

**UNAE.** Universidad Autónoma de Encarnación

Facultad de Ciencia, Arte y Tecnología

Licenciatura en Análisis de Sistemas Informáticos

**DESARROLLO DE SOFWARE**

|  |
| --- |
| **PRESENTE** |

**Profesor**: Profesor Hugo Sendoa.

**Materia**: Ing. Software

**AUTORES:**

* **Alvarez , Cristina.**
* **Fernández Insaurralde, Patricia Lorena.**
* **Mathiasen, Carlos.**
* **Rojas, Jorge.**
* **Sanches, Jorge.**

Encarnación – Paraguay

2015

**INDICE**

Contenido

[PLANIFICACIÓN 4](#_Toc428524192)

[Problemática 5](#_Toc428524193)

[Delimitación y Alcance del Tema. 5](#_Toc428524194)

[**a.** **Alcances:** 5](#_Toc428524195)

[Limitaciones: 5](#_Toc428524196)

[Viabilidad: 6](#_Toc428524197)

[ESPECIFICACIONES TÉCNICAS DEL SISTEMA 6](#_Toc428524198)

[REQUERIMIENTOS 8](#_Toc428524199)

[1.     Identificar  actividad del sistema: 9](#_Toc428524200)

[2. Modelo del negocio 9](#_Toc428524201)

[1.   Cuestionario 11](#_Toc428524202)

[ANÁLISIS DE DISEÑO 13](#_Toc428524203)

[DIAGRAMAS 14](#_Toc428524206)

[**3.** **Identificar los CU** 23](#_Toc428524207)

[**A)** **ABM Usuarios** 23](#_Toc428524208)

[**B)** **Ingresar al sistema** 36](#_Toc428524209)

[**C)** **ABM Facultad** 40](#_Toc428524210)

[**D)** **ABM Carreras** 53](#_Toc428524211)

[**E)** **ABM Empleado** 66](#_Toc428524212)

[**F)** **ABM Alumnos** 79](#_Toc428524213)

[G) ABM Materias. 92](#_Toc428524214)

[**H)** **Crear planilla de asistencia** 105](#_Toc428524215)

[**I)** **Registrar asistencias** 109](#_Toc428524216)

[**J)** **Reporte de asistencia** 114](#_Toc428524217)

[ANÁLISIS DE RIESGO 118](#_Toc428524218)

[Plan de acción 119](#_Toc428524221)

[2. Propósito del plan 120](#_Toc428524223)

[*4. Análisis y Gestión de Riesgos* 121](#_Toc428524224)

[*44.1.2 Declaración de los Riesgos* 125](#_Toc428524225)

[4.2 Análisis y prioridad de los riesgos. 133](#_Toc428524226)

[4.3 Gestión de los Riesgos 141](#_Toc428524227)

[. Líneas de Acción 141](#_Toc428524228)

[PLAN DE CONFIGURACION 149](#_Toc428524229)

[Control de la documentación 150](#_Toc428524232)

[Control de la Configuración. 150](#_Toc428524233)

[Histórico de Versiones. 150](#_Toc428524234)

[Histórico de Cambios. 150](#_Toc428524235)

[Introducción 151](#_Toc428524236)

[Propósito 151](#_Toc428524237)

[Alcance 151](#_Toc428524238)

[Documentos Referenciados 152](#_Toc428524239)

[1.1 Definiciones 152](#_Toc428524240)

[Organización 153](#_Toc428524241)

[Herramientas 154](#_Toc428524242)

[Procedimientos 155](#_Toc428524243)

[Herramientas y Técnicas de Implementación del Software 162](#_Toc428524244)

[PLAN DE TEST DEL SISTEMA 167](#_Toc428524245)

[PRESENTE 167](#_Toc428524246)

[**Historia del Documento** 168](#_Toc428524247)

[Propósito 170](#_Toc428524248)

[Test del Sistema 171](#_Toc428524249)

[Configuración del Test 172](#_Toc428524250)

[Tareas 172](#_Toc428524251)

[Planificación 173](#_Toc428524252)

# PLANIFICACIÓN

## Problemática

La Universidad Autónoma de Encarnación (UNAE) forma profesionales de excelencia en las diferentes ramas del saber y la práctica, con el fin de lograr la gestión y desarrollo del talento humano calificado, que propicie los cambios requeridos en las organizaciones y la sociedad.

Actualmente, el control de asistencia se hace vía manual para un seguimiento específico en cuanto a la asistencia de los alumnos salvo para el primer año donde se utiliza un formulario de google.

El proceso es el siguiente se cargan los alumnos ausentes en una tabla nada más en el sistema. Dicho proceso podría mejorar.

Por esta razón y en vista a la necesidad de que exista un sistema en el que se pueda registrar la asistencia de los alumnos para un mayor seguimiento, se lleva a cabo el proyecto de desarrollar un sistema web para registrar las asistencias de los alumnos de una manera más cómoda y práctica y en el que se lleve un mejor control de la misma.

## Delimitación y Alcance del Tema.

* 1. **Alcances:**
* Implementación de Interfaz de login.
* Implementación de Interfaz de bienvenida.
* Implementación del sistema carga de alumnos
* Implementación del sistema carga de profesores
* Implementación del sistema carga de materias
* Implementación del sistema carga de facultades
* Implementación del sistema carga de usuarios
* Implementación del sistema asignación de roles y permisos a usuarios
* Implementación de la Interfaz de reportes de porcentajes de asistencias de los alumnos.

## Limitaciones:

* No se contemplan más servicios de los ya mencionados anteriormente.
* No se contemplan la unión de este sistema con otro ya utilizado para gestión de alumnos.

Evaluación de:

## Viabilidad:

* + 1  Operacional : Es viable operacionalmente ya que podría ser utilizado fácilmente tanto por el alumno como por la persona que esté a cargo de manipular el sistema (administrador-secretarios)  con un mínimo de instrucciones que serán dadas por el mismo sistema o por una persona capacitada para su uso.
  + 2 Técnica: Es viable técnicamente ya que el software podría ser diseñado, implementado, operado y mantenido.
  + 3 Económica: Es viable económicamente ya que los equipamientos tecnológicos  requeridos para el desarrollo del mismo no son muy costosos. Con las computadoras existentes ya es posible manipular el sistema lo que lo hace más adquisitivo económicamente.

## ESPECIFICACIONES TÉCNICAS DEL SISTEMA

El software se desarrollará mediante la combinación de las siguientes herramientas y los siguientes lenguajes de programación

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Caracteristica** | **Uso** |
| Git | Es un sistema de control de versiones distribuido. | Control de versiones |
| Githut | Es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git. | Servidor de versiones |
| Collabtive | Es un software de gestión de proyectos totalmente open sourse. | Gestor de proyectos |
| Scrum | Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. | Metodología de desarrollo |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Caracteristica** | **Uso** |
| Espiral | Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. | Modelo de proceso |
| Python | Se trata de un lenguaje de programación que soporta orientado a objetos. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. | Desarrollo del software |
| Django | Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo-Vista-Controlador.Permite construir aplicaciones web **más rápido y con menos código.** | Framework |
| Heroku | Es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación. | Servidor de prueba |
| ArgoUML | Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD. | Diagramas de casos de usos |
| GanttProject | Es una aplicación que permite organizer y planificar proyectos a través de diagramas Gannt, como asi también la carga horaria de los integrantes dentro del mismo. | Cronograma de actividades |
| COCOMOII | Modelo de costes constructiva, es una algorítmica modelo de estimación de costos de software. | Estimación de costos. |

# REQUERIMIENTOS

**Requerimientos del Sistema**

## 1.     Identificar  actividad del sistema:

a.    Registrar usuario.

b.   Ingresar al sistema.

c. Agregar alumnos matriculados

d.    Introducir la asistencia diaria de alumnos.

e.   Imprimir informes semanales, mensuales de la asistencia de alumnos.

f. Calcular el porcentaje de asistencia del alumno.

g. Mostrar en forma gráfica el porcentaje de asistencia de los alumnos.

## 2. Modelo del negocio

**Un sistema de contador de asistencia por alumno.**

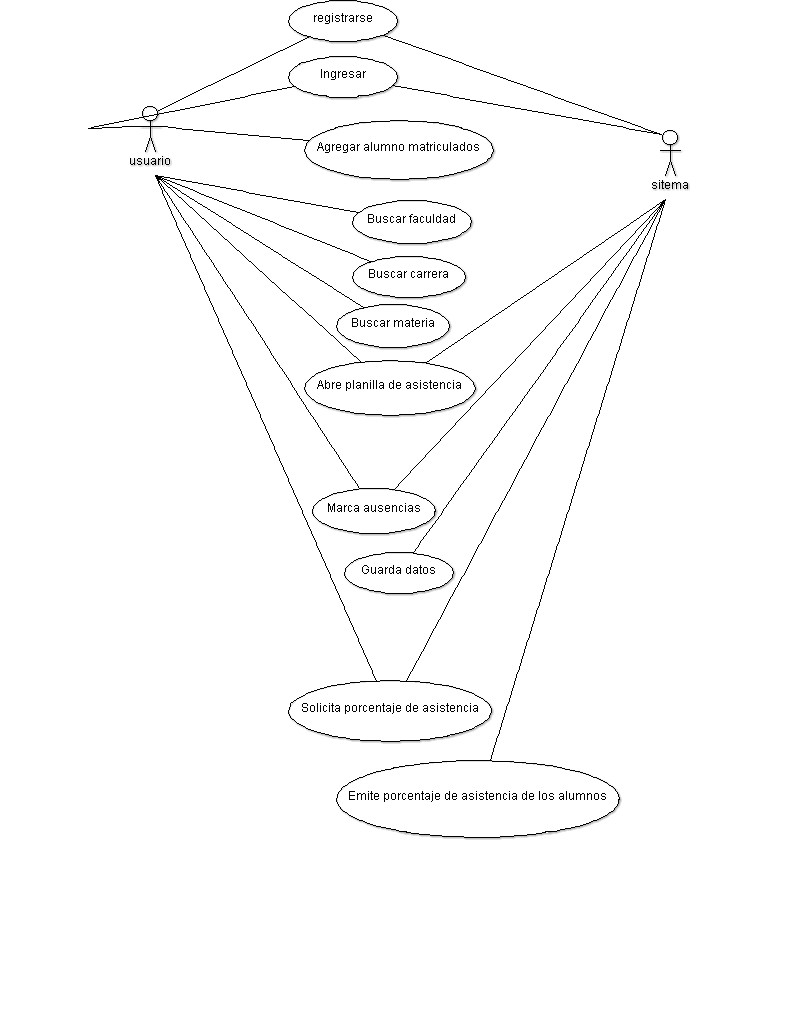
La recepción se encargara de ingresar a los alumnos matriculados, el profesor será un usuario y tendrá permisos para cargar las asistencias y ausencias de los alumnos.  
El alumno podrá solamente ver sus asistencias.

Los usuarios ingresan diariamente la asistencia de cada alumno, de las diferentes carreras.

Las asistencias se van guardando y contando a diario. La interfaz de ingreso de datos será como la de una planilla de cálculos Excel. Cuando el usuario registrado ingresa al sistema aparecen las opciones de carreras y cursos. Una vez cargadas se hace clic en la opción guardar, el sistema emitirá un mensaje aprobando o rechazando los datos ingresados.

Cuando el usuario quisiera saber el total de asistencia de alumnos tendrá la opción de hacerlos mediante los reportes de asistencia.

**Diagrama**

****

## 1.   Cuestionario

**a.**      **Cuestionario Abierto**

1. ¿A qué se dedica la UNAE?

Formar profesionales de excelencia en las diferentes ramas del saber y la práctica, con el fin de lograr la gestión y desarrollo del talento humano calificado, que propicie los cambios requeridos en las organizaciones y la sociedad.

1. ¿Cuáles son las facultades que posee? 3. ¿Cuáles son las carreras que ofrece?

**Facultad**: Ciencias, Arte y Tecnología.

**Carrera:** Análisis de Sistemas Informáticos.

Diseño de Modas.

Diseño Gráfico.

**Facultad:** Ciencias Empresariales.

**Carrera:** Administración.

Contaduría Pública.

Ingeniería Comercial.

Administración de Empresas Agropecuarias.

Relaciones Públicas.

**Facultad:** Ciencias Jurídicas, Humanas y Sociales.

**Carrera:** Derecho.

Trabajo Social.  
 Psicopedagogía.

**Facultad:** Ciencias de la Educación.

**Carrera:** Educación Física.

Lic. Ciencias de la Educación.

Lic. Educación Inicial.

1. ¿Cómo se  realizan las inscripciones de los alumnos para cada carrera?

El interesado deberá presentar los siguientes documentos:

* Certificado de estudio original.
* Copia autenticada del título de bachiller.
* Fotocopias de cedula autenticada (2).
* Foto tipo carnet (2).
* Certifica de nacimiento original.

1. ¿Cuál es el tiempo inicial y límite para la inscripción de un alumno?

Inicio de inscripciones: principios de diciembre.

Tiempo límite: hasta finales de marzo.

1. ¿Cómo se ejecuta el control de asistencia para cada alumno?

El control de asistencia se hace vía manual para un seguimiento específico en cuanto a la asistencia de los alumnos.

1. ¿Cuál es el porcentaje de asistencia requerido para rendir una materia?

Para que el alumno tenga derecho a rendir debe tener una asistencia del 75 %.

Para tener derecho a rendir el examen extraordinario no debe haber rendido la tercera oportunidad.

1. ¿Cuál es el motivo principal objetado por los alumnos al no asistir a clases?

Cuestiones laborales económicas, complejidad de la carrera.

1. ¿Qué sucede cuando el alumno ya no tiene derecho a rendir? A causa de la falta de asistencia ¿Qué solución se le es ofrecida?

Para eso está el extraordinario, lo demás es conversable pero la primera ya no.

1. ¿En caso de que un alumno se vea obligado a re cursar la materia por falta de asistencia, su asistencia en el año perdido es tomada en cuenta para el siguiente año?

No, al alumno debe volver a re cursar la materia por completo, llevando en cuenta nuevamente la asistencia.

**b.**      **Cuestionario Cerrado**

**1. ¿La Universidad Autónoma de Encarnación cuenta con sedes?**

1. Si
2. No
3. ¿Cuántas?

Si cuenta con sedes en Encarnación – Colonias Unidas – Ciudad del Este.

**2. ¿Cuenta con algún sistema informático para el registro de asistencias de alumnos?**

a.    Si

b.   No

No, solo para el primer año que es formulario de google.

El proceso es el siguiente se cargan los alumnos ausentes en una tabla nada más en el sistema.

**3. ¿Cuenta con una representación gráfica (estadística) en donde se pueda observar el porcentaje de asistencia de los alumnos?**

a.    Si

b.      No

No se cuenta con dicha representación, es importante la necesidad de que exista un sistema en el que se pueda registrar la asistencia de los alumnos para un mayor seguimiento, de esa forma se podría evitar que el alumno abandonase la carrera detectando el problema si lo existiese orientándolos de la forma que sea posible.

# ANÁLISIS

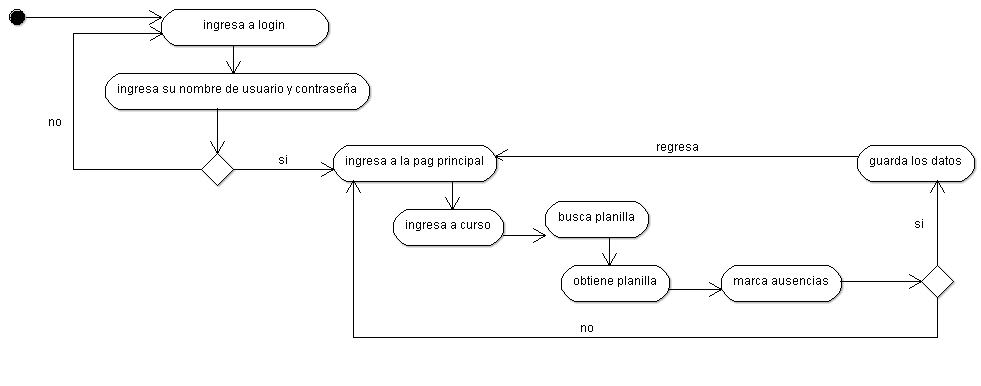
# DE

# DISEÑO

## DIAGRAMAS

**DIAGRAMA DE ACTIVIDADES**

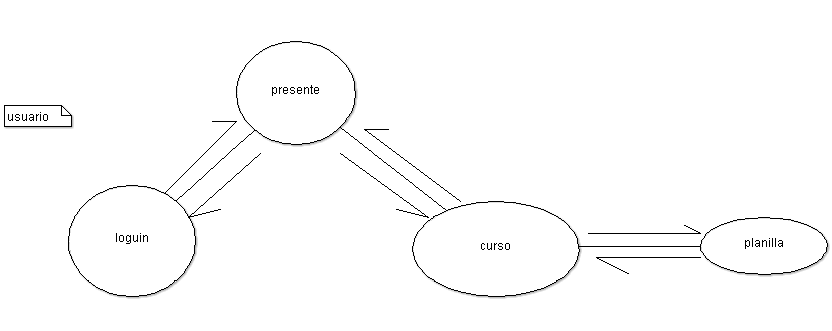
El diagrama de flujo o diagrama de actividades es la representación gráfica del algoritmo o proceso. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.

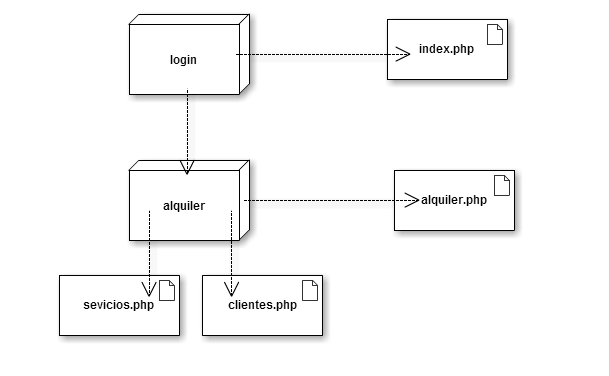
En Lenguaje Unificado de Modelado (UML), un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un diagrama de actividades muestra el flujo de control general.

**DIAGRAMA DE COLABORACION (CUMR)**

Un diagrama de colaboración en las versiones de UML 1.x es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, también llamados diagramas de comunicación, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

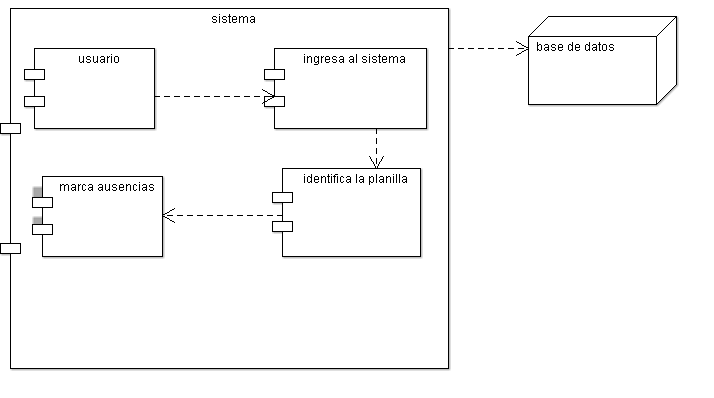
* Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.
* Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dichas implementaciónes llamadas "enlace".



**DIAGRAMA DE COMPONENTES**

Un diagrama de componentes es un diagrama tipo del Lenguaje Unificado de Modelado.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

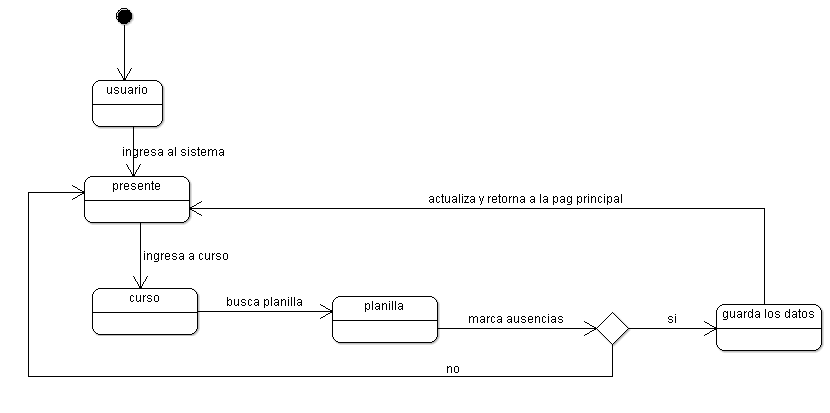


**DIAGRAMA DE ESTADOS (CUMR)**

En UML, un diagrama de estados es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso.

Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación.

El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

**DIAGRAMA DE INTERACCION**

El diagrama de interacción, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre si en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

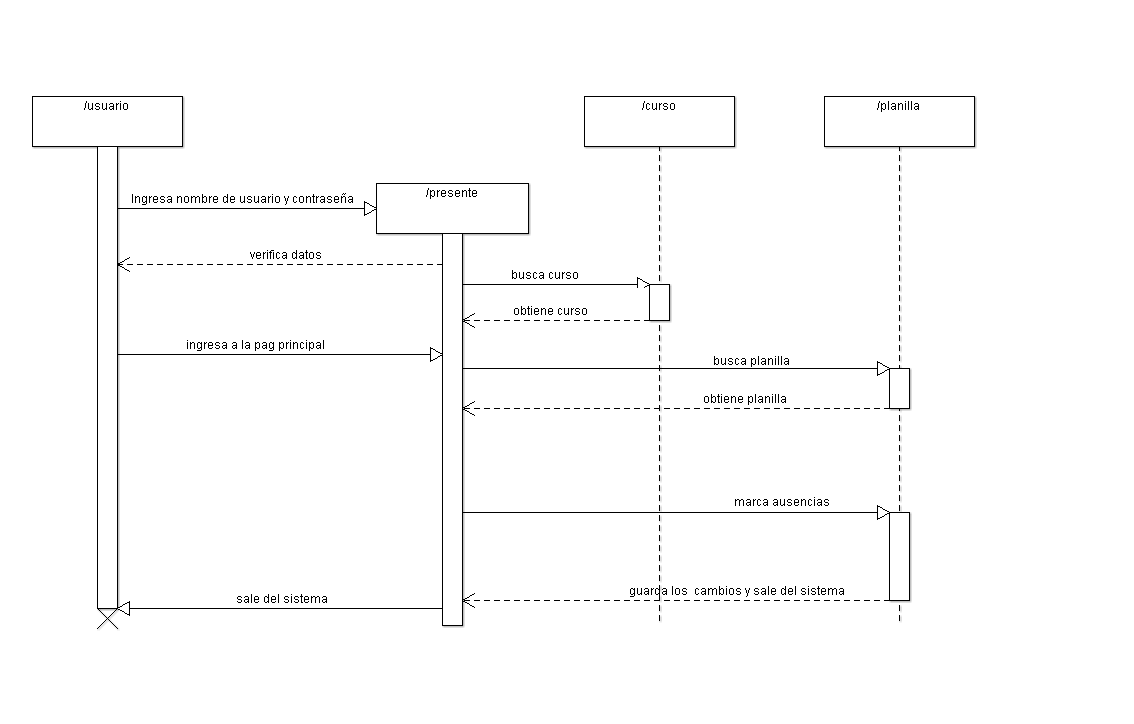
Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso (son diferentes).

Los componentes de un diagrama de interacción son:

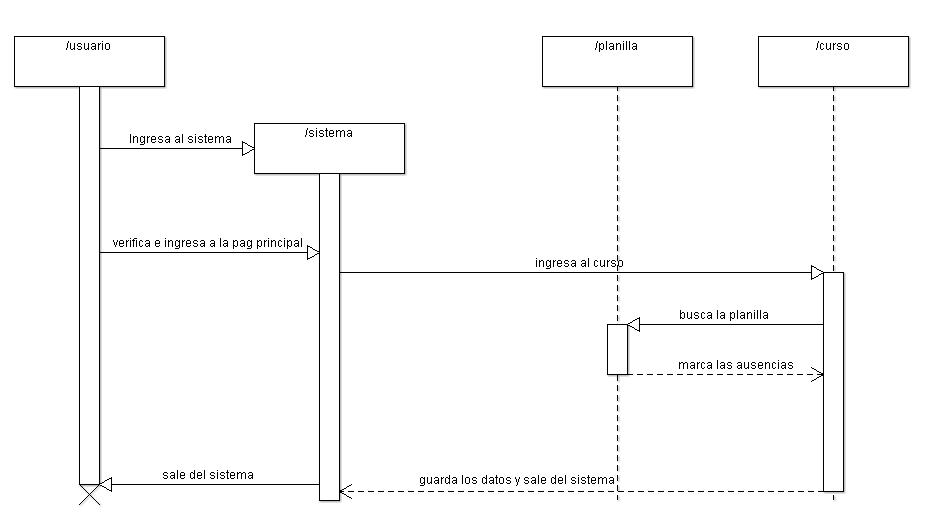
Un Objeto o Actor.

Mensaje de un objeto a otro objeto.

Mensaje de un objeto a si mismo.



**DIAGRAMA DE SECUENCIA (CUMR)**

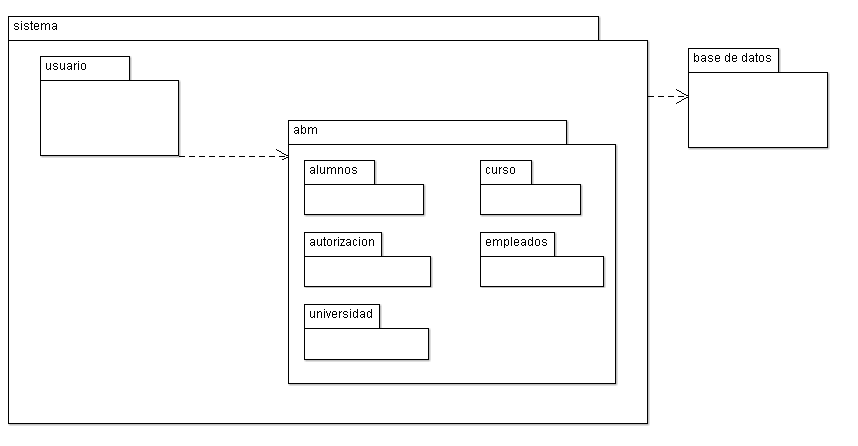
Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el Diagrama de casos de uso permite el modelado de una vista *business* del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

**DIAGRAMA DE PAQUETES**

En el Lenguaje Unificado de Modelado, un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

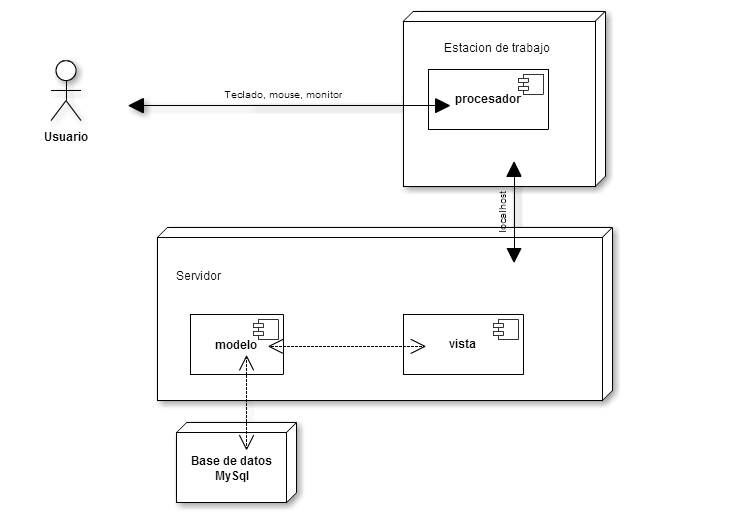


**Diagrama de despliegue**

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

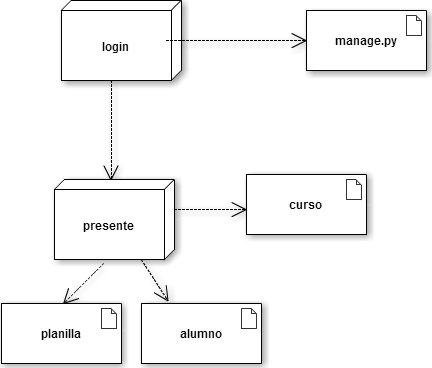
Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

La mayoría de las veces el modelado de la vista de despliegue implica modelar la topología del hardware sobre el que se ejecuta el sistema.



