nivel y orientado a objetos. Es una mezcla de los lenguajes Perl, Smalltalk, Eiffel, Ada y Lisp. [COR 2008, DPV 2007]

**Características:**

- Diseñado para una programación rápida y sencilla.
- Manejo de excepciones para facilitar el manejo de errores.
- Posibilidad de hacer llamadas directamente al sistema operativo.
- Utilización de un recolector de basura para todos los objetos de Ruby.
- Incluye llamadas para embeber Ruby en otros programas, y así usarlo como lenguaje de scripting.
- No requiere de declaración de variables.
- Manejo de hilos (*threading*) independiente del sistema operativo.
- Puede cargar bibliotecas de extensión dinámicamente si lo permite el sistema operativo.
- Uso de métodos Singleton.
- Maneja su propio marco de trabajo para desarrollo de aplicaciones web: Ruby On Rails, el cual es software libre por naturaleza y está basado en el patrón de diseño Modelo Vista Controlador (MVC).
- Es multiplataforma.

✓ **PHP**

**Descripción:** Es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. [TPG 2008, DPV 2007]

**Características:**

- Su código no necesita ser compilado para ejecutarse.
- Para su funcionamiento necesita tener instalado el servidor Apache o IIS con las librerías de PHP.
- La mayor parte de su sintaxis ha sido tomada de los lenguajes C, Java y Perl con algunas características específicas.
- No requiere de declaración de variables.
- Los archivos cuentan con la extensión php.
- Soporta en cierta medida las clases y la herencia.
- Es multiplataforma.
- Por su diseño dinámico, no puede ser compilado y es difícil de optimizar.
- Favorece la creación de código desordenado y complejo de mantener.

✓ **Java**

**Descripción:** Es un lenguaje de programación orientado a objetos y que fue desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de la sintaxis de C y C++, pero tiene un modelo de objetos más simple, además elimina herramientas de bajo nivel que suelen inducir a muchos errores como la manipulación directa de punteros o memoria. [GMC 1996, TJT 2008]

**Características:**

- Las aplicaciones Java están típicamente compiladas en un bytecode. En tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución.

- Utilización de un recolector automático de basura.
- Brinda la posibilidad de desarrollar applets que luego pueden ser incrustados en páginas HTML.
- Permite el manejo de Servlets y JSP en el lado del servidor, para el desarrollo de aplicaciones web dinámicas.
- Posibilita el uso de extensiones como JDBC y JavaMail.
- Permite el manejo de marcos de trabajo como Struts, Webwork y Spring.
- Es multiplataforma.

✓ **Python**

**Descripción:** Es un lenguaje de programación script, interpretado, interactivo y orientado a objetos. Se le compara con lenguajes como Tcl, Perl, Scheme o Java. [PCO 2008, DPV 2007]

**Características:**

- Sintaxis muy sencilla y limpia pero con gran potencia.
- Permite la creación de todo tipo de programas incluyendo los sitios web.
- Su código no necesita ser compilado, por lo que su código es llamado: interpretado.
- Es multiparadigma, lo cual fuerza a los programadores adoptar un estilo de programación particular (orientada a objetos, estructurada, funcional u orientada a aspectos).
- Es multiplataforma.

En la tabla 4.1 se muestra una comparación que revela las ventajas y desventajas de dichos lenguajes de programación, según ciertas características comunes, el nivel de aprendizaje requerido, el nivel de experiencia del equipo, entre otras cosas.

Característica	Ruby	PHP	Java	Python
Multiplataforma	Sí.	Sí.	Sí.	Sí.
Nivel de aprendizaje	Fácil.	Fácil - Intermedio.	Intermedio.	Fácil.
Software libre	Sí.	Sí.	Sí.	Sí.
Documentación existente	Regular.	Regular.	Bastante.	Bastante.
Nivel de experiencia	-	Básico.	Intermedio.	-
Organización por capas	Muy difícil.	Difícil.	Fácil.	Fácil.
Extensiones	Sí.	Sí.	Sí.	Sí.
Rendimiento	Los scripts podrían resultar lentos cuando se trata con bastantes datos.	Muy deficiente para aplicaciones grandes que demanden gran cantidad de solicitudes del servidor.	Penalización en tiempo debido a características propias del lenguaje.	Lentitud por ser un lenguaje interpretado.

**Tabla 4.1.** Comparación de lenguajes de programación.

De los lenguajes de programación presentados, y a pesar que se requiere mayor aprendizaje, se ha optado por utilizar **Java** como lenguaje de programación para el desarrollo de la solución. Esto se justifica primordialmente en el nivel de experiencia que tiene el equipo de desarrollo, la documentación existente que se puede encontrar y la facilidad que ofrece Java para el desarrollo de aplicaciones en capas, tal como se determinó en el capítulo anterior.

### **Entorno de desarrollo**

Considerando el lenguaje de programación definido (Java), se planteó la posibilidad del uso de tres entornos de desarrollo de software libre bastante reconocidos: NetBeans, Eclipse y JDeveloper, cuyas descripciones se muestran a continuación.

#### ✓ **NetBeans**

NetBeans IDE es un entorno de desarrollo integrado y software libre, que permite que desarrolladores de software puedan escribir, compilar, depurar y ejecutar aplicaciones fácilmente. Está totalmente escrito en lenguaje Java, pudiendo brindar la flexibilidad de utilizar casi cualquier lenguaje de programación como Ruby, Java, C/C++, entre otros. [NBC 2008]

Este entorno es de fácil instalación, según apreciación personal, y trabaja bien en diferentes sistemas operativos, dando la posibilidad de añadirse paquetes adicionales para el desarrollo de aplicaciones específicas. Por ejemplo, se tiene el *Visual Web Pack* [NBV 2006] que permite construir aplicaciones web estándar, incluyendo soporte para la tecnología Ajax y componentes JSF.

La interfaz gráfica que brinda es bastante intuitiva, aunque su apariencia visual no es tan atractiva. Además, este entorno de desarrollo es bastante estable y robusto, incorpora varios componentes para el desarrollo de diferentes tipos de aplicaciones, lo que ocasiona que al ejecutarlo consuma bastantes recursos de CPU. [ABI 2008]

#### ✓ **Eclipse**

Eclipse IDE es un entorno de desarrollo integrado y software libre (bajo la licencia GNU GPL), que permite el desarrollo de aplicaciones en casi cualquier lenguaje de programación, incluyendo Java, C/C++ y Python, así como lenguajes de procesamiento de texto como LaTeX. La idea de su construcción es que sea una especie de herramienta universal de código abierto y que sirva para todo y para nada en particular [OTI 2003].

Este entorno emplea módulos (llamados *plug-in*) para proporcionar toda su funcionalidad según el requerimiento de uso del usuario. Esto lo diferencia de otros entornos de desarrollo, puesto que dichos entornos son monolíticos ya que incluyen todas las funcionalidades, así las necesite el usuario o no. Por esta característica, Eclipse IDE es bastante utilizado ya que no consume muchos recursos de CPU [TEF 2008]. Sin embargo, una desventaja de dicha característica, según apreciación personal, es que la instalación de este entorno es un tanto complicada puesto que el usuario debe conocer previamente cuáles son las funcionalidades que necesita según los módulos existentes e instalar cada uno por separado, lo cual implica una pérdida de tiempo.

La interfaz gráfica que brinda no es tan intuitiva pero su apariencia visual es bastante atractiva. Además, este entorno es bastante estable. [ABI 2008]

✓ **JDeveloper**

Oracle JDeveloper es un entorno de desarrollo integrado desarrollado por Oracle Corporation, por lo que es un software propietario pero gratuito desde el año 2005.

Permite el desarrollo de aplicaciones orientadas a servicios, bajo lenguajes como Java, HTML, XML, SQL, PL/SQL, Javascript, PHP, Oracle ADF, UML y otros. Las primeras versiones estaban basadas en el entorno JBuilder de Borland, pero desde la versión 9i del 2001 está basado en Java, desligándose del código anterior de JBuilder. [OTN 2008]

Tiene una integración muy buena con marcos de trabajo como Struts, Ant, JUnit, XDoclets y CVS, ofreciendo características y componentes internos para dichos marcos de trabajo. Además, este entorno da la posibilidad de añadir extensiones para que los usuarios agreguen las funcionalidades que requieran. [SSH 2008]

En la tabla 4.2 se muestra una comparación que revela las ventajas y desventajas del uso de los entornos de desarrollo mencionados.

Característica	NetBeans	Eclipse	JDeveloper
Usabilidad	Muy buena.	Buena.	Buena.
Estabilidad	Buena.	Buena.	Buena.
Instalación	Fácil.	Regular.	Fácil.
Apariencia	Buena.	Muy buena.	Buena.
Nivel de experiencia	Intermedio	Básico.	-
Compatibilidad con servidores	Versión 6.5: Incorpora Tomcat y Glassfish.	Plug-in de JBoss, Tomcat y Glassfish	JBoss, Tomcat
Soporte de frameworks	Struts, Spring, Hibernate, iBatis, Tapestry	Plug-in de: Struts, Spring, Hibernate, Tapestry	Struts, Spring, Hibernate, iBatis, Tapestry
Soporte de DBMS	MySQL, Oracle, DB2, PostgreSQL, SQL Server	MySQL, Oracle, DB2, PostgreSQL, SQL Server	MySQL, Oracle, PostgreSQL, SQL Server
Consumo de recursos	Mucho.	Poco.	Poco.

**Tabla 4.2.** Comparación de entornos de desarrollo.

Considerando las definiciones y el cuadro comparativo, se ha optado por utilizar **NetBeans versión 6.5** ya que ofrece buena estabilidad, usabilidad y primordialmente porque incorpora la instalación de Tomcat como servidor web y Glassfish como servidor de aplicaciones, lo cual ofrece una ventaja en cuanto a menor costo y facilidad de instalación de los componentes requeridos para la arquitectura definida.



