



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB  
PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL  
ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC**

**Sergio Adalberto Rodríguez Sánchez**

Asesorado por el Ing. Edgar Oswaldo Cumes Rodríguez

Guatemala, noviembre de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB  
PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL  
ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**SERGIO ADALBERTO RODRÍGUEZ SÁNCHEZ**

ASESORADO POR EL ING. EDGAR OSWALDO CUMES RODRÍGUEZ

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, NOVIEMBRE DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
EXAMINADOR	Ing. Ricardo Morales Prado
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha agosto de 2011

  
**Sergio Adalberto Rodríguez Sánchez**



Guatemala, 13 de agosto de 2013

Ingeniero

Carlos Azurdia Morales

Escuela de Ciencias y Sistemas

Respetable Ingeniero Azurdia:

Por este medio le informo como asesor del trabajo de graduación del estudiante universitario de la carrera de Ingeniería en Ciencias y Sistemas, SERGIO ADALBERTO RODRÍGUEZ SÁNCHEZ, carné 2003 12682, que he revisado el trabajo de graduación titulado: **“Desarrollo e implementación de una aplicación Web para el control de la evaluación de los estudiantes del área de EPS en la Facultad de Agronomía, USAC”**, y a mi criterio el mismo está completo y actualizado, y cumple con los objetivos propuestos para su desarrollo según el protocolo.

Agradeciendo su atención a la presente,

Atentamente,

A handwritten signature in black ink, appearing to read 'Edgar Cumes'.

*Edgar Oswaldo Cumes Rodríguez  
Ingeniero en Ciencias y Sistemas  
Colegiado No. 4964*

Ing. Edgar Oswaldo Cumes Rodríguez  
Asesor de trabajo de graduación  
Colegiado: 4964



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 4 de Septiembre de 2013

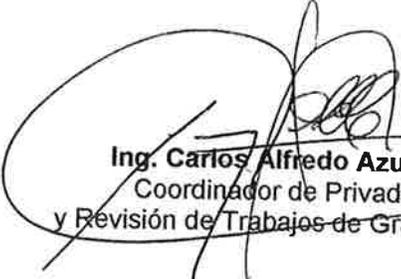
Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **SERGIO ADALBERTO RODRÍGUEZ SÁNCHEZ** carné 2003-12682, titulado: "DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC”**, realizado por el estudiante **SERGIO ADALBERTO RODRÍGUEZ SÁNCHEZ**, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

Ing. Mallón Antonio Pérez Türk  
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 04 de noviembre 2013



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ciencias y Sistemas, al trabajo de graduación titulado: **DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE LA EVALUACIÓN DE LOS ESTUDIANTES DEL ÁREA DE EPS EN LA FACULTAD DE AGRONOMÍA, USAC**, presentado por el estudiante universitario: **Sergio Adalberto Rodríguez Sánchez**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

  
Ing. Murphy Olymoo Paiz Recinos  
Decano



Guatemala, noviembre de 2013

/cc

## **ACTO QUE DEDICO A:**

- Mis padres** Adalberto Rodríguez y Miriam Sánchez, quienes con su esfuerzo y dedicación me permitieron concluir mis estudios y me guiaron por el camino correcto con ejemplo de vida, porque este logro es tanto suyo como mío.
- Mis hermanos** Ana Cecilia, Ana Lucía e Iván Rodríguez Sánchez, con quienes convivo, comparto y han sido un ejemplo en mi camino.
- Mis abuelos** Por su cariño y amor que me brindaron y me siguen brindando, en especial a Marcelino Rodríguez, que recientemente pasó al eterno descanso.
- Mis amigos** Sergio Lemus, Omar Meza, Juan Pablo Roldan, Alexander Barahona, Danilo Escobar, Aura Morán, Virginia Sánchez, Álvaro Donis y especialmente a Astrid Reyes, con quienes he compartido a lo largo de mi vida.

## **AGRADECIMIENTOS A:**

**Ing. Edgar Cumes Rodríguez** Por su asesoría en la realización de este trabajo y fuente de inspiración para perseguir esta maravillosa carrera.

**Facultad de Agronomía** En particular, al área de EPS por brindarme la oportunidad de poder aportar una semilla a tan noble facultad.

**Familia Espina León** Por el apoyo que me brindaron, sin ustedes, este trabajo no hubiera sido posible.



1.2.4.	<i>Business Process Model (BPM)</i> .....	16
2.	DEFINICIÓN DEL SISTEMA REQUERIDO .....	19
2.1.	Modelo del proceso de evaluación del área de EPS .....	19
2.2.	Requerimientos del sistema .....	23
2.2.1.	Descripción general del sistema.....	24
2.2.1.1.	Formas de evaluación del EPS de la Facultad de Agronomía .....	24
2.2.2.	Características de los usuarios .....	26
2.2.3.	Restricciones.....	27
2.2.4.	Evolución a futuro del sistema.....	28
2.2.5.	Requerimientos funcionales .....	29
2.2.6.	Requerimientos no funcionales .....	31
2.2.6.1.	Requerimientos de rendimiento.....	31
2.2.6.2.	Requerimientos de seguridad.....	32
2.2.6.3.	Requerimientos de mantenimiento .....	33
2.2.6.4.	Requerimientos de interfaz de usuario.....	33
2.3.	Casos de uso .....	33
2.3.1.	Casos de uso y el proceso general de evaluación .....	48
2.3.2.	Diagrama de casos de uso.....	50
3.	ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	53
3.1.	Metodología de desarrollo .....	53
3.2.	Herramientas de desarrollo .....	56
3.3.	Arquitectura de software del proyecto .....	58
3.4.	Diagramas de secuencia.....	59
3.5.	Diseño relacional de la base de datos.....	68

3.5.1.	Tipos de datos .....	72
3.6.	Diseño de la interfaz de usuario .....	73
4.	DOCUMENTACIÓN TÉCNICA.....	79
4.1.	<i>Scripts</i> de procesamientos de datos.....	79
4.1.1.	<i>Scripts</i> de las páginas iniciales .....	79
4.1.2.	<i>Scripts</i> de administrador .....	81
4.1.3.	<i>Scripts</i> del profesor .....	85
4.1.4.	<i>Scripts</i> del estudiante .....	88
4.2.	Estructura del sitio web.....	89
	CONCLUSIONES .....	91
	RECOMENDACIONES.....	93
	BIBLIOGRAFÍA.....	95
	APÉNDICE.....	97



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	El modelo lineal secuencial.....	4
2.	El modelo iterativo incremental.....	6
3.	Modelo espiral.....	8
4.	Actores y casos de uso .....	11
5.	Diagrama de caso de uso .....	16
6.	Diagrama de flujo del proceso - página 1.....	21
7.	Diagrama de flujo del proceso - página 2.....	22
8.	Diagrama de flujo del proceso - página 3.....	23
9.	Diagrama de casos de uso del sistema .....	50
10.	Diagrama de la metodología de desarrollo.....	55
11.	Ingreso al sistema .....	60
12.	Administrar usuarios.....	61
13.	Administrar períodos .....	62
14.	Administrar asignación .....	63
15.	Administrar notas y penalizaciones.....	64
16.	Administración de foro de mensajes .....	65
17.	Ver notas.....	66
18.	Modificar datos de usuario .....	67
19.	Generar reportes.....	68
20.	Modelo entidad-relación.....	69
21.	Plantilla de la interfaz de usuario .....	74
22.	Interfaz del administrador.....	75
23.	Interfaz del profesor .....	76

24.	Interfaz del estudiante.....	77
25.	Árbol de la estructura del sitio Web .....	90

## TABLAS

I.	Proceso de evaluación del EPS.....	19
II.	Características de los usuarios del sistema.....	26
III.	Requerimientos funcionales .....	29
IV.	Ingreso de usuarios .....	34
V.	Administración de usuarios.....	34
VI.	Creación de usuarios.....	35
VII.	Búsqueda y modificación de usuarios .....	36
VIII.	Administración de períodos .....	37
IX.	Creación de períodos .....	38
X.	Búsqueda y modificación de períodos .....	39
XI.	Administración de asignaciones .....	39
XII.	Creación de asignaciones .....	40
XIII.	Búsqueda y modificación de asignaciones .....	41
XIV.	Administración de notas y penalizaciones .....	41
XV.	Ingreso de notas.....	42
XVI.	Ingreso de penalizaciones .....	43
XVII.	Administración de foro de mensajes.....	44
XVIII.	Lectura de mensajes del foro .....	44
XIX.	Redactar mensajes en el foro.....	45
XX.	Ver notas de estudiante .....	45
XXI.	Modificar datos de usuario.....	46
XXII.	Modificar datos personales de usuario .....	46
XXIII.	Modificar contraseña de usuario.....	47
XXIV.	Generación de reportes .....	48

XXV. Relación de casos de uso con el proceso general ..... 49



## GLOSARIO

<b>Aplicación Web</b>	Programa de software, el cuál es accedido desde Internet y se aloja en algún servidor especializado para alojar este tipo de aplicaciones.
<b>EPS</b>	Acrónimo que significa Estudio Profesional Supervisado.
<b>Internet</b>	La red mundial que utiliza el protocolo TCP/IP para la comunicación.
<b>Iteración</b>	Se refiere a la repetición de un conjunto de pasos definidos.
<b>RDBMS</b>	Acrónimo que significa en inglés <i>Relational Data Base Management System</i> . Se refiere a cualquier sistema de administración de bases de datos.
<b>Llave foránea</b>	Es un campo dentro de una tabla relacional que hace referencia hacia otro campo en otra tabla de la base de datos.
<b>Llave primaria</b>	Es un campo que identifica de forma única a una tabla dentro del modelo relacional de bases de datos.

<b><i>Open Source</i></b>	Término utilizado para referirse al software que se desarrolla y distribuye de forma libre, compartiendo el código del programa.
<b>Plantilla</b>	Patrón predefinido que permite construir un diseño previamente establecido por medio de la sustitución de los componentes originales por los propios.
<b>Plataforma</b>	Se refiere al sistema operativo o programa de software sobre el cual se ejecuta algún otro sistema.
<b>Requerimiento</b>	Conjunto de características y especificaciones que se desea contenga un sistema.
<b><i>Script</i></b>	Archivo con porciones de código con una función especial.
<b>UML</b>	En inglés <i>Unified Model Language</i> , se refiere al lenguaje de modelado respaldado por el <i>Object Model Group</i> .

## RESUMEN

El desarrollo tecnológico, y en específico en el área del desarrollo de software, ha creado la necesidad, en las empresas, instituciones y organizaciones de cualquier tipo, de contar con sistemas que les permitan automatizar distintas funciones dentro de las mismas, con distintos fines como mejorar la productividad, controlar áreas de la organización de una mejor manera, obtener información valiosa para tomar decisiones importantes.

En el área de EPS de la Facultad de Agronomía de la Universidad de San Carlos de Guatemala, actualmente se maneja toda la información importante como notas de estudiantes, datos personales o de instituciones que apoyan al área, en papel o en medios poco eficientes, por lo que es de vital importancia contar con un sistema que apoye a los profesores supervisores, estudiantes, secretarías y coordinadores del área a realizar un mejor trabajo, y que facilite diversas tareas. Una de estas tareas, es el control de notas de evaluación a los estudiantes de agronomía que llegan a esta fase de su carrera.

El control de notas requiere de un sistema especializado, ya que no se evalúa como algún otro curso de la carrera, sino que lleva un método distinto que requiere una aplicación que cumpla con estas políticas, por lo que se desarrolló un sistema Web para cumplir esta función y algunas más que facilitarán el trabajo de los profesores y la comunicación con los estudiantes.



## **OBJETIVOS**

### **General**

Desarrollar e implementar una aplicación de tipo Web para el control de notas en el área de EPS de la Facultad de Agronomía de la Universidad de San Carlos de Guatemala, que cumpla con los requerimientos de los usuarios del sistema.

### **Específicos**

1. Definir los requerimientos del sistema.
2. Determinar la metodología de desarrollo a utilizar.
3. Documentar el análisis de los requerimientos y el diseño de la aplicación.
4. Diseñar una aplicación que cumpla los requisitos definidos por los usuarios.
5. Desarrollar el software diseñado bajo la metodología establecida.
6. Poner en práctica todos los conocimientos de Ingeniería de Software adquiridos durante la carrera de Ingeniería en Ciencias y Sistemas en una aplicación real.
7. Documentar la información técnica de la aplicación.



## INTRODUCCIÓN

El desarrollo de una aplicación Web, conlleva aplicar diversas metodologías para analizar los requerimientos que el usuario final desea, y así poder diseñar y luego desarrollar una aplicación que cumpla con lo que se estableció y satisfaga al cliente.

Para desarrollar esta aplicación para el área de EPS de la Facultad de Agronomía, se inició por definir el sistema, a través de la toma de requerimientos haciendo uso de distintas herramientas como los casos de uso de UML, y se estableció el alcance y los límites del proyecto.

De este punto en adelante, se procedió a diseñar la aplicación, eligiendo una metodología de desarrollo y creando un diseño para la base de datos que guardará toda la información que los usuarios necesitan en forma persistente, para su consulta o modificación.

Ya definido el diseño de la aplicación desde la base de datos hasta la interfaz de usuario, se procedió a desarrollar la aplicación hasta completar las funcionalidades definidas en la toma de requerimientos y la aprobación final de los interesados en el proyecto, lo que llevó a su implementación y desarrollo de pruebas para corregir los errores cometidos. La documentación del proyecto en este trabajo de investigación, servirá como base para que este proyecto pueda evolucionar y ser mejorado o extendido por alguna otra persona a futuro, sin tener que empezar sin un punto de referencia para la comprensión del funcionamiento de este sistema.



# 1. MARCO TEÓRICO

## 1.1. Software

El desarrollo de una aplicación de software implica primero entender el concepto de esta palabra, que a lo largo de los años se ha hecho común en el idioma español. Se empieza con presentar un panorama general que permita entender de mejor manera este concepto.

### 1.1.1. Definición

Software (Ian Sommerville, 2005) son programas de ordenador y la documentación asociada. Los productos de software se pueden desarrollar para algún cliente en particular o para un mercado general.

### 1.1.2. Ingeniería de software

Ingeniería de *software* (Ian Sommerville, 2005) es una disciplina de ingeniería que comprende todos los aspectos de la producción de software. Comprende las formas prácticas para desarrollar y entregar un software útil.

Según Sommerville, existen 2 tipos de software, dependiendo de a qué tipo de clientes va dirigido, puede ser genérico, dirigido a cualquier cliente que desee utilizarlo o comprarlo, o puede ser personalizado, los cuales son sistemas requeridos por un cliente en particular.

Para el caso particular de este proyecto de desarrollo de software, se clasifica dentro del segundo tipo descrito en el párrafo anterior. Esta aplicación Web está dirigida a un cliente en específico (en este caso los profesores y estudiantes de EPS de Agronomía), por lo que la realización de este tipo de sistemas debe ir apegada a los requerimientos que los usuarios plantean y desean obtener (ver requerimientos).

### **1.1.3. Ciclo de vida del software**

Según la Norma ISO (International Organization for Standardization), el ciclo de vida de un software es un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso. Existen diversas etapas definidas por distintos autores dentro de este ciclo, pero en esencia son indispensables las que siguen:

- Toma de requerimientos del sistema
- Análisis y diseño del sistema
- Codificación o desarrollo del sistema
- Pruebas
- Implementación
- Mantenimiento

El orden en que se realizan estas etapas varía dependiendo del modelo aplicado y pueden también contener más o menos etapas que las listadas anteriormente, ya que el desarrollo de software demanda mucha flexibilidad a lo largo de su ciclo.

#### **1.1.4. Modelos del ciclo de vida del software**

Existe una cantidad bastante amplia de modelos de desarrollo de software, aquí se encuentran los modelos que se utilizarán para formar el modelo de desarrollo utilizado en la construcción de esta aplicación.

##### **1.1.4.1. Modelo cascada**

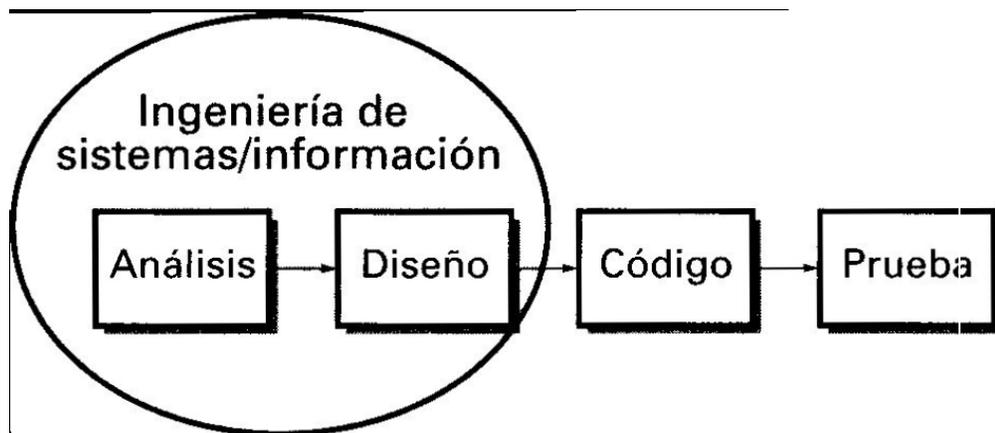
Llamado algunas veces ciclo de vida básico (Pressman, 2002), este modelo sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

Este modelo (Sommerville, 2005), fue el primer modelo de proceso de desarrollo de software que se publicó; se derivó de procesos de ingeniería de sistemas más generales. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

- Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistemas e definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.
- Diseño del sistema y del software. El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.

- Implementación y prueba de unidades. Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
- Integración y prueba del sistema. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.
- Funcionamiento y mantenimiento. Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltarlos servicios del sistema una vez que se descubren nuevos requerimientos.

Figura 1. **El modelo lineal secuencial**



Fuente: Pressman. 2002.