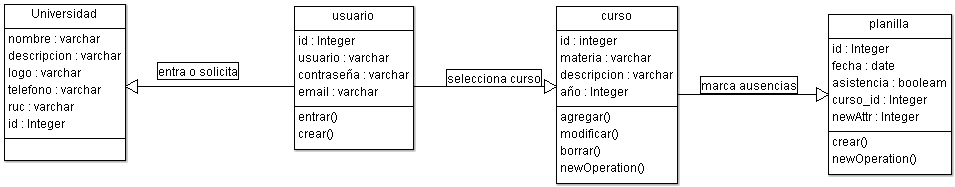
**Diagrama de implementación**

Los diagramas de implementación ofrecen una ilustración de la arquitectura física del hardware, del software y de los artefactos del sistema. Los diagramas de implementación pueden entenderse como lo contrario de los [casos de uso](http://www.altova.com/es/umodel/use-case-diagrams.html), porque ilustran la forma física del sistema, en lugar de representar conceptualmente los usuarios y dispositivos que interactúan con el sistema.



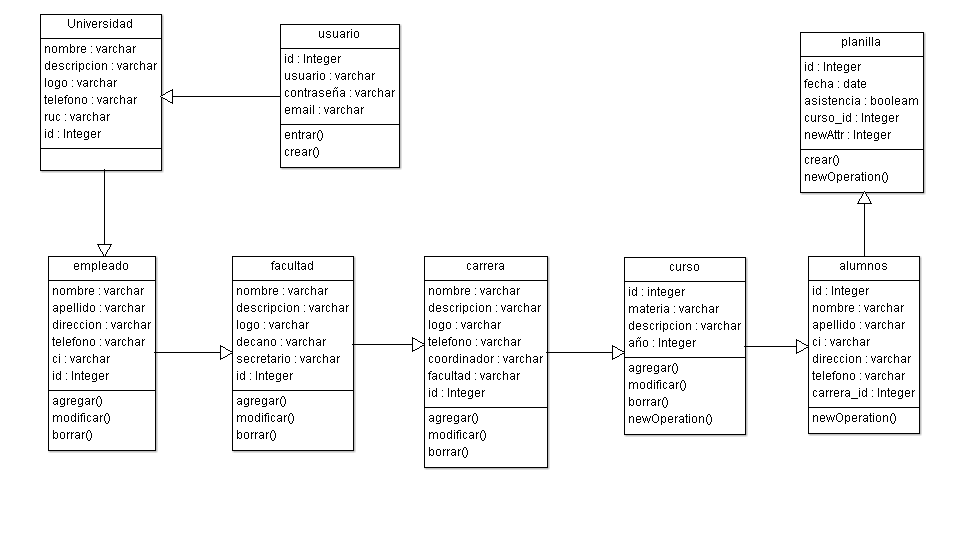
**Diagramas de objeto**

Un diagrama de Objeto se puede considerar un caso especial de un diagrama de clase. Los diagramas de objetos usan un sub conjunto de elementos de un diagrama de clase para enfatizar la relación entre las instancias de las clases en algún punto en el tiempo. Estos son útiles para entender los diagramas de clases. Estos no muestran nada diferente en su arquitectura a los diagramas de secuencia, pero reflejan multiplicidad y roles.



**Diagrama de clases**

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.



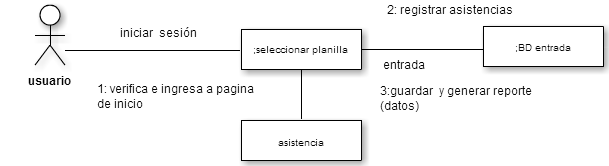
**Diagrama de comunicación**

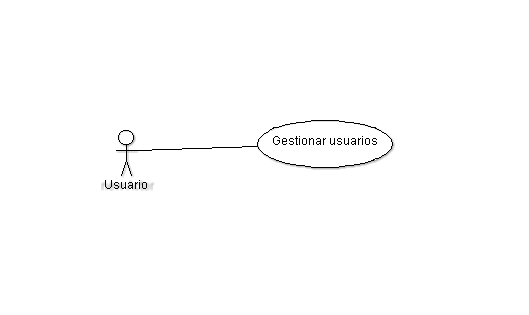
En el [Lenguaje Unificado de Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado) (UML) 2.0, un diagrama de comunicación es una versión simplificada del diagrama de colaboración.

Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de [clases](http://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)), [secuencia](http://es.wikipedia.org/wiki/Diagrama_de_secuencia), y [diagrama de casos de uso](http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso) describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

Los diagramas de comunicación y de secuencia describen información similar, y con ciertas transformaciones, pueden ser transformados unos en otros sin dificultad.

Para mantener el orden de los mensajes en un diagrama de comunicación, los mensajes son etiquetados con un número cronológico y colocado cerca del enlace por el cual se desplaza el mensaje. Leer un diagrama de comunicación conlleva comenzar en el mensaje 1.0, y seguir los mensajes desde un objeto hasta el siguiente, sucesivamente



1. **Identificar los CU**
2. ABM Usuarios
3. Ingresar al sistema
4. ABM facultad
5. ABM Carreras
6. ABM Empleados
7. ABM Alumnos
8. ABM Materias.
9. Crear planilla de asistencia
10. Registrar asistencias
11. Reporte de asistencias
    * 1. **ABM Usuarios**

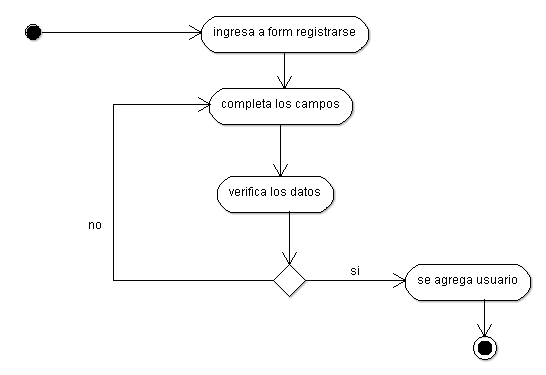
|  |  |
| --- | --- |
| **Código** | CU\_1 |
| **Nombre** | Gestionar usuario |
| **Fecha** | 27-03-15 |
| **Autor** | Patricia Fernández |
| **Actor** | Usuario |
| **Descripción** | Gestionar datos de los usuarios |
| **Precondición** | 1-Poseer datos de los usuarios |
| **Flujo Normal** | Agregar usuarios con los datos recolectados |
| **Pos condición** | Accesibilidad desde el portafolio usuarios |
| **Flujo alternativo** | Conseguir datos de los usuarios |

**Modelo de casos de uso**

|  |  |  |
| --- | --- | --- |
| **Código** | **CU\_1-CUR\_1** | |
| **Nombre** | **Gestionar usuarios - Registrar Usuario** | |
| **Fecha** | **27/03/2015** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | **Mantener ordenada y detallada metódicamente la nómina de usuarios.** | |
| **Descripción** | **Se presenta este caso de uso cuando el usuario se desea registrarse** | |
| **Precondición** | **-       El usuario debe poseer todos sus datos.** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario ingresa sus datos** |
| **2** | **El usuario ingresa una contraseña** |
| **3** | **El sistema registra al usuario** |
| **4** | **El sistema muestra todos los datos del usuario** |
|  |  |
| **Postcondición** | **Se ha registrado un nuevo usuario** | |
| **Excepciones** | **Paso** | **Acción** |
| **1** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **2** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **4** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **Un usuario puede registrarse 1 vez** | |
| **Importancia** | **Alta** | |
| **gestionar_usuario.pngComentarios** | **Una vez agregada el usuario el sistema debe mostrar los detalles de la carga** | |

Diagrama de casos de usos

Diagrama de actividades



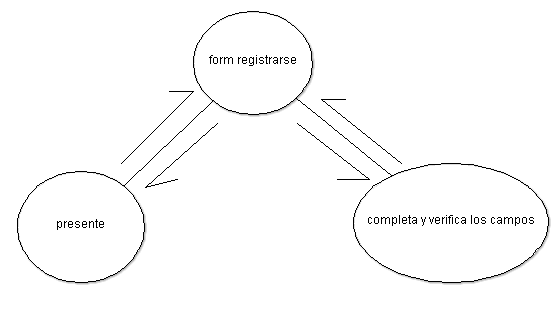
Diagrama de colaboración

Diagrama de componentes

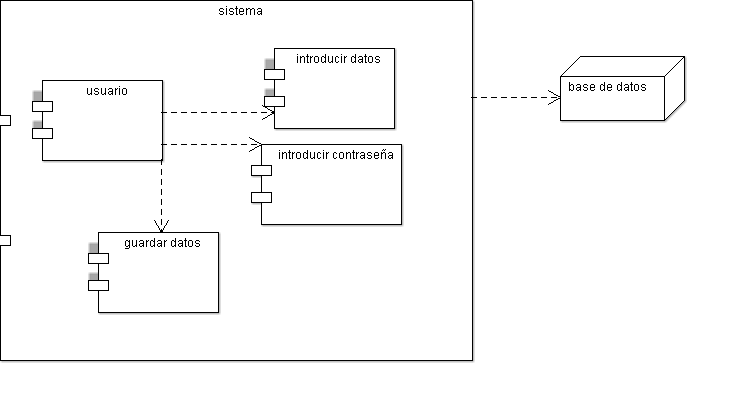


Diagrama de estado

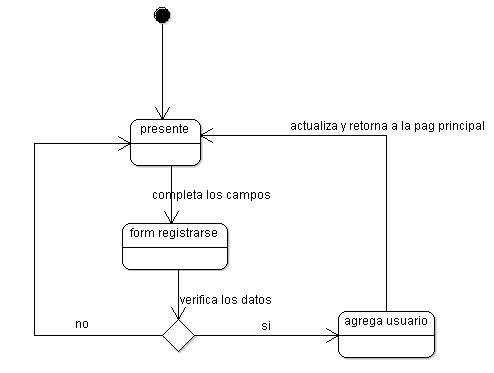
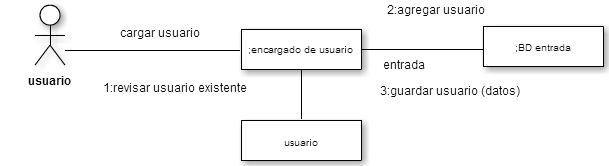


Diagrama de comunicación



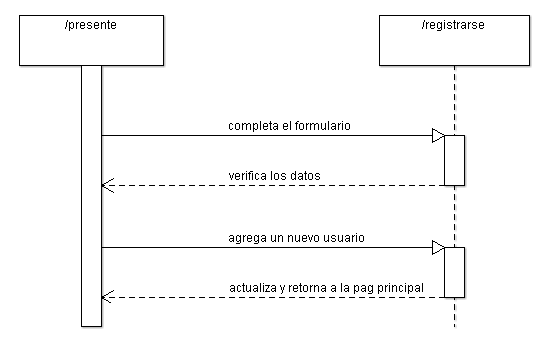
Diagrama de interacción

Diagrama de secuencia

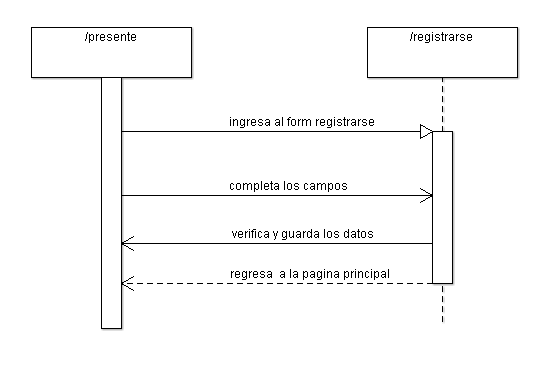
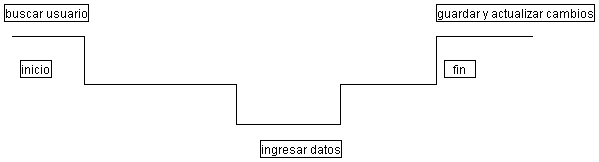


Diagrama de coordinación



|  |  |  |
| --- | --- | --- |
| **Código** | **CU\_1-CUR\_2** | |
| **Nombre** | **Gestionar usuario - Modificar Usuario** | |
| **Fecha** | **27/03/2015** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | **Mantener ordenada y detallada metódicamente la nómina de usuarios.** | |
| **Descripción** | **Se presenta este caso de uso cuando el usuario desea modificar sus datos.** | |
| **Precondición** | **-          El usuario debe estar registrado**  **-          El usuario identifica su dato o contraseña a modificar**  **-          El usuario debe poseer el dato a modificar.** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar el dato a modificar** |
| **2** | **El usuario debe identificar el dato a modificar** |
| **3** | **El usuario modifica sus datos** |
| **4** | **El usuario guarda los cambios realizados** |
| **Postcondición** | **Se ha modificado los detalles del usuario** | |
| **Excepciones** | **Paso** | **Acción** |
| **3** | **El jefe cancela el proceso, el CU queda sin efecto** |
| **4** | **Resaltar campos obligatorios** |
|  |  |
| **Frecuencia esperada** | **El usuario puede modificar sus datos n veces** | |
| **D:\facultad 2015\Ing. Sofware\casos de usos\sistema_asistencia\usuario_modificado.pngImportancia** | **Baja** | |
| **Comentarios** | **Una vez modificada el usuario el sistema debe mostrar los detalles de la carga** | |

Diagrama de casos de usos

Diagrama de actividades



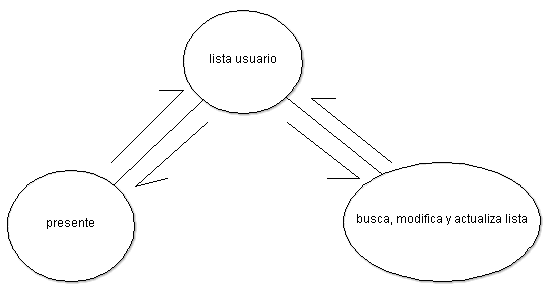
Diagrama de colaboración

Diagrama de componentes

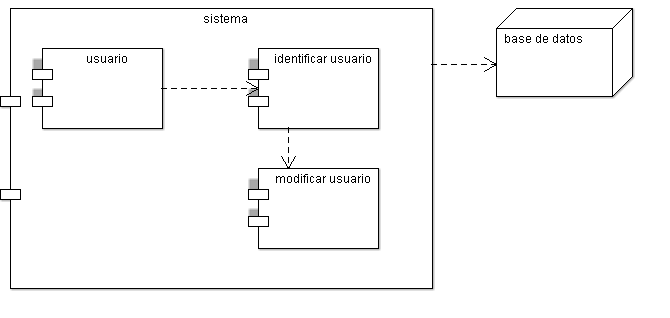


Diagrama de estado

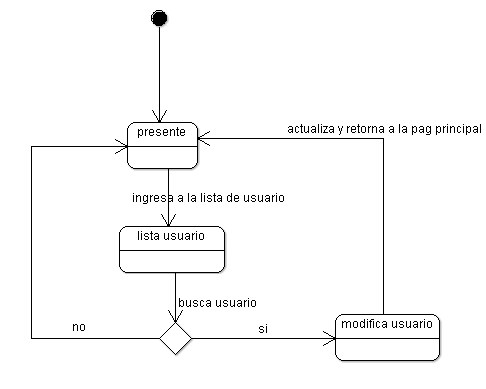
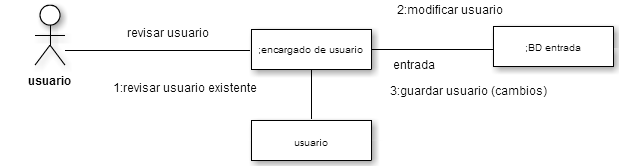


Diagrama de comunicación



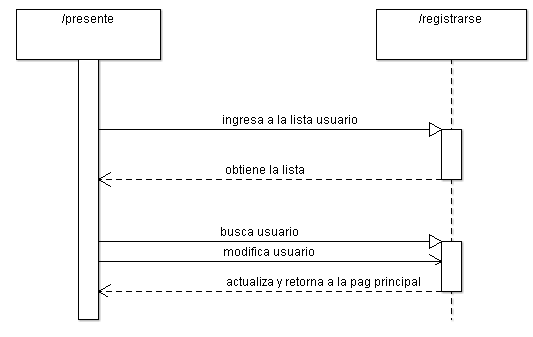
Diagrama de interacción

Diagrama de secuencia

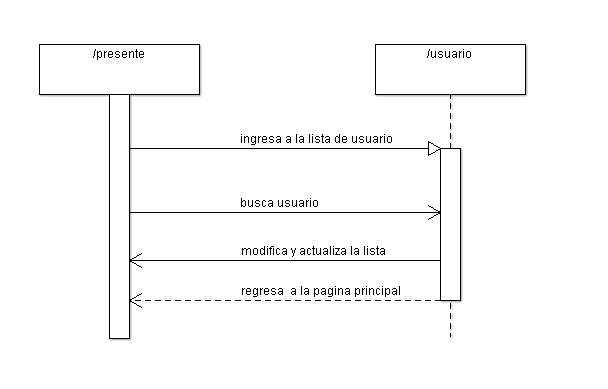
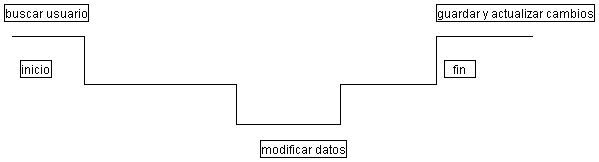


Diagrama de coordinación



|  |  |  |
| --- | --- | --- |
| **Código** | **CU\_1–CUR\_3** | |
| **Nombre** | **Gestionar usuarios – Borrar usuario** | |
| **Fecha** | **27/03/15** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | Mantener ordenada y detallada metódicamente la nómina de usuarios. | |
| **Descripción** | Se presenta este caso de uso cuando se desea eliminar un usuario. | |
| **Precondición** | * El sistema debe tener al usuario registrado * El usuario debe identificar su cuenta * El usuario posee todos sus datos | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario busca su cuenta a eliminar |
| 2 | El usuario identifica su cuenta a eliminar |
| 3 | El usuario elimina su cuenta |
| **Postcondición** | **Se ha eliminado usuario del sistema** | |
| Excepciones | **Paso** | **Acción** |
| 3 | El usuario cancela el proceso, el CU queda sin efecto |
|  |  |
| **Frecuencia esperada** | Un usuario puede darse de baja una vez | |
| **D:\facultad 2015\Ing. Sofware\casos de usos\sistema_asistencia\ingresa_usuario.pngImportancia** | Baja | |
| **Comentarios** | Una vez dado de baja el usuario el sistema debe de mostrar la pagina de login | |

Diagrama de casos de usos

Diagrama de actividades



Diagrama de colaboración

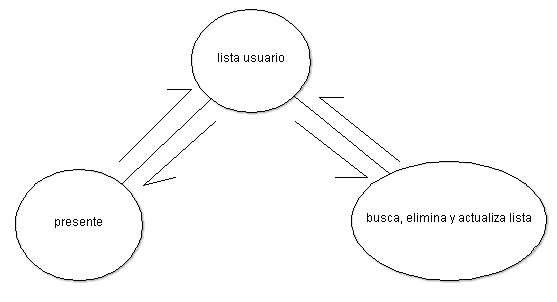


Diagrama de componentes

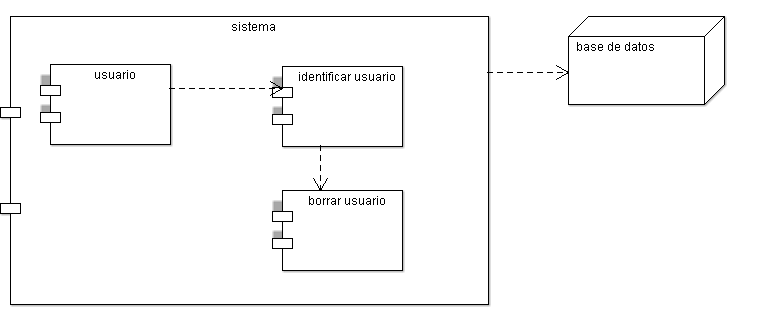


Diagrama de estado

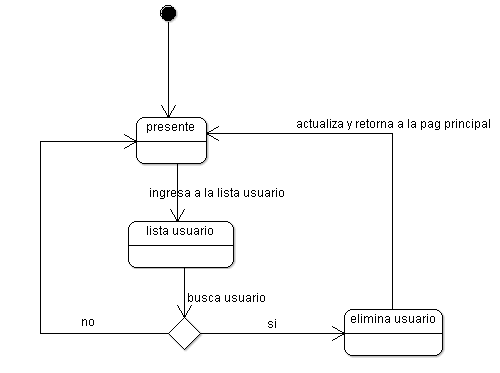


Diagrama de interacción

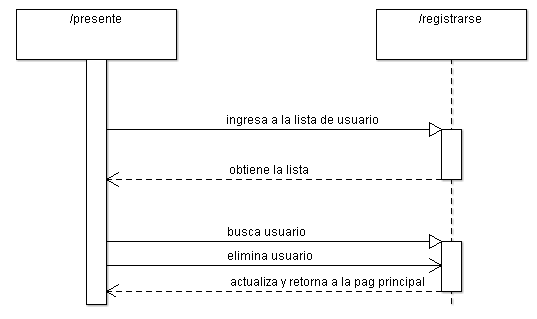


Diagrama de secuencia

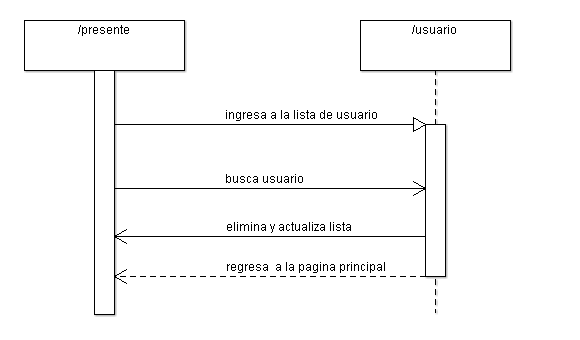


Diagrama de coordinación

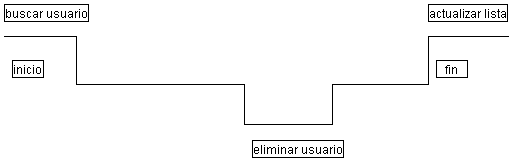
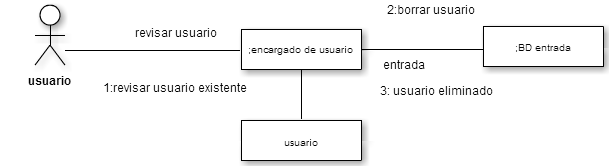


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **CU-2** | **Ingresar al sistema** | |
| **Fecha** | **07/04/2014** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | Ingresar al sistema para marcar o ver asistencias de los alumnos | |
| **Descripción** | Se presenta este caso de uso cuando un usuario desea ingresar al sistema | |
| **Precondición** | El usuario debe poseer su nombre de usuario y contraseña | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario introduce su nombre de usuario |
| 2 | El usuario introduce su contraseña |
| 3 | El usuario entra al sistema |
| **Postcondición** | **Se ha registrado un nuevo cliente** | |
| Excepciones | **Paso** | **Acción** |
| 2 | El usuario cancela el proceso, el CU queda sin efecto |
| 3 | El usuario introduce nombre de usuario o contraseña incorrecta vuelve a intentarlo |
| **Rendimiento** | Paso | Acción |
|  | 1 segundos |
| **Frecuencia esperada** | Un usuario puede intentar entrar a su sistema tres veces | |
| **Importancia** | Media | |
| **Comentarios** |  | |

* + 1. **Ingresar al sistema**

****

**Diagrama de casos de usos**

Diagrama de actividades

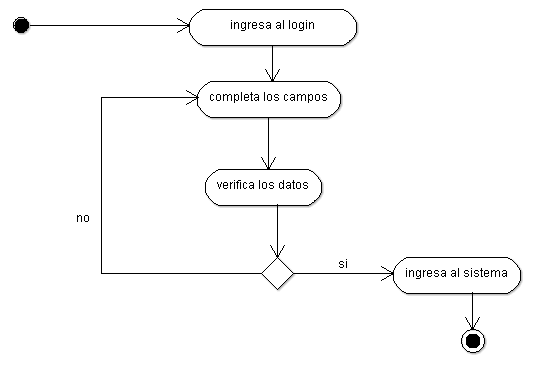


Diagrama de colaboración

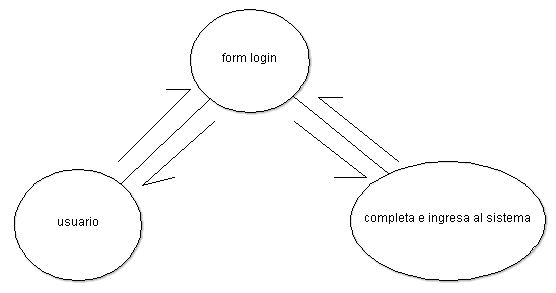


Diagrama de componentes

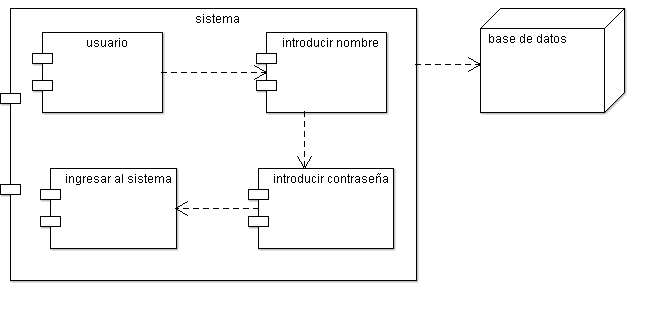


Diagrama de estado

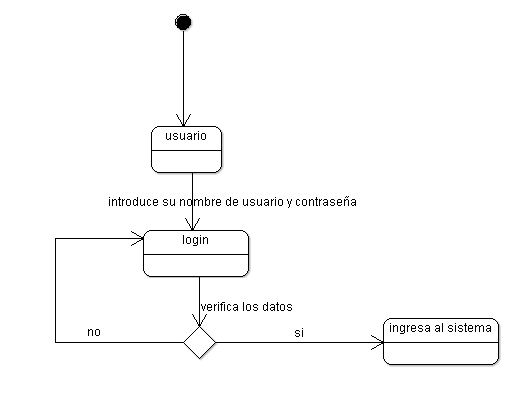


Diagrama de interacción

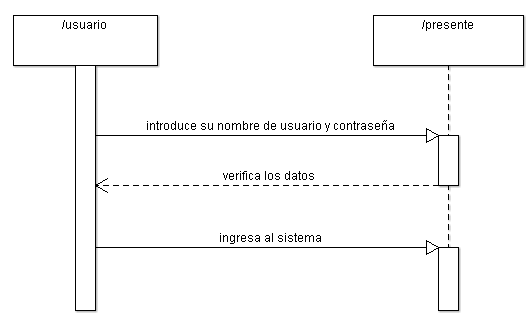


Diagrama de secuencia

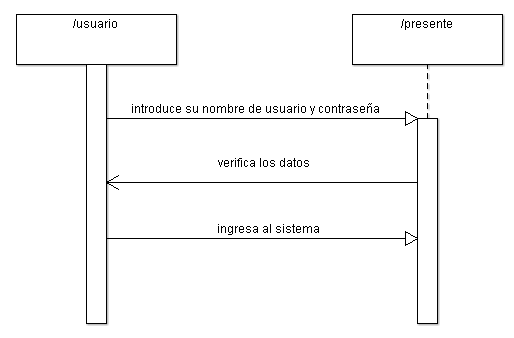
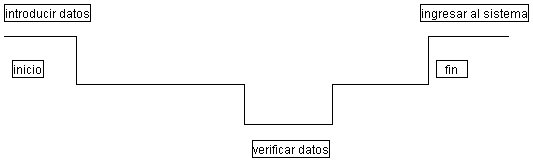
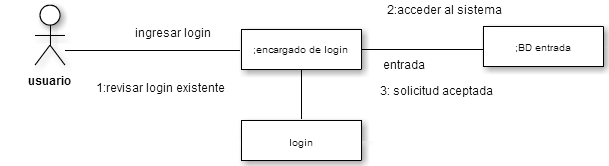