## **Servidor de aplicaciones**

Considerando la arquitectura de la solución, se propuso el uso de un servidor web o un servidor de aplicaciones, ambos con las características de ser software libre y multiplataforma.

Dado que los conceptos son distintos, se definieron las ventajas y desventajas del uso de ambos.

### ✓ **Servidor web**

Programa que implementa el protocolo HTTP (diseñado para transferir hipertextos, páginas web o páginas HTML). Dicho programa se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que responde a estas peticiones adecuadamente mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Instalar un servidor web propio permitiría, entre otras cosas, poder montar la propia página web sin necesidad de contratar *hosting* (servicio de alojamiento web). La desventaja de ello es que se requiere tener el ordenador encendido permanentemente para que la aplicación sea accesible de forma continua, con el consiguiente costo debido al consumo de electricidad.

Se recomienda su uso en casos en los cuales sólo se requiera publicar una aplicación web, sin considerar el alojamiento de otro tipo de aplicaciones. [DRW 2007]

A continuación se muestran algunos servidores web de software libre, con sus respectivas descripciones.

#### ▪ **Apache Tomcat o Jakarta Tomcat**

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems. Dado que fue escrito en lenguaje Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. [AST 2008, JHA 2008]

#### ▪ **Servidor HTTP Cherokee**

Se trata de un servidor web libre, multiplataforma, abierto bajo la licencia GPL. Ha sido desarrollado completamente bajo lenguaje C, con el objetivo de ser un servidor web bastante rápido y escalable. Entre sus características se puede mencionar que puede ejecutar CGI, PHP tanto como PHPCGI o FastCGI. También soporta registro y autenticación de usuarios. [ALO 2008]

### ✓ **Servidor de aplicaciones**

Programa que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general a través de Internet y utilizando el protocolo HTTP. Se distingue de un servidor web por el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos. Adicionalmente, proporciona servicios de

*middleware*, es decir, se encarga de la seguridad y el mantenimiento de la información. [DRA 2007]

El servidor de aplicaciones ejecuta los programas de negocio en lugar del cliente (navegador), del servidor web o sistemas finales. Físicamente separa la lógica del negocio del cliente y los datos dentro de una arquitectura multicapa. Por ello, permite desarrollar y desplegar aplicaciones rápida y fácilmente e incrementar la cantidad de usuarios sin perder la calidad de servicio y sin hacer cambios en la programación.

Se recomienda su uso en casos en los cuales no se requiera simplemente publicar una aplicación web, sino también alojar otras aplicaciones como por ejemplo una base de datos [DRA 2007]. Adicionalmente, se recomienda su uso cuando se necesite que la aplicación realice transferencia de información valiosa, como por ejemplo información de cuentas bancarias o transacciones de dinero; ya que este tipo de servidores utiliza protocolos criptográficos (como SSL y TLS) que proporcionan comunicaciones seguras en Internet.

A continuación se muestran algunos servidores de aplicaciones de software libre, con sus respectivas descripciones.

- **GlassFish**

Servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen dicha especificación. Es gratuito y de código libre, se distribuye bajo la licencia CDDL y la GNU GPL. Tiene como base al servidor Sun Java System Application Server de Sun Microsystems. Algunas ventajas para su uso son: existe mucha documentación al respecto, tiene una consola de administración muy amigable, tiene una integración total con la última versión de NetBeans (v6.5) e incluye gran parte de Tomcat en el Web Container, con especial orientación en los temas de rendimiento. [GFC 2008, JCI 2006]

- **JBoss**

Es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en lenguaje Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. Entre sus características se destaca que tiene una flexibilidad consistente, ofrece servicios de *middleware* para cualquier objeto de Java y está orientado a una arquitectura de servicios (SOA). [JBC 2008, JCA 2007]

Considerando el entorno de desarrollo elegido (NetBeans v6.5) y los conceptos mostrados, en un primer momento podría optarse por utilizar Apache Tomcat como servidor web, ya que ofrece mayores prestaciones que Cherokee, o tal vez GlassFish como servidor de aplicaciones, ya que ofrece mayor robustez que JBoss. Sin embargo, se deben considerar ciertos puntos para la elección de alguno de ellos:

- La aplicación a desarrollar será accesada por distintos usuarios, pero se plantea que no existirán demasiadas conexiones al mismo tiempo.

- La aplicación debe considerar métodos de seguridad debido a que accederán diferentes tipos de usuarios, entonces diferentes permisos. La seguridad debe resguardar la información de la organización.
- La aplicación mantendrá información importante, como notas, estados de las deudas de cada alumno y cuentas bancarias, por lo que se hace indispensable que se provea de autenticación y privacidad de la información.
- Se requiere alojar tanto la aplicación web como la base de datos de la organización.

En la tabla 4.3 se muestra una comparación que revela las ventajas y desventajas del uso de los servidores.

Característica	Apache Tomcat	Glassfish
Implementación de tecnologías Java EE	Sí.	Sí.
Robustez	Buena.	Muy buena.
Orientación a la seguridad	-	Muy buena.
Orientación al rendimiento	-	Muy buena.
Alojamiento de la base de datos	Bueno.	Muy Bueno.
Integración con NetBeans v6.5	Sí.	Sí.

**Tabla 4.3.** Comparación de servidores.

Por todo ello, la mejor opción es utilizar **GlassFish** como servidor de aplicaciones ya que, como se mencionó anteriormente, ofrece robustez, seguridad de la información y permite alojar la base de datos sin inconvenientes.

### **Gestor de base de Datos**

Considerando la cantidad de conexiones concurrentes y el tamaño de la aplicación, se propuso usar un gestor de base de datos relacional que brinde un buen rendimiento, que sea software libre y que se integre muy bien con las tecnologías antes mencionadas. Por ello, se han analizado dos conocidos gestores de base de datos, cuyas descripciones, ventajas y desventajas se muestran a continuación.

#### ✓ **PostgreSQL**

Se trata de un manejador de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Es software libre bajo licencia BSD, utiliza el lenguaje SQL92/SQL99 y posee una gran escalabilidad. [PGD 2008]

Se considera como un gran gestor de bases de datos, capaz de competir con muchos gestores comerciales, aunque carece de un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que administre.

Por otro lado, la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede resultar un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes. En conclusión, la velocidad de respuesta resulta favorable para bases de datos grandes y no pequeñas.

Se recomienda su uso en aplicaciones que tengan una gran cantidad de accesos concurrentes a la base de datos, como por ejemplo un sitio web que posea alrededor de 500.000 peticiones por día. [DPE 2007]

### ✓ MySQL

Se trata de un sistema de gestión de bases de datos relacional, bajo la licencia GPL de GNU. Fue desarrollado con la idea de brindar mayor velocidad de respuesta, aunque se sacrificaron algunas características esenciales. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. [MSA 2008]

Este gestor de bases de datos se puede considerar como el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esto puede deberse adicionalmente a que existe gran cantidad de librerías y herramientas que permiten su uso a través de diferentes lenguajes de programación, además de su fácil instalación y configuración.

Se recomienda su uso en aplicaciones que no tengan una gran cantidad de accesos concurrentes a la base de datos, puesto que el rendimiento puede bajar considerablemente. [DPE 2007]

En la tabla 4.4 se muestra una comparación que revela las ventajas y desventajas de ambas tecnologías.

Característica	PostgreSQL	MySQL
Usabilidad	Buena.	Muy buena.
Robustez	Muy buena.	Regular.
Instalación y configuración	Regular.	Fácil.
Apariencia	Buena.	Muy buena.
Nivel de experiencia	Básico.	Intermedio.
Velocidad de respuesta	Buena.	Muy Buena.
Control de acceso y autenticación de usuarios	Difícil.	Fácil.
Soporte de triggers y procedimientos almacenados	Mejor.	Deficiente.
Consumo de recursos	Mucho.	Poco.

**Tabla 4.4.** Comparación de gestores de base de datos.

Por todo lo mostrado en las definiciones y el cuadro comparativo, se ha optado por utilizar **MySQL** ya que, aunque no ofrece robustez como PostgreSQL, brindaría el rendimiento adecuado para las pocas conexiones concurrentes que se plantea tendrá la aplicación web.

### Lenguaje para creación de páginas Web

Considerando el uso del lenguaje Java, se ha optado por utilizar la tecnología JSP para la creación y manejo de las páginas web. A continuación se muestra una breve descripción.

#### ✓ JSP

JSP es un acrónimo de *Java Server Pages*, que en castellano se puede traducir como “Páginas de Servidor Java”. Es una tecnología orientada a crear páginas web con programación en Java y que permite generar contenido dinámico en forma de documentos HTML, XML o de otro tipo. [SDP 2008]

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del navegador del usuario.

En la figura 4.1 se ilustra el funcionamiento típico de la tecnología JSP.

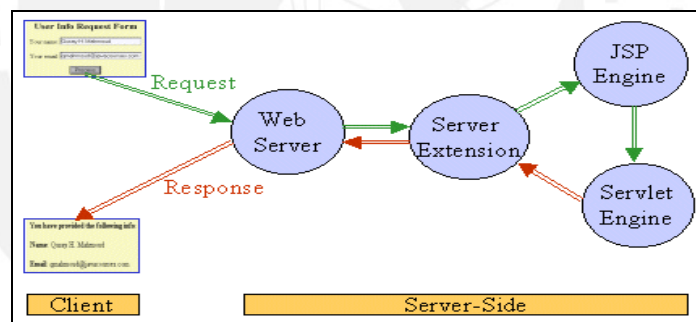


Figura 4.1. Funcionamiento de JSP.

