

**Universidad de las Ciencias Informáticas
Facultad 10**



**Título: Módulo de gestión de asignaturas de la herramienta
educativa SMProg para el EVA - UCI**

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

Autores: Eduardo Failde Fernández
Alberto Gil Hernández

Tutores: Lic. Dunia Suárez Ferreiro
Ing. Alexander Hernández Chapman

Ciudad de La Habana.
Abril de 2010.

Declaración de Autoría.

Declaramos ser autores de la Tesis **Módulo de gestión de asignaturas de la herramienta educativa SMProg para el EVA - UCI** y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Eduardo Failde Fernández
Firma del autor

Alberto Gil Hernández
Firma del autor

Lic. Dunia Suárez Ferreiro
Firma del tutor

Ing. Alexander Hernández Chapman
Firma del tutor

Resumen

Desde la puesta en funcionamiento del Entorno Virtual de Aprendizaje (EVA) en la Universidad de las Ciencias Informáticas (UCI), las asignaturas y contenidos de formación de la carrera del Ingeniero en Ciencias Informáticas, se gestionan a través de dicho entorno. Este proceso se ejecuta a través de módulos, applets, así como otros complementos que conjuntamente llevan a cabo dicha tarea. En un futuro se incorporará al EVA un nuevo módulo denominado SMProg (Software de Motivación a la Programación), el cual se encargará de evaluar aspectos de la asignatura de Programación en todos sus niveles, a través de juegos didácticos, facilitando de esta forma el proceso de interacción de los estudiantes con el entorno.

Uno de los módulos necesarios para el funcionamiento de SMProg es el que realizará la gestión de asignaturas, denominado Virtual_Know, este es una herramienta que realizará dicha gestión a través de la manipulación de uno o varios usuarios, en este caso profesores, quienes conjuntamente con una base de datos propia del sistema, y a través de una interfaz de usuario, llevarán a cabo todas las acciones deseadas, ya sea insertar, modificar, eliminar etc. cualquier materia de clase, que posteriormente se desee evaluar una vez que haya sido creado el juego.

El principal objetivo de la presente investigación es implementar la herramienta Virtual_Know sobre Software Libre, la que permitirá la gestión de asignaturas para el módulo de juegos didácticos SMProg, el cual está asociado a un proyecto de innovación pedagógica de la Universidad de las Ciencias Informáticas.

Índice de Contenidos

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	7
1.1 HERRAMIENTAS PARA LA GESTIÓN DEL CONOCIMIENTO	7
1.2 METODOLOGÍA DE DESARROLLO, LENGUAJE DE MODELADO Y HERRAMIENTAS CASE.	12
1.2.1. <i>Proceso Unificado de Desarrollo de Software</i>	13
<i>Características de la Metodología RUP</i>	15
1.2.2 <i>eXtreme Programming</i>	15
1.2.3. <i>Lenguaje Unificado de Modelado</i>	17
1.3 LENGUAJES DE PROGRAMACIÓN, TENDENCIAS ACTUALES Y HERRAMIENTAS DE DESARROLLO.	19
1.3.1. <i>Programación Orientada a Objetos</i>	19
1.3.2. <i>Lenguaje de Programación Java</i>	20
1.3.2.1. Razones por la cual se escoge Java para la implementación del sistema.....	21
1.3.3. <i>Sistema gestor de Bases de Datos Relacionales Orientado a Objetos PostgreSQL</i>	21
1.3.4 <i>Herramientas de Desarrollo</i>	24
1.4.1 <i>Estilos arquitectónicos</i>	26
CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA.....	29
2.1 MODELO DE DOMINIO.....	29
2.2 LEVANTAMIENTO DE REQUISITOS	31
2.2.1 <i>Requisitos funcionales del sistema</i>	31
2.2.2 <i>Requisitos no funcionales del sistema</i>	32
2.3 MODELO DE CASOS DE USO.....	34
2.3.1 <i>Actores del Sistema</i>	35
2.3.2 <i>Casos de Uso del Sistema</i>	35
2.3.3 <i>Descripción de los Casos de Uso del Sistema</i>	38
CAPÍTULO 3: CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA.....	64
3.1 CARACTERÍSTICAS FUNDAMENTALES DEL ANÁLISIS.....	64
3.1.1 <i>Diagrama de clases del análisis</i>	65
3.2 CARACTERÍSTICAS FUNDAMENTALES DEL DISEÑO.	68
3.2.1 <i>Diagrama de Interacción</i>	68
3.2.3 <i>Descripción de las clases</i>	76
3.3 MODELO ENTIDAD-RELACIÓN	87
CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	96
4.2 DIAGRAMA DE COMPONENTES	96
4.1 DIAGRAMA DE DESPLIEGUE.....	98
4.3 PRUEBAS.....	99
CONCLUSIONES	108
RECOMENDACIONES.....	109
BIBLIOGRAFÍA.....	110

REFERENCIAS BIBLIOGRÁFICAS	111
ANEXOS.....	112
GLOSARIO DE TÉRMINOS	126

Índice de Figuras

FIGURA NO. 1: MODELO DE DOMINIO	30
FIGURA NO. 2: DIAGRAMA DE CASOS DE USO DEL SISTEMA	34
FIGURA NO. 3: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - AUTENTICAR USUARIO	65
FIGURA NO. 4: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - GESTIONAR ASIGNATURA	66
FIGURA NO. 5: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - GESTIONAR TEMA	66
FIGURA NO. 6: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - GESTIONAR PREGUNTA	66
FIGURA NO. 7: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - GESTIONAR RESPUESTA.....	67
FIGURA NO. 8: DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU - CONSULTAR INFORMACIÓN	67
FIGURA NO. 9: DIAGRAMA DE COLABORACIÓN DEL CU - AUTENTICAR USUARIO	69
FIGURA NO. 10: DIAGRAMA DE COLABORACIÓN DEL CU - GESTIONAR ASIGNATURA	69
FIGURA NO. 11: DIAGRAMA DE COLABORACIÓN DEL CU - GESTIONAR TEMA	70
FIGURA NO. 12: DIAGRAMA DE COLABORACIÓN DEL CU - GESTIONAR PREGUNTA	71
FIGURA NO. 13: DIAGRAMA DE COLABORACIÓN DEL CU - GESTIONAR RESPUESTA.....	72
FIGURA NO. 14: DIAGRAMA DE COLABORACIÓN DEL CU - CONSULTAR INFORMACIÓN.....	73
FIGURA NO. 15: DIAGRAMA DE CLASES DEL DISEÑO	75
FIGURA NO. 16: MODELO ENTIDAD-RELACIÓN DE LA BASE DE DATOS DEL SISTEMA.....	89
FIGURA NO. 17: DIAGRAMA DE COMPONENTES DE LA APLICACIÓN VIRTUAL_KNOW.....	97
FIGURA NO. 18: DIAGRAMA DE DESPLIEGUE ÓPTIMO.	99
FIGURA NO. 19: DIAGRAMA DE DESPLIEGUE ACTUAL.....	99

TABLA No. 1: ACTOR DEL SISTEMA.....	35
TABLA No. 2: CASO DE USO AUTENTICAR USUARIO	35
TABLA No. 3: CASO DE USO GESTIONAR ASIGNATURA.....	36
TABLA No. 4: CASO DE USO GESTIONAR TEMA.....	36
TABLA No. 5: CASO DE USO GESTIONAR PREGUNTAS.....	36
TABLA No. 6: CASO DE USO GESTIONAR RESPUESTA	37
TABLA No. 7: CASO DE USO CONSULTAR INFORMACIÓN.....	37
TABLA No. 8: CASO DE USO 1- AUTENTICAR USUARIO	38
TABLA No. 9: CASO DE USO 2-GESTIONAR ASIGNATURA.....	39
TABLA No. 10: CASO DE USO 3- GESTIONAR TEMA.....	43
TABLA No. 11: CASO DE USO 4- GESTIONAR PREGUNTA.....	47
TABLA No. 12: CASO DE USO 5- GESTIONAR RESPUESTA	51
TABLA No. 13: CASO DE USO 6- CONSULTAR INFORMACIÓN	57
TABLA No. 14: DESCRIPCIÓN DE LA CLASE ASIGNATURA.....	76
TABLA No. 15: DESCRIPCIÓN DE LA CLASE TEMA.....	76
TABLA No. 16: DESCRIPCIÓN DE LA CLASE RESPUESTA	77
TABLA No. 17: DESCRIPCIÓN DE LA CLASE SISTEMA_GESTOR	78
TABLA No. 18: DESCRIPCIÓN DE LA CLASE CONSULTA.....	81
TABLA No. 19: DESCRIPCIÓN DE LA CLASE MAIN.....	84
TABLA No. 20: DESCRIPCIÓN DE LA CLASE PANTALLACARGANDOMAIN.....	84
TABLA No. 21: DESCRIPCIÓN DE LA CLASE PANTALLACARGANDOMAIN.....	85
TABLA No. 22: DESCRIPCIÓN DE LA CLASE CONECTARSE.....	87
TABLA No. 23: DESCRIPCIÓN DE LA TABLA: ASIGNATURA	90
TABLA No. 24: DESCRIPCIÓN DE LA TABLA: TEMA	90

Índice de Tablas

TABLA No. 25: DESCRIPCIÓN DE LA TABLA: COMPLEJIDAD	91
TABLA No. 26: DESCRIPCIÓN DE LA TABLA: EVALUACIÓN	92
TABLA No. 27: DESCRIPCIÓN DE LA TABLA: PREGUNTA	92
TABLA No. 28: DESCRIPCIÓN DE LA TABLA: RESPUESTA.....	93
TABLA No. 29: DESCRIPCIÓN DE LA TABLA: SISTEMA_PREGUNTA.....	94
TABLA No. 30: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “AUTENTICAR USUARIO”	100
TABLA No. 31: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR ASIGNATURAS” ESCENARIO “INSERTAR ASIGNATURA”	101
TABLA No. 32: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR ASIGNATURAS” ESCENARIO “MODIFICAR ASIGNATURA”.....	102
TABLA No. 33: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR ASIGNATURAS” ESCENARIO “ELIMINAR ASIGNATURA”.....	103
TABLA No. 34: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR TEMAS” ESCENARIO “INSERTAR TEMA”.	104
TABLA No. 35: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR TEMAS” ESCENARIO “MODIFICAR TEMA”. ...	104
TABLA No. 36: PRUEBAS DE CAJA NEGRA DEL CASO DE USO “GESTIONAR TEMAS” ESCENARIO “ELIMINAR TEMA”.....	106

Introducción

Con el nacimiento y creciente auge de la red de redes, Internet, las Tecnologías de la Informática y las Comunicaciones (TIC) han tomado un alza en la preferencia de los internautas, lo que conlleva entonces al surgimiento de espacios virtuales o sitios online donde, aparte de comunicarse, las personas intercambian entre sí no solo conocimientos sino información de cualquier tipo a través de todo el planeta. Estos espacios han comenzado a jugar un papel trascendental en todos los niveles de la educación, pero fundamentalmente en las universidades es donde se han venido implantando ya que se han creado plataformas virtuales de conocimiento, las cuales son utilizadas con el fin de transmitir e intercambiar informaciones de interés útil a los profesores y estudiantes que interactúan con estas plataformas con el objetivo de mejorar el Proceso de Enseñanza-Aprendizaje (PEA) y promover el interés por el estudio individual entre los estudiantes.

Muchos centros educacionales del mundo y principalmente de Cuba han comenzado a hacer uso de esta variante de aprendizaje como aprovechamiento de las nuevas tecnologías puestas en función de la calidad del PEA, facilitando así la interacción estudiante-profesor y de ambas partes con el medio virtual. Estos entornos son conocidos como EVA cuyo propósito es la enseñanza online, ya que en ellos se pueden publicar todos los contenidos de una o varias asignaturas para luego ser brindados a los estudiantes, de forma que cada vez que estos lo necesiten estén disponibles, además se pueden realizar cuestionarios de evaluación a través de la red sin la necesidad de la presencia del profesor, es por esto que tienen un impacto tan fuerte en los centros educacionales donde se utilizan. Estos entornos son administrados por los profesores, ya que ellos son los que deciden qué materia es la que será publicada, elaboran los cuestionarios, llevan el registro de las evaluaciones de los estudiantes entre otras precisiones.

Por su parte los estudiantes pueden acceder a lo publicado en la plataforma, consultar bibliografías, interactuar con la mensajería instantánea para discutir cualquier tema de una determinada materia, ser evaluados, entre otros aspectos. Ejemplos de Entornos Virtuales de Aprendizaje en el mundo hay muchos pero uno de los más utilizados es el Moodle (*Modular Object-Oriented Dynamic Learning Environment*) el cual está diseñado específicamente para vincularse a la formación de los estudiantes presentando todas las características deseables para un entorno que solo estará dedicado a esta rama social.

En la Universidad de las Ciencias Informáticas existe una infraestructura creada para la teleformación a partir del EVA Moodle como alternativa educativa desde el año 2005, desde su puesta en funcionamiento ha brindado una excelente opción de estudio, constituyendo este un medio donde se recopilan y exponen a la vez todas las materias de los cursos agrupadas por año, además de brindar informaciones y realizar otras operaciones que permitan mejorar la calidad de las clases, también permite la realización de evaluaciones las cuales ahorran recursos por su condición de online. La plataforma Moodle también presenta módulos que han facilitado toda una variedad de opciones entre las que se encuentran: los cuestionarios evaluativos online, los foros de discusión, los registros de evaluaciones entre otras.

Uno de estos módulos está en fase de desarrollo en la UCI, su nombre es SMProg, el cual se encargará de evaluar a través de juegos didácticos, los conocimientos adquiridos por parte de los estudiantes en la asignatura de Programación, ya que los temas o conocimientos que serán usados en estos juegos se extraerán de una base de datos donde previamente hayan sido almacenados. Todavía SMProg no cuenta con una herramienta de gestión de asignaturas la cual se encargue de insertar, eliminar, modificar, así como consultar todas las materias que se utilizarán una vez que dicho módulo esté concluido, dicha herramienta servirá como complemento para el buen funcionamiento de los juegos didácticos a través de la plataforma Moodle ya antes mencionada.

A partir de la situación problemática planteada anteriormente surge el siguiente **problema científico**:

¿Cómo facilitar la gestión de asignaturas de la herramienta educativa SMProg para el EVA-UCI?

Como **objeto de estudio** se tienen las aplicaciones informáticas para la gestión de información académica. Y como **campo de acción** las aplicaciones informáticas para la gestión de información académica desarrolladas utilizando PostgreSQL como sistema gestor de bases de datos.

El **objetivo general** de la investigación es:

Diseñar e implementar un módulo de gestión de asignaturas para la herramienta educativa SMProg para el EVA-UCI.

La **idea a defender** es: Con la creación del módulo de gestión de asignaturas Virtual_Know como complemento de SMProg, se garantizará que los profesores puedan realizar la gestión y administración del contenido que será evaluado en los juegos didácticos de forma correcta.

Los **objetivos específicos** de esta investigación son:

- Realizar un estudio del estado del arte de las aplicaciones existentes para la gestión de información académica como fundamento de la investigación.
- Diseñar una aplicación que permita la gestión de asignaturas para la herramienta SMProg.
- Implementar una aplicación que permita la gestión de asignaturas para la herramienta SMProg.

Para darle cumplimiento a estos objetivos se han definido las siguientes **tareas investigativas**:

- Realizar un estudio del estado del arte. Investigar sobre aplicaciones que se dediquen a la gestión de información académica.
- Estudiar aspectos relacionados con las distintas herramientas a utilizar en la confección de la aplicación, para acelerar y darle cumplimiento a la investigación.
- Realizar el levantamiento de los distintos requisitos con que debe cumplir la aplicación.
- Realizar el análisis, diseño e implementación de la herramienta teniendo en cuenta los aspectos descritos en cada fase.

Para llevar a cabo la presente investigación se utilizaron los siguientes **métodos científicos** de investigación:

Teóricos

- **Analítico - Sintético:** Para realizar el análisis de las características de las herramientas que serán usadas durante la implementación del módulo de gestión, seleccionando la mejor opción entre ellas.
- **Histórico - Lógico:** Para determinar la tendencia actual de los sistemas de gestión de asignaturas y tomar lo positivo de estos para realizar una buena implementación del sistema y una correcta fundamentación de la investigación.
- **Inductivo – Deductivo:** Derivados de este método se encuentran los conocimientos adquiridos durante la investigación sobre los sistemas que gestionan información académica a partir del análisis de lo global del sistema.
- **Modelado:** Se utiliza este método para representar una posible interfaz de la aplicación a través del diseño planteado para modelar el sistema.

Empíricos

- **Observación:** Para realizar una correcta y selectiva percepción de las restricciones y propiedades que el sistema debe tener.

El presente documento está estructurado por un Resumen, la Introducción, 4 Capítulos, las Conclusiones, Recomendaciones, Bibliografía, Referencias Bibliográficas, Anexos y el Glosario de Términos.

Capítulo 1: Fundamentación Teórica de la Investigación.

En este capítulo se aborda la fundamentación teórica de la investigación, destacando aspectos relacionados con las tendencias actuales o estado del arte existente de las aplicaciones informáticas para la gestión de la información académica. Se exponen conceptos relacionados con la gestión y las asignaturas, se realiza un análisis de las tecnologías de software que serán usadas durante la implementación del sistema, los lenguajes de programación y las metodologías candidatas a ser utilizadas en el proceso de desarrollo así como las principales características de la arquitectura de la aplicación.

Capítulo 2: Características de la solución propuesta.

En este capítulo se representa el modelo de dominio, así como el levantamiento de todos los requisitos del sistema; se definen los casos de usos y su descripción textual, además de los roles que intervienen en la aplicación, en este caso el actor, con su correspondiente especificación.

Capítulo 3: Descripción de la solución propuesta.