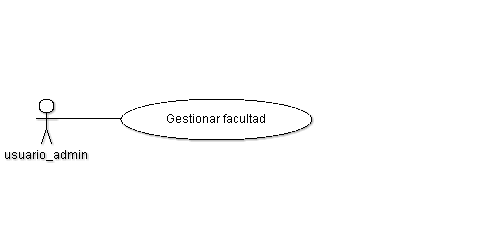
Diagrama de coordinación

****

**Diagrama de comunicación**

****

* + 1. **ABM Facultad**

****

|  |  |
| --- | --- |
| **Código** | CU\_3 |
| **Nombre** | Gestionar facultad |
| **Fecha** | 27-03-15 |
| **Actor** | Cristina Alvarez |
| **Autores** | Usuario (admin) |
| **Descripción** | Gestionar datos de la facultad |
| **Precondición** | 1-Poseer datos de la facultad |
| **Flujo Normal** | Administrar facultades |
| **Pos condición** | Accesibilidad desde el portafolio usuarios |
| **Flujo alternative** | Conseguir datos de las recepciones |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| **Código** | |  | | |
| **Nombre** | | **Agregar Facultad** | | |
| **Fecha** | | **09/04/2015** | | |
| **Versión** | | **2.0** | | |
| **Autor** | | **Cristina Alvarez** | | |
| **Actores** | | **Usuario-Administrador** | | |
| **Objetivos Asociados** | | **Mantener ordenado y detallada metódicamente los registros de las facultades** | | |
| **Descripción** | | **Se presenta este caso de uso cuando se desee agregar o registrar una facultad** | | |
| **Precondición** | | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | | |
| **Secuencia**  **Normal** | | **Paso** | | **Acción** |
| **1** | | **El usuario introduce los datos de la facultad** |
| **2** | | **El sistema registra la facultad** |
| **3** | | **El sistema muestra los datos de la facultad** |
| **Postcondición** | | **Se ha registrado una nueva facultad** | | |
| **Excepciones** | | **Paso** | | **Acción** |
| **4** | | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | | **La facultad puede registrarse una vez** | | |
| **Importancia** | | **Alta** | | |
| **Comentarios** | |  | | |
|  | | |
| Diagrama de casos de usos  C:\Users\MEC\Desktop\Cris_Analisis\aFacultad.png  Diagrama de actividades  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\actividades\5.png  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\colaboracion\5.pngDiagrama de colaboración  Diagrama de componentes  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\diagrama_componentes\5.png  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\estado\5.pngDiagrama de estado  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\interaccion\5.pngDiagrama de interacción  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\secuencia\5.pngDiagrama de secuencia  Diagrama de coordinación  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\zcoordinacion\5.gif  D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\comunicacion\diagrama de comunicacion\agregar facultad.pngDiagrama de comunicación | | |
| **Código** |  | | | |
| **Nombre** | **Modificar Facultad** | | | |
| **D:\facultad 2015\primer semestre\Ing. Sofware\diagramas\diagrama_componentes\5.pngFecha** | **09/04/2015** | | | |
| **Versión** | **2.0** | | | |
| **Autor** | **Cristina Alvarez** | | | |
| **Actores** | **Usuario-Administrador** | | | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las facultades** | | | |
| **Descripción** | **Se presenta este caso de uso cuando se desee modificar datos de una facultad** | | | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | | | |
| **Secuencia**  **Normal** | **Paso** | | **Acción** | |
| **1** | | **El usuario debe buscar el dato a modificar** | |
| **2** | | **El usuario debe identificar el dato a modificar** | |
| **3** | | **El usuario modifica los datos** | |
| **Postcondición** | **Se ha modificado una facultad** | | | |
| **Excepciones** | **Paso** | | **Acción** | |
| **4** | | **El usuario cancela el proceso, el CU queda sin efecto** | |
| **5** | | **Resaltar campos obligatorios** | |
| **Frecuencia esperada** | **La facultad puede ser modificada n veces** | | | |
| **Importancia** | **Alta** | | | |
| **Comentarios** |  | | | |

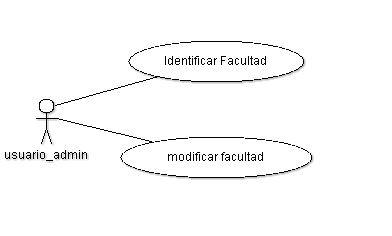


Diagrama de actividades

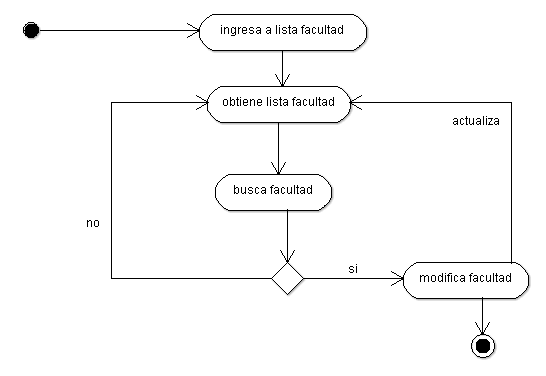


Diagrama de colaboración

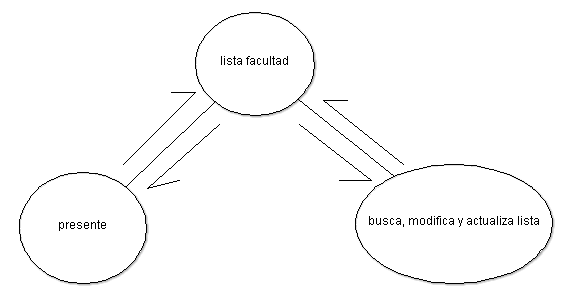


Diagrama de componentes

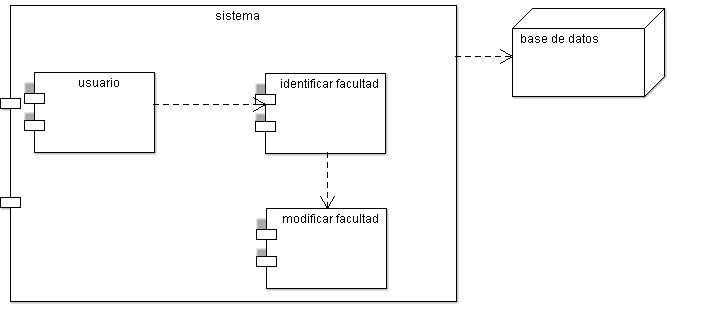


Diagrama de estado

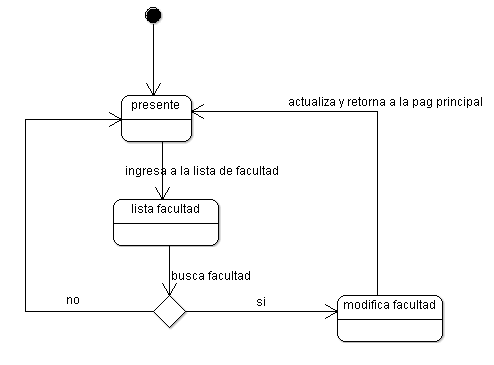


Diagrama de interacción

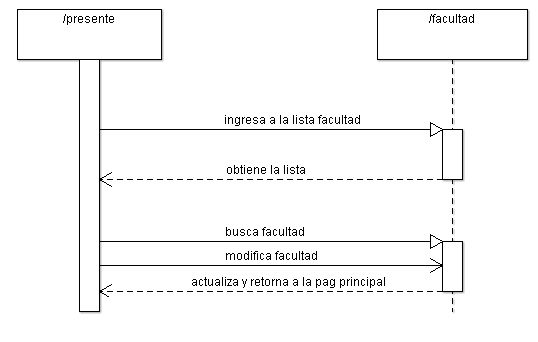


Diagrama de secuencia

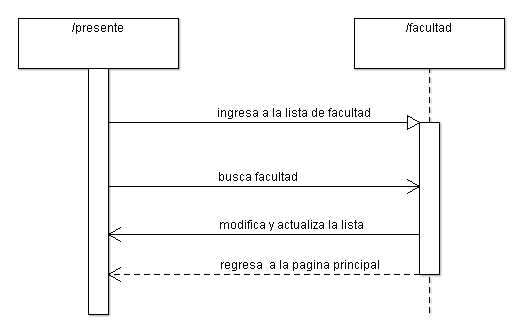


Diagrama de coordinación

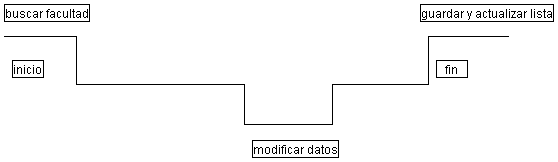
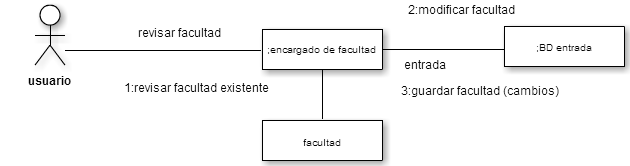


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Eliminar Facultad** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las facultades** | |
| **Descripción** | **Se presenta este caso de uso cuando se requiera eliminar una facultad** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar la facultad a eliminar** |
| **2** | **El usuario debe identificar la facultad a eliminar** |
| **3** | **El usuario elimina la facultad** |
| **Postcondición** | **Se ha eliminado una facultad** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La facultad puede ser eliminada una vez** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

Diagrama de casos de usos

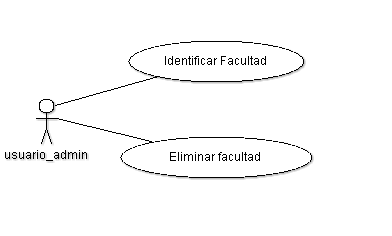


Diagrama de actividades

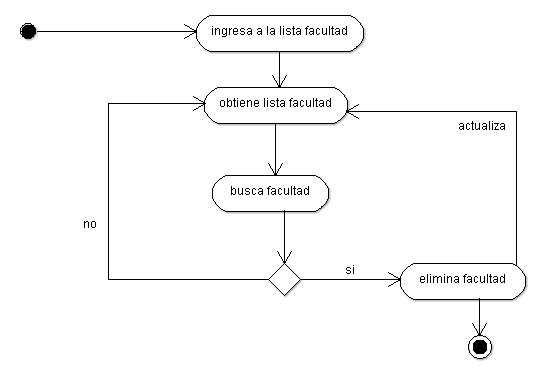


Diagrama de colaboración

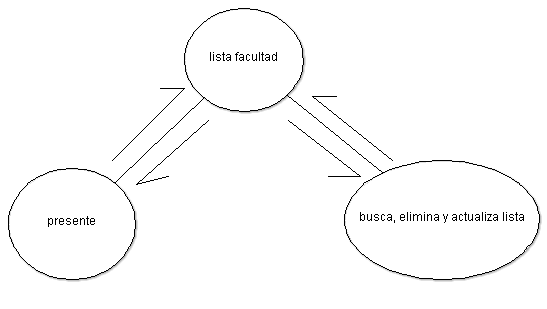


Diagrama de componentes

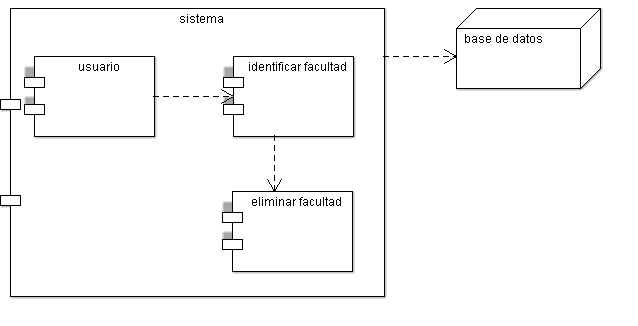


Diagrama de estado

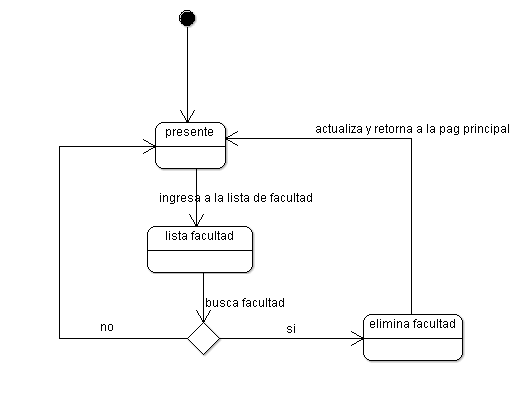


Diagrama de interacción

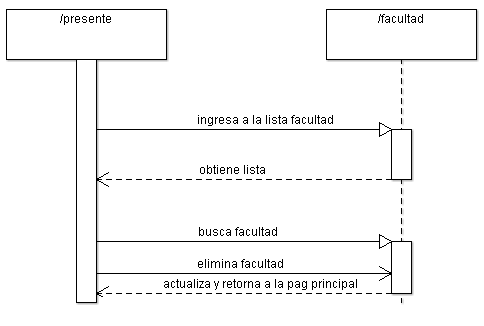


Diagrama de secuencia

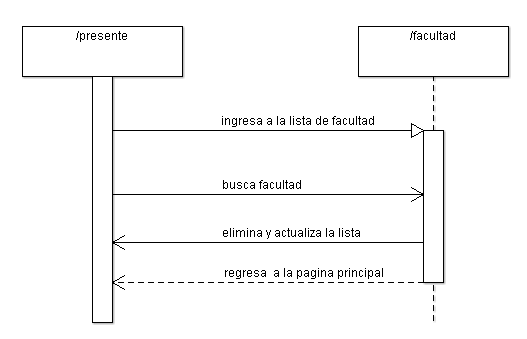


Diagrama de coordinación

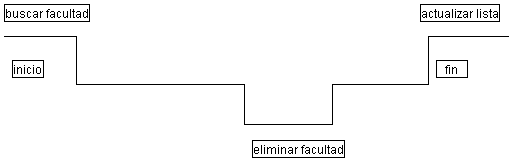
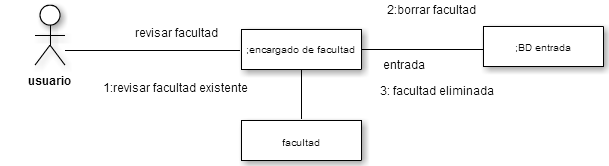


Diagrama de comunicación



* + 1. **ABM Carreras**



|  |  |
| --- | --- |
| **Código** | CU\_4 |
| **Nombre** | Gestionar carreras |
| **Fecha** | 09-04-15 |
| **Actor** | Cristina Alvarez |
| **Autores** | Usuario (admin) |
| **Descripción** | Gestionar datos de las carreras |
| **Precondición** | 1-Poseer datos de la carrera |
| **Flujo Normal** | Administrar carreras |
| **Pos condición** | Accesibilidad desde el portafolio usuarios |
| **Flujo alternative** | Conseguir datos de las recepciones |

|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Agregar Carrera** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las carreras** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee agregar o registrar una carrera** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario introduce los datos de la carrera** |
| **2** | **El sistema registra la carrera** |
| **3** | **El sistema muestra los datos de la carrera** |
| **Postcondición** | **Se ha registrado una nueva facultad** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La carrera puede registrarse una vez** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

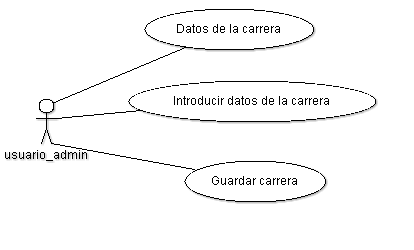


Diagrama de actividades

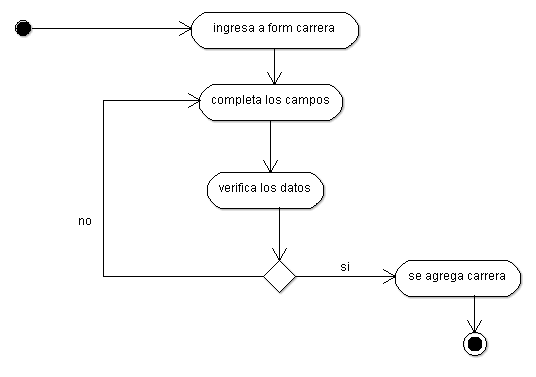


Diagrama de colaboración

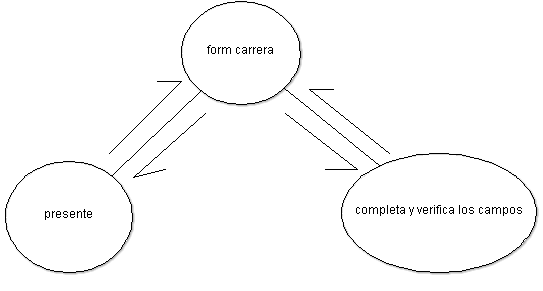


Diagrama de componentes

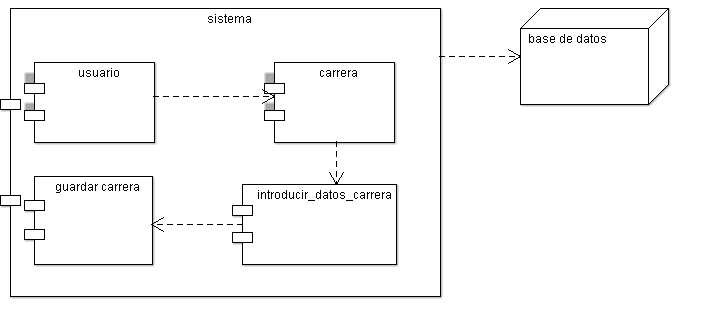


Diagrama de estado

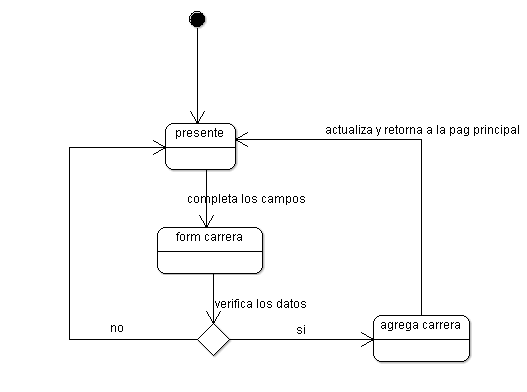


Diagrama de interacción

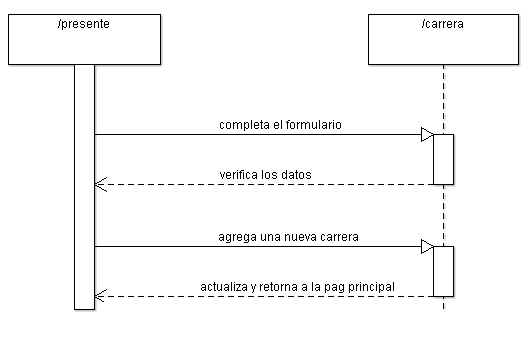


Diagrama de secuencia

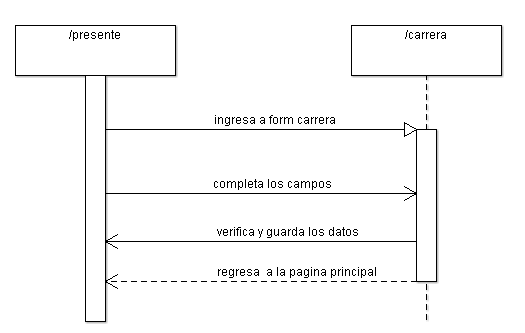


Diagrama de coordinación

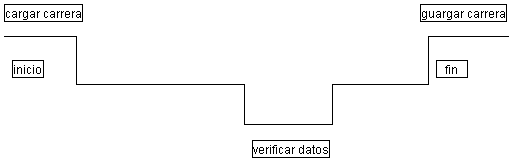
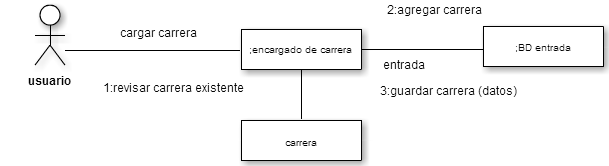


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Modificar Carrera** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las carreras** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee modificar datos de una carrera** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar el dato a modificar** |
| **2** | **El usuario debe identificar el dato a modificar** |
| **3** | **El usuario modifica los datos** |
| **Postcondición** | **Se ha modificado una carrera** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La carrera puede ser modificada n veces** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

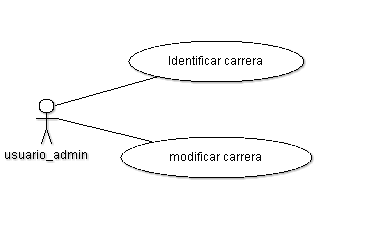


Diagrama de actividades

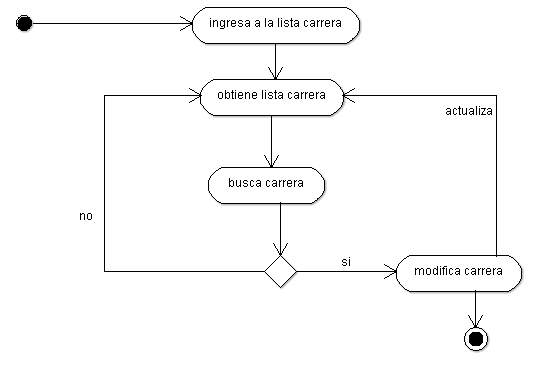


Diagrama de colaboración

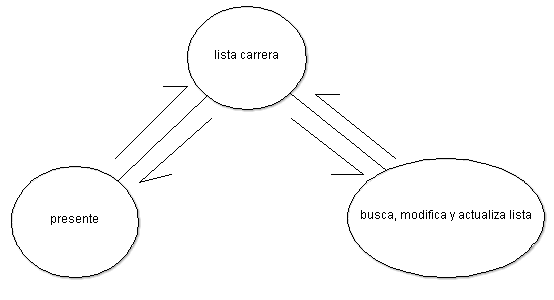


Diagrama de componentes

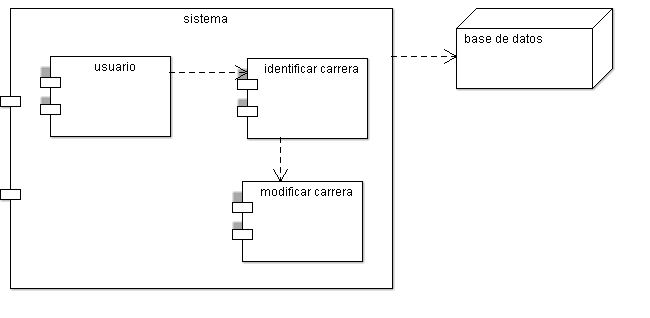


Diagrama de estado

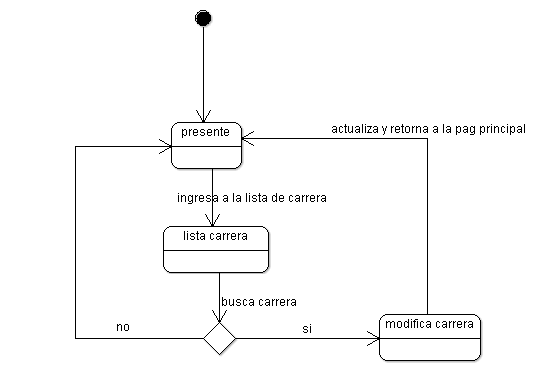


Diagrama de interacción

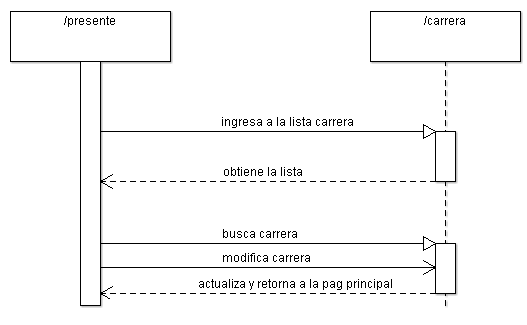


Diagrama de secuencia

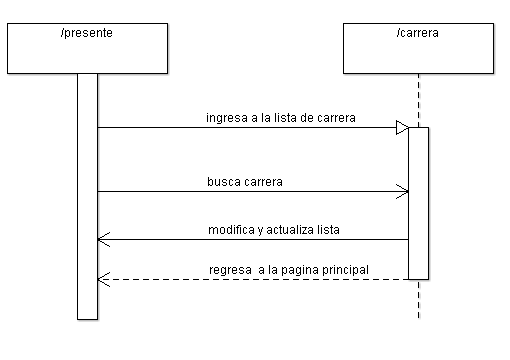


Diagrama de coordinación

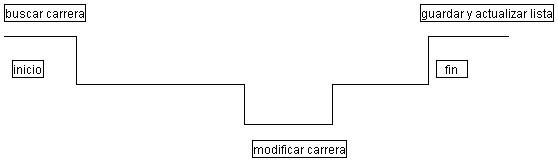
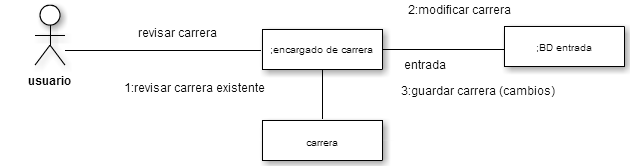


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Eliminar carrera** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las carreras** | |
| **Descripción** | **Se presenta este caso de uso cuando se requiera eliminar una carrera** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar la carrera a eliminar** |
| **2** | **El usuario debe identificar la carrera a eliminar** |
| **3** | **El usuario elimina la carrera seleccionada** |
| **Postcondición** | **Se ha eliminado una carrera** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La carrera puede ser eliminada una vez** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

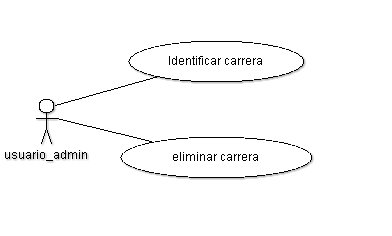


Diagrama de actividades

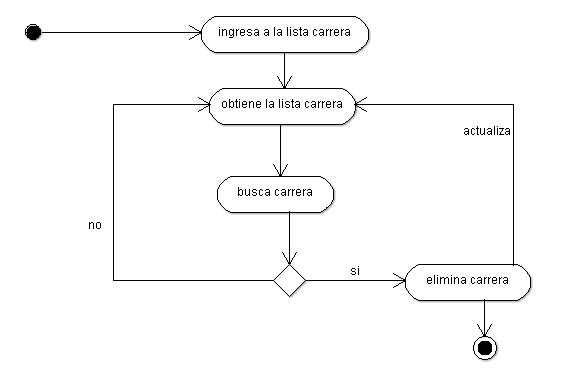


Diagrama de colaboración

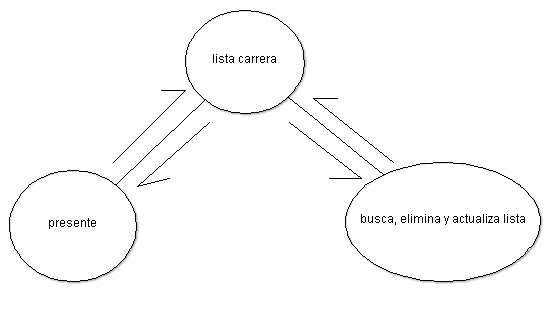


Diagrama de componentes

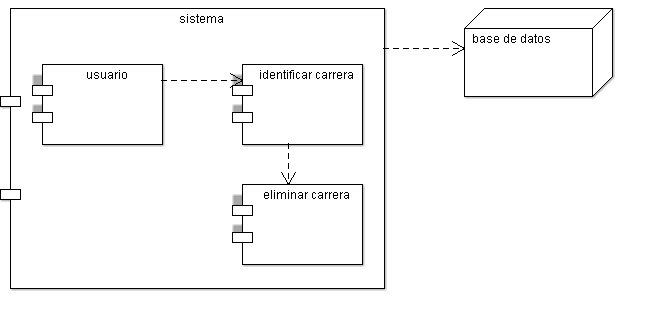


Diagrama de estado

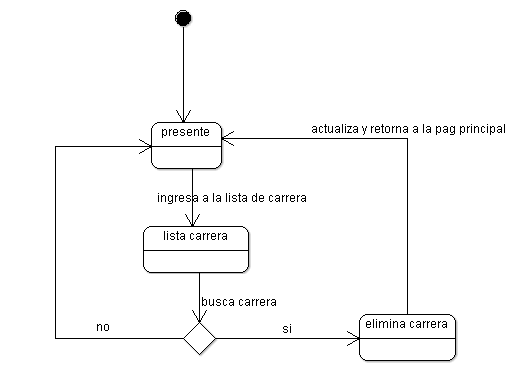


Diagrama de interacción

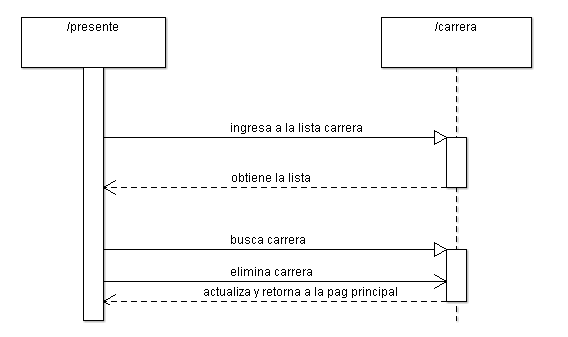


Diagrama de secuencia

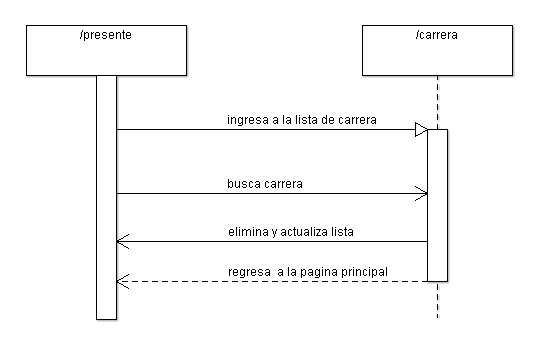


Diagrama de coordinación

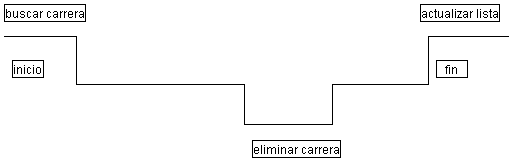
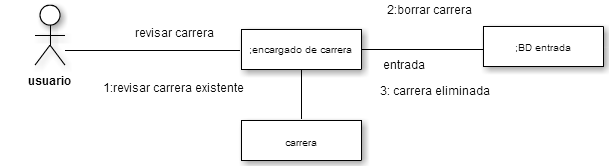
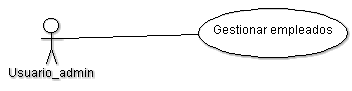