#### **1.1.4.2. Modelo iterativo incremental**

En este modelo se definen las etapas de: descripción, especificación, desarrollo y validación del sistema. La diferencia con el modelo en cascada u otros modelos clásicos es que en este se permite ir entregando versiones y regresando a cada etapa para realizar mejoras y en forma iterativa, alcanzar los objetivos del sistema.

Según Sommerville (Sommerville, 2005), la especificación, el diseño y la implementación del software se dividen en una serie de incrementos, los cuales se desarrollan por tumos. El modelo de desarrollo en cascada requiere que los clientes de un sistema cumplan un conjunto de requerimientos antes de que se inicie el diseño y que el diseñador cumpla estrategias particulares de diseño antes de la implementación.

Los cambios de requerimientos implican rehacer el trabajo de captura de éstos, de diseño e implementación. Sin embargo, la separación en el diseño y la implementación deben dar lugar a sistemas bien documentados susceptibles de cambio. En contraste, un enfoque de desarrollo evolutivo permite que los requerimientos y las decisiones de diseño se retrasen, pero también origina un software que puede estar débilmente estructurado y difícil de comprender y mantener.

La entrega incremental (figura 2) es un enfoque intermedio que combina las ventajas de estos modelos. En un proceso de desarrollo incremental, los clientes identifican, a grandes rasgos, los servicios que proporcionará el sistema. Identifican qué servicios son más importantes y cuáles menos. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema.

Figura 2. **El modelo iterativo incremental**



Fuente: Sommerville. 2005.

#### 1.1.4.3. **Modelo en espiral**

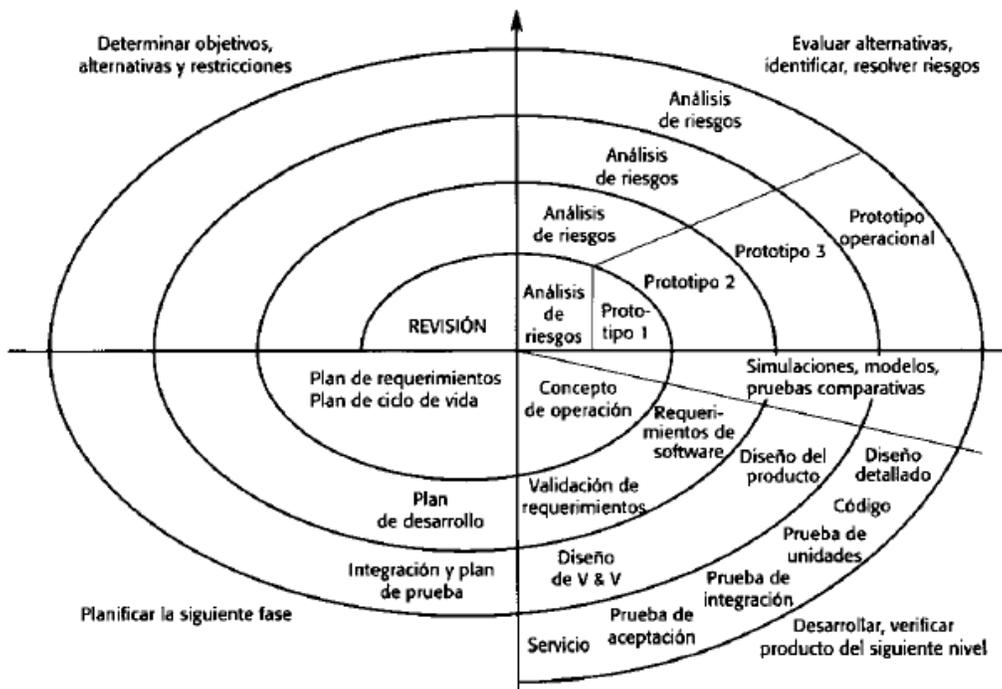
El modelo en espiral (Pressman, 2002), es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

Cada ciclo de la espiral se divide en cuatro sectores (Sommerville, 2005):

- Definición de objetivos. Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso y el producto, y se traza un plan detallado de gestión. Se identifican los riesgos del proyecto. Dependiendo de estos riesgos, se planean estrategias alternativas.

- Evaluación y reducción de riesgos. Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto identificados. Se definen los pasos para reducir dichos riesgo. Por ejemplo, si existe el riesgo de tener requerimientos inapropiados, se puede desarrollar un prototipo del sistema.
- Desarrollo y validación. Después de la evaluación de riesgos, se elige un modelo para el desarrollo del sistema. Por ejemplo, si los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de seguridad son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado, y así sucesivamente. El modelo en cascada puede ser el más apropiado para el desarrollo si el mayor riesgo identificado es la integración de los subsistemas.
- Planificación. El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto.

Figura 3. **Modelo espiral**



Fuente: Sommerville. 1995.

## 1.2. **Lenguaje unificado de modelado**

Tener una notación estándar que permita modelar el proceso, el sistema y el flujo de información permite tener un panorama más claro del producto final que se desea desarrollar. UML, es un estándar utilizado ampliamente en el campo de desarrollo de software.

### **1.2.1. Definición**

Por sus siglas en inglés, UML (*Unified Model Language*), el Lenguaje Unificado de Modelado es la especificación más usada del *Object Model Group* (OMG), y la forma en que el mundo modela no solamente estructuras de aplicaciones, comportamiento y arquitectura, sino también procesos del negocio y estructuras de datos (OMG, 2010).

### **1.2.2. Casos de uso**

Los Casos de Uso (Joaquín Gracia, 2003), son qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Un caso de uso describe un grupo de secuencias en la que cada secuencia representa alguna interacción entre las cosas que se encuentran fuera del sistema (actores) con el sistema en sí. Estos comportamientos son, en efecto, funciones a nivel del sistema que se utilizan para visualizar, especificar, construir y documentar el comportamiento esperado del sistema durante la etapa de análisis y de captura de requerimientos. Un caso de uso representa un requerimiento funcional del sistema como un todo.

Como se mencionó antes, un caso de uso involucra la interacción entre actores y el sistema. Un actor representa un conjunto de roles coherentes que

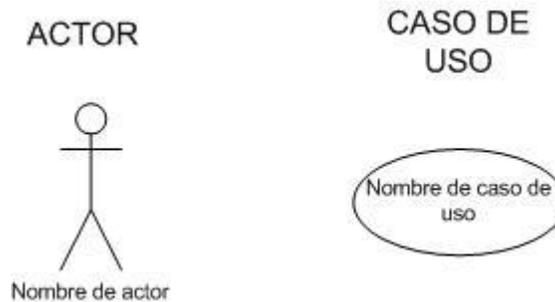
los usuarios de casos de uso juegan cuando interactúan con estos casos de uso. Los actores pueden ser humanos o pueden ser otros sistemas automatizados.

Un caso de uso puede tener variantes. En todos los sistemas que interesen, se encontraran casos de uso que son versiones especializadas de otros casos de uso, como casos de uso que están incluidos como parte de otros casos de uso, y casos de uso que extienden el comportamiento de otros casos de uso básicos. Se puede factorizar el comportamiento común y reutilizable de un conjunto de casos de uso, organizándolos de acuerdo a estas tres clases de relaciones.

Un caso de uso lleva a cabo alguna cantidad tangible de trabajo. Desde la perspectiva de un actor específico, un caso de uso hace algo que es de valor para un actor, como calcular un resultado, generar un nuevo objeto o cambiar el estado de otro objeto.

Se pueden aplicar los casos de uso a todo el sistema y también pueden aplicarse a parte de él, incluyendo subsistemas y hasta clases individuales e interfaces. En cada caso, estos casos de uso no solo representan el comportamiento deseado de estos elementos, sino también pueden ser utilizados como la base de casos de prueba para estos elementos, ya que evolucionan durante el desarrollo. Los casos de uso aplicados a subsistemas son fuentes excelentes de pruebas de sistema y de integración. UML una representación gráfica de un caso de uso y un actor y esta notación permite visualizar un caso de uso independiente de su realización y en contexto con otros casos de uso.

Figura 4. **Actores y casos de uso**



Fuente: elaboración propia.

Los nombres de los casos de uso deben ser únicos dentro del paquete donde se encuentran contenidos para distinguirlos de otros casos de uso. Un nombre debe ser una cadena de caracteres de texto. Ese nombre, solo es conocido como un nombre simple. Un nombre de ruta es el nombre del caso de uso con el nombre del paquete que lo contiene como prefijo. Un caso de uso usualmente se dibuja mostrando solamente su nombre.

#### **1.2.2.1. Casos de uso y actores**

Usualmente un actor representa un rol que un humano, un dispositivo de hardware, o incluso otro sistema juega dentro de un sistema. Una instancia de un actor representa una interacción individual con el sistema de una forma específica. A pesar de que se utilizan los actores en los modelos, éstos no son parte del sistema, habitan afuera de él.

Los actores pueden estar conectados al sistema solamente mediante asociación. Una asociación entre un actor y un caso de uso que estos se

comunican uno con el otro, cada uno probablemente, enviando y recibiendo mensajes.

#### **1.2.2.2. Casos de uso y flujos de eventos**

Un caso de uso describe lo que un sistema hace pero no especifica como lo hace. Cuando se modela es importante mantener clara la separación de lo que concierne entre la vista externa y la interna.

Se puede especificar el comportamiento de un caso de uso describiendo un flujo de eventos en un texto lo suficientemente claro para que alguien ajeno pueda entenderlo fácilmente. Cuando se escriben estos flujos de eventos, se debería incluir como y cuando el caso de uso comienza y termina, cuándo el caso de uso interactúa con los actores y qué objetos son intercambiados, y el flujo básico y los flujos alternativos del comportamiento.

#### **1.2.2.3. Casos de uso y escenarios**

Usualmente, primero se describirá el flujo de eventos para el caso de uso en texto. Mientras se perfecciona el entendimiento de los requerimientos del sistema, sin embargo, se necesitará utilizar diagramas para especificar estos flujos gráficamente, típicamente se utilizaría un diagrama de secuencia para especificar el flujo principal de un caso de uso, y variaciones de ese diagrama para especificar flujos excepcionales de un caso de uso.

Es deseable separar el flujo principal versus los alternativos porque un caso de uso describe un conjunto de secuencias, no solo una secuencia simple, y sería imposible expresar todos los detalles de un caso de uso en tan solo una secuencia. Por ejemplo, en un sistema de recursos humanos, se podría

encontrar un caso de uso Contratar empleado. Esta función general de negocios podría tener muchas variantes posibles; uno podría contratar a una persona de otra compañía, podría transferirse a una persona de una división hacia otra o podría contratar a un extranjero. Cada una de estas variantes puede ser expresada en una secuencia diferente.

Este caso de uso en específico, en realidad describe a un conjunto de secuencias en el que cada secuencia de ese conjunto representa un posible flujo a través de todas estas variaciones. Cada secuencia es llamada un escenario. Un escenario es una secuencia específica de acciones que ilustran un comportamiento. Los escenarios con para los casos de uso como las instancias con para las clases, significando que un escenario es básicamente una instancia de un caso de uso.

#### **1.2.2.4. Casos de uso y colaboraciones**

Un caso de uso captura el comportamiento esperado del sistema a desarrollar, sin tener que especificar como se implementa ese comportamiento. Esta es una separación importante porque el análisis de un sistema debería, en la medida de lo posible, no ser influenciado por los problemas de implementación. Al final, sin embargo, se deben implementar los casos de uso, y se hace creando una asociación de clases y otros elementos que trabajan juntos para implementar el comportamiento de este caso de uso. Esta asociación de elementos, incluyendo tanto su estructura estática como dinámica, es modelada en el UML como una colaboración.

### **1.2.3. Diagramas de casos de uso**

Los casos de uso son uno de los cinco diagramas en el UML para modelar los aspectos dinámicos de los sistemas. Los diagramas de casos de uso son centrales para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso y actores y sus relaciones.

Se aplican los diagramas de casos de uso para modelar la vista del caso de uso de un sistema. Para la mayor parte, esto involucra modelar el contexto del sistema, subsistema o clase, o modelar los requerimientos del comportamiento de estos elementos.

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Hacen a los sistemas, subsistemas y clases, accesibles y entendibles, al presentar una vista externa de cómo esos elementos podrían ser utilizados en contexto. Los diagramas de casos de uso son también importantes para probar sistemas ejecutables a través de ingeniería directa y para comprender estos sistemas a través de ingeniería inversa.

Los diagramas de casos de uso usualmente contienen:

- Casos de uso
- Actores
- Relaciones de dependencia, generalización y asociación

Al igual que muchos otros diagramas, estos pueden contener notas y restricciones, al igual que paquetes, que se utilizan para agrupar elementos del

modelo en grandes pedazos. Ocasionalmente se querrán colocar instancias de casos de uso en los diagramas, especialmente cuando se quiere visualizar un sistema ejecutable específico.

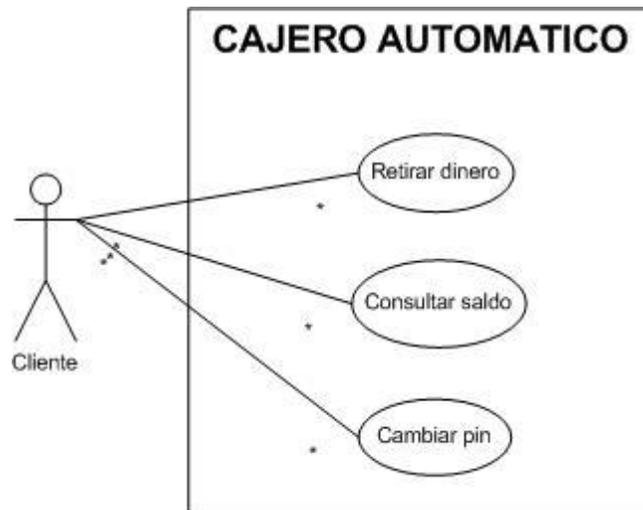
### **1.2.3.1. Usos comunes de los diagramas de casos de uso**

Se aplican los diagramas de casos de uso para modelar la vista del caso de uso estático de un sistema. Esto principalmente, apoya el comportamiento del sistema.

Cuando se modela la vista del caso de uso estático de un sistema, se aplicaran los diagramas de caso de uso en una de estas dos formas:

- Para modelar el contexto del sistema, que involucra dibujar una línea alrededor de todo el sistema y especificando que actores se encuentran afuera del sistema e interactúan con él. Aquí se utilizan los diagramas de caso de uso para especificar a los actores y el significado de sus roles.
- Para modelar los requerimientos del sistema, que involucra especificar lo que ese sistema debería hacer, independientemente de cómo debería hacerlo.

Figura 5. Diagrama de caso de uso



Fuente: elaboración propia.

#### 1.2.4. *Business Process Model (BPM)*

Modelar los procesos del negocio es una parte vital para el diseño de una herramienta que vaya a desarrollarse para solventar algún problema, es por esto mismo que tener documentación sobre los procesos para cualquier actividad dentro de la empresa, es de mucha utilidad.

Documentar uno o varios procesos que vayan a involucrarse en el desarrollo de alguna herramienta de software ayuda al diseñador a entender el funcionamiento real del negocio, los recursos y actores involucrados, y así poder determinar de mejor manera el enfoque del diseño de la herramienta, para que esta esté orientada a mejorar el proceso en general y cumplir con lo que se requiere al final de la herramienta.

Existen diversidad de estándares y notaciones que ayudan a alcanzar esta tarea de modelar procesos, UML es uno de ellos, que desde su versión 2, incluye extensiones para el modelado de procesos. Esta extensión se conoce como *Business Process Model Notation* (BPMN), que no es en sí parte del UML, sino que es una definición que se adapta a los diagramas que posee la definición de UML desde su versión 2 en adelante.

Esta notación define diagramas y notaciones que nos ayudan a entender todas las actividades y elementos involucrados en el proceso. En UML podemos utilizar los diagramas de flujo para representar a los procesos deseados y complementarlos con diagramas de secuencias u otros que se consideren necesarios para tener un panorama más detallado de cómo se realizan los distintos procesos dentro del negocio.

Utilizar diagramas de flujo para modelar los procesos nos permite tener una notación que es ampliamente utilizada para diseñar flujos de sistemas de software, lo que permite asimilar el proceso de una manera que está estrechamente relacionada con el diseño de nuestra aplicación que pretende dar solución al requerimiento solicitado.



## 2. DEFINICIÓN DEL SISTEMA REQUERIDO

Antes de empezar un desarrollo de software, se deben definir las bases sobre las cuales se empezará a construir. Estas bases corresponden a la definición de requerimientos, flujo del proceso y delimitación del sistema requerido, por lo que en esta sección se detallan todos estos detalles esenciales para el diseño de la aplicación.

### 2.1. Modelo del proceso de evaluación del área de EPS

A continuación se presenta la descripción detallada del proceso de evaluación de un estudiante en la fase de EPS, y los involucrados en el flujo de este, seguido del diagrama de flujo correspondiente donde se podrá observar de forma general la forma en que se llevan a cabo cada una de las actividades hasta finalizar el procedimiento, esto toma mayor importancia considerando que mientras mejor se entienda el funcionamiento del sistema actual, mejor orientado estará el diseño del sistema que se desea crear.

