

# Universidad de las Ciencias Informáticas

## Facultad 3



### **Título: “Sistema para la informatización del proceso de Gestión de la caracterización estudiantil en la Facultad 3”**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:** Andrés Barbán Borrero

**Tutor(es):** Ing. Carlos Abel Capeáns Hurtado  
Ing. Dayannis Estrada Duarte  
MsC. Lieen Domínguez Díaz

La Habana, Junio 2014

“Año 56 de la Revolución”



*“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.  
Quien no añade nada a sus conocimientos, los disminuye”*

*Ernesto Guevara de la Serna*

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Andrés Barbán Borrero

Autor

\_\_\_\_\_  
**Ing.** Carlos Abel Capeáns Hurtado

Tutor

\_\_\_\_\_  
**Ing.** Dayannis Estrada Duarte

Tutor

\_\_\_\_\_  
**MsC.** Lieen Domínguez Díaz

Tutor

## DATOS DE CONTACTO

**Carlos Abel Capeáns Hurtado:** Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, año 2008.

Usuario: ccapeans

Cargo: Profesor

**Dayannis Estrada Duarte:** Graduado de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, año 2008.

Usuario: dduarte

Cargo: Profesor

**Lieen Domínguez Díaz:** Graduado de licenciada en educación en la especialidad Marxismo Historia, del ISP Cap. Silverio Blanco Núñez de Sancti Spíritus, año 2003.

Usuario: lieen

Cargo: Profesor

## DEDICATORIA

*Le dedico este trabajo principalmente a mi madre, mi padre a mis hermanos y toda la familia en general que son lo más importante en mi vida. A todas las personas que de una forma u otra contribuyeron a la realización del mismo.*

**Andrés Barbán Borrero**

## AGRADECIMIENTOS

*Primeramente empecemos según el orden en que fui conociendo a todas las personas que de alguna forma u otra hicieron que este trabajo fuera posible. Antes que todo agradecerles a mi mamá y mi papá por ser los principales promotores de que me encuentre aquí hoy. Por brindarme el apoyo y confianza durante todos estos años, los cuales han sido de mucho trabajo y sacrificio por parte de ellos hacia a mí. También por ser ejemplo de lucha en la vida, ya que son luchadores incansables. A mi hermana y hermano por brindarme su apoyo y confianza desde el primer momento que escogí esta carrera. A todos mis amigos de la vieja escuela (IPI), Luis Tomas, El Osbel, Karel que también me ayudaron. Al team del 3506 a todos ellos agradecerles ya que de todos aprendí algo. Bueno y aquí tenemos que hacer un alto para mencionar algunas personas que jugaron un papel protagónico en todo este trabajo: 1ro a la asociación de los mangotes ellos son el Rene que fue un factor clave para la terminación de esta tarea, Randy y Saidel que también contribuyeron, agradecer en especial también al Luis Ma (leleco) por su ayuda y preocupación. A mi amigo y hermano Leandro que no se puede quedar sin mencionar. Con carácter especial a mi novia Yamila por saber aguantarme en este tiempito que llevamos, por su preocupación y todo lo demás que ella sabe, también agradecerle a Liz. A mis tutores por ser de gran ayuda y por su preocupación para con este trabajo. A todos los miembros del tribunal agradecerles por todas las sugerencias brindadas.*

*A todas las personas que confiaron en mi a lo largo de toda mi carrera. Muchas gracias.*

**Andrés Barbán Borrero**

Actualmente la Facultad 3 de la Universidad de las Ciencias Informáticas (UCI) no cuenta con un sistema informático que permita realizar la caracterización de sus estudiantes. Esta caracterización se realiza todos los años teniendo en cuenta los datos personales obtenidos de cada estudiante. Dichos datos son indispensables para realizar las caracterizaciones de los grupos y de los años. Con estas caracterizaciones se detectan posibles problemas que puedan surgir dentro de la brigada además de definir las actividades correspondientes al proyecto educativo de los distintos grupos. Todo este proceso se realiza de forma manual y empleando documentos Word lo cual dificulta su agilización, teniendo en cuenta la cantidad de estudiantes existentes. La UCI concebida para el desarrollo de la informática, debe ser ejemplo en esta rama, al contar con personal calificado para informatizar sus procesos. Por lo anteriormente expuesto, el objetivo de este trabajo es desarrollar un sistema para la informatización del proceso de Gestión de la caracterización estudiantil en la Facultad 3. Para su cumplimiento se realiza un estudio detallado del proceso que será informatizado, se especifican las herramientas y tecnologías que se utilizarán en la solución del problema, se describe la solución propuesta y la validación de la misma. La utilización del sistema permitirá agilizar la Gestión de la caracterización estudiantil en la Facultad 3.

**Palabras claves:** caracterización estudiantil, proyecto educativo, gestión.

	ÍNDICE
Introducción .....	10
CAPÍTULO 1: Fundamentación teórica .....	15
1.1 Conceptos asociados al dominio del problema .....	15
1.1.1 Gestión de la Información .....	15
1.1.2 Caracterización Estudiantil.....	15
1.1.3 Proyecto Educativo .....	15
1.1.4 Características del proyecto educativo.....	15
1.2 Análisis de sistemas homólogos .....	16
1.2.1 Kimkëlen .....	16
1.2.2 SIGA: Software Integrado de Gestión Académica.....	16
1.2.3 Sistema de Gestión Escolar - Rutademic.....	17
1.2.4 Info@tletas.....	18
1.2.5 Sistema de Caracterización Integral del Graduado (SCIG) .....	19
1.3 Tecnologías y herramientas de desarrollo. ....	20
1.3.1 Lenguajes de desarrollo y modelado. ....	20
1.3.2 Marco de trabajo.....	21
1.3.3 Doctrine 2.....	22
1.3.4 JQuery 2.0.....	22
1.4 Herramientas de desarrollo .....	22
1.4.1 Gestor de Base de Datos .....	22
1.4.2 Servidor Web .....	23
1.4.3 Navegador Web.....	23
1.4.4 Herramienta para el modelado .....	23
1.4.5 Entorno de Desarrollo Integrado (IDE) .....	23
1.5 Patrones.....	24
1.5.1 Patrones de diseño .....	24
1.6 Modelo de desarrollo de software del centro CEIGE v1.2.....	25
Conclusiones Parciales .....	27
CAPÍTULO 2: Análisis y diseño del sistema.....	28
2.1 Modelado del negocio .....	28
2.1.1 Modelo Conceptual.....	28
2.1.2 Descripción del proceso de gestión de la caracterización estudiantil en la Facultad	
3. ....	29
2.1.3 Diagramas de procesos de negocio .....	31



## ÍNDICE

2.2 Modelado del sistema.....	31
2.2.1 Definición de estrategias de obtención requisitos.....	32
2.2.2 Requerimientos funcionales .....	32
2.2.3 Especificación de requisito Añadir problemas detectados en el grupo .....	35
2.2.4 Requerimientos no funcionales .....	36
2.2.5 Aplicación de técnicas de validación de requisitos .....	38
2.2.6 Patrones utilizados en el diseño del sistema .....	38
2.2.6 Diagramas de clases del diseño.....	42
2.2.8 Diagramas de secuencia.....	43
2.2.9 Modelo de datos.....	44
Conclusiones Parciales .....	48
CAPÍTULO 3: Implementación y prueba .....	49
3.1 Estándares de codificación.....	49
3.3 Diagrama de componentes.....	50
3.2 Diagrama de despliegue.....	51
3.4 Validación del diseño propuesto .....	52
3.5 Validación de la solución propuesta .....	58
3.5.1 Pruebas de software .....	58
3.5.2 Simulación del proceso de Gestión de la caracterización estudiantil de la Facultad	
3. ....	63
Conclusiones Parciales .....	65
Conclusiones .....	66
Referencias.....	68
ANEXOS.....	71

*“La información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje, es un conocimiento explícito extraído por seres vivos o sistemas expertos como resultado de interacción con el entorno” (1).* La tendencia actual en el mundo es que la información tenga un crecimiento acelerado. Para manejar esta situación se hace necesaria la utilización de los sistemas de gestión de la información, los cuales juegan un papel fundamental en la gestión del conocimiento de las organizaciones.

La Universidad de las Ciencias Informáticas, concebida para contribuir al desarrollo de la informática en Cuba, ha desempeñado un papel fundamental en el desarrollo de software tanto para entidades nacionales como internacionales. Dentro de la universidad se evidencia el auge que ha tenido la informatización, en la misma se han realizado varios sistemas para la gestión de sus procesos; dígase el sistema de guardia, directorio de personas, guía telefónica, gestión universitaria, sistemas para comedores, transportación, entre otros.

A pesar de que se han informatizado una serie de procesos aún se debe seguir trabajando en este sentido, tal es el caso del proceso de la gestión de la caracterización estudiantil. La Facultad 3 también se ve implicada en este proceso el cual resulta de gran importancia dentro de la universidad.

Antes de profundizar en el proceso de gestión de la caracterización estudiantil se debe analizar en qué consiste dicha caracterización. *“La caracterización constituye la vía para profundizar en el conocimiento de las cualidades personales de los estudiantes y las características principales del grupo. Tiene como objetivo fundamental que el estudiante establezca un grado de compromiso con el desarrollo de cualidades afines a los requerimientos y exigencias de su futura profesión y está orientada a profundizar en las cualidades intelectuales, la motivación, las cualidades morales, los hábitos y habilidades para el desempeño profesional, y el desarrollo sociopolítico, dando salida a las expectativas del profesional que se necesita” (2).*

El proceso de gestión de la caracterización es realizado por primera vez cuando el estudiante comienza el primer año, con el objetivo de brindarle una atención diferenciada a partir del análisis de sus características personales y sus principales preferencias. Sin embargo, se debe realizar una actualización anual, teniendo en cuenta, que la situación personal de todo ser humano cambia con el tiempo y cualquier cambio por mínimo que sea puede afectar su comportamiento; en este caso pudiera influir negativamente en su desempeño docente y actitud ante el estudio.

## INTRODUCCIÓN

Pueden presentarse distintos tipos de cambios en la vida del estudiante que impliquen realizar alguna actualización de su caracterización y que constituyen aspectos para realizar su seguimiento. De esta forma se torna más fácil mitigar los problemas docentes que pudieran surgir. Entre los elementos que deberían ser analizados se encuentran los personales (salud, pareja, relaciones interpersonales, preferencias sexuales), familiares (hijos, enfermedad o fallecimiento de familiares cercanos, situación de la vivienda, situación económica), así como sus preferencias docentes, científicas o extracurriculares.

El profesor principal del año y el guía de la brigada deben conocer en todo momento cuáles son los estudiantes que presentan algún problema familiar o personal, por ejemplo, se debe profundizar en la situación económica con el objetivo de analizar posibles préstamos estudiantiles, así como en los alumnos que presenten alguna situación en su familia que necesiten una atención diferenciada por parte de sus profesores.

En muchas ocasiones la información de la caracterización estudiantil se pierde al terminar un curso escolar, lo que provoca que al año siguiente el estudiante deba redactarla íntegramente, teniendo en cuenta que a veces los cambios son irrelevantes, esta actividad resulta molesta para los alumnos y los profesores guías. Además se emplea mucho tiempo desde el momento en que se da la orientación al estudiante hasta que finalmente se logra acumular todas las caracterizaciones del grupo.

La caracterización individual de cada estudiante es de vital importancia a la hora de conformar la caracterización y el proyecto educativo de la brigada. Para el profesor guía resulta arduo el proceso de realización de la caracterización de la brigada, pues debe revisar cada caracterización individual para identificar las características comunes de los estudiantes así como sus preferencias curriculares y extensionistas. Esta información también debe ser actualizada cada año, debido a los cambios de grupo, bajas, reingresos o reincorporaciones, así como cambios en las características de los estudiantes que implican cambios en la brigada.

Usualmente es solicitada determinada información relativa a un estudiante, o grupo de estudiantes por la dirección de la facultad o por los profesores del colectivo pedagógico. La obtención de la información se dificulta debido a que es manejada a través de herramientas tales como Word y Excel, lo que provoca que la búsqueda de datos se atrase por la poca flexibilidad que éstas poseen. Así como problemas en el control de versiones de la documentación, en la actualización de la caracterización y proyecto educativo de la brigada.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema a resolver**:  
*¿Cómo agilizar el proceso de Gestión de la caracterización estudiantil en la Facultad 3?*

## INTRODUCCIÓN

El **objeto de estudio** de la presente investigación lo constituye *Proceso de Gestión de la caracterización estudiantil* y como **campo de acción** *Proceso de gestión de la caracterización estudiantil en la Facultad 3*. Se define como **objetivo general** de la presente investigación: *Desarrollar un Sistema para la informatización del proceso de Gestión de la caracterización estudiantil en la Facultad 3*.

A partir de un análisis del objetivo general se derivan los siguientes **objetivos específicos**:

1. *Fundamentar la investigación mediante la elaboración del Marco Teórico para sustentar los conceptos y la propuesta de desarrollo del sistema de Gestión de la caracterización estudiantil en la Facultad 3.*
2. *Realizar el análisis y diseño de un sistema que informaticice el proceso de Gestión de la caracterización estudiantil en la Facultad 3.*
3. *Realizar la implementación de un sistema que informaticice el proceso de Gestión de la caracterización estudiantil en la Facultad 3.*
4. *Validar la propuesta de solución.*

Para dar cumplimiento a los objetivos específicos anteriores se diseñaron las siguientes **tareas de investigación**:

1. *Elaborar el marco conceptual de la investigación.*
2. *Estudiar el proceso de Gestión de la caracterización estudiantil en la Facultad 3.*
3. *Estudiar y seleccionar técnicas para captura de requisitos funcionales.*
4. *Estudiar y seleccionar patrones de diseño.*
5. *Seleccionar la metodología de desarrollo de software a emplear.*
6. *Definir la notación y/o lenguaje de modelado a utilizar.*
7. *Definir herramientas y tecnologías a utilizar para el desarrollo de la solución.*
8. *Elaborar Modelo conceptual.*
9. *Identificar requisitos funcionales utilizando las técnicas definidas.*
10. *Describir los requisitos funcionales con la plantilla establecida para ello.*
11. *Realizar los diagramas de clases de diseño.*
12. *Diseñar el modelo de datos mediante un diagrama de entidad-relación.*
13. *Validar el diseño aplicando métricas.*
14. *Realizar diagrama de componentes.*
15. *Definir estándares de codificación.*
16. *Realizar el diagrama de despliegue teniendo en cuenta los aspectos técnicos de la solución.*
17. *Validar el software mediante pruebas de caja blanca, caja negra y de aceptación.*
18. *Validar la investigación desarrollada mediante una simulación del proceso.*

## INTRODUCCIÓN

Se plantea la siguiente **idea a defender** para la investigación: *Si se desarrolla un sistema para la informatización del proceso de gestión de la caracterización estudiantil en la Facultad 3, se contribuirá a agilizar la realización de dicho proceso.*

En la presente investigación científica se emplearon los siguientes métodos teóricos y empíricos:

### **Métodos teóricos:**

- **Histórico–lógico:** este método se utiliza en la investigación para el análisis de los sistemas homólogos existentes, el estudio de las características, ventajas y desventajas de dichos sistemas, con el propósito de obtener un mayor conocimiento del objeto de estudio.
- **Analítico–sintético:** este método se emplea para analizar la bibliografía relacionada con el tema, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- **Modelación:** este método permite descubrir y estudiar las relaciones entre los procesos de gestión de la caracterización de los estudiantes, mediante la realización de modelos con vista a investigar la realidad y comprender el objeto de estudio con más claridad. Además permite representar lo que se quiere estudiar de una forma más simple.