En este capítulo se expone el análisis y diseño de la aplicación en cuestión, se muestran los resultados obtenidos en el flujo de trabajo de Diseño del sistema; los cuáles serán usados posteriormente en la implementación de la herramienta Virtual_Know. En este flujo de trabajo se realiza un refinamiento del análisis, teniendo en cuenta los requisitos funcionales y no funcionales, garantizando que el sistema cumpla con los requerimientos deseados, considerando además el entorno de implementación.

Capítulo 4: Construcción de la solución propuesta.

En este capítulo es donde se expone el flujo de trabajo de implementación, el cual comienza a partir de lo obtenido en el flujo de trabajo de Diseño. Se muestran las dependencias software-hardware a través del diagrama de despliegue, además se ilustra la dependencia existente entre los componentes del sistema a través del diagrama de componentes y se exponen además los resultados de algunas de las pruebas realizadas a las funcionalidades más importantes del sistema.

Capítulo 1: Fundamentación Teórica de la Investigación

El problema de la gestión procede de diferentes campos de la vida social. Desde tiempos remotos el hombre se vio en la necesidad de resolver sus propios problemas buscando las alternativas que en su momento consideró más apropiadas. Uno de estos problemas fue y continúa siéndolo hoy en día la gestión del conocimiento, puesto que antaño dichos conocimientos se perdían con el paso de los años al ser transmitidos de persona a persona. Con el creciente desarrollo de la humanidad, la necesidad de conocer lo que le rodeaba, lo que había acontecido con sus antepasados, lo nuevo que surgía, así como el perfeccionamiento de lo ya existente, obligó a la especie humana a crear métodos y herramientas para obtener los conocimientos necesarios. Como resultado de esta necesidad y con el decursar de los años han surgido varias definiciones que hacen referencia al concepto de gestión del conocimiento.

1.1 Herramientas para la gestión del conocimiento

El concepto de gestión del conocimiento nació en la década de los 90s y fue un boom y una total reforma para los conceptos tradicionales del campo administrativo, este surgió básicamente de una necesidad impuesta por el mercado. En síntesis la gestión del conocimiento consiste en la capacidad de generar nuevos conocimientos partiendo de datos y experiencias previas, asimismo de organizar, dirigir y controlar los ya existentes **(1)**.

Seguidamente se expondrán algunos conceptos formales de **Gestión**:

1. Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización. **(2)**.

Capítulo 1: *Fundamentación Teórica de la Investigación*

2. Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se desean obtener. **(2)**.

Todo proceso de gestión que se lleve a cabo lleva consigo un conjunto de objetivos esenciales para su ejecución, entre los cuáles se destacan:

- **Planificación:** Es un método a través del cual se establecen metas o retos que serán posteriormente alcanzados y que definen las posibles acciones que se llevarán a cabo durante la elaboración del producto, las que permitirán alcanzar los fines antes mencionados.
- **Organización:** Es el proceso en el cual se preparan y dividen las acciones que se llevarán a cabo, se hace una correcta asignación de las mismas, se mezclan todos los elementos participantes a fin de obtener los resultados esperados, se combinan en esta fase habilidades, recursos, técnicas y todo lo que pueda convertirse en un futuro resultado.
- **Dirección:** Es en este proceso donde se controla, se administra y se verifican los esfuerzos laborales de los implicados en la elaboración del producto, el guía o líder, en este caso la persona encargada de dirigir, se integra al personal restante brindando su apoyo y ayuda en lo que sea necesario durante todo el desarrollo de dicho proceso.

Capítulo 1: Fundamentación Teórica de la Investigación

- **Control:** Es el que permite el chequeo de las actividades y recursos a través de su correcta manipulación, corrigiendo errores en caso de haberlos, tomando decisiones una vez determinadas una gran variedad de medidas, manifiesta el establecimiento de un conjunto de normas que servirán como base para la medida de los resultados de investigación, análisis, diseño, implantación, así como para otros métodos de verificación directa.

Por consiguiente para el hombre surgieron otras necesidades, como las de una vez obtenidos los conocimientos estos debían organizarse, y posteriormente necesitaban ser guardados de forma tal que sirvieran a las presentes y futuras generaciones, las materias se fueron almacenando en diversos medios duros como por ejemplo, pergaminos, escritos, libros, etc. Pero con el desarrollo del pensamiento humano este proceso también evolucionó de forma tal que se construyeron lugares especiales para acumular este gran volumen de información, estos espacios se creaban con el propósito de concentrar en lugares específicos todos estos libros y pergaminos, los cuales todavía en la actualidad se continúan utilizando, como ejemplo tenemos a las bibliotecas y los archivos nacionales o municipales, los que hoy en día constituyen una de las principales fuentes de sabiduría existentes. No obstante, surgió la revolución de las tecnologías y el desarrollo continuaba, pero a su vez el cúmulo de información se hacía tan grande que era prácticamente imposible almacenarla en su totalidad o simplemente, si se necesitaba acceder a las mismas de forma inmediata el proceso de búsqueda se hacía muy tedioso.

Con el surgimiento de la era digital, una novedosa corriente tecnológica que vino a cambiar la vida de las personas, todo esto se vio solucionado ya que surgieron las computadoras, se agilizaron muchos procesos de los ya existentes y muchas de las necesidades del hombre, como la de almacenar sus conocimientos, fueron resueltas. Enseguida ante lo novedoso, comenzaron a surgir las ideas de cómo todo ese conocimiento se podría almacenar de forma digital, para que una vez almacenado, pudiera ser utilizado y quedarán solucionados los problemas del gran cúmulo de información y del acceso a ésta.

Capítulo 1: Fundamentación Teórica de la Investigación

En el mundo se han concebido gran variedad de sistemas para la gestión de datos o más bien conocidos como Herramientas de Gestión de la Información, cada uno de los cuáles está orientado a una esfera distinta ya que a medida que se fueron acentuando las necesidades, estos se fueron desarrollando y especializándose en su desempeño. Existen numerosos sistemas de gestión de información, como ejemplo se encuentran los que se utilizan para la gestión de la información contable, económica y financiera, sistemas para la gestión gubernamental así como para otras áreas del comercio y la sociedad. Otros de los que pueden ser citados son los destinados a la gestión de la información académica, estos sistemas están diseñados fundamentalmente para ser utilizados en los diversos sistemas de enseñanza, debido a las favorables y numerosas ventajas que ofrecen para este proceso. Cuando se hace referencia a la gestión académica se hace alusión a la gestión de asignaturas principalmente, entendiéndose por asignaturas las materias que forman una carrera o un plan de estudios y que se dictan en los centros de enseñanza. Algunos ejemplos de asignaturas son la Matemática, la Programación, la Filosofía etc. **(3)**.

Las Herramientas de Gestión de la Información se clasifican en dos grandes grupos, Las Herramientas de Trabajo en Grupo y los Portales Colaborativos, a continuación se detallan cada uno de estos grupos y se mencionan algunos ejemplos:

1. **Herramientas de Trabajo en Grupo:** son utilizadas para generar procesos colaborativos, distribuir y sincronizar tareas en la organización de forma que ayudan a reducir el tiempo y aumentar la eficacia en dicha entidad, como ejemplo se encuentran:
 - **SAP Knowledge Warehouse:** contiene un repositorio para almacenar contenidos de determinadas clases o tipos y posee herramientas para crear, modificar, eliminar y consultar dicho contenido.

Capítulo 1: Fundamentación Teórica de la Investigación

- **Interlan Systems:** eficaz herramienta flexible y adaptable a la mayoría de los estándares informáticos actuales al permitir a cualquier empresa clasificar, archivar, así como publicar información de una manera rápida y sencilla.
- **OnBase:** permite la captura electrónica, el almacenamiento y gestión de cada documento nuevo que entra a la empresa donde se encuentra funcionando, se integra con aplicaciones de CRM y ERP.

2. **Portales Colaborativos:** dentro de los que se destacan los denominados portales del conocimiento, son los que permiten aprovechar los estándares de Internet e integrar las distintas herramientas de gestión de datos e información, y así un numeroso conjunto de aplicaciones que su único fin es la gestión de la información **(4)**.

Las aplicaciones mencionadas dentro del grupo de Herramientas de Trabajo en Grupo están desarrolladas sobre software propietario, por lo que el acceso a ellas no está disponible para todas las personas, puesto que el que no esté en condiciones de pagarlas se ve privado de su uso, sin embargo se han implementado bajo Software Libre otras que permiten la gestión del conocimiento académico y están dentro del grupo de los Portales Colaborativos, entre ellas se encuentran **Claroline** y **Moodle** (ambas son CMS usados para crear portales colaborativos), muy similares entre ellas en cuanto a funcionamiento destacándose Moodle por ser la más usada entre todas para estos fines por las características ya antes mencionadas.

A nivel nacional se han venido explotando estas opciones educativas destacándose los portales EVAs de varias universidades, entre los que podemos destacar el de la Universidad de La Habana (UH), el del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE del cual se deriva el portal de la UCI) y el de la Universidad Central de las Villas Marta Abreu (UCLV) respectivamente, contruidos todos sobre Moodle, los

cuales cumplen al 100 % con la función de virtualizar la enseñanza y de mantener un espacio donde a cualquier hora se pueden realizar consultas de los aspectos que en ellos se exponen.

En la UCI también ha comenzado a funcionar este sistema de enseñanza obteniendo satisfactorios resultados, aunque vale destacar que cada día se trabaja más en él con el único propósito de incrementar sus funcionalidades, esto se hace mediante la incorporación de diversos módulos, los que permiten dinamizar su contenido. Con la incorporación de SMProg, a modo de ejemplo, se ve una nueva forma de evaluación online la que constituye un sistema de juegos didácticos a través de los cuáles será evaluado todo el contenido o una gran parte del programa de estudio de un año específico de la carrera de Ingeniero en Ciencias Informáticas.

Como tal la herramienta que permite la gestión de los conocimientos que debe llevar dicho módulo de juegos aún no se ha implementado en la UCI, por lo que surge la necesidad en la Facultad 10 de implementar un **sistema automatizado que permita gestionar los contenidos que serán evaluados en el módulo de juegos didácticos SMProg para el EVA-UCI**, el cuál permita darle solución a la problemática antes planteada.

1.2 Metodología de desarrollo, Lenguaje de modelado y Herramientas CASE.

La tendencia o alcance de este trabajo es el de garantizar, a través de la implementación de la aplicación Virtual_Know, que SMProg cuente con una funcionalidad

que le permita a los profesores gestionar los contenidos o conocimientos que posteriormente serán evaluados a través de los juegos didácticos.

Para el desarrollo de dicha aplicación se realizó un estudio sobre las características de las principales herramientas a utilizar así como de las metodologías a ser empleadas durante su construcción. Analizando las tendencias actuales y las novedades existentes en el campo del desarrollo se llega a las siguientes propuestas acerca de las metodologías, lenguajes de programación así como otras herramientas que serán empleadas durante todo el desarrollo de la aplicación.

1.2.1. Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software RUP (en inglés: *Rational Unified Process*) es una metodología de desarrollo de software no ágil, la cual constituye entre todas las metodologías estándares la más utilizada para el diseño, implementación y documentación de sistemas orientados a objetos, esta no presenta pasos que estén firmemente establecidos sino que cuenta con un conjunto de flujos de trabajo adaptables al contexto y necesidades de cada organización.

RUP cuenta con un ciclo de vida que se caracteriza por ser **(5)**:

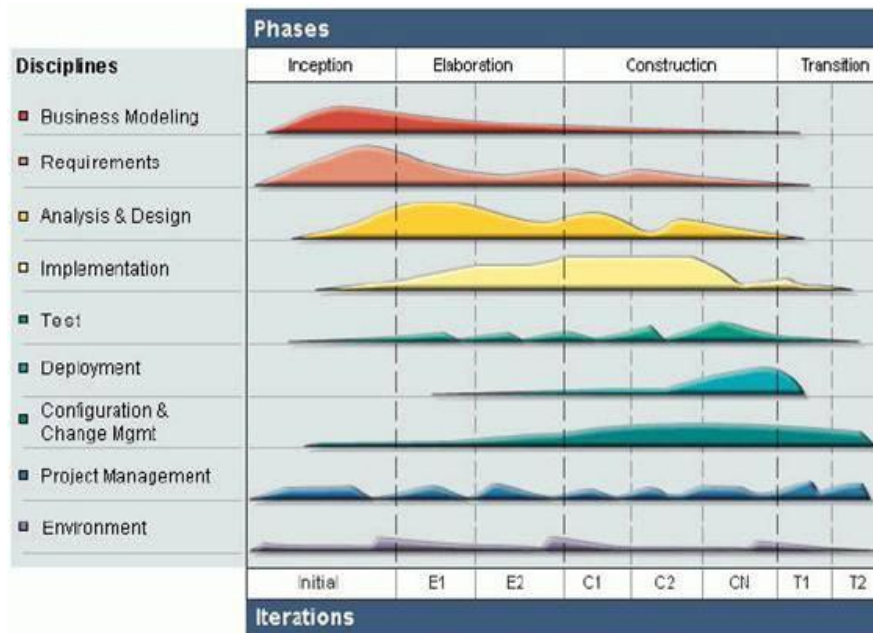
1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros desean, lo cual es identificado cuando se realiza el modelo de negocio y se representa a través de los requerimientos del sistema. Estos casos de uso son los que guían el proceso de desarrollo, ya que los modelos que son arrojados por los diferentes flujos de trabajo son los que representan como se llevan a cabo los casos de uso.

Capítulo 1: *Fundamentación Teórica de la Investigación*

2. **Centrado en la arquitectura:** La arquitectura del sistema muestra la visión completa del mismo ya que los usuarios y el equipo de proyecto deben de estar de acuerdo con ella, describe además los elementos del modelo que son más importantes para su construcción o los que son la base para comprenderlo, desarrollarlo y producirlo con la factibilidad económica que se desea. RUP se desarrolla mediante iteraciones las cuales definen los casos de uso más importantes para el sistema desde el punto de vista arquitectónico. La arquitectura invoca los elementos más significativos para dicho sistema y resulta influenciada por plataformas de software, sistemas operativos, herramientas gestoras de bases de datos, los protocolos, consideraciones y requisitos no funcionales, además cabe destacar que la arquitectura también abarca decisiones importantes con respecto a la organización del sistema, su estructura así como las interfaces que lo componen y sus comportamientos.

3. **Iterativo e Incremental:** La estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre Casos de Uso y Arquitectura se vaya logrando durante cada mini proyecto, siendo así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Todas las fases antes mencionadas están conformadas por 9 flujos de trabajo principales, siendo los 6 primeros conocidos como **flujos de ingeniería** (Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba o Testeo e Instalación o Despliegue) y los 3 últimos como **flujos de apoyo** (Administración de Proyecto, Administración de Configuración y Cambios y el flujo de Ambiente).



Características de la Metodología RUP

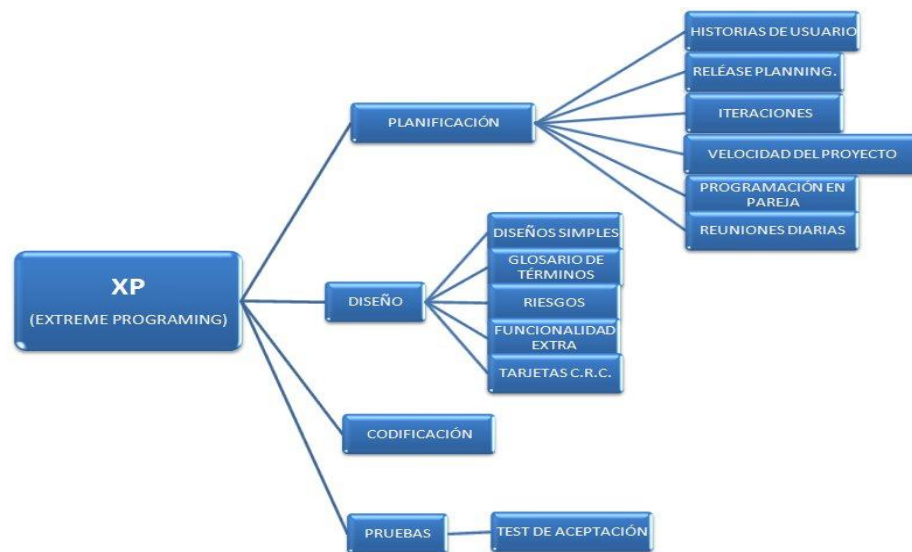
1.2.2 eXtreme Programming

Otra de las metodologías de desarrollo que más se utiliza en el desarrollo de aplicaciones es la eXtreme Programming (XP). Esta es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software, entre sus objetivos principales se encuentran 2 que son muy simples: **la satisfacción del cliente**: esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita, y el segundo objetivo es **potenciar al máximo el trabajo en grupo**: tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Cumple también con fases pero cada una de ellas es específica, por ejemplo en la fase de "Diseño" se plantea que estos deben ser de manera simple, hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácil y entendible por lo que a veces se obvian detalles que son importantes a la hora de la implementación. En la

Capítulo 1: Fundamentación Teórica de la Investigación

fase de "Codificación" no se debe olvidar que los clientes son los que crean las historias de usuario y negocian los tiempos de implementación, y en la fase de "Pruebas" se realiza una verificación de lo realizado para así asegurar el funcionamiento y para comprobar que las distintas historias de usuario cumplen su cometido. Estas pruebas son diseñadas para saber las funcionalidades que debe cumplir la aplicación una vez especificadas en los requisitos.



Características de la metodología XP

Después de realizado un análisis acerca de las metodologías a usar se determinó por parte del grupo de desarrollo que se usaría RUP puesto que esta se adapta más a las necesidades, características y complejidad del software a desarrollar, en ella se hace una correcta captura de requisitos lo cual agiliza el trabajo y a la vez cumple con todas las expectativas y funcionalidades que se exigen para el producto, es de vital importancia esta metodología para proyectos donde se tenga que virar hacia atrás en un determinado momento, ya una vez detectado un posible error, es posible retroceder y corregirlo sin que atente en un futuro con el correcto funcionamiento de la herramienta. Además de que al comienzo del proyecto SMProg se decidió llevarlo a cabo utilizando RUP, y como

la presente investigación constituye un módulo de dicho proyecto se decidió por parte del equipo de desarrollo realizarla con la misma metodología.