Tabla I. Proceso de evaluación del EPS

No	Actividad	Descripción	Encargado
1	Solicitud de EPS	El estudiante solicita realizar su EPS en la coordinación del EPS	Estudiante
2	Aprobación o Rechazo	Se aprueba o rechaza la solicitud de EPS	Coordinación
3	Notificación de EPS asignado	Se notifica al estudiante la asignación y este inicia con el EPS	Estudiante
4	Evaluación primera fase	Se evalúa el trabajo del estudiante en la fase I	Supervisor

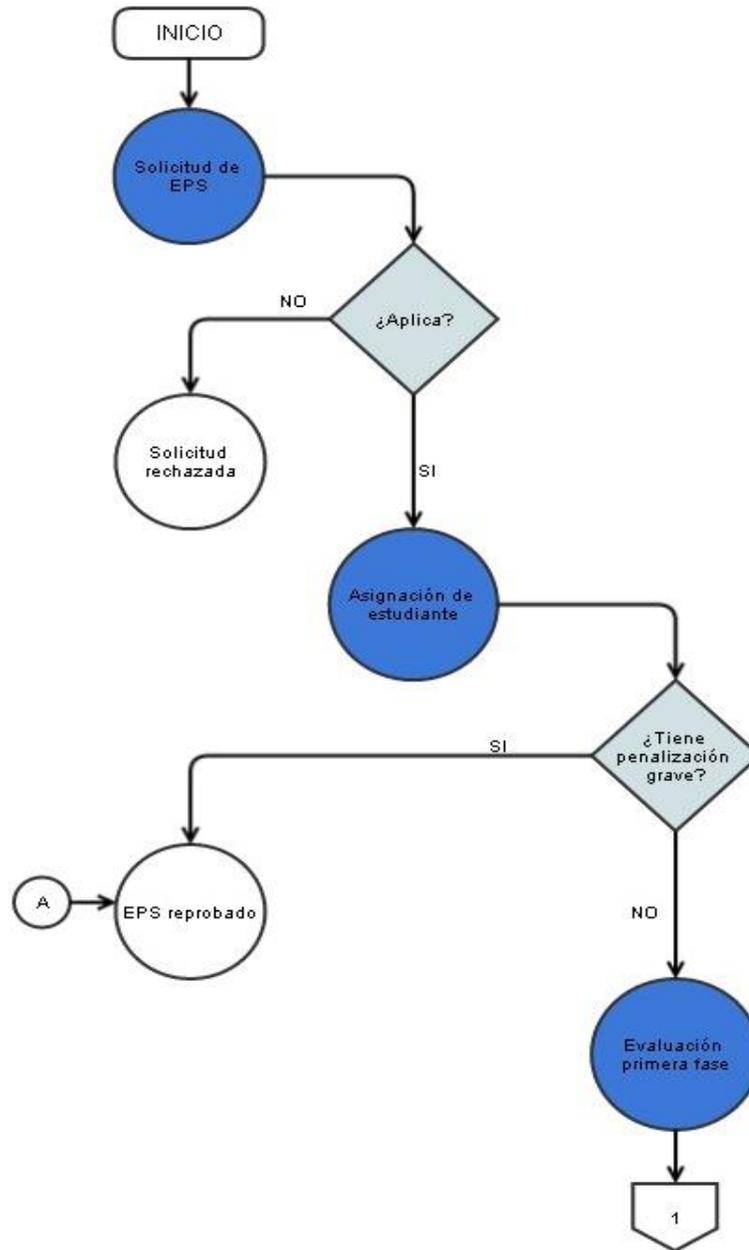
Continuación de la tabla I.

<b>No</b>	<b>Actividad</b>	<b>Descripción</b>	<b>Encargado</b>
5	Evaluación segunda fase	Se evalúa el trabajo del estudiante en la fase II	Supervisor
6	Evaluación tercera fase	Se evalúa el trabajo del estudiante en la fase III	Supervisor
7	Notificación a control académico	Se recibe la evaluación aprobada o reprobada del estudiante según corresponda para su trámite en control académico	Control académico

Fuente: elaboración propia.

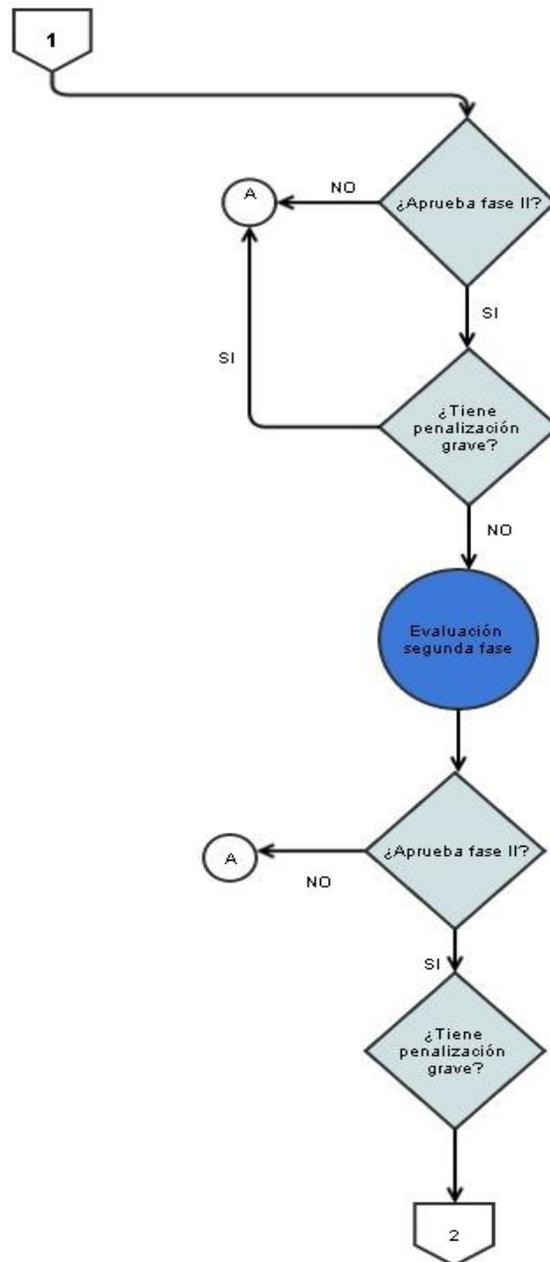
La entrada general al sistema es la aprobación y asignación del estudiante en el período correspondiente, mientras que la salida final es la evaluación del estudiante, la cual es una salida binaria, siendo sus valores Aprobado o Reprobado.

Figura 6. Diagrama de flujo del proceso - página 1



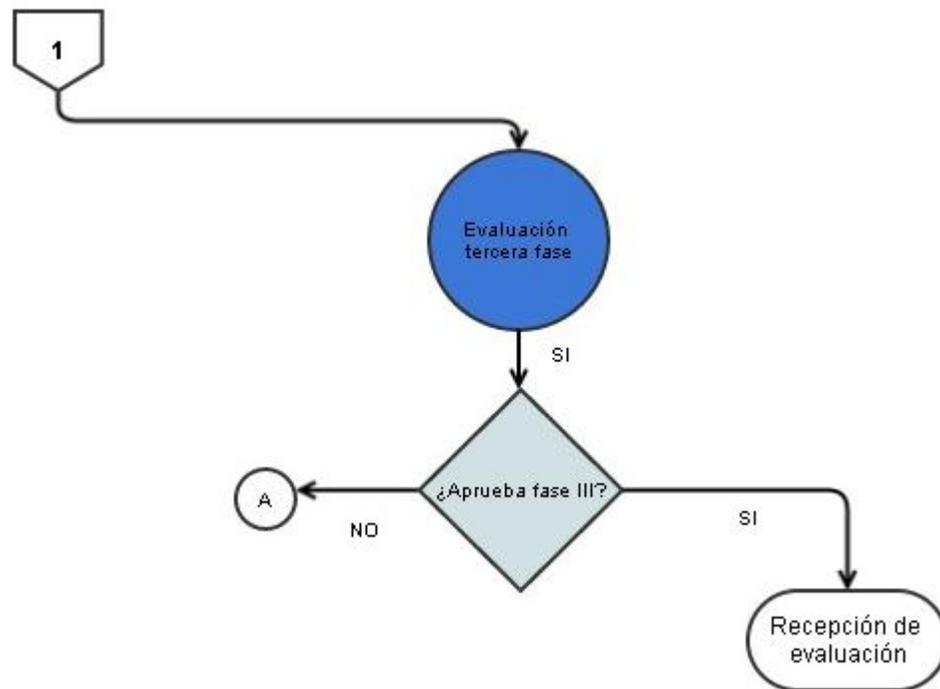
Fuente: elaboración propia.

Figura 7. Diagrama de flujo del proceso - página 2



Fuente: elaboración propia.

Figura 8. Diagrama de flujo del proceso - página 3



Fuente: elaboración propia.

## 2.2. Requerimientos del sistema

Entender lo que se espera de una aplicación es indispensable para el éxito o fracaso de la misma. Una aplicación cuyos requerimientos no están claros o se entienden de manera distinta a como el usuario final los transmite pueden ocasionar serios problemas en fases posteriores, por lo que una definición de requerimientos se hace de suma importancia para poder generar un producto adecuado y que cumpla los objetivos que se pretenden al desarrollar la aplicación.

### **2.2.1. Descripción general del sistema**

El Área Integrada y EPS de la Facultad de Agronomía de la Universidad de San Carlos de Guatemala, es una unidad que al igual que todas las áreas de cualquier facultad universitaria, necesita de un sistema que permita controlar de una mejor forma la evaluación y promoción de estudiantes que cursan esta fase al final de la carrera universitaria, y que permita evolucionar en conjunto con los avances tecnológicos.

Actualmente el EPS de Agronomía no tiene ninguna aplicación que permita tener un control mínimo centralizado en una base de datos de las notas de evaluación para los estudiantes y tampoco un medio que permita a los profesores tener una plataforma de comunicación eficiente entre ellos y los estudiantes que se asignan bajo su supervisión en cada ciclo de trabajo. Debido a esto, se ha realizado una aplicación de tipo Web, que permitirá cubrir estas necesidades básicas de control y comunicación entre profesores y estudiantes.

Después de la definición del proceso general de evaluación (ver sección 2.1), se procedió a realizar la toma de requerimientos y definir exactamente las necesidades urgentes que tiene el Área para establecer el alcance del proyecto y delimitarlo adecuadamente y así poder cumplir con lo que los usuarios del sistema necesitan de esta aplicación.

#### **2.2.1.1. Forma de evaluación del EPS de la Facultad de Agronomía**

El área de EPS de Agronomía evalúa a sus estudiantes durante cualquiera de los 2 ciclos que inician en un año calendario, teniendo el primer ciclo comenzando en el mes de febrero y finalizando en el mes de noviembre del

mismo año en que se inició, mientras que el otro ciclo inicia en agosto y finaliza al año siguiente en el mes de mayo.

El proceso inicia cuando un estudiante solicita realizar su EPS, en donde se evalúa si cumple o no con los requerimientos necesarios para ser aceptado en el programa durante el ciclo actual, de cumplirlos, se procede a registrarlo en el listado de estudiantes que realizaran esta actividad académica.

Después de ser aceptado, el estudiante es asignado a un único catedrático supervisor durante el ciclo, a menos que suceda alguna situación inusual, y cada catedrático tiene a su cargo un número indefinido de estudiantes, los cuáles realizan el EPS en distintas instituciones en cualquier departamento de Guatemala.

En cada ciclo cada estudiante debe aprobar 3 fases distintas y al reprobar alguna de las 3 fases automáticamente el estudiante reprueba el EPS durante el presente ciclo, también se puede reprobar una fase (y por lo tanto el EPS), si se comenten faltas de indisciplina. Existen 2 tipos de faltas, leves y graves. Cuando se comete una falta grave, se reprueba automáticamente el EPS, y también si se acumulan 2 faltas leves, se convierten automáticamente en 1 grave.

Existen 2 tipos distintos de estudiantes que el sistema debe manejar, los estudiantes con punto de tesis aprobado previo a la realización del EPS y los que no lo tienen, en cada caso se utiliza un formulario de evaluación distinto.

El EPS se aprueba con una nota mínima de 61 puntos, pero la promoción final de cada estudiante se establece como Aprobado o Reprobado, tanto para cada fase como para el total final.

### 2.2.2. Características de los usuarios

En este sistema se determinaron tres roles de usuarios que harán uso del sistema, cada uno con funciones distintas dentro del área de EPS, el coordinador, el profesor y el estudiante, que vemos a mayor detalle en el cuadro siguiente:

Tabla II. **Características de los usuarios del sistema**

<b>Usuario</b>	<b>Características y responsabilidades</b>
Administrador	Usuario encargado de administrar los catálogos de la aplicación, tales como la administración de usuarios, períodos y asignaciones entre estudiantes y profesores. Tiene el mayor número de privilegios de acceso a la base de datos y es responsable por los cambios a la información contenida en la base de datos. Este rol está identificado en el diagrama de proceso como coordinación en el diagrama de flujo del proceso en la sección 2.1.
Profesor	Este usuario se encarga de administrar las notas correspondientes a los estudiantes asignados a su supervisión, tiene una limitada cantidad de privilegios de acceso a la base de datos, no puede realizar cambios sobre los catálogos de usuarios, períodos y asignaciones. Este rol corresponde a la entidad supervisor, identificado en el diagrama de flujo del proceso de la sección 2.1.

Continuación de la tabla II.

Usuario	Características y responsabilidades
Estudiante	<p>Usuario con menos privilegios, básicamente tiene acceso de lectura a las notas que le corresponden y al envío y recepción de mensajes entre él y su profesor asignado, no puede hacer ningún tipo de modificación a la base de datos del sistema.</p> <p>Existen 2 tipos de estudiantes, con punto de tesis aprobado y sin punto de tesis aprobado, cada uno con un formulario de evaluación distinto uno del otro. En el diagrama de flujo del proceso de la sección 2.1 se identifica a este rol con el mismo nombre, estudiante</p>

Fuente: elaboración propia.

### 2.2.3. Restricciones

Para el desarrollo de este sistema hay una limitación principal discutida con los interesados del proyecto en cuanto a recursos de hardware. Al momento no se cuenta con ningún recurso en donde se pueda implementar el sistema. Actualmente el EPS de Agronomía no cuenta con ningún medio físico para establecer un servidor para la aplicación, así como tampoco un nombre de dominio ni algún tipo de servicio para el alojamiento de este sistema.

Debido a esta restricción, el diseño y la elección de herramientas de desarrollo se ve afectada directamente por esto, ya que se debe tomar en cuenta que el sistema deberá ser flexible en cuanto a independencia de plataformas.

#### **2.2.4. Evolución a futuro del sistema**

Durante las reuniones realizadas para la toma de requerimientos, se estableció el alcance del proyecto, pero también se establecieron diversos parámetros que se deben tomar en cuenta ya que se pretende que este sistema evolucione con el tiempo y se pueda expandir para abarcar más necesidades del área y conjuntarse con otros sistemas.

Como primer aspecto, se estableció que se desea tener un sistema flexible debido a que en el Área de EPS de Agronomía actualmente se encuentran trabajando en cambios al sistema de evaluación actual; también se está discutiendo el cambio de ciertas políticas actuales en el área, lo que afectaría directamente a la estructura del sistema que se desarrolló, por lo que el sistema se espera cambie y se introduzcan nuevas reglas al mismo a mediano plazo y se debe tomar en cuenta en el diseño del mismo.

El segundo aspecto importante es la integración con el sistema central de notas de la Facultad de Agronomía que actualmente se encuentra en uso, ya que se pretende que en un futuro, el sistema del área de EPS pueda interactuar con este otro sistema, de manera que el envío de notas finales sea realizada por este medio, y no como se hace actualmente en papel impreso. Aunque esto queda fuera del alcance del proyecto, se tomó en cuenta para el diseño de la aplicación. Dicha interacción quedó definida en el diagrama de flujo de proceso (sección 2.1) donde se puede observar que al final del flujo se traslada la evaluación final hacia la entidad control académico.

### 2.2.5. Requerimientos funcionales

El sistema desarrollado cumple con los siguientes requerimientos indispensables definidos en conjunto con los interesados del proyecto, dichos requerimientos se listan a continuación con su respectiva descripción y el usuario el cual utilizará cada funcionalidad dentro del sistema.

Tabla III. **Requerimientos funcionales**

Requerimiento	Descripción	Usuario
R1	El sistema debe poder crear, buscar y modificar cualquier tipo de usuarios del sistema.	Administrador
R2	El sistema debe poder crear, buscar y modificar períodos de evaluación de EPS.	Administrador
R3	El sistema debe poder crear y modificar asignaciones entre estudiantes y un profesor encargado, una asignación por estudiante en cada período deseado.	Administrador
R4	El sistema debe verificar la identidad del usuario que desea utilizar el sistema a través de un sistema de autenticación por nombre de usuario y contraseña única por usuario.	Todos los usuarios
R5	El sistema debe verificar que se muestren las páginas correspondientes al rol del usuario que hace uso del sistema.	Todos los usuarios
R6	El sistema puede crear, modificar y eliminar avisos en la sección inicial del sistema para visualización de cualquier usuario, autenticado o no dentro del sistema.	Administrador, profesor

Continuación de la tabla III.

Requerimiento	Descripción	Usuario
R7	El sistema debe ser capaz de generar una contraseña aleatoria nueva cuando el usuario solicite el cambio y sea enviada al correo de este usuario.	Todos los usuarios
R8	El sistema debe poder ingresar, modificar y guardar notas de evaluación por cada una de las 3 fases que componen el ciclo de EPS.	Profesor
R9	El sistema debe tener un foro de comunicación entre los profesores y los estudiantes asignados a su supervisión que permita mandar y recibir mensajes visibles tanto en la aplicación como en el correo de cada usuario.	Profesor, estudiante
R10	El sistema debe permitir enviar mensajes del profesor al estudiante cuando se le penalice por alguna falta cometida, y debe poder reprobar a un estudiante automáticamente al incumplir las políticas de penalizaciones del área de EPS.	Profesor, estudiante
R11	El sistema debe poder mostrar las notas de cada estudiante dependiendo del tipo del mismo y el período elegido.	Estudiante
R12	El sistema debe verificar que el estudiante que repruebe alguna de las 3 fases del ciclo de EPS sea automáticamente reprobado.	Profesor, estudiante
R13	El sistema debe poder guardar el número actual de la oportunidad en la que los estudiantes sin punto de tesis aprobado, durante la fase 3, entreguen el informe escrito correspondiente a dicha fase.	Profesor

Continuación de la tabla III.