Diagrama de comunicación



* + 1. **ABM Empleado**



|  |  |
| --- | --- |
| **Código** | CU\_5 |
| **Nombre** | Gestionar empleados |
| **Fecha** | 10-04-15 |
| **Actor** | Patricia Fernández |
| **Autores** | Usuario (admin) |
| **Descripción** | Gestionar datos de las empleados |
| **Precondición** | 1-Poseer datos de los empleados |
| **Flujo Normal** | Administrar empleados |
| **Pos condición** | Accesibilidad desde el portafolio empleados |
| **Flujo alternativo** | Conseguir datos de los empleados |

|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Agregar empleados** | |
| **Fecha** | **10/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de los empleados** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee agregar o registrar un empleado** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario introduce los datos del empleado** |
| **2** | **El sistema registra el empleado** |
| **3** | **El sistema muestra los datos del empleado** |
| **Postcondición** | **Se ha registrado una nuevo profesor** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **Un profesor puede registrarse una vez** | |
| **Importancia** | **Media** | |
| **Comentarios** |  | |

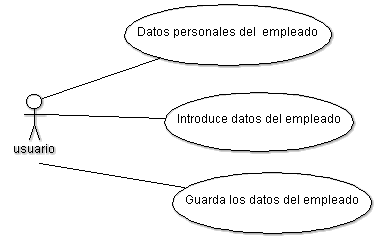


Diagrama de actividades

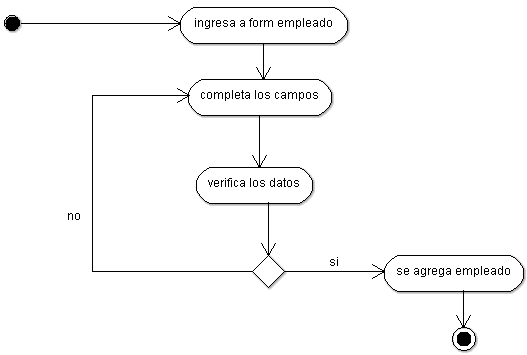


Diagrama de colaboración

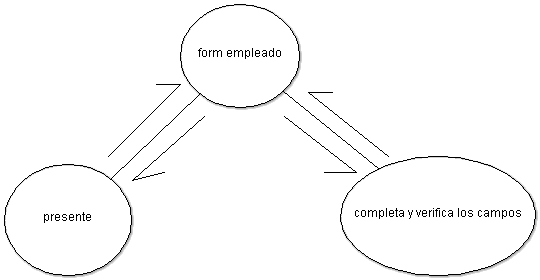


Diagrama de componentes

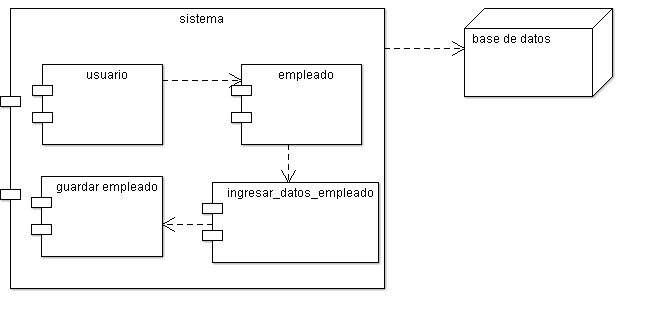


Diagrama de estado

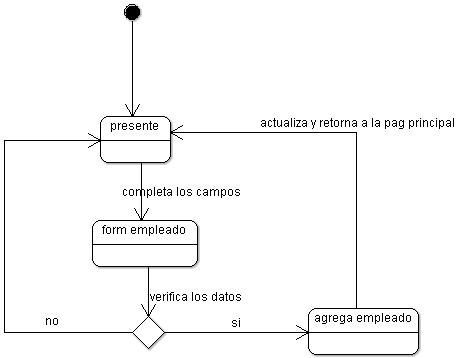


Diagrama de interacción

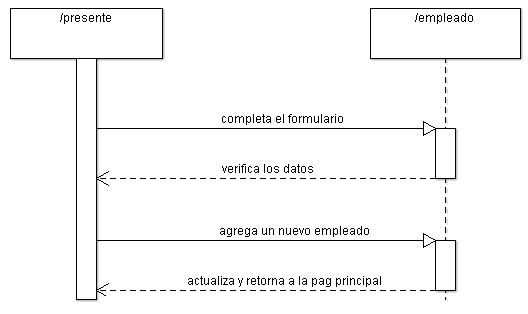


Diagrama de secuencia

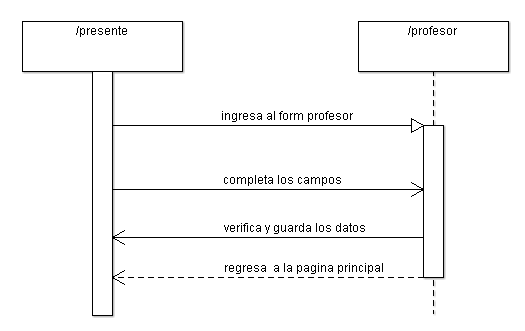


Diagrama de coordinación

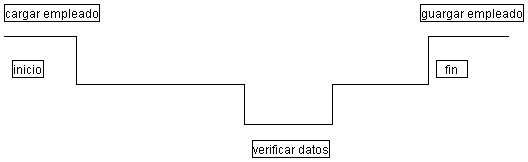
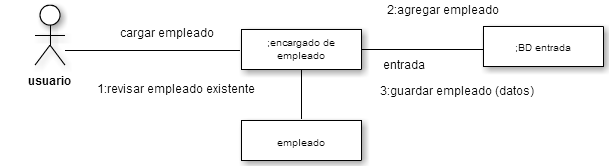


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Modificar empleado** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de los empleados** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee modificar datos de un empleado** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar el dato a modificar** |
| **2** | **El usuario debe identificar el dato a modificar** |
| **3** | **El usuario modifica los datos** |
| **Postcondición** | **Se ha modificado un empleado** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **Un empleado puede ser modificada n veces** | |
| **Importancia** | **Media** | |
| **comentarios** |  | |

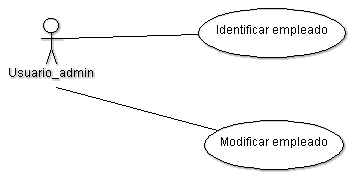


Diagrama de actividades

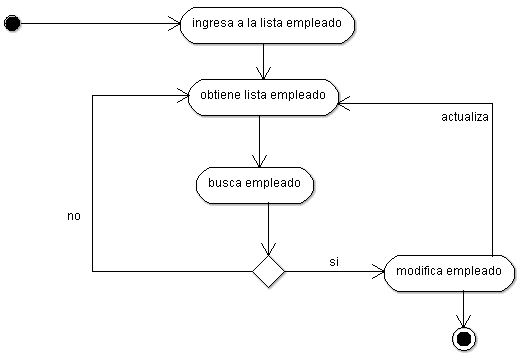


Diagrama de colaboración

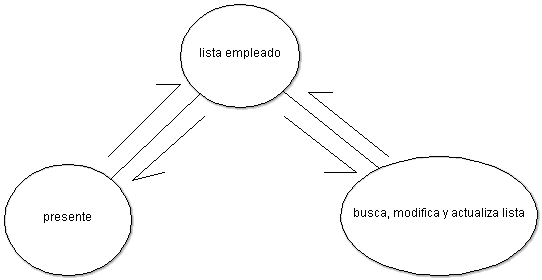


Diagrama de componentes

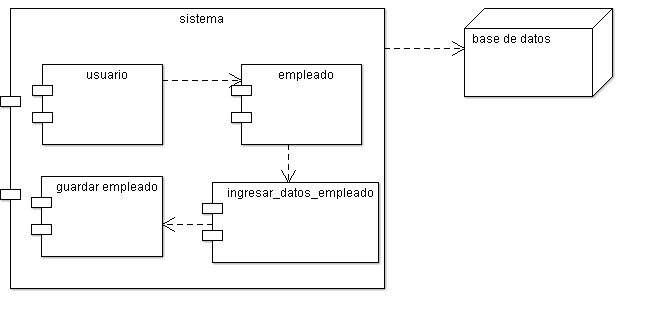


Diagrama de estado

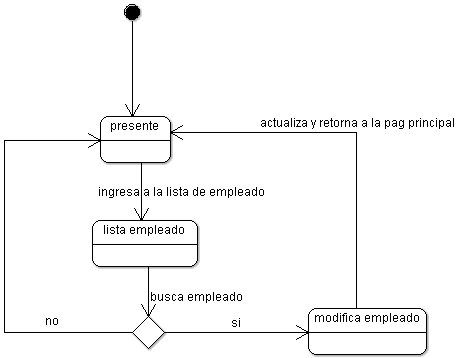


Diagrama de interacción

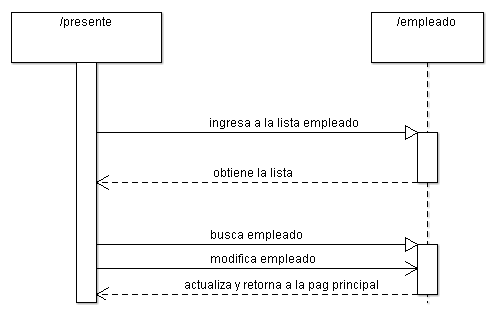


Diagrama de secuencia

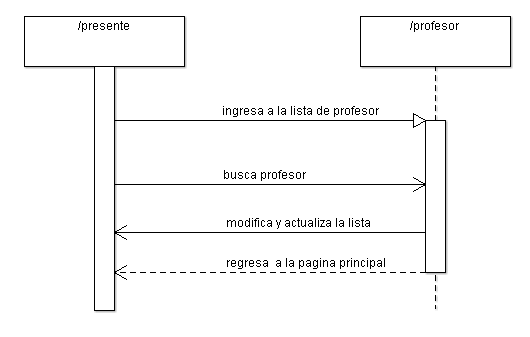


Diagrama de coordinación

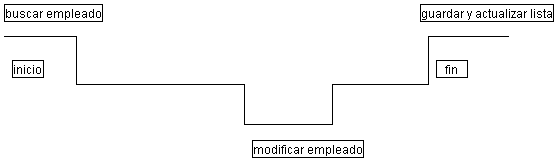
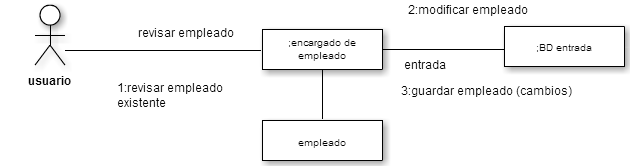


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Eliminar empleado** | |
| **Fecha** | **10/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de los empleados** | |
| **Descripción** | **Se presenta este caso de uso cuando se requiera eliminar un empleado** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El debe tener los datos del empleado a eliminar** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar el empleado a eliminar** |
| **2** | **El usuario debe identificar el empleado a eliminar** |
| **3** | **El usuario elimina el empleado seleccionado** |
| **Postcondición** | **Se ha eliminado un empleado** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **Un empleado puede ser eliminado una vez** | |
| **Importancia** | **Media** | |
| **Comentarios** |  | |

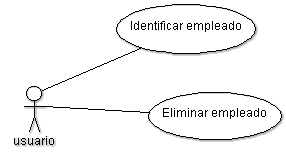


Diagrama de actividades

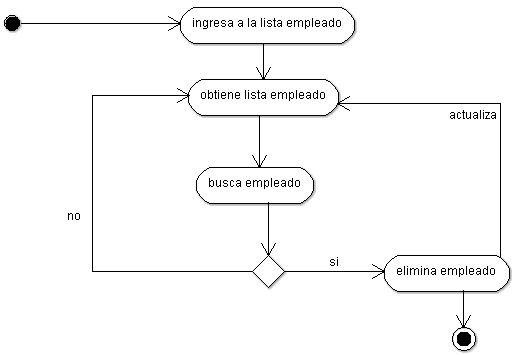


Diagrama de colaboración

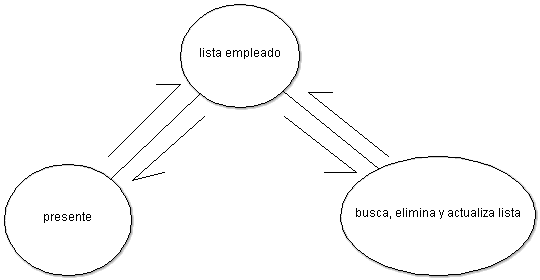


Diagrama de componentes

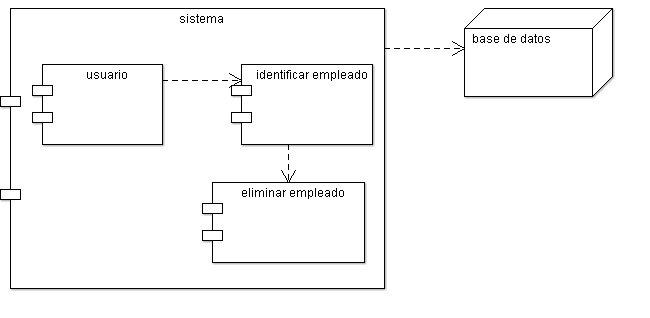


Diagrama de estado

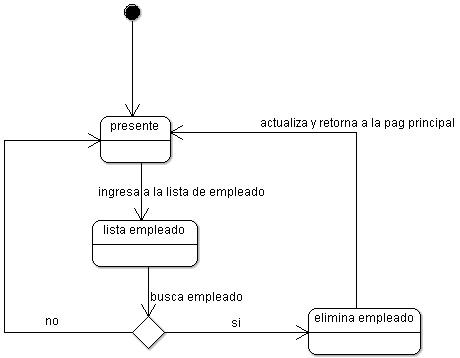


Diagrama de interacción

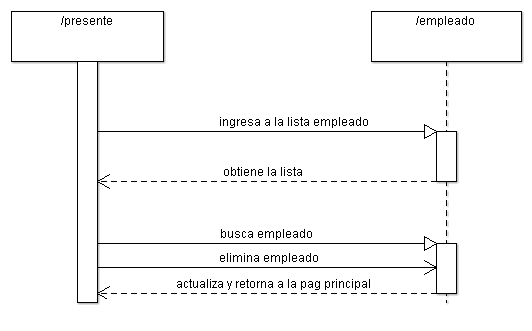


Diagrama de secuencia

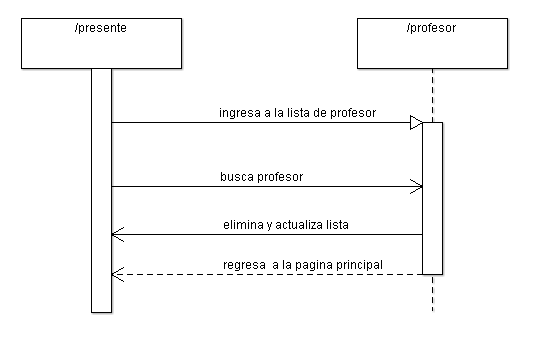


Diagrama de coordinación

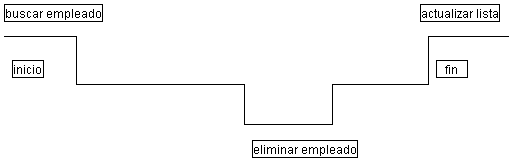
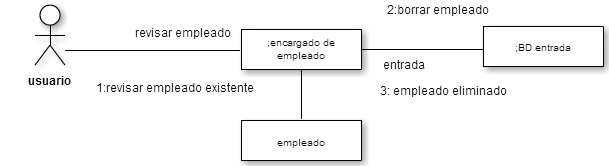
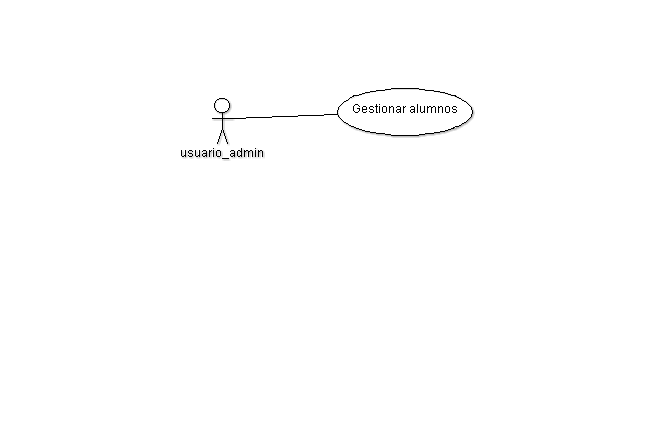


Diagrama de comunicación



* + 1. ******ABM Alumnos**

|  |  |
| --- | --- |
| **Código** | CU\_6 |
| **Nombre** | Gestionar alumnos |
| **Fecha** | 07-04-15 |
| **Autor** | Patricia Fernández |
| **Actor** | Usuario |
| **Descripción** | Gestionar datos de los alumnos |
| **Precondición** | 1-Poseer datos de los alumnos |
| **Flujo Normal** | Agregar alumnos con los datos recolectados |
| **Pos condición** | Accesibilidad desde el portafolio alumnos |
| **Flujo alternativo** | Conseguir datos de los alumnos |

|  |  |  |
| --- | --- | --- |
| **CU-3** | **Agregar Alumno** | |
| **Fecha** | **07/04/2014** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | Mantener ordenada y detallada metódicamente la nómina de alumnos matriculados. | |
| **Descripción** | Se presenta este caso de uso cuando el usuario desea registrar un nuevo alumno | |
| **Precondición** | El usuario debe poseer todos los datos del alumno. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario solicita todos los datos del alumno |
| 2 | El usuario busca al alumno |
| 3 | El usuario introduce todos los datos del alumno |
| 4 | El sistema registra al alumno |
| 5 | El sistema muestra todos los datos del alumno |
| **Postcondición** | **Se ha registrado un nuevo cliente** | |
| Excepciones | **Paso** | **Acción** |
| 2 | El usuario cancela el proceso, el CU queda sin efecto |
| 3 | El usuario no encuentra al alumno, sigue en el paso 3 |
| **Rendimiento** | Paso | Acción |
| 4 | 2 segundos |
| **Frecuencia esperada** | Un usuario puede agregar un alumno n veces | |
| **Importancia** | Alta | |
| **D:\facultad 2015\Ing. Sofware\casos de usos\sistema_asistencia\guardar_alumno.pngComentarios** |  | |

**Diagrama de Caso de Uso**

Diagrama de actividades

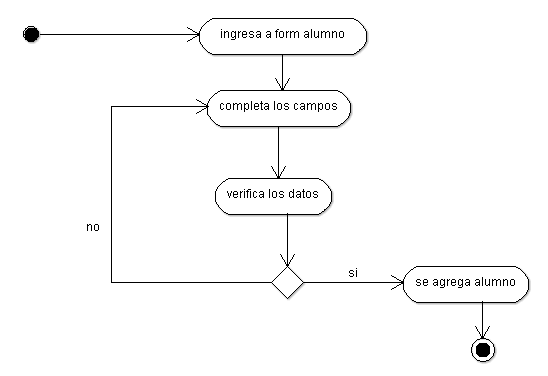


Diagrama de colaboración

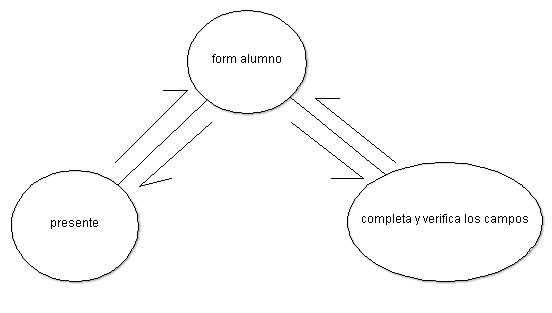


Diagrama de componentes

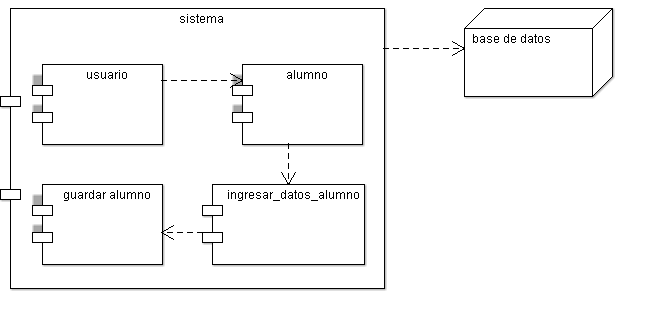


Diagrama de estado

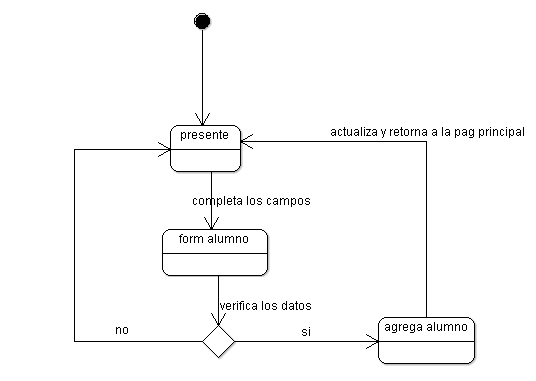


Diagrama de interacción

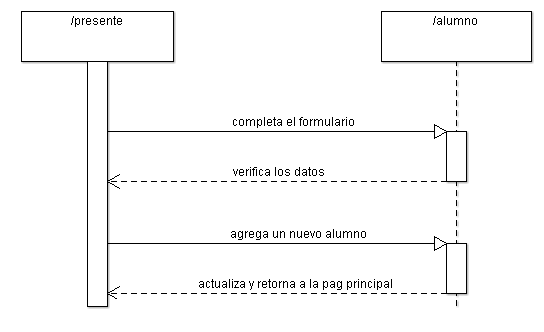


Diagrama de secuencia

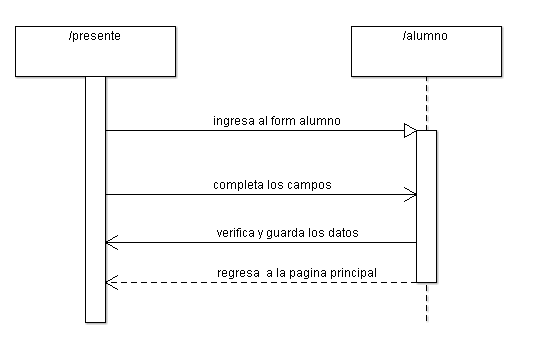
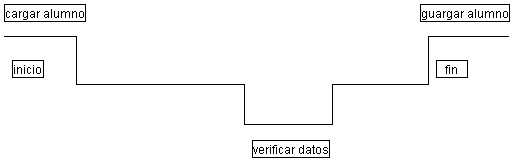
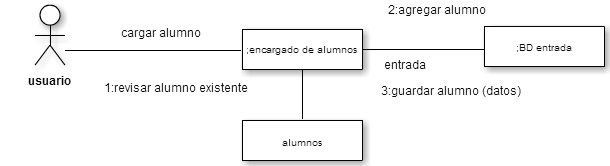


Diagrama de coordinación

****

**Diagrama de comunicación**

****

|  |  |  |
| --- | --- | --- |
| **CU-3** | **Modificar Alumno** | |
| **Fecha** | **07/04/2014** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | Mantener ordenada y detallada metódicamente la nómina de alumnos. | |
| **Descripción** | Se presenta este caso de uso cuando el usuario desea modificar datos del alumno de la lista. | |
| **Precondición** | * El sistema debe tener al alumno registrado. * El usuario debe identificar al alumno a eliminar. * El usuario debe poseer todos los datos del alumnos. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario debe buscar el alumno a modificar |
| 2 | El usuario debe identificar al alumno a modificar |
| 3 | El usuario modifica los datos del alumno |
| 4 | El usuario guarda los cambios realizados |
| **Postcondición** | **Se ha modificado los datos del cliente** | |
| Excepciones | **Paso** | **Acción** |
| 2 | El usuario no encuentra al alumno, cancela el proceso |
| 3 | El usuario cancela el proceso, el CU queda sin efecto |
|  |  |
| **Rendimiento** | Paso | **Acción** |
| 4 | **2 segundos** |
| **Frecuencia esperada** | El usuario puede modificar a un mismo alumno n veces | |
| **Importancia** | Media | |
| **Comentarios** |  | |