### **Métodos empíricos:**

- **Observación:** este método permite la comprensión de la problemática de la investigación ya que posibilita la investigación de los procesos que se ven involucrados en la caracterización estudiantil en la Facultad 3.
- **Entrevista:** este método permitirá conocer las causas y qué soluciones se le dan actualmente a los problemas sobre el registro de información para la caracterización de los estudiantes y si existen normas que rigen dicho registro.

El documento de tesis consta de una introducción, tres capítulos, las conclusiones, recomendaciones, las bibliografías utilizadas y el cuerpo de anexos.

### **Capítulo 1. Fundamentación teórica**

En este capítulo se definen algunos conceptos importantes para la comprensión del objeto de estudio de la investigación. Se presentan los elementos teóricos que sirven de base a la investigación del problema planteado; analizando los principales conceptos relacionados con el objeto de estudio. Se realiza un análisis de otras soluciones existentes y se presentan las herramientas y metodología a utilizar para el desarrollo de la solución propuesta.

## INTRODUCCIÓN

### **Capítulo 2. Análisis y diseño del sistema**

En este capítulo se describe el análisis y diseño de la solución. Se realiza la descripción del proceso de negocio, determinando las personas que interactuarán con el sistema y la descripción general de la propuesta. Se definen los requisitos funcionales y no funcionales del sistema a desarrollar. Por último, se describe la arquitectura, los patrones de diseño y se muestran diagramas de clases, de secuencias y el modelo de la base de datos.

### **Capítulo 3. Implementación y prueba**

En este capítulo se presentan los estándares de diseño y codificación a seguir para la implementación del sistema. Además de los diagramas de despliegue y componentes. Se definen las pruebas de caja negra y caja blanca realizadas al software en aras de detectar errores o defectos relacionados con su funcionalidad y se muestran los resultados obtenidos de la validación del sistema, de los requisitos y el diseño.

El presente capítulo tiene como objetivo abordar los elementos teóricos que en su conjunto constituyen la base para la fundamentación teórica de la presente investigación. Se realiza un análisis de algunos de los principales sistemas que gestionan la información a nivel nacional e internacional con el objetivo de identificar sus principales características. Además, se exponen conceptos asociados al objeto de estudio de la investigación y se detallan las tecnologías y metodologías utilizadas para la implementación de la propuesta de solución.

## 1.1 Conceptos asociados al dominio del problema

**1.1.1 Gestión de la Información:** Proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización. Genera nuevos conocimientos (3).

**1.1.2 Caracterización Estudiantil:** Constituye la vía para profundizar en el conocimiento de las cualidades personales de los estudiantes y las características principales del grupo. Tiene como objetivo fundamental que el estudiante establezca un grado de compromiso con el desarrollo de cualidades afines a los requerimientos y exigencias de su futura profesión y está orientada a profundizar en las cualidades intelectuales, la motivación, las cualidades morales, los hábitos y habilidades para el desempeño profesional, y el desarrollo sociopolítico, dando salida a las expectativas del profesional que se necesita (2).

**1.1.3 Proyecto Educativo:** Los objetivos del proyecto educativo son la expresión concreta, en cada año académico, de la estrategia de trabajo de la facultad relacionada con la labor educativa y se corresponden con los objetivos formulados para el año en cuestión (4).

### 1.1.4 Características del proyecto educativo:

**Tabla 1. Características del proyecto educativo (4).**

OBJETIVOS	Pocos Integradores Que caractericen al año
ACCIONES	Precisas En cada una de las dimensiones

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

	Balanceadas adecuadamente
CRITERIOS DE MEDIDA	Precisos Cuantificables Evaluables sin dificultades

### 1.2 Análisis de sistemas homólogos

A continuación se realizará un análisis valorativo de algunos los sistemas de gestión de la información que existen actualmente en Cuba y el mundo, con el objetivo de verificar que puedan ser posibles soluciones de la problemática planteada, de no ser así revisar las funcionalidades y características que puedan ayudar para la posible propuesta de solución.

#### 1.2.1 Kimkëlen

Es un sistema de gestión integrada de colegios secundarios realizado por el Centro Superior para el Procesamiento de la Información (CesPi) perteneciente a la Universidad Nacional de La Plata (UNLP). El sistema permite administrar y centralizar toda la información y registros propios de una institución educativa secundaria.

Este sistema posibilita la personalización a través de “comportamientos” que responden a los diversos modos de gestionar una institución educativa. Cada comportamiento permite definir formas de evaluación, seguimiento de inasistencias, sanciones disciplinarias, impresión de boletines, generación de reportes, y más.

Kimkëlen funciona de manera online y permite el acceso y la consulta inmediata por parte de todos los usuarios, sean estos docentes o no docentes. Además de estar disponible para su libre utilización, este sistema cuenta con una comunidad de usuarios que dan cuenta de su estabilidad y posibilidad de adaptación (5).

El sistema no puede ser aplicado en la facultad debido a las características específicas que posee, por ejemplo es un sistema destinado para escuelas secundarias. Tampoco cuenta con el registro de la información personal de los estudiantes.

#### 1.2.2 SIGA: Software Integrado de Gestión Académica

La aplicación informática utilizada por la Universidad de Córdoba para la automatización de la gestión académica es el SIGA (Sistema Integral de Gestión Académica). Esta aplicación, desarrollada por el Servicio de Informática de la Universidad, se encuentra en continua evolución, ampliando nuevas funcionalidades y adaptándose a las nuevas tecnologías.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Las funcionalidades soportadas por el SIGA son muy amplias, abarcan desde la gestión del acceso a la universidad hasta la tramitación de los títulos pasando por la matrícula, actas de examen, expedientes académicos, estadísticas, informes, gestión de becas, convalidaciones, etc (6).

Los estudiantes interactúan con el sistema directamente, existen varios módulos destinados a satisfacer las necesidades de estos, como son el módulo de Opciones de Alumnos, donde los estudiantes a través de los submódulos Gestión de Asistencias y Gestión de Calificaciones pueden conocer todos los detalles de su asistencia a clases y evaluaciones. Para los estudiantes matriculados de manera no presencial en el centro, está el submódulo Gestión de Tutorías que controla el número y tipo de contactos del alumno con el centro a través de su tutor. SIGA posee una gran desventaja, y es que su interfaz no es amigable lo cual puede provocar serios problemas de utilización teniendo en cuenta la diversidad de usuarios que interactúan con el sistema (alumnos, profesores, personal del centro, entre otros). Además de no contar con un registro de la información personal de cada estudiante, siendo esta de gran importancia para la elaboración de la caracterización estudiantil.

### 1.2.3 Sistema de Gestión Escolar - Rutademic

Es una herramienta útil que facilita el manejo de centros educativos, mejorando la eficiencia de sus áreas de operación (Gestión Académica, Disciplina, Asistencia, Matriculación, Pensiones, Facturación, Manejo de Usuarios, Comunicación, etc.).

Permite mejorar los procesos de gestión, ahorrando tiempo y recursos, brindando un excelente servicio tanto a profesores y personal administrativo como a padres de familia y estudiantes, es decir a toda la comunidad educativa de la institución (7).

#### **Ventajas:**

- Acceso web: el sistema puede instalarse en la institución educativa o en un servidor remoto en Internet. Todos los usuarios pueden acceder al sistema desde cualquier navegador de internet.
- Es muy parametrizable: se puede adaptar a la estructura que maneja el establecimiento educativo, la institución no debe adaptarse al sistema, el sistema se configura según la estructura de la institución educativa.
- Muy fácil de aprender, su uso es muy intuitivo.
- Facilita el trabajo y ahorra tiempo de docentes, personal administrativo, estudiantes y representantes.
- Es muy seguro. Administración de permisos a través de perfiles de usuarios.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Los estudiantes y representantes pueden revisar notas, actividades pendientes, horarios, lista de compañeros, etc. desde el colegio y desde internet.
- Reducción de tiempo de matriculación.
- Módulo de comunicación muy completo que permite el intercambio de información constante entre todos los miembros de la comunidad educativa.
- Impresión de reportes requeridos por el Ministerio de Educación (Régimen Escolar) y reportes internos para la institución educativa.
- El sistema puede ser instalado sobre cualquier plataforma (7).

A pesar de no ser aplicable como propuesta de solución, debido a que es un sistema para colegios privados. El sistema cuenta con algunas funcionalidades que pudieran tomarse en cuenta, tal es el caso de la gestión de la seguridad.

### 1.2.4 Info@tletas

Es la aplicación web oficial del Instituto Nacional de Deportes, Educación Física y Recreación (INDER). Surge como colofón de varias aplicaciones de escritorio confeccionadas según la solicitud de la dirección de alto rendimiento del INDER. Se encarga principalmente de la administración de los expedientes técnico-acumulativos de atletas y entrenadores nacionales e internacionales. Se encuentra instalada en el servidor del INDER y todo el personal autorizado puede acceder a ella a través de Internet o de la red nacional.

La aplicación cuenta con varios módulos, en el cual se incluye el módulo de Atletas, el cual permite la administración de datos, inscripción y manejo de los expedientes técnico-acumulativos de los atletas. Este módulo mantiene, en forma persistente, todos los datos necesarios para conformar el expediente técnico-acumulativo de un atleta:

- Datos generales.
- Datos sociales.
- Datos de la matrícula actual.
- Caracterizaciones.

El módulo se encarga de la inscripción de nuevos atletas y la gestión del expediente técnico-acumulativo el cual agrupa los datos bajo un criterio, como son las participaciones, méritos y demás aspectos, facilita el paso de un atleta de un centro a otro sin que se pierda la trayectoria que ha tenido el mismo hasta el momento actual.

El sistema cuenta además con un módulo de Reportes, encargado de generar reportes estadísticos de atletas y entrenadores visualizados de forma tabular y a través de gráfica de barras. Cada módulo tiene una parte administrativa y una pública. En cada pedido la

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

plataforma gestiona la cantidad de datos necesaria y suficiente evitando así que tanto el canal de conexión a la base de datos como el establecido entre el servidor web y el cliente, se congestionen. Esto trae consigo un mayor rendimiento de la aplicación y el ahorro de recursos técnicos.

Los reportes se comportan según el tipo de usuario que ha iniciado su sesión en el sistema:

- Supervisor Especial y Supervisor Nacional: visualizan los reportes a nivel nacional.
- Supervisor Provincial: visualizan los reportes a nivel provincial.
- Administrador, Metodólogo y Técnico: visualizan los reportes a nivel de centro (8).

La aplicación no puede emplearse como solución debido a que sus características no son aplicables en la facultad. Este es un sistema para escuelas deportivas donde los estudiantes realizan otro tipo de actividades propias de su formación como futuros atletas. Dentro de las funcionalidades con que cuenta el sistema se puede tener en cuenta el comportamiento de los reportes según el tipo de usuario.

### 1.2.5 Sistema de Caracterización Integral del Graduado (SCIG)

Es un sistema desarrollado por el Centro de Innovación de la Calidad de Educación (CICE) en la UCI, con el objetivo de automatizar el proceso de registro para contribuir a la caracterización integral de los estudiantes de quinto año. Este sistema permite el registro de datos generales y resultados obtenidos en las distintas esferas evaluadas en la universidad, mostrando un por ciento de las caracterizaciones realizadas por cada centro, para la toma de decisiones por parte de los directivos que están al frente del proceso. Luego se exporta la plantilla de caracterización en formato PDF para su almacenamiento o impresión. Esta plantilla de caracterización se le entrega a los empleadores para que conozcan las habilidades poseídas por el futuro profesional.

Este sistema desarrollado en la universidad a pesar de hacer un uso organizado de la gestión de la información no cuenta con muchas de las especificidades que se plantea en la problemática. Como por ejemplo: la obtención de la información personal, realizar la caracterización a otros años en la carrera, gestionar actividades del Proyecto Educativo, etc. Por lo tanto no se puede aplicar como propuesta de solución.

### 1.2.6 Conclusiones sobre el estudio de los sistemas homólogos

**Tabla 2. Análisis de los sistemas estudiados**

<b>Características/Sistemas</b>	<b>Kimkëlen</b>	<b>SIGA</b>	<b>Rutademic</b>	<b>Info@tletas</b>	<b>SCIG</b>
Gestionar caracterización por grupos de estudiante (brigada o año)	<b>no</b>	<b>no</b>	<b>no</b>	<b>no</b>	<b>no</b>

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Gestionar caracterización por estudiante	no	no	no	no	no
Gestión de actividades del Proyecto Educativo	no	no	no	no	no
Basado en la tecnologías libres	si	si	no	si	si
Manejo efectivo de la seguridad de la información	sí	sí	sí	sí	sí
Basado en tecnologías web	si	si	si	si	si

La investigación realizada arrojó como resultado que no existe ningún sistema que cumpla con las características definidas para la gestión de la caracterización estudiantil en la Facultad 3. Aunque cubran aspectos comunes los sistemas nacionales e internacionales, gestionan y planifican dependiendo de sus características propias; no obstante, pueden ser de gran utilidad algunas de las funcionalidades que manejan esos sistemas. Estos elementos evidencian la necesidad de elaborar una solución que responda acertadamente a la Gestión de la caracterización estudiantil en la Facultad 3. Con la propuesta de este sistema se pretende lograr que este proceso se realice de manera adecuada.

### 1.3 Tecnologías y herramientas de desarrollo.

La selección correcta de las herramientas y tecnologías que se utilizan en el desarrollo de un software se traduce en ahorro de tiempo y trabajo dentro de cualquier proyecto. Las herramientas utilizadas en el desarrollo del sistema son las que establece la plataforma especificada por la facultad (Plataforma Tecnológica Facultad 3). Se describen a continuación:

#### 1.3.1 Lenguajes de desarrollo y modelado.

**JavaScript 1.8:** Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (9).

**PHP 5.3:** Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. El código fuente escrito en PHP es invisible al

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

navegador y al cliente, esto hace que la programación sea segura y confiable. A esto se añaden valores como el hecho de ser un proyecto de código abierto, gratuito y multiplataforma (10).

Entre sus principales ventajas se puede mencionar:

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con PostgreSQL.
- Posee una amplia documentación en su página oficial, entre la cual sobresale que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite las técnicas de Programación Orientada a Objeto (POO).
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

**Lenguaje Unificado de Modelado (UML):** UML son las siglas para Unified Modeling Language, que en castellano quiere decir: Lenguaje de Modelado Unificado. Para comprender qué es el UML, basta con analizar cada una de las palabras que lo componen, por separado.

- Lenguaje: el UML es, precisamente, un lenguaje. Lo que implica que éste cuenta con una sintaxis y una semántica. Por lo tanto, al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.
- Modelado: el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.
- Unificado: unifica varias técnicas de modelado en una única.

Ya que el UML proviene de técnicas orientadas a objetos, se crea con la fuerte intención de que este permita un correcto modelado orientado a objetos (11).

### 1.3.2 Marco de trabajo

Symfony2 es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony2 está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows (12).

### 1.3.3 Doctrine 2

Es un asignador de objeto relacional (ORM) para PHP 5.3.0+ que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa capa de abstracción de base de datos (DBAL por DataBase Abstraction Layer). La principal tarea de los asignadores objeto↔relacionales es la traducción transparente entre objetos (PHP) y las filas relacionales de la base de datos.

Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos llamado Lenguaje de Consulta Doctrine (DQL por Doctrine Query Language), inspirado en Hibernate HQL. Además DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible (13).

### 1.3.4 JQuery 2.0

Es un software libre y de código abierto. Posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Es una biblioteca o Framework de JavaScript.

Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas Web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (14).

## 1.4 Herramientas de desarrollo

### 1.4.1 Gestor de Base de Datos

**PostgreSQL 9.2:** Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (15).

### 1.4.2 Servidor Web

**Apache 2.2.6:** Apache HTTP Server es un servidor web que se carga como un servicio más del sistema operativo. Cuando está activo, convierte tu máquina en un servidor capaz de enviar contenido a cualquier navegador.