Fuente: [CDG 2008]

#### ✓ ASP

ASP es un acrónimo de *Active Server Pages*, que en castellano se puede traducir como “Páginas de Servidor Activo”. Esta tecnología, similar a JSP, ha sido creada por Microsoft y es recomendada para desarrollo en plataforma .Net con su versión ASP .Net.

Permite generar páginas web con contenido dinámico, utilizando código de scripts del lado del servidor con lenguaje HTML. Además, hace uso de componentes ya desarrollados por su fabricante, como por ejemplo los controles ActiveX.

ASP permite la programación de las páginas web mediante lenguajes compilados como Visual Basic, C# y otros lenguajes soportados por la



plataforma .Net. Adicionalmente, puede utilizarse lenguaje Java con esta tecnología para desarrollo web pero no es recomendable. [MCO 2008]

En la tabla 4.5 se muestra una comparación que revela las ventajas y desventajas del uso de los lenguajes para creación de páginas web.

Característica	JSP	ASP
Soporte de lenguaje Java	Sí.	Sí, pero no recomendable.
Generación de contenido dinámico	Sí.	Sí.
Integración con NetBeans v6.5	Sí.	No.
Integración con componentes adicionales	Muy buena.	Sólo con componentes creados por su fabricante (Microsoft).
Portabilidad	Varios sistemas operativos.	Solo sistemas operativos de Microsoft.
Usabilidad	Fácil.	Regular.

**Tabla 4.5.** Comparación de lenguajes para creación de páginas web.

Por todo lo anteriormente expuesto, **JSP** ofrece mayores ventajas puesto que la parte dinámica se realiza con Java, lo cual ofrece facilidad de uso, y es portable a otros sistemas operativos y servidores web que soporten una máquina virtual de Java.

#### 4.1.2. Frameworks

##### *Framework de presentación*

Según la investigación de documentación existente, se recomienda que, para la capa de presentación de la arquitectura definida, la mejor práctica es elegir un marco de trabajo existente y probado en vez de diseñar y construir un marco de trabajo personalizado [DYS 2007]. Por ese motivo, a continuación se muestran las descripciones de varios marcos de trabajo de este tipo entre los que se puede elegir: Struts, WebWork, y JSF.

##### ✓ JSF

JSF es el acrónimo de Java Server Faces. Se trata de un marco de trabajo de interfaces de usuario del lado de servidor para aplicaciones web basadas en lenguaje Java, que facilita la creación y mantenimiento de páginas web bajo la tecnología JSP, aunque también se acomoda a otras tecnologías como XUL.

Este marco de trabajo es software libre y sus principales componentes son [SDF 2008]:

- Un conjunto de interfaces y una implementación de referencia para: representar componentes de interfaz de usuario y manejar su estado; manejo de eventos, validación del lado del servidor y conversión de

datos; definir la navegación entre páginas; soportar internacionalización y accesibilidad; y proporcionar extensibilidad para todas estas características.

- Una librería de etiquetas Java Server Pages (JSP) personalizadas para dibujar componentes de interfaz de usuario dentro de una página JSP.

Todo ello facilita significativamente la tarea de construcción y mantenimiento de aplicaciones web con interfaces de usuario del lado del servidor. Adicionalmente, permite la creación de validadores personalizados y la definición de navegabilidad entre páginas, entre otras características.

#### ✓ **Struts**

Se trata de un marco de trabajo de soporte para el desarrollo de aplicaciones web utilizando el patrón MVC bajo la plataforma J2EE (Java 2 Enterprise Edition). [ASS 2008]

Entre sus principales características se puede mencionar:

- Configuración del control centralizada.
- Interrelaciones entre acciones y página u otras acciones se especifican por tablas XML en lugar de codificarlas en los programas o páginas.
- Uso de componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
- Manejo de librerías de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas JSP.
- Uso de herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página (en JavaScript) hasta las validaciones de fondo hechas a nivel de las acciones.

Struts permite reducir el tiempo de desarrollo. Su carácter de software libre y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible, lo convierte en una herramienta altamente disponible. Sin embargo, actualmente no es muy utilizado puesto que otros marcos de trabajo pueden ofrecer mayores prestaciones [RBA 2004].

#### ✓ **WebWork**

Se trata de un marco de trabajo de software libre que ha sido diseñado para ser conceptualmente simple, interoperable y sencillo de usar, proporcionando un soporte robusto para construir interfaces de usuario reusables [OSY 2007].

Entre sus principales características se puede mencionar [SCI 2007]:

- Se centra en la capa de presentación y negocio.
- Por un lado permite asociar páginas con clases-acción, pero también proporciona una serie de componentes reusables que pueden ser invocados desde cualquier página.
- Puede configurarse para usar en la capa de presentación tecnologías como FreeMarker o JasperReports.
- Proporciona un sistema de validación basado en XWork y dispone de una serie de validadores predefinidos y permite agregar validadores creados por el desarrollador.

- En el archivo de configuración principal se define la navegabilidad entre páginas, donde para cada acción existe una vista de entrada y una vista de salida a la que redirigir en caso de ejecutarse con éxito.

Este marco de trabajo intenta reducir al mínimo la cantidad de código necesario para trabajar, permitiendo a los desarrolladores centrarse en la lógica de negocio y el modelado. En su versión 2.2, presentada en el año 2006, WebWork se fusionó con la comunidad Struts dando lugar a Struts Action Framework 2.0. Este nuevo marco de trabajo mantiene la filosofía de WebWork y se aprovecha de la gran comunidad de usuarios de Struts, además de estar bajo el nombre de la Fundación Apache.

Una de sus grandes ventajas es que es simple y flexible de utilizar. Además, permite trabajar con las tecnologías JSP y Velocity fácilmente. [ESW 2003]

En la tabla 4.6 se muestra una comparación que resume las ventajas y desventajas de utilizar ambas tecnologías.

Característica	JSF	Struts	WebWork
Usabilidad	Muy buena.	Buena.	Muy buena.
Apariencia	Buena.	Regular.	Buena.
Nivel de experiencia	-	Básico.	-
Compatibilidad con JSP	Muy buena.	Buena.	Buena.
Navegabilidad entre páginas	Fácil.	Regular.	Fácil.
Nivel de aprendizaje	Fácil.	Regular.	Fácil.
Documentación existente	Bastante.	Bastante.	Poca.

**Tabla 4.6.** Comparación de frameworks de capa de presentación.

Por todo ello, se ha optado por utilizar el marco de trabajo **Struts** en su versión *Struts Action Framework 2.0* puesto que ofrece buena usabilidad, estabilidad, integración con JSP y existe bastante documentación al respecto.

### **Framework de lógica de negocio**

Según la investigación de documentación existente, para la construcción de la capa de lógica de negocio se puede utilizar EJB (Enterprise Java Beans) o POJO (Plain Old Java Objects). Dichos conceptos se muestran a continuación:

✓ **EJB**

Es un acrónimo de Enterprise Java Beans. Es una interfaz que forma parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJBs:

- Comunicación remota utilizando CORBA.
- Transacciones.
- Control de la concurrencia.
- Eventos utilizando JMS (Java messaging service).
- Servicios de nombres y de directorio.
- Seguridad.
- Ubicación de componentes en un servidor de aplicaciones.

Los EJBs proporcionan un modelo de componentes distribuido estándar del lado del servidor, cuyo objetivo es permitir al programador abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, entre otros) para centrarse en el desarrollo de la lógica de negocio. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. [MBL 2004].

✓ **POJO**

Es un acrónimo de Plain Old Java Object, sigla creada por Martin Fowler, Rebecca Parsons y Josh MacKenzie en septiembre del año 2000 y utilizada por programadores lenguaje de programación Java para enfatizar el uso de clases simples y que no dependen de un marco de trabajo en especial.

Surgió como una reacción en el mundo Java a los marcos de trabajo cada vez más complejos, y que requieren un complicado armazón que esconde el problema que realmente se está modelando. En particular surgió en oposición al modelo planteado por los estándares EJB anteriores a la versión 3.0, con el objetivo de lograr una revalorización de la programación simplemente orientada a objetos. [RIC 2006].

Considerando las descripciones mostradas, para la aplicación a desarrollar bastaría utilizar POJO puesto que no se requiere que sea un sistema distribuido.

**4.1.3. Patrones de diseño**

Dado que existe gran cantidad de patrones de diseño [CAJ 2007], a continuación se describirán y compararán dos de los patrones de diseño más conocidos.

**Modelo – Vista – Controlador (MVC)**

MVC es el patrón de diseño arquitectural recomendado para aplicaciones interactivas Java. Entre sus características se tiene que separa los conceptos de diseño, y por lo tanto decreta la duplicación de código, la centralización del control y hace que la aplicación sea más extensible. [SDM 2002, JGU 2006]

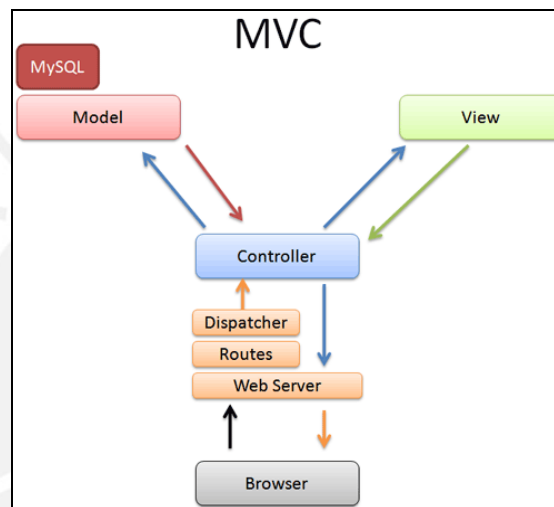
Los componentes en los que se separan los datos de una aplicación, la interfaz de usuario y la lógica de control son los siguientes:

**Modelo:** Encapsula los datos y las funcionalidades del negocio. Accede a la capa de acceso a datos.

**Vista:** Se encarga de mostrar la información recibida al usuario por medio de la interfaz gráfica de usuario. Así como también notifica al Controlador que se ha producido algún evento.