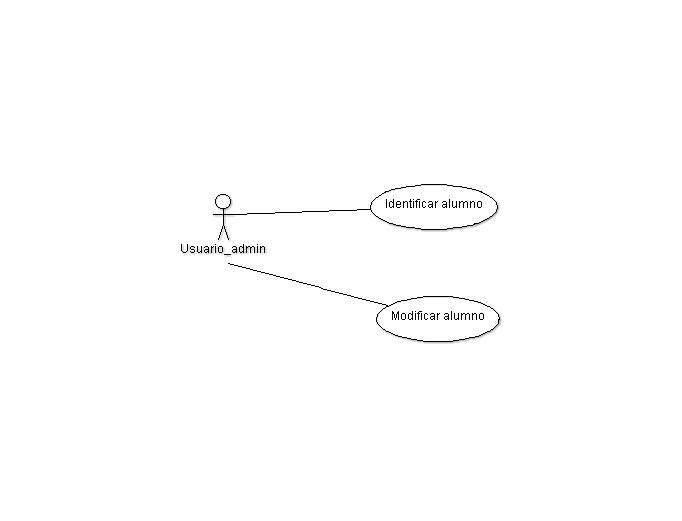
**Diagrama de CU**

Diagrama de actividades

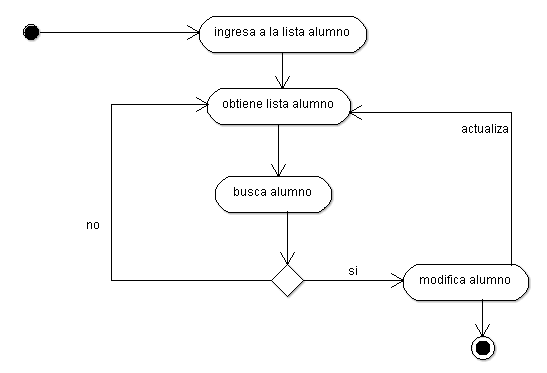


Diagrama de colaboración

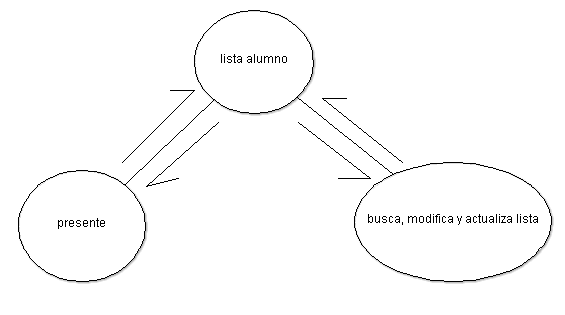


Diagrama de componentes

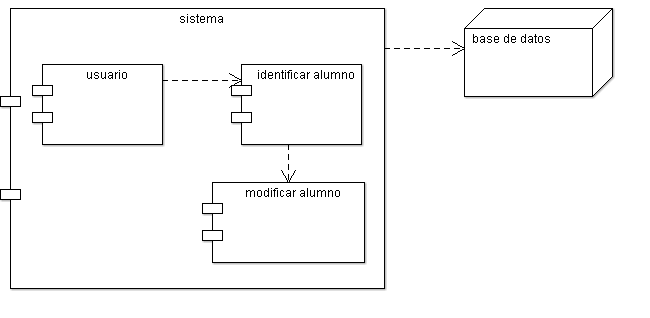


Diagrama de estado

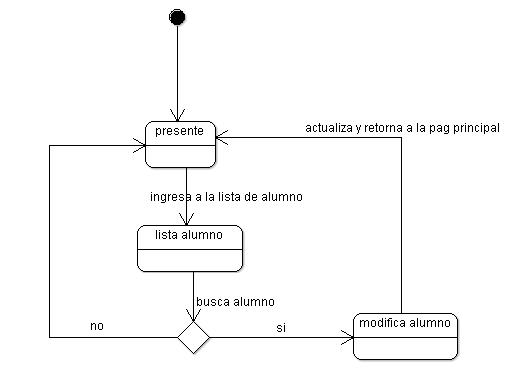


Diagrama de interacción

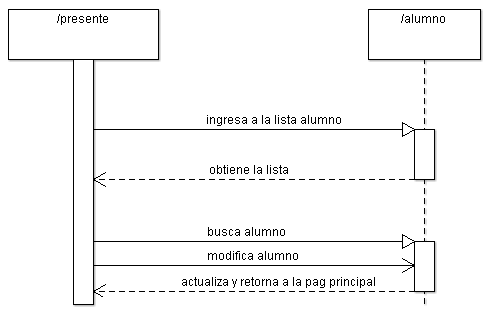


Diagrama de secuencia

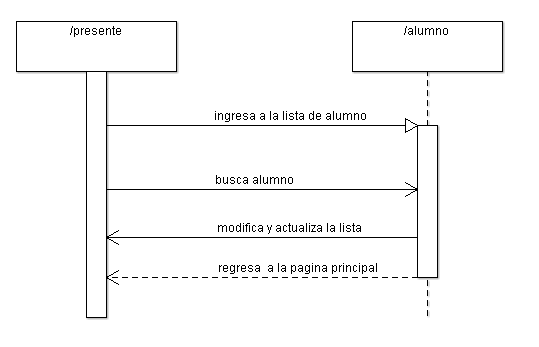
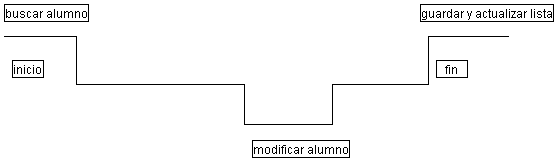
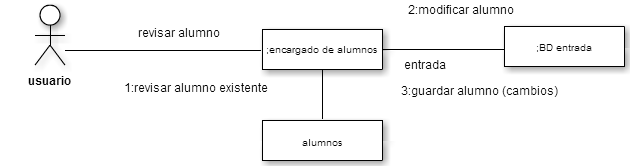


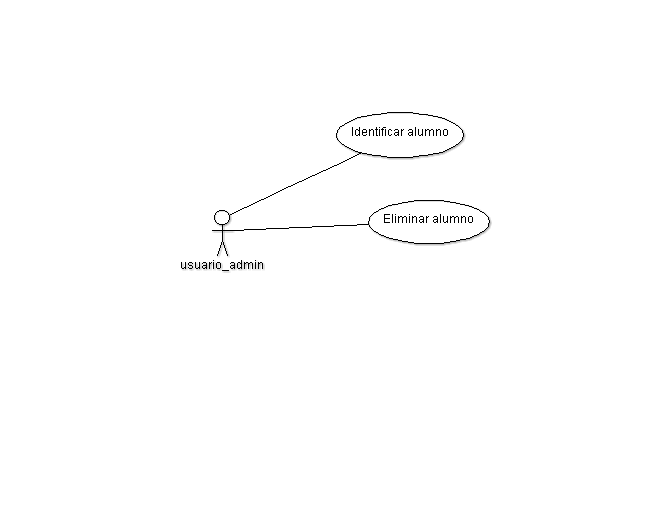
Diagrama de coordinación

****

**Diagrama de comunicación**

****

|  |  |  |
| --- | --- | --- |
| **CU-3** | **Borrar Alumno** | |
| **Fecha** | **07/04/2014** | |
| **Versión** | **1.0** | |
| **Autor** | **Patricia Fernández** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | Mantener ordenada y detallada metódicamente la nómina de alumnos. | |
| **Descripción** | Se presenta este caso de uso cuando se desea eliminar un alumno del listado. | |
| **Precondición** | * El sistema debe tener al alumno registrado * El usuario debe identificar al alumno a eliminar * El usuario debe poseer todos los datos del alumno. | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | El usuario debe buscar al alumno a eliminar |
| 2 | El usuario debe identificar al alumno a eliminar |
| 3 | El usuario elimina al alumno de la lista |
| **Pos condición** | **Se ha eliminado al alumno de la lista** | |
| Excepciones | **Paso** | **Acción** |
| 1 | El usuario no encuentra al alumno, cancela el proceso |
| 3 | El usuario cancela el proceso, el CU queda sin efecto |
| **Rendimiento** | Paso | **Acción** |
| 3 | **3 segundos** |
| **Frecuencia esperada** | Un usuario puede borrar a un mismo alumno 1 veces | |
| **Importancia** | Baja | |
| **Comentarios** |  | |

****

**Diagrama de CU**

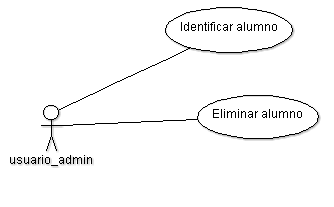
****

Diagrama de actividades



Diagrama de colaboración

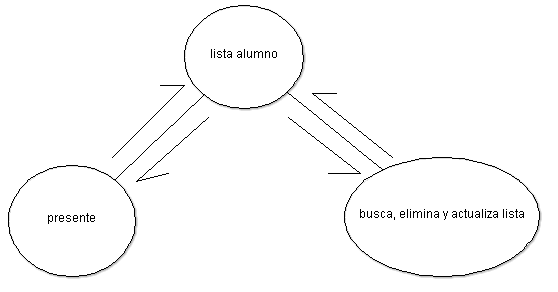


Diagrama de componentes

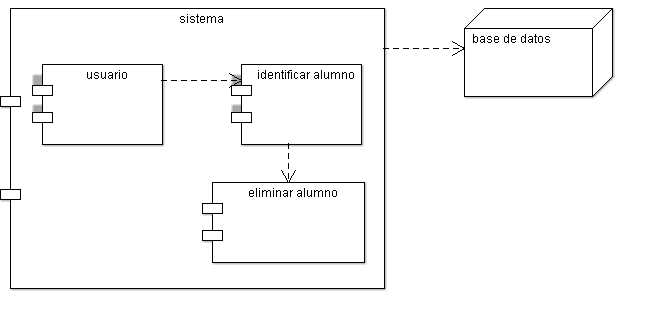


Diagrama de estado

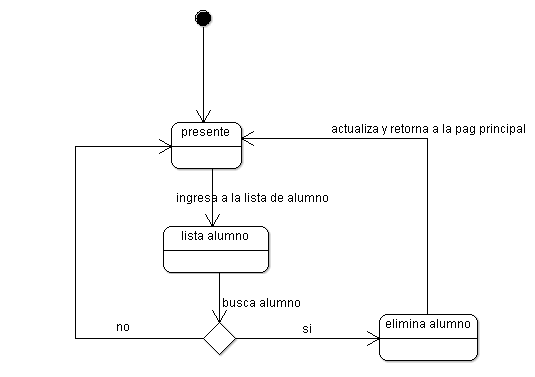


Diagrama de interacción

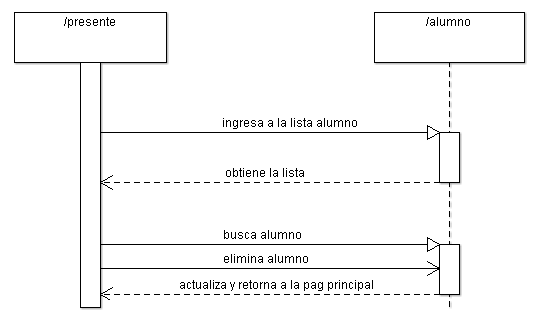


Diagrama de secuencia

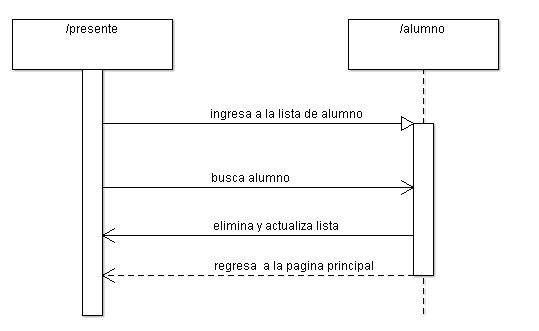
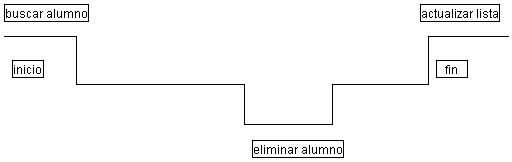
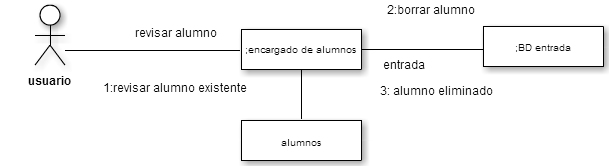


Diagrama de coordinación



**Diagrama de comunicación**



* + 1. ABM Materias

|  |  |
| --- | --- |
| **Código** | CU\_7 |
| **Nombre** | Gestionar Materias |
| **Fecha** | 27-03-15 |
| **Actor** | Cristina Alvarez |
| **Autores** | Usuario (admin) |
| **Descripción** | Gestionar materias |
| **Precondición** | 1-Poseer datos de la materia |
| **Flujo Normal** | Administrar materias |
| **Pos condición** | Accesibilidad desde el portafolio usuarios |
| **Flujo alternative** | Conseguir datos de las recepciones |

|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Agregar Materia** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las materias** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee agregar o registrar una carrera** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario introduce los datos de la materia** |
| **2** | **El sistema registra la materia** |
| **3** | **El sistema muestra los datos de la materia** |
| **Postcondición** | **Se ha registrado una nueva materia** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La materia puede registrarse una vez** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

Diagrama de casos de usos

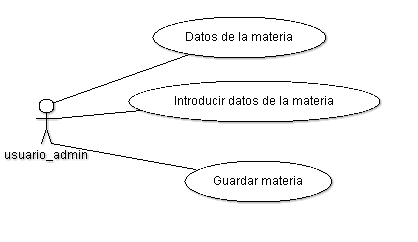


Diagrama de actividades



Diagrama de colaboración

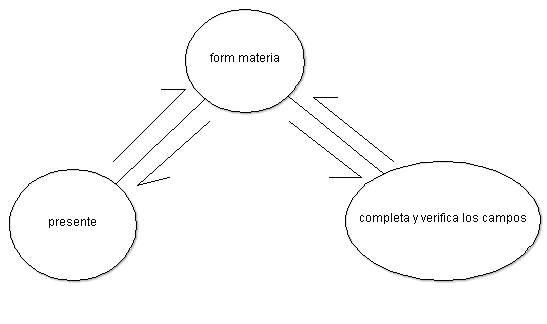


Diagrama de componentes

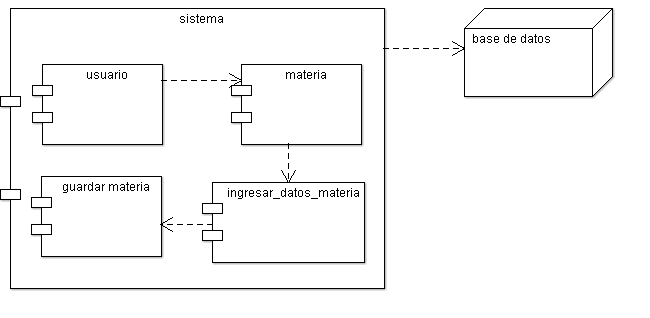


Diagrama de estado

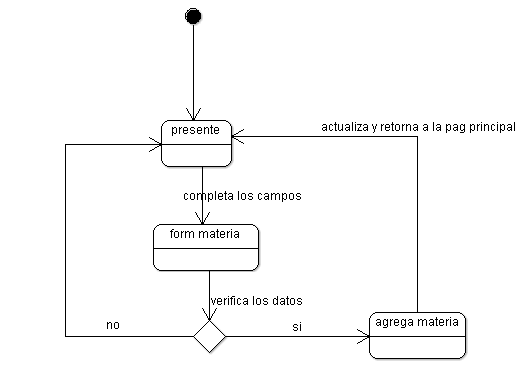


Diagrama de interacción

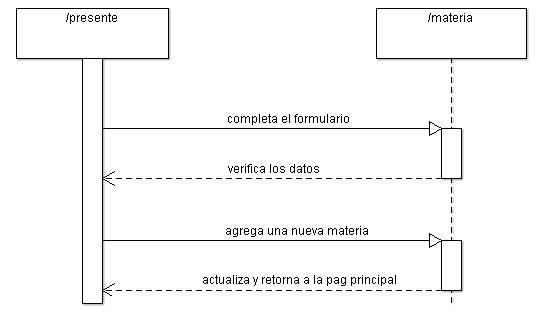


Diagrama de secuencia

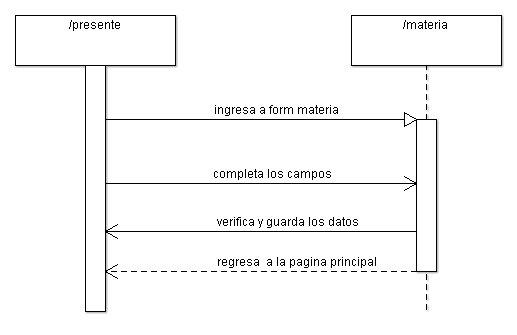
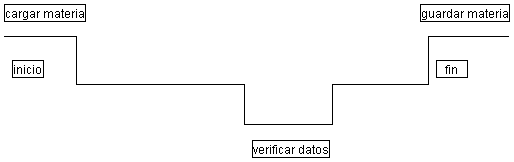
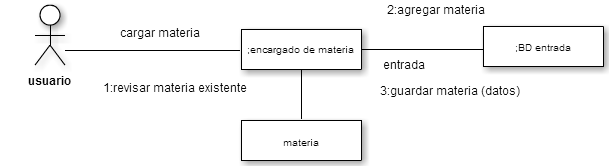


Diagrama de coordinación

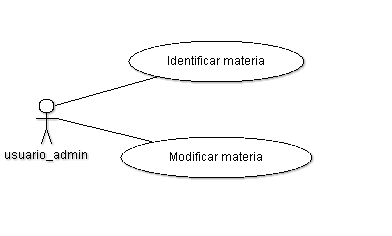


**Diagrama de comunicación**



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Modificar Materia** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las materias** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee modificar datos de una materia** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar el dato a modificar** |
| **2** | **El usuario debe identificar el dato a modificar** |
| **3** | **El usuario modifica los datos** |
| **Postcondición** | **Se ha modificado una materia** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La materia puede ser modificada n veces** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

Diagrama de casos de usos

  
Diagrama de actividades

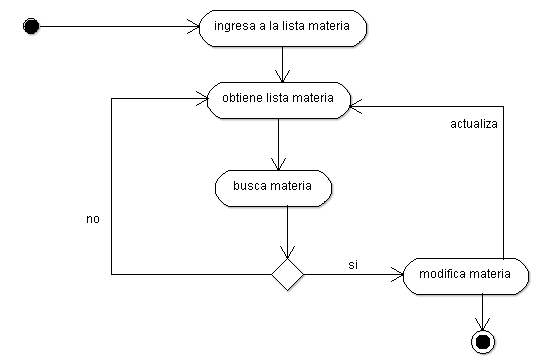


Diagrama de colaboración

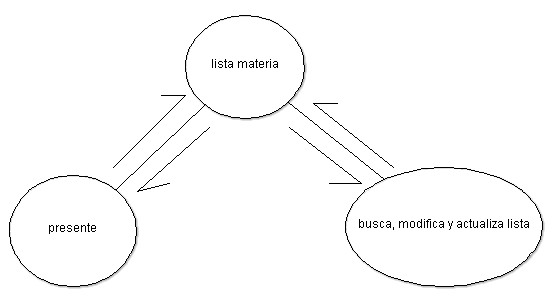


Diagrama de componentes

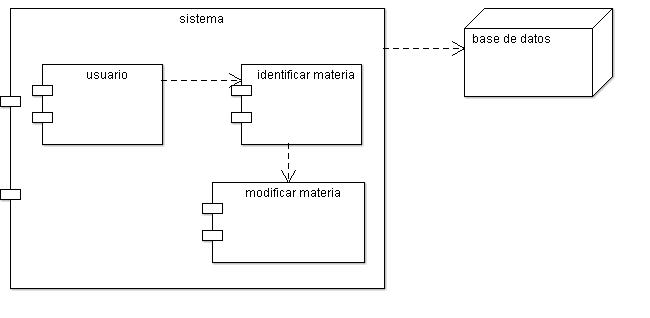


Diagrama de estado

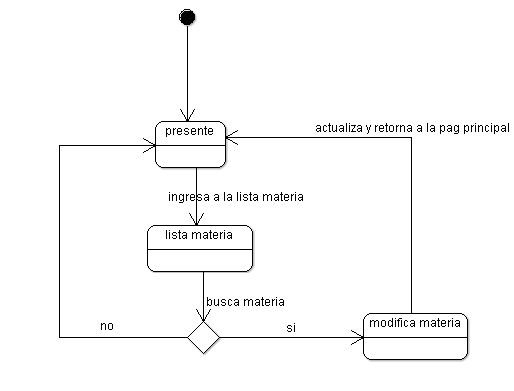


Diagrama de interacción

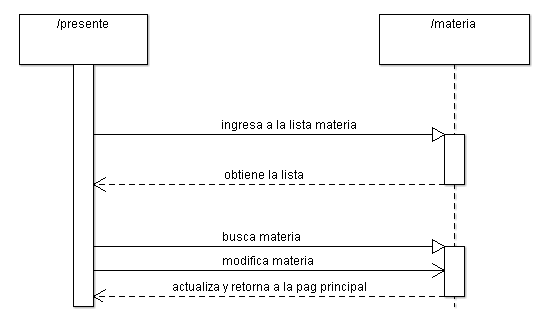


Diagrama de secuencia

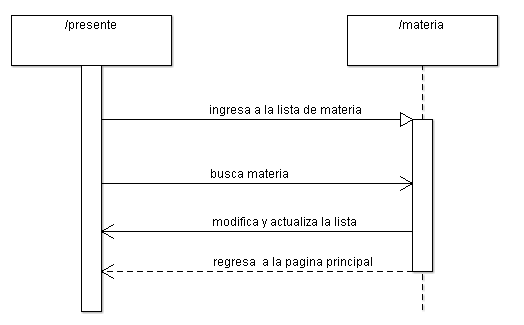


Diagrama de coordinación

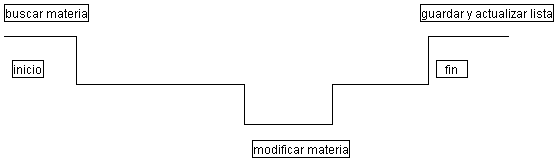
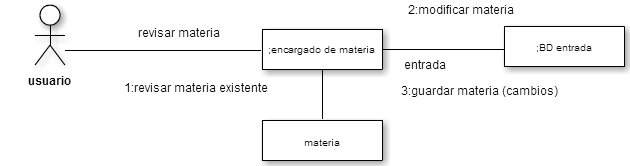


Diagrama de comunicación



|  |  |  |
| --- | --- | --- |
| **Código** |  | |
| **Nombre** | **Eliminar materia** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario-Administrador** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las materias** | |
| **Descripción** | **Se presenta este caso de uso cuando se requiera eliminar una materia** | |
| **Precondición** | **- El usuario posee los permisos para administrar el sistema**  **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario debe buscar la materia a eliminar** |
| **2** | **El usuario debe identificar la materia a eliminar** |
| **3** | **El usuario elimina la materia seleccionada** |
| **Postcondición** | **Se ha eliminado una carrera** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La materia puede ser eliminada una vez** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

Diagrama de casos de uso

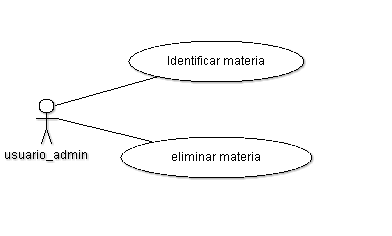


Diagrama de actividades

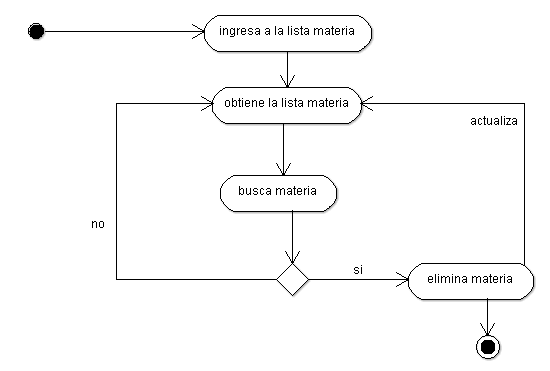


Diagrama de colaboración

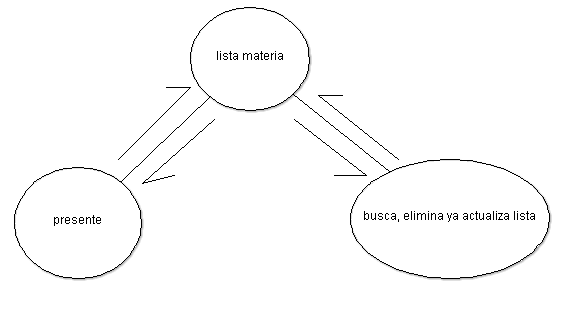


Diagrama de componentes

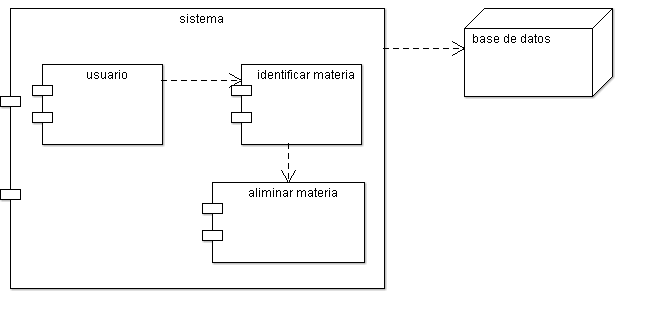


Diagrama de estado

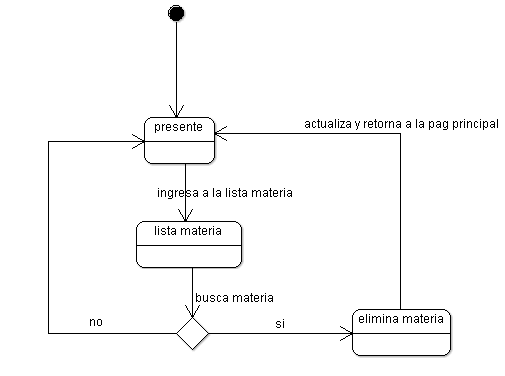


Diagrama de interacción

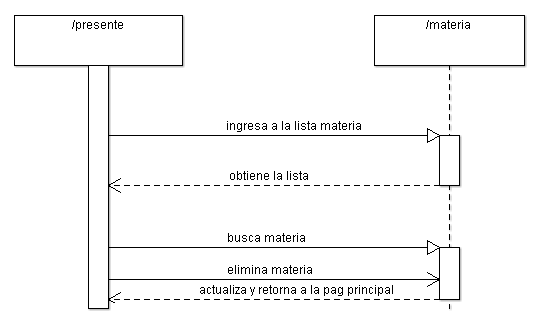


Diagrama de secuencia

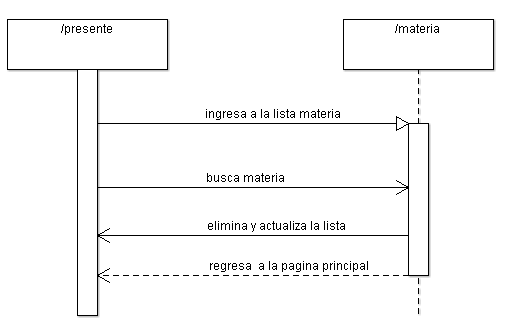


Diagrama de coordinación

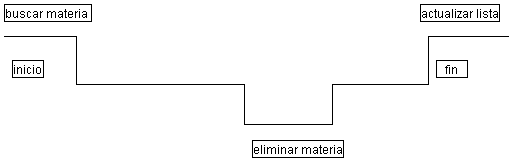
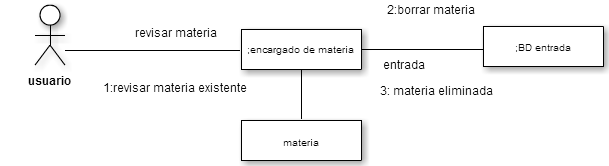


Diagrama de comunicación



* + 1. **Crear planilla de asistencia**

|  |  |  |
| --- | --- | --- |
| **Código** | CU\_8 | |
| **Nombre** | **Crear planilla** | |
| **Fecha** | **17/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente las asistencias** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee agregar o registrar una carrera** | |
| **Precondición** | **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario selecciona la materia correspondiente (las materia aparecerán en las opciones de acuerdo a cada profesor)** |
| **2** | **El sistema genera planilla** |
| **3** | **El sistema muestra la planilla** |
| **Postcondición** | **Se ha creado nueva planilla** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La planilla se genera una vez por materia** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