1.2.3. Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado más conocido por todos como UML (por sus siglas en inglés *Unified Modeling Language*), es un lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software **(6)**, es el más usado ya que ofrece un estándar para describir un plano o modelo del sistema en el cual se incluyen los conceptos de proceso de negocio así como funciones del sistema. UML es un lenguaje para especificar y no para escribir métodos o procesos, por lo que es como tal el lenguaje que describe el modelo de un determinado sistema.

UML en su funcionamiento emplea una serie de diagramas que son fundamentales para el correcto uso del mismo, los cuáles son **(6)**:

- **Diagramas de estructura estática:** Describen las propiedades estructurales del sistema, dentro de estos diagramas podemos encontrar los de clases, los de objetos y los de casos de uso.
- **Diagramas de Comportamiento:** Son los diagramas que definen como tal el comportamiento del sistema mostrando para esto las interacciones ocurridas entre los objetos en un escenario o parte del sistema, entre ellos podemos encontrar los de interacción (secuencia y colaboración), los de estado así como los de actividad.
- **Diagramas de Implementación:** Son los que definen la configuración final, la organización entre componentes, incluyen el código fuente del sistema etc., entre estos podemos encontrar los diagramas de componentes y de despliegue.

1.2.4. Herramientas CASE

Las herramientas CASE (*por sus siglas en inglés: **Computer Aided Software Engineering**, o Ingeniería de Software Asistida por Ordenador*) son aplicaciones informáticas cuyo objetivo es el de aumentar la productividad y la eficiencia en el desarrollo de software minimizando de esta forma el costo en términos de tiempo y dinero. Entre estas aplicaciones se encuentran el ArgoUML, el Umbrello así como el Visual Paradigm en la cual estará centrada toda la atención por ser una suite de trabajo (entiéndase por suite: varias herramientas dentro de una sola) fácil de usar, su entorno es de ágil comprensión por el usuario, permite la importación de ficheros desde otras herramientas como el Rational Suite, permite realizar el proceso de ingeniería inversa así como la generación de código entre otras opciones que facilitan el trabajo y aumentan la preferencia de los usuarios por la misma. Entre sus más potentes herramientas están los diseñadores de diagramas con que cuenta y la generación de artefactos durante los ciclos de desarrollo del software, así como la integración con otros IDEs de desarrollo como por ejemplo el NetBeans y el Eclipse por solo citar algunos.

La aplicación que se desarrollará utilizará el Visual Paradigm para la modelación del UML y para la creación del modelo entidad-relación de la base de datos desarrollada sobre PostgreSQL así como también las técnicas de Programación Orientada a Objetos (POO), conformándose de esta forma la base del funcionamiento visual de RUP. Teniendo en cuenta además, para la selección de esta suite aparte de las ya mencionadas, las siguientes características:

1. Permite la automatización y mantenimiento de las tareas de desarrollo lo cual es un factor importante dentro del proceso de elaboración del software.
2. En la UCI se cuenta con la licencia para el uso de dicha suite, opción esta que facilita las condiciones para el uso de la misma.

1.3 Lenguajes de programación, tendencias actuales y herramientas de desarrollo.

Un lenguaje de programación no es más que un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar, por lo tanto es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo electrónico o computadora en este caso **(8)**. Las acciones a las cuales se hace referencia son conocidas como código fuente del programa, dichas acciones o instrucciones consisten en un conjunto de reglas sintácticas que definen la estructura del programa y en un conjunto de reglas semánticas que definen el significado de los elementos.

1.3.1. Programación Orientada a Objetos

La POO ha tenido tanto auge en su desarrollo que hoy en día es soportada por la mayoría de los lenguajes de programación de alto nivel existentes, ya que es una técnica de estructuración en la cual los objetos son los principales elementos de construcción y se conectan entre sí por medio de mensajes, siendo estos últimos los que describen el comportamiento de dichos objetos. Esta técnica posee grandes ventajas como son:

- 1. Encapsulamiento:** Es el proceso bajo el cual se mantienen agrupados bajo una misma cápsula la estructura y el comportamiento de los objetos que intervienen en el sistema.
- 2. Herencia:** Permite la construcción de clases nuevas a partir de las ya existentes, además de potenciar la reutilización de código.
- 3. Polimorfismo:** Es como tal la capacidad que tienen los objetos que pertenecen a determinadas clases de comportarse de distintas (*poli*) formas (*morfos*) diferentes ante la ocurrencia de un evento determinado.

1.3.2. Lenguaje de Programación Java.

Java no es más que un lenguaje creado por la compañía extranjera Sun Microsystem y que en su sintaxis toma mucho de otros lenguajes como son C y C++, presenta características propias como son la eliminación de herramientas de bajo nivel así como un modelo de objetos más simple, los cuáles han venido a solucionar problemas como la incorrecta manipulación de los punteros y a su vez de la memoria del sistema.

Uno de los objetivos principales del presente trabajo de diploma es desarrollar la aplicación Virtual_Know sobre el lenguaje de programación Java para su posterior muestreo y su integración con la herramienta educativa SMProg, por lo que se tuvo que realizar un exhaustivo análisis de los lenguajes de programación existentes, como es el caso de C++, Python, CSharp, para determinar las razones por la cual se escogió a Java como candidato.

1.3.2.1. Razones por la cual se escoge Java para la implementación del sistema.

Entre las principales razones por la cual se escoge a Java para la implementación de la herramienta Virtual_Know se encuentran las siguientes:

1. La portabilidad de su plataforma.

La portabilidad del sistema está basada principalmente en que el compilador del Java genera un código binario conocido como *bytecode* el cual es interpretado por la Máquina Virtual de Java (JVM por sus siglas en inglés: *Java Virtual Machine*), y esta a su vez se encarga de ejecutar dicho código. Esto permite que un programa escrito en Java pueda ser ejecutado en cualquier ordenador independientemente de la plataforma que este use, ya sea Windows o Linux, ya que la máquina virtual es independiente de la plataforma.

2. La propiedad de poder incrustarse en otros programas.

Otra de las razones por las cuáles Java es uno de los lenguajes más usados entre los desarrolladores es por la condición de poder incrustarse en cualquier otro programa en forma de applet, un applet es un componente de Java que se ejecuta en el contexto de otro programa, el más común de los ejemplos es la incrustación de estos en los navegadores web, ya que los applets se utilizan para representar algún contenido dentro del mismo.

Además el archivo ejecutable .jar creado luego de la compilación de la aplicación puede ser ejecutado en cualquier contexto ya que también es independiente de la plataforma pues solamente necesita de la existencia de la máquina virtual antes mencionada. Mediante estas funcionalidades o característica del lenguaje Java es que la aplicación que se deriva de esta investigación será desarrollada.

1.3.3. Sistema gestor de Bases de Datos Relacionales Orientado a Objetos PostgreSQL

Capítulo 1: Fundamentación Teórica de la Investigación

PostgreSQL está derivado del paquete Postgres escrito en el Departamento de Ciencias de la Universidad de California en Berkeley, pero su implementación surgió como tal en el año 1986 con el nombre de Postgres, en 1994 recibió el nombre de Postgres95 y el código de este fue adaptado a ANSI C lo cual llevó a una considerable reducción de errores con respecto a las versiones anteriores. No es hasta 1996 cuando se decide nombrarlo PostgreSQL como tal ya que incorporó en su código a SQL mejorando una serie de factores en el mismo, como por ejemplo en el motor de búsqueda se incluyeron subconsultas, valores por defecto así como restricciones a valores en algunos campos.

Como cualquier otro sistema de gestión, Virtual_Know trabaja con una base de datos en la cual se llevarán a cabo todas las operaciones con el contenido que se procesará, ya sea insertar, eliminar, modificar o consultar dicho contenido, PostgreSQL es un potente sistema gestor de bases de datos, el cuál presenta ciertas ventajas sobre otros sistemas gestores de bases de datos como Oracle, MySql, Fortran etc., entre ellas se pueden citar algunas como por ejemplo instalación ilimitada, mejor soporte que cualquier otra versión puesto que cuenta con una amplia comunidad de desarrolladores, es multiplataforma, es extensible al estar su código fuente disponible para todos sin coste alguno, es estable en comparación con los demás sistemas gestores ya que hasta ahora no se ha reportado nunca una caída de un sistema que use este gestor, es ideal para manejar amplios volúmenes de información entre otras características que se mencionan a continuación.

1.3.3.1 Razones por las cuáles se escoge PostgreSQL para el diseño de la base de datos de la herramienta Virtual Know

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional ORDBMS, (en inglés: *Object Relational Database Management System*) que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Presenta una alta concurrencia a través de un sistema denominado Control de Concurrencia Multiversión (MVCC por sus siglas en inglés *Multi-Version Concurrency Control*) ya que permite que mientras un proceso está escribiendo datos en una tabla otros accedan a la misma tabla sin necesidad de bloqueos, esta estrategia es superior al uso de bloqueo por tablas o por filas muy común entre otras bases de datos puesto que elimina la necesidad del uso de bloqueos explícitos. A pesar de esto cabe destacar que PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Las principales características de este sistema gestor de bases de datos son las siguientes:

1. Implementación del estándar SQL92/SQL99.
2. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
3. Incorpora una estructura de datos array.
4. Incorpora funciones de diversa índole: manejo de fechas, geométricas etc.
5. Permite la declaración de funciones propias, así como la definición de disparadores.
6. Soporta el uso de índices, reglas y vistas.
7. Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

Capítulo 1: Fundamentación Teórica de la Investigación

8. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
9. Posee interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.

Cabe destacar además que PostgreSQL es el gestor de bases de datos de código abierto más avanzado que existe hoy en día, soportando casi toda la sintaxis del lenguaje SQL (incluyendo subconsultas, transacciones etc.), contando también con un amplio conjunto de funciones incorporadas como son respaldo al cliente, índices parciales y funcionales, autenticación Kerberos nativa, soporte para consultas con UNION, UNION ALL y EXCEPT entre otras.

1.3.4 Herramientas de Desarrollo

Las herramientas de desarrollo que se analizarán se dividirán en 2 grupos, el primero estará conformado por las herramientas para el trabajo con el lenguaje Java y el segundo por las herramientas para el trabajo con PostgreSQL.

Para Java:

Un IDE de desarrollo (en inglés: *Interface Development Environment*) no es más que una interfaz mediante la cual se puede interactuar con un lenguaje determinado, para el caso del lenguaje Java existen en la actualidad diversos IDEs, como ejemplo se pueden mencionar el Eclipse, el JBuilder, el Rational Application Developer, el Visual Studio así como el NetBeans en el cuál se centrará toda la atención el ser el IDE con el que se desarrollará la herramienta Virtual_Know, sobre todas las cosas porque es además un IDE Open Source o de código abierto.

NetBeans es un entorno de programación para varios lenguajes, creado por la Sun Microsystem que incluye a Java, PHP y C++ dentro de los lenguajes soportados. Este IDE es de código abierto, por lo que su fuente es proporcionada a los desarrolladores

Capítulo 1: Fundamentación Teórica de la Investigación

para que se pueda modificar de acuerdo a las necesidades o exigencias de los usuarios, siempre respetando ciertos parámetros de licencia que no deben ser violados. Es además una plataforma de ejecución para aplicaciones que proporciona una serie de servicios disponibles desde el propio IDE. Destacándose además que NetBeans viene integrado con un conjunto de herramientas necesarias para el desarrollo de aplicaciones Web o Desktop a diferencia del Eclipse, al cual es necesario ir instalándole progresivamente *plug-ins* según sea necesario.

Para PostgreSQL:

En el caso de las bases de datos estas herramientas se conocen como clientes gráficos, entre estos clientes se encuentran el EMS PostgreSQL Manager así como el pgAdmin, siendo este el escogido para la creación y administración de la base de datos de la aplicación Virtual_Know, así como para otras funciones que se necesite. El pgAdmin es conocido como el máximo exponente de cliente gráfico para PostgreSQL, aparte de que es una herramienta libre y multiplataforma, posee una serie de características que a continuación se mencionan:

1. Puede trabajar con todos los objetos de la Base de Datos, examinar sus propiedades así como realizar tareas administrativas en la misma, (entiéndase por objetos de la Base de Datos a columnas, dominios, funciones, grupos, índices etc.).
2. Cada vez que se realiza una modificación en un objeto escribe las sentencias SQL correspondientes a dicha modificación, lo que lo convierte aparte de una herramienta útil en una herramienta didáctica.
3. Incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos.

Todas estas características hacen que pgAdmin sea la herramienta que se necesite para el trabajo con PostgreSQL, ya que tanto desde el punto de vista del administrador como

de los usuarios, las acciones que se llevarán a cabo a través de este cliente serán de acuerdo a los privilegios definidos en la base de datos para el usuario que ha establecido conexión con ella.

1.4 Arquitectura de software

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Es el esqueleto o base de una aplicación la que es analizada desde varios puntos de vista. Constituye la nueva visión que los desarrolladores tienen de los sistemas de software en la última década. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su arquitectura (2).

1.4.1 Estilos arquitectónicos

Arquitectura n-Capas

Con el objetivo de solventar los problemas de escalabilidad, disponibilidad, seguridad e integración que puedan presentar las aplicaciones compactas, se ha generalizado la división de las aplicaciones en capas. La capa que se agrega es la que surge de separar definitivamente las reglas del negocio de la de Acceso a Datos. Esta arquitectura trae consigo la ventaja de aislar definitivamente la lógica del negocio de todo lo que tenga que ver con el origen de datos, de modo que ante cualquier eventual cambio, solo se deberá tocar un módulo específico, así como al momento de plantear la escalabilidad del sistema no se afrontarán grandes modificaciones. Dentro de este tipo de arquitectura una de las divisiones muy utilizada en el desarrollo de sistemas es la arquitectura en tres capas.

Patrón de arquitectura Modelo Vista Controlador (MVC)

El MVC (por sus siglas en inglés: **Model View Controller**) fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. MVC es un patrón de arquitectura de software clásico que está formado por tres niveles:

- **El modelo:** que representa la información con que trabaja la aplicación.
- **La vista:** que transforma el modelo en una interfaz a través de la cual el usuario puede interactuar con ella.
- **El controlador:** que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o la vista.

Este patrón presenta ciertas ventajas como son:

- Hay una clara separación entre los componentes de un programa, lo que permite implementarlos por separado.
- La conexión entre el Modelo y sus Vistas es dinámica, se produce en tiempo de ejecución no en tiempo de compilación.

Para el desarrollo de la aplicación Virtual_Know se escogió el patrón de arquitectura Modelo-Vista-Controlador ya que al incorporar el modelo de arquitectura MVC a un diseño de aplicaciones, las piezas de estas se pueden elaborar de forma independiente y luego podrán ser mezcladas en tiempo de ejecución, en caso de que uno de los componentes elaborados funcione de forma incorrecta o se decida realizarle una posterior modificación, podrá ser reemplazado sin que los otros componentes se vean afectados por esta acción.

Capítulo 1: Fundamentación Teórica de la Investigación

Tomando como punto de partida todo lo antes analizado se propone la creación de un módulo que se encargue de la gestión de asignaturas para el SMProg o módulo de juegos didácticos del EVA-UCI. En el presente capítulo se realizó un detallado análisis de las tecnologías, herramientas de diseño y desarrollo así como de la metodología y la arquitectura a seguir para el correcto desarrollo de la herramienta Virtual_Know.

Capítulo 2: Características de la solución propuesta

En este capítulo serán presentados el modelo de dominio, el diagrama de casos de uso así como el levantamiento de todos los requisitos funcionales y no funcionales con que debe cumplir la aplicación, se definirán y detallarán los casos de usos presentes en el sistema, los roles que intervienen, en este caso los actores, y también se representarán algunos artefactos que se desprendan de las distintas fases por la que atraviese.

2.1 Modelo de Dominio

Debido a que en la presente investigación y en la metodología de desarrollo de software aplicada no se tiene bien definido el negocio como tal, ya que no se logra establecer lo planteado para conformar el modelo de negocio, entonces es necesario identificar los diversos componentes que serán usados en el modelo de dominio del sistema así como posteriormente realizar las definiciones de los mismos:

Capítulo 2: Características de la solución propuesta

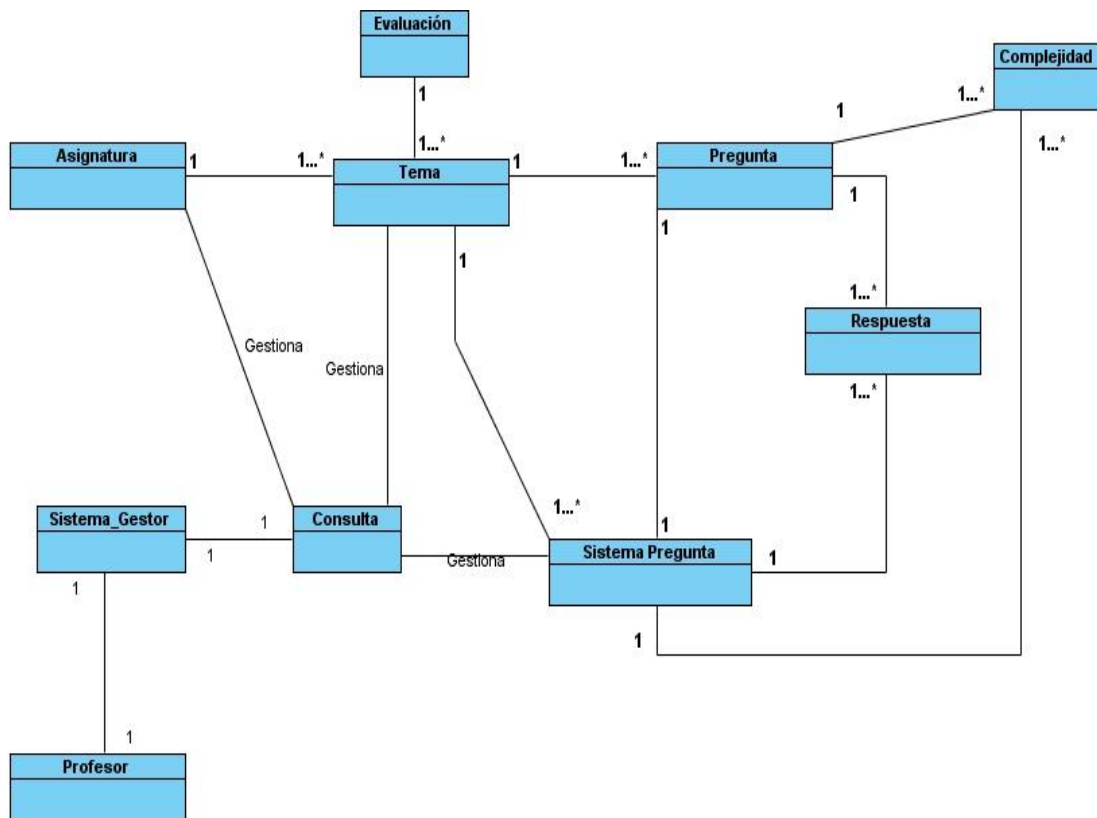


Figura No. 1: Modelo de dominio

En la figura No. 1 se muestra la relación existente entre las entidades o clases las que se encuentran definidas por el modelo de dominio, en la misma se aprecia como el profesor es el encargado de gestionar la información contenida en la base de datos a través de dos clases controladoras o principales, Sistema_Gestor y Consulta, las que se encargan de llevar a cabo las acciones deseadas por el usuario, primeramente el profesor interactúa con la clase Sistema_Gestor la cual una vez insertados y validados los datos de entrada, los envía a la clase Consulta la cual se encargará de llevar a cabo las acciones sobre los distintos campos de la base de datos, entre las operaciones que pueden efectuar sobre el contenido almacenado se destacan la de insertar, modificar, eliminar así como realizar consultas sobre él.

