



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

**DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN *WEB* PARA EL
CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA
FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

Henry Waldemar Sontay Chan

Asesorado por el Ing. Miguel Marín de León

Guatemala, mayo de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN *WEB* PARA EL CONTROL DE
AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

HENRRY WALDEMAR SONTAY CHAN
ASESORADO POR EL ING. MIGUEL MARÍN DE LEÓN

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Angel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	P.A. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Cesar Augusto Fernández
EXAMINADOR	Ing. Edgar Estuardo Santos
EXAMINADOR	Ing. Cesar Batz
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

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

DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN *WEB* PARA EL CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2010.

Henry Waldemar Sontay Chan

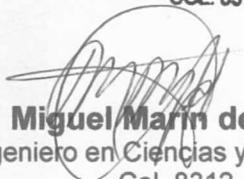
Guatemala, 6 de Noviembre de 2010

Ing. Carlos Azurdia
Coordinador de Revisión de Trabajos de Graduación
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **HENRRY WALDEMAR SONTAY CHAN**, titulado: **"DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Atentamente,

Miguel Marín de León
INGENIERO EN CIENCIAS Y SISTEMAS
COL. 8312


Miguel Marín de León
Ingeniero en Ciencias y Sistemas
Col. 8312



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

Guatemala, 23 de Febrero de 2011

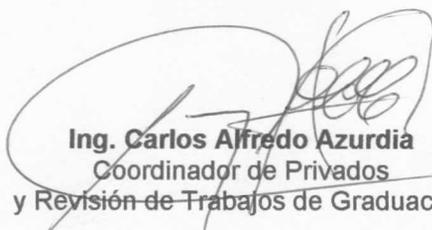
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **HENRRY WALDEMAR SONTAY CHAN**, carné 1997-12993, titulado: "**DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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

Atentamente,


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



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

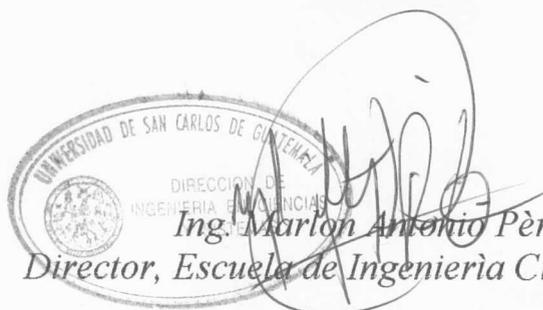
UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL.: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, presentado por el estudiante HENRRY WALDEMAR SONTAY CHAN, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”


Ing. *Marlon Antonio Pérez Turk*
Director, Escuela de Ingeniería Ciencias y Sistemas

Guatemala, 25 de mayo 2011



DTG. 151.2010

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE AUXILIARES DE LA ESCUELA DE SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario **Henry Waldemar Sontay Chan**, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Recinos
Decano

Guatemala, 25 de mayo de 2011.



/gdech

ACTO QUE DEDICO A:

Dios	Inteligencia suprema y causa primera de todas las cosas.
Mis padres	Atanacio Sontay Tzunux y Margarita Chan Capriel.
Hermanos	Vilma, Gerberth, Danilo, Mayra, Luis, Ana, Yovany, Randolpho y Dinora.

AGRADECIMIENTOS A:

- Dios** Por haberme dado la oportunidad de venir nuevamente a este plano terrenal y poner a mi alcance la antorcha espírita para ver mis imperfecciones que retardan mi adelantamiento en esta vida y en la otra.
- Mis padres** Por la vida que me concedieron y los sacrificios realizados.
- Guías espirituales** Por ser mis protectores en las pruebas de esta vida.
- Mis amigos** Por el apoyo brindado.
- Asesor** Por el apoyo recibido en el área de especialización.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	III
GLOSARIO	V
RESUMEN.....	VII
OBJETIVOS.....	IX
INTRODUCCIÓN	XI
1. METODOLOGÍA DE DESARROLLO.....	1
1.1. Definición	1
1.2. Historia.....	1
1.3. Características	2
1.3.1. Proceso dirigido por casos de uso.....	3
1.3.2. Proceso centrado en la arquitectura.....	4
1.3.3. Proceso iterativo e incremental	6
1.4. Estructura del proceso	7
1.4.1. Estructura dinámica del proceso	8
1.4.2. Estructura estática del proceso	12
1.5. Configuración de <i>RUP</i> al proyecto.....	18
1.5.1. Entregables del proyecto.....	18
1.5.2. Esquema de trazabilidad.....	20
2. REQUISITOS DEL SISTEMA	23
2.1. Modelo de casos de uso	23
2.1.1. Catedrático	23
2.1.2. Administrador	24
2.1.3. Auxiliar.....	25

2.1.4.	Estudiante	26
2.2.	Descripción de casos de uso	27
2.3.	Interfaces <i>web</i>	47
2.3.1.	Interfaces de contenido de información	47
2.3.2.	Interfaces de estructura de navegación	48
2.3.3.	Interfaces de entrada de datos.....	49
3.	ANÁLISIS DEL SISTEMA.....	51
3.1.	Diagrama de clases	51
3.2.	Trazabilidad de casos de uso a clases	53
4.	DISEÑO DEL SISTEMA.....	55
4.1.	Diseño lógico de la base de datos	55
4.2.	Arquitectura de la aplicación.....	56
4.3.	Arquitectura del código fuente	57
4.3.1.	Estructura de la aplicación	57
4.3.2.	Estructura de módulos	58
4.3.3.	Estructura del sitio <i>web</i>	59
5.	HERRAMIENTAS UTILIZADAS	61
	CONCLUSIONES.....	63
	RECOMENDACIONES.....	65
	BIBLIOGRAFÍA.....	67

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Historia de <i>RUP</i>	2
2.	Los casos de uso integran el trabajo	3
3.	Trazabilidad a partir de los casos de uso.....	4
4.	Evolución de la arquitectura del sistema.....	5
5.	Proceso iterativo e incremental.....	7
6.	Estructura de <i>RUP</i>	8
7.	Fases e hitos en <i>RUP</i>	9
8.	Distribución de recursos por fases en <i>RUP</i>	12
9.	Relación entre roles, actividades y artefactos.....	12
10.	Detalle de un <i>workflow</i>	13
11.	Trazabilidad entre artefactos.....	21
12.	Caso de uso asociado al catedrático	23
13.	Casos de uso asociado al administrador	24
14.	Caso de uso asociado al auxiliar	25
15.	Caso de uso asociado al estudiante	26
16.	Esquema de las interfaces de contenido de información.....	47
17.	Esquema de las interfaces de estructura de navegación.....	48
18.	Esquema de las interfaces de entrada de datos	49
19.	Interfaces del sitio <i>web</i>	50
20.	Diagrama de clases <i>Framework Symfony</i>	51
21.	Diagrama de clases de estructura de la aplicación.....	52

22.	Diseño lógico de la base de datos.....	55
23.	Arquitectura de la aplicación <i>MVC</i>	56

TABLAS

I.	Esfuerzo en actividades según fase del proyecto.....	11
II.	Descripción de caso de uso gestionar usuario.....	27
III.	Descripción de caso de uso gestionar curso.....	29
IV.	Descripción de caso de uso gestionar auxiliar.....	32
V.	Descripción de caso de uso gestionar catedrático.....	35
VI.	Descripción de caso de uso gestionar asignación.....	38
VII.	Descripción de caso de uso gestionar estadísticas.....	43
VIII.	Descripción de caso de uso gestionar reportes.....	45
IX.	Trazabilidad de casos de uso del diagrama de clases.....	53
X.	Estructura de la aplicación.....	57
XI.	Estructura de módulos.....	58
XII.	Estructura del sitio <i>web</i>	59

GLOSARIO

Caso de uso	Técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de <i>software</i> . Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.
Clase	Declaración o abstracción de objeto, lo que significa, que una clase es la definición de un objeto.
<i>Framework</i>	Estructura de soporte definida en la cual otro proyecto de <i>software</i> puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado por otro <i>software</i> para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
MySQL	Sistema de gestión de base de datos relacional, multihilo y multiusuario.

ORM	Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.
Patrones de diseño	Soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.
Sitio web	Conjunto de páginas <i>web</i> , típicamente comunes a un dominio o subdominio en la <i>World Wide Web</i> .
XML	<i>Extensible Markup Language</i> (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el <i>World Wide Web Consortium (W3C)</i> .
PHP	Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas <i>web</i> dinámicas.

RESUMEN

La información de las actividades realizadas por los auxiliares de la Escuela de Sistemas de la Facultad de Ingeniería y las estadísticas respecto a las evaluaciones de tales actividades son muy importantes, debido a que permiten, tanto al catedrático ver el comportamiento de las evaluaciones, como a la coordinación de auxiliares evaluar el rendimiento de los auxiliares.

Actualmente esta información es presentada en un informe mensual escrito, lo cual repercute en tiempo de elaboración, revisión y calificación. Por lo que, es necesario contar con una herramienta que permita obtener vía *web* la información de una manera más fácil, y poder hacer uso de esta información en cualquier momento y en cualquier lugar.

Este trabajo consiste en implementar una aplicación *web* para mejorar el control de actividades y estadísticas presentadas por los auxiliares de cátedra de la Escuela de Ciencias y Sistemas.

OBJETIVOS

General

Implementar un proyecto *web* aplicando los conocimientos adquiridos durante la carrera, y el cual permita mejorar el control de reportes estadísticos presentados por los auxiliares de cátedra de la Escuela de Ciencias y Sistemas.

Específicos

1. Obtener información vía *web* de las distintas actividades y estadísticas realizadas por los auxiliares de cátedra.
2. Diseñar una interfaz que permita visualizar en forma gráfica las estadísticas obtenidas vía *web*.
3. Aplicar la metodología *Rational Unified Process (RUP)* en el contexto de una arquitectura de implementación.

INTRODUCCIÓN

Dada la importancia del uso de una metodología para el desarrollo de un proyecto de *software*, se ha optado por la metodología *RUP* (*Rational Unified Process*), para el desarrollo del Sistema de Control de Auxiliares.

El presente trabajo de graduación consta de cinco capítulos. En el capítulo uno se describe el uso de la metodología, sus características esenciales, su estructura de proceso y la configuración de la metodología a la aplicación.

En el capítulo dos se establecen los requisitos del sistema, definiendo los casos de usos necesarios.

En el capítulo tres se realiza el análisis del sistema, definiendo el modelo de clases basados en los casos de usos del capítulo anterior.

En el capítulo cuatro se realiza el diseño del modelo relacional y se define la arquitectura de la aplicación.

En el capítulo cinco se describen las herramientas utilizadas para la implementación del sistema.

1. METODOLOGÍA DE DESARROLLO

1.1. Definición

Para el desarrollo y documentación del proyecto se ha utilizado como referencia la metodología *Rational Unified Process*, *RUP*, el cual es un producto desarrollado por *Rational*, hoy propiedad de IBM. En este capítulo se presenta una breve introducción a esta metodología, sus características principales y estructura de proceso.

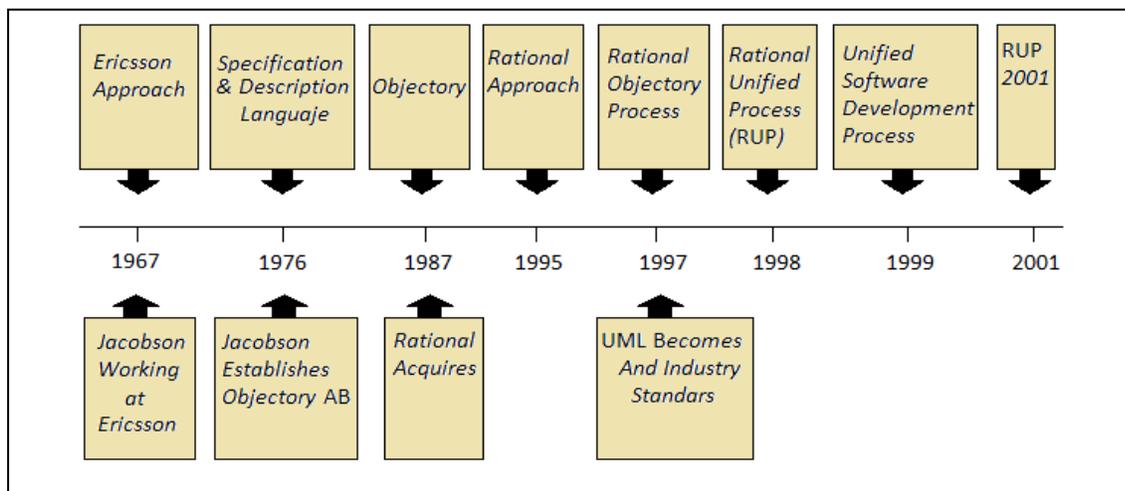
1.2. Historia

La figura 1 ilustra la historia de *RUP*. El antecedente más importante se ubica en 1967 con la Metodología Ericsson (*Ericsson Approach*) elaborada por Ivar Jacobson, una aproximación de desarrollo basada en componentes que introdujo el concepto de caso de uso. Entre los años de 1987 a 1995 Jacobson fundó la compañía *Objectory AB* y lanza el proceso de desarrollo *Objectory* (abreviación de *Object Factory*).