<b>Requerimiento</b>	<b>Descripción</b>	<b>Usuario</b>
R14	El sistema debe emitir reportes en formato web a convenir, de la información para cada usuario del sistema.	Todos los usuarios
R15	El sistema debe guardar toda la información de un usuario, siendo el identificador para los usuarios, el número de carné para los estudiantes y el número de registro de personal del administrador o del profesor.	Todos los usuarios
R16	El sistema no debe ser capaz de eliminar a ningún usuario, período o asignación de la base de datos.	Todos los usuarios

Fuente: elaboración propia.

## **2.2.6. Requerimientos no funcionales**

Los requerimientos no funcionales, son también muy importantes en la definición de un sistema. Usualmente están ligados a aspectos que no son perceptibles inmediatamente por el usuario pero que pueden serlo conforme se va utilizando la solución.

### **2.2.6.1. Requerimientos de rendimiento**

En el área de EPS por cada ciclo de evaluación la cantidad de estudiantes y profesores es baja, aproximadamente 25 estudiantes por ciclo y 10 profesores en promedio. Debido a esto la carga esperada de conexiones y accesos a la base de datos es muy baja, no solo por la cantidad de usuarios sino también porque cada estudiante se encontraría localizado en áreas rurales de

Guatemala donde actualmente no se cuenta con acceso a internet o se cuenta con un acceso muy limitado.

En el peor de los casos se espera un promedio de conexiones concurrentes al sistema de:

- 25 estudiantes + 10 profesores + 1 administrador = 36 terminales conectadas.

El sistema debe ser capaz de soportar como mínimo esta carga de 36 conexiones en dado caso sean concurrentes.

#### **2.2.6.2. Requerimientos de seguridad**

El sistema debe manejar la seguridad básica de un sistema Web, ya que este aspecto es indispensable hoy en día para garantizar la integridad de la información, en este desarrollo se tomaron en cuenta aspectos como:

- Encriptación de contraseñas de usuarios
- Privilegios de acceso a tablas de la base de datos
- Ocultamiento de las contraseñas de roles de usuarios para acceso a la base de datos

Debido a que no se ha definido en donde se alojará la aplicación, la seguridad de acceso en la Web como firewalls y restricciones de acceso en el servidor Web, quedan fuera del alcance del proyecto y estará sujeta a la elección final del destino de la aplicación.

### **2.2.6.3. Requerimientos de mantenimiento**

El mantenimiento quedará a cargo del usuario administrador del sistema, pero en el caso de extensión de las funcionalidades y reestructuración del sistema en general, se debe requerir a un desarrollador experto para llevar a cabo dichas tareas y a un administrador de base datos para realizar los mantenimientos a la misma. Esto queda fuera del alcance de este proyecto y queda a discreción de los interesados del proyecto la designación de dicha persona o personas encargadas de realizar estas tareas de mantenimiento del sistema.

### **2.2.6.4. Requerimientos de interfaz de usuario**

La interfaz de usuario es la parte que verán los clientes, la parte en donde ellos interactuarán con la base de datos, por eso es esencial un diseño adecuado para las necesidades y capacidades de los usuarios.

El requerimiento principal discutido con los interesados del proyecto fue simpleza en el diseño, sin sobrecarga de *banners*, animaciones o adornos, con el logotipo de la facultad de Agronomía y del área de EPS en la parte superior de la página.

La interfaz debe ser intuitiva y con una curva de aprendizaje pequeña, para que los usuarios finales se adapten a ella en un corto tiempo.

## **2.3. Casos de uso**

Cada caso de uso representa una funcionalidad del sistema desarrollado, en las siguientes tablas se describen los casos de uso para el sistema

desarrollado, que captura toda la funcionalidad deseada por parte de los interesados del proyecto.

Tabla IV. **Ingreso de usuarios**

<b>C-01</b>	Ingreso y autenticación de usuarios	
<b>Descripción</b>	El sistema debe autenticar por medio de un nombre de usuario y una contraseña a los usuarios que deseen hacer uso de la aplicación.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario escribe el nombre de usuario, contraseña y elige el tipo de usuario correspondiente.
	2	El usuario oprime el botón de ingreso al sistema.
		2a Si la autenticación falla se muestra una página de error informando al usuario sobre el mismo-
		2b Si la autenticación es correcta se muestra la página inicial correspondiente al tipo de usuario.
3	Se termina la sesión en el sistema por decisión del usuario o por tiempo de inactividad.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si falla la conexión a la base de datos se muestra una página informando al usuario del error y se recomienda intentar de nuevo el paso 1.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla V. **Administración de usuario**

<b>C-02</b>	Administración de catálogo de usuarios
<b>Descripción</b>	El sistema debe permitir al administrador crear, buscar o modificar cualquier usuario en la base de datos

Continuación de la tabla V.

<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador ingresa al sistema, realizar caso C-01
	2	El administrador elige la opción de administración de usuarios
	3	El administrador elige la opción deseada (crear o buscar).
	3a	Si se elige la opción de creación de usuario, se muestra la página para la creación de nuevos usuarios, realizar el caso C-03
	3b	Si elige la opción de buscar se muestra la página de búsqueda de usuarios, realizar el caso C-04
	4	Se termina la sesión del administrador
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla VI. **Creación de usuarios**

<b>C-03</b>	Creación de nuevos usuarios.	
<b>Descripción</b>	El sistema debe permitir al administrador crear usuarios de cualquier rol dentro del sistema.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de creación de nuevo usuario, realizar caso C-02.
	2	El administrador elige el rol del usuario a crear.
	3	El administrador ingresa los datos del usuario a crear.
	4	El administrador oprime el botón de creación de usuario.
	5	Se muestra un mensaje de éxito al crear el nuevo usuario.

Continuación de la tabla VI.

<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario ingresa información errónea se muestra un mensaje en la página informando del parámetro ingresado que produjo el error.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla VII. **Búsqueda y modificación de usuarios**

<b>C-04</b>	Búsqueda y modificación de usuarios ya creados.	
<b>Descripción</b>	El sistema debe permitir al administrador buscar usuarios dentro del sistema para su modificación o visualización de la información guardada de los mismos.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de búsqueda de usuario, realizar caso C-02.
	2	El administrador elige el rol del usuario a buscar.
	3	El administrador escribe los parámetros de búsqueda del usuario.
	4	El administrador oprime el botón de búsqueda.
	5	Se muestra la página de resultados de la búsqueda con las posibles opciones.
	6	El administrador elige uno de los resultados.
	7	Se muestran los datos del usuario elegido modificables.
	8	El administrador modifica la información deseada del usuario elegido.
	8a	Si el usuario oprime el botón para modificar usuario, se guarda la información modificada del usuario.

Continuación de la tabla VII.

		8b	Si el usuario oprime el botón para generar una nueva contraseña, se envía la contraseña al correo del usuario.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	5	Si no hay resultados se muestra una página advirtiendo que no se encontró coincidencia en la base de datos de la búsqueda realizada.	
	8a	Si se modifica la información del usuario con datos erróneos, se muestra un mensaje de error en la página mostrando el parámetro que generó el error.	
<b>Importancia</b>	Media		
<b>Comentario</b>	La contraseña de cada usuario no se muestra en la información del usuario elegido en la búsqueda.		

Fuente: elaboración propia.

Tabla VIII. **Administración de períodos**

<b>C-05</b>	Administración de los períodos del ciclo evaluación.		
<b>Descripción</b>	El sistema debe permitir al administrador crear y modificar períodos para cada ciclo de EPS.		
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El administrador ingresa al sistema, realizar caso C-01.	
	2	El administrador elige la opción de administración de períodos.	
		2a	Si elige la opción de nuevo período, realizar caso C-06.
		2b	Si elige la opción de buscar período, realizar el caso C-07.

Continuación de la tabla VIII.

	3	Se finaliza la sesión del administrador.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla IX. **Creación de períodos**

<b>C-06</b>	Creación de nuevos períodos de evaluación.	
<b>Descripción</b>	El sistema debe permitir al administrador crear nuevos períodos de evaluación.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de nuevo período, realizar caso C-05.
	2	El administrador ingresa el año de inicio y el año de finalización del período.
	3	Se muestra un mensaje de éxito en la creación del nuevo período.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si ingresa años incorrectos o con una diferencia de más de 1 año entre el de inicio y el de finalización se muestra un mensaje de error para informar al usuario.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla X. **Búsqueda y modificación de períodos**

<b>C-07</b>	Búsqueda y modificación de períodos de evaluación.	
<b>Descripción</b>	El sistema debe permitir al administrador buscar períodos en la base de datos para su modificación.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de buscar período, realizar caso C-05.
	2	El administrador ingresa los parámetros de búsqueda del período deseado.
	3	Se muestran los posibles resultados.
	4	Se elige un resultado de las coincidencias de búsqueda.
	5	Se modifica el período elegido.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5	Si ingresa años incorrectos o con una diferencia de más de 1 año entre el de inicio y el de finalización se muestra un mensaje de error para informar al usuario.
<b>Importancia</b>	Media	

Fuente: elaboración propia.

Tabla XI. **Administración de asignaciones**

<b>C-08</b>	Administración de las asignaciones.	
<b>Descripción</b>	El sistema debe permitir al administrador crear y modificar asignaciones entre estudiantes y profesores para un período específico.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador ingresa al sistema, realizar caso C-01.
	2	El administrador elige la opción de administración de asignaciones.

Continuación de la tabla XI.

		2a	Si elige la opción de nueva asignación, realizar caso C-09.
		2b	Si elige la opción de buscar asignación realizar el caso C-10.
	3	Se finaliza la sesión del administrador.	
<b>Importancia</b>	Alta		

Fuente: elaboración propia.

Tabla XII. **Creación de asignaciones**

<b>C-09</b>	Creación de una nueva asignación.	
<b>Descripción</b>	El sistema debe permitir al administrador crear una asignación entre un estudiante y un profesor, previamente creados en la base de datos.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de nueva asignación, realizar caso C-08.
	2	El administrador ingresa la información solicitada para la asignación.
	3	Se muestra un mensaje de éxito en la asignación.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si ingresa datos incorrectos se muestra un mensaje de error en la página informando el parámetro que causó el error.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla XIII. **Búsqueda y modificación de asignaciones**

<b>C-10</b>	Búsqueda y modificación de asignaciones	
<b>Descripción</b>	El sistema debe permitir al administrador buscar asignaciones en la base de datos para su modificación.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El administrador elige la opción de buscar asignación, realizar caso C-08.
	2	El administrador ingresa los parámetros de búsqueda de la asignación deseada.
	3	Se muestran los posibles resultados.
	4	Se elige un resultado de las coincidencias de búsqueda.
	5	Se modifican los datos de la asignación elegida.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5	Si ingresa datos incorrectos se muestra un mensaje de error en la página informando el parámetro que causó el error.
<b>Importancia</b>	Media	

Fuente: elaboración propia.

Tabla XIV. **Administración de notas y penalizaciones**

<b>C-11</b>	Administración de las notas de evaluación y penalizaciones	
<b>Descripción</b>	El sistema debe permitir al profesor ingresar y modificar las notas y enviar penalizaciones por faltas cometidas durante el EPS al estudiante bajo su supervisión.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor ingresa al sistema, realizar caso C-01.
	2	El profesor elige la opción de administración de notas.

Continuación de la tabla XIV.

		2a	Si elige la opción de ingreso de notas, realizar caso C-12.
		2b	Si elige la opción de penalización realizar el caso C-13.
	3	Se finaliza la sesión del profesor.	
<b>Importancia</b>	Alta		

Fuente: elaboración propia.

Tabla XV. **Ingreso de notas**

<b>C-12</b>	Ingreso de notas	
<b>Descripción</b>	El sistema debe permitir al profesor ingresar o modificar las notas correspondientes a cualquiera de los estudiantes bajo su supervisión en un período elegido.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor elige la opción de ingreso de notas, realizar caso C-11
	2	El profesor elige al estudiante y el período de asignación.
	3	El profesor elige la fase en donde se desea ingresar o modificar notas.
	4	El profesor ingresa o modifica las notas en el formulario.
	5	El profesor oprime el botón para guardar las notas y se muestra un mensaje de éxito.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el profesor elige una fase habiendo reprobado ya el estudiante la fase previa, se inhabilita el ingreso o modificación de notas en ese formulario.

Continuación de la tabla XV.

	4	Si el profesor ingresa notas fuera del rango posible, se debe mostrar un mensaje de error en la página.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla XVI. **Ingreso de penalizaciones**

<b>C-13</b>	Ingreso de penalizaciones	
<b>Descripción</b>	El sistema debe permitir al profesor enviar mensajes de penalizaciones al estudiante y guardarlas en la base de datos con la descripción de la misma.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor elige la opción de penalizaciones, realizar caso C-11.
	2	El profesor elige el estudiante y el período de asignación.
	3	El profesor elige el tipo de penalización y su descripción.
	4	El profesor envía la penalización al estudiante y a la base de datos.
<b>Excepción</b>	<b>Paso</b>	<b>Acción</b>
	4	Si el estudiante acumula las penalizaciones necesarias para reprobado el EPS, se modifica el estado del estudiante a REPROBADO.
<b>Importancia</b>	Baja	

Fuente: elaboración propia.

Tabla XVII. **Administración de foro de mensajes**

<b>C-14</b>	Administración del foro de mensajes	
<b>Descripción</b>	El sistema debe permitir al profesor y al estudiante leer y enviar mensajes entre ellos.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al sistema, realizar caso C-01.
	2	El usuario elige la opción de foro de mensajes.
	2a	Si elige la opción de bandeja de entrada, realizar el caso C-15.
	2b	Si elige la opción de redactar mensaje, realizar el caso C-16.
3	Se finaliza la sesión del usuario.	
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla XVIII. **Lectura de mensajes del foro**

<b>C-15</b>	Lectura de mensajes en el foro	
<b>Descripción</b>	El sistema debe permitir al profesor y estudiante leer los mensajes enviados entre ellos en un período elegido	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario elige la opción de bandeja de entrada, realizar caso C-14.
	2	El usuario elige el período.
	3	Se muestran los mensajes.
<b>Importancia</b>	Media	

Fuente: elaboración propia.

Tabla XIX. **Redactar mensajes en el foro**

<b>C-16</b>	Redactar mensajes en el foro	
<b>Descripción</b>	El sistema debe permitir al profesor y estudiante redactar mensajes para su envío entre ellos	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario elige la opción de redactar mensaje, realizar caso C-14.
	2	El usuario elige el período.
	3	Se ingresa el título y el mensaje y se envía al correo del destinatario y a la base de datos.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si no se ingresa alguno de los 2 parámetros del mensaje, se debe mostrar un error en la página solicitando se llene la información necesaria para el envío del mensaje.
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla XX. **Ver notas de estudiante**

<b>C-17</b>	Ver notas de estudiante	
<b>Descripción</b>	El sistema debe permitir al estudiante ver sus notas en la aplicación en cada fase detalladamente.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante elige la opción de ingreso al sistema, realizar caso C-01.
	2	El estudiante elige el período de asignación.

Fuente: elaboración propia.

Tabla XXI. **Modificar datos de usuario**

<b>C-18</b>	Modificar datos de usuario	
<b>Descripción</b>	El sistema debe permitir al administrador, profesor y estudiante modificar sus datos personales.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al sistema, realizar caso C-01.
	2	El usuario elige la opción modificar datos.
	2a	Si elige la opción de modificar datos personales, realizar caso C-19.
	2b	Si elige la opción de modificar contraseña, realizar caso C-20.
3	El usuario finaliza sesión.	
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

Tabla XXII. **Modificar datos personales de usuario**

<b>C-19</b>	Modificar datos personales de usuario.	
<b>Descripción</b>	El sistema debe permitir al administrador, profesor y estudiante modificar sus datos personales.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario elige la opción de modificar datos personales.
	2	El usuario modifica sus datos.
	3	Se guarda la información en la base de datos y se muestra un mensaje de éxito.

Continuación de la tabla XXII.

<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si se ingresa información incorrecta se muestra un mensaje indicando el parámetro que provocó el error.
<b>Importancia</b>	Baja	

Fuente: elaboración propia.

Tabla XXIII. **Modificar contraseña de usuario**

<b>C-20</b>	Modificar la contraseña de usuario	
<b>Descripción</b>	El sistema debe permitir al administrador, profesor y estudiante modificar su contraseña de acceso al sistema.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario elige la opción de modificar contraseña.
	2	El usuario ingresa su antigua contraseña y la nueva.
	3	Se envía la contraseña nueva al correo del usuario y se muestra un mensaje de éxito.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el usuario ingresa una contraseña anterior incorrecta o una contraseña inválida se muestra un mensaje de error en la página.
<b>Importancia</b>	Media	

Fuente: elaboración propia.

Tabla XXIV. **Generación de reportes**

<b>C-21</b>	Generar reportes	
<b>Descripción</b>	El sistema debe permitir al administrador y profesor generar distintos reportes basados en la información almacenada den la base de datos.	
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario ingresa al sistema, realizar caso C-01.
	2	El usuario elige la opción de reportes.
	3	El usuario elige el reporte que desea ver.
	4	El sistema muestra el reporte en formato PDF.
5	El usuario finaliza sesión.	
<b>Importancia</b>	Alta	

Fuente: elaboración propia.

### 2.3.1. **Casos de uso y el proceso general de evaluación**

Cada caso de uso de la sección anterior, se encuentra encapsulado dentro de una actividad del proceso general descrito en la sección 2.1, es importante conocer dentro de qué actividades se enmarcan los casos de uso, ya que estos casos se deben apegar al proceso general de evaluación del EPS y deben realizar funciones que permitan ejecutar cada actividad descrita en este proceso. A continuación se puede observar dicha correspondencia en la tabla siguiente.