La instalación de Apache HTTP Server es muy sencilla gracias al nuevo asistente, que activa el servicio y añade un icono de notificación en la barra de tareas. Un cómodo panel, el Apache Service Monitor, sirve para controlar el estado del servidor (16).

### 1.4.3 Navegador Web

**Mozilla Firefox 28.0:** Firefox es un navegador de Internet que fue lanzado el año 2002 por sus creadores de la Corporación Mozilla.

Este navegador de Internet corresponde al proyecto, con interfaz gráfica, desarrollado por la corporación Mozilla. En sus comienzos, lo que hoy se conoce por Mozilla Firefox, tuvo varios nombres, tales como Phoenix y Firebird.

Por otra parte, este novedoso navegador corresponde a un proyecto “Open Source”, o en español, “Código Abierto”, lo que quiere decir que todo desarrollador puede modificar el código y así poder mejorarlo. Se trata de un navegador multiplataforma, es decir, que se encuentra disponible en diferentes versiones y así es posible utilizarlo con Microsoft Windows, Mac OS X, y también con GNU/Linux (17).

### 1.4.4 Herramienta para el modelado

**Visual Paradigm8.0:** Es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (18).

### 1.4.5 Entorno de Desarrollo Integrado (IDE)

**NetBeans 7.4:** Es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: **Java**, **PHP**, **Groovy**, **C/C++**, **HTML5**,... Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS (19).

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.5 Patrones

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software.

#### 1.5.1 Patrones de diseño

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables (20).

##### ➤ **Patrones GRASP**

Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades) (20).

**Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad.

**Creador:** Este patrón es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.

**Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta.

**Bajo Acoplamiento:** Este patrón es el encargado de asignar una responsabilidad para conservar bajo acoplamiento.

**Controlador:** Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

##### ➤ **Patrones GOF**

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides) se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento (20).

**Creacionales:** Los patrones creacionales abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

**Estructurales:** Los patrones estructurales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionar nuevas funcionalidades.

**Comportamiento:** Los patrones de comportamiento están relacionados con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos.

### 1.6 Modelo de desarrollo de software del centro CEIGE v1.2

El ciclo de vida de los proyectos del CEIGE (**Figura 1**) tiene en cuenta las actividades de cada una de las fases y áreas de procesos que plantea el nivel dos de CMMI establecido en la UCI. Este abarca el total de acciones que se realizan en las distintas líneas de desarrollo para la elaboración del servicio o producto final, sin embargo, se debe adaptar a las características particulares del proyecto que puede que no ejecute determinada disciplina, así como la elaboración de determinados artefactos del total aquí definido (21).

En la **Figura 1** se muestra el Ciclo de vida de proyectos de CEIGE. Constituido por 2 fases: Inicio y Desarrollo, dentro de cada una se especifican las disciplinas.



**Figura 1.** Ciclo de vida de proyectos del CEIGE (21)

#### Descripción general de la disciplina Estudio preliminar

Se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel. Se realiza un estudio inicial de la organización cliente que permite obtener información

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo, y decidir si se ejecuta o no el proyecto.

### **Descripción general de la disciplina Modelado del negocio**

Es la disciplina destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito.

### **Descripción general de la disciplina Requisitos**

El esfuerzo principal en la disciplina de Requisitos es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del sistema. Se especifican los requisitos funcionales y no funcionales.

### **Descripción general de la disciplina Análisis y diseño**

Durante esta disciplina es modelado el sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima disciplina. Los artefactos generados en esta etapa son más formales y específicos de una implementación. En caso de llevarse a cabo la reutilización de componentes software ya desarrollados, durante esta disciplina se ajusta el modelado existente a los requisitos actuales.

### **Descripción general de la disciplina Implementación**

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes.

### **Descripción general de la disciplina Pruebas internas**

Durante esta disciplina se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple.

### **Descripción general de la disciplina Pruebas de liberación**

Se aplican pruebas diseñadas e implementadas por el Laboratorio Industrial de Pruebas de Software a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### **Conclusiones Parciales**

Con la realización del presente capítulo se concluye lo siguiente:

- Los conceptos utilizados ayudan a lograr un mejor entendimiento de la investigación científica en cuestión, conformando la fundamentación teórica que sustenta el desarrollo de la investigación.
  
- La realización del estudio del estado del arte acerca de los sistemas de gestión existentes, tanto en Cuba como en el mundo, permite arribar a la conclusión de que no existe un sistema de gestión que satisfaga las necesidades planteadas por la Facultad 3, en aras de un mejor control y agilización del proceso de caracterización.
  
- La descripción de las tecnologías, lenguajes y herramientas con las cuales se va a desarrollar la solución propuesta permite lograr un mejor entendimiento de la misma.



En el presente capítulo se realiza la descripción de la propuesta para dar solución a la problemática existente. Para ello se formularon todas las características que tendrá el sistema, partiendo del estudio del proceso actual del negocio donde se van a identificar los actores, el rol y la responsabilidad que tiene cada uno con el sistema. Además se lleva a cabo el modelado del negocio del proceso informatizado, además de realizarse una descripción de dicho proceso. Se identifican los requisitos funcionales y no funcionales, se define el diagrama de procesos, lo que permite dar paso al diseño donde se realizan los diagramas de clases del diseño, diagramas de secuencias, modelo conceptual, modelo físico de datos y además, se especificará la utilización de un conjunto de patrones dentro del diseño del sistema.

## **2.1 Modelado del negocio**

La modelación de procesos de negocio permite realizar una exploración del dominio del problema, con el fin de lograr comprensión por parte del equipo de desarrollo de los procesos que se realizan actualmente en la entidad y la relación que existe entre estos. De esta forma se van determinando necesidades operacionales, así como restricciones que presenta la entidad, obteniéndose finalmente un entendimiento del negocio.

### **2.1.1 Modelo Conceptual**

En el modelo conceptual se trata de obtener el esquema conceptual de la base de datos a partir de la lista descriptiva de objetos y asociaciones identificadas en la organización durante el análisis.

En la **Figura 2** se muestra el Modelo Conceptual definido para el negocio estudiado.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

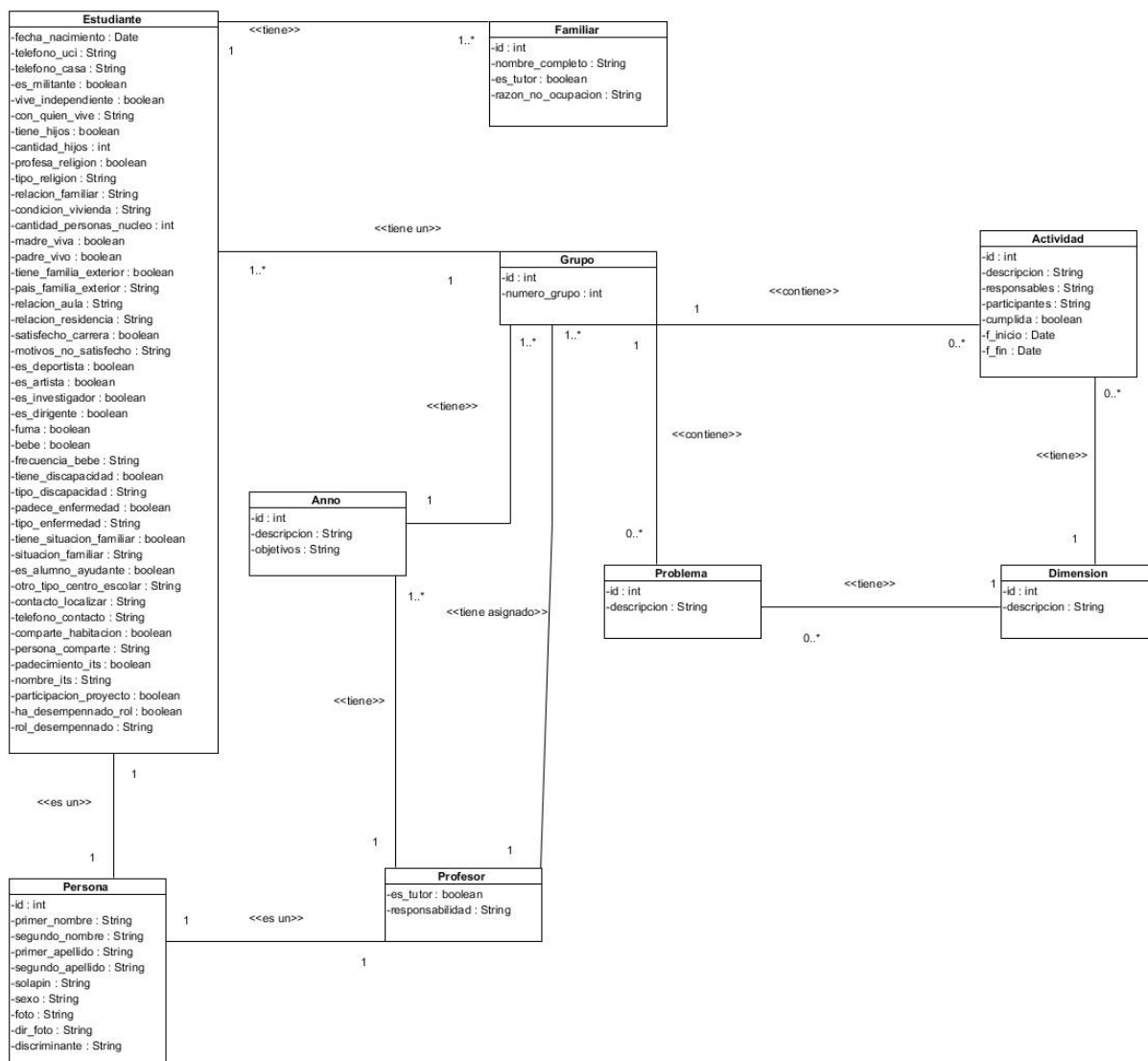


Figura 2. Modelo Conceptual

### 2.1.2 Descripción del proceso de gestión de la caracterización estudiantil en la Facultad 3.

La gestión de la caracterización se empieza a realizar desde el 1er año. Este proceso comienza cuando el profesor principal del año le solicita al profesor guía la caracterización de los estudiantes. El profesor guía es el encargado enviar las planillas de caracterizaciones a los estudiantes. El estudiante es el encargado de llenar la planilla de caracterización y se la entrega al profesor guía de la brigada. El profesor guía aprueba la caracterización y luego pasa a detectar los problemas del grupo y define las actividades del Proyecto Educativo, para luego entregárselos al profesor principal del año.

A continuación se muestra en la **Tabla 3** la descripción de las personas relacionadas con el negocio.

Tabla 3. Personas relacionadas con el negocio

Rol	Descripción
-----	-------------

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

Profesor Principal del año	Encargado de asociar el grupo al profesor guía correspondiente.
Profesor Guía	Persona encargada de revisar la caracterización, detectar los problemas de la brigada y elaborar las actividades del proyecto educativo del grupo que tiene asociado como profesor guía. Además realizar reportes para ver las características de sus estudiantes.
Estudiante	Persona que registra sus datos y consulta la información generada en la caracterización.

**Tabla 4. Descripción del proceso de negocio Gestión de la caracterización estudiantil en la Facultad 3.**

<b>Objetivo</b>	Realizar la caracterización de los estudiantes de la Facultad 3.
<b>Evento(s) que lo genera(n)</b>	N/A
<b>Pre condiciones</b>	N/A
<b>Marco legal</b>	N/A
<b>Clientes internos</b>	N/A
<b>Clientes externos</b>	N/A
<b>Entradas</b>	
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1.	Solicitar caracterización: El profesor principal le solicita al profesor guía la caracterización de los estudiantes.
2.	Solicitar llenado de caracterización: El profesor guía es el encargado enviar las planillas de caracterización a los estudiantes.
3.	Llenar caracterización: El estudiante es el encargado de llenar la planilla de caracterización y verificar si están correctos y completos los datos.
4.	Entregar caracterización: El estudiante se encarga de entregarle la planilla actualizada al profesor guía.
5.	Revisar caracterización del estudiante: El profesor guía revisa y aprueba la caracterización.
6.	Procesar datos para caracterización del grupo: El profesor guía realiza una búsqueda de los datos de los estudiantes para obtener información de la brigada.
7.	Identificar problemas del grupo: El profesor guía se encarga de detectar los problemas de la brigada.
8.	Definir actividades para Proyecto Educativo: El profesor guía se encarga de definir las actividades de la brigada.
9.	Entregar Proyecto Educativo: El profesor guía se encarga de entregar el Proyecto Educativo al profesor principal.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### 10. Recibir Proyecto Educativo: El profesor principal recibe el Proyecto Educativo

#### Pos-condiciones

Se ha realizado la caracterización del estudiante.

#### Salidas

Caracterización del estudiante.

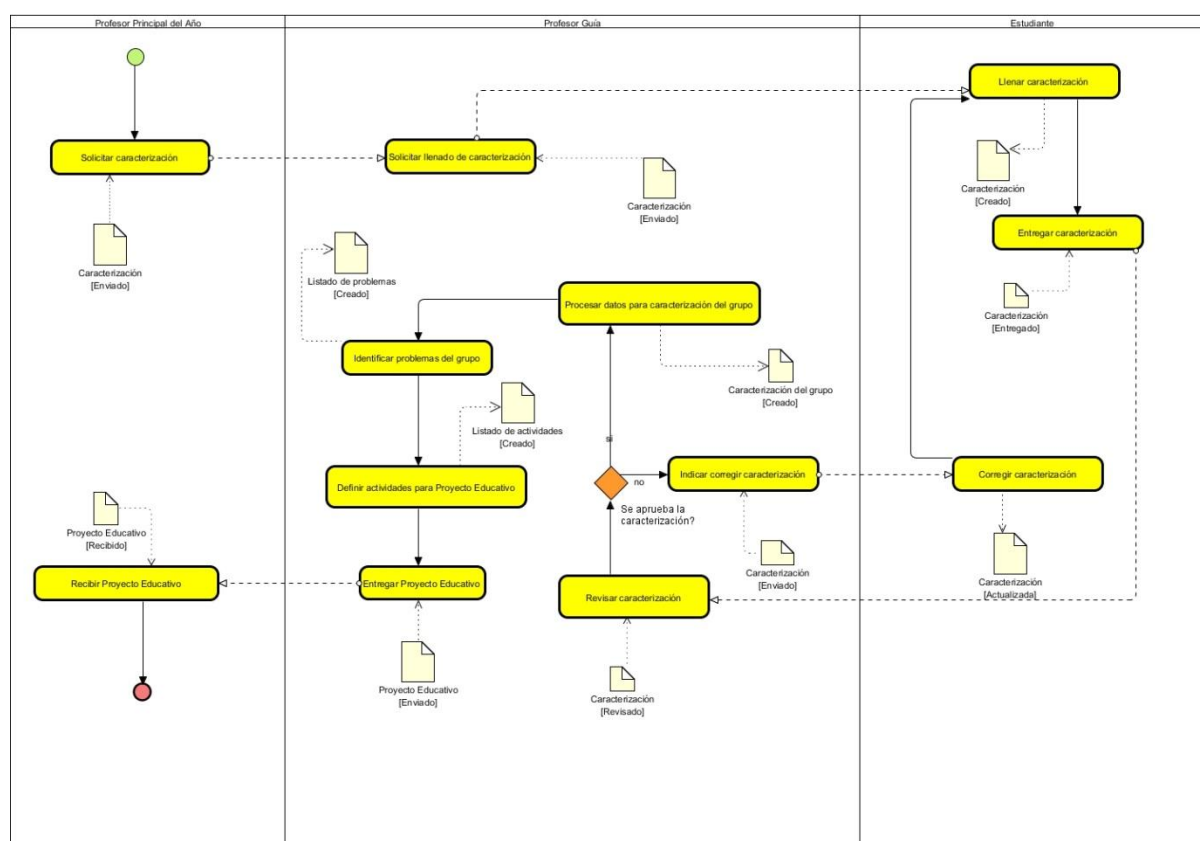
#### Asuntos pendientes

N/A

### 2.1.3 Diagramas de procesos de negocio

Un proceso de negocio es un conjunto de actividades relacionadas que permiten crear un producto o servicio final y tiene como objetivo fundamental satisfacer al cliente.