2.2 Levantamiento de requisitos

En el levantamiento de los requisitos se definen 2 tipos de requisitos, los funcionales, que no son más que condiciones que el sistema debe cumplir, y los requisitos no funcionales, los cuáles no son más que cualidades que el sistema debe tener, los mismos especifican las propiedades del sistema como son por ejemplo la dependencia de plataformas, mantenibilidad y confiabilidad, estos requisitos son los que permiten que el sistema sea atractivo, usable, rápido, confiable, etc. En muchas ocasiones los requisitos no funcionales son fundamentales en el éxito del producto.

2.2.1 Requisitos funcionales del sistema.

RF1- Validar la entrada de los usuarios al sistema

RF2- Administrar Asignatura

RF2.1- Insertar Asignatura en la base de datos

RF2.2- Eliminar Asignatura de la base de datos

RF2.3- Modificar Asignatura existente en la base de datos

RF3- Administrar Tema

RF3.1- Insertar Tema en la base de datos

RF3.2- Eliminar Tema de la base de datos

RF3.3- Modificar Tema existente en la base de datos

RF4- Administrar Preguntas

RF4.1- Insertar pregunta

RF4.2- Eliminar pregunta

RF4.3- Modificar pregunta

RF5- Administrar Respuestas

Capítulo 2: Características de la solución propuesta

RF5.1- Insertar respuestas

FR5.2- Eliminar todas las respuestas de una pregunta dada

RF5.3- Eliminar una respuesta

RF5.4- Modificar respuesta

RF6- Mostrar la Información contenida en las tablas de la base de datos

RF6.1- Mostrar el listado de asignaturas de la base de datos

RF6.2- Mostrar el listado de tema de la base de datos

RF6.3- Mostrar el listado de preguntas de la base de datos

RF6.4- Mostrar el listado de las respuestas presentes en la base de datos.

RF6.5- Mostrar el listado de los Sistemas de Preguntas existentes en la base de datos.

2.2.2 Requisitos no funcionales del sistema.

- **Requisito no funcional de Apariencia o Interfaz Externa.**

- Cada funcionalidad debe estar descrita explícitamente con un texto aclaratorio en correspondencia con la función que realiza.
- Las ventanas de trabajo deben ser lo más sencilla pero funcional posible, es decir no estar cargadas en su composición para que sea más fácil de entender por el usuario.

- **Requisito no funcional de Usabilidad.**

- El sistema debe permitir la concurrente utilización por parte de varios usuarios a la vez.

- **Requisito no funcional de Rendimiento.**

- El sistema deberá garantizar la gestión de toda la información así como atender y responder a una determinada petición por parte de los usuarios en un período de tiempo no mayor de 1 segundo.

- **Requisito no funcional de Portabilidad.**

Capítulo 2: Características de la solución propuesta

- La aplicación debe ser independiente de la plataforma en que se ejecute, ya sea Windows o Linux.
 - **Requisito no funcional de Seguridad.**
- Solo se podrá acceder a los datos almacenados una vez autenticado el usuario.
- **Disponibilidad:** Significa que a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
 - **Requisito no funcional de Legales.**
- Las herramientas de desarrollo deben ser herramientas libres o de Open Source en su mayoría y para las que no lo son, las licencias deben estar avaladas para su uso.
 - **Requisito no funcional de Confiabilidad.**
- Debe validar la entrada de datos para evitar entradas incorrectas de los mismos.
- La precisión en los reportes obtenidos debe ser la más exacta posible, midiéndose en términos de calidad y exactitud no así de cantidad.
 - **Requisito no funcional de Software Necesario.**
- Máquina Virtual de Java instalada versión 5.0 o superior.
- Servidor PostgreSQL como SGBD para establecer y mantener la base de datos, en su versión 8.3.
 - **Requisito no funcional de Hardware Recomendado para PC Servidor.**
- 40 o más GB de capacidad en el disco duro.
- Microprocesador de 2.4 GHz o superior.
- Memoria RAM de 2 GB o superior.
- Tarjeta de Red de 100 Mbps o superior.

- **Requisito no funcional de hardware recomendado para PC Cliente.**
 - 15 o más GB de capacidad en el disco duro.
 - Microprocesador 1 GHz o superior
 - Memoria RAM de 512 MB o superior.
 - Tarjeta de red de 100 Mbps o superior.
- **Requisito no funcional de Restricciones de la Implementación.**
 - Se utilizará como lenguaje de programación el Java usando como IDE de desarrollo el NetBeans 6.0.

2.3 Modelo de Casos de Uso

En el caso de la aplicación Virtual_Know, el profesor es el único actor del sistema, ya que es el que interactuará con la aplicación y a su vez será el iniciador de todos los casos de uso presentes en la aplicación. En la figura No. 2 se observan los casos de usos definidos para representar el flujo de los eventos los cuáles permitirán obtener a los actores un resultado de valor apreciable durante la gestión de la información.

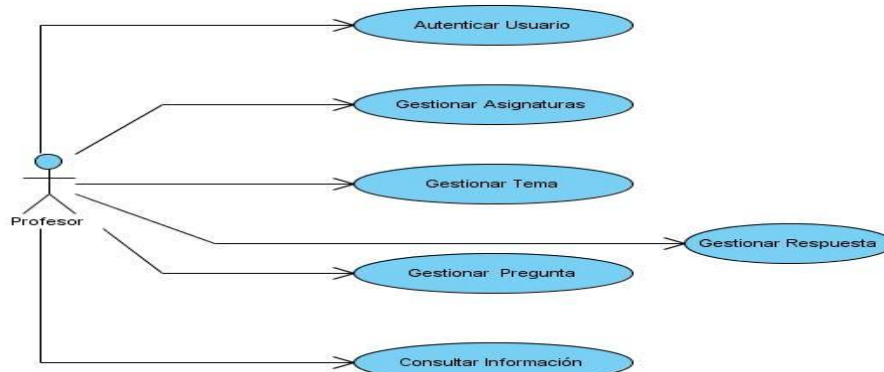


Figura No. 2: Diagrama de casos de Uso del Sistema

Capítulo 2: Características de la solución propuesta

2.3.1 Actores del Sistema

Actor	Descripción
Profesor	Es el encargado de interactuar con el sistema, así como de gestionar todo lo referente a las asignaturas y demás información que manipule.

Tabla No. 1: Actor del Sistema

2.3.2 Casos de Uso del Sistema

CU-1	Autenticar Usuario
Actor	Profesor
Descripción	El profesor selecciona la opción de autenticarse la cual le establecerá la conexión con la base de datos del sistema para posteriores funcionalidades sobre la misma.
Referencia	RF1

Tabla No. 2: Caso de Uso Autenticar Usuario

CU-2	Gestionar Asignatura
Actor	Profesor
Descripción	El profesor realiza sobre las asignaturas múltiples operaciones como pueden ser insertar,

Capítulo 2: Características de la solución propuesta

	modificar y/o eliminar.
Referencia	RF2

Tabla No. 3: *Caso de Uso Gestionar Asignatura*

CU-3	Gestionar Tema
Actor	Profesor
Descripción	El profesor realiza sobre los temas determinadas acciones como son insertar nuevos temas, eliminar así como modificar los ya existentes.
Referencia	RF3

Tabla No. 4: *Caso de Uso Gestionar Tema*

CU-4	Gestionar Preguntas
Actor	Profesor
Descripción	El profesor realiza sobre las preguntas determinadas acciones como son insertar nuevas preguntas, eliminar así como modificar las ya existentes.
Referencia	RF4

Tabla No. 5: *Caso de Uso Gestionar Preguntas*

Capítulo 2: Características de la solución propuesta

CU-5	Gestionar Respuesta
Actor	Profesor
Descripción	El profesor realiza sobre las respuestas determinadas acciones como son insertar nuevas respuestas, eliminar, así como modificar las ya existentes.
Referencia	RF5

Tabla No. 6: *Caso de Uso Gestionar Respuesta*

CU-6	Consultar Información
Actor	Profesor
Descripción	El profesor selecciona lo que desea consultar de la base de datos, las opciones de selección son sobre Asignatura, Tema, Pregunta, Respuesta así como Sistema de Preguntas.
Referencia	RF6

Tabla No. 7: *Caso de Uso Consultar Información*

Capítulo 2: Características de la solución propuesta

2.3.3 Descripción de los Casos de Uso del Sistema

Tabla No. 8: Caso de Uso 1- Autenticar Usuario

CU-1	Autenticar Usuario
Actores:	Profesor
Propósito:	Establecer la conexión con la base de datos para comenzar a trabajar.
<u>Resumen:</u> Se crea la conexión a la base de datos con los parámetros especificados por el usuario.	
Referencias:	RF 1
Precondiciones:	Debe existir una base de datos y un servidor previamente instalado y configurado correctamente donde los datos de conexión coincidan con los especificados por el usuario.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1. El profesor selecciona el menú operaciones.	2. Muestra la opción Conectarse.
3. El profesor selecciona la opción Conectarse.	4. Inicia la forma de autenticación con la base de datos.
5. El profesor inserta los datos requeridos para establecer la conexión.	6. Toma los datos insertados por el usuario, los valida y establece la conexión.

Capítulo 2: Características de la solución propuesta

Curso alterno de los eventos	
	6.1 Si los datos introducidos son incorrectos o existen campos vacíos el sistema retorna al curso normal de los eventos 4.
Pos condiciones:	Se establece la conexión con la base de datos para poder realizar posteriormente las operaciones deseadas.

Tabla No. 9: Caso de uso 2-Gestionar Asignatura

CU-2	Gestionar Asignatura
Actores:	Profesor
Propósito:	Insertar, modificar y eliminar asignaturas.
<p><u>Resumen:</u> Permite al usuario insertar nuevas asignaturas así como eliminar y modificar las ya existentes en la base de datos. Esto está sujeto a la acción que desee realizar el usuario una vez entrada a la aplicación.</p>	
Referencias:	RF 2

Capítulo 2: Características de la solución propuesta

Precondiciones:	El usuario debe previamente haber establecido la conexión a la base de datos, debe existir la tabla asignatura para el caso de insertar y para el caso de modificar y eliminar la tabla no puede estar vacía.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1. El profesor selecciona del menú la opción gestionar información.	2. El sistema muestra las secciones con las que puede trabajar y las opciones a realizar sobre las asignaturas: <ul style="list-style-type: none">• Insertar• Eliminar• Modificar
3. El profesor selecciona la opción deseada.	4. Si el profesor selecciona la opción Insertar, ver sección “Insertar Asignatura” . Si el profesor selecciona la opción Eliminar, ver sección “Eliminar Asignatura” . Si el profesor selecciona la opción Modificar, ver sección “Modificar Asignatura” .
Sección Insertar Asignatura	
	1. Muestra la interfaz con los campos necesarios para la inserción además de la tabla de asignaturas existentes en

Capítulo 2: Características de la solución propuesta

	la base de datos.
2. El profesor inserta el nombre de la asignatura en el campo correspondiente y presiona el botón insertar.	3. Valida los datos y almacena los datos temporalmente en memoria.
4. Presiona el botón Guardar Información.	5. Inserta los datos en la base de datos. 6. Actualiza la base de datos y la tabla mostrando las nuevas asignaturas insertadas.
Curso Alterno de los eventos	
	3.1 Si están vacíos o incompletos los campos, muestra un mensaje de error, el sistema regresa al curso normal de los eventos 1.
Sección Eliminar Asignatura	
	1. Muestra la interfaz de la opción seleccionada además de la tabla de asignatura existente en la base de datos.
2. El profesor selecciona la asignatura a eliminar.	3. La aplicación carga automáticamente en el campo de texto correspondiente el identificador de la asignatura a eliminar.

Capítulo 2: Características de la solución propuesta

4. El profesor presiona el botón Eliminar.	5. El sistema muestra un mensaje de confirmación para verificar si desea verdaderamente eliminar la asignatura. 6. Si acepta se elimina la asignatura seleccionada de la base de datos. 7. Se actualiza la tabla de preguntas así como la base de datos.
Curso alternativo de los eventos	
	5.1 Si selecciona NO el sistema retoma el curso normal de los eventos 1.
Sección Modificar Asignatura	
	1. Muestra la interfaz de la opción seleccionada y la tabla con las asignaturas que existen en la base de datos.
2. El profesor selecciona en la tabla la asignatura a modificar.	3. La aplicación coloca los valores actuales de dicha asignatura en los campos correspondientes.
4. El usuario reemplaza en los campos los valores de la asignatura seleccionada por los	5. Reemplaza los valores existentes en la base de datos por los nuevos valores

Capítulo 2: Características de la solución propuesta

nuevos valores y presiona el botón Modificar.	introducidos por el usuario. 6. Actualiza la tabla con los nuevos valores de la asignatura insertados en la base de datos.
Curso Alternativo de los eventos	
	5.1 Si están vacíos o incompletos los campos, muestra un mensaje de error, y retorna al curso normal de los eventos 2.
Pos condiciones:	Se gestionan las asignaturas.

Tabla No. 10: Caso de Uso 3- Gestionar Tema

CU-3	Gestionar Tema
Actores:	Profesor
Propósito:	Insertar, modificar y eliminar Tema.
<p><u>Resumen:</u> Permite al usuario insertar nuevos temas. Además de eliminar y modificar los ya existentes en la base de datos. Esto está sujeto a la acción que desee realizar el usuario una vez entrada a la aplicación.</p>	
Referencias:	RF3

Capítulo 2: Características de la solución propuesta

Precondiciones:	Se debe haber establecido una previa conexión a la base de datos además debe existir la tabla tema, para el caso de eliminar y modificar debe existir el tema con el que va a trabajar.	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>	
1. El profesor selecciona el menú gestionar información.	2. El sistema muestra el panel correspondiente de las operaciones sobre los temas <ul style="list-style-type: none">• Insertar• Eliminar• Modificar	
3. El profesor selecciona la opción deseada.	4. Si el profesor selecciona la opción Insertar, ver sección “Insertar Tema” . 5. Si el usuario selecciona la opción Eliminar, ver sección “Eliminar Tema” . 6. Si el usuario selecciona la opción Modificar, ver sección “Modificar Tema” .	
Sección Insertar Tema		
	1. Muestra la interfaz de la opción seleccionada además de la tabla que muestra los temas que existen en la base de datos.	

Capítulo 2: Características de la solución propuesta

2. El profesor inserta los valores correspondientes en los campos y presiona el botón Insertar.	3. Valida los datos y los almacena temporalmente en memoria.
4. Presiona el botón Guardar Información.	5. Inserta los datos en la base de datos. 6. Actualiza la base de datos y la tabla de la interfaz mostrando en ella los nuevos temas insertados.
Curso Alterno de los eventos	
	3.1 Si los campos están vacíos o incompletos, muestra un mensaje de error, el sistema retorna al curso normal de los eventos 1.
Sección Eliminar Tema	
	1. Muestra la interfaz de la opción seleccionada además de la tabla de los temas que existen en la base de datos.
2. El profesor selecciona de la base de datos el tema a eliminar.	3. La aplicación carga automáticamente el identificador del tema seleccionado en el campo correspondiente.

Capítulo 2: Características de la solución propuesta

4. El profesor presiona el botón Eliminar.	5. El sistema muestra un mensaje de confirmación para verificar si desea verdaderamente eliminar el tema seleccionado. 6. Si selecciona SI se elimina el tema correspondiente y se actualiza la tabla.
Curso alternativo de los eventos	
	6.1 Si selecciona NO se retorna al curso normal de los eventos 1.
Sección Modificar Tema	
	1. Muestra la interfaz de la opción seleccionada y la tabla con los temas que tiene la base de datos.
2. El profesor selecciona en la tabla el tema a modificar.	3. La aplicación coloca cada valor del tema seleccionado en los campos correspondientes.
4. El usuario modifica los valores de los campos de textos reemplazando los valores que en ellos aparecen por los nuevos y presiona el botón Modificar.	5. La aplicación valida los datos de entrada, si están correctos el sistema reemplaza los valores existentes por los nuevos valores modificados por el usuario.

Capítulo 2: Características de la solución propuesta

	6. Actualiza la tabla con los nuevos valores.
Curso alternativo de los eventos.	
	5.1 Si están vacíos o incompletos muestra un mensaje de error y se retorna al curso normal de los eventos 1.
Pos condiciones:	Se gestionan los temas.