Posteriormente en 1995 *Rational Software Corporation* adquiere *Objectory AB* y entre 1995 y 1997 se desarrolla *Rational Objectory Process (ROP)* a partir de *Objectory 3.8* y del Enfoque *Rational (Rational Approach)* adoptando UML como lenguaje de modelado.

Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, *Rational Software* desarrolló e incorporó diversos elementos para expandir *ROP*, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza *Rational Unified Process*.

Figura 1. Historia de *RUP*



Fuente: elaboración propia.

1.3. Características

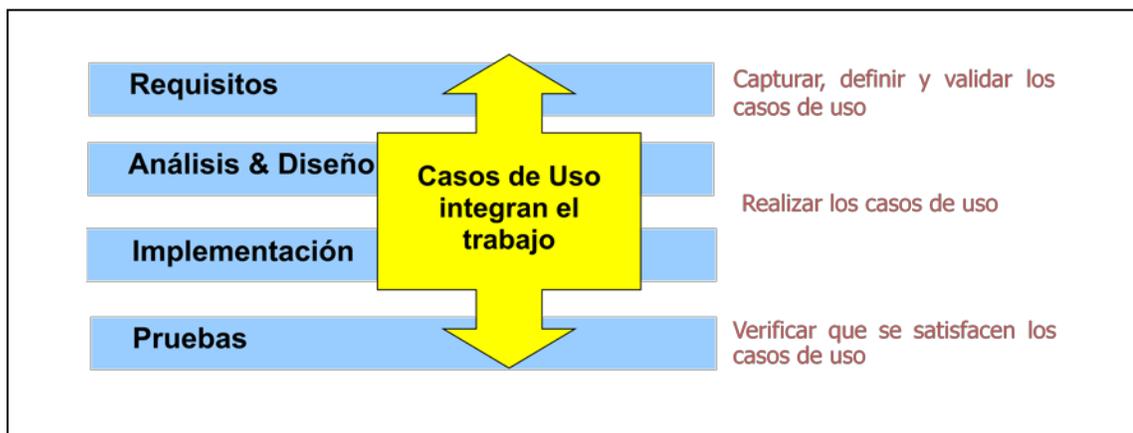
Los autores de *RUP* destacan que el proceso de *software* propuesto por *RUP* tiene tres características esenciales: está dirigido por los casos de uso, está centrado en la arquitectura, y es iterativo e incremental.

1.3.1. Proceso dirigido por casos de uso

El caso de uso es una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones a contemplar. Los casos de uso representan los requisitos funcionales del sistema.

RUP esta manejado por casos de uso, porque el desarrollador crea modelos de implementación y diseño que harán efectivos a los casos de uso planteados en la etapa de análisis. Cada modelo del proceso tendrá una correspondencia con el modelo de casos de uso; éstos constituyen un elemento integrador como se ilustra en la figura 2.

Figura 2. Los casos de uso integran el trabajo

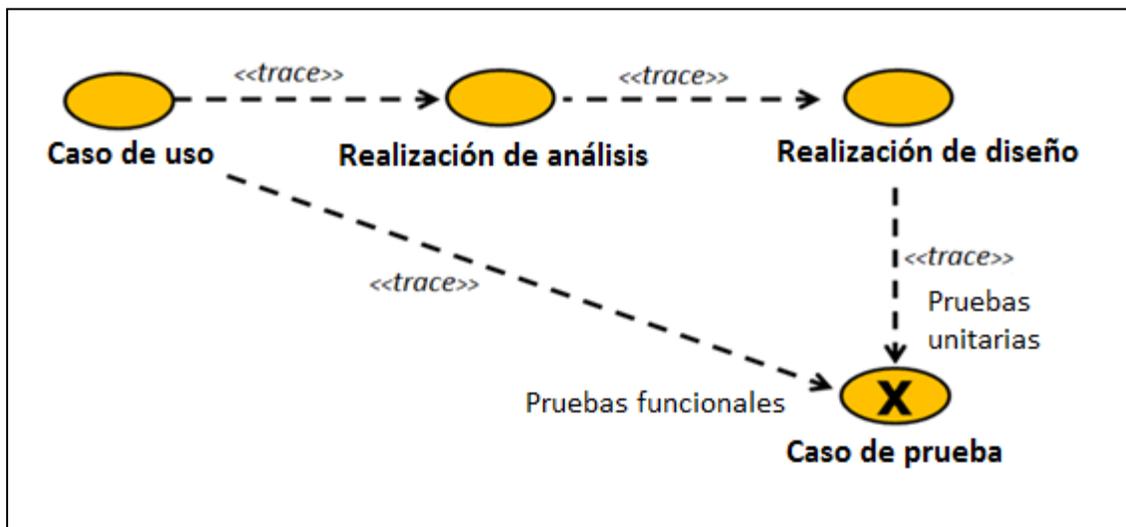


Fuente: elaboración propia.

Los casos de uso no sólo inician el proceso de desarrollo, sino que, proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como se muestra en la figura 3 basándose en los casos de uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso. Todos los modelos deben estar sincronizados con el modelo de casos de uso.

Figura 3. Trazabilidad a partir de los casos de uso



Fuente: elaboración propia.

1.3.2. Proceso centrado en la arquitectura

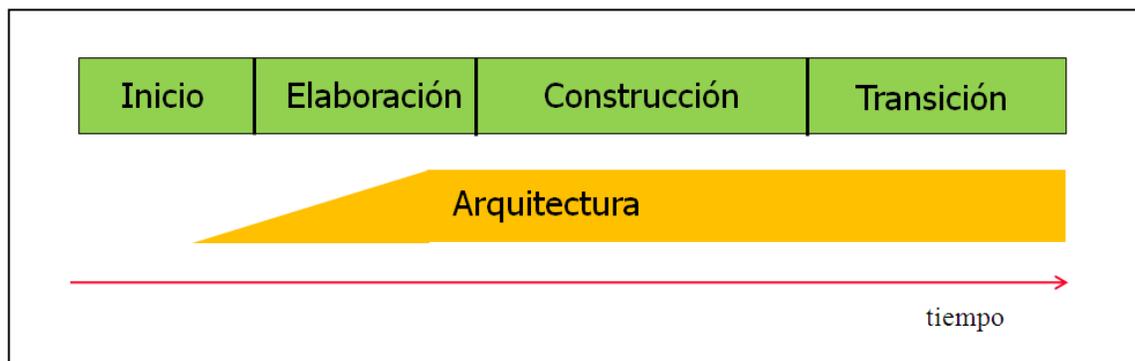
La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema. En el caso de *RUP*, además de utilizar los casos de uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los casos de uso y la forma la proporciona la arquitectura. Existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como casos de uso deban evolucionar en paralelo durante todo el proceso de desarrollo de *software*.

En la figura 4 se ilustra la evolución de la arquitectura durante las fases de *RUP*. Se tiene una arquitectura más robusta en la fase final del proyecto. En la fase inicial se va consolidando la arquitectura por medio de *baselines* y se va modificando dependiendo de las necesidades del proyecto.

Figura 4. **Evolución de la arquitectura del sistema**



Fuente: elaboración propia.

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño, por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para *RUP*, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de casos de uso que es la que da cohesión a todas.

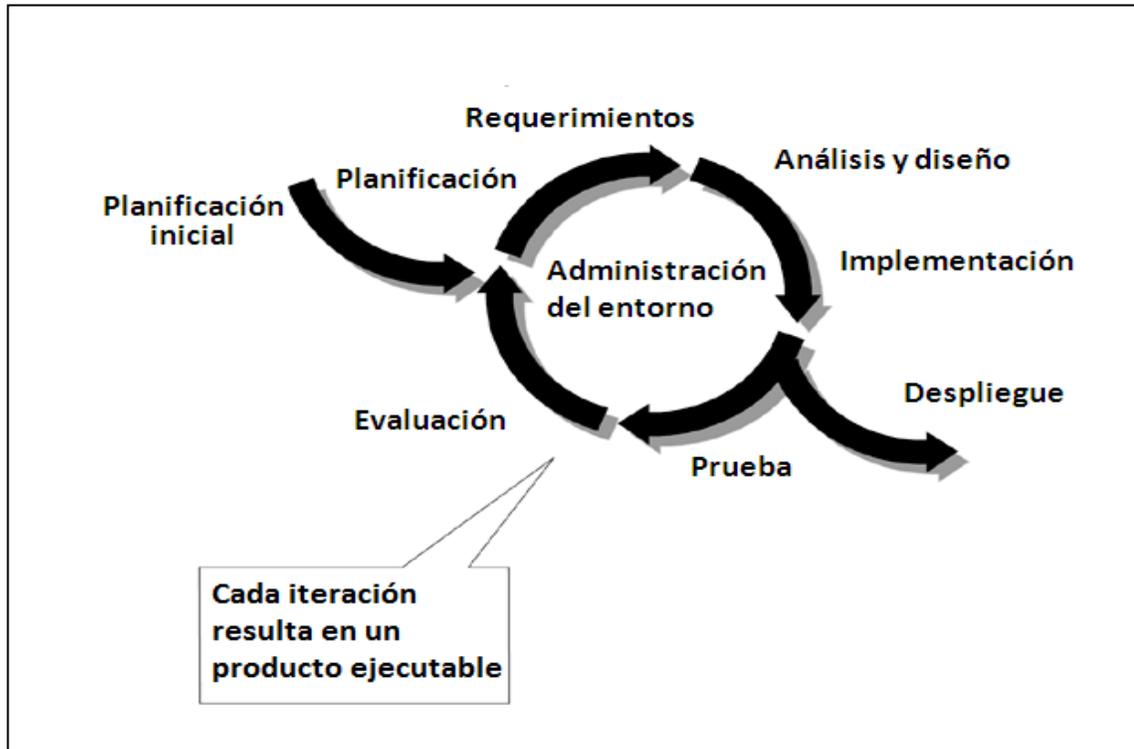
Al final de la fase de elaboración se obtiene una *baseline* de la arquitectura donde fueron seleccionados una serie de casos de uso arquitectónicamente relevantes (aquéllos que ayudan a mitigar los riesgos más importantes, aquéllos que son los más importantes para el usuario y aquéllos que cubran las funcionalidades significativas)

1.3.3. Proceso iterativo e incremental

El proceso unificado es un marco de desarrollo iterativo e incremental, compuesto de cuatro fases denominadas inicio, elaboración, construcción y transición. Cada una de estas fases es a su vez, dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: análisis de requisitos, diseño, implementación y prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

Figura 5. **Proceso iterativo e incremental**



Fuente: elaboración propia.

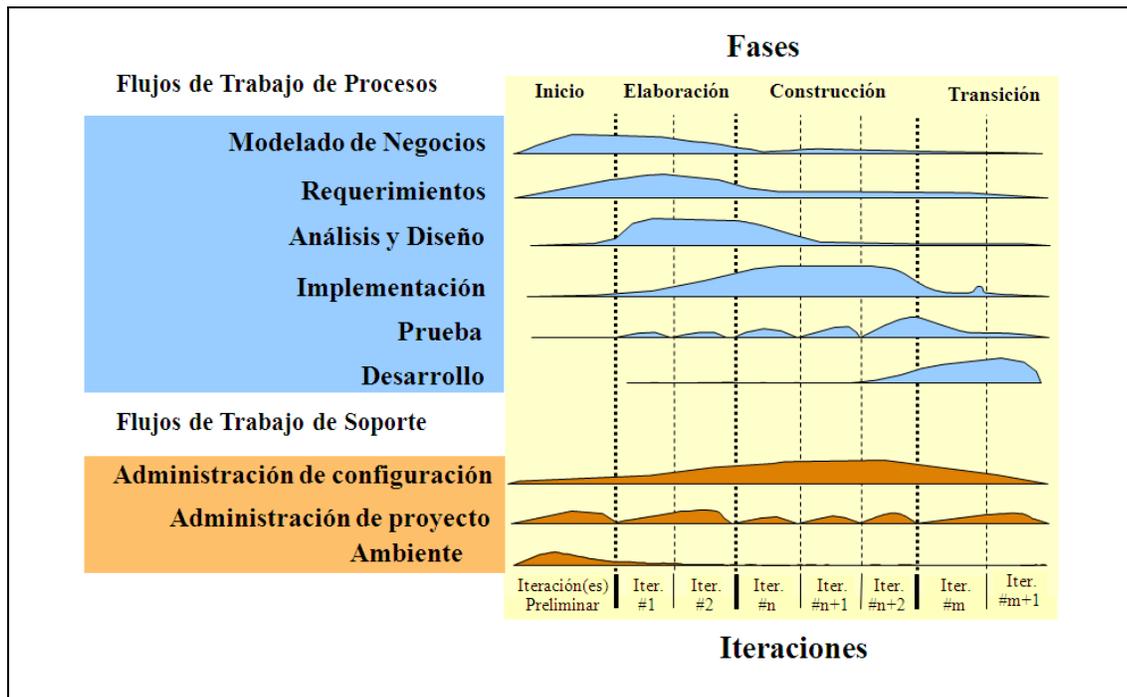
1.4. Estructura del proceso

El proceso puede ser descrito en dos dimensiones o ejes [1]:

- Eje horizontal: representa el eje dinámico del proceso, expresado en términos de ciclo, fases, iteraciones e hitos.
- Eje vertical: representa el aspecto estático del proceso. Describe el proceso en términos de componentes de proceso, actividades, disciplinas, artefactos y roles.