**Controlador:** Recibe los eventos de entrada y se encarga de realizar peticiones de actualización al modelo o a la vista según sea el caso.

En la figura 4.2 se muestra la interacción de dichos componentes.



**Figura 4.2.** Componentes del patrón MVC.  
Fuente: [KAZ 2007]

### **Modelo – Vista – Presentador (MVP)**

MVP es un patrón de diseño arquitectural derivado del patrón MVC. Está más enfocado a que la interfaz de usuario tenga la menor cantidad de código posible y cuya lógica sea administrada por una clase “Presentador” que no dependa de los componentes de la interfaz gráfica y, así, sea más fácil de probar. [TSN 2007]

Sus componentes son los siguientes:

**Modelo:** Compuesto por los objetos que manejan los datos dentro de la aplicación.

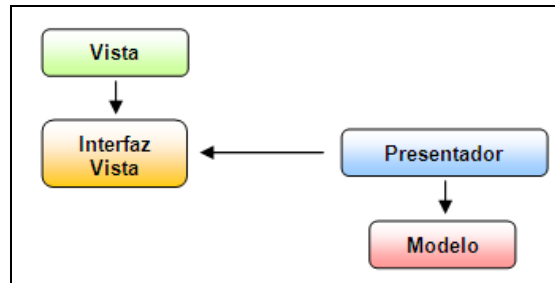
**Interfaz de vista (IView):** Interfaz con la que el presentador se comunica con la vista.

**Vista:** Implementa la interfaz IView y se encarga de manejar los aspectos visuales. Mantiene una referencia a su presentador, al cual le delega la responsabilidad del manejo de los eventos.



**Presentador:** Contiene la lógica para responder a los eventos y manipula el estado de la vista mediante una referencia a la interfaz IView. Además, utiliza el modelo para saber cómo responder a los eventos.

En la figura 4.3 se muestra la interacción de dichos componentes.



**Figura 4.3.** Componentes del patrón MVP.

En la tabla 4.7 se muestra una comparación que revela las principales ventajas y desventajas del uso de ambos patrones de diseño.

Característica	MVC	MVP
Clara separación de conceptos	Sí.	Sí.
Reutilización de código	Sí.	Sí.
Flexibilidad	Sí.	Sí.
Facilidad de probar	Regular.	Fácil.
Facilidad de manejo de múltiples vistas	Fácil.	Regular.
Nivel de aprendizaje	Fácil.	Regular.
Nivel de conocimientos previos	Regular.	-
Documentación existente	Bastante.	Poca.

**Tabla 4.7.** Comparación de patrones de diseño.

Por todo ello, se ha optado por utilizar el patrón MVC ya que, aunque no ofrece tanta facilidad de realizar pruebas como el patrón MVP, la curva de aprendizaje y el nivel de conocimientos previos son favorables para los miembros del equipo de desarrollo.

## 4.2. Pruebas

En esta sección se presenta la estrategia de pruebas a utilizar, los tipos de prueba a realizar, el catálogo de pruebas de los procesos principales y el reporte de la ejecución de dichas pruebas.

### 4.2.1. Estrategia de pruebas

La estrategia a utilizar aplicará un esquema de pruebas incrementales, el cual aborda las pruebas a partir de los componentes atómicos o componentes base hacia los componentes más complejos de acuerdo a la planificación efectuada para el desarrollo del presente proyecto de fin de carrera.

### 4.2.2. Tipos de prueba

Los tipos de prueba que se realizarán son los siguientes:

- **Pruebas Unitarias**

Las pruebas unitarias centrarán al proceso de verificación en la menor unidad de diseño dentro de la arquitectura planteada.

Estas pruebas permitirán la determinación de la obligatoriedad de los campos y sus respectivas validaciones en los formularios. Para ello se identificarán clases válidas y no válidas por formulario.

- **Pruebas de Integración**

Las pruebas de integración son una técnica sistemática para construir la arquitectura mientras que, al mismo tiempo, se llevan a cabo pruebas para la detección de errores en la interacción de sus componentes.

Estas pruebas se realizarán en forma ascendente, de tal manera que se realicen a partir de los componentes atómicos, hasta llegar a cubrir a todos los componentes dentro de la arquitectura.

Adicionalmente, se establecerán determinados escenarios para la pruebas de integración, los cuales facilitarán la validación de todo el sistema atravesando la precedencia de manera transversal.

### 4.2.3. Catálogo de pruebas

En esta sección se describirán los principales casos de prueba de la aplicación, los cuales se realizarán para asegurar la calidad del producto final.

Primero se presentarán las pruebas unitarias por caso de prueba, incluyendo las clases de equivalencia; y posteriormente se presentarán las pruebas de integración de los procesos más complejos del sistema.

En el Anexo H: Catálogo de Pruebas Unitarias, se pueden encontrar algunos otros desarrollos de casos de prueba.

✓ **Caso de prueba: Mantener postulante**

Con este caso de prueba se podrá verificar si el sistema realiza con éxito la administración de la información del postulante.

Dado que existen muchas clases de equivalencia, se mostrarán solamente aquellas pruebas unitarias de mayor importancia.

▪ **Clases de Equivalencia:**

En la tabla 4.8 se muestran las clases válidas y clases no válidas para el formulario que permite el mantenimiento de la información del postulante.

Condición de entrada	Clases válidas	Clases no válidas
Código	1. Cadena de caracteres alfanuméricos. 2. Valores entre 1 y 15 caracteres.	3. Cadena vacía. 4. Valores menores a 1 caracter y mayores a 15 caracteres.
Nombres, Apellido Paterno, Apellido Materno	5. Cadena de caracteres alfanuméricos. 6. Valores entre 1 y 20 caracteres.	7. Cadena con valores menores a 1 caracter y mayores a 20 caracteres.
Fecha de Nacimiento	8. Cadena de caracteres alfanuméricos con formato DD/MM/AAAA.	9. Cadena de caracteres sin formato de fecha.
Lugar de Nacimiento	10. Cadena de caracteres alfanuméricos. 11. Valores entre 1 y 40 caracteres.	12. Cadena con valores menores a 1 caracter y mayores a 40 caracteres.
N° DNI o Libreta Militar	13. Cadena de caracteres numéricos. 14. Valores entre 1 y 10 caracteres.	15. Cadena vacía. 16. Valores menores a 1 caracter y mayores a 10 caracteres.
Estado Civil	17. Valores: 'Soltero', 'Casado', 'Viudo', 'Divorciado'.	18. Cadena vacía.
Edad	19. Cadena de caracteres numéricos. 20. Valores entre 1 y 2 caracteres.	21. Cadena de caracteres no numéricos. 22. Valores menores a 1 caracter y mayores a 2 caracteres.
Sexo	23. Valores: 'F' o 'M'.	24. Cadena vacía.
Correo1, Correo2	25. Cadena de caracteres alfanuméricos, con 1 caracter '@' y al menos 1 punto. 26. Valores entre 1 y 35 caracteres.	27. Cadena de caracteres alfanuméricos sin el formato especificado. 28. Valores menores a 1 caracter y mayores a 35 caracteres.
Teléfono1, Teléfono2	29. Cadena de caracteres numéricos. 30. Valores entre 1 y 15 caracteres.	31. Cadena de caracteres no numéricos. 32. Valores menores a 1 caracter y mayores a 15 caracteres.
Dirección1, Dirección2	33. Cadena de caracteres alfanuméricos. 34. Valores entre 1 y 50 caracteres.	35. Cadena con valores menores a 1 caracter y mayores a 50 caracteres.
Login	36. Cadena de caracteres alfanuméricos. 37. Valores entre 1 y 10	38. Cadena con valores menores a 1 caracter y mayores a 10 caracteres

	caracteres.	
Facultad de Estudios	39. Cadena de caracteres alfanuméricos, registrada por las instituciones.	40. Cadena vacía.
Carrera de Estudios	41. Cadena de caracteres alfanuméricos, registrada por las instituciones.	42. Cadena vacía.
Ciclos en institución, Ciclos faltantes para culminación	43. Cadena de caracteres numéricos. 44. Valores entre 1 y 2 caracteres.	45. Cadena de caracteres no numéricos. 46. Valores menores a 1 caracter y mayores a 2 caracteres.
Total créditos faltantes para culminación, Promedio de último ciclo, Ponderado general	47. Cadena de caracteres alfanuméricos, con formato de número decimal.	48. Cadena vacía. 49. Cadena de caracteres alfanuméricos, sin formato de número decimal.
Puesto en la carrera, Puesto en la facultad	50. Cadena de caracteres numéricos. 51. Valores entre 1 y 10 caracteres.	52. Cadena de caracteres no numéricos. 53. Valores menores a 1 caracter y mayores a 10 caracteres.
Monto de boleta por ciclo	54. Cadena de caracteres alfanuméricos, con formato de número decimal.	55. Cadena vacía. 56. Cadena de caracteres alfanuméricos, sin formato de número decimal.
Escala de pago	57. Cadena de caracteres alfanuméricos. 58. Valores entre 1 y 5 caracteres.	59. Cadena vacía. 60. Valores menores a 1 carácter y mayores a 5 caracteres.
% Préstamo solicitado	61. Cadena de caracteres numéricos.	62. Cadena con caracteres no numéricos.
Monto solicitado por ciclo	63. Cadena de caracteres alfanuméricos, con formato de número decimal.	64. Cadena vacía. 65. Cadena de caracteres alfanuméricos, sin formato de número decimal.
Nombre Idioma1, Nivel Idioma1, Nombre Idioma2, Nivel Idioma2	66. Valores: - Nombre: 'Inglés', 'Español', 'Portugués', 'Alemán', 'Japonés', 'Ruso', 'Francés', 'Italiano'. - Nivel: 'Básico', 'Intermedio', 'Avanzado'.	67. Cadena vacía.
Nombre completo del padre, Nombre completo de la madre	68. Cadena de caracteres alfanuméricos. 69. Valores entre 1 y 50 caracteres.	70. Cadena vacía. 71. Valores menores a 1 caracter y mayores a 50 caracteres.
Ocupación del padre, Ocupación de la madre	72. Cadena de caracteres alfanuméricos. 73. Valores entre 1 y 20 caracteres.	74. Cadena vacía. 75. Valores menores a 1 caracter y mayores a 20 caracteres.
Ingresos anuales del padre, Ingresos anuales de la madre	76. Cadena de caracteres alfanuméricos, con formato de número decimal.	77. Cadena vacía. 78. Cadena de caracteres alfanuméricos, sin formato de número decimal.
Personas que viven con el estudiante	79. Cadena de caracteres alfanuméricos.	80. Cadena vacía.