Tabla No. 11: Caso de Uso 4- Gestionar Pregunta

CU-4	Gestionar Pregunta
Actores:	Profesor
Propósito:	Insertar, modificar así como eliminar Pregunta.
<p><u>Resumen:</u> Permite al usuario insertar las preguntas. Además de eliminar y modificar las ya existentes en la base de datos. Esto está sujeto a la acción que desee realizarse una vez autenticado el usuario en la aplicación. A la hora de insertar, modificar o eliminar aparecerá la ventana para realizar la acción deseada sobre la pregunta.</p>	
Referencias:	RF4
Precondiciones:	Debe existir una conexión con la base de datos previamente ejecutada.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>

Capítulo 2: Características de la solución propuesta

1. El profesor selecciona el menú Gestionar Información.	2. El sistema muestra el panel correspondiente de las operaciones sobre el sistema de preguntas. <ul style="list-style-type: none">• Insertar• Eliminar• Modificar
3. El profesor selecciona la opción que desee.	4. Si el profesor selecciona la opción insertar, ver sección “Insertar Pregunta” . 5. Si el usuario selecciona la opción eliminar, ver sección “Eliminar Pregunta” . 6. Si el usuario selecciona la opción modificar, ver sección “Modificar Pregunta” .
Sección Insertar Pregunta	
	1. Muestra la interfaz con 2 pestañas, la primera con las tablas Tema, Pregunta y en la otra la opción para insertar la pregunta.
2. Introduce los datos requeridos para conformar la pregunta y presiona el botón Insertar.	3. El sistema verifica la entrada de datos en los campos si están correctos los adiciona de forma temporal en memoria en caso contrario ver curso

Capítulo 2: Características de la solución propuesta

	<p>alterno 1.</p> <p>4. Activa el panel para insertar las respuestas asociadas a esa pregunta.</p>
Curso alterno de los eventos 1	
	<p>3.1 Si están vacíos o incompletos, muestra un mensaje de error y se retorna al curso normal de los eventos 1.</p>
Sección Eliminar Preguntas	
	<p>1. Muestra la interfaz con 2 pestañas, la primera con las tablas Pregunta, Respuesta y en la otra los paneles para eliminar cada elemento según desee el usuario:</p> <ul style="list-style-type: none">• Eliminar Respuestas dada Pregunta.• Eliminar Respuesta.• Eliminar Pregunta
<p>2. Selecciona la operación “Eliminar Pregunta”.</p> <p>3. El usuario selecciona la pregunta a eliminar de la tabla Pregunta marcándola.</p>	<p>4. Coloca el identificador de la pregunta a eliminar en el campo correspondiente.</p>
<p>5. Presiona el botón Eliminar</p>	<p>6. El sistema muestra un</p>

Capítulo 2: Características de la solución propuesta

	<p>mensaje de confirmación para verificar si desea verdaderamente eliminar la pregunta.</p> <p>7. Si selecciona SI se elimina la pregunta de la base de datos.</p> <p>8. Se actualiza la tabla Pregunta de la interfaz de trabajo.</p>
Curso alterno de los eventos.	
	<p>7.1 Si selecciona NO se retorna a la acción 3.</p>
Sección Modificar Pregunta	
<p>1. El profesor selecciona de la tabla Preguntas, marcando la pregunta que desea modificar.</p>	<p>2. La aplicación carga de forma automática en los campos de textos correspondientes los datos de la pregunta para ser modificados e inactiva ciertos campos que no permiten ser modificados.</p>
<p>3. Modifica los valores de los campos que estén habilitados y presiona el botón Modificar.</p>	<p>4. El sistema verifica la entrada de datos en los campos, si están correctos modifica los valores correspondientes de la base de datos reemplazando los datos existentes por los nuevos parámetros introducidos. En caso contrario ver curso</p>

Capítulo 2: Características de la solución propuesta

	<p>alterno de los eventos.</p> <p>5. Se resetean los campos de textos.</p>
<p>Curso alterno de los eventos.</p>	
	<p>4.1 Si están vacíos o incompletos, muestra un mensaje de error, retorna al curso normal de los eventos 1.</p>
Pos condiciones:	Se gestionan las preguntas.

Tabla No. 12: Caso de Uso 5- Gestionar Respuesta

CU-5	Gestionar Respuesta
Actores:	Profesor
Propósito:	Insertar, modificar así como eliminar Respuestas.
<p><u>Resumen:</u> Permite al usuario insertar las respuestas. Además de eliminar y modificar las ya existentes en la base de datos. Esto está sujeto a la acción que desee realizar el usuario una vez autenticado en la aplicación.</p>	
Referencias:	RF # 5

Capítulo 2: Características de la solución propuesta

Precondiciones:	Debe existir una conexión con la base de datos previamente y las tablas de la base de datos deben contener elementos. Para insertar las respuestas debe haber insertado la pregunta anteriormente.	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>	
1. El profesor selecciona el menú Gestionar Información.	2. El sistema muestra el panel correspondiente de las operaciones sobre el sistema de preguntas <ul style="list-style-type: none"> • Insertar • Eliminar • Modificar 	
3. El profesor selecciona la opción que desee	4. Si el profesor selecciona la opción insertar, ver sección “Insertar” . 5. Si el usuario selecciona la opción eliminar, ver sección “Eliminar” . 6. Si el usuario selecciona la opción modificar, ver sección “Modificar” .	
Sección Insertar		
1. El profesor inserta los datos correspondientes en el panel Insertar Respuesta y presiona el	2. El sistema verifica la entrada de datos en los campos, si están correctos los adiciona	

Capítulo 2: Características de la solución propuesta

botón Insertar.	de forma temporal en memoria en caso contrario ver curso alterno de los eventos 1 . 3. Se resetean los campos de textos para adicionar nuevos valores.
4. Si no desea insertar más respuestas presiona el botón Guardar Información.	5. Activa el panel para insertar el sistema de preguntas y desactiva el panel de las respuestas.
6. Inserta los datos necesarios para conformar el sistema de preguntas en los campos correspondientes y presiona el botón Insertar.	7. El sistema verifica la entrada de datos en los campos, si están correctos los inserta en la base de datos en caso contrario ver curso alterno de los eventos 2 . 8. Inactiva los campos de textos dejando activado solamente el botón Insertar Nueva Pregunta.
Curso alterno de los eventos 1	
	2.1 Si están vacíos o incompletos, muestra un mensaje de error y retorna al curso normal de los eventos 1.

Capítulo 2: Características de la solución propuesta

Curso alterno de los eventos 2	
	7.1 Si están vacíos o incompletos, muestra un mensaje de error y retorna al curso normal de los eventos 6.
Sección Eliminar	
	<p>1. Muestra la interfaz con 2 pestañas, la primera con las tablas Pregunta, Respuesta y en la otra los paneles para eliminar cada elemento según desee el usuario:</p> <ul style="list-style-type: none">• Eliminar Respuestas dada Pregunta.• Eliminar Respuesta.
2. El profesor escoge la operación que desee.	<p>3. Si desea eliminar respuesta ver sección “Eliminar Respuesta”.</p> <p>4. Si desea eliminar respuesta dada pregunta ver sección “Eliminar Respuestas dada Pregunta”.</p>
Sección Eliminar Respuesta	

Capítulo 2: Características de la solución propuesta

1. El usuario introduce el identificador de la pregunta en el campo situado cerca de la tabla de la respuesta para realizar la búsqueda y presiona el botón Buscar.	2. Busca en la base de datos las respuestas asociadas a esa pregunta.
3. Selecciona la respuesta a eliminar marcándola en la tabla de Respuestas.	4. Coloca automáticamente el identificador en el campo de texto correspondiente.
5. Presiona el botón Eliminar.	6. El sistema muestra un mensaje de confirmación para verificar si desea verdaderamente eliminar la respuesta. 7. Si selecciona SI se elimina de la base de datos la respuesta seleccionada, en caso contrario ver curso alternativo de los eventos . 8. Se actualizan los datos en la tabla Respuestas.
Curso alternativo de los eventos	
	7.1 Si selecciona NO se retorna a la acción 3.
Sección Eliminar Respuestas dada Pregunta	
1. El profesor selecciona de la tabla Preguntas la pregunta a la cual desea	2. La aplicación carga automáticamente en el campo de texto correspondiente el

Capítulo 2: Características de la solución propuesta

eliminarle todas las respuestas	identificador de la pregunta de la cual se desea eliminar todas sus respuestas.
3. El profesor presiona el botón Eliminar.	<p>4. El sistema muestra un mensaje de confirmación para verificar si desea verdaderamente eliminar las respuestas.</p> <p>5. Si selecciona Si se eliminan de la base de datos todas las respuestas asociadas a la pregunta seleccionada, en caso contrario ver curso alternativo de los eventos.</p> <p>6. Se actualiza la tabla Respuestas de la base de datos.</p> <p>7. Se resetea el campo del identificador para eliminar otro si así lo desea.</p>
Curso alternativo de los eventos	
	5.1 Si selecciona NO se retorna a la acción 1.
Sección Modificar	
	1. Muestra la interfaz con 2 pestañas, la primera con las tablas Pregunta, Respuesta y en la otra los paneles para modificar cada elemento

Capítulo 2: Características de la solución propuesta

	<p>según desee el usuario:</p> <ul style="list-style-type: none"> • Modificar Respuesta. • Modificar Pregunta
<p>2. El profesor selecciona la acción “Modificar Respuesta”.</p> <p>3. El profesor selecciona de la tabla la respuesta que desea modificar marcando la misma.</p>	<p>4. La aplicación carga automáticamente en los campos de textos correspondientes los datos para ser modificados e inactiva ciertos campos que no permiten ser modificados.</p>
<p>5. Modifica los valores de los campos que estén habilitados y presiona el botón Modificar.</p>	<p>6. El sistema verifica la entrada de datos en los campos, si están correctos, reemplaza los valores existentes por los nuevos insertados, en caso contrario ver curso alternativo de los eventos.</p> <p>7. Se resetean campos de textos.</p>
Curso alternativo de los eventos	
	<p>6.1 Si están vacíos o incompletos, muestra un mensaje de error, retorna al curso normal de los eventos 3.</p>
Pos condiciones:	Se gestionan las respuestas

Tabla No. 13: Caso de Uso 6- Consultar Información

Capítulo 2: Características de la solución propuesta

CU-6	Consultar Información	
Actores:	Profesor	
Propósito:	Realizar una búsqueda en la base de datos de acuerdo a determinados criterios de búsqueda.	
<p><u>Resumen:</u> Permite al usuario buscar cualquier asignatura, tema, pregunta, respuesta o sistema de preguntas dentro de la base de datos de acuerdo a algunos criterios de búsqueda que se especifican en la aplicación. El usuario inserta los valores en los campos correspondientes y presiona el botón buscar, no necesariamente el usuario debe llenar todos los campos de textos para realizar la búsqueda.</p>		
Referencias:	RF # 6	
Precondiciones:	Debe existir una conexión con la base de datos previamente y las tablas de la base de datos deben contener elementos.	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>	
1. El profesor selecciona el menú Gestionar Información.	2. El sistema muestra la sección Consultar con la opción para seleccionar cuales de los elementos desea consultar. <ul style="list-style-type: none"> • Consultar Asignatura. • Consultar Tema. • Consultar Pregunta • Consultar Respuesta. • Consultar Sistema de 	

Capítulo 2: Características de la solución propuesta

	Preguntas
3. El profesor selecciona la opción que desee y presiona el botón Siguiente.	<p>4. Si selecciona la opción Consultar Asignatura ver sección “Consultar Asignatura”.</p> <p>5. Si selecciona la opción Consultar Tema ver sección “Consultar Tema”.</p> <p>6. Si selecciona la opción Consultar Pregunta ver sección “Consultar Pregunta”.</p> <p>7. Si selecciona la opción Consultar Respuesta ver sección “Consultar Respuesta”.</p> <p>8. Si selecciona la opción Consultar Sistema de Preguntas ver sección “Consultar Sistema de Preguntas”.</p>
Sección Consultar Asignatura	
	1. El sistema muestra la sección Consultar Asignatura.
2. Introduce los datos por los criterios de búsqueda que vaya a realizar y presiona el botón Buscar.	3. El sistema verifica la entrada de datos en los campos si están correctos realiza la búsqueda y muestra los

Capítulo 2: Características de la solución propuesta

	<p>resultados en la tabla que aparece en la parte inferior de la interfaz, en caso contrario ver curso alternativo de los eventos.</p> <p>4. Se resetean los campos de textos.</p>
Curso alternativo de los eventos	
	<p>3.1 Si los datos están incorrectos (cambios de tipo de entrada, Ej. texto donde van números etc.), muestra un mensaje de error, se retorna al curso normal de los eventos 1.</p>
Sección Consultar Tema	
	<p>1. Muestra la interfaz con el panel para realizar la búsqueda del Tema deseado.</p>
<p>2. Introduce los datos por los criterios de búsqueda que vaya a realizar y presiona el botón Buscar.</p>	<p>3. El sistema verifica la entrada de datos en los campos, si están correctos realiza la búsqueda y muestra los resultados en la tabla que aparece en la parte inferior de la interfaz, en caso contrario ver curso alternativo de los</p>

Capítulo 2: Características de la solución propuesta

	eventos. 4. Se resetean los campos de textos.
Curso alternativo de los eventos	
	3.1 Si los datos están incorrectos (cambios de tipo de entrada, Ej. texto donde van números etc.), muestra un mensaje de error, se retorna al curso normal de los eventos 1.
Sección Consultar Preguntas	
	1. Muestra la forma con el panel para realizar la búsqueda.
2. Introduce los datos por los criterios de búsqueda que vaya a realizar (no es necesario que llene todos los campos) y presiona el botón Buscar.	3. El sistema verifica la entrada de datos en los campos, si están correctos realiza la búsqueda y muestra los resultados en la tabla que aparece en la parte inferior de la interfaz, en caso contrario ver curso alternos de los eventos. 4. Se resetean los campos de textos.
Curso alternativo de los eventos	
	3.1 Si los datos están incorrectos

Capítulo 2: Características de la solución propuesta

	(cambios de tipo de entrada, Ej. texto donde van números etc.), muestra un mensaje de error, se retorna al curso normal de los eventos 1.
Sección Consultar Respuestas	
	1. Muestra la forma con el panel para realizar la búsqueda.
2. Selecciona del combobox la pregunta que desea.	3. Se carga automáticamente en el campo correspondiente el identificador de la pregunta.
4. Presiona el botón Buscar.	5. Muestra en la tabla las respuestas asociadas a la pregunta seleccionada por el usuario. 6. Resetea el campo del identificador de la pregunta.
Sección Consultar Sistema de Preguntas	
	1. Muestra la forma con el panel para realizar la búsqueda.
2. Introduce los datos por los criterios de búsqueda que vaya a realizar (no es necesario que llene todos los campos) y presiona el botón Buscar.	3. El sistema verifica la entrada de datos en los campos, si están correctos realiza la búsqueda y muestra los resultados en la tabla que aparece en la parte inferior de

Capítulo 2: Características de la solución propuesta

	la interfaz, en caso contrario ver curso alternativo de los eventos. 4. Se resetean los campos de textos.
Curso alternativo de los eventos	
	3.1 Si los datos están incorrectos (cambios de tipo de entrada, Ej. Texto donde van números etc.), muestra un mensaje de error, se retorna al curso normal de los eventos 1.
Pos condiciones:	Se obtiene una detallada búsqueda a partir de los distintos criterios introducidos por el usuario para consultar.

En el presente capítulo fueron expuestas las principales características del sistema basándose principalmente en “lo que hace” y no “en cómo lo hace”, quedaron definidas las principales funcionalidades que debe tener el software a desarrollar, mostradas a través de sus requisitos funcionales, estructurándolos en casos de usos así como realizando una descripción textual de cada uno de ellos, proporcionando todo esto una visión general de lo que se debe hacer.

Capítulo 3: Descripción de la solución propuesta

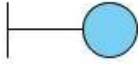
En este capítulo serán abordados los principales aspectos del Análisis y Diseño de la herramienta para la gestión de asignaturas Virtual_Know a través de los diagramas de clases del diseño, de clases del análisis, además de los diagramas de interacción y del modelo entidad-relación como representación del diseño de la base de datos, se explica el flujo de los procesos y se ofrece una representación de los Casos de Uso más significativos representados en UML y diseñados con la herramienta Visual Paradigm.

3.1 Características fundamentales del análisis

Análisis: Consiste en obtener una visión del sistema que se preocupa de ver “qué hace”, de modo que solo se interesa por los requisitos funcionales. Las actividades dentro del análisis se hacen con el objetivo fundamental de facilitar la entrada al diseño, por lo que son un paso inicial y una primera aproximación conceptual para que, una vez comprendidos los requisitos a este nivel, se pueda aumentar el nivel de especificidad en aras de garantizar el cubrimiento de los requisitos funcionales y no funcionales **(9)**.

Las clases que se utilizan para el modelado de los diagramas de clases de análisis son de tres tipos: Interfaz, Control y Entidad, las que representan conceptos y relaciones del dominio, poseen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización así como tipos asociativos **(9)**.

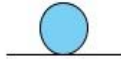
Clase Interfaz: Modelan la interacción entre el sistema y sus actores.



Clase Control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.



Clase Entidad: Modelan la información que posee larga vida y que es a menudo persistente.



3.1.1 Diagrama de clases del análisis.

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real y no de la implementación automatizada que dicho problema requiere (9).