Se puede observar en la figura 6 que *RUP* consta de cuatro fases: inicio, elaboración, construcción y transición y que cada fase se subdivide a la vez en iteraciones.

Figura 6. Estructura de *RUP*



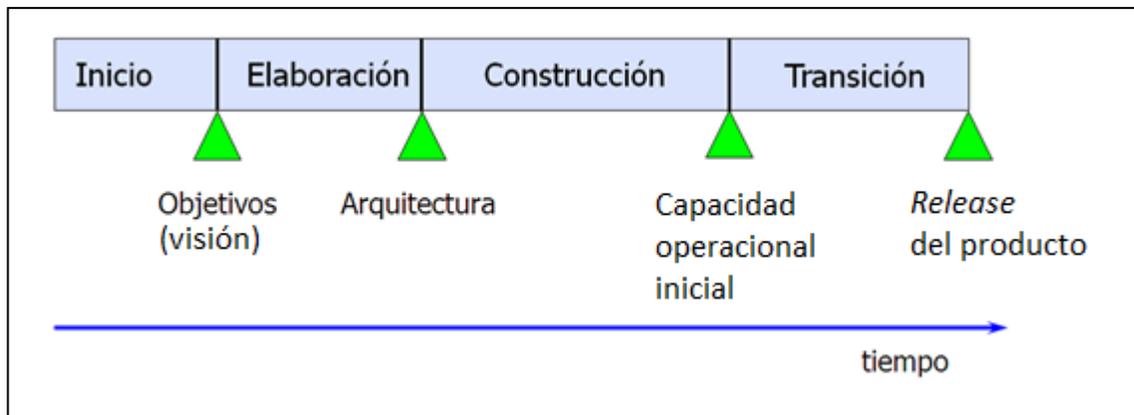
Fuente: elaboración propia.

1.4.1. Estructura dinámica del proceso

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Las fases y sus respectivos hitos se ilustran en la figura 7.

Figura 7. Fases e hitos en *RUP*



Fuente: elaboración propia.

- Fase de inicio: suele ser la más corta del desarrollo, y no debería alargarse demasiado en el tiempo. En caso contrario, podríamos encontrarnos en una situación de excesiva especificación inicial, yendo en contra del enfoque relativamente ágil del proceso unificado.

En esta fase se realizan las siguientes tareas:

- Desarrollar una descripción del producto final y presentar el análisis de negocio
- Realizar una identificación inicial de riesgos

- Establecen las principales funciones del sistema para los usuarios más importantes, la arquitectura a grandes rasgos y un plan de proyecto

La fase de inicio termina con el hito de los objetivos del desarrollo.

- Fase de elaboración: durante esta fase deberían capturarse la mayoría de requisitos del sistema, aunque los objetivos principales son tratar los riesgos ya identificados y establecer y validar la base de la arquitectura del sistema. Esta base se llevará a cabo a través de varias iteraciones, y servirá de punto de partida para la fase de construcción. La fase de elaboración termina, por tanto, al alcanzar el hito de la arquitectura del sistema.
- Fase de construcción: es la más larga del proyecto y completa la implementación del sistema tomando como base la arquitectura obtenida durante la fase de elaboración. A partir de ella, las distintas funcionalidades son incluidas en distintas iteraciones, al final de cada una de las cuales se obtendrá una nueva versión ejecutable del producto. Por tanto, esta fase concluye con el hito de obtención de una funcionalidad completa, que capacite al producto para funcionar en un entorno de producción.
- Fase de transición: en la fase final del proyecto se lleva a cabo el despliegue del producto en el entorno de los usuarios, lo que incluye la formación de éstos.

En lo relativo a la evolución del propio producto *software*:

- Gracias a las opiniones obtenidas de versiones preliminares, evoluciona desde la fase *beta* a una versión final;
- Se resuelven incidencias en la implantación e integración, y si existen, se clasifican aquellas que podrían justificar una nueva versión del producto.

Esta fase concluye con el hito de publicación del producto.

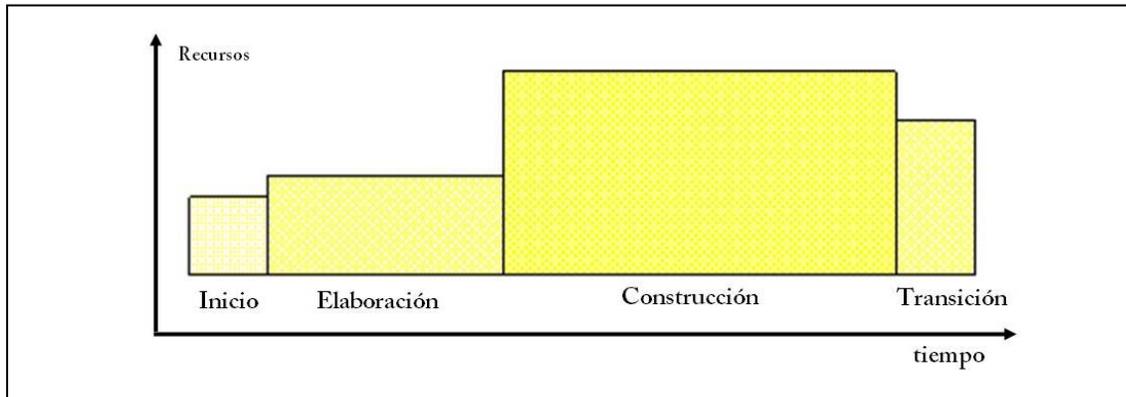
La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, la tabla 1 ilustra porcentajes frecuentes al respecto; consecuente con el esfuerzo señalado; la figura 8 ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

Tabla I. **Esfuerzo en actividades según fase del proyecto**

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10%
Tiempo dedicado	10 %	30 %	50 %	10%

Fuente: elaboración propia.

Figura 8. **Distribución de recursos por fases en RUP**

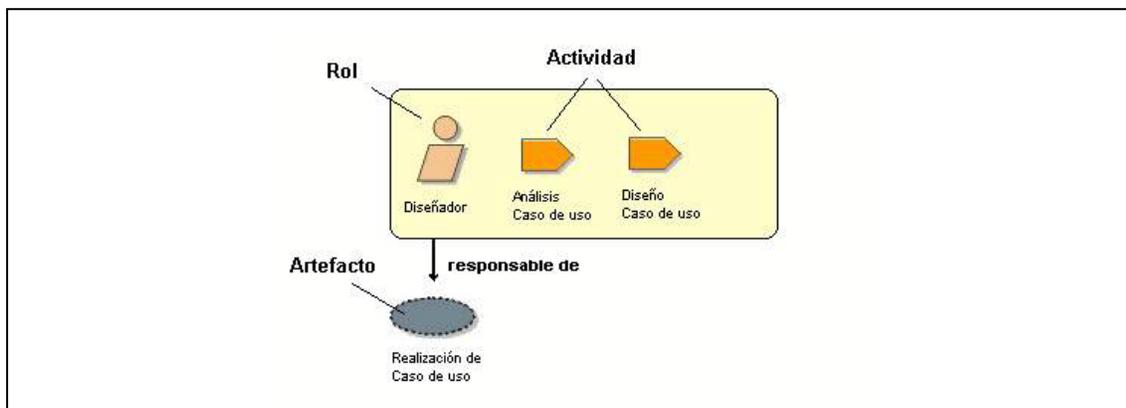


Fuente: elaboración propia.

1.4.2. Estructura estática del proceso

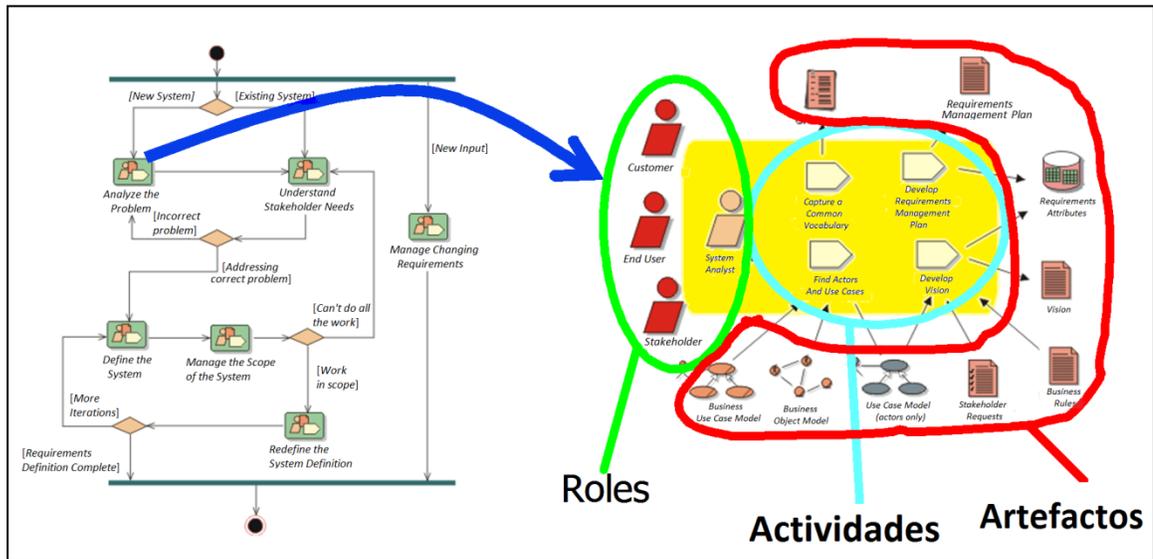
Un proceso de desarrollo de *software* define quién hace qué, cómo y cuándo. *RUP* define cuatro elementos: los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? (ver figura 9 y 10).

Figura 9. **Relación entre roles, actividades y artefactos**



Fuente: elaboración propia.

Figura 10. Detalle de un *workflow*



Fuente: elaboración propia.

- Roles: un rol define el comportamiento y responsabilidades de un individuo o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.
- Actividades: actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.
- Artefactos: un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de *software*. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final.

- **Flujos de trabajo:** con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables.
- **Modelado del negocio:** describe como desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de casos de uso del negocio y un modelo de objetos del negocio. Complementario a estos modelos, se desarrollan otras especificaciones tales como un glosario.
- **Requerimientos:** es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que construyamos. En esta línea los requerimientos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requerimientos que especifiquemos.

Los requerimientos se dividen en dos grupos: los requerimientos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de casos de uso. Los requerimientos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requerimientos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requerimientos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

- Análisis y diseño: consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requerimientos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requerimientos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los casos de uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes. Además, si el sistema usara una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas.

Otro producto importante de este flujo es la documentación de la arquitectura de *software*, que captura varias vistas arquitectónicas del sistema.

- Implementación: en este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

- Pruebas: flujo de trabajo encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

Las actividades de este flujo comienzan pronto en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de las prueba en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

El desarrollo del flujo de trabajo consistirá en planificar que es lo que hay que probar, diseñar cómo se va a hacer, implementar lo necesario para llevarlos a cabo, ejecutarlos en los niveles necesarios y obtener los resultados, de forma que la información obtenida nos sirva para ir refinando el producto a desarrollar.

- **Despliegue:** en este flujo de trabajo se desarrolla con mayor intensidad en la fase de transición, ya que el propósito del flujo es asegurar una aceptación y adaptación sin complicaciones del *software* por parte de los usuarios. Su ejecución inicia en fases anteriores para preparar el camino, sobre todo con actividades de planificación, en la elaboración del manual de usuario y tutoriales.

El objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios.

- **Gestión del proyecto:** es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.
- **Configuración y control de cambios:** la finalidad de este flujo de trabajo es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

- Entorno: la finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

1.5. Configuración de *RUP* al proyecto

En este apartado se describe la configuración de *RUP* que se ha utilizado para el proyecto. Por las características del proyecto, se han incluido muy pocos artefactos, roles y actividades de la metodología, manteniendo los más esenciales. Dicha configuración está basada en la siguiente selección de artefactos:

1.5.1. Entregables del proyecto

A continuación se describen brevemente cada uno de los artefactos que se generarán y usarán durante el proyecto.