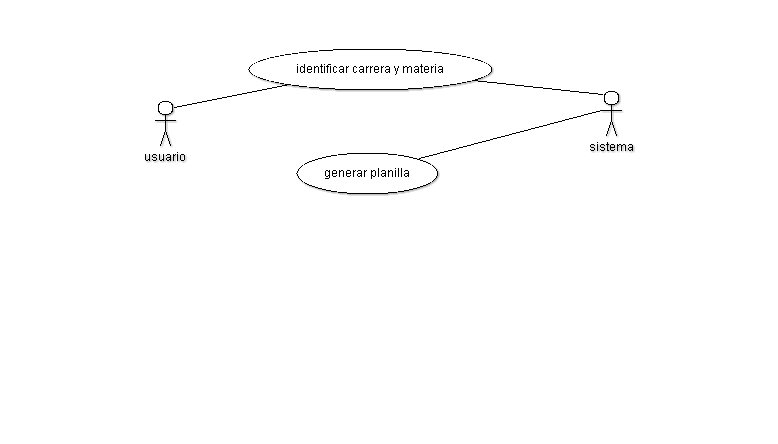
Diagrama de casos de uso

Diagrama de actividades



Diagrama de colaboración

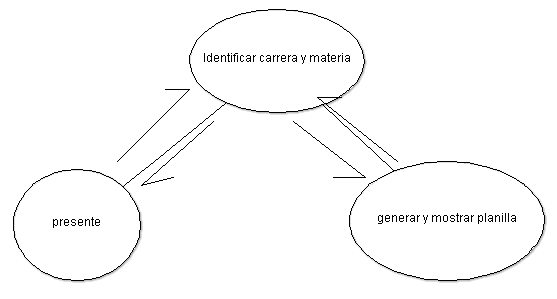


Diagrama de componentes

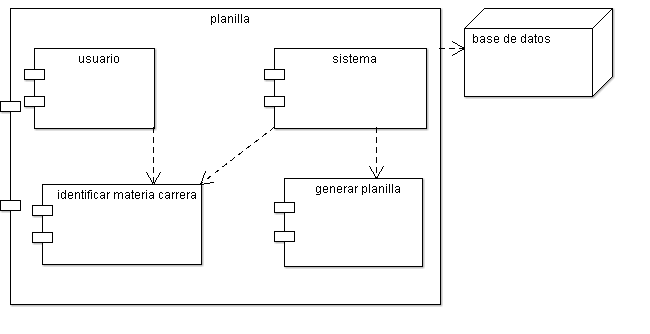


Diagrama de estado

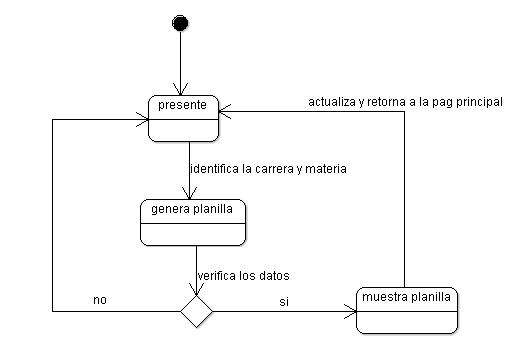


Diagrama de interacción

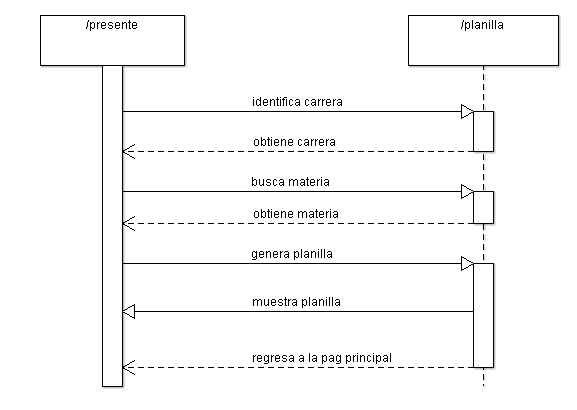


Diagrama de secuencia

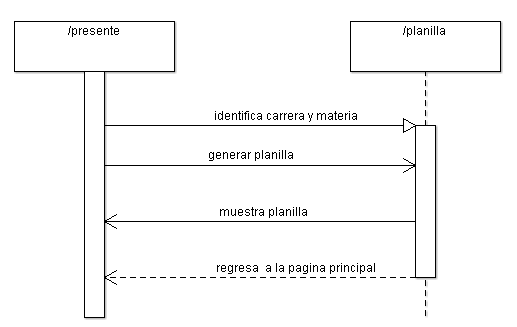


Diagrama de coordinación

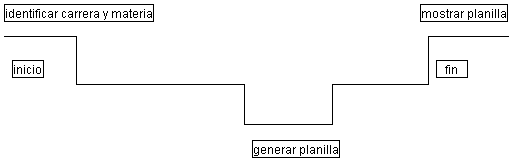
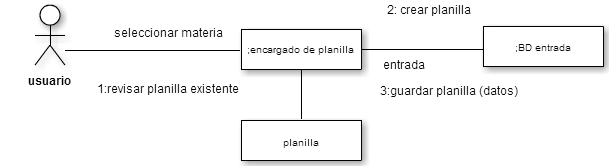


Diagrama de comunicación



1. **Registrar asistencias**

|  |  |  |
| --- | --- | --- |
| **Código** | CU\_9 | |
| **Nombre** | Registro de asistencias | |
| **Fecha** | **17/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente las asistencias** | |
| **Descripción** | **Se presenta este caso de uso cada vez que se tenga que registrar las asistencias** | |
| **Precondición** | **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario identifica la planilla correspondiente** |
| **2** | **El usuario marca las asistencias y elige guardar** |
| **3** | **El sistema guarda la información** |
| **Postcondición** | **Se ha registrado las asistencias** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **La materia puede registrar las asistencia cada semana** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

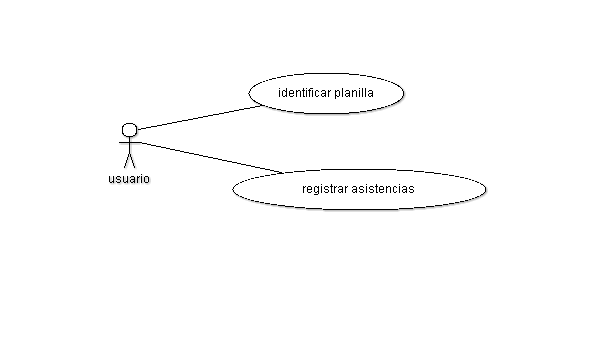


Diagrama de actividades

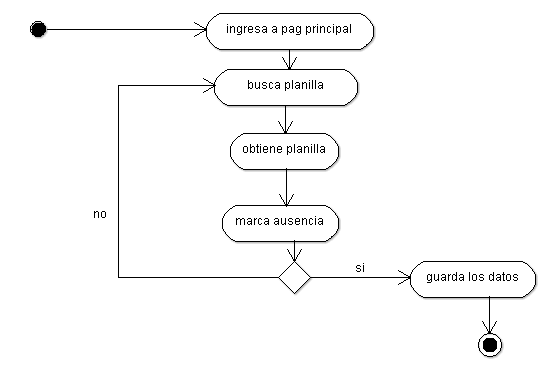


Diagrama de colaboración

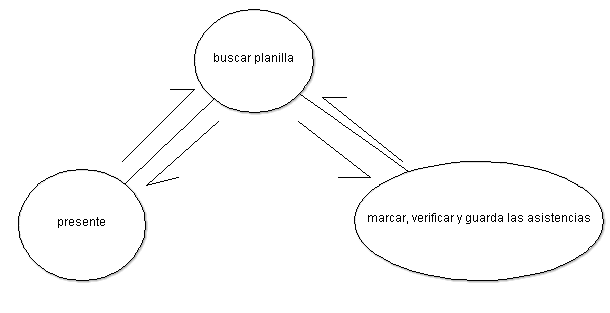


Diagrama de componentes

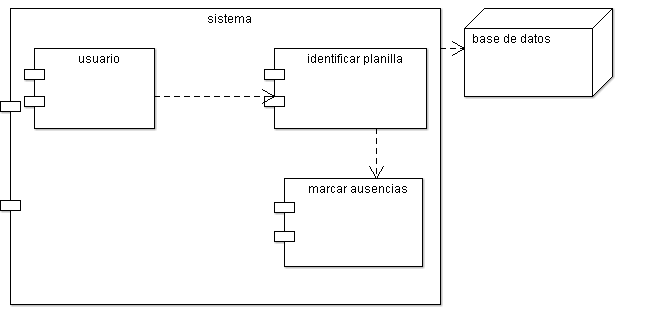


Diagrama de estado

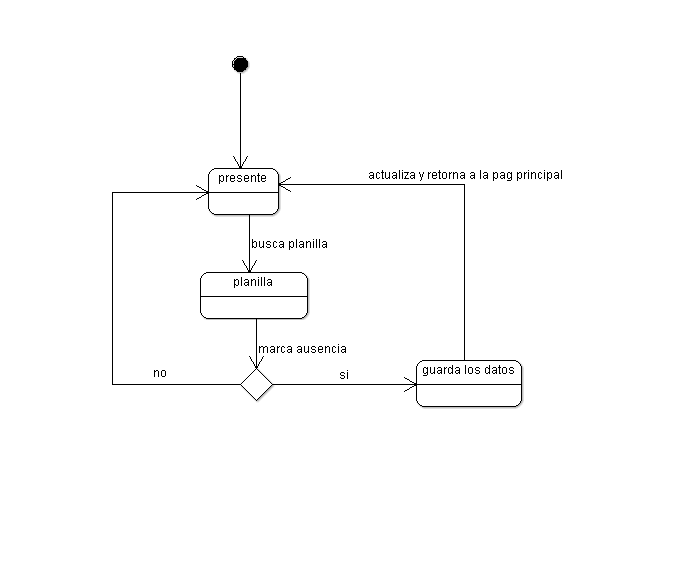


Diagrama de interacción

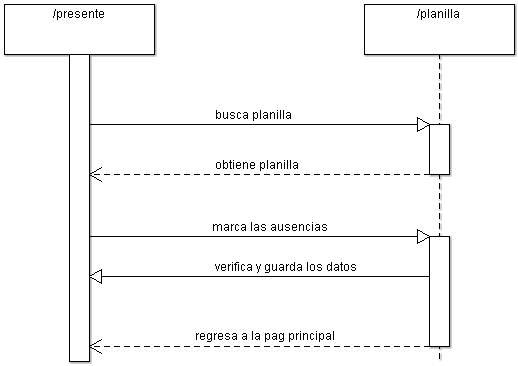


Diagrama de secuencia

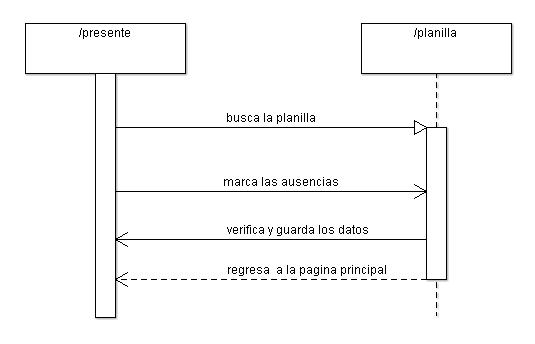


Diagrama de coordinación

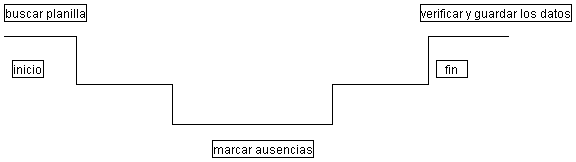
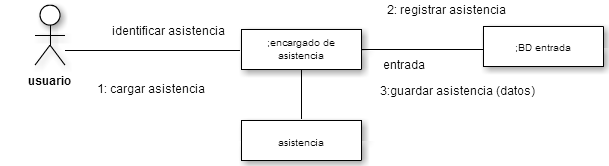


Diagrama de comunicación



* + 1. **Reporte de asistencia**

|  |  |  |
| --- | --- | --- |
| **Código** | CU\_10 | |
| **Nombre** | **Mostrar reporte** | |
| **Fecha** | **09/04/2015** | |
| **Versión** | **2.0** | |
| **Autor** | **Cristina Alvarez** | |
| **Actores** | **Usuario** | |
| **Objetivos Asociados** | **Mantener ordenado y detallada metódicamente los registros de las materias** | |
| **Descripción** | **Se presenta este caso de uso cuando se desee agregar o registrar una carrera** | |
| **Precondición** | **- El usuario ingresa su usuario y contraseña** | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| **1** | **El usuario identifica la planilla correspondiente** |
| **2** | **El sistema genera un informe mostrando las asistencias.** |
| **Postcondición** | **Se muestra asistencias** | |
| **Excepciones** | **Paso** | **Acción** |
| **4** | **El usuario cancela el proceso, el CU queda sin efecto** |
| **5** | **Resaltar campos obligatorios** |
| **Frecuencia esperada** | **Las asistencias pueden mostrarse n veces** | |
| **Importancia** | **Alta** | |
| **Comentarios** |  | |

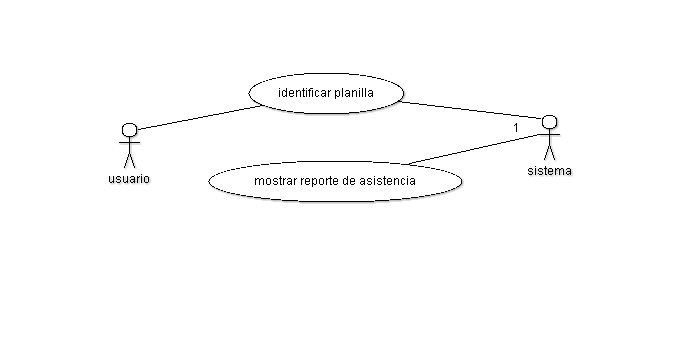


Diagrama de actividades

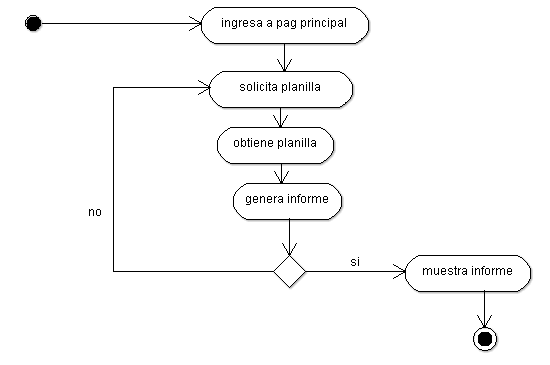


Diagrama de colaboración

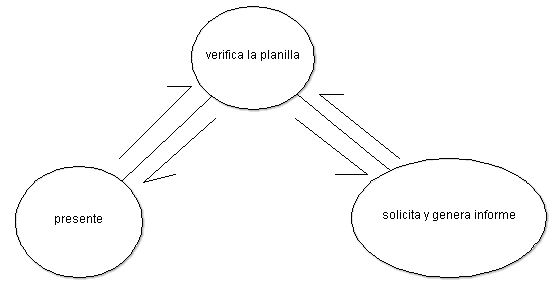


Diagrama de componentes

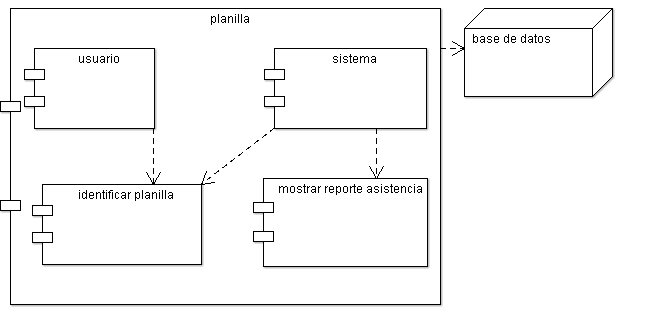


Diagrama de estado

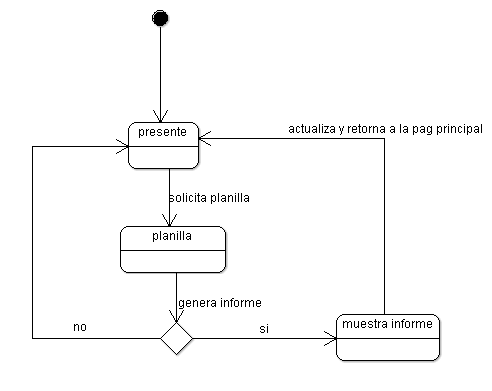


Diagrama de interacción

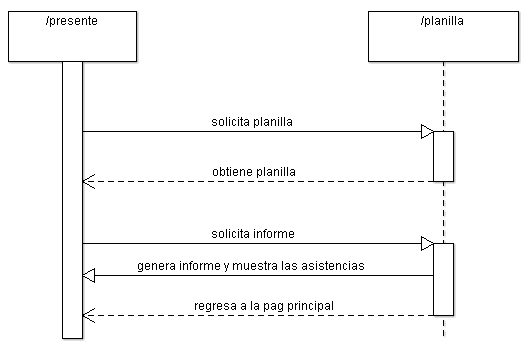


Diagrama de secuencia

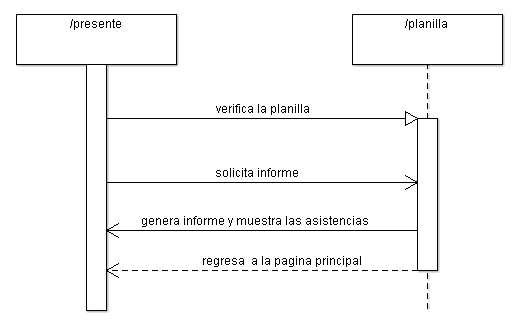


Diagrama de coordinación

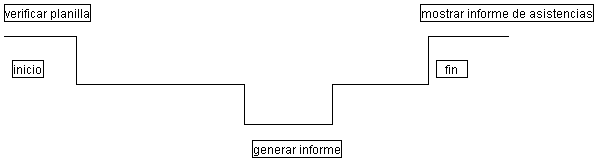


Diagrama de comunicación



# ANÁLISIS

# DE

# RIESGO

PRESENTE

Análisis de Riesgo

1. **Inventario completo**

Inventariar Los activos a proteger

## Plan de acción

**Análisis y Gestión de Riesgos**

**1. Introducción:**

Un riesgo es una variable del proyecto que pone en peligro o impide el éxito del mismo. Es la “probabilidad de que un proyecto experimente sucesos no deseables, como retrasos en las fechas, excesos de costes, o la cancelación directa”.

Se han producido amplios debates sobre la definición adecuada para riesgo de software, y hay acuerdo común en que el riesgo siempre implica dos características:

• Incertidumbre: el acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100 por ciento de probabilidad.   
  
• Pérdida: Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

Es por esta razón que, para que un proyecto de desarrollo pueda llevarse a cabo dentro de los tiempos establecidos y los costos previstos, esos riesgos deben ser identificados y controlados, es decir se debe realizar un adecuado “Análisis y Gestión de Riesgos”.

Mediante el análisis de los riesgos podremos identificar las amenazas que acechan a los distintos componentes pertenecientes o relacionados con el Sistema de información (conocidos como ‘activos’); para determinar la vulnerabilidad del sistema ante esas amenazas y para estimar el impacto o grado de perjuicio que una seguridad insuficiente puede tener para la organización, obteniendo cierto conocimiento del riesgo que se corre.

La gestión de los riesgos, basada en los resultados obtenidos en el análisis anterior, permite seleccionar e implantar las medidas o ‘salvaguardas’ de seguridad adecuadas para conocer, prevenir, impedir, reducir o controlar los riesgos identificados y así reducir al mínimo su potencialidad o sus posibles perjuicios.

**2. Propósito del plan**

El propósito del presente plan es identificar los riesgos que se puedan presentar en el desarrollo del proyecto, analizarlos, calcular la exposición al riesgo (combinando información de su probabilidad de ocurrencia con la magnitud de su impacto), en base a ello poder priorizarlos, para establecer estrategias de control y resolución y ejercer una correcta supervisión de los mismos.

**3. Visión general**

**3.1. Objetivos**

El objetivo del presente plan es asegurar que el proyecto se concrete dentro de los plazos previstos y observando todos los requerimientos fijados por los futuros usuarios. Permitiendo, al mismo tiempo, la detección oportuna de los problemas técnicos que se puedan presentar y la ejecución de un adecuado control y gestión de los cambios que se vayan presentando durante el desarrollo.

**3.2. Prioridades de aversión al riesgo**

Se definen para el presente plan las siguientes prioridades de aversión al riesgo:

1. Volatilidad de requisitos.
2. Cumplimiento de requisitos.
3. Cumplimiento de la planificación temporal.
4. Problemas de personal.
5. Problemas tecnológicos.
6. Problemas económicos.

**3.3. Organización**

**3.3.1. Gestión**

La gestión del presente plan deberá ser llevada a cabo en forma directa por el Gestor de Riesgos del Proyecto con el apoyo de todos los personales con sus respectivos roles.

**3.3.2. Responsabilidades**

* Del Gestor de Riesgos del proyecto:
  + Gestionar el presente plan.
  + Adoptar las medidas necesarias tendientes a evitar retrasos en la planificación realizada.
  + Tener un plan de contingencia para los diferentes tipos de problemas que se puedan presentar.
* Del Lider de Proyecto:
  + Evaluar con el equipo de gestión de configuración los cambios solicitados en el caso que se presenten.
  + Ordenar al equipo de desarrollo la implementación de los cambios aprobados.
  + Supervisar el cumplimiento de la planificación de desarrollo del proyecto.
* Del Gestor de configuracion:
  + Interactuar con los futuros usuarios del Sistema para discutir posibles solicitudes de cambio y variación de los requisitos ya establecidos.
  + Comprobar que el producto satisfaga los requerimientos establecidos.
  + Evaluar con el equipo de gestión de configuración los cambios solicitados en el caso que se presenten.
* De los documentadores del proyecto:
  + Interactuar con el equipo de trabajo para detectar tempranamente problemas técnicos o de personal.
  + Gestionar los informes de incidencia.

## *4. Análisis y Gestión de Riesgos*

***4.1. Identificación***

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto. Los riesgos deben ser identificados y expresados en forma clara e inequívoca para que el equipo de desarrollo pueda llegar a un consenso y continuar hacia la fase de análisis y prioridad.

La meta en la identificación de riesgos es la elaboración de una lista de los riesgos con los que el equipo deberá enfrentarse. Esta lista debe ser lo más *extensa* posible y deberá cubrir todas las áreas del proyecto, para ello se hará uso la clasificación de riesgos.

***4.1.1. Clasificación (o Taxonomía) de los Riesgos – Fuentes***

La clasificación de los riesgos -también denominadas taxonomías de riesgos- puede servir de ayuda para elaborar un enfoque coherente, reproducible y medible. Las listas de clasificación permiten al equipo pensar con mayor amplitud sobre los riesgos que pueden afectar al proyecto porque se dispone de una lista de áreas del proyecto susceptibles de esconder riesgos.

Existen muchas taxonomías o clasificaciones para los riesgos de proyectos generales de desarrollo de software. Para el presente trabajo se ha escogido la clasificación propuesta por el Software Risk Management (SRM) desarrollado por el Software Engineering Institute.

La siguiente tabla muestra una clasificación de alto nivel de las fuentes de riesgo de los proyectos siguiendo la taxonomía propuesta por la metodología SRM organizadas en tres niveles: clases, elementos y atributos.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Elemento** | **Riesgo** | **Fuente** |
| RI-01 | Planificación | Errores en la estimación de presupuesto | Lider de Proyecto |
| RI-02 | Planificación | Errores en la estimación de la calendarización | Lider del proyecto |
| RI-03 | Planificación | Adelantar fecha de entrega | Usuario/  Lider del proyecto |
| RI-04 | Requerimientos | Formulación de nuevos requisitos | Usuarios/  Documentadores |
| RI-05 | Requerimientos | Eliminación de requisitos aprobados | Usuarios/  Documentadores |
| RI-06 | Requerimientos | Errores en la definición de requisitos. | Usuarios/  Documentadores |
| RI-07 | Requerimientos | Necesidades de los usuarios no recogidas de forma completa | Usuarios/  Documentadores |
| RI-08 | Diseño | Existencia de funciones que presentan dificultades técnicas | Lider de Proyecto  Desarrolladores |
| RI-09 | Diseño | Diseño inadecuado para la realización del proceso de pruebas | Tester |
| RI-10 | Diseño | Problemas con la definición de la interfase de usuario | Lider de Proyecto  Desarrolladores |
| RI-11 | Diseño | Aparición de errores por la reutilización de software preexistente | Lider de Proyecto  Desarrolladores |
| RI-12 | Gestión de  Configuración | Solicitud de cambios no controlados o incorrectamente evaluados. | Usuarios  Gestor de configuración |
| RI-13 | Usuarios | Poca Comunicación con futuros usuarios | Lider de Proyecto  Desarrolladores |
| RI-14 | Usuarios | Ausencia de usuarios finales para validar el producto | Lider de Proyecto  Desarrolladores |
| RI-15 | Proceso | Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software | Lider de Proyecto/  Gestor de configuración |
| RI-16 | Proceso | No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados | Gestor de configuración |
| RI-17 | Proceso | No definir reglas específicas para la documentación del código fuente. | Gestor de configuración |
| RI-18 | Tecnología | Dificultades con herramientas y lenguajes nuevos de programación | Lider de Proyecto  Desarrolladores |
| RI-19 | Tecnología | Problemas con los equipos de hardware para el desarrollo | Lider de Proyecto  Desarrolladores |
| RI-20 | Equipo de Trabajo | Despido o renuncia de personal para desarrollar el producto en los tiempos estimados. | Lider de Proyecto/  Desarrolladores |
| RI-21 | Equipo de Trabajo | Dificultad de comunicación por la distribución del personal. | Lider de Proyecto  Desarrolladores |
| RI-22 | Equipo de Trabajo | Inexperiencia del equipo con la metodología de desarrollo | Lider de Proyecto  Desarrolladores |
| RI-23 | Equipo de Trabajo | Perdida de documentos y archivos del proyecto | Lider el proyecto/  Gestor de configuración |
| RI-24 | Equipo de trabajo | Reposo a causa de problemas de salud | Lider de Proyecto/  Desarrolladores |
| RI-25 | Equipo de trabajo | Permiso de ausencias laborales a causa de lutos | Lider de Proyecto/  Desarrolladores |
| RI-26 | Equipo de trabajo | Problemas financieros | Lider de Proyecto/  Desarrolladores |

## *44.1.2 Declaración de los Riesgos*

Las definiciones genéricas de un riesgo no hacen desaparecer la incertidumbre y dan lugar a distintas interpretaciones del riesgo. Las definiciones que no dejan lugar a dudas permiten a los equipos:

* Asegurarse de que todos los miembros del equipo comprenden el riesgo de la misma forma.
* Comprender la causa o causas del riesgo y la relación con los problemas que puedan surgir.
* Disponer de una base para realizar un análisis formal y cuantitativo y planear los esfuerzos.

A continuación se definen en forma mas precisa los riesgos identificados anteriormente siguiendo un proceso de declaración en dos partes (condición – consecuencia). La primera parte de la declaración de riesgo se denomina **condición** y describe una situación o atributo del proyecto existente que el equipo prevé que puede resultar en una pérdida en el proyecto o en una reducción de beneficios. La segunda

***44.1.2 Declaración de los Riesgos***

Las definiciones genéricas de un riesgo no hacen desaparecer la incertidumbre y dan lugar a distintas interpretaciones del riesgo. Las definiciones que no dejan lugar a dudas permiten a los equipos:

* Asegurarse de que todos los miembros del equipo comprenden el riesgo de la misma forma.
* Comprender la causa o causas del riesgo y la relación con los problemas que puedan surgir.
* Disponer de una base para realizar un análisis formal y cuantitativo y planear los esfuerzos.

A continuación se definen en forma más precisa los riesgos identificados anteriormente siguiendo un proceso de declaración en dos partes (condición – consecuencia). La primera parte de la declaración de riesgo se denomina **condición** y describe una situación o atributo del proyecto existente que el equipo prevé que puede resultar en una pérdida en el proyecto o en una reducción de beneficios. La segunda parte de la declaración de riesgo se denomina **consecuencia** y describe el atributo o situación no deseable del proyecto. Además se incluyen los **efectos** que tendrían estos riesgos de no controlarse debidamente.

**RI-01 Errores en la estimación del presupuesto.**

Condición: Errores en los cálculos, no estimar bien los factores que influyen en el cálculo (archivos, funciones, etc.) O el caso contrario la sobre estimación.

Consecuencia: No disponer de los recursos necesarios para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el presupuesto para desarrollar todas las funciones requeridas, baja calidad del producto.

**RI-02 Errores en la estimación de la calendarización.**

Condición: Errores en los cálculos de estimaciones de fechas para el desarrollo del proyecto, no estimar bien los factores que influyen en el desarrollo (archivos, funciones, etc.).

Consecuencia: No disponer del tiempo necesario para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el tiempo necesario para desarrollar todas las funciones requeridas, baja calidad del producto.

**RI-03 Adelantar fecha de entrega.**

Condición: Se requiere que el sistema sea entregado antes de la fecha de entrega que fue fijada al iniciar su desarrollo.

Consecuencia: No disponer del tiempo necesario para terminar el proyecto a tiempo, sobrecarga de tareas al personal.

Efecto: No contar con el tiempo necesario para desarrollar todas las funciones requeridas, baja calidad del producto.