Figura No. 3: Diagrama de clases del análisis del CU - Autenticar Usuario



Capítulo 3: Descripción de la solución propuesta

Figura No. 4: Diagrama de clases del análisis del CU - Gestionar Asignatura

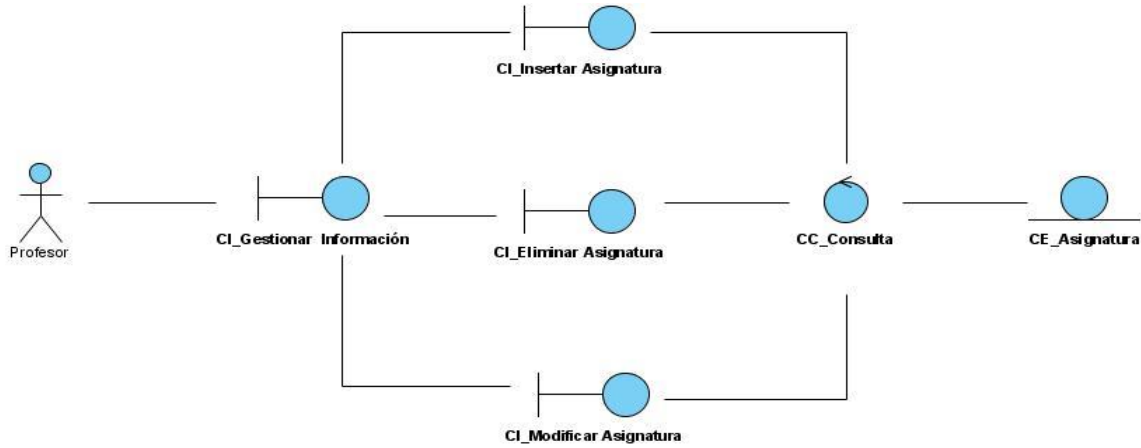


Figura No. 5: Diagrama de clases del análisis del CU - Gestionar Tema

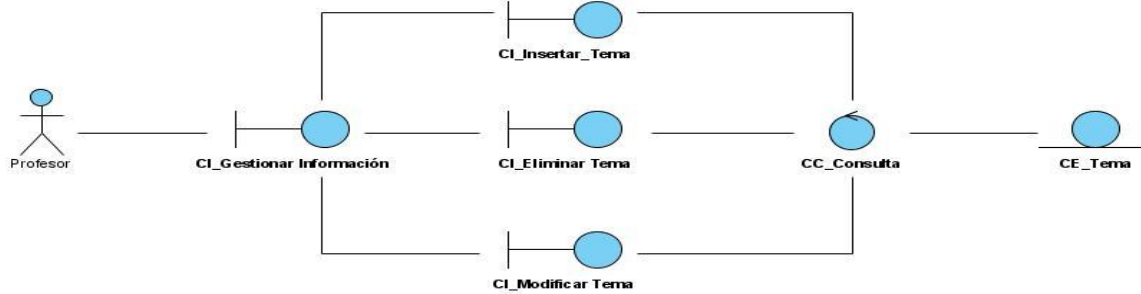
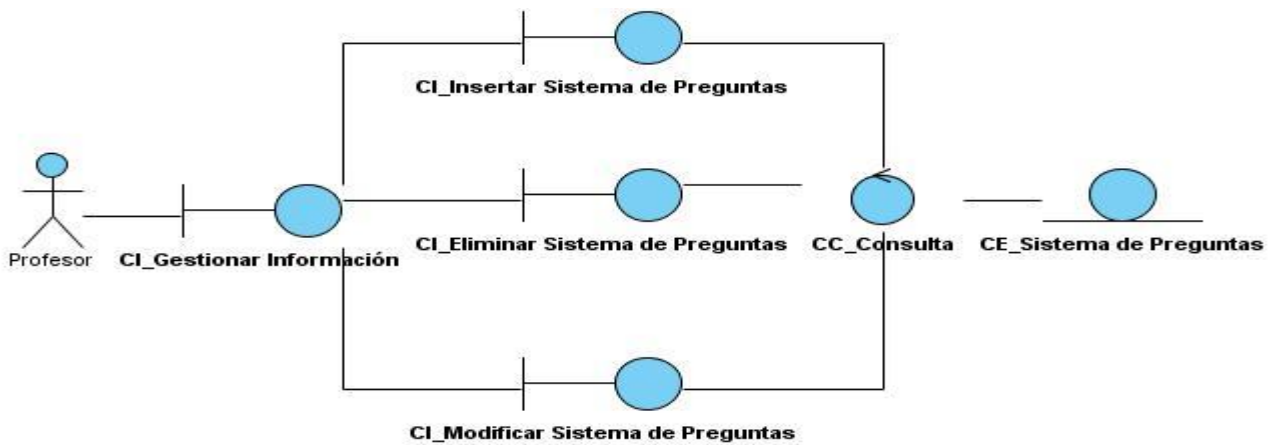


Figura No. 6: Diagrama de clases del análisis del CU - Gestionar Pregunta



Capítulo 3: Descripción de la solución propuesta

Figura No. 7: Diagrama de clases del análisis del CU - Gestionar Respuesta

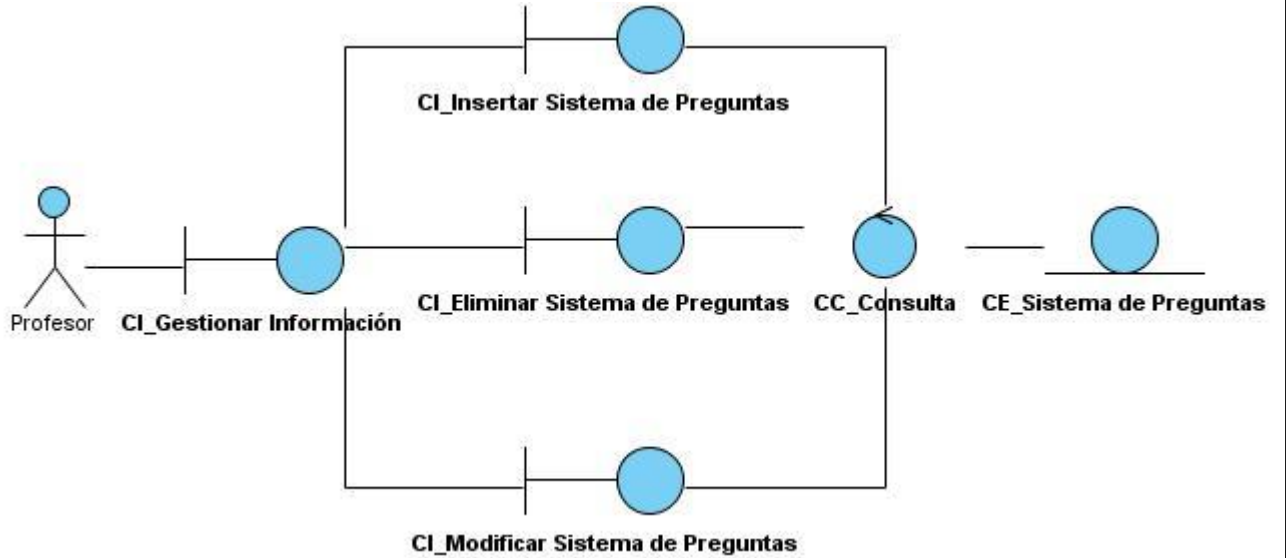
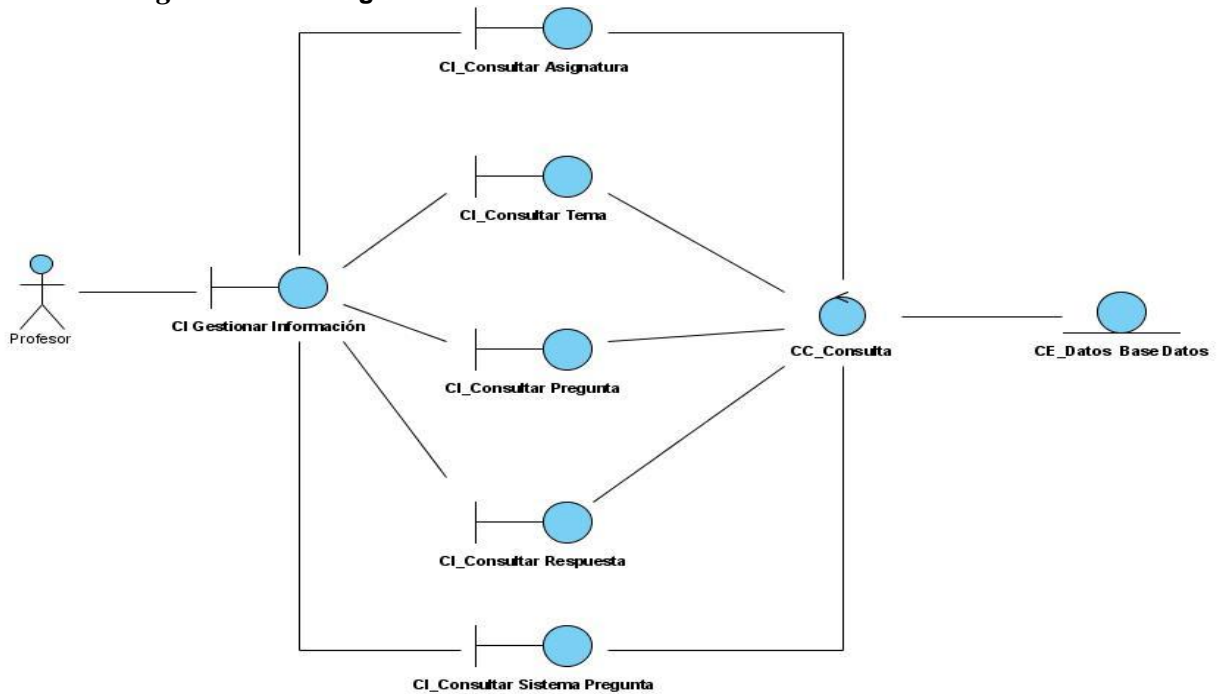


Figura No. 8: Diagrama de clases del análisis del CU - Consultar Información



3.2 Características fundamentales del Diseño.

Diseño: Es un refinamiento del Análisis que tiene en cuenta los requisitos no funcionales, es decir como cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Tiene como propósitos fundamentales transformar los requerimientos en un diseño de cómo debe ser el sistema y adaptarlo para que se corresponda con el entorno de implementación, diseñando sus funcionalidades **(9)**.

3.2.1 Diagrama de Interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que trae consigo modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos junto con los mensajes enviados entre ellos, todo esto es denominado flujo de control. Estas operaciones se llevan a cabo en el contexto de un escenario que ilustra un comportamiento de dicho sistema. Existen dos tipos de diagramas de interacción en UML, los Diagramas de Colaboración (dimensión estructural) y los Diagramas de Secuencia (dimensión temporal).

Diagramas de Colaboración: Destacan la organización de los objetos que participan en una interacción, se construyen colocando en primer lugar los objetos que participan en la colaboración como nodos del grafo, a continuación se representan los enlaces que conectan esos objetos como arcos del grafo y por último estos enlaces se adornan con los mensajes que envían y reciben los objetos. Poseen dos características que los distinguen de los diagramas de secuencia, una de ellas es el camino, el cual indica cómo se enlaza un objeto a otro y la otra es el número de secuencia el que indica la ordenación temporal de un mensaje o el orden en que estos se llevan a cabo **(9)**.

Capítulo 3: Descripción de la solución propuesta

Figura No. 9: Diagrama de colaboración del CU - Autenticar Usuario

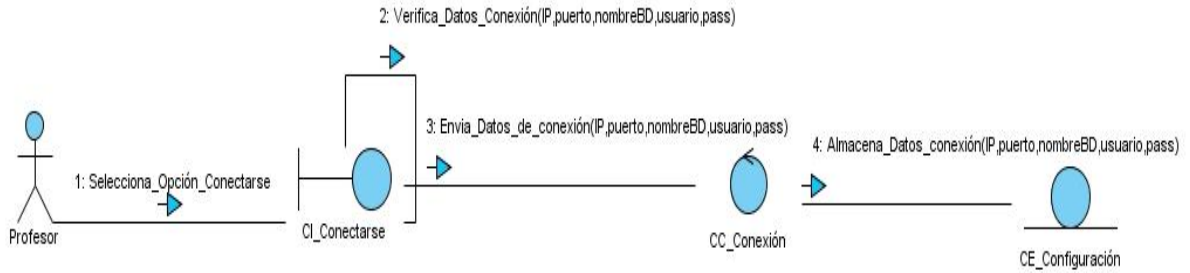
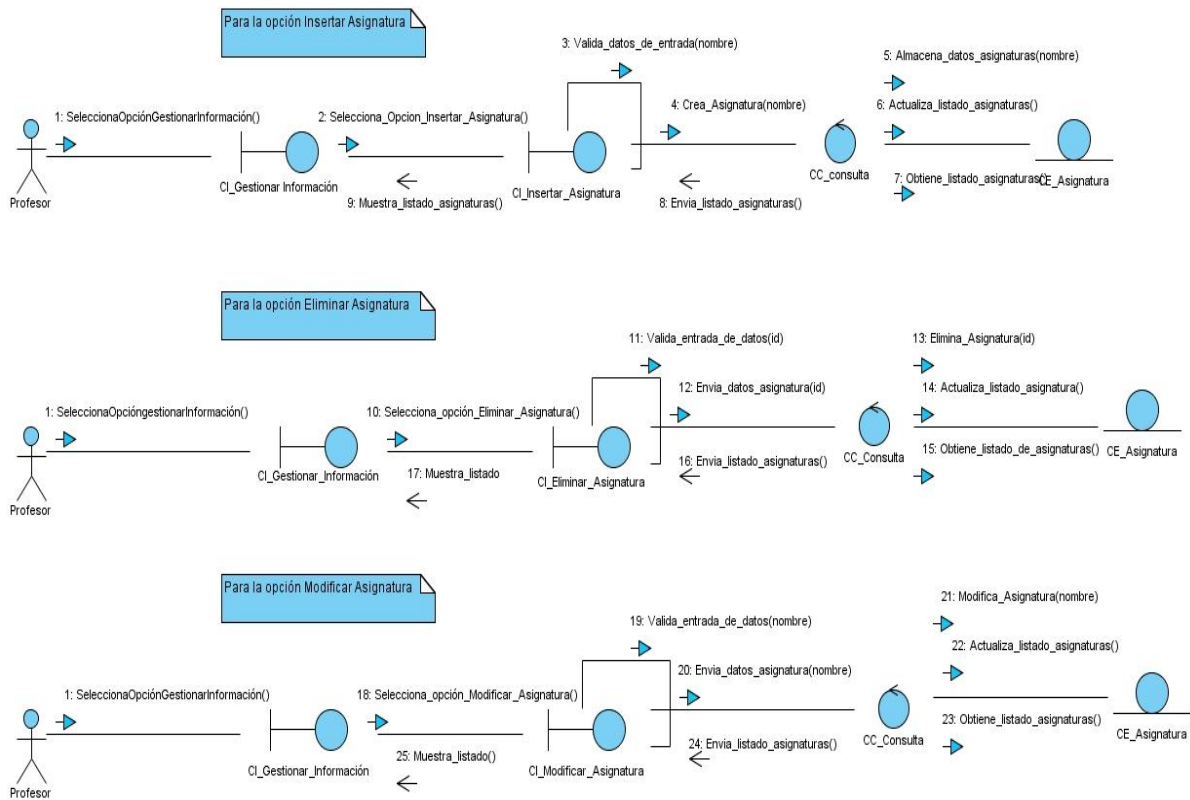
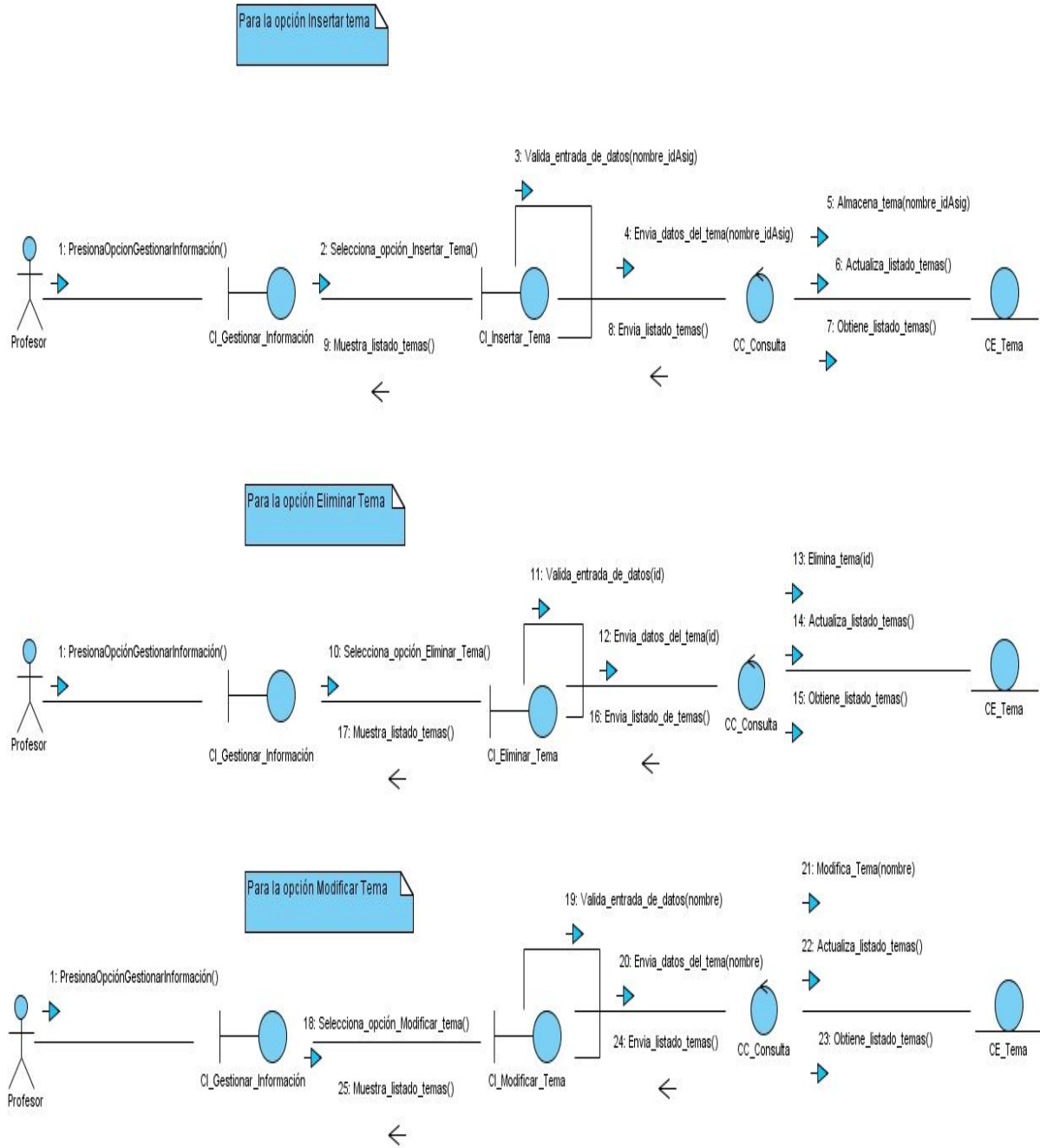