- Flujos de trabajo: se utilizarán diagramas de actividades para modelar los flujos de trabajo (*workflows*) propuesto que serán soportados por el sistema desarrollado.
- Características del producto *software*: es una lista de las características principales del producto, deseables desde una perspectiva de las necesidades del cliente.

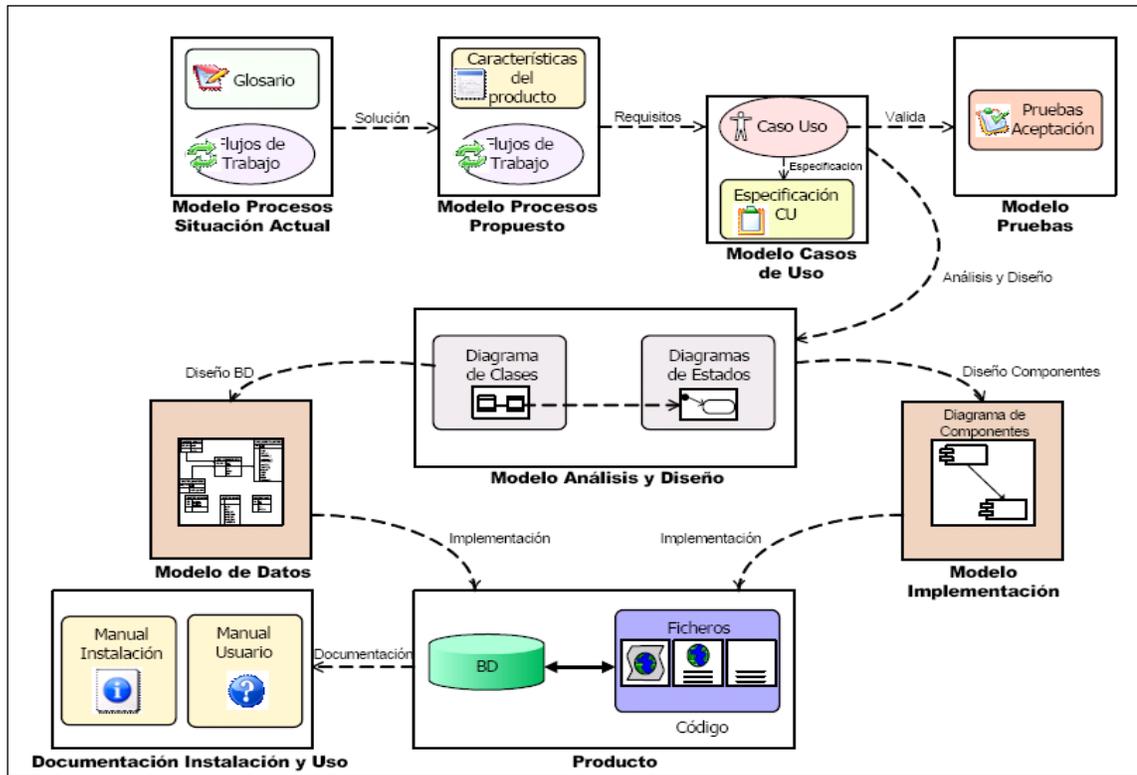
- **Glosario:** es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.
- **Modelo de casos de uso:** presenta la funcionalidad del sistema y los actores que hacen uso de ella. Es representado mediante diagramas de casos de uso.
- **Especificaciones de casos de uso:** para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados.
- **Modelo de análisis y diseño:** este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación). Está constituido esencialmente por un diagrama de clases y algunos diagramas de estados para las clases que lo requieran.
- **Modelo lógico relacional:** previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un diagrama de tablas donde se muestran las tablas, claves, etc.

- Modelo de implementación: éste es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.
- Manual de instalación: este documento incluye las instrucciones para realizar la instalación del producto.
- Material de usuario: corresponde a un conjunto de documentos y facilidades de uso del sistema.
- Producto: todos los ficheros fuente y ejecutable del producto.

1.5.2. Esquema de trazabilidad

La figura 11 ilustra las relaciones de trazabilidad entre artefactos del proyecto, y según la configuración antes mencionada.

Figura 11. Trazabilidad entre artefactos



Fuente: elaboración propia.

Las relaciones de trazabilidad son enlaces entre artefactos que establecen cómo se generan unos a partir de otros. Esto permite por ejemplo asegurar la cobertura de los requisitos o determinar el posible impacto de los cambios. En la figura 11 se ilustran los modelos y artefactos utilizados, indicando las relaciones de trazabilidad entre ellos, lo cual se resume a continuación:

- Se modelarán los procesos de negocio de la situación actual utilizando diagramas de actividad para representar flujos de trabajo actuales. Esto se complementará mediante un glosario que establecerá la terminología.

- Los requisitos serán establecidos mediante un modelo de casos de uso que incluirá diagramas de casos de uso, prototipos de interfaces de usuario y especificaciones de casos de uso.
- El modelo de análisis y diseño establecerá el particionamiento interno del sistema. Estará compuesto por un diagrama de clases y algunos diagramas de estados. Las clases determinarán la estructura y las operaciones necesarias para implementar las funcionalidades descritas en los casos de uso. Los diagramas de estados detallarán el comportamiento para las clases que lo requieran.
- A partir del diagrama de clases, y considerando las clases que requieran persistencia, se derivará el modelo lógico relacional, representado mediante diagramas de tablas.
- En el modelo de implementación se organizarán las operaciones de las clases en términos de componentes de dicha arquitectura. Esto se representará mediante diagramas de componentes.
- La implementación del modelo lógico relacional y de los componentes de la aplicación constituirá el producto, el cual se complementará con el manual de instalación y el manual de usuario.

2. REQUISITOS DEL SISTEMA

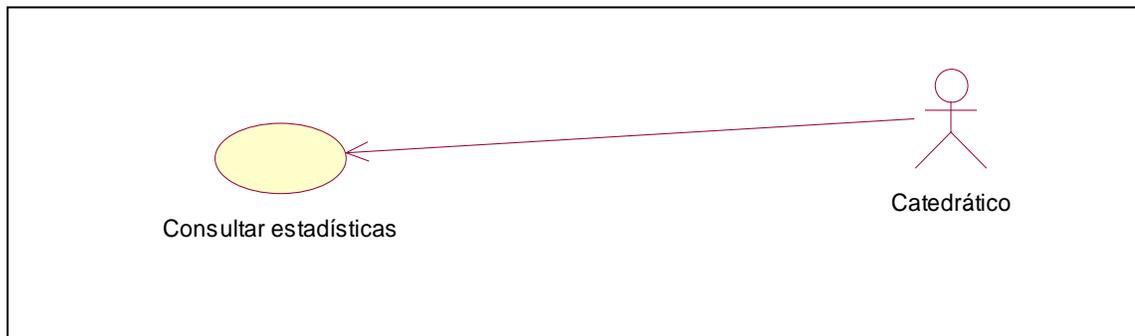
2.1. Modelo de casos de uso

Los casos de uso identificados están organizados por actor. Se ha optado por la utilización de una plantilla para la descripción de cada caso de uso.

2.1.1. Catedrático

En la figura 12 se muestra el diagrama de casos de uso asociado al catedrático.

Figura 12. Caso de uso asociado al catedrático

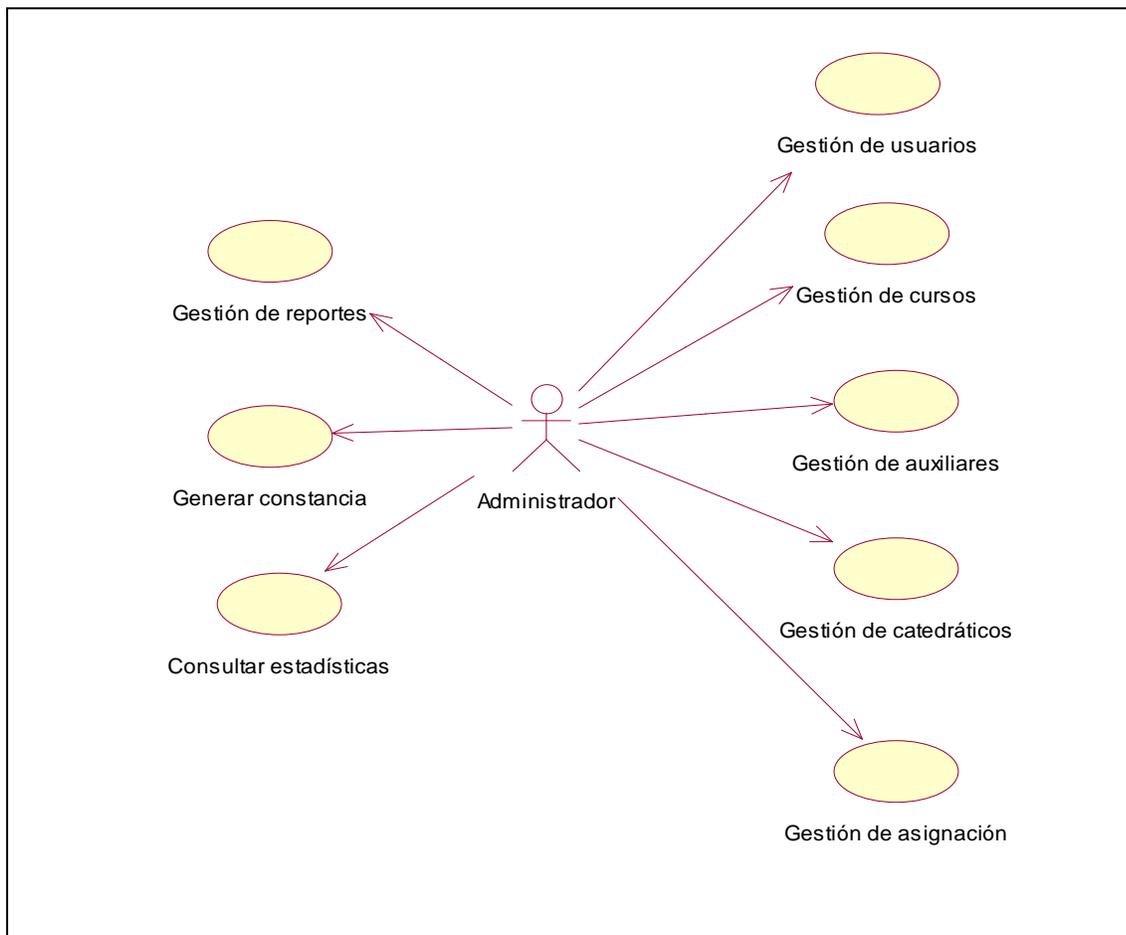


Fuente: elaboración propia.

2.1.2. Administrador

En la figura 13 se muestra el diagrama de casos de uso asociado al Administrador.

Figura 13. Casos de uso asociado al administrador

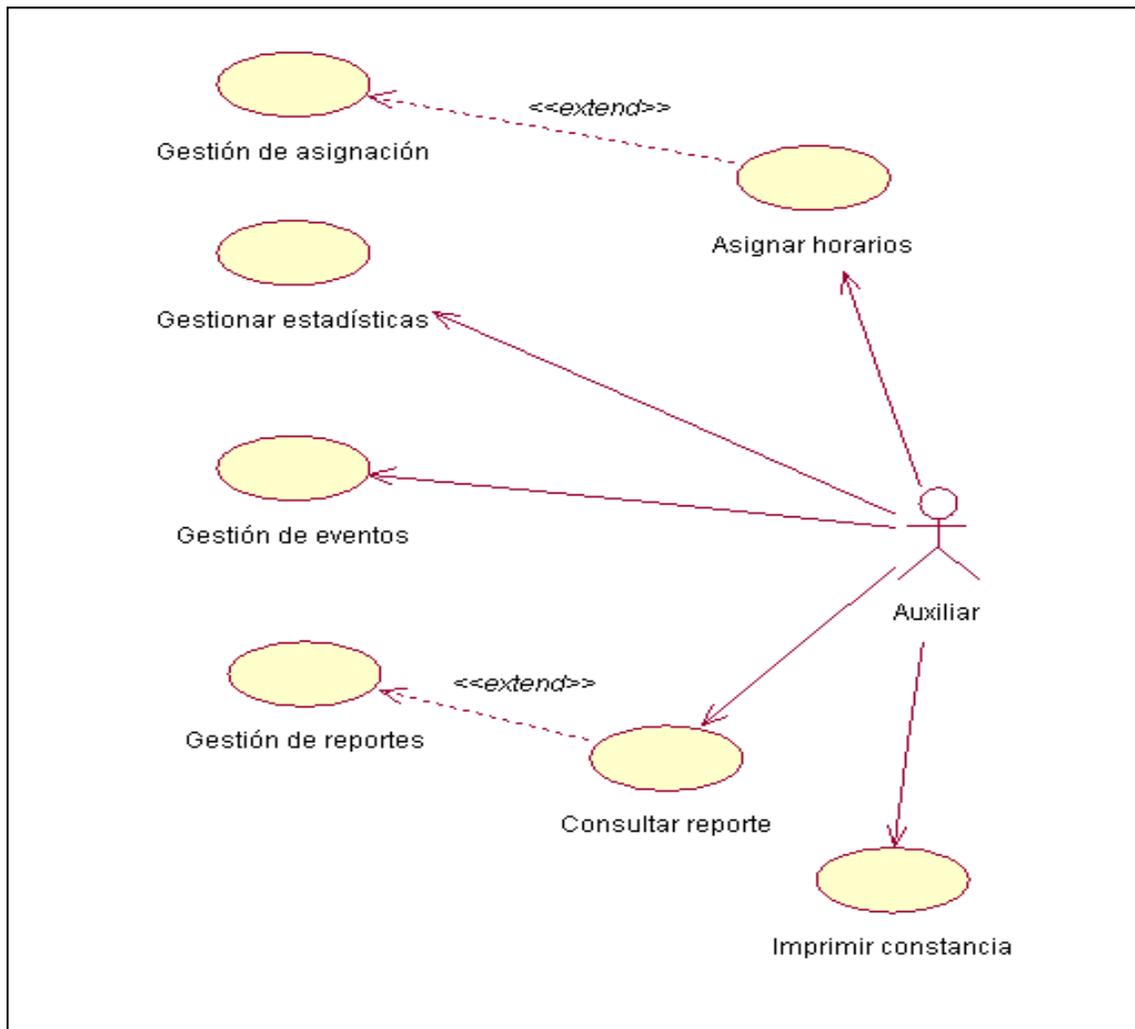


Fuente: elaboración propia.