Tabla XXV. **Relación de casos de uso con el proceso general**

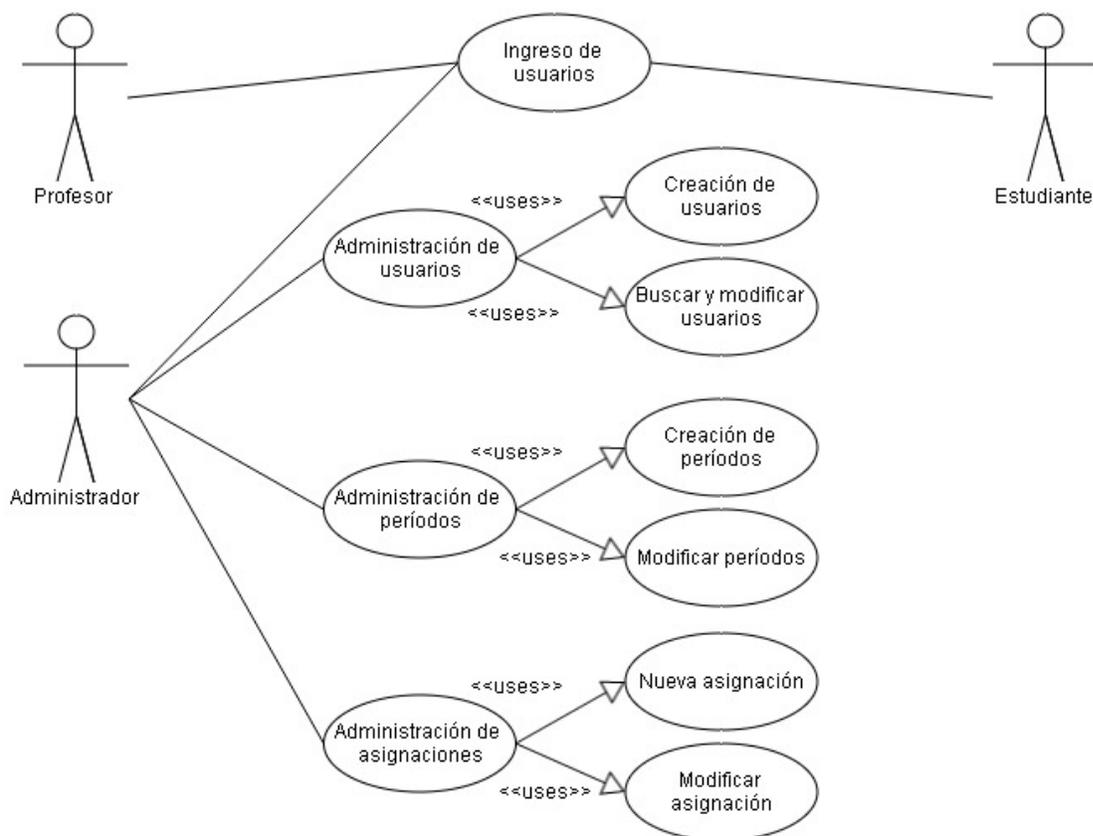
<b>Caso de uso</b>	<b>Actividades en el proceso (Tabla I)</b>
C-01. Ingreso y autenticación de usuarios	Actividades 3-6
C-02. Administración de catálogo de usuarios	Actividades 3-6
C-03. Creación de nuevos usuarios	Actividad 3
C-04. Búsqueda y modificación de usuarios ya creados	Actividad 3
C-05. Administración de los períodos de evaluación	Actividad 3
C-06. Creación de nuevos períodos de evaluación	Actividad 3
C-07. Búsqueda y modificación de períodos de evaluación	Actividad 3
C-08. Administración de las asignaciones	Actividad 3
C-09. Creación de una nueva asignación	Actividad 3
C-10. Búsqueda y modificación de asignaciones	Actividad 3
C-11. Administración de las notas de evaluación y penalizaciones	Actividades 4-6
C-12. Ingreso de notas	Actividades 4-6
C-13. Ingreso de penalizaciones	Actividades 4-6
C-14. Administración del foro de mensajes	Actividades 4-6
C-15. Lectura de mensajes en el foro	Actividades 4-6
C-16. Redactar mensajes en el foro	Actividades 4-6
C-17. Ver notas de estudiante	Actividades 4-6
C-18. Modificar datos de usuario	Actividad 3
C-19. Modificar datos personales de usuario	Actividades 3-6
C-20. Modificar la contraseña de usuario	Actividades 3-6
C-21. Generar reportes	Actividades 3-6

Fuente: elaboración propia.

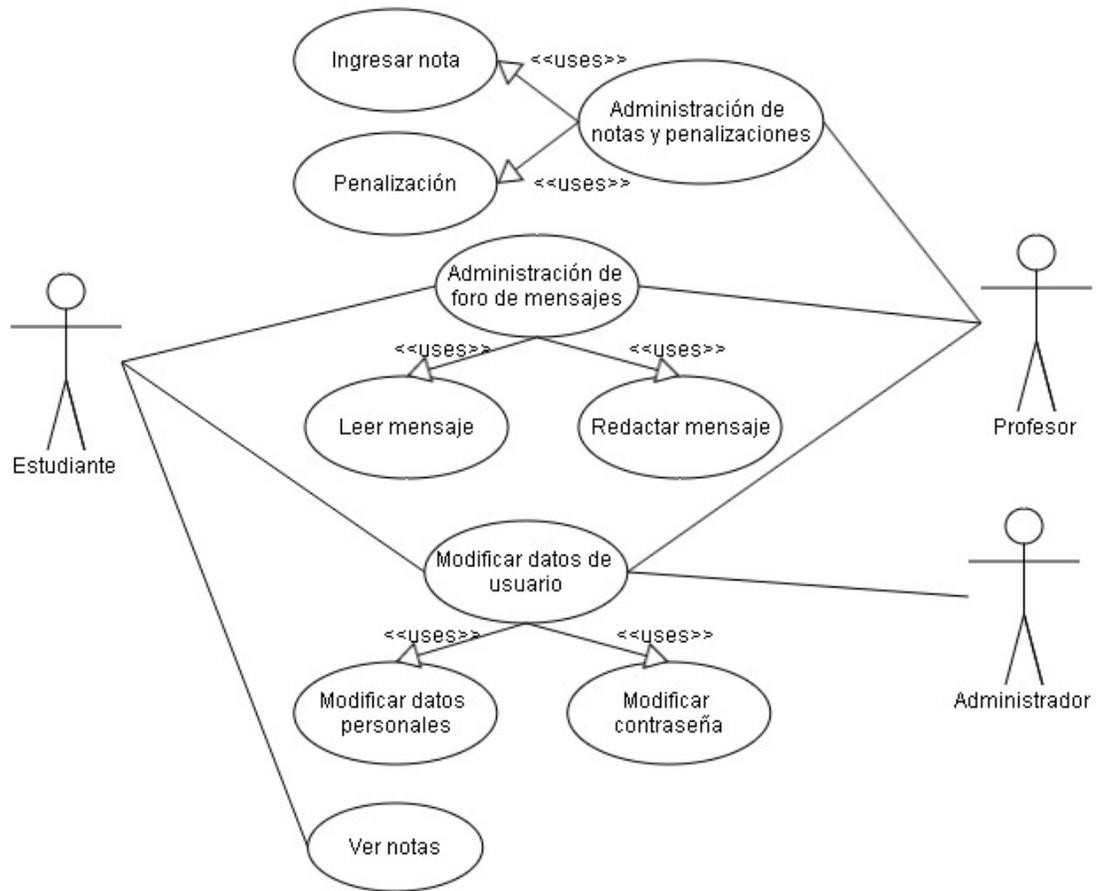
### 2.3.2. Diagrama de casos de uso

La notación de casos de uso, es un estándar utilizado para mostrar tanto al usuario como el desarrollador, la interacción entre los distintos actores que harán uso del sistema, y este.

Figura 9. Diagrama de casos de uso del sistema



Continuación de la figura 9.



Fuente: elaboración propia.



### **3. ANÁLISIS Y DISEÑO DE LA APLICACIÓN**

Después de definir los requerimientos y tener claro el proceso que desea cubrirse con la aplicación, se debe analizar dichos requerimientos y empezar con el diseño del sistema.

#### **3.1. Metodología de desarrollo**

La elección de la metodología para desarrollar este proyecto no se apega a una solamente, sino a un conjunto de varias metodologías que permitieron tener flexibilidad y al mismo tiempo mantener un orden efectivo en la realización de todas las tareas de análisis, diseño y desarrollo del sistema. En el capítulo 1 se pueden ver las distintas metodologías tomadas en cuenta para desarrollar este proyecto, en esta parte se mostrarán las partes tomadas de cada una de esas metodologías y el diagrama de la metodología resultante con la cual se trabajó durante el transcurso de todo el proyecto.

Para el desarrollo de este sistema, se tomaron 3 partes fundamentales el ciclo de vida del software del modelo en cascada:

- Toma de requerimientos
- Análisis y diseño de la aplicación
- Desarrollo y pruebas

Ya que este enfoque nos da un orden pero no provee de la flexibilidad necesaria para realizar cambios y evolucionar el software de manera que los requerimientos se vayan cumpliendo progresivamente, el enfoque del modelo

iterativo incremental en conjunto con el modelo en espiral, dieron esta flexibilidad, que en conjunto con el orden establecido con las partes sustraídas del modelo en cascada, permitieron definir un nuevo modelo adaptable a las necesidades de este proyecto y a su vez permitió aprovechar la interacción constante con los usuarios finales que se tuvo durante el transcurso del desarrollo, desde su inicio hasta su entrega final.

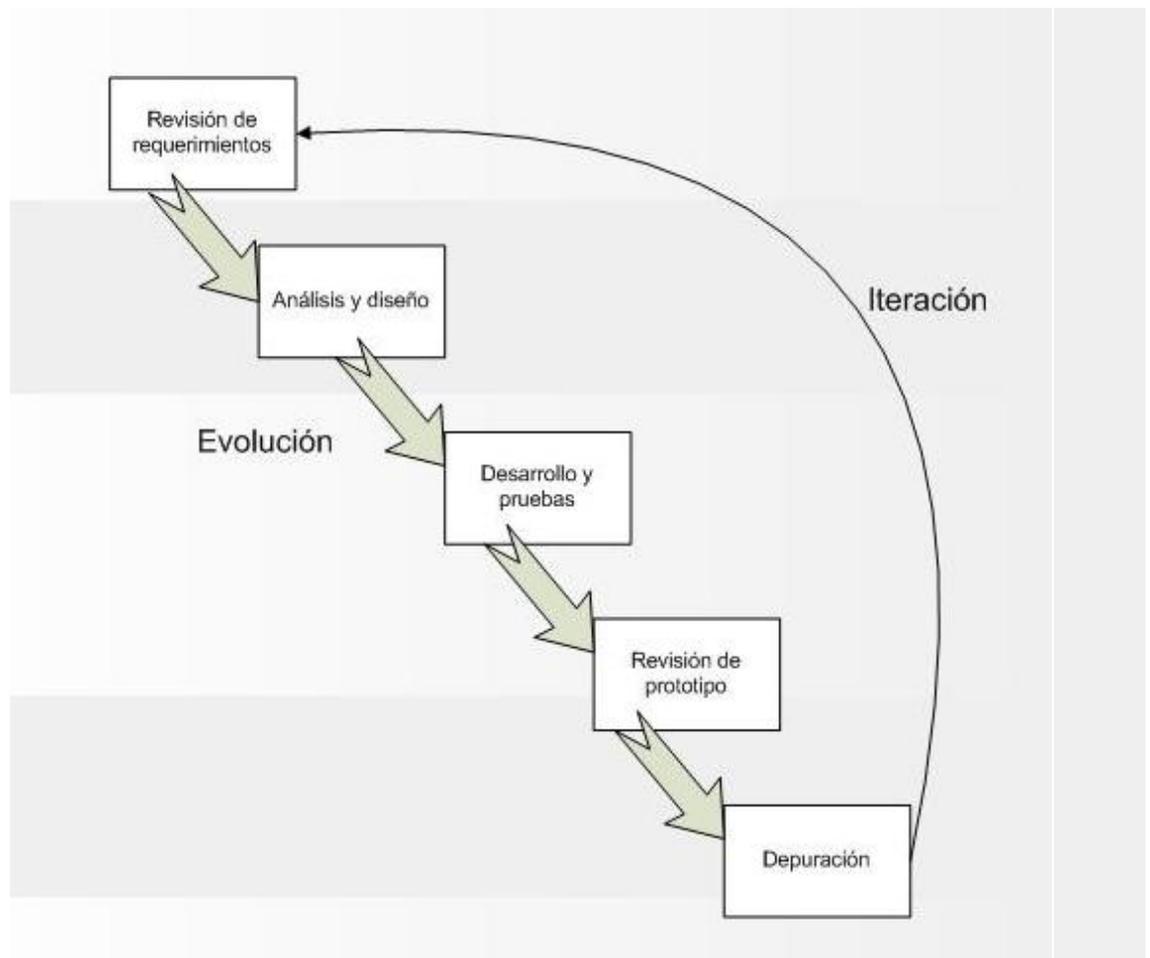
Tomando la evolución en espiral con las 3 partes centrales definidas anteriormente, se creó un modelo híbrido donde se hicieron varias iteraciones, en cada iteración se elaboró una revisión de los requerimientos, del análisis y el diseño establecido. De esta manera se pudo, en conjunto con los usuarios finales, ajustar al final de cada iteración los puntos en los cuáles los requerimientos no estaban siendo cumplidos en su totalidad, presentando prototipos funcionales que fueron aprobados y así dieron paso a una nueva iteración, reduciendo así el riesgo de atrasos por cambios profundos o posibles errores en la toma de requerimientos.

Al hacer diversas iteraciones con revisiones al final de cada iteración se logró que los usuarios finales tuvieran una idea más clara de lo que querían en cuanto a funcionalidades y diseño de la interfaz de usuario y evitar realizar cambios a esto en puntos más avanzados del proyecto que obligarían a alterar casi en su totalidad todo lo realizado anteriormente.

En la siguiente gráfica se muestra de forma general y simple, como se unieron estos modelos de desarrollo de software en uno solo que permitió realizar un proyecto que cumpliera con los requerimientos establecidos. Ahí se observa como la base del modelo en cascada modificado, se une a las iteraciones evolutivas de estos otros dos modelos para lograr este modelo que si bien es mucho más simple y deja afuera a muchos elementos que toman en

cuenta cada uno de los modelos anteriores, se ajusta a las necesidades de este proyecto, que por ser un proyecto de baja envergadura, no necesita de todas esas partes involucradas en cada modelo, ya que no permitirían un desarrollo más ágil y preciso, si se apega el desarrollo estrictamente a los componentes de cada modelo.

Figura 10. **Diagrama de la metodología de desarrollo**



Fuente: elaboración propia.

Se debe anotar que en la primera iteración, la toma de requerimientos, el análisis y el diseño del proyecto ocupan una porción más grande de tiempo en la evolución, lo cual no se puede observar en la gráfica pero es importante tener en cuenta ya que en el inicio estas dos fases son críticas para que el proyecto evolucione de manera correcta. Durante el transcurso del resto de iteraciones, hasta la iteración final, la toma de requerimientos, el análisis y el diseño, consisten en revisiones con los usuarios finales para ajustes menores que no afectan lo ya desarrollado anteriormente pero sí a las fases siguientes.

### **3.2. Herramientas de desarrollo**

La elección de las herramientas de software para el desarrollo de la aplicación va ligada a las restricciones de hardware descritas en la sección 2.1.3, hay una limitante importante que se tomó en cuenta, siendo esta la falta de recursos de hardware y de software en el área de EPS adecuados para la instalación e implementación del proyecto.

No se cuenta con ninguna computadora para realizar las funciones de servidor, así como ningún nombre de dominio ni dirección ip pública para acceder desde internet a la aplicación, pero se manejan 3 distintas opciones que se definirán posteriormente por parte de los interesados del proyecto.

- Instalar la aplicación sobre una computadora a adquirir por parte del área de EPS y obtener una ip y dominio por parte del sistema central de la USAC.
- Contratar un servicio de *hosting* y dominio que permita la instalación y acceso al sistema.

- Instalar la aplicación en los servidores del centro de cálculo de la Facultad de Agronomía y obtener el acceso a través de su página central.

Debido a la restricción de estas opciones, se tomó en cuenta que las herramientas que se utilizarían pudieran ser lo suficientemente flexibles para poder adaptarse a cualquiera de estas opciones, además de que el área no cuenta con un presupuesto definido para pago de licencias de software.

Esto hizo que se eligieran herramientas *Open Source* tanto en lenguaje de programación como de administrador de base de datos y servidor Web, que fueran independientes de la plataforma, y pudieran, si se diera el caso, poder instalarse en sitios que proveen alojamiento y dominios para aplicaciones Web. Esto, principalmente, fue la causa de la elección de herramientas que se verán a continuación, ya que hay un gran número de sitios en internet que ofrecen soporte para estas herramientas.

Lenguajes de programación:

- PHP: se utilizó este lenguaje en el desarrollo de todo el código de la aplicación porque es ampliamente usado y soportado por los sitios que proveen servicios de alojamiento y dominios en internet, y permite trabajar con páginas dinámicas que interactúan con el cliente y el servidor Web de una manera sencilla.
  - Versión utilizada: PHP 5.3.5
- HTML: al ser esta una aplicación Web, HTML es una parte fundamental de la aplicación, que presenta la interfaz de usuario y permite la interacción en conjunto con PHP del cliente con el servidor.

- Versión utilizada: HTML 4.0

Servidor Web:

- Apache Server: al igual que PHP, este servidor Web es ampliamente utilizado y soportado en internet, tiene una gran flexibilidad y cumple con todas las funcionalidades, como soporte para PHP y también es adaptable a diversos sistemas operativos como Microsoft Windows o cualquier distribución de Linux, que se necesitan en este proyecto.}
- Versión utilizada: Httpd 2.2.17

Administrador de Base de Datos

- MySQL: Este sistema de base de datos es uno de los más utilizados por aplicaciones de baja escala y más soportados en internet, además de que el lenguaje PHP provee muchas funciones para interactuar con este RDBMS lo que facilitó el rápido desarrollo de la aplicación y permitió tener esa independencia de plataforma necesaria.
- Versión utilizada: MySQL 5.5.15

### **3.3. Arquitectura de software del proyecto**

La arquitectura es la parte fundamental del diseño de la aplicación, a partir de este punto, si se hace de manera correcta, podemos desarrollar una aplicación en la que hay menos probabilidad de incumplimiento con los requisitos establecidos y permita conjuntar las herramientas de desarrollo

elegidas y presentar un mejor orden en el desarrollo que permita al sistema crecer y evolucionar de una forma más rápida.