Figura No. 10: Diagrama de colaboración del CU - Gestionar Asignatura



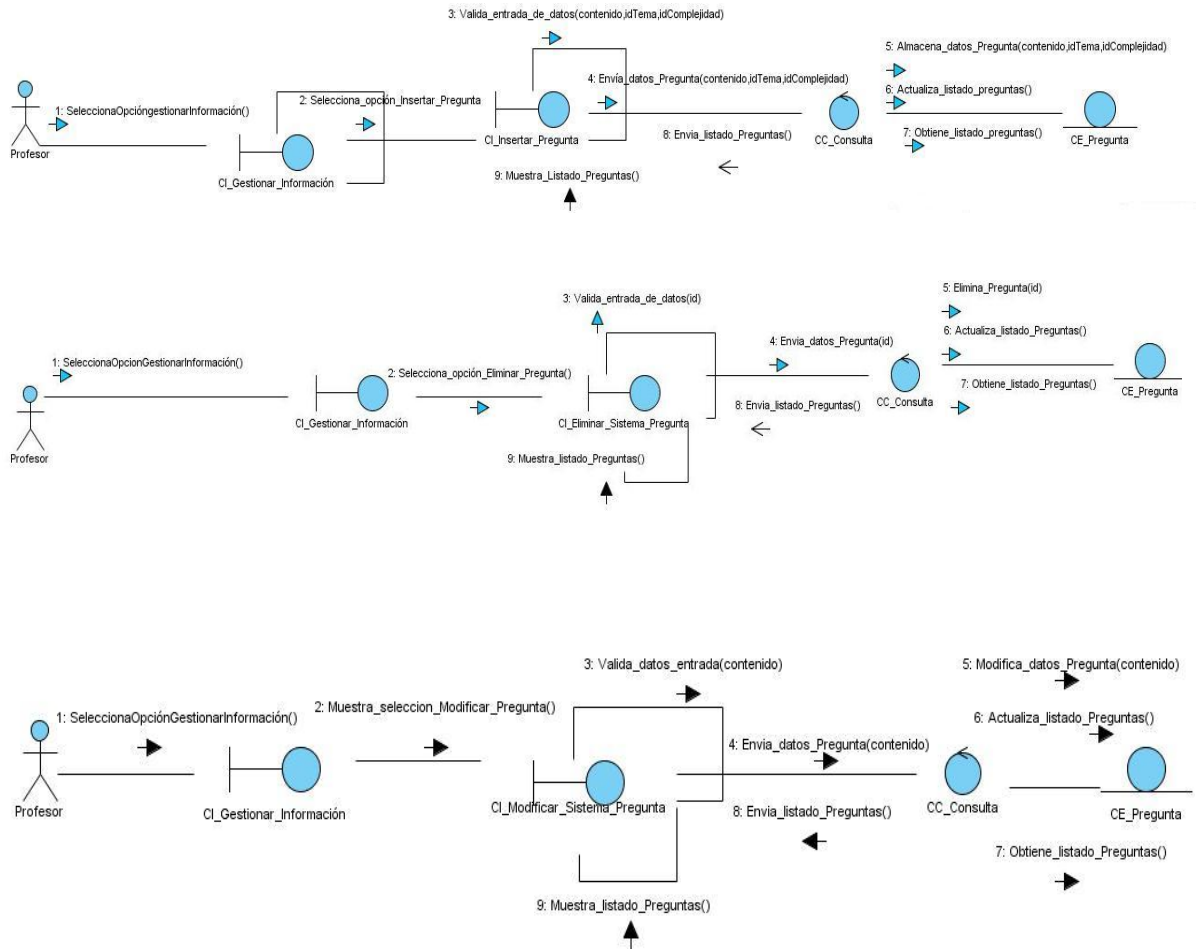
Capítulo 3: Descripción de la solución propuesta

Figura No. 11: Diagrama de colaboración del CU - Gestionar Tema



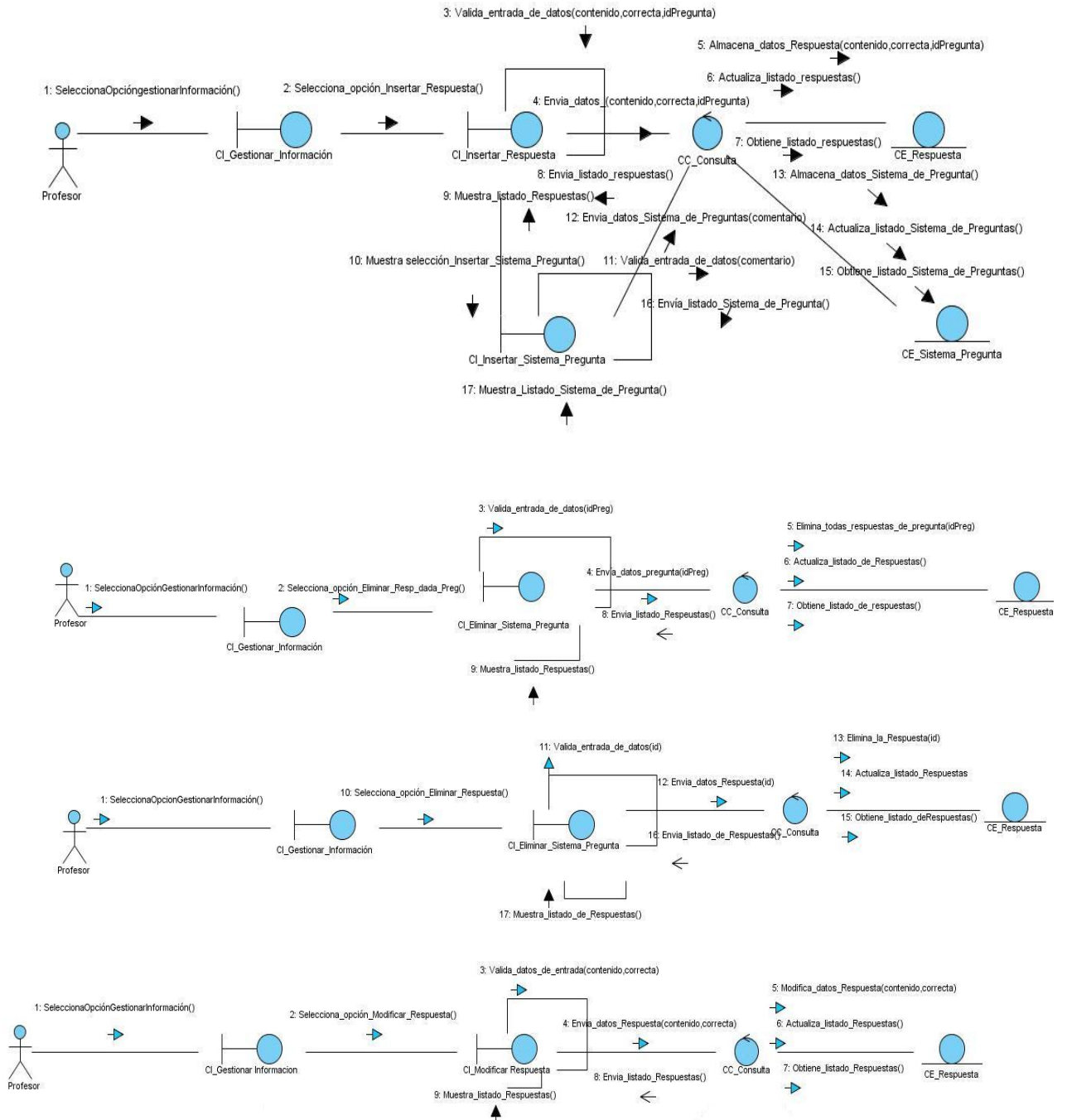
Capítulo 3: Descripción de la solución propuesta

Figura No. 12: Diagrama de colaboración del CU - Gestionar Pregunta



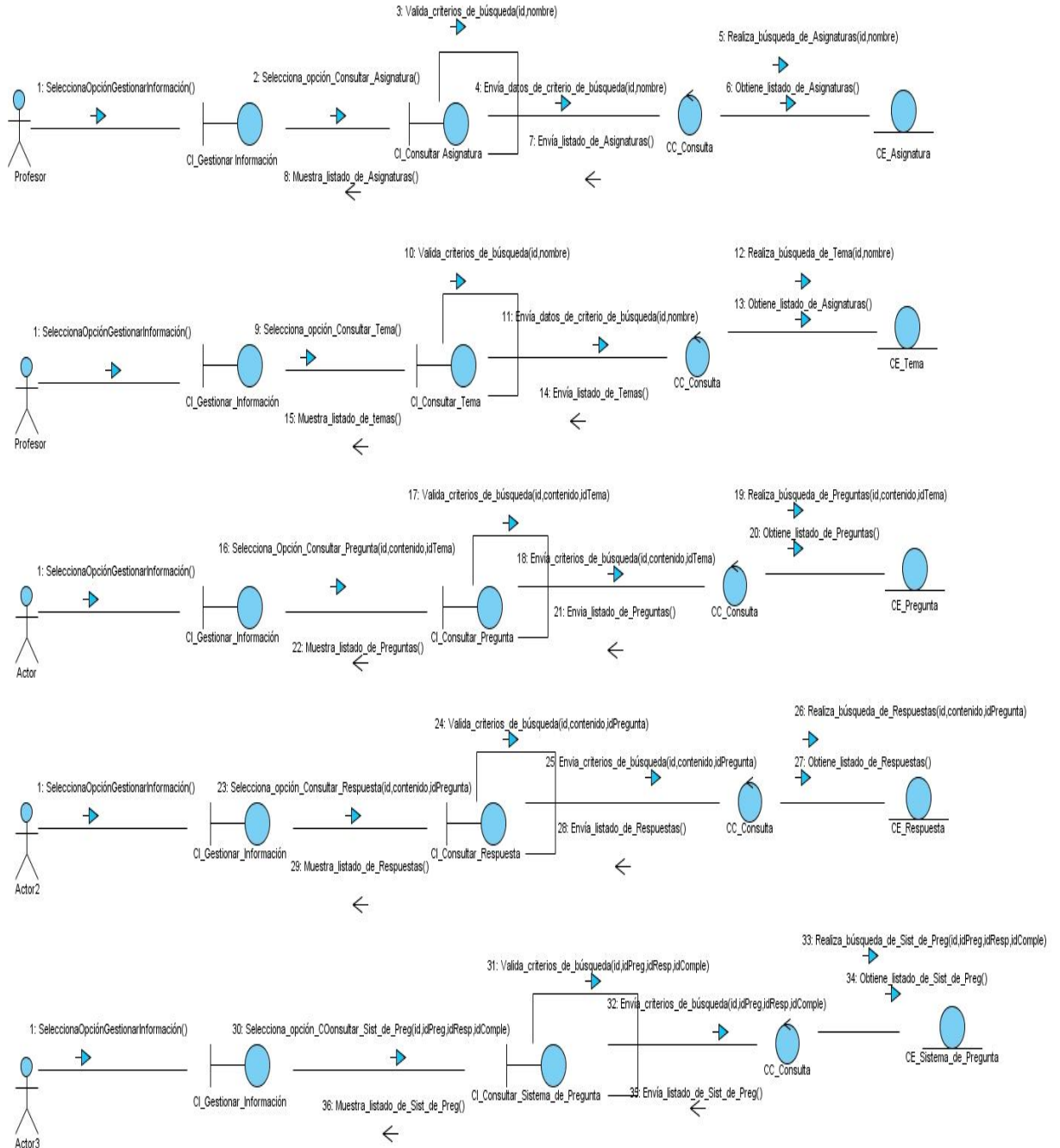
Capítulo 3: Descripción de la solución propuesta

Figura No. 13: Diagrama de colaboración del CU - Gestionar Respuesta



Capítulo 3: Descripción de la solución propuesta

Figura No. 14: Diagrama de colaboración del CU - Consultar Información



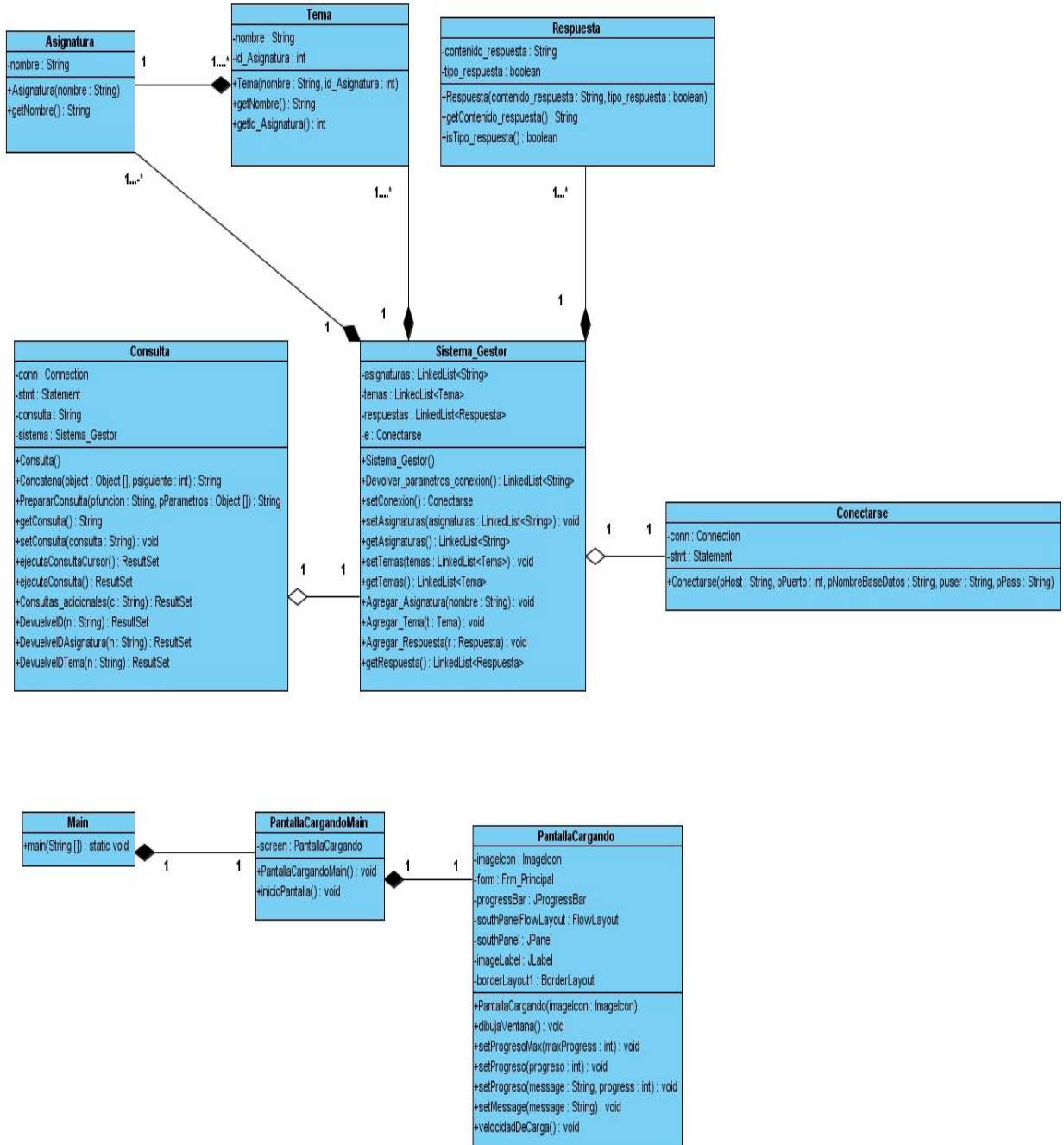
3.2.2 Diagrama de clases del diseño.

Un diagrama de clases del diseño es un diagrama que describe gráficamente las especificaciones de las clases existentes en un producto de software así como de las interfaces involucradas. Estos diagramas contienen información útil para el usuario como por ejemplo clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, información sobre los tipos de atributos, navegabilidad y dependencia existente. A diferencia de un modelo conceptual, un diagrama de clases del diseño contiene las definiciones de las entidades del software en vez de conceptos del mundo real. Pero como en UML no está bien definido el concepto de diagrama de clases diseño, este se sirve de un término genérico denominado Diagrama de Clases **(10)**.

En el siguiente diagrama de clases se observan las clases que dan solución al problema planteado así como sus relaciones, encontrándose la clase Asignatura, Tema, Respuesta, Consulta, Conectarse, Sistema Gestor, Main, PantallaCargando y PantallaCargandoMain respectivamente, las cuales forman el eje central de la aplicación Virtual_Know debido a la fuerte relación existente entre ellas.

Capítulo 3: Descripción de la solución propuesta

Figura No. 15: Diagrama de clases del diseño



3.2.3 Descripción de las clases.

A continuación se realiza una detallada descripción de las clases más importantes que conforman el sistema, especificándose en cada caso su composición a través de sus atributos y las operaciones que realizan.