En la **Figura 3** se muestra el Diagrama de Proceso de negocio.



**Figura 3.** Diagrama del proceso de negocio Gestión de la caracterización estudiantil en la Facultad 3.

## 2.2 Modelado del sistema

Los conocimientos adquiridos hasta el momento sobre los conceptos que rodean al campo de acción, permitirán analizar las características que debe tener el sistema para garantizar que se cumplan los objetivos planteados.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### 2.2.1 Definición de estrategias de obtención requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan y entienden sus requisitos (22).

Existen varias técnicas para la identificación de requisitos, dentro de las que se encuentran:

- Entrevistas
- Sesiones de tormentas de ideas (Brainstorming)
- Revisión de la documentación técnica
- Cuestionarios
- Observación
- Prototipos
- Casos de uso

De las técnicas antes mencionadas, para la captura de requisitos de la solución propuesta se utilizaron las siguientes:

- Sesiones de tormentas de ideas (Brainstorming): se realizaron reuniones en grupo, con la participación del cliente, con el objetivo de generar ideas respecto a las funcionalidades requeridas para el sistema.
- Entrevistas: se realizaron reuniones con el cliente, en las cuales se efectuaron una serie de preguntas (no estructuradas), para lograr un mayor entendimiento de la problemática existente, y a partir de ahí poder identificar los requisitos del software.

### 2.2.2 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe realizar, es decir, define qué es lo que el sistema debe hacer. Especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema.

A continuación se listan los requisitos funcionales identificados para el sistema:

#### Listado de requisitos funcionales por agrupaciones.

##### Gestionar Estudiante

- ✓ RF 1: Mostrar caracterización.
- ✓ RF 2: Editar caracterización.
- ✓ RF 3: Listar estudiante.
- ✓ RF 4: Asociar estudiante a un grupo.
- ✓ RF 5: Disociar estudiante de un grupo.
- ✓ RF 6: Promover estudiante

##### Gestionar Grupo

- ✓ RF 7: Actualizar datos del grupo.



## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

- ✓ RF 8: Añadir grupo.
- ✓ RF 9: Editar grupo.
- ✓ RF 10: Eliminar grupo.
- ✓ RF 11: Listar grupo.

### **Gestionar Grupos de Roles**

- ✓ RF 12: Agregar grupo de roles.
- ✓ RF 13: Editar grupo de roles.
- ✓ RF 14: Eliminar grupo de roles.
- ✓ RF 15: Listar grupo de roles.

### **Gestionar Actividad**

- ✓ RF 16: Añadir actividad del grupo.
- ✓ RF 17: Editar actividad.
- ✓ RF 18: Eliminar actividad.
- ✓ RF 19: Listar actividad.
- ✓ RF 20: Mostrar detalles de la actividad.

### **Gestionar Problemas**

- ✓ RF 21: Añadir problema detectado en el grupo.
- ✓ RF 22: Editar problema.
- ✓ RF 23: Eliminar problema.
- ✓ RF 24: Listar problema.
- ✓ RF 25: Mostrar detalles del problema.

### **Gestionar Nomencladores**

#### **Gestionar deporte**

- ✓ RF 26: Añadir deporte.
- ✓ RF 27: Editar deporte.
- ✓ RF 28: Eliminar deporte.
- ✓ RF 29: Mostrar deporte.

#### **Gestionar dimensión**

- ✓ RF 30: Añadir dimensión.
- ✓ RF 31: Editar dimensión.
- ✓ RF 32: Eliminar dimensión.
- ✓ RF 33: Mostrar dimensión.

#### **Gestionar año académico**

- ✓ RF 34: Añadir año académico.
- ✓ RF 35: Editar año académico.
- ✓ RF 36: Eliminar año académico.
- ✓ RF 37: Mostrar año académico.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### **Gestionar semestre**

- ✓ RF 38: Añadir semestre.
- ✓ RF 39: Editar semestre.
- ✓ RF 40: Eliminar semestre.
- ✓ RF 41: Mostrar semestre.

### **Gestionar evento**

- ✓ RF 42: Añadir evento.
- ✓ RF 43: Editar evento.
- ✓ RF 44: Eliminar evento.
- ✓ RF 45: Mostrar evento.

### **Gestionar ocupación**

- ✓ RF 46: Añadir ocupación.
- ✓ RF 47: Editar ocupación.
- ✓ RF 48: Eliminar ocupación.
- ✓ RF 49: Mostrar ocupación.

### **Gestionar Vía de ingreso**

- ✓ RF 50: Añadir vía de ingreso.
- ✓ RF 51: Editar vía de ingreso.
- ✓ RF 52: Eliminar vía de ingreso.
- ✓ RF 53: Mostrar vía de ingreso.

### **Gestionar provincia**

- ✓ RF 54: Añadir provincia.
- ✓ RF 55: Editar provincia.
- ✓ RF 56: Eliminar provincia.
- ✓ RF 57: Mostrar provincia.

### **Gestionar municipio**

- ✓ RF 58: Añadir municipio.
- ✓ RF 59: Editar municipio.
- ✓ RF 60: Eliminar municipio.
- ✓ RF 61: Mostrar e municipio.

### **Gestionar organización**

- ✓ RF 62: Añadir organización.
- ✓ RF 63: Editar organización.
- ✓ RF 64: Eliminar organización.
- ✓ RF 65: Mostrar e organización.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### Gestionar nivel escolar

- ✓ RF 66: Añadir nivel escolar.
- ✓ RF 67: Editar nivel escolar.
- ✓ RF 68: Eliminar nivel escolar.
- ✓ RF 69: Mostrar e nivel escolar.

### Gestionar asignatura

- ✓ RF 70: Añadir asignatura.
- ✓ RF 71: Editar asignatura.
- ✓ RF 72: Eliminar asignatura.
- ✓ RF 73: Mostrar e asignatura.

### Gestionar manifestación cultural

- ✓ RF 74: Añadir manifestación cultural.
- ✓ RF 75: Editar manifestación cultural.
- ✓ RF 76: Eliminar manifestación cultural.
- ✓ RF 77: Mostrar manifestación cultural.

### Gestionar parentesco

- ✓ RF 78: Añadir parentesco.
- ✓ RF 79: Editar parentesco.
- ✓ RF 80: Eliminar parentesco.
- ✓ RF 81: Mostrar parentesco.

### Reportes

- ✓ RF 82: Mostrar reporte por criterio(s) de búsqueda.

### 2.2.3 Especificación de requisito Añadir problemas detectados en el grupo

A continuación se muestra la especificación de requisito Añadir problema.

**Tabla 5. Descripción textual del requisito**

<b>Precondiciones</b>	El usuario ha sido validado.
<b>Flujo de eventos</b>	
<b>Flujo básico Añadir problema</b>	
1.	Se introducen los datos del problema: Dimensión Descripción
2.	El sistema valida (ver validación 1) los datos introducidos.
3.	Si los datos son correctos el sistema los registra.
4.	El sistema confirma el registro de los datos.
5.	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se registró en el sistema un problema.
<b>Flujos alternativos</b>	

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### Flujo alternativo 3.a Información errónea

- 1 El sistema señala los datos erróneos y permite corregirlos.
- 2 El usuario corrige los datos.
- 3 Volver al paso 2 del flujo básico.

### Flujo alternativo 3.b Información incompleta

- 1 El sistema señala los campos vacíos y permite corregirlos.
- 2 El usuario corrige los datos.
- 3 Volver al paso 2 del flujo básico.

### Flujo alternativo 1.a El usuario vuelve a la lista

- 1 Concluye el requisito.

### Pos-condiciones

- 1 N/A

### Validaciones

- 1 N/A

Conceptos	Problema	Visibles en la interfaz:
		Dimensión
		Descripción

**Requisitos especiales** N/A

**Asuntos pendientes** N/A

A continuación se muestra en la **Figura 4** el prototipo elemental de interfaz gráfica de usuario del requisito.

### Adicionar problemas al grupo 3101

El prototipo muestra un formulario con los siguientes elementos:

- Un campo de texto con el título "Dimensión \*" y el contenido "Seleccione...".
- Un campo de texto con el título "Descripción del problema \*".
- Una barra de botones con tres opciones: "Guardar y editar", "Guardar y regresar al listado" y "Guardar y agregar otro".

**Figura 4.** Adicionar problemas

Como ejemplo ilustrativo y para una mayor comprensión del lector ver especificación de requisitos en el expediente del proyecto.

### 2.2.4 Requerimientos no funcionales

Los requerimientos no funcionales hacen relación a las características del sistema que aplican de manera general como un todo, más que a rasgos particulares del mismo. Estos

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema, y corresponden a aspectos tales como la disponibilidad, mantenibilidad, flexibilidad, seguridad, facilidad de uso, etc (23).

### **Apariencia o interfaz externa.**

- El sistema debe que ofrecer una interfaz amigable, fácil de operar.
- Debe ser interactivo con el usuario.
- La comunicación entre el servidor Web y las máquinas clientes será mediante el HTTP y entre el servidor y el directorio activo mediante LDAP.

### **Usabilidad.**

- Tiene que poseer una interfaz agradable para el cliente.

### **Rendimiento.**

- La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos y la velocidad de la consultas de la base de datos.
- El sistema debe ser capaz de funcionar al ser instalados en una PC todos sus componentes tales como Gestor de bases de Datos, Aplicación Web, etc.

### **Disponibilidad.**

- El sistema debe estar disponible las 24 horas del día sin ninguna interrupción.

### **Mantenibilidad.**

- El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.

### **Soporte.**

- Tiene que brindar soporte para grandes volúmenes de datos y velocidad de procesamiento.
- El sistema tiene que ser multiplataforma.
- Confiabilidad y seguridad.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al rol del usuario que esté activo.
- Tiene que brindar una protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Permitir autenticarse a través de un Servidor de Dominio.
- El sistema tiene que mantener en todo momento la seguridad de la información asegurando la autenticidad de la misma.

### **Hardware.**

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

### Cliente:

- Requerimientos mínimos: procesador Pentium IV a 800 MHz, 128mb de memoria RAM y un navegador web.
- Las computadoras clientes deben estar conectadas en red.

### Servidor:

- Requerimientos mínimos: procesador Dual Core a 3.00 GHz, 1gb de memoria RAM y una capacidad de 40gb de disco duro.
- El servidor debe tener al menos una tarjeta de red para establecer la conexión.

### **Software.**

#### Cliente:

- Sistema operativo con interfaz gráfica y soporte para red.
- Las interfaces deben ser compatibles con Mozilla Firefox 3.0 o superior.

#### Servidor:

- Servidor web Apache 2.2.6 o superior.
- Gestor de base de datos PostgreSQL 9.2 o superior.

### **2.2.5 Aplicación de técnicas de validación de requisitos**

Las técnicas que se utilizaron para la validación de los requisitos identificados fueron las siguientes:

#### **La Revisión Técnica Formal (RTF)**

Una vez terminada las especificaciones de los requisitos se realizaron reuniones con los clientes, donde revisaron cada una de estas especificaciones. Ante los errores detectados se volvieron a analizar para una nueva revisión en caso que fuese necesario. Con este proceso se pudo validar que la interpretación de cada una de las especificaciones no fuera ambigua, y que cada uno de los requisitos cumplía con lo que necesitaba el usuario final.

#### **Prototipo de interfaz**

Se elaboraron prototipos de interfaz, con la intención de simular las funcionalidades que necesita el sistema. Las interfaces obtenidas permitieron a los especialistas obtener una idea de cómo quedaría la estructura de la aplicación.

### **2.2.6 Patrones utilizados en el diseño del sistema**

El diseño de la propuesta de solución se realizó aplicando los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns), GOF (por sus siglas en inglés, Gang of Four, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, definen estructuras de clases que tratan una

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

situación en particular y especifican respuestas deseadas a sucesos o solicitudes de datos concretas respectivamente.

### 2.2.6.1 Patrones Grasp

El uso del Symfony 2, evidencia el uso de varios patrones de diseño que este trae incluidos por defecto en su arquitectura, además de estar concebidos de tal manera que obliga al programador a aplicarlos. Con su uso, se pretende establecer un lenguaje común entre los programadores, contribuir a la reutilización, ahorrar tiempo en la implementación y obtener un producto con calidad.

- ✓ **Experto:** Es uno de los patrones que más se utiliza cuando se trabaja con Symfony 2, con la inclusión de Doctrine para mapear la base de datos. Symfony 2 utiliza este ORM para implementar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad y contienen la información necesaria de la tabla que representan.
- ✓ **Creador:** En las clases controladoras del SiceBundle se encuentran las funcionalidades con el sufijo “action”, la cuales contiene las acciones y son las encargadas de ejecutarlas. Dentro de estas existen varias funcionalidades en las cuales se crean varios objetos o instancias de las clases que representan cada una de las tablas existentes en la base de datos.
- ✓ **Controlador:** Todas las peticiones web son manipuladas por un controlador frontal, ejemplo de esto son las clases “app.php” y “app\_dev.php”, que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases controladoras del sistema. En Symfony 2 hay una capa específicamente para los controladores, que son el núcleo del mismo, aquí cada clase en esta capa tiene su responsabilidad y es única, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros “yml” de configuración. Esto se aplica a la clase “app.php” que es el controlador frontal.
- ✓ **Alta Cohesión:** Se emplea en las clases controladoras ya que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea. Lo cual permite que el software sea flexible a cambios sustanciales con efecto mínimo.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

- ✓ **Bajo acoplamiento:** Este patrón se evidencia dentro del marco de trabajo Symfony 2 en las clases que implementan la lógica del negocio y de acceso a datos que se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja. Como existe poca dependencia entre esas clases se permite una mayor reutilización.

### Patrones GoF

Con el uso del marco de trabajo Symfony 2 se aplican los siguientes patrones GoF:

- ✓ **Fachada:** El uso de este patrón se evidencia en la implementación de las interfaces que brindará el sistema.
- ✓ **Command:** Este patrón se observa en las clases “app.php” y “app\_dev.php”, son las encargadas de establecer el módulo y la acción que se va a usar según la petición del usuario. Esto se aplica en la clase “RoutingManipulator.php”, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el marco de trabajo, la cual se puede activar o desactivar. En este método se comprueba la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el action que debe responder a la petición.
- ✓ **Decorator:** Este patrón se observa en la clase “base.html.twig”, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado “layout.php.twig” es el que contiene el layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este procedimiento es una implementación del patrón Decorator.
- ✓ **Registry:** Este patrón se aplica en la clase “GlobalVariables.php”, que es la encargada de acumular todas las variables de uso global en el sistema.

#### 2.2.6.2 Estilo arquitectónico

**Modelo-Vista-Controlador (MVC):** Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.



## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

### **El modelo es el responsable de:**

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc (24)).

### **El controlador es responsable de:**

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W".

### **Las vistas son responsables de:**

- Recibir los datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

En la **Figura 5** se muestra el funcionamiento del estilo arquitectónico Modelo-Vista-Controlador.