Casa familiar	81. Valores: 'Propia', 'Alquilada'.	82. Cadena vacía.
Costo mensual	83. Cadena de caracteres alfanuméricos, con formato de número decimal.	84. Cadena vacía. 85. Cadena de caracteres alfanuméricos, sin formato de número decimal.
Otras propiedades, Descripción de la situación económica	86. Cadena de caracteres alfanuméricos, con valores entre 1 y 100 caracteres. 87. Cadena vacía.	88. Cadena de caracteres alfanuméricos, con valores mayores a 100 caracteres.
Nivel económico	89. Cadena de caracteres alfanuméricos, registrada por la asociación.	90. Cadena vacía.
Informe de la asistenta social, Informe del decano de la facultad, Conclusiones	91. Cadena de caracteres alfanuméricos, con valores entre 1 y 200 caracteres.	92. Cadena vacía.

**Tabla 4.8.** Clases de equivalencia del caso de prueba Mantener Postulante.

- **Pruebas Unitarias:**

En la tabla 4.9 se muestran las pruebas unitarias para el mantenimiento de la información del postulante, según las clases de equivalencia descritas anteriormente.

Id	Objetivo	Clases asociadas	Precondición	Descripción	Resultados esperados
1	Verificar el correcto registro de la información personal del postulante, haciendo uso de la pantalla 'Datos personales del postulante'.	CV1, CV2, CV5 CV6, CV25, CV26, CV36, CV37.	El usuario es de la categoría <i>Institución</i> .	<p>Se ingresarán los siguientes valores:</p> <ol style="list-style-type: none"> <li>1. Código: 20022255</li> <li>2. Nombres: <i>Marlessi Marilyn</i></li> <li>3. Apellido paterno: <i>Chanca</i></li> <li>4. Apellido materno: <i>De la Cruz</i></li> <li>5. Correo: <i>marlessi.chanca@pucp.edu.pe</i></li> <li>6. Login: <i>a20022255</i></li> </ol> <p>Hacer clic en 'Guardar'.</p>	Se registra correctamente la información personal del postulante.
2	Verificar el correcto registro de la información de la ficha del postulante, haciendo uso de la pantalla 'Ficha del postulante'.	CV39, CV41, CV43, CV44, CV47, CV50, CV51, CV54, CV57, CV58, CV61, CV63, CV66.	El usuario es de la categoría <i>Postulante</i> .	<p>Se ingresarán los siguientes valores:</p> <ol style="list-style-type: none"> <li>1. Facultad de estudios: <i>Ciencias e Ingeniería</i></li> <li>2. Carrera de estudios: <i>Ingeniería Informática</i></li> <li>3. Ciclos en la institución: 7</li> <li>4. Ciclos faltantes: 4</li> <li>5. Total créditos faltantes: 50</li> <li>6. Promedio de último ciclo: 15</li> <li>7. Ponderado general: 14</li> <li>8. Puesto en la carrera: 100 de 1500</li> <li>9. Puesto en la facultad: 200 de 4000</li> <li>10. Monto de boleta por ciclo: 400</li> <li>11. Escala de pago: 2</li> <li>12. % Préstamo solicitado: 100</li> <li>13. Monto solicitado: 200</li> <li>14. Nombre idioma 1: <i>Inglés</i></li> <li>15. Nivel idioma 1: <i>Intermedio</i></li> <li>16. Nombre idioma 2: <i>Portugués</i></li> <li>17. Nivel idioma 2: <i>Básico</i></li> </ol> <p>Hacer clic en 'Guardar'.</p>	Se registra correctamente la información de la ficha del postulante.



Id	Objetivo	Clases asociadas	Precondición	Descripción	Resultados esperados
3	Verificar el correcto registro de la información socioeconómica del postulante, haciendo uso de la pantalla 'Información socio-económica del postulante'.	CV68, CV69, CV72, CV73, CV76, CV79, CV81, CV83, CV86, CV87, CV89.	El usuario es de la categoría <i>Postulante</i> .	<p>Se ingresarán los siguientes valores:</p> <ol style="list-style-type: none"> <li>1. Apellidos y nombres del padre: <i>Chanca Portuguez Juan</i></li> <li>2. Ocupación del padre: <i>Director de escuela</i></li> <li>3. Ingresos anuales del padre: <i>10000</i></li> <li>4. Apellidos y nombres de la madre: <i>De la Cruz Pérez Catalina</i></li> <li>5. Ocupación de la madre: <i>Ama de casa</i></li> <li>6. Ingresos anuales de la madre: <i>0</i></li> <li>7. Casa familiar: <i>Alquilada</i></li> <li>8. Costo mensual: <i>600</i></li> <li>9. Otras propiedades: <i>vacío</i>.</li> <li>10. Descripción de la situación económica: <i>no es muy buena</i>.</li> <li>11. Nivel económico: <i>Low</i></li> <li>12. Personas que viven con el estudiante: <i>Chanca de la Cruz Katherine, Hermana, Estudiante, 200</i></li> </ol> <p>Hacer clic en 'Guardar.'</p>	Se registra correctamente la información socioeconómica del postulante.
4	Verificar el correcto registro de los informes del postulante, haciendo uso de la pantalla 'Informes del postulante'.	CV91.	El usuario es de la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	<p>Se ingresarán los siguientes valores:</p> <ol style="list-style-type: none"> <li>1. Conclusiones: <i>se admitirá al postulante tras haber analizado su situación económica</i>.</li> </ol> <p>Hacer clic en 'Enviar e-mail a representante'.</p>	Se registra correctamente las conclusiones en la sección de informes del postulante.

**Tabla 4.9.** Pruebas unitarias para el caso de pruebas Mantener Postulante.

## ✓ Pruebas de Integración

### ▪ Proceso de Postulación

El escenario del flujo de este proceso se describe a continuación:

1. El usuario de categoría *Asociación* y rol *Administrativo Perú*, ingresa al sistema, publica la información de las vacantes y entrevistas por institución, y envía e-mails a los representantes de cada institución con la información publicada.
2. El usuario de categoría *Institución* ingresa al sistema, registra los datos personales de cada postulante y envía una invitación con información de usuario y contraseña para cada postulante.
3. El usuario de categoría *Postulante* ingresa al sistema, registra la información faltante y envía e-mail a la institución para su aprobación.
4. El usuario de categoría *Institución* ingresa al sistema, verifica los datos ingresados por los postulantes, preselecciona a los postulantes que seguirán en el proceso de postulación y envía la información a la secretaria ejecutiva de la asociación.
5. El usuario de categoría *Asociación* y rol *Administrativo Perú*, ingresa al sistema, revisa la información completada de los postulantes y registra las conclusiones sobre cada ficha de postulante, preseleccionando así a los que serán entrevistados. Luego, envía e-mails, a los representantes de las instituciones, con el listado de postulantes que se entrevistarán.
6. El usuario de categoría *Institución* ingresa al sistema, revisa la lista de postulantes seleccionados para ser entrevistados y les asigna una fecha y hora de entrevista a cada uno, de acuerdo al rango de horas en el que la asociación puede entrevistar. Luego, envía e-mails a cada postulante seleccionado con información de la entrevista asignada y también a la secretaria ejecutiva la lista de horas asignadas a los postulantes.
7. El usuario de categoría *Postulante* ingresa al sistema y visualiza los datos de la entrevista.
8. El usuario de categoría *Asociación* y rol *Administrativo Perú*, ingresa al sistema y selecciona a los postulantes que, tras haber asistido a la entrevista, podrían ser nuevos becarios. Luego, envía e-mails a los representantes del directorio suizo con la lista de postulante seleccionados para ser posibles becarios.
9. El usuario de categoría *Asociación* y rol *Administrativo Suiza*, ingresa al sistema y selecciona a los postulantes que, según su criterio, serán admitidos como becarios en la asociación, aprobando un porcentaje de préstamo por cada uno. Luego, envía e-mail a la secretaria ejecutiva de la asociación con la lista de los postulantes admitidos.
10. El usuario de categoría *Asociación* y rol *Administrativo Perú*, ingresa al sistema y envía e-mails a los representantes de las instituciones la lista de los postulantes admitidos por el directorio suizo.

En la tabla 4.10 se muestran las pruebas de integración de este proceso.

Cabe señalar que en el Anexo I: Catálogo de Pruebas de Integración, se pueden encontrar los casos de prueba de los demás procesos principales.