La arquitectura para este proyecto se diseñó con base en el modelo Cliente-Servidor de 3 capas (ver sección 1.3.1), básicamente con este modelo se estableció la interacción entre el cliente y el servidor que tiene acceso a la base de datos. Estas capas son:

- Capa de cliente: compuesta por la interfaz de usuario, realizada en HTML, en la que el usuario envía peticiones e información a la capa siguiente y obtiene resultados de la misma.
- Capa de servidor: en esta capa se procesa toda la información enviada desde la capa del cliente y se procesa para ser enviada a la base de datos y devolver resultados de nuevo a la capa del cliente. Se encarga de la comunicación con la base de datos en MySQL por medio del lenguaje PHP.
- Capa de datos: compuesta por el RDBMS MySql, en esta capa se almacenan los datos en la base de datos y a la que se accede a través de la capa de servidor.

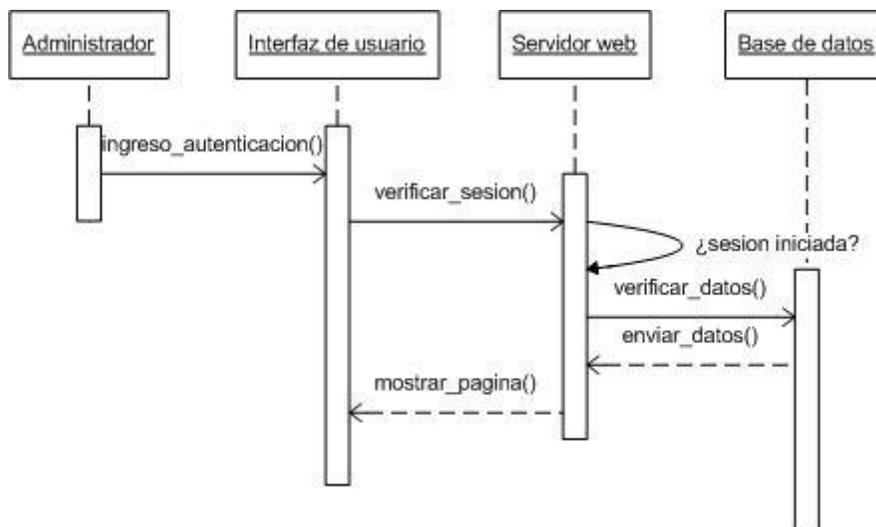
### **3.4. Diagramas de secuencia**

Al momento de obtener los requerimientos y de establecer los casos de uso para estos requerimientos, se define la base sobre la cual se puede construir el sistema, pero los casos de uso solo dan una vista general de la funcionalidad del sistema y no del funcionamiento interno del mismo, por lo que en la etapa de diseño se hicieron los distintos diagramas de secuencia para los

casos de uso descritos en el capítulo 2. Estos diagramas brindan una perspectiva más clara para desarrollar el código de la solución.

Para poder gestionar el proceso de evaluación se debe ingresar primero al sistema, esta acción es parte inicial de cada actividad dentro del flujo del proceso general de evaluación. Dentro de cada actividad del proceso (sección 2.1) se hará uso de la aplicación para gestionar y dar vida a cada una de estas actividades, es por eso que es el diagrama inicial que se presenta.

Figura 11. **Ingreso al sistema**

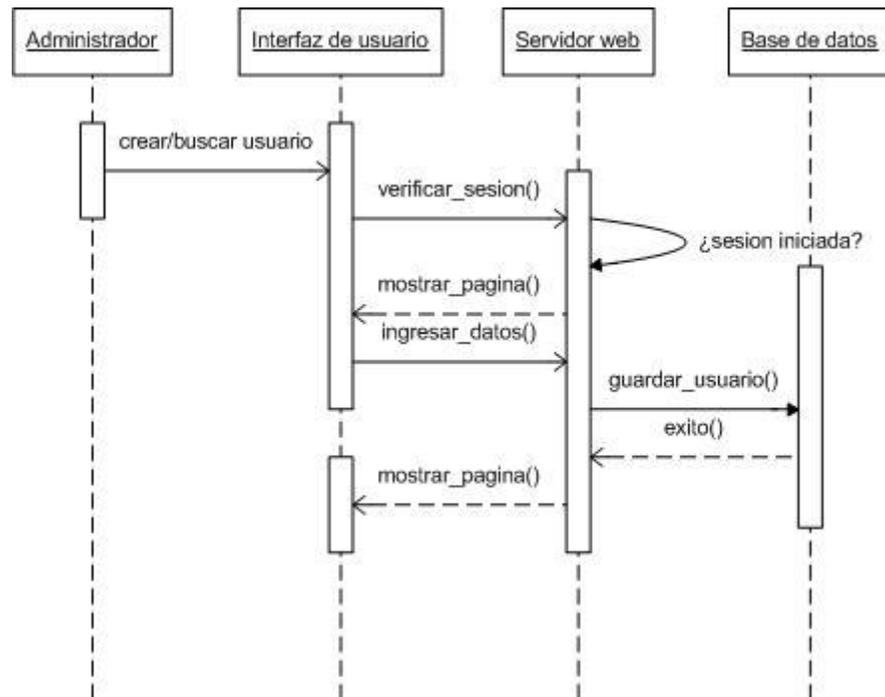


Fuente: elaboración propia.

Siguiendo con el proceso general, la actividad inicial es la aceptación y asignación del estudiante en el EPS, esta actividad comprende todas las acciones de administración y cada uno de los mantenimientos que conforman dichas acciones. Dentro del sistema, esto abarca la administración de usuarios, períodos y asignaciones, que cumplen con las funciones necesarias para dar vida a esta primera actividad dentro del flujo proceso.

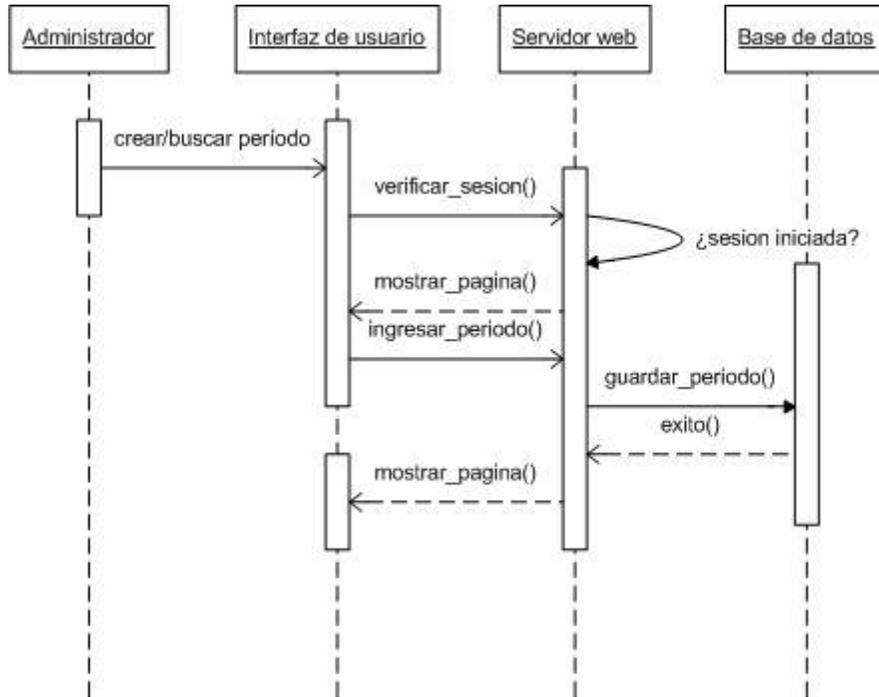
Estas acciones dentro del sistema pueden, sin embargo, ser ejecutadas dentro de actividades posteriores, ya que las administraciones pueden realizarse en cualquier momento, pero es en esta actividad inicial en específico donde se hacen críticas, ya que si no se realizan en este momento, no se puede continuar con las actividades siguientes y no se puede completar, por lo tanto, el flujo del proceso.

Figura 12. **Administrar usuarios**



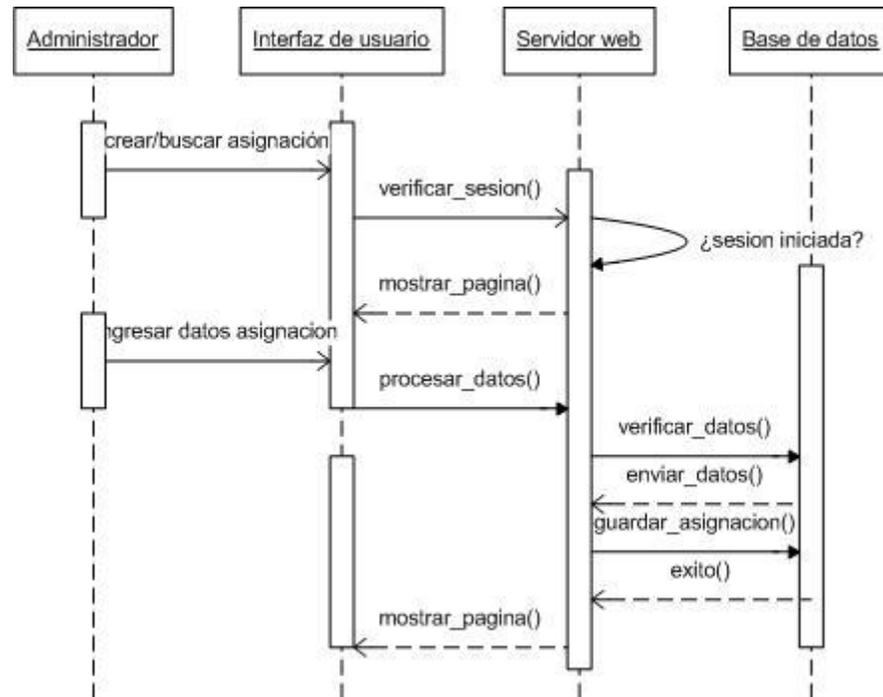
Fuente: elaboración propia.

Figura 13. Administrar períodos



Fuente: elaboración propia.

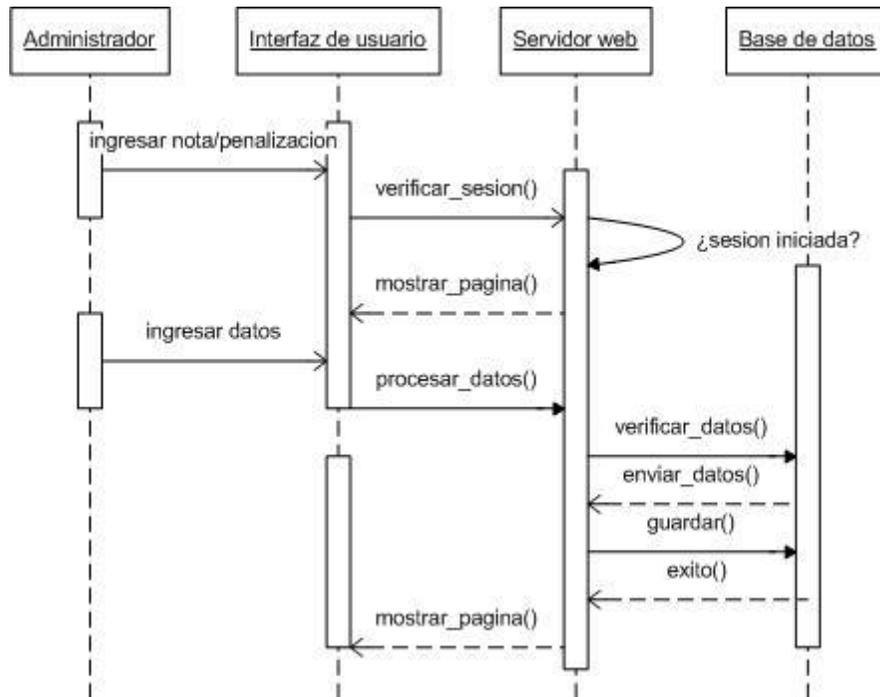
Figura 14. Administrar asignación



Fuente: elaboración propia.

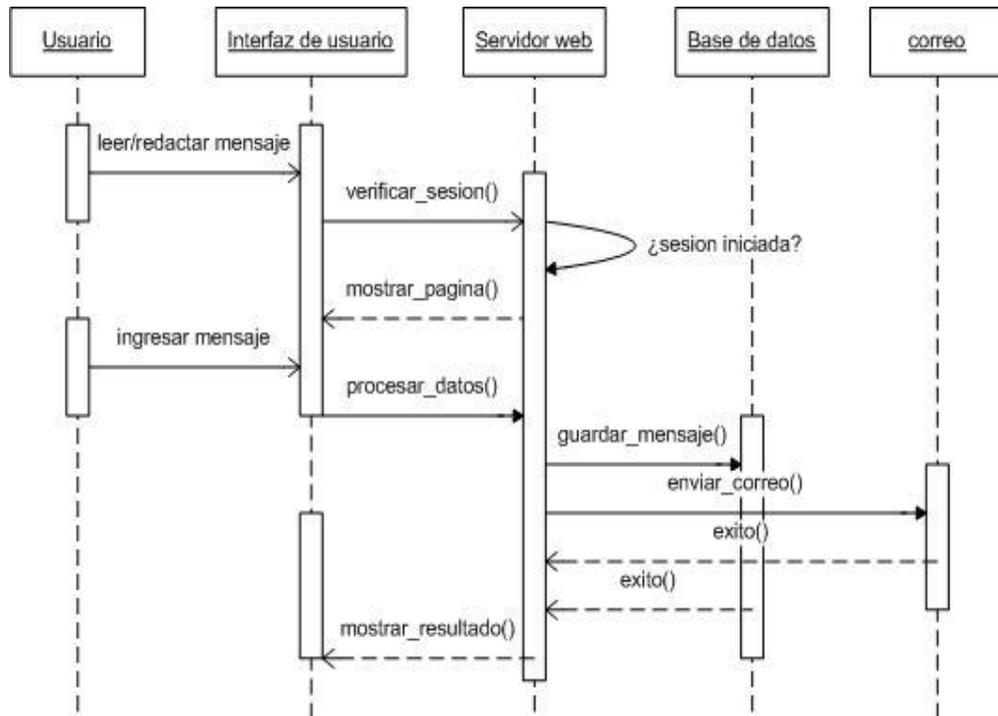
Después de realizar todos los procesos administrativos iniciales, las siguientes actividades del proceso general corresponden a la evaluación como tal del estudiante que realiza el EPS. Para estas actividades corresponden las administraciones de notas, penalizaciones, interacciones en el foro de mensajes y la visualización en sí de las notas correspondientes a cada estudiante. Estas actividades son el núcleo central del proceso, donde se realizan las actividades que hacen que se cumpla el objetivo general del sistema, que es administrar las evaluaciones de cada estudiante que cursa el EPS.

Figura 15. **Administrar notas y penalizaciones**



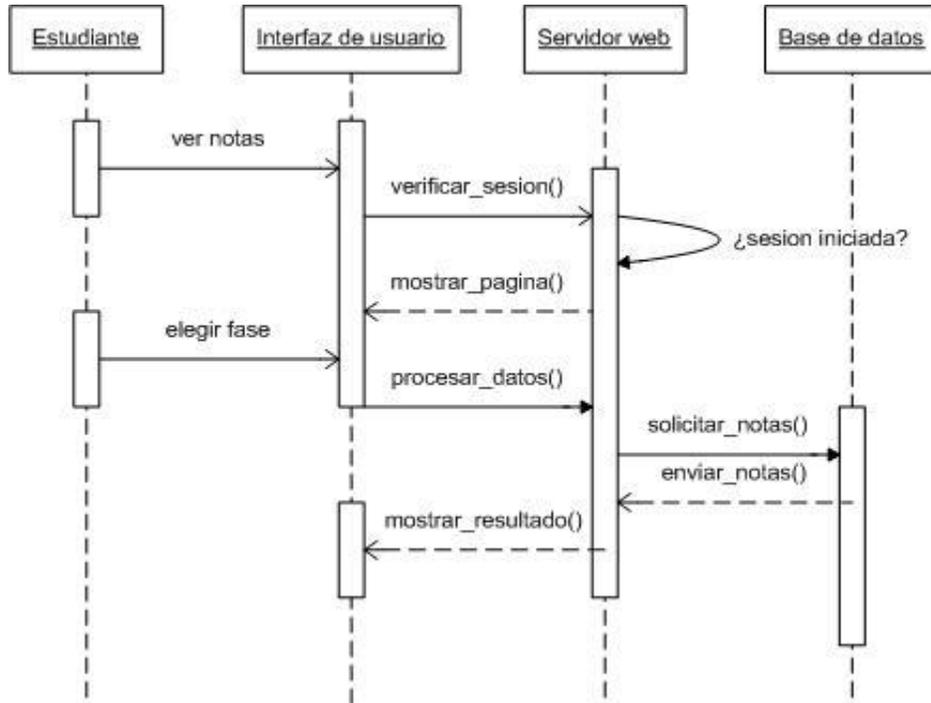
Fuente: elaboración propia.