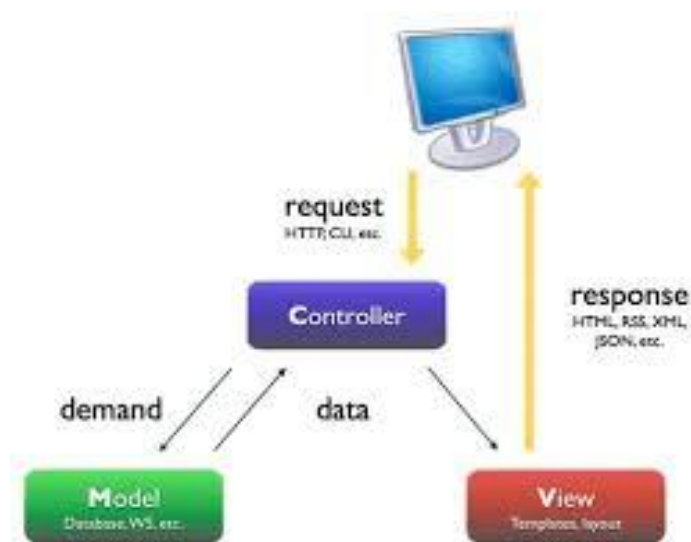


Figura 5. Estilo Modelo-Vista-Controlador (17).

**A continuación se muestra el Modelo-Vista-Controladora aplicado en Symfony2**

Symfony2 basa su funcionamiento interno en la arquitectura Modelo-Vista-Controlador (MVC), utilizada por un gran número de frameworks web. No obstante, según su creador Fabien Potencier: "Symfony2 no es un framework MVC. Symfony2 sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad de los desarrolladores, aunque existen librerías para integrar fácilmente los ORM más conocidos, como Doctrine y Propel. En cualquier caso, resulta esencial conocer cómo se aplican los principios fundamentales de la arquitectura MVC a las aplicaciones Symfony2 (Consultado del manual de Symfony2).

En la Figura 6 se observa el esquema simplificado del funcionamiento interno de Symfony2.

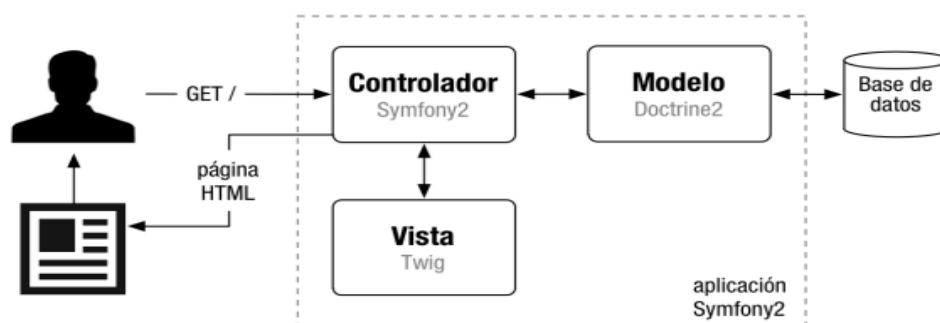


Figura 6. Funcionamiento interno de Symfony2 (Consultado del manual de Symfony2).

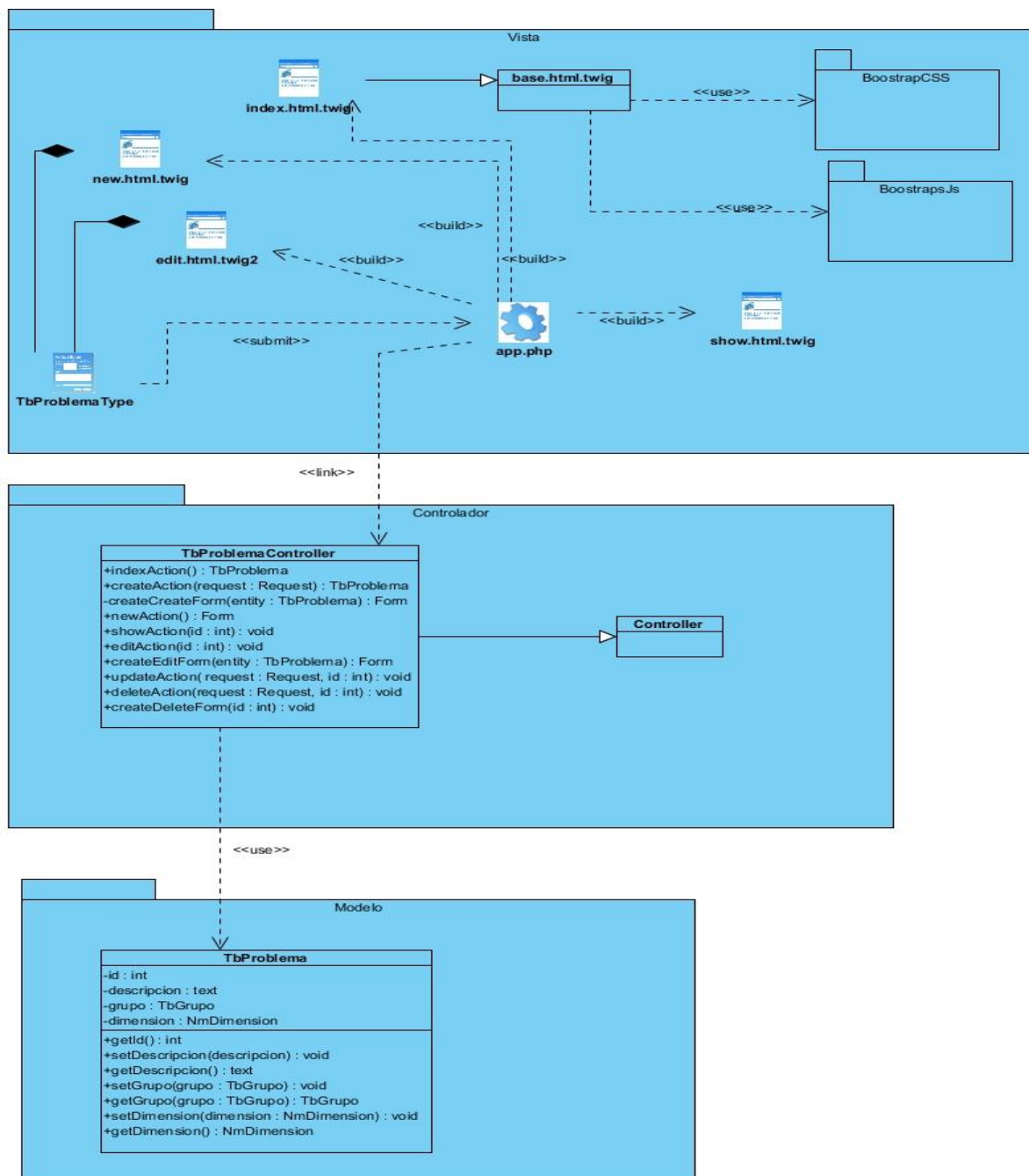
**2.2.6 Diagramas de clases del diseño**

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Nos sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

convencimiento. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso (25).

Se muestra a continuación el diagrama de clases del diseño obtenido para la agrupación de requisitos Gestionar Problema **Figura 7**:



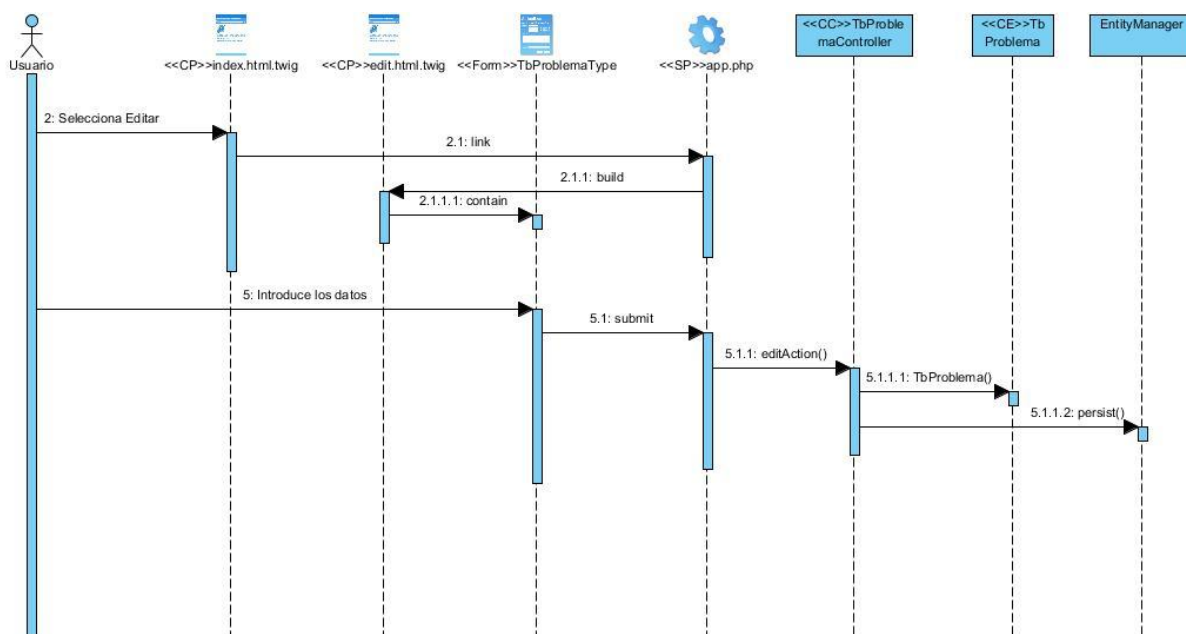
**Figura 7.** Diagrama de clases Gestionar Problemas

### 2.2.8 Diagramas de secuencia

Un diagrama de secuencia muestra una interacción, que representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. El tiempo fluye hacia abajo en el diagrama y muestra el flujo de control de un participante a otro (26).

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

Se muestra en la **Figura 8** el diagrama de secuencia relacionado con el requisito Editar Problema.



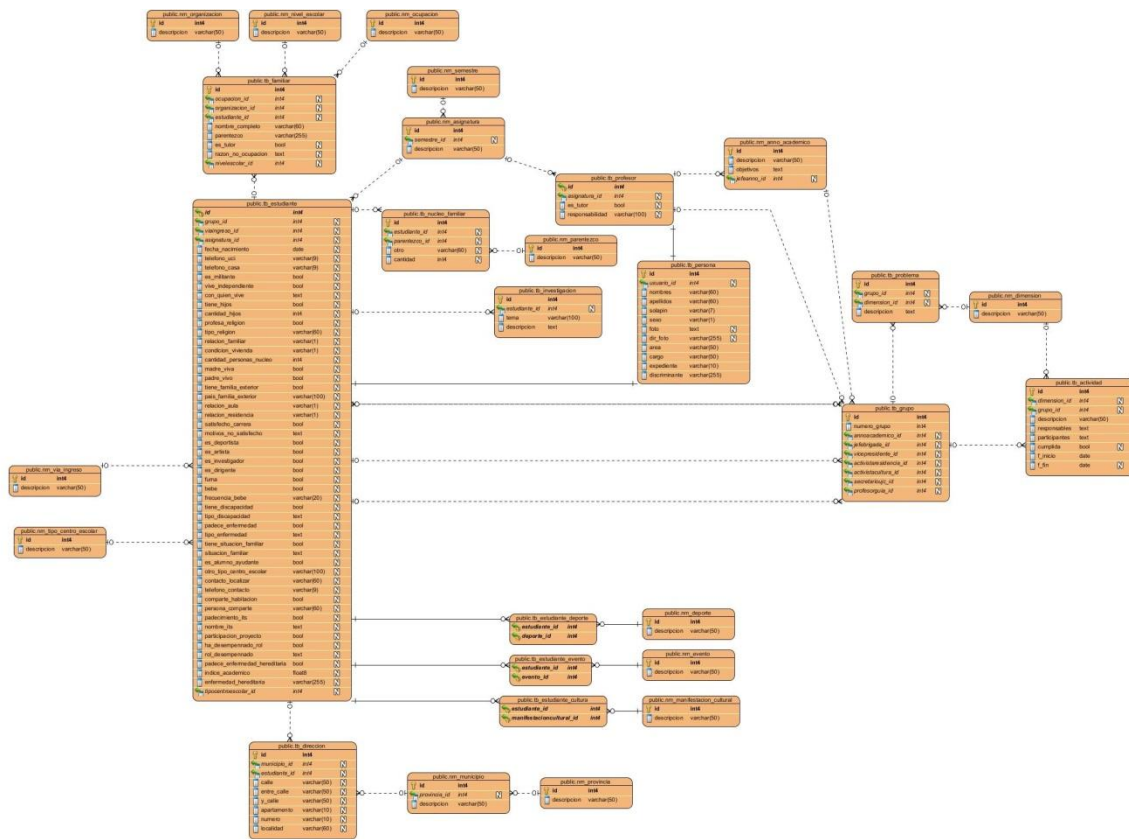
**Figura 8.** Diagrama de secuencia Editar problema

### 2.2.9 Modelo de datos

El modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Permite describir las estructuras de datos de la base de datos (el tipo de los datos que incluye la base de datos y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base de datos (27).

A continuación se presenta en la **Figura 9** el modelo de datos del sistema el cual cuenta con 28 tablas.

# CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA



**Figura 9. Modelo de datos**

## 2.2.9.1 Descripción de las tablas

Se muestran a continuación las descripciones de las principales tablas del modelo de datos.

**Tabla 6. Descripción de la tabla “tb\_estudiante”**

Nombre: “tb_estudiante”		
Tipo: Entidad		
Descripción: almacena los datos correspondientes a cada uno de los estudiantes.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
grupo_id	int	Identificador de la tabla tb_grupo.
raza_id	int	Identificador de la tabla nm_raza.
asignatura_id	int	Identificador de la tabla nm_asignatura.
direccion_id	int	Identificador de la tabla tb_direccion (Llave foránea).
fecha_nacimiento	date	Fecha de nacimiento del estudiante.
telefono_uci	varchar	Teléfono del apartamento en la UCI.
telefono_casa	varchar	Teléfono de la casa del estudiante.
es_militante	bool	Indica si el estudiante es militante o no.
vive_independiente	bool	Indica si el estudiante vive independiente o no.
con_quien_vive	text	Describe personas que viven con el estudiante.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

tiene_hijos	bool	Indica si el estudiante tiene hijos o no.
cantidad_hijos	int	Cantidad de hijos del estudiante.
profesa_religion	bool	Indica si el estudiante profesa alguna religión.
tipo_religion	varchar	Descripción de la religión.
relacion_familiar	varchar	Relación con la familia.
condicion_vivienda	varchar	Condición de la vivienda.
cantidad_personas_nucleo	int	Cantidad de personas en el núcleo.
madre_viva	bool	Indica si la madre del estudiante es viva o no.
padre_vivo	bool	Indica si el padre del estudiante es vivo o no.
tiene_familia_exterior	bool	Indica si el estudiante tiene familia en el exterior.
pais_familia_exterior	varchar	Nombre del país.
relacion_aula	varchar	Relación en el aula.
relacion_residencia	varchar	Relación en la residencia.
satisfecho_carrera	bool	Indica si el estudiante está satisfecho con la carrera.
motivos_no_satisfecho	text	Motivo en caso de que no esté satisfecho con la carrera.
es_deportista	bool	Indica si el estudiante es deportista o no.
es_artista	bool	Indica si el estudiante es artista o no.
es_investigador	bool	Indica si el estudiante es investigador o no.
es_dirigente	bool	Indica si el estudiante es dirigente o no.
fuma	bool	Indica si el estudiante fuma o no.
bebe	bool	Indica si el estudiante bebe o no.
frecuencia_bebe	varchar	Frecuencia con que bebe.
tiene_discapacidad	bool	Indica si el estudiante tiene alguna discapacidad o no.
tipo_discapacidad	text	Tipo de discapacidad.
padece_enfermedad	bool	Indica si el estudiante tiene alguna enfermedad o no.
tipo_enfermedad	text	Tipo de enfermedad.
tiene_situacion_familiar	bool	Indica si el estudiante tiene alguna situación familiar.
situacion_familiar	text	Situación familiar.
es_alumno_ayudante	bool	Indica si el estudiante es alumno ayudante o no.
otro_tipo_centro_escolar	varchar	Otro tipo de centro escolar.
contacto_localizar	varchar	Contacto a localizar.
telefono_contacto	varchar	Teléfono del contacto.
comparte_habitacion	bool	Indica si el estudiante comparte habitación o no.
persona_comparte	varchar	Persona con la que comparte habitación.
padecimiento_its	bool	Indica si el estudiante padece alguna ITS o no.