2.1.3. Auxiliar

En la figura 14 se muestra el diagrama de casos de uso asociado al auxiliar.

Figura 14. Caso de uso asociado al auxiliar

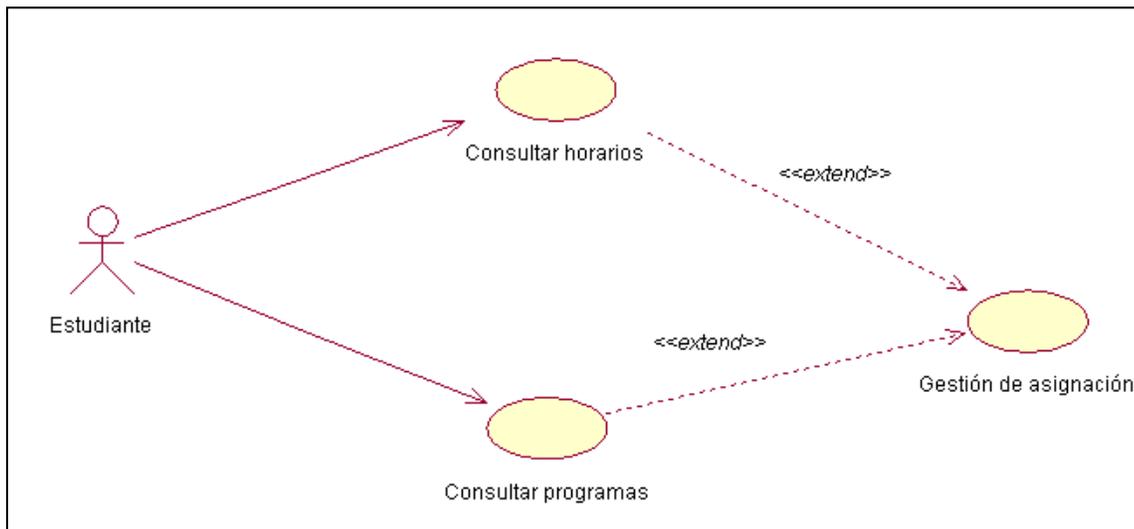


Fuente: elaboración propia.

2.1.4. Estudiante

En la figura 15 se muestra el diagrama de casos de uso asociado al estudiante.

Figura 15. Caso de uso asociado al estudiante



Fuente: elaboración propia.

2.2. Descripción de casos de uso

Tabla II. Descripción de caso de uso gestionar usuario

Nombre	Gestionar usuario
Descripción	Este caso de uso lo ejecuta el administrador del sistema. Se usa para la gestión de usuario (agregar, modificar y eliminar).
Pre-condiciones	El administrador debe de haber ingresado correctamente al sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none">1. El administrador selecciona la opción de gestión de usuarios.2. El sistema muestra las opciones de agregar, modificar y eliminar usuarios.3. El administrador selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado.	
Sub-flujos	
Caso 1: Agregar usuario <ol style="list-style-type: none">1. El sistema muestra la pantalla para agregar a un nuevo usuario.2. El administrador ingresa los datos nombre de usuario, contraseña y selecciona el tipo de usuario.3. El administrador presiona el botón agregar usuario.4. El sistema valida los datos ingresados (FA1), verifica que el usuario no exista (FA2) y agrega al usuario en el sistema.5. El sistema muestra un mensaje indicando que se ha ingresado al usuario al sistema.	

Fuente: elaboración propia.

Continuación Tabla II

<p>Caso 2: Modificar usuario</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para modificar información de usuario.2. El administrador ingresa el nombre del usuario, contraseña y tipo de usuario.3. El administrador presiona el botón modificar usuario.4. El sistema valida los datos ingresados (FA1), verifica que el usuario exista (FA3) y actualiza la información del usuario. <p>El sistema muestra un mensaje de la operación realizada.</p>
<p>Caso 3: Eliminar usuario</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar usuario.2. El administrador ingresa el nombre del usuario.3. El administrador presiona el botón eliminar usuario.4. El sistema valida los datos ingresados (FA1), valida que el usuario exista (FA3) y procede a la eliminación del usuario en el sistema.5. El sistema muestra un mensaje indicando que se ha eliminado al usuario.
<p>Flujos alternos</p>
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información. <p>FA2: Usuario existente</p> <ul style="list-style-type: none">- Si el usuario ya existe en el sistema, se muestra un mensaje indicando que el usuario ya existe, y se vuelve a pedir de nuevo la información al administrador. <p>FA3: Usuario no existe</p> <ul style="list-style-type: none">- Si el usuario no existe en el sistema, se muestra un mensaje de error indicando que el usuario no existe, y se vuelve a pedir de nuevo la información al administrador.

Fuente: elaboración propia.

Tabla III. **Descripción de caso de uso gestionar curso**

Nombre	Gestionar curso
Descripción	Este caso de uso lo ejecuta el administrador del sistema. Se usa para la gestión de cursos (agregar, modificar y eliminar).
Pre-condiciones	El administrador debe de haber ingresado correctamente al sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción de gestión de cursos. 2. El sistema muestra las opciones de agregar, modificar y eliminar cursos. 3. El administrador selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Ingresar curso</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para agregar un nuevo curso. 2. El administrador ingresa los datos código del curso, nombre del curso y selecciona el semestre al que pertenece el curso. 3. El administrador presiona el botón agregar nuevo curso. 4. El sistema valida los datos ingresados (FA1), verifica que el código del curso no exista (FA2) y procede a agregar el curso en el sistema. 5. El sistema muestra un mensaje indicando que se ha agregado un nuevo curso. 	

Fuente: elaboración propia.

Continuación Tabla III

Caso 2: Modificar curso

1. El sistema muestra la pantalla para modificar información del curso.
2. El administrador ingresa el código del curso, y presiona el botón aceptar.
3. El sistema valida el dato ingresado (FA1) y verifica que el código del curso exista (FA3).
4. El sistema devuelve los datos del curso y permite el cambio de los datos nombre del curso y semestre al que pertenece el curso.
5. El administrador presiona el botón modificar datos del curso.
6. El sistema valida los datos ingresados (FA1) y verifica que el código del curso exista (FA3) y procede a la actualización de la información del curso.
7. El sistema muestra un mensaje indicando que se ha actualizado la información del curso.

Caso 3: Eliminar curso

1. El sistema muestra la pantalla para eliminar curso.
2. El administrador ingresa el código del curso y presiona el botón aceptar.
3. El sistema valida el dato ingresado (FA1), y verifica que el código del curso exista (FA3).
4. El sistema devuelve la información del curso que se desea eliminar.
5. El administrador presiona el botón eliminar curso.
6. El sistema valida el dato ingresado (FA1), valida que el código del curso exista (FA3) y procede a la eliminación del curso en el sistema.
7. El sistema muestra un mensaje indicando que se ha eliminado el curso.

Fuente: elaboración propia.

Continuación Tabla III

Flujos alternos
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información.
<p>FA2: Código del curso existente</p> <ul style="list-style-type: none">- Si el código del curso ya existe en el sistema, se muestra un mensaje indicando que el código del curso ya existe y por lo tanto no se puede ingresar, y se vuelve a pedir de nuevo la información al administrador.
<p>FA3: Código del curso no existe</p> <ul style="list-style-type: none">- Si el código del curso no existe en el sistema, se muestra un mensaje de error indicando que el código del curso no existe, y se vuelve a pedir de nuevo la información al administrador

Fuente: elaboración propia.

Tabla IV. **Descripción de caso de uso gestionar auxiliar**

Nombre	Gestionar auxiliar
Descripción	Este caso de uso se encarga de agregar, modificar y eliminar información de auxiliares.
Pre-condiciones	El administrador debe de haber ingresado correctamente al sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción de gestión de auxiliares. 2. El sistema muestra las opciones de agregar, modificar y eliminar auxiliar. 3. El administrador selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Ingresar auxiliar</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para agregar a un nuevo auxiliar. 2. El administrador ingresa los datos número de carnet, nombre, dirección, teléfono, celular, <i>e-mail</i> y fecha de ingreso. 3. El administrador presiona el botón agregar auxiliar. 4. El sistema valida los datos ingresados (FA1), verifica que el número de carnet no exista (FA2) y procede a agregar al auxiliar al sistema. 5. El sistema muestra un mensaje indicando que se ha agregado a auxiliar. 	

Fuente: elaboración propia.

Continuación Tabla IV

Caso 2: Modificar auxiliar

1. El sistema muestra la pantalla para modificar información del auxiliar.
2. El administrador ingresa el número de carnet del auxiliar y presiona el botón aceptar.
3. El sistema valida el dato ingresado (FA1) y verifica que el número de carnet del auxiliar exista (FA3).
4. El sistema devuelve los datos del auxiliar y permite el cambio de los datos nombre, dirección, teléfono, celular, *e-mail* y fecha de ingreso.
5. El administrador presiona el botón modificar datos de auxiliar.
6. El sistema valida los datos ingresados (FA1), verifica que el número de carnet exista (FA3) y procede a la actualización de la información del auxiliar.
7. El sistema muestra un mensaje indicando que se ha actualizado la información del auxiliar.

Caso 3: Eliminar auxiliar

1. El sistema muestra la pantalla para eliminar auxiliar.
2. El administrador ingresa el número de carnet del auxiliar y presiona el botón aceptar.
3. El sistema valida el dato ingresado (FA1), y verifica que el número de carnet exista (FA3)
4. El sistema devuelve la información del auxiliar que se desea eliminar.
5. El administrador presiona el botón eliminar auxiliar.
6. El sistema valida el dato ingresado (FA1), valida que el número de carnet exista (FA3) y elimina la información en el sistema.
7. El sistema muestra un mensaje indicando que se ha eliminado al auxiliar.

Fuente: elaboración propia.

Continuación Tabla IV

Flujos alternos
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información.
<p>FA2: Número de carnet existente</p> <ul style="list-style-type: none">- Si el número de carnet ya existe en el sistema, se muestra un mensaje indicando que el número de carnet ya existe y por lo tanto no se puede ingresar, y se vuelve a pedir de nuevo la información al administrador.
<p>FA3: Número de carnet no existe</p> <ul style="list-style-type: none">- Si el número de carnet no existe en el sistema, se muestra un mensaje de error indicando que el número de carnet no existe, y se vuelve a pedir de nuevo la información al administrador.

Fuente: elaboración propia.

Tabla V. **Descripción de caso de uso gestionar catedrático**

Nombre	Gestionar catedrático
Descripción	Este caso de uso se encarga de agregar, modificar y eliminar información de catedráticos.
Pre-condiciones	El administrador debe de haber ingresado correctamente al sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción de gestión de catedráticos. 2. El sistema muestra las opciones de agregar, modificar y eliminar catedrático. 3. El administrador selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Ingresar catedrático</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para agregar a un nuevo catedrático. 2. El administrador ingresa los datos número de registro de personal, nombre, dirección, teléfono, celular, <i>e-mail</i> y fecha de ingreso. 3. El administrador presiona el botón agregar catedrático. 4. El sistema valida los datos ingresados (FA1), verifica que el número de registro de personal no exista (FA2) y procede a agregar al catedrático al sistema. 5. El sistema muestra un mensaje indicando que se ha agregado a catedrático. 	

Fuente: elaboración propia.