Id	Objetivo	Precondición	Descripción	Resultados esperados
1	Validar el ingreso al sistema por parte de un usuario, según la categoría que le pertenece.	El usuario se encuentra registrado en el sistema.	Esta prueba se aplica a cada usuario que intente ingresar al sistema durante el escenario descrito. Se considera el Caso de Prueba: Validar Ingreso al Sistema. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 1.	Se muestra la página principal del sistema.
2	Verificar la publicación de la información de vacantes y entrevistas.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 1 del escenario descrito. Se considera el Caso de Prueba: Publicar Información. Detalle en Anexo G: Ventanas del Proceso de Postulación - Gráfico 2 y Gráfico 3.	Se registra información de vacantes por institución, así como datos de fechas y horas de entrevistas.
3	Verificar el envío de la información publicada, a las instituciones.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 1 del escenario descrito. Se considera el Caso de Prueba: Enviar e-mail con información publicada. Detalle en Anexo G: Ventanas del Proceso de Postulación - Gráfico 4 y Gráfico 5.	Se envió la información de vacantes por institución, así como datos de fechas y horas de entrevistas a los representantes de las instituciones.
4	Verificar el registro de información básica de un postulante para enviar invitación.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 2 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 6, Gráfico 7 y Gráfico 8.	Se registra información básica del postulante.
5	Verificar el envío de una invitación a un postulante.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 2 del escenario descrito. Se considera el Caso de Prueba: Enviar Invitación. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 9, Gráfico 10 y Gráfico 11.	Se envía la invitación al postulante, con datos de usuario y contraseña para que complete su información. Se cambia de estado a cada postulante invitado.
6	Verificar el registro de la información faltante por parte de un postulante.	El usuario pertenece a la categoría <i>Postulante</i> .	Esta prueba se aplica al punto 3 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 12, Gráfico 13, Gráfico 14 y Gráfico 15.	Se actualiza la información del postulante.

Id	Objetivo	Precondición	Descripción	Resultados esperados
7	Verificar el envío de confirmación de haber completado la información faltante por parte del postulante.	El usuario pertenece a la categoría <i>Postulante</i> .	Esta prueba se aplica al punto 3 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 16 y Gráfico 17.	Se envía con éxito un e-mail de confirmación de haber completado la información faltante del postulante.
8	Verificar el registro de la información del postulante, modificada por la institución a la que pertenece.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 4 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 18, Gráfico 19, Gráfico 20 y Gráfico 21.	Se actualiza la información del postulante.
9	Verificar que se realice con éxito la preselección de los postulantes que continuarán en el proceso de postulación.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 4 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 22 y Gráfico 23.	Se preselecciona con éxito a un postulante y se cambia su estado.
10	Verificar el envío de e-mail con la lista de estudiantes preseleccionados por la institución a la secretaria ejecutiva de la asociación.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 4 del escenario descrito. Se considera el Caso de Prueba: Enviar listado de preseleccionados. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 24, Gráfico 25 y Gráfico 26.	Se envía el listado de postulantes preseleccionados por la institución.
11	Verificar el registro de las conclusiones en la sección de Informes de la ficha de un postulante.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 5 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 27, Gráfico 28, Gráfico 29 y Gráfico 30.	Se actualiza la información del postulante.
12	Verificar que se realice con éxito la selección de postulantes preseleccionados por las instituciones, que serán entrevistados.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 5 del escenario descrito. Se considera el Caso de Prueba: Mantener Postulante. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 31 y Gráfico 32.	Se actualiza la información del postulante y se cambia su estado.

Id	Objetivo	Precondición	Descripción	Resultados esperados
13	Verificar el envío de e-mail a cada institución con el listado de los postulantes seleccionados para ser entrevistados.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 5 del escenario descrito. Se considera el Caso de Prueba: Enviar listado de entrevistados. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 33, Gráfico 34 y Gráfico 35.	Se envía el listado de los postulantes seleccionados para pasar a una entrevista.
14	Verificar la asignación de fecha y hora de la entrevista a un postulante.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 6 del escenario descrito. Se considera el Caso de Prueba: Asignar Entrevista. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 36, Gráfico 37, Gráfico 38 y Gráfico 39.	Se asigna la fecha y hora de la entrevista al postulante seleccionado.
15	Verificar el envío de e-mail a la secretaria ejecutiva, con la lista de horas asignadas a los postulantes a entrevistar.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 6 del escenario descrito. Se considera el Caso de Prueba: Enviar datos de entrevista. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 40 y Gráfico 41.	Se envían a la secretaria ejecutiva la lista de horas de entrevista asignadas.
16	Verificar el envío de e-mail a cada postulante seleccionado para ser entrevistado, con información de la fecha y hora asignadas.	El usuario pertenece a la categoría <i>Institución</i> .	Esta prueba se aplica al punto 6 del escenario descrito. Se considera el Caso de Prueba: Enviar datos de entrevista. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 42 y Gráfico 43.	Se envían los datos de la entrevista a cada postulante seleccionado.
17	Verificar la visualización de los datos de la entrevista para un postulante.	El usuario pertenece a la categoría <i>Postulante</i> .	Esta prueba se aplica al punto 7 del escenario descrito. Se considera el Caso de Prueba: Ver Datos de Entrevista. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 44.	El postulante puede visualizar todos los datos referentes a la entrevista asignada.
18	Verificar que se realice con éxito la selección de los postulantes que podrían ser admitidos por el directorio suizo.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 8 del escenario descrito. Se considera el Caso de Prueba: Seleccionar postulantes. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 45 y Gráfico 46.	Se actualiza el estado de cada postulante seleccionado.



Id	Objetivo	Precondición	Descripción	Resultados esperados
19	Verificar el envío de e-mail al directorio suizo con el listado de los postulantes seleccionados para ser posibles nuevos becarios.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 8 del escenario descrito. Se considera el Caso de Prueba: Enviar lista de postulantes seleccionados. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 47 y Gráfico 48.	Se envía la lista de postulantes seleccionados al directorio suizo.
20	Verificar que se realice con éxito la admisión de un postulante que ha sido seleccionado por la secretaria ejecutiva.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Suizo</i> .	Esta prueba se aplica al punto 9 del escenario descrito. Se considera el Caso de Prueba: Admitir postulantes. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 49, Gráfico 50, Gráfico 51, Gráfico 52, Gráfico 53 y Gráfico 54.	Se admite a un postulante, registrándose como un nuevo becario.
21	Verificar el envío de e-mail a la secretaria ejecutiva con el listado de los postulantes admitidos.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Suizo</i> .	Esta prueba se aplica al punto 9 del escenario descrito. Se considera el Caso de Prueba: Enviar listado de admitidos. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 55, Gráfico 56 y Gráfico 57.	Se envía la lista de postulantes admitidos a la secretaria ejecutiva.
22	Verificar el envío de e-mail a cada institución con el listado de los postulantes admitidos.	El usuario pertenece a la categoría <i>Asociación</i> y rol <i>Administrativo Perú</i> .	Esta prueba se aplica al punto 10 del escenario descrito. Se considera el Caso de Prueba: Enviar listado de admitidos. Detalle en Anexo G: Ventanas del Proceso de Postulación – Gráfico 58, Gráfico 59 y Gráfico 60.	Se envía un e-mail a cada representante de las instituciones con el listado de los alumnos admitidos como nuevos becarios.

**Tabla 4.10.** Pruebas de integración del proceso de Postulación.

#### 4.2.4. Reporte de pruebas

En la tabla 4.11 se muestra el formato que se usará para registrar los resultados de las pruebas realizadas sobre el producto.

Número de Prueba	Fecha	Resultado Esperado	Resultado Real	Comentarios
1	15/09/2008	Exitoso	Exitoso	El postulante ha sido correctamente registrado, de acuerdo a los datos consultados posteriormente.

**Tabla 4.11.** Formato del reporte de pruebas.

Donde:

- **Número de prueba:** número asignado a la prueba en el catálogo de pruebas.
- **Fecha:** representa la fecha en la que se realiza la prueba.
- **Resultado esperado:** estado final que se espera que tenga la prueba, el cual puede ser: exitoso (en caso se espere que la prueba finalice con éxito) o fallido (en caso se espere que la prueba no finalice con éxito).
- **Resultado real:** estado final real de la prueba, el cual puede ser: exitoso (en caso la prueba finalice con éxito), pendiente (en caso aún no se haya realizado la prueba) o fallido (en caso la prueba no finalice con éxito).
- **Comentarios:** columna en la que se colocarán las anotaciones adicionales que puedan ser consideradas pertinentes sobre los casos de prueba finalizados.

## 5. Observaciones, conclusiones y recomendaciones

El presente capítulo contendrá tres secciones que se describen a continuación:

- En la primera sección (Observaciones), se listarán todos aquellos puntos importantes ocurridos en el proyecto.
- En la segunda sección (Conclusiones), se mencionarán las acciones realizadas para seguir con los objetivos propuestos inicialmente.
- En la tercera sección (Recomendaciones y trabajos futuros), se listarán una serie de recomendaciones que se debieran seguir para mejorar la calidad del producto, en base a la experiencia vivida por las dos tesis del presente proyecto.

### 5.1. Observaciones

- Las continuas reuniones con el usuario final permitieron reevaluar los requerimientos y el alcance real de la solución planteada inicialmente.
- En relación al punto anterior, se observó que el número de tesis asignados inicialmente para la elaboración del presente proyecto no era suficiente para culminar en el tiempo estimado.
- La utilización de *software libre* para el desarrollo de la solución fue una de las características que hizo posible que el proyecto sea viable, dado que las licencias de *software comercial* son costosas, evitando así el pago por las licencias que, muchas veces, organizaciones como la asociación educativa para la que se realiza este proyecto, no pueden costear.
- La desventaja de la utilización de herramientas libres están en la falta de soporte y documentación. Ello significó en algunos casos, como la integración de la solución con la herramienta de exportación de archivos, invertir tiempo en pruebas de integración de tecnologías hasta conseguir una arquitectura estable.