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

nombre_its	text	Nombre de la ITS.
participacion_proyecto	bool	Indica si el estudiante participa en proyecto o no.
ha_desempenado_rol	bool	Rol que ha desempeñado.
rol_desempenado	text	Rol desempeñado.
viaingreso_id	int	Identificador de la tabla nm_via_ingreso.
estadocivil_id	int	Identificador de la tabla nm_estado_civil.
tipocentroscolar_id	int	Identificador de la tabla nm_tipo_centro_escolar.

**Tabla 7. Descripción de la tabla “tb\_profesor”**

<b>Nombre:</b> “tb_profesor”		
<b>Tipo:</b> Entidad		
<b>Descripción:</b> almacena los datos correspondientes al profesor.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
asignatura_id	int	Identificador de la tabla nm_asignatura.
es_tutor	bool	Indica si un profesor es tutor o no.
responsabilidad	varchar	Responsabilidad del profesor.

**Tabla 8. Descripción de la tabla “tb\_grupo”**

<b>Nombre:</b> “tb_grupo”		
<b>Tipo:</b> Entidad		
<b>Descripción:</b> almacena los datos correspondientes al grupo.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
numero_grupo	int	Número del grupo.
annoacademico_id	int	Identificador de la tabla nm_anno_academico.
jefebrigada_id	int	Identificador del jefe de brigada.
vicepresidente_id	int	Identificador del vicepresidente.
activistaresidencia_id	int	Identificador del activista de residencia.
activistacultura_id	int	Identificador del activista de cultura.
secretarioujc_id	int	Identificador del secretario de la ujc.
profesorguia_id	int	Identificador del profesor guía.

**Tabla 9. Descripción de la tabla “tb\_persona”**

<b>Nombre:</b> “tb_persona”		
<b>Tipo:</b> Entidad		
<b>Descripción:</b> almacena los datos correspondientes a una persona.		
Atributo	Tipo	Descripción

## CAPÍTULO 2: ANALISIS Y DISEÑO DEL SISTEMA

id	int	Identificador de la tabla.
primer_nombre	varchar	Primer nombre de la persona.
segundo_nombre	varchar	Segundo nombre de la persona.
primer_apellido	varchar	Primer apellido de la persona.
segundo_apellido	varchar	Segundo apellido de la persona.
solapin	varchar	Solapín de la persona.
sexo	varchar	Sexo de la persona.
foto	text	Foto de la persona.
dir_foto	varchar	Dirección de la foto.

### Conclusiones Parciales

Con la realización del presente capítulo se concluye lo siguiente:

- ✓ Se elaboró el modelo conceptual para lograr un mayor entendimiento del dominio del problema, los principales conceptos y sus relaciones.
- ✓ Se definieron todos los requisitos funcionales y no funcionales del sistema, aplicando algunas técnicas de captura y validación de requisitos, los mismos fueron validados satisfactoriamente por el cliente
- ✓ Se realizó el modelo de datos del sistema con el objetivo de que este sea capaz de almacenar toda la información referente al proceso de negocio tratado, mostrándose además las relaciones existentes entre las diferentes tablas de la base de datos.
- ✓ Se realizaron los diagramas de clases de diseño, empleando patrones de diseño para alcanzar un entendimiento mejor y más profundo de cómo se realizará la implementación del sistema.



En este capítulo se realiza una breve descripción de los estándares de código a utilizar durante la implementación del sistema. También se realizar el diagrama de componentes y de despliegue del sistema. Por último, se validará el diseño propuesto a través de las métricas TOC y RC, se describirán las pruebas de caja negra y caja blanca realizadas al software además de las pruebas de aceptación, mostrándose los resultados obtenidos de su aplicación. También se muestra la simulación realizada al proceso Gestión de la caracterización estudiantil en aras de comprobar si se logra su agilización con el empleo del sistema desarrollado.

### 3.1 Estándares de codificación

Los estándares de codificación no son más que una guía que define la estructura y composición sobre la cual se debe regir la codificación de una solución informática. Su objetivo fundamental es lograr un correcto, común y comprensible formato del código que compone un sistema. Este estándar debe servir de apoyo a quien interactúa con dicho código para identificar de forma sencilla cuál es el objetivo y las funcionalidades que brinda cada una de las clases, funciones y demás componentes de software, además que debe servir de guía para posteriores implementaciones y/o modificaciones del sistema.

En general, un estándar de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que otros programadores se les facilite entender el código (como identificar las variables, las funciones o métodos, etc (28).

Se describen a continuación los estándares empleados durante la implementación:

#### **PascalCasing**

El estándar PascalCasing establece que los identificadores, nombres de clases, variables, métodos o funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula. La nomenclatura de las clases se realizó sobre la base de este estándar, usando palabras compuestas sugerentes acordes al propósito de la misma.

#### Nomenclatura de clases según su tipo

TbEstudianteController, TbEstudiante, NmAsignatura.

#### **CamelCasing**

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

El estándar CamelCasing es parecido al PascalCasing con la particularidad de que la letra inicial del identificador no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones y atributos.

### Nomenclatura de las funciones

El identificativo a emplear para las funciones o métodos se escribe con la primera palabra en minúscula utilizando la notación CamelCasing y nombres que deduzcan su propósito. Ejemplo: obtenerEstudiantesConHijos. Los denominadores de las acciones de las clases controladoras tienen la peculiaridad de ir seguidos por la palabra Action. Ejemplo: caracterizacionAction.

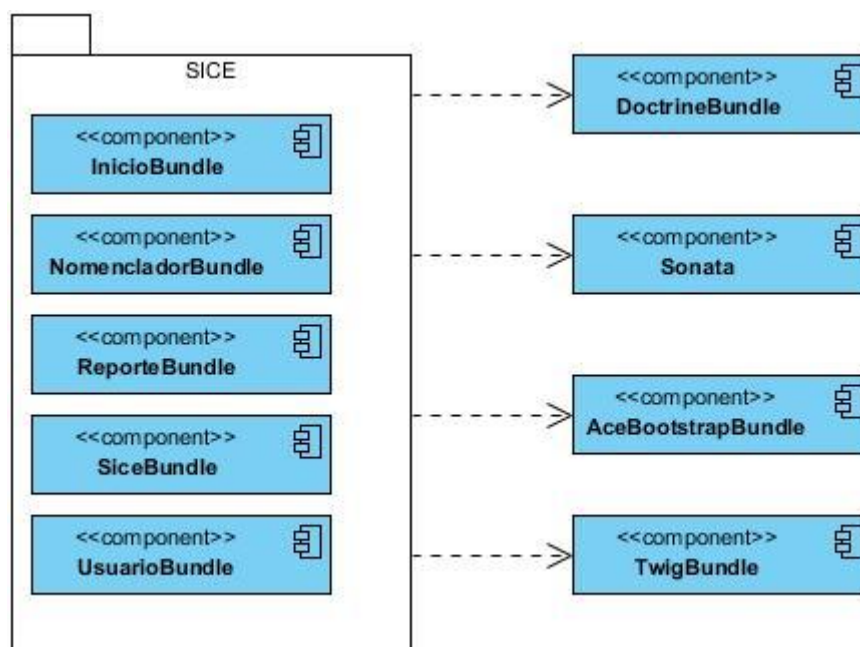
### Nomenclatura de los atributos

Las variables se nombran convenientemente de acuerdo con el estándar CamelCasing. Ejemplo: \$fechaNacimiento, \$telefonoUci.

### 3.3 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre los mismos. 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.

A continuación se muestra el diagrama de componentes en la **Figura 10**.



**Figura 10.** Diagrama de componentes

### 3.2 Diagrama de despliegue

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos. Está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos nodos que son representados. Se definirá una arquitectura cliente-servidor como se muestra en la **Figura 11**.

A continuación se muestra el diagrama de despliegue del sistema en la **Figura 11**:



**Figura 11.** Diagrama de despliegue del sistema.

**PC Cliente:** computadora en la cual la aplicación se ejecutará a través de un navegador, en este caso se debe usar el mozilla firefox.

**Servidor Web:** radica la lógica de negocio de la aplicación. Servidor web Apache 2.2.6, utilizando el lenguaje de programación del lado del servidor PHP 5.

**Servidor de Autenticación:** servidor a través del cual se realiza la autenticación de los usuarios del sistema pertenecientes al dominio uci.cu, en aras de lograr que no accedan al sistema usuarios no autorizados.

**Servidor de base de datos:** servidor de datos PostgreSQL 9.2, donde se encuentra la base de datos que utiliza el sistema, este puede estar instalado en la misma computadora donde se encuentra el servidor web.

**Impresora:** utilizada para imprimir los reportes del sistema en caso de ser necesario.

### 3.4 Validación del diseño propuesto

La validación del diseño se realizó aplicando métricas que permitieron evaluar la calidad del diseño propuesto.

Las métricas contribuirán a la evaluación de los modelos de análisis y de diseño, en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente. Lo cual ayudará en el diseño de pruebas más efectivas. Se tuvieron en cuenta las métricas orientadas a objetos que contienen varios atributos de calidad, dentro de los cuales se encuentran:

- **Responsabilidad:** *consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.*
- **Complejidad de implementación:** *consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.*
- **Reutilización:** *consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.*
- **Acoplamiento:** *consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.*
- **Complejidad del mantenimiento:** *consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.*
- **Cantidad de pruebas:** *consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto diseñado.*

Las métricas concebidas como instrumento para evaluar la calidad del diseño y su relación con los atributos de calidad definidos son las siguientes:

- ✓ **Tamaño Operacional de Clase (TOC):** se refiere al número de métodos pertenecientes a una clase. Está determinada por los atributos: Responsabilidad, Complejidad de implementación y la Reutilización, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado.

**Tabla 10. Tamaño operacional de clase (TOC)**

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

**Tabla 11. Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC**

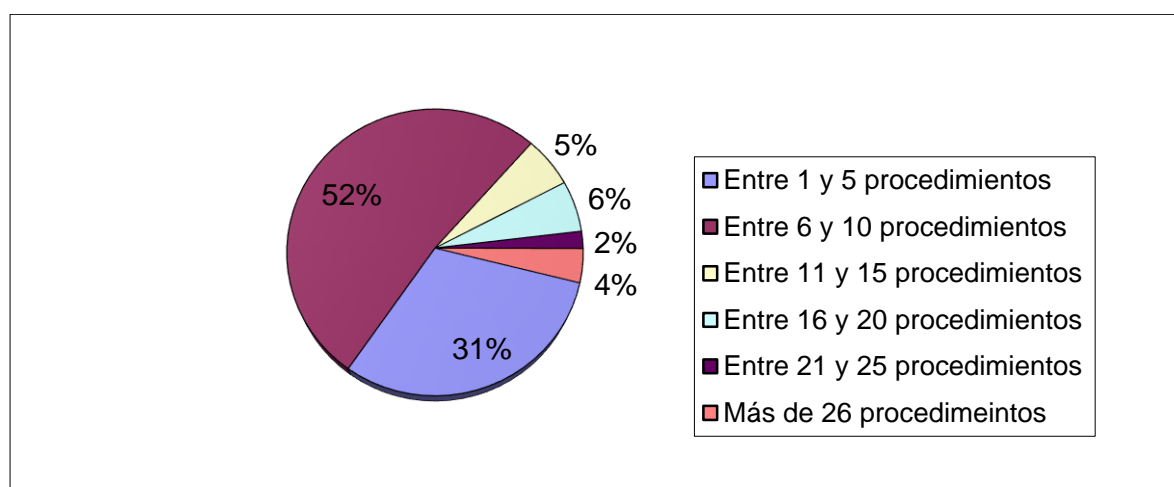
	Categoría	Criterio
Responsabilidad	Baja	$\leq$ Prom. (5.1)
	Media	Entre Prom. Y 2* Prom
	Alta	$>$ 2* Prom
Complejidad de implementación	Baja	$\leq$ Prom
	Media	Entre Prom. y 2* Prom
	Alta	$>$ 2* Prom
Reutilización	Baja	$>$ 2*Prom
	Media	Entre Prom. y 2* Prom
	Alta	$\leq$ Prom

**Tabla. 12 Umbrales para el TOC**

Tamaño operacional de la clase	Criterio
Pequeña	$\leq$ Prom.(5.1)
Media	Entre Prom. y 2 * Prom.
Grande	$>$ 2* Prom.

**Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clase (TOC)**

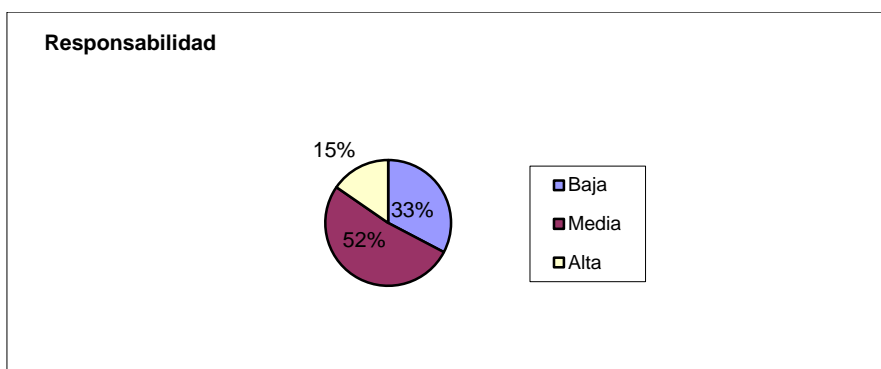
Representación en % de los resultados obtenidos en el instrumento, agrupados en los intervalos definidos, ver **Figura 12**:



**Figura 12.** Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

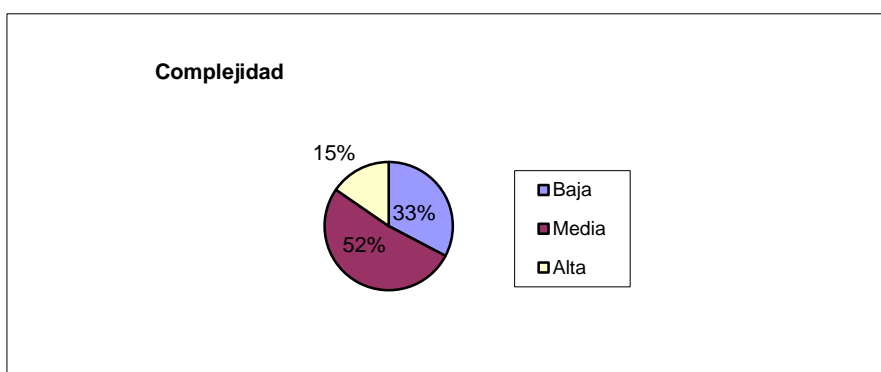
### CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad, ver **Figura 13**:



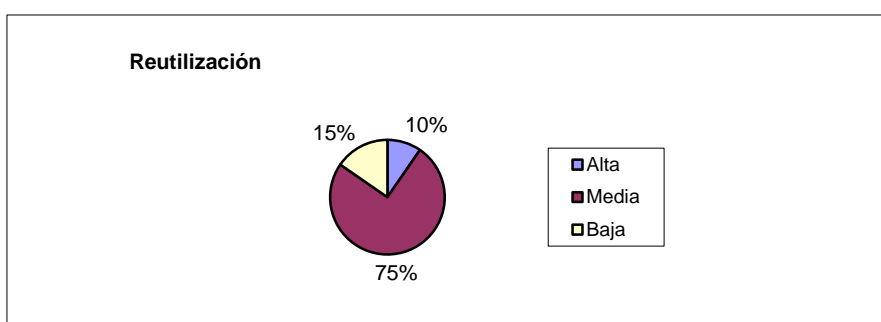
**Figura 13.** Resultado del atributo Responsabilidad.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación, ver **Figura 14**:



**Figura 14.** Resultado del atributo Complejidad de implementación.

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización, ver **Figura 15**:



**Figura 15.** Resultado del atributo Reutilización.

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica TOC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable. Teniendo en cuenta que la mayoría de las clases (83%) cuenta con una cantidad de operaciones entre 1 y 10.

### CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

Los atributos de calidad se encuentran en un nivel aceptable, en el 33% de las clases tienen una baja responsabilidad. En el atributo de Complejidad de implementación el 33% de las clases tienen una baja complejidad. En el atributo de Reutilización el 75% de las clases posee una reutilización media y un 10% alta.

**Relaciones entre Clases (RC):** dado por el número de relaciones de uso de una clase. Está determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Cantidad de pruebas y Reutilización, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado.

**Tabla 13. Atributos de calidad evaluados por la métrica RC**

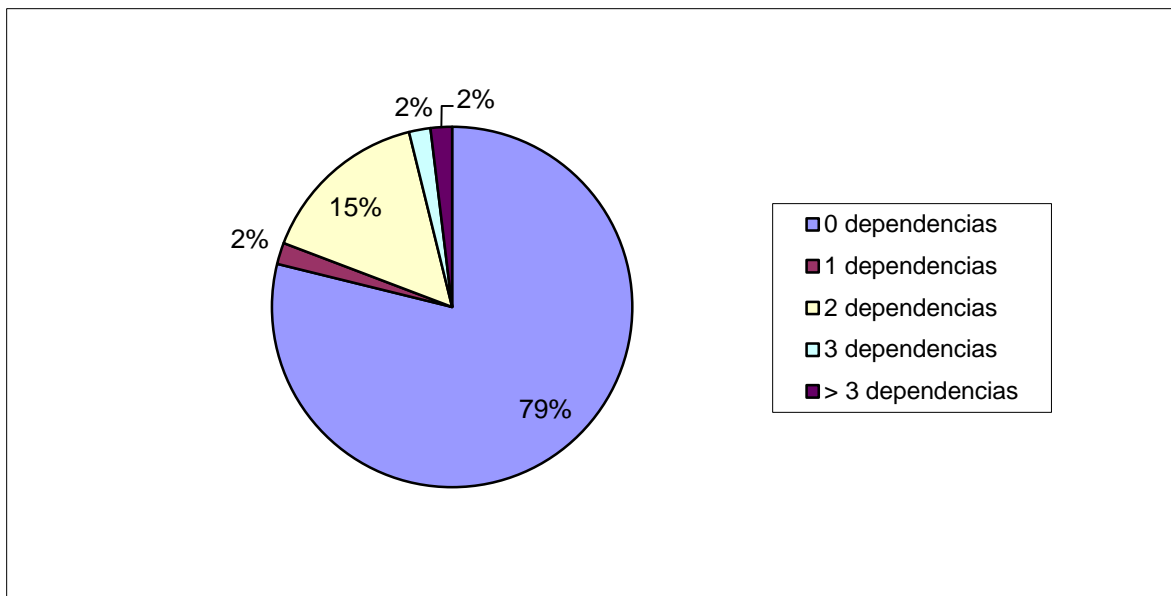
Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

**Tabla 14. Criterios de evaluación para la métrica RC**

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	> 2
Complejidad de mantenimiento	Baja	< =Prom.
	Media	Entre Prom y 2* Prom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom y 2* Prom
	Alta	<= Prom.
Cantidad de pruebas	Baja	< =Prom.
	Media	Entre Prom y 2* Prom.

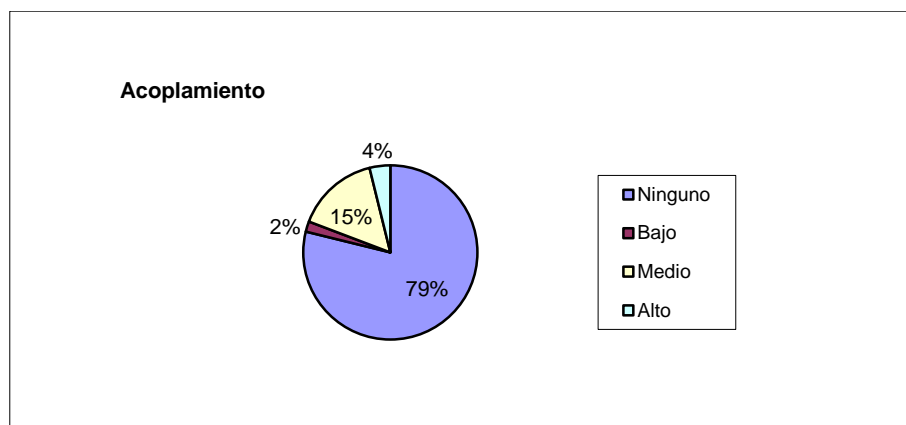
	Alta	> 2* Prom.
--	------	------------

**Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC).** Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos, ver **Figura 16**:



**Figura 16.** Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento, ver **Figura 17**:

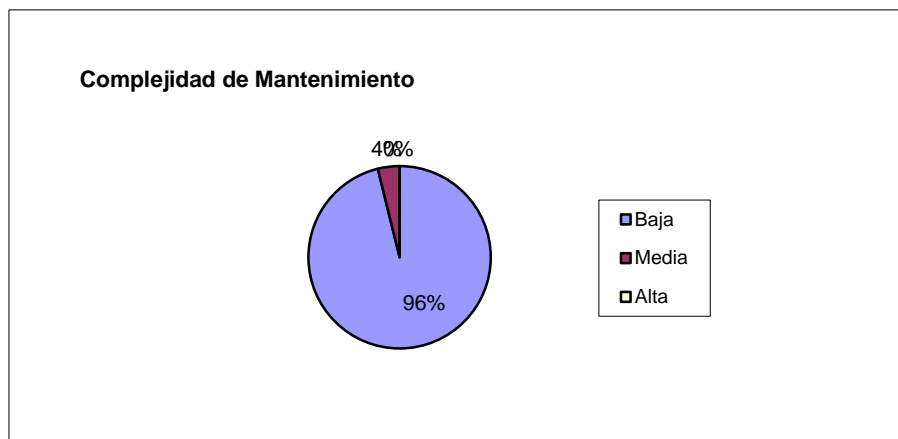


**Figura 17.** Resultado del atributo Acoplamiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento, ver **Figura 18**:

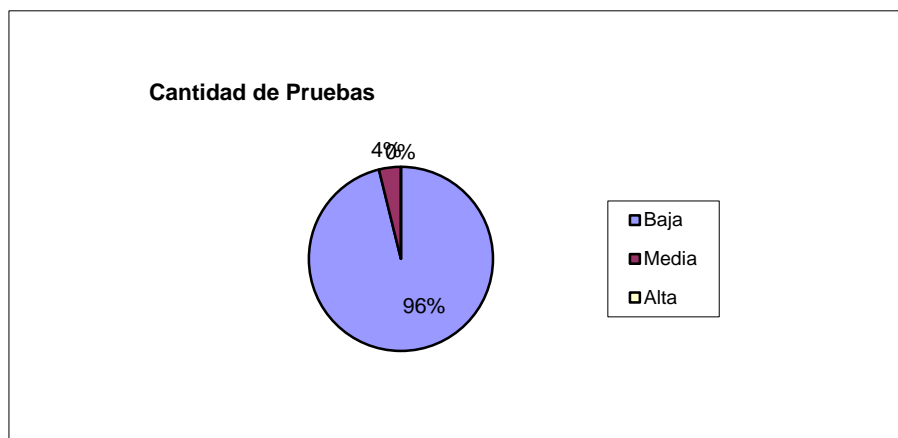


### CAPÍTULO 3: IMPLEMENTACION Y PRUEBA



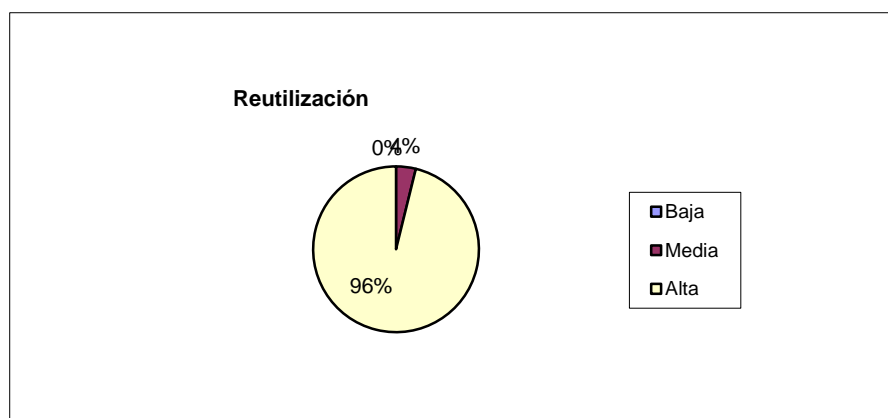
**Figura 18.** Resultado del atributo Complejidad de mantenimiento.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas, ver **Figura 19**:



**Figura 19.** Resultado del atributo atributo Cantidad de pruebas.

Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización, ver **Figura 20**:



**Figura 20.** Resultado del atributo Reutilización.

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, teniendo en cuenta que el 79% de las clases posee cero dependencias. Los atributos de calidad se encuentran en un nivel satisfactorio, en el 79% de las clases el grado de acoplamiento es mínimo, la Complejidad de mantenimiento, la Cantidad de pruebas y la Reutilización se comportan favorablemente para un 96% de las clases.

### 3.5 Validación de la solución propuesta

Se define una estrategia de pruebas regida según el modelo de desarrollo establecido para realizar el proceso de la aplicación de pruebas, con el fin de garantizar la calidad del software. Se concretan los casos de pruebas, a partir de la construcción de todos los posibles caminos de ejecución, o escenarios de cada componente desarrollado, basados en los requisitos implementados, comparando cada funcionalidad descrita con lo implementado para verificar hasta qué punto concuerdan y cumplen con las necesidades del cliente. Se realizan pruebas de aceptación a clientes e interesados.

#### 3.5.1 Pruebas de software

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos (29).

##### 3.5.1.1 Prueba de Caja Blanca

Las pruebas de caja blanca, también conocidas como pruebas de caja transparente o pruebas estructurales, se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Este método intenta garantizar que se ejecutan al menos una vez todos los caminos independientes que presente el módulo y que todas las estructuras de datos internas serán usadas (31).

###### 3.5.1.1.1 Técnicas de caja blanca

De las técnicas de caja blanca, en el trabajo se tuvo en cuenta la de prueba del camino básico, la cual se especifica a continuación:

**Prueba del camino básico:** El método del camino básico (propuesto por McCabe) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez (32).

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

El algoritmo a analizar es el que se presenta a continuación **Figura 21**:

```
public function loadUserByUsername($username) {
    $username = $this->container->get('request')->get('_username');//1
    $password = $this->container->get('request')->get('_password');//1

    $ws = new \SoapClient($this->container->getParameter('uci.webservice'));//1

    $user = $ws->autenticarUsuario($username, $password, 'uci.cu', true);//1

    if ($user->Usuario === $username && $user->Autenticado && $user->Area->NombreArea=='Facultad 3'
        && (in_array($user->Cargo->NombreCargo,array('Estudiante','Profesor','Jefe de Departamento')))) { //2,3,4,5

        $em=$this->container->get('doctrine')->getManager();//6
        $repository=$em->getRepository('ApplicationSonataUserBundle:User');//6
        $usuario=$repository->findOneBy(array('username'=>$username,'credencial'=>(string)$user->Credencial));//6
        if(!$usuario){ //7
            $usuario=$this->createUser($username,$password,$user);//8
        }
        return $usuario; //9
    }

    throw new UsernameNotFoundException(sprintf('Username "%s" does not exist.', $username));//10
}
```

**Figura 21.** Fragmento de código

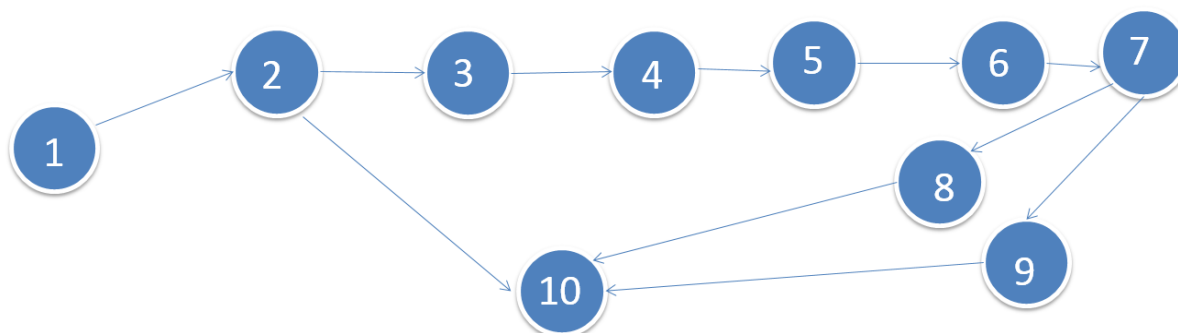
Después de este paso, es necesario representar el grafo de flujo asociado, en el cual se representan distintos componentes como es el caso de:

**Nodo:** son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

**Aristas:** son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

**Regiones:** son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

A continuación se muestra el grafo de flujo en la **Figura 22**.



**Figura 22.** Grafo de flujo

**Fórmulas para calcular la complejidad ciclomática:**

**1.  $V(G) = (A - N) + 2$**

Donde “A” es la cantidad de aristas y “N” la cantidad de nodos.

$V(G) = (11 - 10) + 2$

$V(G) = 3$

**2.  $V(G) = P + 1$**

Siendo “P” la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$V(G) = 2 + 1$

$V(G) = 3$

**3.  $V(G) = R$**

Donde “R” representa la cantidad de regiones en el grafo.

$V(G) = 3$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de 3, lo que significa que existen tres posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

**Tabla 15. Caminos básicos**

Número	Caminos básicos
1	1-2-10
2	1-2-3-4-5-6-7-8-10
3	1-2-3-4-5-6-7-9-10

**3.5.1.2 Prueba de Caja Negra**

Las pruebas que se le realizan al sistema para la validación de sus funcionalidades son pruebas de caja negra, estas pruebas se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del sistema.

Los casos de prueba de caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

A continuación se derivan conjuntos de condiciones de entrada que utilizan todos los requisitos funcionales de una aplicación.

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

Las pruebas de caja negra pretenden encontrar errores como:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en la estructura de datos o en accesos a base de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación (32).

### 3.5.1.2.1 Técnicas de caja negra

En el trabajo se aplican los diseños de casos de pruebas como técnicas de caja negra. A continuación se explica en qué consisten estos y se muestra un caso de prueba para un requisito funcional. Casos de pruebas: un caso de prueba es una especificación, usualmente formal, de un conjunto de, entradas de prueba condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de aspectos particulares de un elemento objeto de prueba.