Continuación Tabla V

<p>Caso 2: Modificar catedrático</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para modificar información del catedrático.2. El administrador ingresa el número de registro de personal, y presiona el botón aceptar.3. El sistema valida el dato ingresado (FA1) y verifica que el número de registro de personal exista (FA3).4. El sistema devuelve los datos del catedrático y permite el cambio de los datos nombre, dirección, teléfono, celular, <i>e-mail</i> y fecha de ingreso.5. El administrador presiona el botón modificar datos de auxiliar.6. El sistema valida los datos ingresados (FA1), verifica que el número de registro de personal exista (FA3) y procede a la actualización de la información del catedrático.7. El sistema muestra un mensaje indicando que se ha actualizado la información del catedrático.
<p>Caso 3: Eliminar catedrático</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar catedrático.2. El administrador ingresa el número de registro de personal del catedrático y presiona el botón aceptar.3. El sistema valida el dato ingresado (FA1), y verifica que el número de registro de personal exista (FA3)4. El sistema devuelve la información del catedrático que se desea eliminar.5. El administrador presiona el botón eliminar catedrático.6. El sistema valida el dato ingresado (FA1), valida que el número de registro de personal exista (FA3) y elimina la información en el sistema.7. El sistema muestra un mensaje indicando que se ha eliminado al catedrático.

Fuente: elaboración propia.

Continuación Tabla V

Flujos alternos
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información.
<p>FA2: Número de registro de personal existente</p> <ul style="list-style-type: none">- Si el número de registro de personal ya existe en el sistema, se muestra un mensaje indicando que el número de registro de personal ya existe y por lo tanto no se puede ingresar, y se vuelve a pedir de nuevo la información al administrador.
<p>FA3: Número de registro de personal no existe</p> <ul style="list-style-type: none">- Si el número de registro de personal no existe en el sistema, se muestra un mensaje de error indicando que el número de registro de personal no existe, y se vuelve a pedir de nuevo la información al administrador.

Fuente: elaboración propia.

Tabla VI. **Descripción de caso de uso gestionar asignación**

Nombre	Gestionar asignación
Descripción	Este caso se encarga de la asignación de catedráticos, auxiliares, horarios de los diferentes cursos.
Pre-condiciones	<ul style="list-style-type: none"> - El administrador y el auxiliar deben de haber ingresado correctamente al sistema. - Debe de haberse ingresado cursos, catedráticos y auxiliares (haber usado los casos de uso correspondiente)
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción de asignación. 2. El sistema muestra las opciones dependiendo del usuario, si es administrador se visualiza las opciones asignación de cursos, asignación de auxiliares, modificación y eliminación de asignación de curso, si es auxiliar se mostrará las opciones asignación de horarios (curso, laboratorio, atención al estudiante). 3. El usuario selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Asignación de catedrático</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para la asignación de catedrático. 2. El administrador ingresa el registro de personal (FA2). 3. El administrador selecciona el curso, sección, titularidad. 4. El sistema valida los datos ingresados (FA1) y procede a la asignación. 	

Fuente: elaboración propia.

Continuación Tabla VI

<p>Caso 2: Asignación de auxiliar</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para la asignación de auxiliar.2. El administrador ingresa los datos carnet del auxiliar.3. El administrador selecciona el curso, sección, titularidad.4. El administrador presiona el botón guardar asignación.5. El sistema valida los datos ingresados (FA1) y procede a la asignación del curso.6. El sistema muestra un mensaje indicando que se ha asignado el curso al auxiliar.
<p>Caso 3: Asignación de horarios</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para la asignación de horarios2. El sistema muestra los cursos asignados al auxiliar.3. El auxiliar selecciona el curso al cual desea asignar horarios (curso, laboratorio, atención)4. El auxiliar selecciona el horario (curso, laboratorio, atención), día, edificio e ingresa el salón y el horario de inicio y finalización.5. El sistema valida los datos ingresado (FA1) y verifica que el carnet del auxiliar exista (FA3).6. El sistema muestra un mensaje indicando que se ha asignado horario.
<p>Caso 4: Modificar asignación de catedrático</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para la modificación de asignación de catedrático.2. El sistema muestra la información de la asignación del curso, y procede a modificar los datos necesarios, y presiona el botón modificar asignación.3. El sistema valida los datos ingresado (FA1), y procede a la a la actualización de la asignación de curso.4. El sistema muestra un mensaje de actualización de asignación.

Fuente: elaboración propia.

Continuación Tabla VI

<p>Caso 5: Modificar asignación de auxiliar</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para la modificación de asignación de auxiliar.2. El sistema muestra la información de la asignación del auxiliar, y procede a modificar los datos necesarios, y presiona el botón modificar asignación.3. El sistema valida los datos ingresado (FA1), y procede a la a la actualización de la asignación.4. El sistema muestra un mensaje indicando que se ha actualizado la información.
<p>Caso 6: Modificar asignación de horario</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para la modificación de asignación de horario.2. El sistema muestra la información de la asignación de horario, y procede a modificar los datos necesarios, y presiona el botón modificar asignación de horario.3. El sistema valida los datos ingresado (FA1), y procede a la a la actualización de la asignación de horario.4. El sistema muestra un mensaje indicando que se ha actualizado la asignación.
<p>Caso 7: Eliminar asignación de catedrático</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar asignación de catedrático.2. El administrador ingresa el número de registro de personal y presiona el botón aceptar.3. El sistema devuelve la información de las asignaciones del catedrático que se desea eliminar.4. El administrador selecciona las asignaciones a eliminar y presiona el botón eliminar asignación.5. El sistema valida que se pueda eliminar la información (FA4) y procede a eliminarla del sistema.6. El sistema muestra un mensaje indicando que se ha eliminado la asignación.

Fuente: elaboración propia.

Continuación Tabla VI

<p>Caso 8: Eliminar asignación de auxiliar</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar asignación de auxiliar.2. El administrador ingresa número de carnet y presiona el botón aceptar.3. El sistema devuelve la información de la asignación del auxiliar que se desea eliminar.4. El administrador selecciona las asignaciones a eliminar y presiona el botón eliminar asignación.5. El sistema valida que se pueda eliminar la información (FA4) y procede a eliminarla del sistema.6. El sistema muestra un mensaje indicando que se ha eliminado la asignación.
<p>Caso 9: Eliminar Asignación de Laboratorio</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar asignación de horarios.2. El administrador ingresa carnet del auxiliar y presiona el botón aceptar.3. El sistema devuelve la información de la asignación de horarios que desea eliminar.4. El administrador presiona el botón eliminar horarios.5. El sistema valida que se pueda eliminar la información (FA4) y procede a eliminarla del sistema.
<p>Flujos alternos</p>
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información. <p>FA2: Número de registro de personal no existente</p> <ul style="list-style-type: none">- Si el número de registro de personal no existe en el sistema, se muestra un mensaje indicando que el número de registro de personal no existe y por lo tanto no se vuelve a pedir que ingrese de nuevo el registro de personal.

Fuente: elaboración propia.

Continuación Tabla VI

FA3: Número de carnet no existe

- Si el número de carnet no existe en el sistema, se muestra un mensaje de error indicando que el carnet no existe, y regresa a la lista de cursos asignados al auxiliar.

FA4: Eliminación de asignación no es posible

- Si la información que se desea eliminar se encuentra relacionado con otra información se muestra un mensaje indicando que la información no puede ser eliminada, debido a dependencias que existe.

Fuente: elaboración propia.

Tabla VII. **Descripción de caso de uso gestionar estadísticas**

Nombre	Gestionar estadísticas
Descripción	Este caso de uso se encarga de ingresar, modificar, eliminar y visualizar estadísticas de las actividades de los cursos.
Pre-condiciones	<ul style="list-style-type: none"> - El administrador, auxiliar y catedrático deben de haber ingresado correctamente al sistema. - Debe de haberse ingresado cursos, catedráticos y auxiliares (haber usado los casos de uso correspondiente)
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción de gestión de estadísticas. 2. El sistema muestra las opciones dependiendo del usuario, si es administrador se visualiza las opciones de ingresar, modificar, eliminar y ver estadísticas del cursos, si es auxiliar se mostrará las opciones ingresar, modificar y eliminar estadísticas, si es catedrático la opción ver estadísticas. 3. El usuario selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Ingresar estadísticas</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para ingresar estadísticas. 2. El usuario ingresa los estadísticos. 3. El usuario presiona el botón agregar estadística. 4. El sistema valida los datos ingresados (FA1). 5. El sistema muestra un mensaje que se ha ingresado las estadísticas. 	

Fuente: elaboración propia.

Continuación Tabla VII

<p>Caso 2: Modificar estadísticas</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para modificar estadísticas.2. El usuario selecciona el curso, sección y actividad al cual desea modificar las estadísticas, y presiona el botón actualizar información.3. El sistema valida los datos ingresado (FA1), y procede a la actualización de las estadísticas. <p>El sistema muestra un mensaje indicando que se ha actualizado las estadísticas.</p>
<p>Caso 3: Eliminar estadísticas</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar estadísticas.2. El usuario selecciona el curso, sección y la actividad al cual desea eliminar las estadísticas y presiona el botón eliminar estadísticas.3. El sistema procede a la eliminación de las estadísticas y muestra un mensaje de que se ha eliminado la estadística seleccionada.
<p>Caso 4. Ver estadísticas del curso</p> <ol style="list-style-type: none">1. El sistema muestra en pantalla el listado de cursos y secciones y un enlace para visualizar estadísticas del curso y sección.2. El usuario presiona el enlace del curso y sección para ver estadísticas.3. El sistema muestra las estadísticas del curso y sección.
<p>Flujos alternos</p>
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al administrador de nuevo la información. <p>FA4: Eliminación de asignación no es posible</p> <ul style="list-style-type: none">- Si la información que se desea eliminar se encuentra relacionado con otra información se muestra un mensaje de que no se puede eliminar.

Fuente: elaboración propia.

Tabla VIII. Descripción de caso de uso gestionar reportes

Nombre	Gestionar reportes
Descripción	Este caso de uso se encarga del ingresar, modificación y eliminación de los reportes presentados por los auxiliares.
Pre-condiciones	- El administrador, auxiliar y catedrático deben de haber ingresado correctamente al sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción de gestión de reportes. 2. El sistema muestra las opciones dependiendo del usuario, si es administrador se visualiza las opciones de ingresar, modificar, eliminar reportes, si es auxiliar se mostrará las opciones ingresar, modificar y eliminar reportes y si el usuario es catedrático se mostrará las opción reporte de actividades. 3. El usuario selecciona una de las opciones, dependiendo de la opción seleccionada, uno de los siguientes sub-flujos es ejecutado. 	
Sub-flujos	
<p>Caso 1: Ingresar reporte</p> <ol style="list-style-type: none"> 1. El sistema muestra la pantalla para el ingreso de reportes. 2. El auxiliar selecciona el curso e ingresa la descripción de actividades, anomalías y dudas y presiona el botón ingresar reporte. 3. El sistema valida los datos ingresados (FA1), y procede a ingresar el reporte. 4. El sistema muestra un mensaje indicando que se ha ingresado el reporte. 	

Fuente: elaboración propia.

Continuación Tabla VIII

<p>Caso 2: Modificar reporte</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para modificar reportes.2. El usuario selecciona el reporte que desea modificar y procede a modificar la información necesaria y presiona el botón actualizar reporte.3. El sistema valida los datos ingresado (FA1), y procede a la actualización del reporte presentado.4. El sistema muestra un mensaje indicando que se ha actualizado la información.
<p>Caso 3: Eliminar Reportes</p> <ol style="list-style-type: none">1. El sistema muestra la pantalla para eliminar reportes.2. El administrador ingresa el carnet del auxiliar y selecciona el reporte que desea eliminar y presiona el botón eliminar reporte.3. El sistema procede a la eliminación del reporte y muestra un mensaje de que se ha eliminado el reporte.
<p>Caso 4: Ver reporte</p> <ol style="list-style-type: none">1. El sistema muestra el listado de reportes ingresados por el auxiliar.2. El usuario selecciona el reporte que desea visualizar.3. El sistema muestra el reporte de las actividades realizadas, estadísticas de las actividades, estadísticas de deserción.
<p>Flujos alternos</p>
<p>FA1: Datos no válidos</p> <ul style="list-style-type: none">- Falta llenar información indispensable o una de las entradas es incorrecta, se le vuelve a pedir al usuario de nuevo la información.