**RI-04 Formulación de nuevos requisitos.**

Condición:Este riesgo tiene que ver con la volatilidad de los requerimientos de los usuarios. Al estar el producto enmarcado en estándares internacionales constantemente actualizados, este riesgo se reduce notablemente ya que con ello se tienen en cuenta los requisitos primordiales y de gran costo para el proyecto.

Consecuencia: Mayor tiempo de desarrollo, el proyecto puede sufrir retrasos y será más crítico cuando el desarrollo esté más avanzado.

Efecto: No terminar el producto en los plazos establecidos, carecer de las funcionalidades requeridas por los usuarios, el producto podría volverse inconsistente ya que no se han considerado de un comienzo estos requisitos y puede traer problemas para integrarlos.

**RI-05 Eliminación de requisitos aprobados.**

Condición: Existen muchas causas por las cuales puede ser factible eliminar requisitos que en algún momento se los consideraba necesarios. Entre ellas se puede citar: relevamiento pobre de las necesidades de los futuros usuarios (lo que puede llevar a creer que son necesarias ciertas funciones que en realidad no lo son para los usuarios o la política de la empresa), ausencia de un estudio de antecedentes, o porque ya no es un requisito de los usuarios.

Consecuencia: Si se ha avanzado considerablemente sobre estos requisitos, puede significar un desperdicio de los recursos disponibles para el proyecto (humanos, financieros, etc.)

Efecto: El producto podría no contar con las funciones verdaderamente requeridas por los usuarios en un tiempo considerable, ya que se han empleado recursos en el desarrollo de funciones que en realidad no beneficia a los usuarios y, posiblemente postergar tareas que eran vitales.

**RI-06 Errores en la definición de requisitos.**

Condición: Falta de experiencia del personal del proyecto, poca documentación para el análisis de los requisitos, etc.

Consecuencia: Desperdicio de recursos, más aún en fases avanzadas del proyecto. Clientes insatisfechos.

Efecto: Baja calidad del producto, ya que puede no ser detectado a tiempo, gastos innecesarios de recursos ya sea por el desarrollo de los mismos o por las correcciones que se deberían realizar al producto para enmendar errores.

**RI-07 Necesidades de los clientes no recogidas de forma completa.**

Condición: El equipo de desarrollo no se ha contactado con suficientes usuarios del sistema, de modo que no tiene una visión global de cuales son los principales requisitos que demandan los usuarios finales.

Consecuencia: Pérdida de tiempo en el desarrollo de necesidades que podrían no ser las más importantes para los usuarios.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales, con lo que se desestima su uso.

**RI-08 Existencia de funciones que presentan dificultades técnicas**

Condición: Errores en la determinación de ciertos factores en alguna de las etapas de desarrollo (análisis, diseño) son frecuentemente la causa de dificultades técnicas, poca experiencia en el lenguaje de programación utilizado, falta de experiencia en sistemas expertos y su tecnología de soporte, no considerar las limitaciones de la plataforma utilizada.

Consecuencia: Retraso del proyecto debido al tiempo necesario para superar estas dificultades, lo que también puede llevar a la inconsistencia de todo el sistema, con la posibilidad de tener que contratar personal adicional (altamente calificado en el tema) para solucionar el problema.

Efecto: En caso de no ser controlado el producto puede presentar serios defectos con lo cual puede producir el rechazo de los usuarios.

**RI-09 Diseño inadecuado para la realización del proceso de pruebas**

Condición: Escaso conocimiento de las funciones del producto por parte de los diseñadores del proceso de prueba. Falta de experiencia de los mismos.

Consecuencia: Esfuerzo adicional para la depuración del producto. No eliminación de la mayoría de los errores más importantes.

Efecto: Lanzar un producto con muchos defectos, retraso en la terminación de una versión estable del mismo.

**RI-10 Problemas con la definición de la interfase de usuario.**

Condición: Escasa experiencia del equipo de desarrolladores con aplicaciones afines para definir las interfaces adecuadas para los usuarios.

Consecuencia: Errores de diseño y retraso en la implementación del sistema.

Efecto: Desarrollar un producto con poca funcionalidad, dificultad en el manejo del sistema o no poder aprovechar todos los beneficios del mismo.

**RI-11 Aparición de errores por la reutilización de software preexistente**

Condición: Incompatibilidad del software, el software preexistente no es soportado completamente por la arquitectura diseñada, manejo de distintos formatos de datos por los diversos componentes (preexistentes y desarrollados por el equipo).

Consecuencia: Perdida de tiempo en la solución de los errores, en la adaptación del software. Probabilidad de desarrollar ciertos módulos desde cero.

Efecto: Mal funcionamiento y bajo rendimiento del software en su conjunto, falta de funciones requeridas por los usuarios.

**RI-12 Solicitud de cambios no controlados o incorrectamente evaluados.**

Condición: Los cambios a solicitud del usuario o bien del equipo de desarrollo surgirán de manera inevitable.

Consecuencia: Complicaciones a la hora de gestionar los cambios, con posibilidades de incompatibilidades o inconsistencias.

Efecto: Si no se aplica un debido proceso de Gestión de Configuración del Software es posible desarrollar un producto con muchos defectos.

**RI-13 Poca comunicación con los futuros usuarios.**

Condición: El equipo de desarrollo no ha contactado con suficientes usuarios finales para definir sus necesidades.

Consecuencia: No tener una visión global de cuales son los principales requisitos que demandan los usuarios finales, resistencia de los usuarios a implementar ciertas funciones.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales, con lo que se puede desestimar su uso por gran parte de los futuros usuarios, ya que no sienten que el producto se adapta medianamente a su entorno.

**RI-14 Ausencia de usuarios finales para validar el producto.**

Condición: El equipo de desarrollo no cuenta con usuarios finales para validar los requisitos definidos, no contar con un grupo usuarios de un entorno real para la validación de los requisitos.

Consecuencia: Desarrollar un producto que posea funciones que no son necesarias para la mayoría de los usuarios y en contraste carencia de otras.

Efecto: Desarrollo de un producto incompleto, que no satisface de forma plena las necesidades específicas de los usuarios finales. O conformado por funciones que no le sirve al usuario.

**RI-15 Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software.**

Consecuencia: Dificultad para armar la carpeta del proyecto, puede llegar a ser muy difícil la utilización de dichos documentos o imposible. Ya que el equipo de desarrollo trabaja en forma distante, y a la vez existen sub-equipos.

**RI-16 No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados**

Consecuencia: Es muy importante documentar el resultado de las revisiones (y datos relacionados) y ponerlos a disposición de todo el equipo de trabajo. Ya que puede servir, en el caso de los errores, por ejemplo: para no cometerlos nuevamente, y para utilizar esta información en proyectos similares.

**RI-17 No definir reglas específicas para la documentación del código fuente.**

Condición: Este riesgo se origina al no definir un estándar de programación.

Consecuencia: Dificultades para leer, comprender, detectar errores, realizar modificaciones del código fuente.

Efecto: Lanzar un producto con muchos defectos y baja calidad.

**RI-18 Dificultades con herramientas y lenguajes nuevos de programación.**

Condición: Escaso conocimiento y experiencia de algunos de los integrantes del proyecto sobre las herramientas utilizadas y los lenguajes de programación.

Consecuencia: Mayor tiempo de desarrollo para investigar sobre el tema, o en la capacitación del personal.

Efecto: Lanzar un producto con baja calidad, con muchos defectos.

**RI-19 Problemas con los equipos de hardware para el desarrollo.**

Condición: Equipos con los que cuentan los personales empiezan a fallar, a volverse lento, se descomponen o ya no funcionan.

Consecuencia: Invertir en reparaciones o nuevos equipos de hardware para el desarrollo.

**RI-20 Despido o renuncia de personal para desarrollar el producto en los tiempos estimados.**

Condición: Este riesgo se podría presentar como una consecuencia de RI-01, pero no necesariamente, también por la falta de presupuesto o por diversos motivos.

Consecuencia: Una de las consecuencias más evidentes es la demora del proyecto. También se puede dar la ausencia de personal calificado para tareas específicas con lo que las mismas no se podrán realizar hasta contar con el personal requerido.

Efecto: Que el software desarrollado tenga carencias, ya sea técnicas o funcionales. O mala calidad del mismo en caso de no utilizar el personal calificado para llevar a cabo determinadas tareas.

**RI-21 Dificultad de comunicación del equipo.**

Condición: El equipo de desarrollo se encuentra distribuido en las ciudades de Encarnación y Cambyreta.

Consecuencia: Dificultades de comunicación, la comunicación podría no ser lo suficientemente fluida.

Efecto: Errores de comunicación, posibilidad de avanzar en el desarrollo de una actividad sin la validación requerida.

**RI-22 Inexperiencia del equipo con la Metodología de desarrollo.**

Condición: Algunos participantes del equipo de desarrollo pueden no contar con la experiencia suficiente en cuanto al uso de la Metodología.

Consecuencia: Retraso en el desarrollo de actividades definidas por la metodología, no aprovechar por completo las prestaciones de la metodología.

Efecto: Retraso en la entrega del producto.

**RI-23 Perdida de documentos y archivos del proyecto.**

Condición: puede dar el caso en que el disco de una computadora se queme y se borren todos los archivos o un virus lo destruya todo.

Consecuencia: Retraso y pérdida de documentos y archivos pertenecientes al desarrollo de actividades definidas.

Efecto: Retraso en la entrega del producto.

**RI-24 Reposo a causa de problemas de salud.**

Condición: puede darse el caso en el que el personal este con problemas de salud y requiera de reposo para su recuperación.

Consecuencia: Retraso en las tareas asignadas al personal o sin poder terminarse.

Efecto: Retraso en la entrega del producto.

**RI-25 Permiso de ausencias laborales a causa de lutos.**

Condición: puede darse el caso en el que un familiar cercano ha fallecido y el personal requiera de días de ausencia para darle luto al familiar.

Consecuencia: Retraso en las tareas asignadas al personal.

Efecto: Retraso en la entrega del producto.

**RI-26 Problemas financieros.**

Condición: no se cuentan con fondos necesarios para seguir llevando a cabo el desarrollo del proyecto.

Consecuencia: No poder pagar a los personales que llevan a cabo el desarrollo del proyecto.

Efecto: Retraso en la entrega del producto, baja calidad del producto.

## 4.2 Análisis y prioridad de los riesgos.

La meta principal del análisis de riesgos consiste en establecer las prioridades de los elementos de la lista de riesgos y determinar cuál de ellos justifica la reserva de recursos para el planeamiento. Por otro lado la asignación de prioridades a los riesgos permitirá tratar en primer lugar los riesgos más importantes del proyecto.

**4.2.1 Estimación de la probabilidad.**

La probabilidad del riesgo es una medida que calcula la probabilidad de que la situación descripta en el apartado de consecuencias de los riesgos de la declaración de riesgos llegue a producirse de verdad.

Para cuantificar la incertidumbre acerca de la ocurrencia de los riesgos se emplearán las categorizaciones expresadas en lenguaje natural, en base a un rango de probabilidades establecido, como se muestra en la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Rango de probabilidad** | **Promedio para el calculo** | **Expresión de lenguaje natural** | **Valor numérico** |
| de 1% a 10% | 5 % | Baja | 1 |
| de 11 % a 25% | 18 % | Poco probable | 2 |
| de 26% a 55% | 40 % | Media | 3 |
| de 56% a 80% | 68 % | Altamente probable | 4 |
| de 81% a 99% | 90 % | Casi seguro | 5 |

En la siguiente tabla se expresan los riesgos identificados para el proyecto con las probabilidades estimadas subjetivamente para cada uno de ellos.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Riesgo** | **Expresión** | **Probabilidad** |
| RI-01 | Errores en la estimación del presupuesto | Poco probable | 18% |
| RI-02 | Errores en la estimación de la calendarización | Poco probable | 18% |
| RI-03 | Adelantar fecha de entrega | Altamente Probable | 68% |
| RI-04 | Formulación de nuevos requisitos | Poco probable | 18% |
| RI-05 | Eliminación de requisitos aprobados | Poco probable | 18% |
| RI-06 | Errores en la definición de requisitos. | Media | 40% |
| RI-07 | Necesidades de los clientes no recogidas de forma completa | Media | 40% |
| RI-08 | Existencia de funciones que presentan dificultades técnicas | Media | 40% |
| RI-09 | Diseño inadecuado para la realización del proceso de pruebas | Baja | 5% |
| RI-10 | Problemas con la definición de la interfase de usuario | Baja | 5% |
| RI-11 | Aparición de errores por la reutilización de software preexistente | Media | 40% |
| RI-12 | Solicitud de cambios no controlados o incorrectamente evaluados. | Poco Probable | 18% |
| RI-13 | Poca Comunicación con futuros usuarios | Poco Probable | 18% |
| RI-14 | Ausencia de usuarios finales para validar el producto | Baja | 5% |
| RI-15 | Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software | Media | 40% |
| RI-16 | No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados | Media | 40% |
| RI-17 | No definir reglas específicas para la documentación del código fuente. | Media | 40% |
| RI-18 | Dificultades con herramientas y lenguajes nuevos de programación | Media | 40% |
| RI-19 | Problemas con los equipos de hardware para el desarrollo | Media | 40% |
| RI-20 | Despido o renuncia de personal para desarrollar el producto en los tiempos estimados. | Altamente Probable | 68% |
| RI-21 | Dificultad de comunicación por la distribución del personal. | Poco Probable | 18% |
| RI-22 | Inexperiencia del equipo con la metodología de desarrollo | Media | 40% |
| RI-23 | Perdida de documentos y archivos del proyecto | Poco Probable | 18% |
| RI-24 | Reposo a causa de problemas de salud | Poco Probable | 18% |
| RI-25 | Permiso de ausencias laborales a causa de lutos | Baja | 5% |
| RI-26 | Problemas financieros | Media | 40% |

**4.2.2 Estimación del impacto.**

El impacto del riesgo calcula la gravedad de los efectos adversos, la magnitud de una pérdida o el costo potencial de la oportunidad si el riesgo llega a producirse dentro del proyecto.

Para el presente análisis se empleará la escala de medición subjetiva expresada en la siguiente tabla.

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Retraso en la planificación** | **Valor numérico** |
| Insignificante | 1 semana | 1 |
| Marginal | 2 semanas | 2 |
| Medio | 1 mes | 3 |
| Crítico | 2 meses | 4 |
| Catastrófico | Mas de 2 meses | 5 |

En la siguiente tabla se definen el impacto que producirían la ocurrencia de los riegos citados anteriormente:

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgo** | **Impacto** | **Riesgo** | **Impacto** |
| RI-01 | Marginal | RI-14 | Critico |
| RI-02 | Marginal | RI-15 | Medio |
| RI-03 | Medio | RI-16 | Medio |
| RI-04 | Crítico | RI-17 | Medio |
| RI-05 | Medio | RI-18 | Medio |
| RI-06 | Crítico | RI-19 | Medio |
| RI-07 | Media | RI-20 | Critico |
| RI-08 | Medio | RI-21 | Media |
| RI-09 | Marginal | RI-22 | Critico |
| RI-10 | Medio | RI-23 | Catastrófico |
| RI-11 | Medio | RI-24 | Marginal |
| RI-12 | Marginal | RI-25 | Marginal |
| RI-13 | Critico | RI-26 | Medio |
|  |  |  |  |

**4.2.3 Exposición al riesgo.**

La exposición al riesgo calcula la amenaza general que supone el riesgo combinando la información que expresa la probabilidad de una pérdida real con información que indica la magnitud de la pérdida potencial en un único valor numérico.

Para el presente estudio la exposición al riesgo se calcula multiplicando la probabilidad de riesgo por el impacto. Luego se utilizará la magnitud de la exposición al riesgo para clasificar los riesgos.

Magnitud de exposición al riesgo:

Aprox. 1 = bajo riesgo.

Aprox. 2 = riesgo medio.

Aprox. 3 = alto riesgo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Exposición** |
| RI-01 | Errores en la estimación del presupuesto | 5% | 2 | 0.10 |
| RI-02 | Errores en la estimación de la calendarización | 5% | 2 | 0.10 |
| RI-03 | Adelantar fecha de entrega | 68% | 3 | **2.04** |
| RI-04 | Formulación de nuevos requisitos | 18% | 4 | 0.72 |
| RI-05 | Eliminación de requisitos aprobados | 18% | 3 | 0.54 |
| RI-06 | Errores en la definición de requisitos. | 40% | 4 | **1.60** |
| RI-07 | Necesidades de los clientes no recogidas de forma completa | 40% | 3 | 1.20 |
| RI-08 | Existencia de funciones que presentan dificultades técnicas | 40% | 3 | 1.20 |
| RI-09 | Diseño inadecuado para la realización del proceso de pruebas | 5% | 2 | 0.10 |
| RI-10 | Problemas con la definición de la interfase de usuario | 5% | 3 | 0.15 |
| RI-11 | Aparición de errores por la reutilización de software preexistente | 40% | 4 | **1.60** |
| RI-12 | Solicitud de cambios no controlados o incorrectamente evaluados. | 18% | 2 | 0.36 |
| RI-13 | Poca Comunicación con futuros usuarios | 18% | 4 | 0.72 |
| RI-14 | Ausencia de usuarios finales para validar el producto | 5% | 4 | 0.20 |
| RI-15 | Falta de definición de formatos y ejemplos de documentos para todas las entregas definidas como parte del proceso del software | 40% | 3 | 1.20 |
| RI-16 | No documentar todos los resultados de las revisiones técnicas, incluyendo los errores encontrados y recursos empleados | 40% | 3 | 1.20 |
| RI-17 | No definir reglas específicas para la documentación del código fuente. | 40% | 3 | 1.20 |
| RI-18 | Dificultades con herramientas y lenguajes nuevos de programación | 40% | 4 | **1.60** |
| RI-19 | Problemas con los equipos de hardware para el desarrollo | 40% | 3 | 1.20 |
| RI-20 | Despido o renuncia de personal adecuada para desarrollar el producto en los tiempos estimados. | 68% | 4 | **2.72** |
| RI-21 | Dificultad de comunicación por la distribución del personal. | 18% | 4 | 0.72 |
| RI-22 | Inexperiencia del equipo con la metodología de desarrollo | 40% | 3 | 1.20 |
| RI-23 | Perdida de documentos y archivos del proyecto | 18% | 5 | 0.90 |
| RI-24 | Reposo a causa de problemas de salud | 18% | 2 | 0.36 |
| RI-25 | Permiso de ausencias laborales a causa de lutos | 18% | 2 | 0.36 |
| RI-26 | Problemas financieros | 40% | 3 | 1.20 |

## 4.3 Gestión de los Riesgos

La lista de riesgos presentada anteriormente servirá como punto de partida al proceso de administración de riesgos. Cabe mencionar que debido a que la gestión de riesgos es una actividad continua, esta lista es susceptible de presentar modificaciones a medida que avance el proyecto, y como consecuencia de ello deberá ser actualizada a lo largo de todo el ciclo de análisis, planeamiento y supervisión de los riesgos.

El análisis de riesgos presentado en el punto 4.2 examina la amenaza que representa cada riesgo como una ayuda para decidir en qué riesgos es conveniente aplicar una acción.

## . Líneas de Acción

Para ejercer una adecuada gestión y supervisión de los riesgos mencionados anteriormente, se elaborará un Plan de Acción y Un Plan de Contingencias para cada unos de ellos.

El **Plan de Acción** será utilizado para minimizar los riesgos mediante acciones preventivas. La probabilidad que un riesgo ocurra así como el impacto que el mismo podría ocasionar en el proyecto pueden ser mitigados encarando los problemas en forma proactiva.

El **Plan de Contingencia**, por el contrario intenta implementar respuestas rápidas para mitigar los efectos en el caso que los riesgos se concreten, es decir reducir el impacto de los mismos mediante una reacción planeada. Este plan, además definirá ciertos indicadores que permitirán poner en marcha las acciones previstas, es decir, en caso que se verifiquen ciertos disparadores se adoptarán las medidas indicadas.

1. **Riesgo RI-03 Adelantar fecha de entrega.**
   1. *Aspectos a considerar*:
      1. **Por que** el riesgo es importante: Es probable que se adelante la fecha de entrega ya que ha sucedido con otros proyectos que se estuvieron desarrollando
      2. **Que información** se necesita para seguir el estado del riesgo**:**
         * Calendarización de sistema
         * Documento de especificaciones de requisitos de software.
      3. **Quien es responsable** de realizar las actividades de control del riesgo: el responsable es el líder del proyecto junto con los futuros usuarios del sistema.
      4. **Que recursos** se necesitan para realizar las actividades de control del riesgo: para realizar las actividades de control del riesgo no se necesitan recursos económicos extras, pero si herramientas y contacto con los futuros usuarios para analizar las fechas de entrega.
   2. **Plan de Acción**
      1. Contactarse con posibles usuarios del sistema para dejar escrito y firmado la fecha de entrega del proyecto.
      2. Realizar reuniones con el grupo de desarrollo para exponer claramente la calendarización.
   3. **Plan de Contingencia**
      1. Disparador: Adelanto de la fecha de entrega del proyecto:

* Interrogar a quien posee dudas ante la fecha de entrega del proyecto
* Realizar una reunión entre usuarios y desarrolladores para constatar cuál es la nueva fecha de entrega del proyecto y a su vez:
* Asignar a los desarrolladores más carga de tareas ajustando la calendarización del proyecto con el objetivo de terminar el desarrollo del software en la nueva fecha establecida**.**

1. **Riesgo RI-06 Errores en la definición de requisitos.**
   1. *Aspectos a considerar*:
      1. **Por que** el riesgo es importante: Es probable que en primera instancia no se interpreten debidamente los verdaderos requisitos de los futuros usuarios del sistema. Si este riesgo se verifica podría generarse graves inconvenientes en la implementación del sistema.
      2. **Que información** se necesita para seguir el estado del riesgo:
         * Documento de estudio de viabilidad de sistema.
         * Documento de especificaciones de requisitos de software.
      3. **Quien es responsable** de realizar las actividades de control del riesgo: el responsable es el líder del proyecto junto con personal encargado de los requisitos.
      4. **Que recursos** se necesitan para realizar las actividades de control del riesgo: para realizar las actividades de control del riesgo no se necesitan recursos económicos extras, pero si herramientas y contacto con los futuros usuarios para analizar los recursos.
   2. **Plan de Acción**
      1. Contactarse con posibles clientes y usuarios del sistema.
      2. Realizar las consultas y entrevistas necesarias con usuarios para lograr un relevamiento completo de sus necesidades.
      3. Analizar detenidamente cada uno de los requisitos para comprobar la existencia de requerimientos encubiertos o mimetizados con otros requerimientos ya identificados.
      4. Desarrollar y mantener un catálogo de los requisitos identificados donde aparezcan detalles sobre cada uno de ellos, el estado de desarrollo en el que se encuentra, como así también un informe de excepciones.
      5. Realizar reuniones con el grupo de desarrollo para exponer claramente los requisitos identificados.
      6. Construir prototipos para lograr que los usuarios, clientes y desarrolladores conozcan plenamente el producto y validen las características y funcionalidades del mismo antes de iniciar la fase del diseño definitivo del primer prototipo operacional.
      7. Implementar planes de prueba y cruzar información para corroborar que el producto en desarrollo cumple todos y cada uno de los requisitos.
   3. **Plan de Contingencia**
      1. Disparador: Confusión en la interpretación funcional de los prototipos desarrollados: Ante diferencias de interpretación de las funcionalidades del prototipo en desarrollo se deberá:

* Interrogar a quien posee dudas acerca de las funcionalidades del producto.
* Realizar una reunión entre usuarios y desarrolladores para constatar cuál es la correcta visión por parte del usuario de la función donde se encontraron diferencias.
* En caso de necesidad de implementar cambios proceder como lo establece el Plan de Gestión de Configuración**.**

1. **Riesgo: RI-11 Aparición de errores por la reutilización de software preexistente.**
   1. *Aspectos a considerar:*
      1. ***Por que*** el riesgo es importante: para mejorar la eficiencia y economizar costos es aconsejable reutilizar software preexistente. Lo que frecuentemente se utiliza son módulos con determinadas funciones que son utilizadas en el desarrollo, también se puede aprovechar programas que se los puede integrar con el sistema. Pero puede que el software a reutilizar no se integre de manera consistente, lo que puede ser crítico.
      2. ***Que información*** se necesita para corregir el estado del riesgo: la información que se necesita para mantener al riesgo bajo control será la documentación que apoye la construcción de los módulos a reutilizar y la definición de los requisitos del sistema a desarrollar.
      3. ***Quién es responsable*** de realizar las actividades de control del riesgo. El responsable es el líder del proyecto y el personal afectado a tareas relacionadas con el proyecto.
      4. **Que recursos** se necesitan para realizar las actividades de control del riesgo. Para controlar este recurso no son necesarios recursos económicos, pero si es necesaria la documentación antes mencionada y la participación de cierto personal.
   2. *Plan de acción:* para mantener controlado RI-11 deberán adoptar las siguientes medidas de prevención y seguimiento.
      1. Antes de utilizar el software preexistente para implementar funciones al sistema se deberá analizar detenidamente la documentación de diseño del componente.
      2. Seguidamente, se deberá estudiar y analizar si el software a reutilizar cumple con las especificaciones de diseño del sistema.
      3. Se determinarán también los ajustes y modificaciones necesarios para adaptar el modulo a los requerimientos del sistema.
      4. Con esta información el encargado de la integración del módulo, precederá a consultar al líder del proyecto la conveniencia o no de la reutilización del software mencionado.
      5. En caso de que se decida implementarlo se deberá realizar los ajustes necesarios y la documentación que será incluida a la del sistema, tomando como base la del software original.
   3. *Plan de contingencias*: en caso de que se verifique los siguientes disparadores se adoptará las medidas indicadas:
      1. *Disparador:* Aparición de errores en las etapas de implementación o pruebas:
         * Se informará al líder de desarrollo quien se encargará de llevar adelante el control del caso, con la ayuda del equipo de desarrollo.
         * Realizar pruebas del sistema sin la implementación del módulo reutilizado para comprobar si efectivamente es éste el que produce los errores.
         * En el caso de que se compruebe este hecho se deberá evaluar si es conveniente modificar el software reutilizado para que se adapte al sistema, sin generar errores.
2. ***Riesgo*: RI-18 Dificultades con herramientas y lenguajes nuevos de programación.**
   1. *Aspectos a considerar:*
      1. ***Por que*** el riesgo es importante: La falta de conocimientos por parte del personal son importantes ya que pueden demorar el desarrollo en los tiempos estimados.
      2. ***Que*** información se necesita para corregir el estado del riesgo:
         * Bases y conocimientos del personal.
         * Antecedentes de trabajo del personal.
         * Manejo del lenguaje de programación y base de datos por parte del personal.
      3. ***Quien es responsable*** de realizar las actividades de control del riesgo: El responsable de realizar las actividades de control del riesgo es el líder del proyecto.
      4. ***Que recursos*** se necesitan para realizar las actividades de control del riesgo: para realizar un adecuado control de este riesgo se necesitará personal capacitado para validar las funciones desde el punto de vista técnico. Este personal deberá tener amplios conocimientos en cuanto a Sistemas de Bases de Datos, lenguajes de programación, Sistemas expertos, etc.
   2. *Plan de acción:* para mantener controlado el riesgo RI-18 se deberá adoptar las siguientes medidas de prevención:
      1. Antes de integrar nuevos personales para el desarrollo del software, se beberá analizar la capacitación y el conocimiento en cuanto a lenguajes de programación, base de datos, sistemas expertos, etc. Para determinar su integración y que no presente dificultades o perjudique otras partes del sistema.
      2. Una vez realizada la etapa anterior se deberá comprobar que el personal se encuentra lo suficientemente capacitado para llevar a cabo el desarrollo del software.
      3. Con esta información el líder del proyecto decidirá la conveniencia o no de la integración del nuevo personal.
      4. Si se decide integrar al nuevo personal, se deberá comunicar a los demás desarrollares para que le pongan al tanto del sistema y asignarle sus tareas de acuerdo a las necesidades del proyecto.
   3. *Plan de contingencias*: en caso de que se verifiquen los siguientes disparadores se adoptaran las medidas indicadas.
      1. *Disparador*: Dificultades con herramientas y lenguajes nuevos de programación:
         * Informar al líder del proyecto quien se encargará de tomar el control del caso llevando adelante, con la participación del equipo de desarrolladores para llevar adelante las siguientes acciones:
         * Elaborar cursos prácticos para el manejo de herramientas y lenguajes de programación con el fin de que el personal tenga los conocimientos necesarios lo más pronto posible.
3. ***Riesgo*: RI-20 Despido o renuncia de personal para desarrollar el producto en los tiempos estimados.**
   1. *Aspectos a considerar:*
      1. ***Por que*** el riesgo es importante: La ausencia de personal calificado son importantes ya que pueden demorar el desarrollo en los tiempos estimados.
      2. ***Que*** información se necesita para corregir el estado del riesgo:
         * Condiciones económicas, personales y sociales del personal.
         * Manejo del lenguaje de programación y base de datos por parte del personal.
      3. ***Quien es responsable*** de realizar las actividades de control del riesgo: El responsable de realizar las actividades de control del riesgo es el líder del proyecto.
      4. ***Que recursos*** se necesitan para realizar las actividades de control del riesgo: para realizar un adecuado control de este riesgo se necesitará conocer las condiciones económicas y personales del personal.
   2. *Plan de acción:* para mantener controlado el riesgo RI-20 se deberá adoptar las siguientes medidas de prevención:
      1. Tener un informe acerca del estado económico, personal, y social de los personales.
   3. *Plan de contingencias*: en caso de que se verifiquen los siguientes disparadores se adoptaran las medidas indicadas.
      1. *Disparador*: Despido o renuncia del personal en las etapas de desarrollo:
      2. Informar al líder del proyecto quien se encargará de tomar el control del caso llevando adelante, con la participación del equipo de desarrolladores para llevar adelante las siguientes acciones:
      3. Se dividirán las tareas en partes iguales a los demás desarrolladores lo que correspondería al personal ausente con el fin de terminar el desarrollo de software en el tiempo estimado.
4. **Riesgo: RI-23 Perdida de documentos y archivos del programa.**

*Aspectos a considerar*:

* + 1. **Por que** el riesgo es importante: la pérdida de documentos son importantes ya que puede atrasar catastróficamente el desarrollo de software.
    2. **Que** información se necesita para seguir el estado del riesgo:
* Documentos y archivos del desarrollo del software.
  + 1. **Quien** es responsable de realizar las actividades de control del riesgo: El responsable es el líder de proyecto.
    2. **Que Recursos** se necesitan para realizar las actividades de control del riesgo: para tratar este riego se necesitan disponer de herramientas para guardar los archivos. Se puede dedicar un servidor, el cual servirá de repositorio de archivos.
  1. *Plan de acción*: para mantener controlado el riesgo RI-23 se deberán adoptar las siguientes medidas de prevención y seguimiento.
     1. El líder de proyecto deberá realizar reuniones con todos los integrantes del grupo de trabajo, a fin de dilucidar los posibles problemas relacionados con la protección de archivos.
     2. Establecer un mecanismo para llevar adelante las tareas grupales en forma virtual.
     3. Seleccionar herramientas que se adapten a las necesidades.
     4. Utilizar un repositorio con el fin de guardar archivos.
  2. *Plan de Contingencias*: En caso de que se verifiquen los siguientes disparadores se adoptarán las medidas indicadas:
     1. **Disparador**: Perdidade archivos del desarrollo del software.
        + El líder de proyecto deberá realizar una reunión con el grupo responsable, para discutir la manera en que se puede adelantar el proceso de desarrollo y realizar de nuevo todos los archivos perdidos.