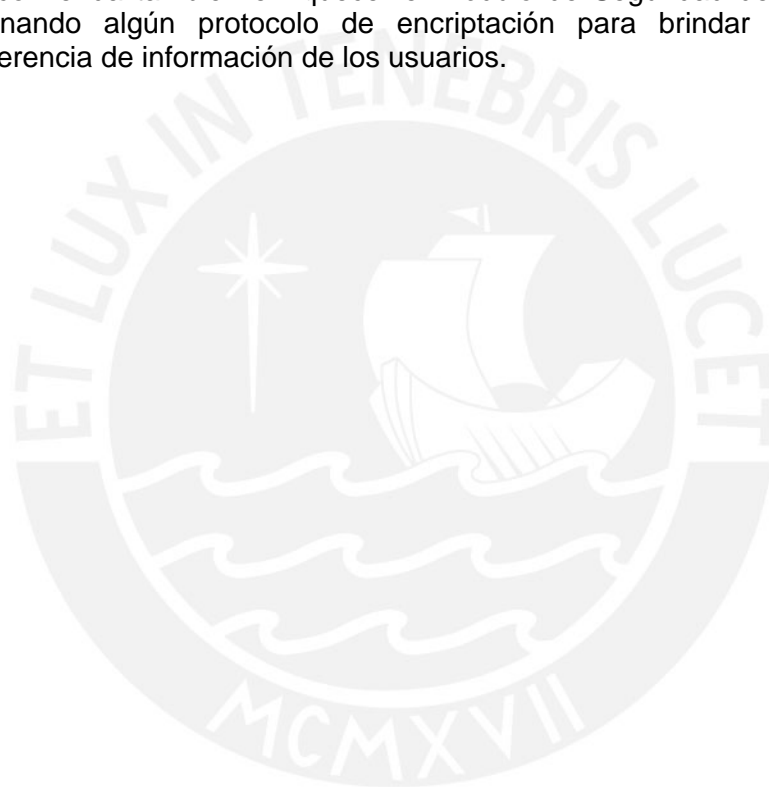
## 5.2. Conclusiones

- La utilización de metodologías ágiles en el presente proyecto han brindado un buen resultado, pues a diferencia de las convencionales (por ejemplo: RUP), proporcionan mayor flexibilidad ante los cambios inesperados y constantes de los requerimientos por parte de los usuarios finales, así como también brindan la posibilidad de tener un código de mejor calidad (mejor diseño, menos defectos) al trabajar en pares y permiten modificarlo sin alterar el comportamiento externo para mejorar su diseño y estructura interna.
- La realización y presentación constante de un prototipo de la aplicación final, hacia los usuarios finales, permitió que éstos puedan visualizar y validar lo que los tesisistas interpretaban de las reuniones. Así, la información presentada en el producto final fue totalmente aprobada por los usuarios finales y de acuerdo a sus preferencias.
- La realización de un plan de pruebas unitarias y de integración, brinda la posibilidad de obtener un producto que cumpla con los requisitos especificados por el usuario, reduciendo así costos durante la implementación de la aplicación y la posibilidad de errores.
- La aplicación ha sido planteada de forma tal que no sólo soluciona las necesidades encontradas sino que también mejora ciertos procesos, en cuanto a tiempo de realización y manera de visualizar la información crítica de forma efectiva. Ello es debido a que actualmente todos los procesos son manuales, con información incongruente en la mayoría de los casos y la única manera de visualizarlos es físicamente; sin embargo, con el presente proyecto se mejorará la gestión de la información, no solo de los procesos de negocio críticos sino también procesos que sirven de apoyo, a la vez que se mejorará la comunicación y el nivel de participación de las instituciones y los alumnos.
- La interfaz gráfica de usuario fue planteada y diseñada siguiendo las necesidades y preferencias de los usuarios finales, las cuales fueron identificadas en las distintas reuniones realizadas con dichos usuarios.
- Si bien es cierto que las últimas tecnologías de desarrollo favorecerán el mantenimiento posterior del producto, éstas también implicaron – en el período de desarrollo – considerar un tiempo extra para el tiempo de investigación y aplicación de dichas tecnologías en el sistema planteado.
- Las reuniones con los usuarios finales son sumamente necesarias si se desea enriquecer el producto, esto es levantar información de acuerdo a las metodologías usadas.

## 5.3. Recomendaciones y trabajos futuros

- Se recomienda el uso de los resultados de este proyecto como base para sucesivos desarrollos de sistemas similares a mayor escala y bajo contextos pertinentes.
- Para centrarse más en lo que quiere el usuario final (producto) se recomienda utilizar una metodología ágil, ya que se enfoca más en el producto que en la documentación. Para el presente proyecto fue de gran ayuda utilizar metodologías ágiles para conocer los verdaderos requerimientos de los usuarios finales, haciendo las reuniones más participativas e informales, de modo que los usuarios tengan la mayor comodidad de expresar sus ideas de cómo quisieran que funcione y se vea el producto final.
- El manejo de un diagrama de Gantt a lo largo del proyecto es indispensable para manejar el avance del mismo de forma ordenada, eficaz y que permita tomar precauciones y soluciones a tiempo.

- Se recomienda que para trabajos futuros, la etapa de implementación se realice en un entorno similar al final (características similares a las de la plataforma que albergará al sistema). Ello para reducir tiempo en adaptar el producto obtenido (sistema) a su entorno final.
- Un sistema web debe ser probado en varios navegadores y versiones de estos. Así, se verifica que en todos se vea la misma estructura y disposición de la información, con lo cual se satisface al 100% a los usuarios finales.
- Se recomienda modularizar y comentar todo el código posible, ello facilitará el mantenimiento posterior del software. Además posibilita a cualquier desarrollador hacerlo, no necesariamente al equipo que lo desarrolló inicialmente.
- Para trabajos futuros, se recomienda implementar un módulo de Ayuda y soporte técnico que permita a los usuarios registrar las ocurrencias durante el uso del sistema, de modo que se brinde información adecuada sobre las posibles soluciones o descripción de los distintos mensajes de error que el sistema pueda mostrar.
- Se recomienda también enriquecer el módulo de Seguridad del presente proyecto, adicionando algún protocolo de encriptación para brindar confiabilidad en la transferencia de información de los usuarios.





## Bibliografía

- [ABI 2008] Andrew Binstock. Eclipse 3.3 or NetBeans 6.0?. 2008. Estados Unidos de América.  
En: <http://www.javaworld.com/javaworld/jw-03-2008/jw-03-java-ides0308.html>  
Último acceso: viernes 12/09/2008
- [AFR 2002] Andrés Fernando Ruiz, La Importancia de la educación, 2002, Colombia.  
En <http://www.gestiopolis.com/canales/economia/articulos/34/educacion.htm>  
Último acceso: viernes 18/04/2008
- [ALO 2008] Álvaro López Ortega. Cherokee 0.8 Documentation: Getting Started. 2008. Web.  
En: <http://www.cherokee-project.com/doc/basics.html>  
Último acceso: viernes 12/09/2008
- [ANS 2008] Ansueta, SCRUM: metodología “ágil” para tus proyectos, 2008, Web.  
En: <http://pymecrunch.com/scrum-metodologia-agil-para-tus-proyectos>  
Último acceso: viernes 26/04/2008
- [ASS 2008] The Apache Software Foundation. Struts. 2008. Web.  
En: <http://struts.apache.org/>  
Último acceso: domingo 14/09/2008
- [AST 2008] The Apache Software Foundation. Apache Tomcat. 2008. Web.  
En: <http://tomcat.apache.org/>  
Último acceso: sábado 13/09/2008
- [BCP 2004] Banco de crédito, Información para tu desarrollo - Becas - INSTITUCIONES INTERNACIONALES, 2004, Perú.  
En: <https://encontacto.bcp.com.pe/reclutamiento/bcp/inf-desarrollo/becas/inst-inter.asp>  
Último acceso: viernes 18/04/2008
- [CAJ 2007] Ciberaula Java. Patrones de Diseño en aplicaciones Web con Java J2EE. 2007. España.  
En: [http://java.ciberaula.com/articulo/disenio\\_patrones\\_j2ee/](http://java.ciberaula.com/articulo/disenio_patrones_j2ee/)  
Último acceso: sábado 13/09/2008
- [CHR 2007] Chuidiang Roxas, SCRUM, 2007, Web.  
En: <http://www.chuidiang.com/ood/metodologia/scrum.php>  
Último acceso: viernes 26/04/2008
- [CNE 2005] Concejo Nacional de Educación, Renovado Sistema de Educación Superior Articulado al Desarrollo Educación Superior: De qué situación partimos, 2005, Perú.  
En: <http://pen.cne.gob.pe/upload/RESULTADO9.doc>  
Último acceso: viernes 18/04/2008
- [CSG 2004] Carlos Sánchez Gonzales, Ciclo de vida de un proyecto XP, 2004, España.  
En: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>  
Último acceso: viernes 18/04/2008