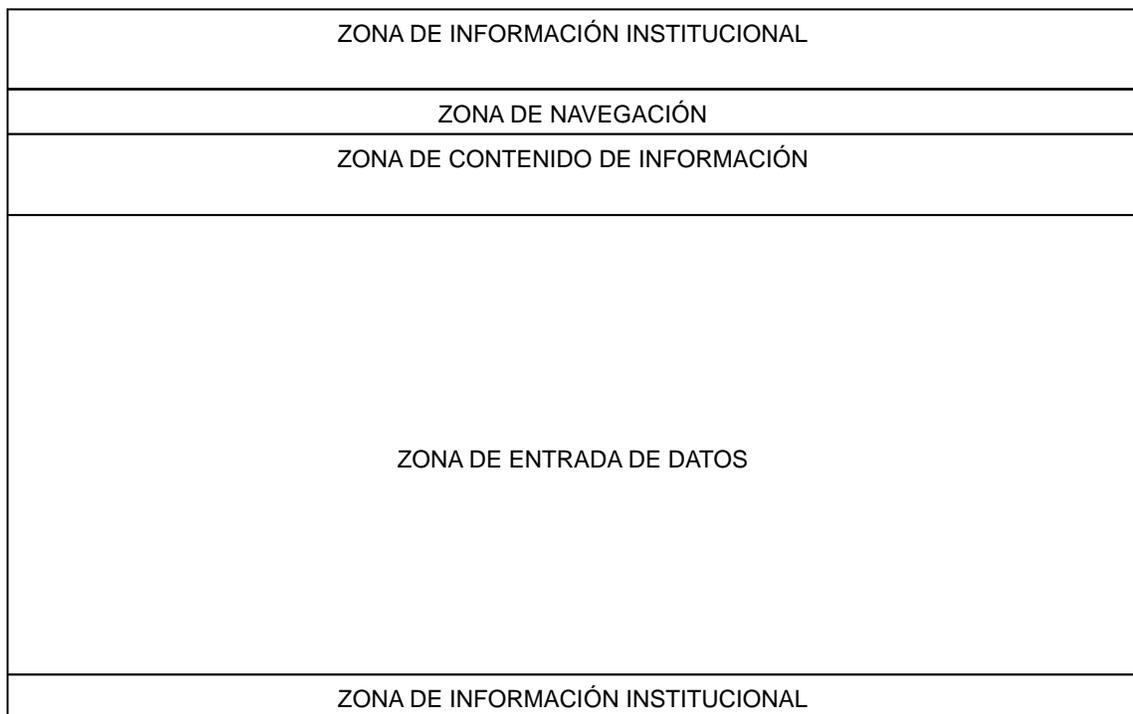
Fuente: elaboración propia.

2.3. Interfaces web

2.3.1. Interfaces de contenido de información

Las páginas de información son las responsables de proporcionar al usuario información y funcionalidad, el objetivo de estas páginas es informar al usuario de parte del sistema.

Figura 16. **Esquema de las interfaces de contenido de información**

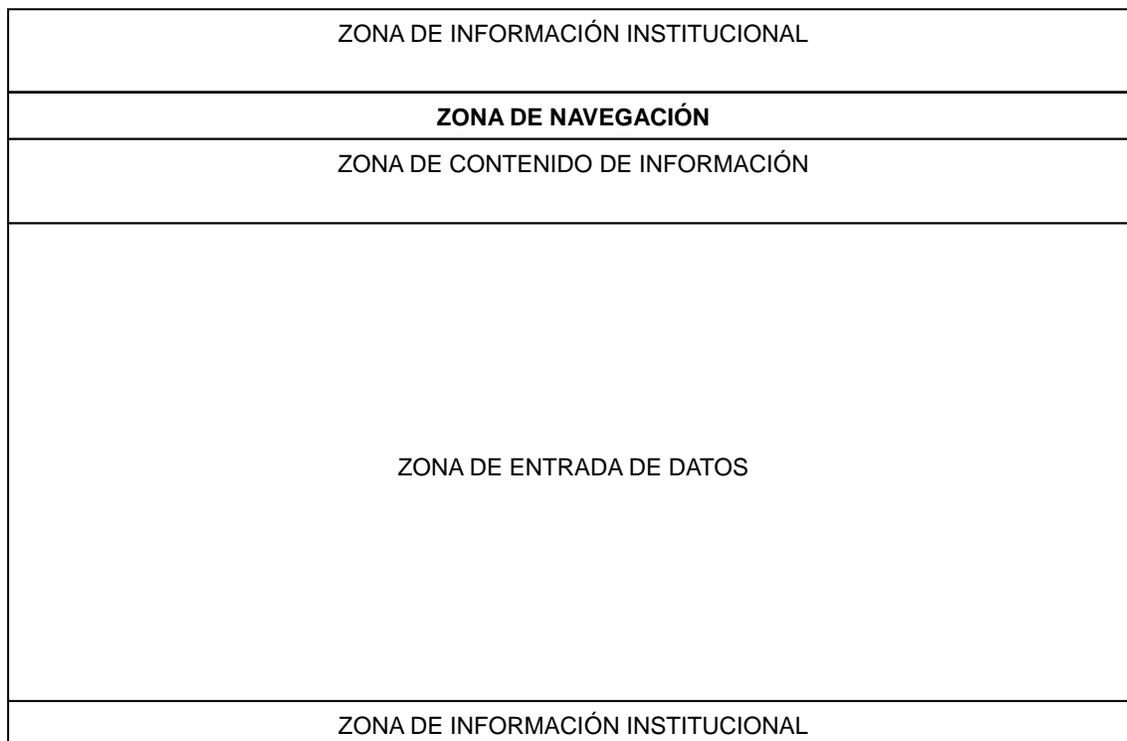


Fuente: elaboración propia.

2.3.2. Interfaces de estructura de navegación

Son páginas especializadas en estructurar la navegabilidad dentro de la aplicación, dando acceso a páginas que sólo son accesibles a partir de éstas.

Figura 17. **Esquema de las interfaces de estructura de navegación**

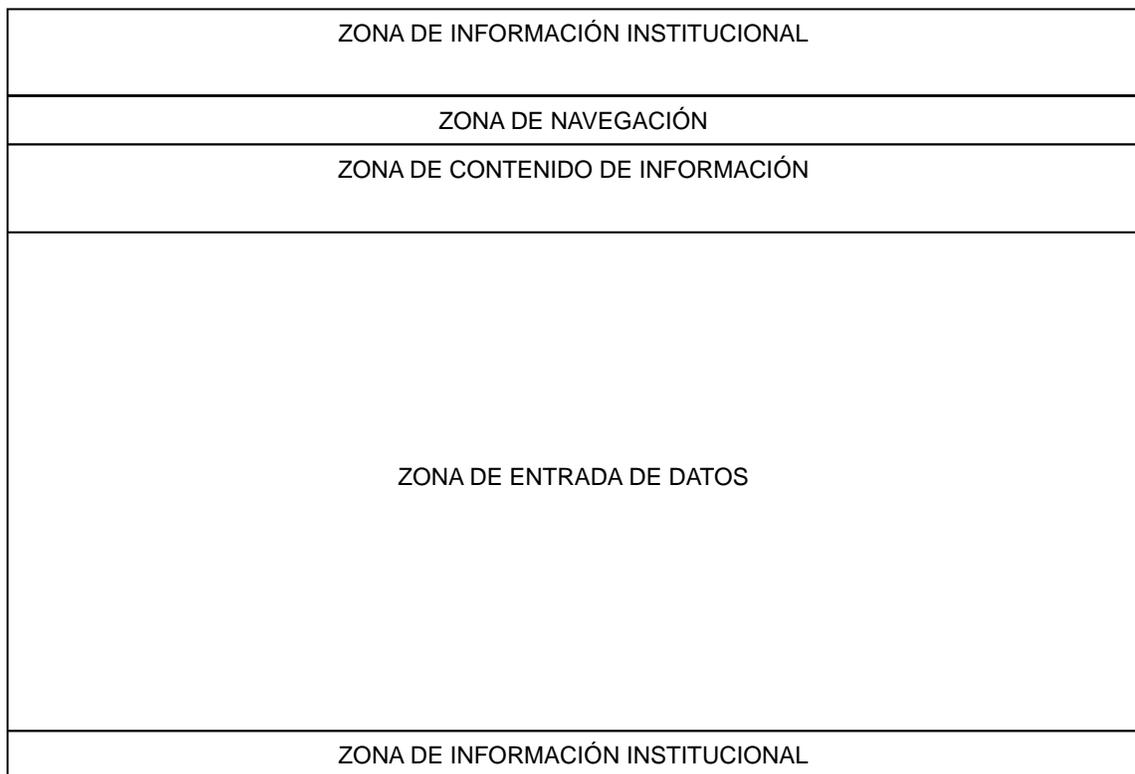


Fuente: elaboración propia.

2.3.3. Interfaces de entrada de datos

Este tipo de interfaz surge cuando el usuario necesita introducir datos en páginas para llevar a cabo cierta funcionalidad. En este caso para el ingreso de información de auxiliares, catedráticos, actividades y estadísticas.

Figura 18. **Esquema de las interfaces de entrada de datos**



Fuente: elaboración propia.

A continuación en la figura 19 encontramos los esquemas anteriormente descritos aplicados a una interfaz real.

Figura 19. **Interfaces del sitio web**



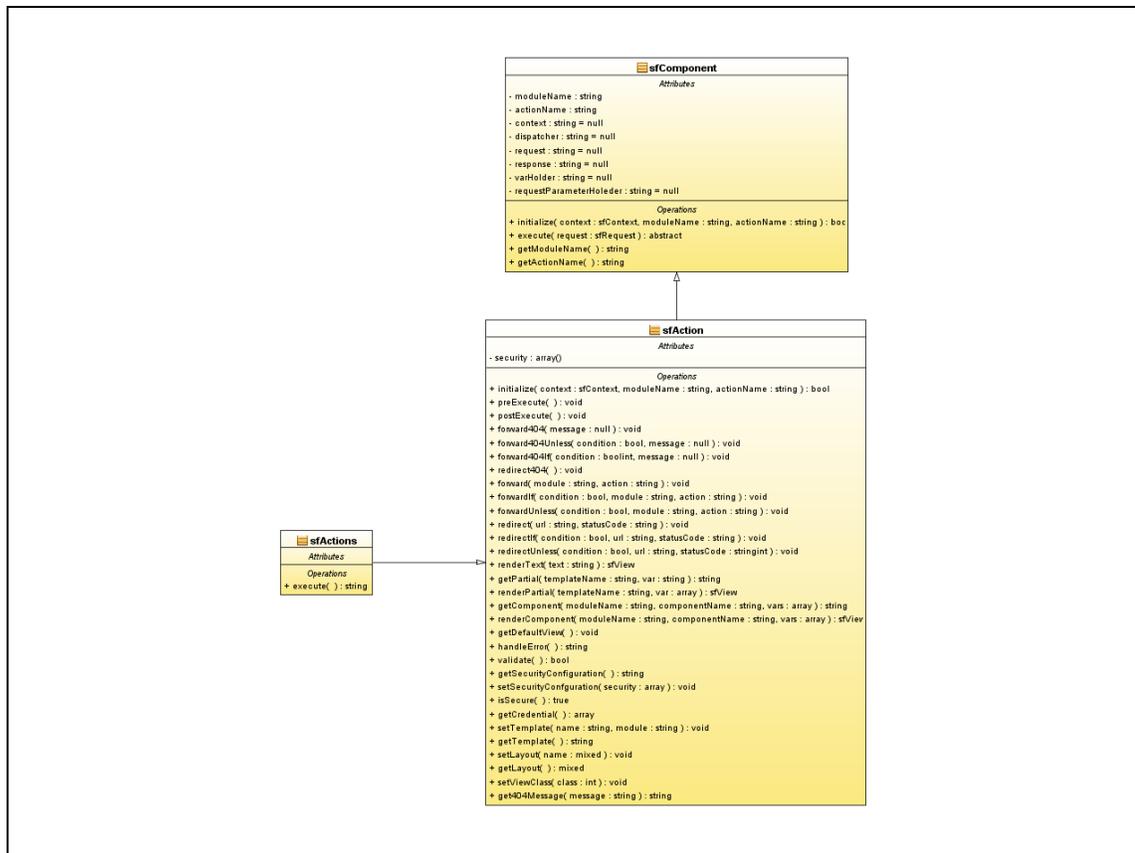
Fuente: elaboración propia.