**Tabla 16. Descripción de requisito**

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Añadir problema.	Se añade un nuevo problema.	EP 1.1: Añadir un nuevo problema.	<ul style="list-style-type: none"><li>• Se presiona el botón Añadir problema.</li><li>• Se introducen los datos del problema.</li><li>• Se presiona el botón Crear y regresar al listado de la ventana y esta se cierra mostrando el mensaje de información. Se cierra el mensaje.</li></ul>
		EP1.2: Añadir un nuevo problema dejando campos requeridos en blanco.	<ul style="list-style-type: none"><li>• Se presiona el botón Añadir problema.</li><li>• Se introducen los datos dejando campos requeridos en blanco.</li><li>• Se presiona el botón Crear y regresar al listado. El sistema muestra un mensaje de error y señala los campos requeridos en blanco. Se cierra el mensaje.</li></ul>
		EP1.3: Añadir un nuevo problema introduciendo errores en los datos.	<ul style="list-style-type: none"><li>• Se presiona el botón Añadir problema.</li><li>• Se introducen los datos del problema introduciendo errores en los mismos.</li><li>• El sistema señala los campos con errores.</li></ul>

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

El sistema fue sometido a dos iteraciones de pruebas de caja negra. En la primera iteración se detectaron 35 no conformidades, de las cuales 29 fueron de aplicación y 6 de documentación. Dentro de las no conformidades referentes a las de aplicación se encuentran 16 de validación, 12 de interfaz y 1 de funcionalidad. En cuanto a las de documentación se encontraron 2 no conformidades de redacción, 1 de ortografía y 3 de correspondencia. Todas estas no conformidades fueron resueltas seguidamente de haberlas detectado, lo que permitió que el componente pasara a una segunda iteración en la que no se detectaron no conformidades, obteniendo resultados satisfactorios.

### **3.5.1.3 Pruebas de Aceptación/Calificación**

Hace una comparación entre el comportamiento del sistema con los requisitos del cliente. El cliente se encarga de realizar, o especificar, las tareas típicas para verificar que se cumplen sus requisitos o que el grupo de trabajo los ha identificado para el mercado adonde se va a dirigir el software. En estas pruebas se puede incluir o no a los desarrolladores del sistema (30).

A continuación se muestra en la **Figura 23** el Acta de aceptación como evidencia de las pruebas de aceptación realizadas al cliente

**ACTA DE ACEPTACIÓN**

Por el presente documento damos constancia que la aplicación obtenida para la gestión de la caracterización estudiantil (Sistema integral para la caracterización estudiantil - SICE) cumple con las funcionalidades necesarias que posibilitan obtener con mayor facilidad informaciones de los estudiantes de la Facultad 3.

Se empleó para el desarrollo de la aplicación tecnologías libres contribuyendo con el proceso de independencia tecnológica por el que se está abogando en la UCI.

Se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

- *Modelo conceptual*
- *Diagrama de procesos de negocio*
- *Especificaciones de Requisitos de Software*
- *Modelo de Datos*
- *Diagramas de clases del diseño*
- *Diagramas de secuencias*
- *Diagrama de componentes*
- *Diagrama de despliegue*
- *Ficheros de código fuente*
- *Diseños de casos de pruebas*

Y para constancia de ello se firma el presente aval a los **11** del mes de junio del 2014.

Entrega:	Recibe:	Firma:
Andrés Barbán Borrero	Ing. Carlos Abel Capeans Hurtado	<i>Capue</i>
	Ing. Dayannis Estrada Duarte	<i>Dayannis</i>
	MsC. Elizabeth Rodríguez Stiven	<i>Beth</i>
	MsC. Pedro Manuel Nogales	<i>Manuel</i>
	Ing. Berenice Guevara Delgado	<i>Berenice</i>



**Figura 23.** Acta de aceptación

**3.5.2 Simulación del proceso de Gestión de la caracterización estudiantil de la Facultad 3.**

Con objetivo de validar la investigación se realizó la simulación del proceso de Gestión de la caracterización estudiantil, analizando si el tiempo en el desarrollo de las actividades

### CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

comprendidas en dicho proceso disminuye haciendo uso del sistema obtenido. Se tomó como promedio un grupo de 20 estudiantes para el desarrollo de las actividades.

**Tabla 17: Actividades del proceso de Gestión de la caracterización estudiantil de la Facultad 3**

<b>Actividades del proceso de gestión de la caracterización estudiantil</b>				
No	<b>ACTIVIDADES</b>		Antes (min)	Después (min)
	<b>Sin el sistema</b>	<b>Con el sistema</b>		
1	Solicitar caracterización	Solicitar caracterización	2.35	2.05
2	Solicitar llenado de caracterización	Solicitar llenado de caracterización	8.15	7.05
3	Llenar caracterización	Llenar caracterización (o actualizar en caso que haya introducido datos en años anteriores)	7	6
4	Entregar caracterización	No se hace	1.05	--
5	Revisar caracterización	Revisar caracterización	41.55	40
6	Indicar corregir caracterización	Indicar corregir caracterización	igual	igual
7	Corregir caracterización	Actualizar caracterización	igual	igual
8	Procesar datos para caracterización del grupo	Mostrar reporte por criterio(s) de búsqueda	197.25	28
9	Identificar problemas del grupo	Identificar problemas del grupo	igual	igual
10	Definir actividades para Proyecto Educativo	Definir actividades para Proyecto Educativo	igual	igual
11	Entregar Proyecto Educativo	No se hace	1.05	--
12	Recibir proyecto educativo	No se hace	0.25	--
	<b>TOTAL</b>		<b>258.65</b>	<b>83.1</b>

Como se puede apreciar en la tabla con el uso del sistema se logra agilizar el proceso de Gestión de la caracterización estudiantil de la Facultad 3, teniendo en cuenta que el tiempo total disminuye en aproximadamente 175.5 minutos.



## CAPÍTULO 3: IMPLEMENTACION Y PRUEBA

### **Conclusiones Parciales**

Una vez finalizado el capítulo se concluye que:

- Los resultados arrojados por las métricas TOC y RC permiten valorar principalmente la complejidad en la implementación de la solución que se presenta con la actual investigación.
- La realización de las pruebas de caja blanca y caja negra realizadas al sistema, permitieron verificar su adecuado funcionamiento.
- Los resultados obtenidos por la simulación demostró una mayor agilización del proceso, como resultado de la utilización del sistema.
- Las pruebas de aceptación realizadas hasta el momento por el colectivo de asesores de año han permitido verificar y validar los requisitos identificados, arrojando que la solución propuesta proporciona un avance en la ejecución del proceso de gestión de la caracterización estudiantil.

Una vez terminado el presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- ✓ El análisis de sistemas informáticos, tanto nacionales como internacionales, vinculados a la gestión de la información estudiantil evidencian la no existencia de una solución informática capaz de cubrir todas las funcionalidades requeridas para realizar una adecuada gestión de la caracterización estudiantil.
- ✓ El uso de las técnicas para la captura de los requisitos permitió identificar los requisitos funcionales del sistema; los cuales fueron validados satisfactoriamente por los clientes para su posterior desarrollo.
- ✓ El análisis y el diseño del Sistema de gestión para la caracterización estudiantil de la Facultad 3, permitió generar un conjunto de artefactos que sirvieron de base para la implementación de los requisitos definidos.
- ✓ Las pruebas desarrolladas permitieron encontrar y corregir los errores no detectados durante la implementación posibilitando cumplir con las especificaciones requeridas y la validación del sistema implementado.

## RECOMENDACIONES

### RECOMENDACIONES

Teniendo como base los resultados de este trabajo y la experiencia adquirida durante el desarrollo del mismo, se recomienda:

- ✓ Que el sistema se haga extensivo a toda la comunidad universitaria.
- ✓ Aumentar el factor de reutilización del tamaño operacional de las clases.
- ✓ Integrar la solución al sistema DATAFEU para obtener más información de los estudiantes.

1. Palacio, Carlos Augusto Muriel. Renovación Tecnológica y Literatura. *Renovación Tecnológica y Literatura*. [En línea] 12 de Enero de 2014. [Citado el: 25 de Enero de 2014.] [http://12357carlosaugusto.blogspot.com/2014/01/que-es-informacion\\_12.html](http://12357carlosaugusto.blogspot.com/2014/01/que-es-informacion_12.html).
2. Benítez, Lic. Idarmis González. Scielo. *Scielo*. [En línea] 6 de Enero de 2000. [Citado el: 26 de Enero de 2014.] [http://scielo.sld.cu/scielo.php?pid=S0864-21412000000100008&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S0864-21412000000100008&script=sci_arttext).
3. Barroso. *Propuesta de pautas para el diseño de un Sistema de Gestión de Información en la empresa ECIMETAL. Tesis de Licenciatura*. Habana : Departamento de Bibliotecología y Ciencia de la Información, Universidad de La Habana, 2009.
4. *Revista Pedagogía Universitaria*. Silva, Dr. C. Pedro Horruitiner. 4, Habana : s.n., 2007, Vol. XII.
5. Cespi. *Cespi*. [En línea] [Citado el: 1 de Febrero de 2014.] <http://www.cespi.unlp.edu.ar/kimkelen>.
6. Servicio de Informática de la Universidad de Córdoba. *Servicio de Informática de la Universidad de Córdoba*. [En línea] [www.gestion.uco.es/gestion/aplicaciones/siga](http://www.gestion.uco.es/gestion/aplicaciones/siga).
7. Sistema de Gestión Escolar. *Sistema de Gestión Escolar*. [En línea] [Citado el: 1 de Febrero de 2014.] <http://www.rutatec.com/pagina/index.php/productos/servicios/sgescolar>.
8. Info@tletas. *Info@tletas*. [En línea] [Citado el: 2 de Febrero de 2014.] <http://www.inder.cu/FichaAtleta/documentacion/index.html>.
9. LibrosWeb. *LibrosWeb*. [En línea] 2014. [Citado el: 5 de Febrero de 2014.] [http://librosweb.es/javascript/capitulo\\_1.html](http://librosweb.es/javascript/capitulo_1.html).
10. Ecured. *Ecured*. [En línea] [Citado el: 7 de Febrero de 2014.] <http://www.ecured.cu/index.php/PHP>.
11. Epidata. *Epidata*. [En línea] [Citado el: 10 de Febrero de 2014.] [http://www.epidataconsulting.com/tikiwiki/tiki-read\\_article.php?articleId=15](http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15).
12. Symfony para todos. *Symfony para todos*. [En línea] [Citado el: 15 de Febrero de 2014.] <http://symfony.cubava.cu/introduccion/symfony-2/>.
13. Introducción — Doctrine 2. *Introducción — Doctrine 2*. [En línea] [Citado el: 20 de Febrero de 2014.] <http://sf2-es.net16.net/doctrine/orm-documentation/reference/introduction.html>.
14. JQuery. *JQuery*. [En línea] [Citado el: 26 de Febrero de 2014.] <http://jquery.com/>.
15. Postgresql-es. *Postgresql-es*. [En línea] [Citado el: 1 de Abril de 2014.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
16. Softonic. *Softonic*. [En línea] [Citado el: 2 de Abril de 2014.] <http://apache-http-server.softonic.com>.

## REFERENCIAS

17. Mis respuestas.com. *Mis respuestas.com*. [En línea] [Citado el: 10 de Abril de 2014.] <http://www.misrespuestas.com/que-es-firefox.html>.
18. Software.com.ar. *Software.com.ar*. [En línea] [Citado el: 15 de Abril de 2014.] <http://www.software.com.ar/visual-paradigm-para-uml.html>.
19. Genbeta. *Genbeta*. [En línea] [Citado el: 16 de Abril de 2014.] <http://www.genbetadev.com/herramientas/netbeans-1>.
20. IngenieroSoftware. *IngenieroSoftware*. [En línea] [Citado el: 16 de Abril de 2014.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
21. Entidades, Centro de Informatización de Gestión de. *CEIGE-Modelo de Desarrollo de Software v1*. Habana : s.n., 2014.
22. ClubEnsayos. *ClubEnsayos*. [En línea] [Citado el: 17 de Abril de 2014.] <http://clubensayos.com/Tecnolog%C3%ADa/T%C3%A9cnicas-De-Obtenci%C3%B3n-De-Requerimientos/269660.html>.
23. Procuraduría. *Procuraduría*. [En línea] [Citado el: 25 de Abril de 2014.] [http://www.procuraduria.gov.co/infosim/media/file/VERSIONES\\_EN\\_PDF/Etapa4-ReqNoFunc.pdf](http://www.procuraduria.gov.co/infosim/media/file/VERSIONES_EN_PDF/Etapa4-ReqNoFunc.pdf).
24. [En línea] [Citado el: 30 de Abril de 2014.] [http://www.google.com/cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fsi.ua.es%2Fes%2Fdocumentacion%2Fasp-net-mvc-3%2F1-dia%2Fmodelo-vista-controlador-mvc\\_pdf.pdf&ei=MRGRU8WGD6HgsATXhIKoDg&usg=AFQjCNH1EMdQTIIm8hlw6Ra5HokXBs4AnYg&](http://www.google.com/cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fsi.ua.es%2Fes%2Fdocumentacion%2Fasp-net-mvc-3%2F1-dia%2Fmodelo-vista-controlador-mvc_pdf.pdf&ei=MRGRU8WGD6HgsATXhIKoDg&usg=AFQjCNH1EMdQTIIm8hlw6Ra5HokXBs4AnYg&).
25. Diseño UML. *Diseño UML*. [En línea] 22 de Mayo de 2009. [Citado el: 16 de Abril de 2014.] <http://egdamar877.blogspot.com/2009/05/expocicion.html>.
26. Microsoft. *Microsoft*. [En línea] [Citado el: 21 de Abril de 2014.] <http://msdn.microsoft.com/es-es/library/dd409377.aspx>.
27. Definición.de. *Definición.de*. [En línea] [Citado el: 27 de Abril de 2014.] <http://definicion.de/modelo-de-datos/>.
28. Yahoo. *Yahoo*. [En línea] [Citado el: 28 de Abril de 2014.] <https://mx.answers.yahoo.com/question/index?qid=20090122200540AAOI1N8>.
29. Pruebas de Software. *Pruebas de Software*. [En línea] [Citado el: 1 de Mayo de 2014.] <http://www.pruebasdesoftware.com/laspruebasdesoftware.htm>.
30. Swebok. *Swebok*. [En línea] [Citado el: 2 de Mayo de 2014.] [ftp://ftpdcente.uniautonoma.edu.co/mnieto/Ingenieria\\_software\\_I/SWEBOK/SWEBOK%20html%20uniautonoma/Cap%204-5/prueba\\_niveles.html](ftp://ftpdcente.uniautonoma.edu.co/mnieto/Ingenieria_software_I/SWEBOK/SWEBOK%20html%20uniautonoma/Cap%204-5/prueba_niveles.html).
31. Pressman. Un enfoque práctico. *Ingeniería de software*. 2010.
32. [En línea] [Citado el: 3 de Mayo de 2014.] [https://docs.google.com/viewer?a=v&q=cache:L0\\_moMWzYTYJ:indalog.ual.es/mtorres/LP/P](https://docs.google.com/viewer?a=v&q=cache:L0_moMWzYTYJ:indalog.ual.es/mtorres/LP/P)

## REFERENCIAS

rueba.pdf+pruebas+de+caja+negra&hl=es&gl=cu&pid=bl&srcid=ADGEEShyg3ux79zqPpID1  
O92UU\_M3GdGAlm2GwqSjSr2t\_-  
ZPK92JW5\_hBCcumq1FcxUBeqrpgyuQ6QzRHEujVuzZ1T961tR-Omog9-fZtsh1JqlqFE.

33. CECES. *CECES*. [En línea] [Citado el: 30 de Enero de 2014.]  
[http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Revistas/Revista%20Pedagogia%20Universitaria/A\\_o%201998/1998-3/189498304.pdf](http://ftp.ceces.upr.edu.cu/centro/repositorio/Textuales/Revistas/Revista%20Pedagogia%20Universitaria/A_o%201998/1998-3/189498304.pdf).