# PLAN

# DE

# CONFIGURACION

## Control de la documentación

### Control de la Configuración.

|  |  |
| --- | --- |
| Título: |  |
| Referencia: | Ninguna |
| Autores: | Jorge Rojas  Patricia Fernández |
| Fecha: | 08/05/2015 |

### Histórico de Versiones.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Fecha | Estado | Responsable | Nombre de Archivo |
| 1.0 | 18/04/2015 | Corregido | Jorge Rojas | Plan\_de\_configuracion1.0.docx |
| 2.0 | 08/05/2015 | Revisión | Patricia Fernández | Plan\_de\_configuracion2.0.docx |

### Histórico de Cambios.

|  |  |  |
| --- | --- | --- |
| Versión | Fecha | Cambios |
| 1.0 | 18/04/2015 | Primera versión. |
| 2.0 | 08/05/2015 |  |

## Introducción

## Propósito

El objetivo de este Plan de Gestión de la Configuración (CM), es proporcionar una visión general de la organización, actividades, tareas en general, y los objetivos de Gestión de la Configuración. Se aborda la identificación de los elementos de configuración (CI), control de cambios y las auditorías de configuración en un alto nivel; se proporcionan detalles adicionales sobre las actividades de CM, técnicas y herramientas en los procedimientos de relacionados a CM.

## Alcance

El plan de gestión de la configuración debe involucrar todas las fases del ciclo de vida del software. El documento permitirá mostrar los estándares de etiquetación de los productos de trabajo. Así mismo esclarecerá el tipo de nomenclatura utilizada para el control de las versiones de los documentos que se encuentran dentro de los elementos de la gestión.

Por otra parte cualquier stakeholder podrá presentar cualquiera de los siguientes tipos de peticiones de cambio sobre el sistema, para el control de cambios:

* Petición de cambios en los requerimientos (adiciones, supresiones, modificaciones, aplazamientos) en el software actualmente en desarrollo
* Informes de los problemas en la producción corriente o sistemas de pruebas beta
* Petición de mejoras en los sistemas actuales de producción
* Petición de nuevos proyectos de desarrollo

Este proceso de control de cambio se aplica a los productos de línea base creados o gestionados por los miembros del sistema, incluyendo:

* El software que se ha lanzado a la producción o se encuentra en versión beta
* Requisitos de las especificaciones del sistema
* Grupo de procedimientos y procesos
* Usuarios y documentación técnica

Las siguientes clases de productos de trabajo están exentos de este proceso de control de cambios:

* Los productos de trabajo que están todavía en desarrollo, a excepción de cambios en los requerimientos solicitados en nuevos proyectos.

## Documentos Referenciados

|  |  |
| --- | --- |
| Archivo | Nombre |
| Proceso de Gestión de la Configuración | Plan\_de\_configuraciñon.docx |

Acrónimos

CCB Configuration Control Board

CM Control Management

GCS Gestión de la Configuración del Software

ECS Elementos de la Configuración de Software

CMO Configuration Management Office

CI Configuration Item

## Definiciones

Línea Base Un documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.

Configuration Control Board Revisa y aprueba los cambios sugeridos a un producto.

Petición de Cambio Una solicitud que alguien ha presentado al sistema de control de cambio que describe un problema de software, una mejora solicitada, una propuesta de cambio en los requisitos de un producto en fase de desarrollo, o un nuevo proyecto que se propone.

Stakeholder Persona que directa o indirectamente se ve afectada por el sistema y que puede afectar el  proyecto.

ConfigurationItem Los elementos que son puestos bajo el control de gestión de la configuración.

## Organización

Sistema de Gestión de la Configuración

Git, el Sistema de control de versiones, es una herramienta que se utiliza para almacenar todas las versiones del software y dar seguimiento de los cambios y líneas de base del proyecto.

Githut Es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git.

Personal, Roles y Responsabilidades

**CCB**

La CCB, está integrada por el Desarrollador Líder (Carlos Mathiasen), esta en cuestión tomará decisiones sobre los principales cambios necesarios. Cualquier cambio en los requisitos deberá ser aprobado por el CCB.

* Revisar todas las peticiones de cambio y proporcionar los datos necesarios para determinar la disposición, tal como se describe en la Sección 3.4.
* Asignar las peticiones de cambio aprobadas una fecha de implementación y un equipo.
* Asegurar que las acciones tomadas por las peticiones de cambio son de manera oportuna.

**Presidente CCB**

El presidente de la CCB es el Administrador del Proyecto (Patricia Fernández).

* Dirigir las reuniones CCB.
* Asignar/aprobar la disposición de cada petición de cambio y la asignación de la implementación de la petición de cambio aprobada.
* Asegurarse de que la acción se toma en las solicitudes de cambio de manera oportuna.

**Administrador de la Herramienta de Gestión de la Configuración**

Es el encargado de administrar el sistema de gestión de la configuración (Patricia Fernández), introducir las líneas base, otorgar permisos y administración de usuarios.

**CMO (**Patricia Fernández**)**

* Mantener el Plan de CM
* Identificar los elementos de configuración (CI) y documentar sus características.
* Controlar y facilitar los cambios a las características de un CI.
* Realizar auditorías para verificar el cumplimiento del Plan CM.
* Realizar auditorías para verificar la preparación para la liberación.
* Administrar la base de datos de administración de configuración.
* Informar a la CCB el estado de aprobación de todos los cambios propuestos y el estado de ejecución de todos los cambios aprobados.
* Trabajar con el Presidente y los miembros de CCB para programar reuniones periódicas de CCB, y preparar la agenda para cada reunión.

## Herramientas

Las siguientes herramientas son usadas para administrar las líneas base del proyecto:

* Para gestionar los requisitos, documentos y código fuente. La herramienta utilizada para la gestión de configuración de repositorios es Git.

## Procedimientos

Estimación de tiempo para identificación de Elementos

Con base al ERS del proyecto, el CMO, determino que el tiempo estimado para la identificación de los elementos tomara un total de 1 semana a partir del día 28 de marzo.

Identificar elementos de configuración

La identificación de los elementos de configuración (CI) consiste en determinar los componentes del sistema y documentos, que se pondrán bajo la identificación exclusiva de un identificador único, y hacerlas accesibles al equipo. Un esquema de correcta identificación, otorga a cada elemento trazabilidad entre el elemento y su información de estado.

Cada elemento se identificara de la siguiente manera Proyecto\_TipoDeDocumento\_NombreDelElemento\_Versión (*GCPV*\_CMP\_2.3.1), por ejemplo:

***Proyecto*** *SGRI (Sistema de Gestion para restauran )*

***Tipo de Documento*** *PP (Planes del Proyecto)*

***Nombre del Elemento*** *Plan\_de\_configuracion*

***Versión del Elemento*** *1.0 (Primera Version)*

A continuación se presentan los elementos que se pondrán bajo la gestión de la configuración:

* Diseño
* Código Fuente
* Herramientas
* Documentación del Sistema
* Procedimientos de Prueba
* Resultados de Prueba

Establecer un sistema de administración de configuración

Para seleccionar el sistema que servirá como gestor de la configuración, se tomara en cuenta los siguientes puntos:

* Que la versión del software no sea de prueba o de paga.
* Permita administrar a los usuarios que tendrán acceso al sistema
* Permita otorgar permisos a los usuarios que accederán al sistema
* Que sea un sistema fácil de usar
* Que no sea un plugin de un ambiente de desarrollo (IDE).
* Que se pueda utilizar en distintos sistemas operativos
* Que permita solucionar los conflictos que surjan de una manera eficaz

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sistema | Prueba | Usuarios | Permisos | Usabilidad | Plugin | MultiOS | Resolución de Conflictos | Total |
| Git | No | Si | Si | Si | No | Si | Si | 10 |
| Assembla | No | Si | Si | Si | Si | Si | No | 8 |
| GoogleCode | No | Si | Si | Si | Si | Si | No | 8 |

Tabla 1. Comparación de sistemas de gestión de la configuración

Con respecto a la configuración, se puede ver que el sistema que se usara es Git, ya que es el más completo según el análisis que se realizó.

Crear o liberar las líneas base

Para la creación o liberación de una línea base se debe seguir el proceso que se presenta en la , la persona interesada en la creación o liberación de líneas base debe obtener la autorización de la CCB, cualquier integrante del equipo puede presentar una propuesta para creación o liberación de línea base pero esta solo puede ser autorizada por la CCB, esta propuesta debe de realizarse a través del formato correspondiente, (tal formato se encuentra disponible en el proceso de gestión de la configuración en el apéndice A8).

Si la petición de liberación fue aprobada por el CCB, el administrador de la Herramienta de Gestión de la Configuración deberá de liberar los elementos de línea base que se le soliciten y deberá de registrar la salida de dichos elementos.

El CCB informará qué el conjunto actual de líneas base esté disponible a los interesados.



Ilustración 1. Proceso de liberación de líneas base

Seguir las peticiones de cambio

Cualquier cambio que afecta a los requisitos de línea base debe ser presentado a la CCB como una petición de cambio. La muestra el diagrama de flujo típico de una solicitud de cambio. El CCB debe analizar y considerar cada petición de cambio. Si la solicitud es rechazada, puede ser redefinida y volver a presentarla. Si se acepta, los requisitos serán de nuevo línea base y todos los potencialmente afectados por el cambio serán informados. Esta petición debe de realizarse a través del formato correspondiente, (tal formato se encuentra disponible en el proceso de gestión de la configuración en el apéndice A9).

Cada petición de cambio presentada que sea aceptada se debe revisar o asignar una prioridad, y proporcionar una evaluación de impacto (una estimación aproximada del nivel de esfuerzo requerido para la ejecución, y el impacto de otras actividades actuales y previstas).

Se asigna una prioridad a cada petición de cambio cuando se recibe. La prioridad de una petición de cambio es asignado, ya sea por el autor o por el desarrollador líder. El desarrollador líder tiene la autoridad para modificar la prioridad de cualquier petición de cambio. Un cambio necesario tan pronto como sea posible, sin pasar por el calendario de lanzamientos regulares, se le asigna una prioridad urgente. Todos los cambios, independientemente de su prioridad, siguen el mismo proceso de aprobación.

Al recibir una petición de cambio, el CCB evalúa el cambio, en caso de ser necesario se pone en contacto con el autor del cambio, procesa la solicitud del cambio, y recomienda un calendario para la aplicación de los cambios aprobados.

Presentado

Evaluado

Rechazado

Aprobado

Cambio Hecho

Verificado

Cerrado

Cancelado

Ilustración 2. Proceso de petición de cambio

Controlar los elementos de configuración

La CMO recoge los datos necesarios para producir informes útiles para el CCB, y el administrador del proyecto.

Para la gestión del cambio, la CMO recoge información de identificación correspondiente a cada petición de cambio recibida y su estado en la base de datos de peticiones de cambio, tal como se define en el procedimiento de peticiones de cambio. Después la CMO prepara sus respectivos informes.

Para el estado de los elementos de configuración, la CMO recoge información de identificación correspondiente a cada elemento de configuración controlada, es decir, la versión actual, el historial de revisiones, el subsistema asociado. Al final de cada entrega, los elementos de configuración se actualizan como se define en el procedimiento de identificación de elementos bajo configuración. La CMO prepara informes conforme a lo solicitado sobre el estado de CI, que detalla las nuevas solicitudes de cambio, las solicitudes de cambio recientemente aprobadas, y las solicitudes de cambio cerradas. Se preparan informes de liberación por la CMO para que se ingresen en los informes de la liberación.

Realizar auditorías de configuración

Las auditorías de configuración consisten en revisiones donde se compara el proceso de CM o una configuración de un producto con los requisitos para determinar si estos se cumplen.

Con base a lo anterior, se identifican las siguientes fechas para las auditorias de la gestión de la configuración:

* 05-05-2015
* 19-06-2015

**Auditorías de línea base**

La CMO trabaja con el desarrollador líder para llevar a cabo una auditoría a las líneas base por cada liberación. Esta auditoría de base verifica el contenido de la versión completa, y todos los cambios que deberían realizarse han sido verificados por el equipo. Las revisiones de configuración se realizaran periódicamente para verificar la exactitud de la información de los elementos de configuración. El objetivo de la revisión es para verificar que todos los componentes del sistema han sido correctamente identificados y que todos los cambios en él han sido debidamente administrados. Las revisiones periódicas de configuración también permitirán evaluar la eficacia del proceso de CM y para identificar posibles modificaciones. El CMO seguirá de cerca la configuración de los elementos de configuración y trabajara con los involucrados para asegurar que los procedimientos de gestión de configuración se utilizan adecuadamente. Antes de las liberaciones principales del software, el Jefe de Desarrollo llevará a cabo una auditoría de la configuración mediante la realización de una construcción completa del software desde cero, y volverá a correr el conjunto de pruebas.

A continuación se presentan los tres diferentes tipos de auditorías a líneas base para llevar a cabo:

1. Las auditorías de líneas base:
   * Una auditoría de línea base de se llevará a cabo por la CMO cuando cada línea base se halla establecido y se esté listo para pasar a la siguiente fase del ciclo de vida. Los resultados de esta auditoría se documentarán y se proveen al equipo de desarrollo.
   * Las auditorias de construcción de código serán llevadas a cabo por la CMO cuando una línea base construida esté lista para avanzar a la fase de pruebas. Esta auditoría verificara el contenido de la construcción en comparación con el contenido previsto en la planeación. Los resultados de esta auditoría se documentarán y se suministra con el equipo de pruebas para que conozcan que la aplicación está lista para ser probada.
2. Cerca de la conclusión del proyecto, una auditoría funcional de configuración (FCA) se llevará a cabo por la CMO. La FCA valida que un producto de trabajo cumple con sus requisitos de rendimiento. La CMO analizará la:
   * Matriz de trazabilidad de Requisitos
   * Hojas de certificación y evidencia objetiva de que cada requerimiento se validó
3. Cerca de la conclusión del proyecto, una auditoria física de configuración (PCA) se llevará a cabo por la CMO. El PCA valida la coherencia entre el producto y su documentación técnica. La CMO analizara:
   * Lista de elementos a ser inspeccionados (inventario)
   * Registro de estado (inventario actualizado)

## Herramientas y Técnicas de Implementación del Software

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Caracteristica** | **Uso** |
| Git | Es un sistema de control de versiones distribuido. | Control de versiones |
| Githut | Es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git. | Servidor de versiones |
| Collabtive | Es un software de gestión de proyectos totalmente open sourse. | Gestor de proyectos |
| Scrum | Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. | Metodología de desarrollo |
| Espiral | Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. | Modelo de proceso |
| Python | Se trata de un lenguaje de programación que soporta orientado a objetos. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. | Desarrollo del software |
| Django | Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo-Vista-Controlador.Permite construir aplicaciones web **más rápido y con menos código.** | Framework |
| Heroku | Es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación. | Servidor de prueba |
| ArgoUML | Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD. | Diagramas de casos de usos |
| GanttProject | Es una aplicación que permite organizer y planificar proyectos a través de diagramas Gannt, como asi también la carga horaria de los integrantes dentro del mismo. | Cronograma de actividades |
| COCOMOII | Modelo de costes constructiva, es una algorítmica modelo de estimación de costos de software. | Estimación de costos. |

**Nomenclatura para los archivos del sistema**

El establecimiento de unas normas de nomenclatura comunes para toda la documentación gestionada es un aspecto fundamental dentro del proceso de gestión de la configuración. Con estas normas se intentará establecer de forma lógica la forma de nombrar cualquier entregable documental atendiendo a las siguientes directrices:

• Los nombres de los documentos deben tener una estructura común: Esto ayuda a la identificación rápida de los archivos y a la comprensión de su contenido, ya que se identifica cada parte del nombre con su significado.

◦ Ejemplo: En este caso, cada documento parece utilizar una norma, lo que hace difícil identificar cada dato.

▪ Acta\_20100301.doc

▪ Subvenciones\_ACT\_Marzo2010.doc

▪ Prestaciones\_Acta\_Seguimiento.doc

• Deben poderse identificar de forma unívoca: Si todos los documentos existentes se ubicaran en una misma carpeta, se deberían cumplir que ningún archivo sobrescribiría a otro: lo cual quiere decir que no habría dos archivos con el mismo nombre.

◦ Ejemplo: “Acta\_20100301.doc” si se celebran dos reuniones de distintos proyectos el mismo día, se repetiría este documento.

• Deben ser fáciles de identificar: Hay que buscar un término medio entre un simple código y una descripción detallada para el nombre de los archivos.

◦ Ejemplo: “ANEGDP\_023X23c\_KM.doc”

• No deben ser excesivamente largos: No se debe tender a escribir el contenido del documento en su nombre.

◦ Ejemplo: “Análisis de los cambios solicitados por el usuario para introducir la nueva propuesta de resolución.doc”

• Deben incluir el número de versión como parte del archivo.

**Nomenclatura para los códigos y base de datos**

Nombres de variables

* Se utilizaran nombres significativos.
* Los nombres deben estar en minúsculas, excepto la primera letra de cada palabra a partir de la segunda.
* Se utilizara notación camello.

Ejemplo:

* nuevoEmpleado

Nombres de clases

Cuando se crea una clase esta siempre debe empezar con mayúscula y continuar con notación camello.

Ejemplo:

class CursoAdmin

Nombres de los archivos

Todo en minúscula excepto la primera letra de cada palabra a partir de la segunda. Todos los programas deben tener la extensión .py preferiblemente.

Ejemplo:

cursoAlumno.py

Nombres de las carpetas

Todo en minúscula excepto la primera letra de cada palabra a partir de la segunda.

Ejemplo:

curso

Corchetes e indentación

La indentación es algo que ayuda a darle claridad a un programa y es INDISPENSABLE que se haga bien. Debe hacerse con "tabs" y no con espacios en blanco. Y pytthon te obliga a usarlos o el programa no funciona.

Ejemplo de indentación apropiada:

from django.db import models

from alumnos.models import Alumno

from universidad.models import Materia

class Curso(models.Model):

materia = models.ForeignKey(Materia)

fecha = models.DateField()

created\_at = models.DateField(auto\_now\_add=True)

updated\_at = models.DateField(auto\_now=True)

def \_\_str\_\_(self):

return str(self.fecha)

class CursoPorAlumno(models.Model):

curso = models.ForeignKey(Curso)

alumno = models.ForeignKey(Alumno)

PLAN DE TEST DEL SISTEMA

PRESENTE

**Prefacio**

|  |  |
| --- | --- |
|  |  |
|  | Este documento describe el plan de test del sistema Presente, cuyo objetivo principal desarrollar un sistema web para registrar las asistencias de los alumnos de una manera más cómoda y práctica y en el que se lleve un mejor control de la misma. |
| **Alcance** | Este documento de plan de test es la base de la fase de test del proyecto. Describe los siguientes tópicos:   * Implementación de Interfaz de login. * Implementación de Interfaz de bienvenida. * Implementación del sistema carga de alumnos * Implementación del sistema carga de profesores * Implementación del sistema carga de materias * Implementación del sistema carga de facultades * Implementación del sistema carga de usuarios * Implementación del sistema asignación de roles y permisos a usuarios * Implementación de la Interfaz de reportes de porcentajes de asistencias de alumnos por semestre y materias   Este documento no describe:   * No se contemplan más servicios de los ya mencionados anteriormente. * No se contemplan la unión de este sistema con otro ya utilizado para gestión de alumnos. |

**Historia del Documento**

|  |  |  |  |
| --- | --- | --- | --- |
| **FECHA** | **VERSION** | **DESCRIPCION** | **AUTOR** |
| **08/04/2015** | 1.0 | El software cuenta con todas las interfaces funcionales a excepción de la interfaz de reportes de asistencias. | **Carlos Mathiasen** |
| **01/05/ 2015** | 1.0 | Documentación de Pruebas | **Jorge Sánchez** |
| **20/05/2015** | 2.0 | Documentación de Pruebas | **Patricia Fernández** |

INTRODUCCION

## Propósito

El software a desarrollar se basa en la necesidad de contar con un sistema web de asistencia en el cual se podrán cargar las asistencias de los alumnos de cada carrera y materia correspondiente de la Universidad Autónoma de Encarnación (UNAE).

También se podrán ver las asistencias de manera gráfica y saber si cada alumno tiene derecho a rendir para lo cual debería tener un 75% de asistencia, de esta manera será mucho más fácil controlar las asistencias y se podrá brindar una mejor atención a los alumnos.

Objetivos del Plan

EL Plan de Test del Sistema especifica los procesos de test y de verificación que se realizaran con el objeto de:

* Identificar defectos y fallas.
* Medir rendimiento.
* Evaluar la calidad
* Determinar el cumplimiento de los requerimientos.

Los objetivos de este plan son:

* Definir y detallar todas las tareas que se desarrollarán para probar el sistema.
* Definir el plan y la persona o grupo responsable de cada tarea.
* Definir las herramientas de prueba y el ambiente necesario a la conducción de las actividades de test.

Definir los ítems y funcionalidades que serán probados.

Alcance del Testing

El Plan de Testing del Sistema es una especificación de alto nivel de los requerimientos funcionales y de calidad que serán probados, del ambiente de testing, de la estrategia de testing, de las responsabilidades y de los criterios de éxito.

El comportamiento de un producto bajo testing será comparado con las especificaciones de los requerimientos que fueron usados para implementar el sistema, incluyendo todos los cambios que han sido aprobados e implementados.

Los casos de prueba y los criterios de éxito serán derivados de este plan general y serán especificados en el documento de Especificaciones de Testing del Sistema.

El alcance del test del sistema es probar la funcionalidad completa y el rendimiento del Sistema Presente.

Criterios de Entrada

Para poder comenzar la fase de pruebas del sistema, se deben cumplir los siguientes criterios:

* Test unitarios realizados y completados para cada componente del sistema.
* Sistema completamente integrado.
* Software congelado.

Hardware congelado.

Criterios de Salida

Los criterios de salida deberán ser:

* Menor cantidad de errores posibles por cada test unitario.
* No contar con errores de sentencias sql.
* La longitud de los campos de cada interfaz deben se limitados.
* El tipo de dato debe ser el correcto para cada caso.
* En la interfaz de login solo deberá ingresar el usuario con la contraseña correcta, en caso de ingresar de forma incorrecta uno de los campos se deberá pedir que se reingresen los dos campos.
* Los registros ingresados en tablas deberán ser visualizados sin inconvenientes.
* La eliminación de alumnos en la interfaz alumnos deberá realizada con una alerta de ejecución.
* En las demás interfaces todos los requisitos del testeo deberán cumplirse, ya sea longitud de campos, tipo de datos, etc.

## Test del Sistema

El sistema a probar se define como Presente

Estrategia de Test del Sistema

El sistema será testeado a través de planes de testeos unitarios, tratando de testear metódica y sistemáticamente cada interfaz del sistema.

Pruebas en Operación Normal

* El sistema debe tener como mínimo la funcionalidad de ingresar al sistema con el usuario determinado.
* Debe poder guardar lo ingresado en los campos en una base de datos.
* Debe poder filtrar la búsqueda en las listas de usuario, materias, etc.
* Debe poder ingresar a cada interfaz sin problemas mayores.

Pruebas en Condiciones de Excepción

* Manejará como mínimo las excepciones de alertar al usuario al ingresar al sistema si uno de los campos recibe valores no admitidos o si se excede la longitud permitida para cada campo.
* Al elegir la opción eliminar ya sea alumnos, usuarios, materias, etc, el sistema deberá alertar al usuario de la acción a ejecutarse y deberá pedirle que acepte dicha acción para ejecutar lo requerido.

Criterios de Éxito/Fracaso de Pruebas

Se puede considerar una prueba pasada satisfactoriamente al cumplir con los requisitos necesarios que se listaron en este documento.

Entregables

El documento en donde se plasmarán los resultados de las pruebas realizadas al sistema contara con las pruebas pasadas, y en caso de que no hayan pasado algunas de las pruebas requeridas se le informará al programador de haber hecho las funciones y también al programador de turno. Al resolverse las falencias del sistema se ejecutaran nuevamente cada una de las pruebas.

## Configuración del Test

Esta sección establece los componentes del ambiente de testing

Hardware

* Sistema Operativo de 32 bits Windows 7
* Disco duro 250
* Memoria RAM 4GB

Software

* Google Chrome versión 42.0.2311.152
* Python 3.4
* Django
* Consola cmd

## Tareas

Actividades

La secuencia de actividades para probar el sistema es:

* Tener hardware necesario para las pruebas
* Ejecutar cada uno de los planes de testeo unitarios
* Reportar los errores de manera a corregirlos y volver a testear
* Desarrollar las pruebas finales

Responsabilidades

Esta sección establece las responsabilidades de cada grupo que participa en la fase de pruebas.

Responsabilidades del Grupo de Desarrollo

* Ejecutar las pruebas unitarias
* Ejecutar y probar la integración de bajo nivel
* Corregir los problemas reportados

Responsabilidades del Grupo de Testing

* Planificar las pruebas del sistema
* Configurar el ambiente de prueba
* Ejecutar las pruebas del sistema
* Escribir el reporte de test

Responsabilidades de la Gerencia

* Proveer recursos
* Aceptación final y aprobación de la liberación del producto

## Planificación

Para cada testeo unitario se estima un promedio de 2 días para ejecutar y generar el reporte de errores a ser corregidos.