3. ANÁLISIS DEL SISTEMA

3.1. Diagrama de clases

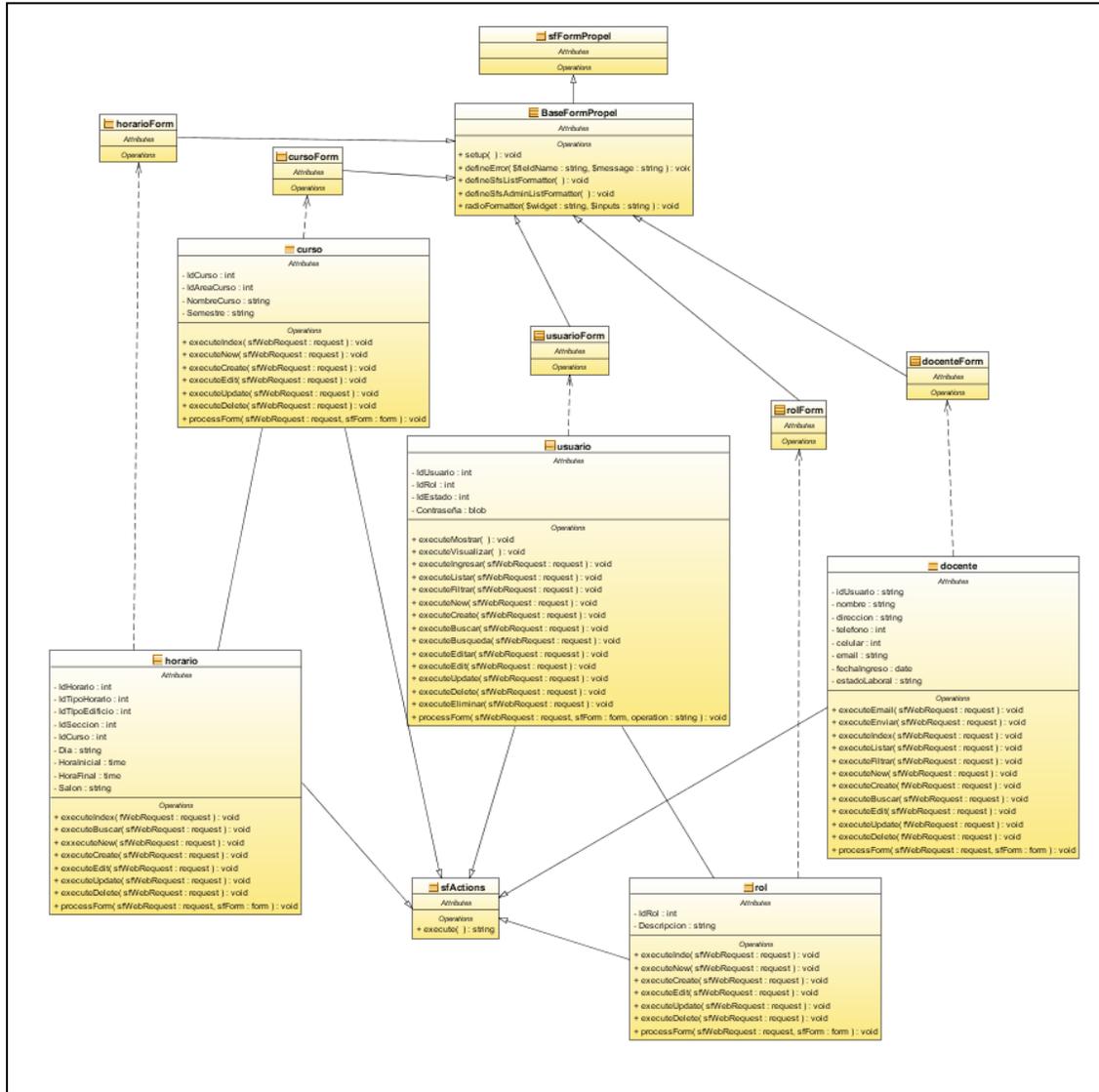
A continuación se muestra el diagrama de clases que representa la estructura del sistema.

Figura 20. Diagrama de clases *Framework Symfony*



Fuente: elaboración propia.

Figura 21. Diagrama de clases de estructura de la aplicación



Fuente: elaboración propia.

3.2. Trazabilidad de casos de uso a clases

En esta sección se ilustra la trazabilidad entre los casos de uso y las clases del diagrama de la figura 21.

Tabla IX. Trazabilidad de casos de uso del diagrama de clases

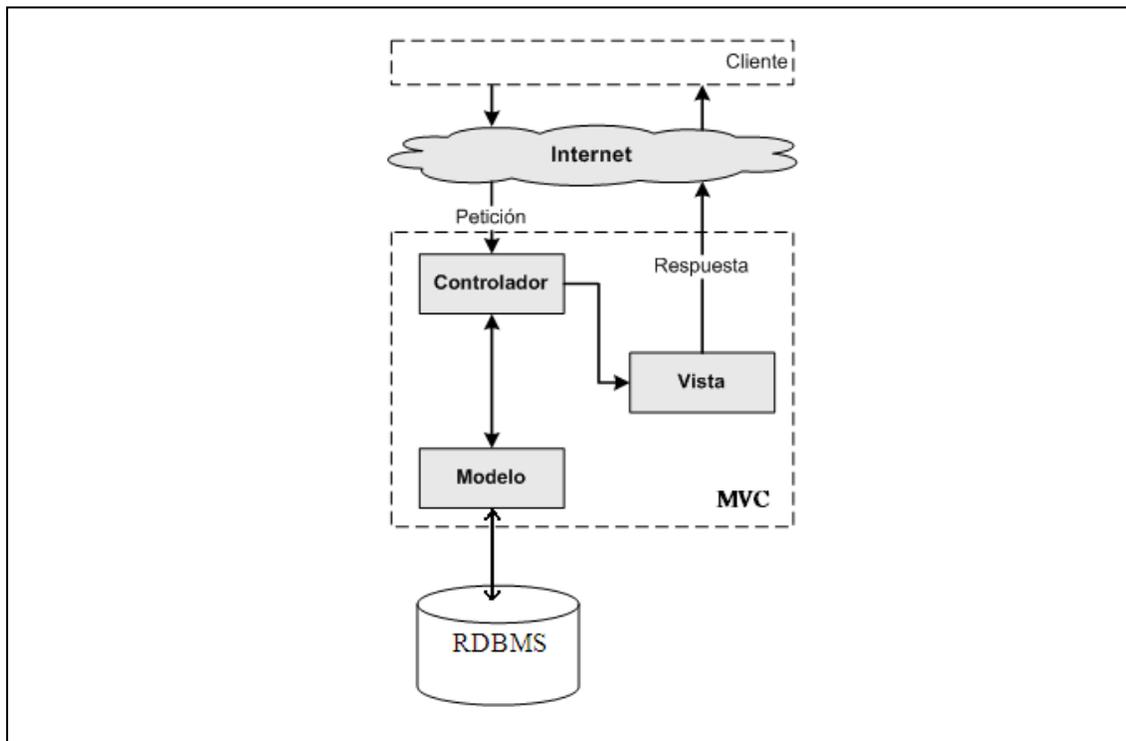
Clase Casos de uso	Usuario	Curso	Docente	Horario	Estadísticas	Reportes	Asignación
Gestionar usuarios	X						
Gestionar curso		X					
Gestionar auxiliar			X				
Gestionar catedrático			X				
Consultar horarios				X			
Asignar horarios				X			
Gestionar estadísticas					X		
Consultar estadísticas					X		
Gestionar reportes						X	
Consultar reportes						X	
Generar constancia						X	
Imprimir constancia						X	
Gestionar asignación							X

Fuente: elaboración propia.

4.2. Arquitectura de la aplicación

La arquitectura de aplicación estaba basada en el modelo *MVC*, el cual separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (*HTTP*, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. En la siguiente figura se ilustra el funcionamiento de este patrón arquitectónico.

Figura 23. **Arquitectura de la aplicación MVC**



Fuente: elaboración propia.

4.3. Arquitectura del código fuente

La aplicación está formada por uno o más módulos. Un módulo normalmente representa a una página *web* o a un grupo de páginas con un propósito relacionado. Los módulos almacenan las acciones, que representan cada una de las operaciones que se puede realizar en un módulo.

4.3.1. Estructura de la aplicación

Tabla X. Estructura de la aplicación

Directorio	Descripción
apps/	Contiene un directorio por cada aplicación del proyecto.
cache/	Contiene la versión cacheada de la configuración y (si está activada) la versión cacheada de las acciones y plantillas del proyecto.
config/	Almacena la configuración general del proyecto
data/	En este directorio se almacenan los archivos relacionados con los datos, es en donde se almacena el esquema la base de datos.
lib/	Almacena las clases y librerías externas. Se guardan todo el código común a todas las aplicaciones del proyecto. El subdirectorio <i>model/</i> guarda el modelo de objetos del proyecto.
log/	Guarda todos los archivos de log generados por <i>Symfony</i> . También se utiliza para guardar los <i>logs</i> del servidor <i>web</i> , de la base de datos o de cualquier otro componente del proyecto.
plugins/	Almacena los <i>plugins</i> instalados en la aplicación.
web/	La raíz del servidor <i>web</i> . Los únicos archivos accesibles desde Internet son los que se encuentran en este directorio.

Fuente: elaboración propia.

4.3.2. Estructura de módulos

Tabla XI. Estructura de módulos

Directorio	Descripción
config/	Contiene un conjunto de archivos de configuración creados con YAML. Aquí se almacena la mayor parte de la configuración de la aplicación, salvo los parámetros propios del <i>Framework</i> . También es posible redefinir en este directorio los parámetros por defecto si es necesario.
i18n/	Contiene todos los archivos utilizados para la internacionalización de la aplicación, sobre todo los archivos que traducen la interfaz.
lib/	Contiene las clases y librerías utilizadas exclusivamente por la aplicación
modules/	Almacena los módulos que definen las características de la aplicación
templates/	Contiene las plantillas globales de la aplicación, es decir, las que utilizan todos los módulos. contiene un archivo llamado layout.php, que es el <i>layout</i> principal con el que se muestran las plantillas de los módulos

Fuente: elaboración propia.

4.3.3. Estructura del sitio web

Tabla XII. Estructura del sitio web

Directorio	Descripción
css/	Contiene los archivos de hojas de estilo creados con CSS (archivos con extensión .css)
images/	Contiene las imágenes del sitio con formato .jpg, .png o .gif
js/	Contiene los archivos de <i>JavaScript</i> con extensión .js
uploads/	Se almacenan los archivos subidos por los usuarios.

Fuente: elaboración propia.

5. HERRAMIENTAS UTILIZADAS

En esta sección se describen las herramientas utilizadas para el desarrollo de la aplicación.

- SYMFONY: *framework* para trabajar con *PHP* que sigue la mayoría de mejores prácticas y patrones de diseño de la *web*. La versión utilizada es la 1.2
- MySQL: sistema de gestión de base de datos relacional.
- PROPEL: capa que traduce la lógica objeto a la lógica relaciona llamada *ORM (object relational mapping)*. PROPEL utiliza un generador, este generador lee los archivos *XML* y construye las clases *PHP* que interactúan con el modelo de datos y el archivo de definiciones *SQL*, con el cual se pueden crear las tablas, las relaciones, las llaves, etc.
- FusionCharts: librería que permite crea gráficas *flash* animadas e iterativas las cuales puede ser incrustada dentro de diferentes lenguajes en nuestro caso *PHP*.

CONCLUSIONES

1. El sistema *web* hará más eficiente la recepción de reportes y estadísticas presentadas por los auxiliares.
2. La utilización de una metodología en el contexto de implementación permitió que se cumpliera con los objetivos propuestos.
3. El uso del *Framework Symfony* permite que nuevos módulos sean integrados al sistema sin las dificultades que se dan al no tener separado el código fuente en varias capas.

RECOMENDACIONES

1. Conocer el *Framework Symfony* el cual permitirá desarrollar y optimizar aplicaciones *web* complejas, separando la lógica del negocio, la lógica de servidor y la presentación.
2. Hacer uso de herramientas *OpenSource* para el desarrollo de futuras aplicaciones sociales y educativas.
3. Dar seguimiento al sistema para poder mejorarlo o adaptarle nuevos módulos.

BIBLIOGRAFÍA

1. GRADY BOOCH, Ivar Jacobson. *The Unified Modeling Language User Guide*. 1a. edición, Estados Unidos: Addison-Wesley, 1998. p. 14-33.
2. GRADY BOOCH, Ivar Jacobson. *The Unified Software Development Process*, 1a. edición, Estados Unidos: Addison-Wesley, 1999. p. 217-250.
3. KRUCHTEN, Philippe. *Architectural Blueprints. The "4+1" View Model of Software Architecture*, Rational Software Corp. IEEE Software 12 (6), 1995. p. 42-50.
4. KRUCHTEN, Philippe. *The Rational Unified Process: an introduction*. 2a. edición, Estados Unidos: Addison-Wesley, 2000. p. 17-32.
5. SYMFONY, *Open-Source PHP Web Framework* [en línea]. The SensioLabs Network. [Consulta: 30 de Julio de 2010]. Disponible en Web: <<http://www.symfony-project.org/>>
6. UNIVERSIDAD POLITÉCNICA DE VALENCIA. *Diseño y Construcción de un sistema para la monitorización de campos electromagnéticos* [en línea]. Departamento de Sistemas Informáticos y Computación. [Consulta: 30 de julio de 2010]. Disponible en Web:

<<https://pid.dsic.upv.es/C1/Material/default.aspx>>

7. WIKIPEDIA. *Unified Process* [en línea]. Software development process [Consulta: 30 Octubre de 2010]. Disponible en Web: <http://en.wikipedia.org/wiki/Unified_Process>