Figura 16. Administración de foro de mensajes



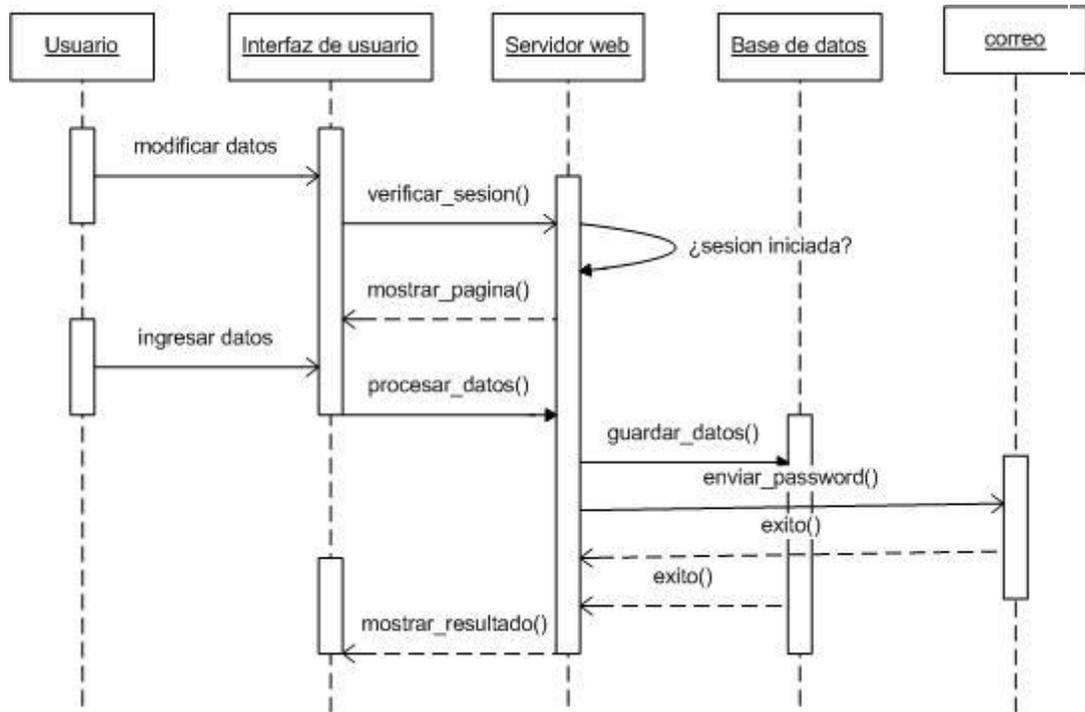
Fuente: elaboración propia.

Figura 17. Ver notas



Fuente: elaboración propia.

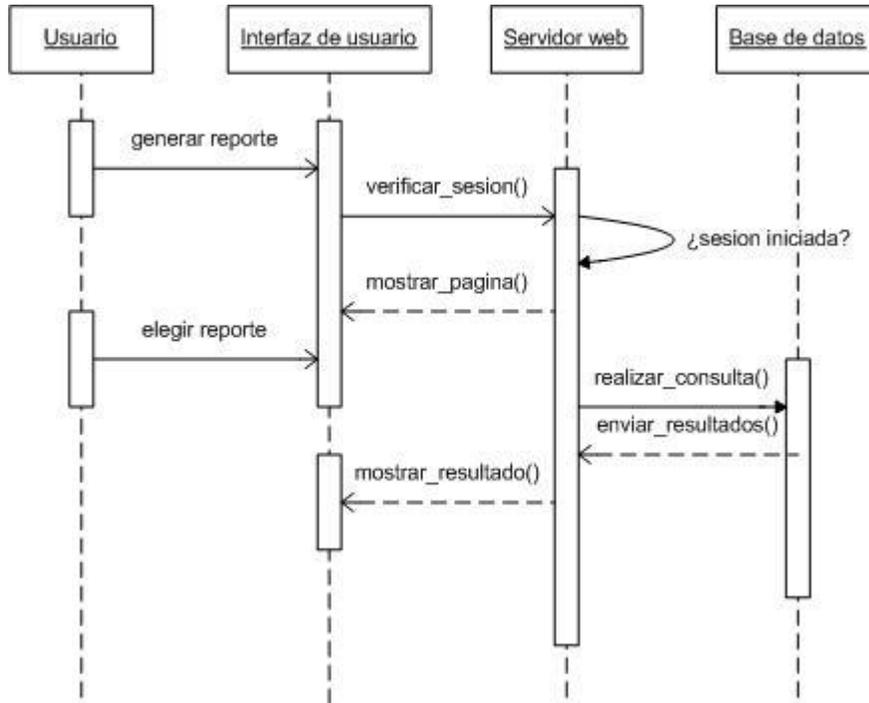
Figura 18. **Modificar datos de usuario**



Fuente: elaboración propia.

Al final de las evaluaciones, los reportes ya tienen los datos suficientes para ser generados y mostrar valores que pueden servir ya sea al administrador o al profesor supervisor en la toma de decisiones.

Figura 19. **Generar reportes**

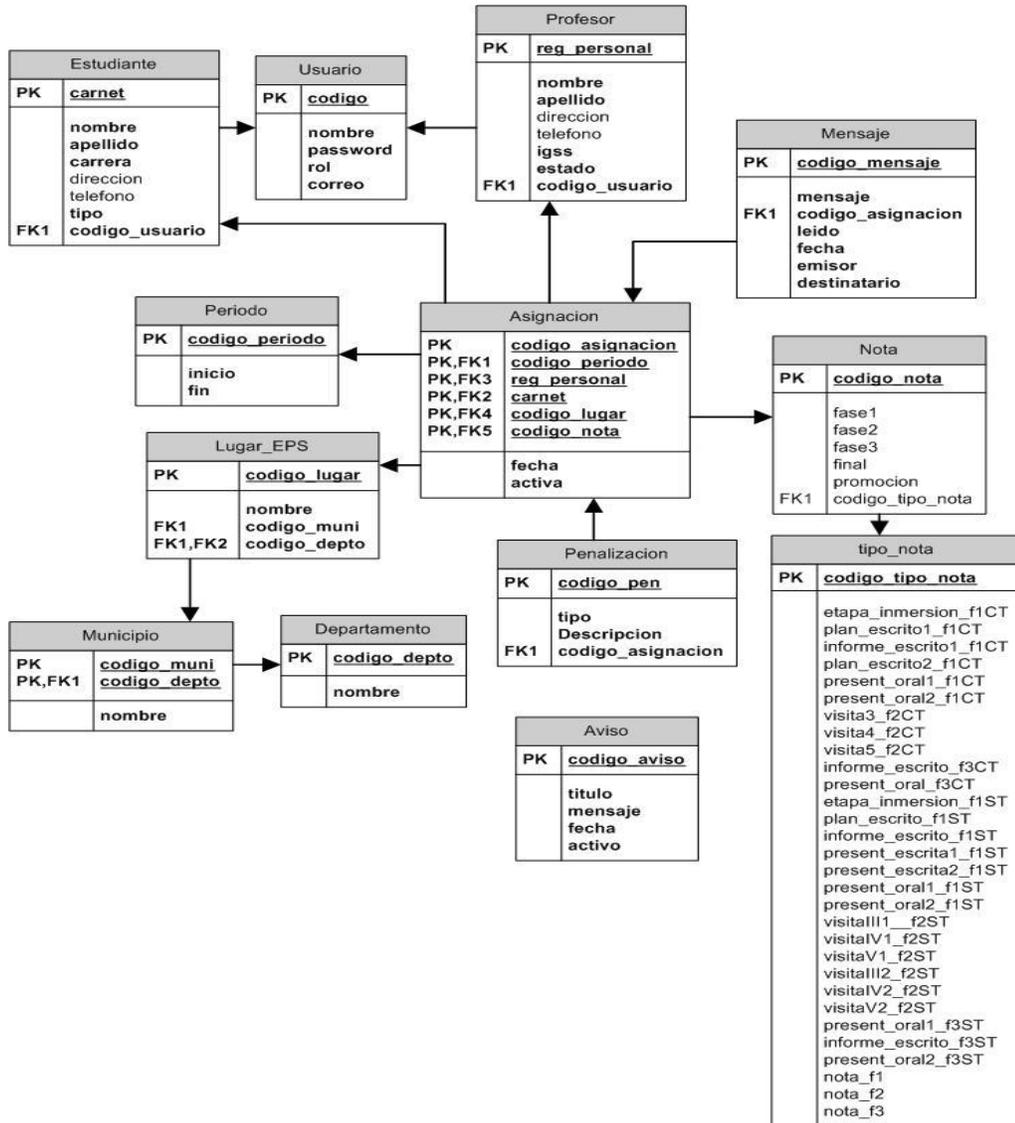


Fuente: elaboración propia.

### 3.5. **Diseño relacional de la base de datos**

La parte central de la aplicación desarrollada, y crítica para el cumplimiento de los requisitos es el diseño de la base de datos relacional, se debe guardar toda la información necesaria para cumplir con los requerimientos. En este caso se definió el modelo relacional como se puede observar en el diagrama siguiente, utilizando el modelo Entidad-Relación (E-R), que permite mostrar la información a almacenar así como las llaves primarias y foráneas necesarias para relacionar las distintas tablas de la base de datos.

Figura 20. Modelo entidad-relación



Fuente: elaboración propia.

En este modelo se puede observar trece tablas que contendrán la información necesaria, así como las relaciones entre cada una de ellas. Las veremos a mayor detalle a continuación.

- Estudiante: esta tabla contiene todos los datos requeridos del estudiante.
  - Llave primaria: número de carné
  - Llaves foráneas: código de usuario
  
- Profesor: la tabla con toda la información de los catedráticos supervisores del área de EPS.
  - Llave primaria: número de registro de personal
  - Llaves foráneas: código de usuario
  
- Usuario: contiene la información de los roles de los usuarios, así como las contraseñas protegidas por encriptación.
  - Llave primaria: código de usuario
- Período: esta tabla contiene los años de inicio y finalización del ciclo de evaluación.
  - Llave primaria: código de período
  
- Lugar de EPS: guarda la información del lugar donde se realiza el EPS, así como la institución financiante.
  - Llave primaria: código del lugar de EPS
  - Llaves foráneas: código de departamento, código de municipio
  
- Departamento: catálogo con el nombre de todos los departamentos de Guatemala
  - Llave primaria: código de departamento

- Municipio: catálogo con todos los municipios de todos los departamentos de Guatemala.
  - Llave primaria: código de municipio, código de departamento
  - Llave foránea: código de departamento
  
- Penalización: tabla que contiene la descripción y el tipo de penalización que se puede otorgar a un estudiante de EPS
  - Llave primaria: código de penalización
  - Llave foránea: código de asignación
  
- Aviso: tabla que guarda los mensajes que se publican al público en la página inicial del sitio.
  - Llave primaria: código de aviso
  
- Nota: en esta tabla se guarda la información de las notas finales y por fase.
  - Llave primaria: código de nota
  - Llaves foráneas: código de tipo de nota
  
- Tipo nota: esta tabla sirve para guardar todas las notas por fases y por tipo de estudiante.
  - Llave primaria: código de tipo de nota
  
- Mensaje: tabla que guarda los mensajes que se envían entre estudiantes y profesores en el foro de mensajes.
  - Llave primaria: código de mensaje
  - Llave foránea: código de asignación

- **Asignación:** esta es la tabla central del modelo, en la que se relacionan todas las demás entre sí, se guarda la información de una asignación estudiante-profesor en un período específico.
  - **Llave primaria:** código de asignación, código de período, registro de personal, carné, código de lugar de EPS, código de nota.
  - **Llaves foráneas:** código de período, registro de personal, carné, código de lugar de EPS, código de nota.

### 3.5.1. Tipos de datos

Los tipos de datos usados para cada campo de las tablas en la base de datos se definieron conforme a los tipos de datos propios de MySQL, el RDBMS escogido. Estos tipos de datos son:

- **INT:** tipo de dato entero
- **VARCHAR:** tipo de dato de caracteres, para guardar cadenas de texto como nombres de usuarios con tamaño limitado o muy pequeño.
- **DATE:** tipo de datos con formato para fechas
- **TEXT:** tipo de dato de texto para guardar cantidades más grandes de caracteres que un VARCHAR definido no soporta, como en las los mensajes en el foro.

Para ver el tipo de datos utilizado en cada campo de cada tabla en la base de datos, revisar el *script* de creación de la base de datos en el apéndice A.

### **3.6. Diseño de la interfaz de usuario**

Entre los requerimientos no funcionales (sección 2.1.6.4) se solicitaba una interfaz sencilla y sin sobrecarga de adornos o animaciones, para esto se diseñó una interfaz con 3 partes, un panel superior, para el logotipo del área; un panel izquierdo, para el menú de navegación principal que contiene todas las opciones disponibles para cada usuario; y el panel derecho, que contiene el menú de navegación dependiente del menú principal y donde se muestra toda la información que maneja el sistema.

Se eligió este diseño debido a la simpleza que representa, los colores elegidos son el verde en tonalidades pastel, con letras azules principalmente, negras y rojas, que contrastan de buena manera y así permiten tener una suavidad en el fondo que no distrae de la información de la página, mientras que este contraste con las letras permite una mejor lectura y enfoque en la parte central de la página, donde se maneja toda la información.

Figura 21. Plantilla de la interfaz de usuario



Fuente: elaboración propia.

Los botones del menú de navegación secundarios, se diseñaron en color verde en un tono contrastante al verde utilizado en el fondo, con letras negras para identificar a cada uno que facilitan identificar las opciones posibles.

Se estableció la resolución de las páginas en 1 024 x 768 pixeles, que es una de las medidas más utilizadas actualmente en los monitores de computadoras.

Figura 22. Interfaz del administrador



Fuente: elaboración propia.

Figura 23. Interfaz del profesor



Fuente: elaboración propia.

Figura 24. Interfaz del estudiante



Fuente: elaboración propia.



## 4. DOCUMENTACIÓN TÉCNICA

Tener una referencia del código contenido en cada página del sistema es de vital importancia para detectar errores y poder dar mantenimientos o ampliaciones a la aplicación. Documentar cada página del sistema es el punto de partida para otros desarrolladores para empezar a entender cómo funciona internamente a nivel de código de programación cada *script* utilizado en las páginas para enviar, recibir y procesar información entre la base de datos y la interfaz de usuario.

En este capítulo se puede ver el significado de las variables utilizadas, así como de las funciones y *scripts* utilizados para el desarrollo de la aplicación.

### 4.1. **Scripts de procesamiento de datos**

En esta sección se describirán las páginas que cumplen con la programación de la lógica del negocio. Dichas páginas corresponden a la capa de negocio que se enlaza con la base de datos y devuelve los resultados a la capa web con la que interactúa el usuario.

#### 4.1.1. **Scripts de las páginas iniciales**

Estas páginas corresponden a la parte inicial del sitio, que se encarga de la verificación de credenciales para autenticación y mostrar los datos generales del sitio a la que cualquier usuario puede acceder.

- Conexión.php: en este *script* se realiza la conexión a la base de datos, a través de las funciones siguientes:
  - *Mysql\_connect*(dirección,usuario,contraseña):esta función tiene 3 parámetros, el nombre o dirección ip del servidor donde se encuentra la base de datos, el nombre de usuario para acceder a la base de datos y su contraseña.
  - *Mysql\_select\_db*(nombre de BD):con esta función se especifica la base de datos a la que se quiere conectar, enviando este nombre como parámetro.
  
- Enviar\_mail.php: este *script* contiene la función enviar\_correo, la cual envía un correo con el destinatario deseado y un mensaje definido. Recibe los siguientes parámetros:
  - \$to: dirección de correo del destinatario.
  - \$from: dirección de correo del emisor.
  - \$from\_name: nombre asociado a la dirección del emisor
  - \$subject: asunto del mensaje.
  - \$body: el mensaje a enviar.
  
- Validar\_user.php: en este *script* se verifican los datos de autenticación ingresados en la pantalla inicial y muestra la página correspondiente al usuario autenticado, o la página principal si ha ocurrido un error. Recibe las siguientes variables para realizar esta tarea:
  - \$usuario: recibe el nombre del usuario ingresado en el formulario de autenticación de la página inicial.
  - \$password: recibe la contraseña del usuario ingresado en el formulario de autenticación de la página inicial.

- \$rol: recibe el rol elegido en el formulario de autenticación de la página principal
- Logout.php: este *script* se llama cuando el usuario que hace uso de la aplicación cierra la sesión, contiene una sola función que destruye todas las variables de sesión en uso: `sesión_destroy()`.

#### 4.1.2. **Scripts del administrador**

Corresponden a las páginas que se encargan de toda la programación de las funcionalidades asociadas al rol de administrador y su procesamiento en base de datos.