- [CDG 2008] Carlos D. González. Introducción a las Java Server Pages (JSP). 2008. Argentina.  
En: <http://usabilidadweb.com.ar/jsp.php>  
Último acceso: domingo 14/09/2008
- [COR 2008] Comunidad Ruby. Acerca de Ruby. 2008. Web  
En: <http://www.ruby-lang.org/es/about/>  
Último acceso: martes 09/09/2008
- [DKL 2004] Kay Lewis – Director de la oficina de ayuda financiera para estudiantes de la Universidad de Washington, Scholarship System Scoping Study, 2004, Estados Unidos.  
En: [http://www.washington.edu/president/tacs/itac/meetings/2003-04/materials/proposals/scholarship\\_system\\_proposal.pdf](http://www.washington.edu/president/tacs/itac/meetings/2003-04/materials/proposals/scholarship_system_proposal.pdf)  
Último acceso: viernes 25/04/2008
- [DOW 2006] Don Wells, Extreme Programming: A gentle introduction, 2006, Web.  
En: <http://www.extremeprogramming.org/rules.html>  
Último acceso: domingo 31/08/2008
- [DPE 2007] Daniel Pecos. PostgreSQL vs. MySQL. 2007. España.  
En: [http://www.netpecos.org/docs/mysql\\_postgres/x108.html#AEN110](http://www.netpecos.org/docs/mysql_postgres/x108.html#AEN110)  
Último acceso: sábado 13/09/2008
- [DPV 2007] Damián Pérez Valdés. Los diferentes lenguajes de programación para la web. 2007. Web.  
En: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>  
Último acceso: martes 09/09/2008
- [DRA 2007] Diego F. Romero. ¿Qué es un servidor de aplicaciones?. 2007. México.  
En: <http://www.editum.org/Que-Es-Un-Servidor-De-Aplicaciones-p-473.html>  
Último acceso: sábado 13/09/2008
- [DRW 2007] Diego F. Romero. ¿Qué es un servidor web?. 2007. México.  
En: <http://www.editum.org/Que-Es-Un-Servidor-Web-p-401.html>  
Último acceso: sábado 13/09/2008
- [DYS 2007] Derek Yang Shen. Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real. 2007. Web.  
En: [http://www.programacion.com/java/tutorial/jap\\_jsfwork/](http://www.programacion.com/java/tutorial/jap_jsfwork/)  
Último acceso: domingo 14/09/2008
- [ESW 2003] Erik Swenson. Web apps in a snap. 2003. Web.  
En: <http://www.javaworld.com/javaworld/jw-03-2003/jw-0307-opensourceprofile.html>  
Último acceso: domingo 14/09/2008
- [GFC 2008] GlassFish Community. GlassFish Users and Application Developers. 2008. Web.  
En: <https://glassfish.dev.java.net/public/users.html>  
Último acceso: sábado 13/09/2008
- [GMC 1996] James Gosling, Henry McGilton. White Paper: The Java Lenguaje Environment. 1996. Web.  
En: <http://java.sun.com/docs/white/langenv/> Último acceso: martes 09/09/2008

- [IPN 2008] Instituto Politécnico Nacional, Sistema de Administración de Becas escolares (SABE), 2008, México.  
En: <http://www.sabe.ipn.mx/becas/menuprincipalacceso.asp>  
Último acceso: viernes 25/04/2008
- [JBC 2008] JBoss Community. JBoss Application Server. 2008. Web.  
En: <http://www.jboss.org/jbossas/>  
Último acceso: sábado 13/09/2008
- [JCA 2007] Jonathan Campbell. JBoss, Geronimo, or Tomcat. 2007. Estados Unidos de América.  
En: <http://www.javaworld.com/javaworld/jw-12-2007/jw-12-appservers.html>  
Último acceso: sábado 13/09/2008
- [JCI 2006] Jaime Cid. JBoss AS vs Sun Application Server. 2006. Web.  
En: [http://blogs.sun.com/jaimecid/entry/jboss\\_vs\\_sun\\_application\\_server](http://blogs.sun.com/jaimecid/entry/jboss_vs_sun_application_server)  
Último acceso: sábado 13/09/2008
- [JHA 2008] Jeff Hanson. Is Tomcat an application server?. 2008. Estados Unidos de América.  
En: <http://www.javaworld.com/javaworld/jw-01-2008/jw-01-tomcat6.html>  
Último acceso: viernes 12/09/2008
- [JGU 2006] Javier Gutiérrez. ¿Qué es un framework web?. 2006. España.  
En: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)  
Último acceso: domingo 31/08/2008
- [JWI 2008] Joaquín Windmuller, Desarrollo de software: probando Extreme programming, 2008, Venezuela.  
En: <http://aikon.com.ve/metodologias-desarrollo-software-extreme-programming/>  
Último acceso: martes 19/08/2008
- [KAZ 2007] Kalid Azad. Intermediate Rails: Understanding Models, Views and Controllers. 2007. Estados Unidos de América.  
En: <http://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/>  
Último acceso: domingo 14/09/2008
- [MBL 2004] Monson-Haefel, Richard; Burke, Bill; Labourey, Sacha. "Enterprise JavaBeans", 4ta edición. O'Reilly Media, Inc. 2004.
- [MCO 2008] Microsoft Corporation. ASP .Net Overview. 2008. Web.  
En: <http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>  
Último acceso: miércoles 17/09/2008
- [MDE 2008] Ministerio de educación, Oficina de Becas y Crédito Educativo - Créditos Educativos, 2008, Perú.  
En: <http://www.minedu.gob.pe/becas/creditos.php>  
Último acceso: viernes 18/04/2008
- [MSA 2008] MySQL AB. Reference Manual: Panorámica del sistema de gestión de base de datos MySQL. 2008. Web.  
En: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>  
Último acceso: sábado 13/09/2008

- [NBC 2008] NetBeans Community. NetBeans IDE 6.5 Features. 2008. Web.  
En: <http://www.netbeans.org/features/index.html>  
Último acceso: miércoles 25/11/2008
- [NBV 2006] NetBeans Community. NetBeans Visual Web Pack 5.5. 2006. Web.  
En: [http://www.netbeans.org/kb/55/vwp-intro\\_es.html](http://www.netbeans.org/kb/55/vwp-intro_es.html)  
Último acceso: miércoles 17/09/2008
- [NRC 2004] Centro de cómputos de Universidad Nacional de Río Cuarto (UNRC), Sistema de Becas (SIBE), 2004, Argentina.  
En: <http://www.cdc.unrc.edu.ar/sistemas/sisbe.htm>  
Último acceso: viernes 25/04/2008
- [OSY 2007] OpenSymphony. WebWork. 2007. Web.  
En: <http://www.opensymphony.com/webwork/>  
Último acceso: domingo 14/09/2008
- [OTI 2003] Object Technology International. Eclipse Platform Technical Overview. 2003. Estados Unidos de América.  
En: <http://www.eclipse.org/whitepapers/eclipse-overview.pdf>  
Último acceso: miércoles 17/09/2008
- [OTN 2008] Oracle Technology Network. Oracle JDeveloper. 2008. Web.  
En: <http://www.oracle.com/technology/products/jdev/index.html>  
Último acceso: miércoles 17/09/2008
- [PCO 2008] Python Community. About Python. 2008. Web.  
En: <http://www.python.org/about/>  
Último acceso: martes 09/09/2008
- [PGD 2008] PostgreSQL Global Development Group. PostgreSQL: About. 2008. Web.  
En: <http://www.postgresql.org/about/>  
Último acceso: sábado 13/09/2008
- [RBA 2004] Roland Barcia. JavaServer Faces (JSF) vs Struts. 2004. Estados Unidos de América.  
En: <http://websphere.sys-con.com/node/46516>  
Último acceso: domingo 14/09/2008
- [RIC 2006] Richardson, Chris. "POJOs in Action", 1ra edición. Manning Publications. 2006.
- [SCI 2007] SeamCity. Comparativa de Frameworks: WebWork 2. 2007. España.  
En: <http://seamcity.madeinxpain.com/archives/comparativa-webwork-2>  
Último acceso: domingo 14/09/2008
- [SDF 2008] Sun Developer Network. JavaServer Faces Overview. 2008. Web.  
En: <http://java.sun.com/javasee/javaserverfaces/overview.html>  
Último acceso: domingo 14/09/2008
- [SDM 2002] Sun Developer Network. Design Patterns: Model-View-Controller. 2002. Web.  
En: <http://java.sun.com/blueprints/patterns/MVC.html>  
Último acceso: sábado 13/09/2008



- [SDP 2008] Sun Developer Network. JavaServer Pages Overview. 2008. Web.  
En: <http://java.sun.com/products/jsp/overview.html>  
Último acceso: domingo 14/09/2008
- [SSH 2008] Shay Shmeltzer. Oracle JDeveloper Overview. 2008. Estados Unidos de América.  
En:  
[http://www.oracle.com/technology/products/jdev/collateral/papers/1013/jdev1013\\_overview.pdf](http://www.oracle.com/technology/products/jdev/collateral/papers/1013/jdev1013_overview.pdf)  
Último acceso: jueves 11/09/2008
- [TEF 2008] The Eclipse Foundation. Eclipse IDE. 2008. Web.  
En: <http://www.eclipse.org/org/>  
Último acceso: miércoles 17/09/2008
- [TJT 2008] Sun Developer Network - The Java™ Tutorial. About the Java Technology. 2008. Web.  
En: <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>  
Último acceso: martes 09/09/2008
- [TPG 2008] The PHP Group. PHP Manual. 2008. Web.  
En: <http://www.php.net/manual/en/index.php>  
Último acceso: martes 09/09/2008
- [TSN 2007] Todd Snyder. MVC or MVP Pattern – What’s the difference?. 2007.  
En: [http://blogs.infragistics.com/blogs/todd\\_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx](http://blogs.infragistics.com/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx)  
Último acceso: lunes 06/07/2009
- [WIG 2007] Wigahluk, Percepciones acerca de la programación ágil y XP, 2007, Web.  
En: <http://wigahluk.wordpress.com/2007/06/14/percepciones-acerca-de-la-programacion-ágil-y-xp/>  
Último acceso: viernes 19/08/2008