- Act\_elim\_aviso.php: *script* que modifica o elimina un aviso de la página inicial, escogido por el usuario, recibe los siguientes parámetros:
  - \$titulo: El título del aviso a guardar o eliminar.
  - \$mensaje: el texto del mensaje que se va a guardar o eliminar
  - \$date: la fecha en que se modifica o elimina el mensaje
- Conexión\_admin.php: *script* de conexión específico para el usuario de rol administrador, utiliza las mismas funciones que el *script* de las páginas iniciales.
- Crear\_asignación.php: en este *script* se crean las nuevas asignaciones, guardando los parámetros siguientes en la base de datos.
  - \$codigo\_lugar: es el código del lugar donde se realiza el EPS.
  - \$periodo: recibe el código del período de asignación.
  - \$carne: recibe el número de carné del estudiante a asignar.
  - \$profesor: recibe el número de

- \$codigo\_depto: recibe el código del departamento donde se realiza el EPS.
  - \$muni: recibe el código del municipio que pertenece al departamento donde se realiza el EPS.
  - \$lugar: recibe el nombre del lugar en donde se realiza el EPS
  - \$financiante: recibe el nombre de la institución que financia el EPS.
  - \$codigo\_nota: guarda el código de la nota asociada a la asignación
  - \$fecha: guarda la fecha en que se realiza la asignación
  - \$valido: bandera booleana para procesar los datos antes de ingresar a la base de datos.
- Crear\_período.php: aquí se procesa la creación de nuevos períodos que se guardan en la base de datos, recibe los siguientes parámetros:
    - \$error: bandera para procesar errores en los datos enviados.
    - \$inicio: recibe el año de inicio del período a guardar.
    - \$fin: recibe el año de finalización del período a guardar.
    - \$permitidos: variable estática que guarda los dígitos del 0 al 9 permitidos en la validación numérica
    - \$valido, \$valido2: banderas booleanas utilizadas para la validación de datos enviados.
- Crear\_user.php:script para la creación de usuarios del sistema de cualquiera de los tres roles. Recibe los siguientes parámetros:
    - \$carnet: carné del estudiante a crear de ser necesario
    - \$password: contraseña del usuario a crear.
    - \$nombre: nombre del usuario a guardar
    - \$apellido: apellido del usuario a guardar
    - \$mail: correo del usuario a guardar
    - \$carrera: identificador de la carrera del estudiante de ser necesario

- \$tel: número de teléfono del usuario.
  - \$dir: dirección del usuario.
  - \$tipo\_est: tipo del estudiante de ser necesario (con tesis o sin tesis aprobada)
  - \$estado: el estado del estudiante, al momento de crearse, tiene el valor: creado.
- Generar\_pass.php: en este *script* se genera una contraseña aleatoria para un usuario especificado, se guarda en la base de datos y se envía al correo del usuario. Solamente recibe dos parámetros:
    - \$usuario: código del usuario al que se le generará la contraseña.
    - \$nombre: nombre del usuario al que se le generará la contraseña.
- Modificar\_asignacion.php: aquí se guardan los cambios realizados sobre alguna asignación creada previamente, en la base de datos. Recibe los parámetros siguientes:
    - \$codigo\_asignacion: código de la asignación a modificar.
    - \$codigo\_lugar: código del lugar modificado en la asignación.
    - \$periodo: código del período modificado en la asignación.
    - \$carne: carné del estudiante modificado en la asignación.
    - \$profesor: registro de personal del profesor modificado en la asignación.
    - \$codigo\_depto: código del departamento modificado en la asignación.
    - \$muni: código del municipio modificado en la asignación.
    - \$lugar: nombre del lugar de EPS modificado en la asignación.
    - \$financiante: nombre de la institución financiera del EPS.
    - \$codigo\_nota: código de las notas asociadas a la asignación.
    - \$fecha: fecha en que se modifica la asignación.

- \$valido: bandera booleana para validación de los datos recibidos.
- Modificar\_periodo.php: en este *script* se realizan cambios a cualquiera de los años de inicio o finalización de un período previamente creado en la base de datos. Recibe como parámetros los años del período:
  - \$inicio: año de inicio modificado.
  - \$fin: año de finalización modificado.
  - \$codigo: código del período a modificar.
  - \$permitidos: variable estática con los dígitos del 0 al 9 para validación numérica de los datos recibidos.
  - \$valido, \$valido2: banderas booleanas.
- Modificar\_user.php: en este *script* se realizan los cambios enviados a través de los parámetros siguientes, de la información del usuario que se desee:
  - \$codigo\_usuario: código del usuario a modificar.
  - \$carnet: nuevo carné de estudiante.
  - \$nombre: nuevo nombre del usuario.
  - \$apellido: nuevo apellido del usuario.
  - \$mail: nuevo correo del usuario.
  - \$carrera: nueva carrera del estudiante.
  - \$tel: nuevo número de teléfono del usuario.
  - \$dir: nueva dirección del usuario.
  - \$tipo\_est: nuevo tipo de estudiante.
  - \$reg\_personal: nuevo registro de personal del profesor.
- S\_publicar\_aviso.php: la publicación de avisos en la sección inicial del sitio se realiza en este *script* con los siguientes parámetros:
  - \$titulo: título del aviso a publicarse.

- \$mensaje: texto del aviso a publicarse.
- \$date: fecha de publicación del aviso.

#### 4.1.3. **Scripts del profesor**

Estas páginas se encargan del procesamiento de la información y tareas propias del rol correspondiente a los catedráticos supervisores y de la comunicación con la base de datos.

- Cargar\_mensajes.php: se encarga de mostrar los mensajes entre el profesor y todos los estudiantes bajo su supervisión en un período específico. Contiene la función siguiente:
  - Cambiar\_fecha(\$fecha1): esta función cambia el formato de la fecha dispuesto en MySQL (Inglés) al formato en español, recibe una fecha en formato del tipo DATE de MySQL y retorna la fecha en el formato distinto.
- Enviar\_mensaje\_profesor.php: en este *script* se envía el mensaje a un estudiante especificado por parte del profesor encargado. Recibe los parámetros siguientes:
  - \$periodo: período en el que se envía el mensaje.
  - \$carne: carné del estudiante al que se le envía el mensaje.
  - \$mensaje: texto del mensaje que se envía.
  - \$leido: estado del mensaje.
  - \$fecha: fecha en que se envía el mensaje.
  - \$emisor: nombre del profesor emisor.
  - \$destinatario: nombre del estudiante destinatario.
  - \$codigo\_asignacion: código de asignación entre el estudiante y el profesor.

- \$correo: correo del destinatario.
- Enviar\_penalización: en este *script* se guardan las penalizaciones dadas a estudiantes, pudiendo ser de tipo leve o grave. Recibe estos parámetros:
  - \$carne: carné del estudiante a penalizar.
  - \$periodo: período en el cual se penaliza.
  - \$reg\_personal: registro de personal del profesor que envía la penalización.
  - \$tipo: tipo de penalización, leve o grave.
  - \$descripcion: texto con la descripción de la causa de la penalización.
  - \$graves: contador de penalizaciones de tipo grave.
  - \$leves: contador de penalizaciones de tipo leve.
- Guardar\_notaf1.php: en este *script* se procesan las notas ingresadas en el formulario de la fase 1, para cualquiera de los 2 tipos de estudiante. Cada parámetro de este script representa al valor numérico de una nota que compone esta fase, para mayor referencia revisar el apéndice B que contiene los formularios oficiales de evaluación con los campos para cada fase.
- Guardar\_notaf2.php: en este *script* se procesan las notas ingresadas en el formulario de la fase 2, para cualquiera de los 2 tipos de estudiante. Cada parámetro de este script representa al valor numérico de una nota que compone esta fase, para mayor referencia revisar el apéndice B que contiene los formularios oficiales de evaluación con los campos para cada fase.
- Guardar\_notaf3.php: en este *script* se procesan las notas ingresadas en el formulario de la fase 3, para cualquiera de los 2 tipos de estudiante. Cada

parámetro de este script representa al valor numérico de una nota que compone esta fase, para mayor referencia revisar el apéndice B que contiene los formularios oficiales de evaluación con los campos para cada fase.

- Modificar\_pass.php: aquí se realiza el cambio de contraseña para el profesor, recibiendo estos parámetros para el efecto:
  - \$anterior: contraseña actualmente guardada en la base de datos.
  - \$nuevo1: contraseña nueva a guardar.
  - \$nuevo2: contraseña nueva escrita de nuevo para asegurarnos que se haya escrito la contraseña que se desea realmente.
  
- Modificar\_profesor.php: en este *script* podemos modificar los datos personales del profesor en la sesión actual. Recibe los siguientes parámetros:
  - \$codigo\_usuario: código del usuario relacionado al registro de personal del profesor a modificar.
  - \$regPersonal: registro de personal del profesor a modificar.
  - \$nombre: nombre nuevo del profesor.
  - \$apellido: apellido nuevo del profesor.
  - \$mail: correo nuevo del profesor.
  - \$igss. número nuevo de carné de IGSS del profesor.
  - \$tel: número de teléfono nuevo del profesor.
  - \$dir: nueva dirección del profesor.
  - \$tipo: nuevo tipo de profesor.
  - \$error: bandera booleana.
  - \$permitidos: variable estática que guarda los dígitos del 0 al 9 permitidos en la validación numérica.

- `Selec_tipo.php`: *script* para procesar la petición de ingreso de notas y re direccionar a la página del tipo correspondiente
  - `$periodo`: el período elegido para el ingreso de notas.
  - `$carne`: el carné del estudiante al que se le ingresaran las notas.

#### 4.1.4. **Scripts del estudiante**

Estas páginas corresponden al rol del estudiante, que básicamente tiene funcionalidades de consulta y modificación de datos personales, y estos *scripts* se encargan de llevar a cabo todas estas tareas.

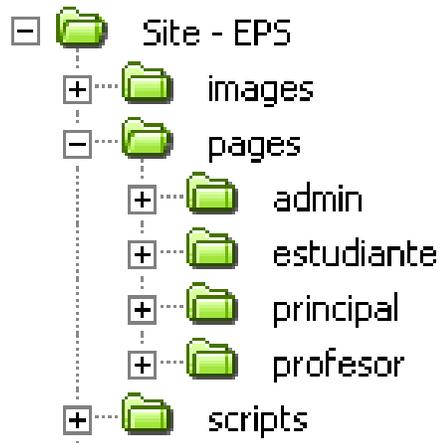
- `Enviar_mensaje_estudiante.php`: en este *script* el estudiante puede enviar mensajes al profesor encargado, funciona con los siguientes parámetros:
  - `$periodo`: período en el cuál se envía el mensaje.
  - `$reg_personal`: registro de personal del profesor destinatario.
  - `$mensaje`: texto del mensaje a enviar al profesor.
  - `$leido`: guarda el estado del mensaje.
  - `$fecha`: fecha de envío del mensaje.
  - `$emisor`: nombre del estudiante emisor.
  - `$destinatario`: nombre del profesor destinatario.
  - `$codigo_asignacion`: código de la asignación.
  - `$correo`: el correo del profesor destinatario.
- `Modificar_estudiante.php`: en este *script* podemos modificar los datos personales del estudiante en la sesión actual. Recibe los siguientes parámetros:
  - `$codigo_usuario`: código del usuario relacionado al carné del estudiante a modificar.
  - `$carnet`: carné del estudiante a modificar.

- \$nombre: nombre nuevo del estudiante.
  - \$apellido: apellido nuevo del estudiante.
  - \$mail: correo nuevo del estudiante.
  - \$tel: número de teléfono nuevo del estudiante.
  - \$dir: nueva dirección del estudiante.
  - \$tipo: nuevo tipo de estudiante.
  - \$error: bandera booleana.
  - \$permitidos: variable estática que guarda los dígitos del 0 al 9 permitidos en la validación numérica.
- Modificar\_pass.php: aquí se realiza el cambio de contraseña para el estudiante, recibiendo estos parámetros para el efecto:
    - \$anterior: contraseña actualmente guardada en la base de datos.
    - \$nuevo1: contraseña nueva a guardar.
    - \$nuevo2: contraseña nueva escrita de nuevo para asegurarnos que se haya escrito la contraseña que se desea realmente.
- Selec\_tipo.php: *script* para ver las notas y re direccionar a la página del tipo correspondiente para el estudiante en sesión.
    - \$periodo: el período elegido para lectura de notas.
    - \$carne: el carné del estudiante.

#### **4.2. Estructura del sitio Web**

En la siguiente gráfica se puede observar de mejor manera la estructura de carpetas y subcarpetas que conforman la estructura de archivos que forman el proyecto.

Figura 25. **Árbol de la estructura del sitio Web**



Fuente: elaboración propia.

En este diagrama se puede identificar 3 partes principales, la carpeta *images*, que contiene todas las imágenes de botones y banners que se utilizan en todo el sitio; la carpeta *pages*, que contiene a todos los archivos de código fuente en PHP y HTML de las páginas que conforman el sitio; y la carpeta *scripts* que contiene los *scripts* globales para el sitio, explicados en la sección anterior.

La carpeta *pages* se subdivide a su vez en 4 subcarpetas, siendo estas la carpeta *admin* que contiene las páginas correspondientes al rol de administrador; la carpeta *estudiante*, que contiene las páginas para este rol; al igual que la carpeta *profesor*, que contiene las páginas de este rol. La última carpeta, *principal*, contiene las páginas de inicio del sistema.

## CONCLUSIONES

1. La ingeniería de software es una ciencia fundamental para poder desarrollar soluciones que cumplan con los requerimientos del cliente.
2. La base de una solución de calidad es la toma adecuada de requerimientos, a partir de esto, se puede analizar y diseñar software adecuado para las necesidades del cliente.
3. La base de la toma de requerimientos es la definición del flujo del proceso.
4. Al entender el flujo del proceso general del sistema, se puede definir las partes que este sistema afectarán.
5. Las metodologías de desarrollo de software se deben adaptar a las necesidades propias del proyecto y deben permitir flexibilidad en el desarrollo.
6. La documentación realizada del análisis y diseño del proyecto ayudará a futuros desarrolladores que quieran modificar o ampliar este software a lograr de una mejor forma estas tareas.
7. Se desarrolló un software aprobado por los usuarios que cumple con los requerimientos establecidos y con la definición del proceso general de evaluación del EPS.

8. La documentación técnica realizada da un indicio del funcionamiento del código interno de la aplicación.

## RECOMENDACIONES

1. Nombrar a un administrador de la base de datos con conocimiento del RDBMS escogido, ya que el programa en sí tiene limitaciones para manipular la información guardada en la base de datos directamente, debido a las políticas de seguridad establecidas.
2. Una recomendación importante es manejar las contraseñas de acceso a la base de datos con todas las precauciones que este tipo de información demanda. No revelar contraseñas a otros usuarios o a personas ajenas al área.
3. Mejorar las búsquedas en la base de datos, ya que actualmente utilizan consultas básicas que tienen ciertas limitantes.
4. No se debe alterar directamente sin conocimiento, el código HTML contenido en las páginas del sitio, ya que hasta un cambio menor en la interfaz, aunque pareciera no afectar, podría ser causa de errores en la aplicación.
5. Realizar un período de pruebas por parte de los usuarios para detectar errores que al desarrollador pudieran habersele escapado, y tener un desarrollador capacitado para resolverlos.
6. Es importante definir un plan de mantenimiento y respaldo, tanto de la aplicación como de base de datos.



## BIBLIOGRAFÍA

1. DATE, C.J. *Introducción a los Sistemas de bases de datos*. 7a ed. México: Pearson Education. 2001. 936 p.
2. SOMMERVILLE, Ian. *Ingeniería del Software*. Madrid, España: Pearson Education. 2005. 687 p. ISBN: 84-7829-074-5.
3. GRACIA, Joaquín. *UML: Casos de Uso. Use case*; Desarrollo de software orientado a objetos. Zaragoza, España: [ref. 27 de septiembre de 2003], Disponible en web: <<http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>>. [Consulta: 16 de marzo de 2013].
4. PRESSMAN, Roger S. *Ingeniería del Software: un enfoque práctico*. Madrid, España: McGraw Hill, 2002. 601 p. ISBN: 0-07-709677-0.
5. RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. *The unified mode language reference manual*. Reading, Massachusetts: Addison Wesley, 1999. 550 p. ISBN: 0-201-30998-X.



## APÉNDICE



# FORMULARIOS DE EVALUACIÓN DE LOS ESTUDIANTES DE EPS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE AGRONOMIA  
AREA INTEGRADA -EPS-

EPSA/AGT01

Válida desde el 22/06/05

CONTROL PERSONAL DE EPSA

(Estudiantes que han iniciado su trabajo de tesis aprobado por el IIA)

Periodo: \_\_\_\_\_

No. de orden	Carné	Nombre del estudiante	FASE DE INDUCCIÓN Y PLANIFICACIÓN 20/100							FASE DE EJECUCIÓN 50/100					FASE FINAL 30/100			NOTA FINAL	PROMOCIÓN FINAL A/R			
			Inducción 05/100		Diagnóstico 30/100		Proyecto de Servicio = 25/100		Presentación de Diagnóstico y Servicio (Reunión Comunal o Empresarial I) 15/100		SERVICIOS 100/100			Nota parcial 100/100		Reunión Comunal o Empresarial II 100/100				Nota parcial 100/100	Promoción de la Fase A/R	
			Etapa de inducción	Plan escrito	Informe escrito	Plan escrito	Presentación Oral (Diagnóstico)	Presentación Oral (Servicio)	Nota parcial 100/100	Promoción de la Fase A/R	Visita III	Visita IV	Visita V	Nota parcial 100/100	Promoción de la Fase A/R	Informe escrito	Presentación Oral			Nota parcial 100/100	Promoción de la Fase A/R	
			Ponderación		65/5	10/30	30/30	15/25	30/40	30/40	100/100	A/R	30/100	30/100	40/100	100/100	A/R			40/60	40/40	100/100
1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						

Supervisor-Docente -EPSA-

Coordinador Área Integrada -EPSA-

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE AGRONOMIA  
AREA INTEGRADA -EPS-

EPSA/AGT01

Metodología 22/06/05

CONTROL PERSONAL DE EPSA

(Estudiantes sin punto de tesis)

Periodo: \_\_\_\_\_

No. de orden	Carné	Nombre del estudiante	FASE DE INDUCCIÓN Y PLANIFICACIÓN 20/100							FASE DE EJECUCIÓN 50/100					FASE FINAL DE ELABORACIÓN Y EVALUACIÓN DEL INFORME FINAL 30/100			NOTA FINAL	PROMOCIÓN FINAL A/R							
			Inducción 05/100		Diagnóstico Comunal o Empresarial 30/100		Plan escrito Proyecto Servicio e Investigación 35/100		Reunión Comunal o Empresarial I 15/100		Revisión Académica 15/100		Proyecto de Servicios 50/100			Proyecto de Investigación 50/100				Reunión Comunal o Empresarial II 30/100		Presentación Académica II 70/100	Nota parcial 100/100	Promoción de la Fase A/R		
			Etapa de inducción	Plan escrito	Informe escrito	Presentación escrita (Comunal)	Presentación escrita (Investigación)	Presentación oral	Presentación oral	Nota parcial 100/100	Promoción de la Fase A/R	Visita III	Visita IV	Visita V	Visita III	Visita IV	Visita V			Nota parcial 100/100	Promoción de la Fase A/R	Presentación oral	Informe escrito	Presentación oral	Nota parcial 100/100	Promoción de la Fase A/R
			Ponderación		5/5	10/30	30/30	15/25	30/35	15/15	15/15	100/100	A/R	15/50	15/50	30/50	15/50			15/50	30/50	100/100	A/R	30/30	35/30	30/70
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8																										
9																										
10																										

Supervisor-Docente -EPSA-

Coordinador Área Integrada -EPSA-