Tabla No. 14: Descripción de la clase Asignatura

Nombre: Asignatura	
Tipo de clase: Interface	
Atributo	Tipo
nombre	String
Para cada responsabilidad:	
Nombre:	Asignatura(String nombre)
Descripción:	Inicializa los atributos de este objeto con los parámetros recibidos.
Nombre:	getNombre()
Descripción:	Obtiene el nombre de una asignatura.

Tabla No. 15: Descripción de la clase Tema

Nombre: Tema
Tipo de clase: Interface

Capítulo 3: Descripción de la solución propuesta

Atributo	Tipo
nombre	String
id_asignatura	int
Para cada responsabilidad:	
Nombre:	Tema(String nombre, int id_asignatura)
Descripción:	Inicializa los atributos de este objeto con los parámetros recibidos
Nombre:	getNombre()
Descripción:	Obtiene el nombre de un tema.
Nombre:	getId_Asignatura()
Descripción:	Obtiene identificador de la asignatura a la cual está ligado el tema.

Tabla No. 16: Descripción de la clase Respuesta

Nombre: Respuesta	
Tipo de clase: Interface	
Atributo	Tipo
contenido_respuesta	String

Capítulo 3: Descripción de la solución propuesta

tipo_respuesta	boolean
Para cada responsabilidad:	
Nombre:	Respuesta(String contenido_respuesta, boolean tipo_respuesta)
Descripción:	Inicializa los atributos de este objeto con los parámetros recibidos
Nombre:	getContenido_respuesta()
Descripción:	Obtiene el contenido o texto de la respuesta.
Nombre:	isTipo_respuesta()
Descripción:	Devuelve el tipo de la respuesta, si es correcta o falsa.

Tabla No. 17: Descripción de la clase Sistema_Gestor

Nombre: Sistema_Gestor	
Tipo de clase: Controladora	
Atributo	Tipo
asignaturas	LinkedList<String>
temas	LinkedList<Tema>
respuestas	LinkedList<Respuesta>

Capítulo 3: Descripción de la solución propuesta

c	Conectarse
Para cada responsabilidad:	
Nombre:	Sistema_Gestor()
Descripción:	Inicializa los parámetros de la clase.
Nombre:	Devolver_parámetros_conexión()
Descripción:	Devuelve en una lista los parámetros de la conexión previamente establecida extrayéndolos del fichero de configuración config.txt.
Nombre:	setConexión()
Descripción:	Establece la conexión con la base de datos utilizando para ello los datos extraídos previamente del fichero
Nombre:	setAsignaturas(LinkedList<String> asignaturas)
Descripción:	Iguala el parámetro local asignaturas a la lista que recibe por parámetro.
Nombre:	getAsignaturas()
Descripción:	Devuelve una lista con las asignaturas existentes para posteriormente almacenarlas en la base de datos.
Nombre:	getTemas()

Capítulo 3: Descripción de la solución propuesta

Descripción:	Devuelve una lista con los temas existentes en memoria.
Nombre:	setTemas(LinkedList<String> asignaturas)
Descripción:	Iguala el parámetro local temas a la lista que recibe por parámetro.
Nombre:	Agregar_Asignatura(String nombre)
Descripción:	Añade a la lista de asignaturas el nuevo valor introducido por parámetro.
Nombre:	Agregar_Tema(Tema t)
Descripción:	Añade a la lista de temas el nuevo valor introducido por parámetro.
Nombre:	Agregar_Respuesta(Respuesta r)
Descripción:	Añade a la lista de Respuestas el nuevo valor introducido por parámetro.
Nombre:	getRespuestas()
Descripción:	Devuelve una lista con las respuestas existentes en memoria.

Capítulo 3: Descripción de la solución propuesta

Tabla No. 18: Descripción de la clase Consulta

Nombre: Consulta	
Tipo de clase: Controladora	
Atributo	Tipo
conn	Connection
stmt	Statement
consulta	String
sistema	Sistema_Gestor
Para cada responsabilidad:	
Nombre:	Consulta()
Descripción:	Inicializa los atributos de la clase.
Nombre:	concatena(Object [] Object, int pSiguiete)
Descripción:	Concatena las distintas partes de la consulta que se formula preparándola para llevarla a cabo.

Capítulo 3: Descripción de la solución propuesta

Nombre:	prepararConsulta(String pFuncion, Object [] pParametros)
Descripción:	Prepara la consulta que se va a ejecutar.
Nombre:	getConsulta()
Descripción:	Devuelve la consulta.
Nombre:	setConsulta(String consulta)
Descripción:	Cambia el valor del atributo de la clase por el valor que recibe por parámetro.
Nombre:	ejecutaConsultaCursor()
Descripción:	Función que ejecuta la consulta con cursores siendo esta previamente generada en la funcionalidad prepararConsulta.
Nombre:	ejecutaConsulta()
Descripción:	Ejecuta la consulta previamente generada en la funcionalidad prepararConsulta pero esta vez sin los cursores.

Capítulo 3: Descripción de la solución propuesta

Nombre:	Consultas_adicionales()
Descripción:	Ejecuta una consulta simple en la base de datos sin necesidad de pasar por las funcionalidades anteriores.
Nombre:	DevuelveIDpregunta(String n)
Descripción:	Devuelve el identificador de la pregunta cuyo contenido coincide con el parámetro introducido.
Nombre:	DevuelveIDAsignatura(String n)
Descripción:	Devuelve el identificador de la asignatura cuyo nombre coincide con el parámetro recibido.
Nombre:	DevuelveIDTema(String n)
Descripción:	Devuelve el identificador del tema cuyo nombre coincide con el parámetro recibido.

Capítulo 3: Descripción de la solución propuesta

Tabla No. 19: Descripción de la clase Main

Nombre: Main	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Main(String [] args)
Descripción:	Inicializa la interfaz de la pantalla de bienvenida del sistema.

Tabla No. 20: Descripción de la clase PantallaCargandoMain

Nombre: PantallaCargandoMain	
Tipo de clase: Interface	
Atributo	Tipo
screen	PantallaCargando
Para cada responsabilidad:	
Nombre:	PantallaCargandoMain()
Descripción:	Inicializa los atributos de la clase.
Nombre:	inicioPantalla()
Descripción:	Establece los valores de la pantalla de inicio como son imagen, posición etc.

Capítulo 3: Descripción de la solución propuesta

Tabla No. 21: Descripción de la clase *PantallaCargandoMain*

Nombre: PantallaCargandoMain	
Tipo de clase: Interface	
Atributo	Tipo
imapelcon	Imapelcon
form	Frm_Principal
progressBar	JProgressBar
southPanelFlowLayout	FlowLayout
southPanel	JPanel
imageLabel	JLabel
borderLayout1	BordeLayout
Para cada responsabilidad:	
Nombre:	PantallaCargando(Imapelcon imapelcon)
Descripción:	Inicializa los atributos de la clase y del objeto con los parámetros recibidos.

Capítulo 3: Descripción de la solución propuesta

Nombre:	dibujaVentana()
Descripción:	Dibuja los contornos de la ventana indicándoles posición, color, etc.
Nombre:	setProgresoMax(int maxProgress)
Descripción:	Establece el máximo valor al cual llegará la barra de progreso.
Nombre:	setProgreso(int progreso)
Descripción:	Establece el valor que irá tomando progresivamente la barra de progreso.
Nombre:	setProgreso(String message, int progreso)
Descripción:	Establece los números de progreso de la barra y un mensaje de fondo de la misma (ej. Loading o Cargando)
Nombre:	setMessage(String message)
Descripción:	Establece solamente el mensaje que tendrá de fondo la barra de progreso.
Nombre:	VelocidadDeCarga()

Capítulo 3: Descripción de la solución propuesta

Descripción:	Establece la velocidad a la que se irá completando la barra de progreso.
---------------------	--

Tabla No. 22: Descripción de la clase Conectarse

Nombre: Conectarse	
Tipo de clase: Controladora	
Atributo	Tipo
conn	Connection
stmt	Statement
Para cada responsabilidad:	
Nombre:	Conectarse(String pHost, int Ppuerto, String pNombreBaseDatos, String pUser, String pPass)
Descripción:	Establece la conexión con la BD

3.3 Modelo Entidad-Relación

El modelo entidad-relación es un modelo conceptual o teórico que se utiliza para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para dicho sistema así como sus interrelaciones y propiedades, están

Capítulo 3: Descripción de la solución propuesta

basados además en una percepción de un mundo real que consiste en un grupo de elementos básicos llamados entidades y relaciones, dichas relaciones se llevan a cabo a través de la cardinalidad.

1. Entidad: Es un elemento o conjunto de estos que existen y se distinguen de otros elementos gracias a que tienen características propias (atributos) que los describen.

2. Atributos: Son las características por las cuáles se puede describir una entidad y que a su vez no tienen características propias representativas, pueden ser únicos de cada entidad o puede estar repetido dentro del conjunto, en caso de ser único recibe el nombre de llave primaria.

3. Llave Primaria: Es un atributo propio distintivo de una sola entidad el cual no puede repetirse en el conjunto de valores.

4. Cardinalidad: Es la forma través de la cual se relacionan las entidades que conforman el modelo entidad-relación. Existen 3 formas básicas de cardinalidad, Uno a Uno (1: 1), Uno a Varias (1: N) y de Varias a Varias (N: N). La cardinalidad depende de las condiciones de la realidad y de la lógica que exista, por cada dos entidades que existan debe establecerse, de existir, la cardinalidad entre ellas **(11)**.

A continuación se representa el modelo entidad-relación de la base de datos que usa la aplicación Virtual_Know en la cual se reflejan las relaciones existentes entre cada tabla así como la cardinalidad entre ellas.

Capítulo 3: Descripción de la solución propuesta

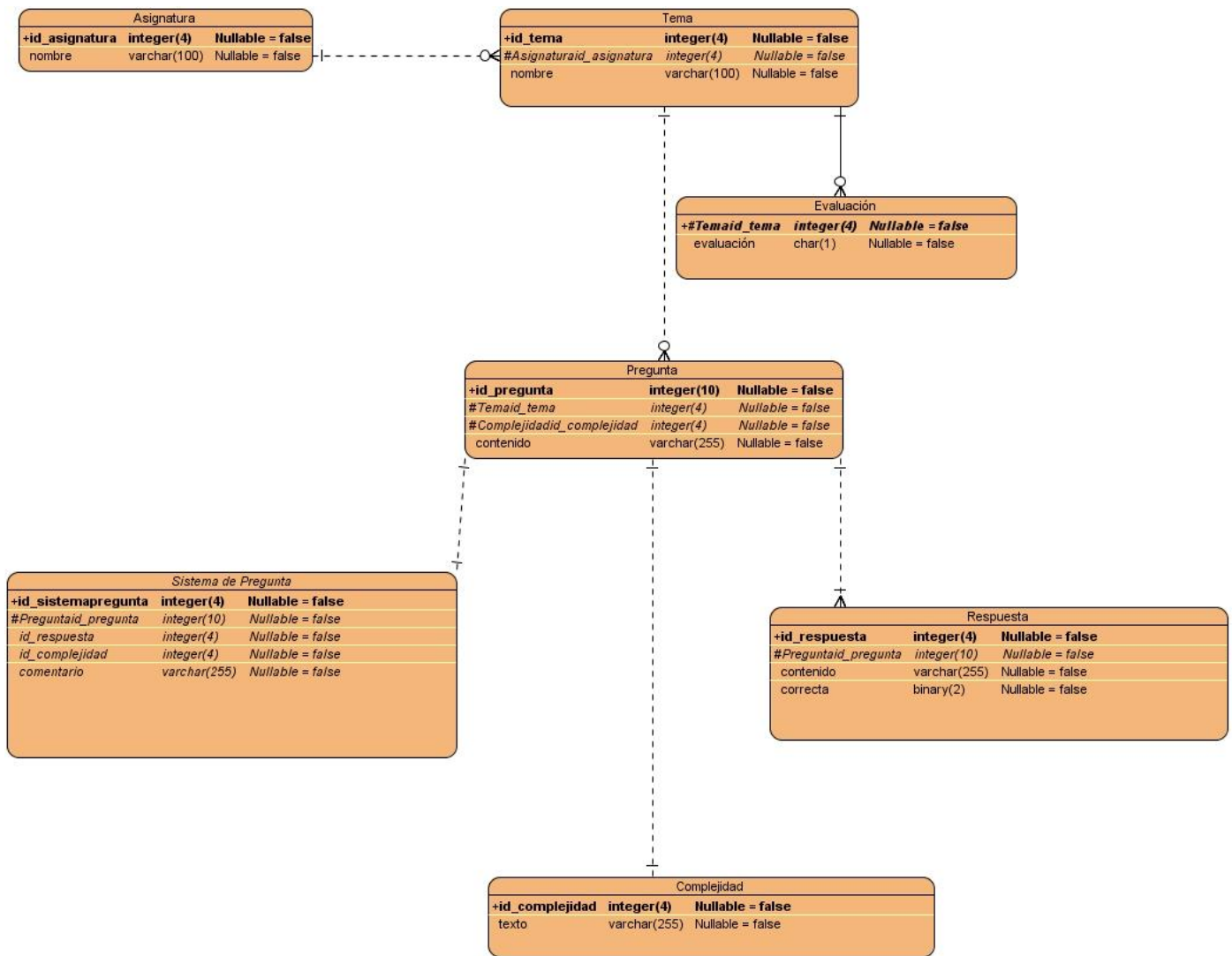


Figura No. 16: Modelo entidad-relación de la Base de Datos del sistema.

3.3.1 Descripción de las tablas

Tabla No. 23: Descripción de la Tabla: Asignatura

Nombre: Asignatura		
Descripción: Contiene todas las asignaturas con sus atributos.		
Atributo	Tipo	Descripción
id_asignatura (PK)	integer (no nulo)	Identificador que se genera e incrementa automáticamente a medida que se insertan nuevas asignaturas en la tabla
p_nombre	character varying (50)	Nombre que tiene la asignatura

Tabla No. 24: Descripción de la Tabla: Tema

Nombre: Tema		
Descripción: Contiene todos los temas de las distintas asignaturas con sus atributos.		
Atributo		Descripción
id_tema (PK)	integer (no nulo)	Identificador que se genera e incrementa

Capítulo 3: Descripción de la solución propuesta

		automáticamente a medida que se insertan nuevos temas en la tabla
id_asignatura (FK)	integer (no nulo)	Llave foránea que viene de la tabla asignatura, indicando a que asignatura pertenece el tema
nombre	character varying (100)	Nombre del tema

Tabla No. 25: Descripción de la Tabla: Complejidad

Nombre: Complejidad		
Descripción: Almacena la complejidad de las preguntas.		
Atributo		Descripción
id_complejidad (PK)	integer (no nulo)	Identificador que se genera e incrementa automáticamente a medida que se insertan nuevas complejidades en la tabla
texto	character varying (40)	Texto de la complejidad

Capítulo 3: Descripción de la solución propuesta

Tabla No. 26: Descripción de la Tabla: Evaluación

Nombre: Evaluación		
Descripción: Almacena la evaluación obtenida en cada tema.		
Atributo		Descripción
id_tema (PK)	integer (no nulo)	Identificador del tema que se desea evaluar
evaluación	character varying (40)	Texto de la evaluación

Tabla No. 27: Descripción de la Tabla: Pregunta

Nombre: Pregunta		
Descripción: Almacena las preguntas de los determinados temas.		
Atributo		Descripción
id_pregunta (PK)	integer (no nulo)	Identificador de la pregunta generado automáticamente cada vez que se inserta una nueva en la tabla.
contenido	text	Contenido de la pregunta
id_tema (FK)	integer (no nulo)	Llave foránea procedente de la tabla

Capítulo 3: Descripción de la solución propuesta

		Tema indicando a que tema pertenece una pregunta determinada.
id_complejidad (FK)	integer (no nulo)	Llave foránea procedente de la Tabla Complejidad indicando la complejidad de la pregunta.

Tabla No. 28: Descripción de la Tabla: Respuesta

Nombre: Respuesta		
Descripción: Almacena las respuesta de las preguntas.		
Atributo		Descripción
id_respuesta (PK)	integer (no nulo)	Identificador de la respuesta generado automáticamente cada vez que se inserta una nueva en la tabla.
contenido	text	Contenido de la respuesta
correcta	boolean	Indica si la respuesta de la pregunta es correcta o no.
id_pregunta (FK)	integer (no nulo)	Llave foránea

Capítulo 3: Descripción de la solución propuesta

		procedente de la Tabla Pregunta indicando a que pregunta corresponde la respuesta
--	--	---

Tabla No. 29: Descripción de la Tabla: Sistema_Pregunta

Nombre: Sistema_Pregunta		
Descripción: Almacena los sistemas de preguntas conformados con preguntas, respuestas y complejidad.		
Atributo		Descripción
id_sistemapregunta (PK)	integer (no nulo)	Identificador del sistema de pregunta generado automáticamente cada vez que se inserta uno nuevo en la tabla.
id_pregunta (FK)	integer (no nulo)	Llave foránea procedente de la Tabla Pregunta indicando el identificador de la pregunta que conforma el sistema de pregunta.
id_respuesta	integer (no nulo)	Llave foránea procedente de la Tabla Respuesta indicando el

Capítulo 3: Descripción de la solución propuesta

		identificador de la respuesta que conforma el sistema de pregunta.
id_complejidad	integer (no nulo)	Llave foránea procedente de la Tabla Complejidad indicando la complejidad del sistema de pregunta.
comentario	character varying(100)	Breve descripción del sistema de pregunta

En el presente capítulo se realizó una detallada explicación de la herramienta Virtual_Know desde el punto de vista del análisis y del diseño, realizando para ello los diagramas de clases del análisis y del diseño respectivamente así como el modelo entidad-relación de la base de datos que usará el sistema, exponiéndose además la explicación detallada de cada clase que interviene en su confección y en la base de datos.

Capítulo 4: Construcción de la solución propuesta

En el desarrollo de este capítulo se dará a conocer el flujo de trabajo de Implementación comenzando para ello a partir del resultado de la etapa de Diseño y describiendo cómo los elementos del modelo de diseño se implementan en términos de componentes y estos son organizados de acuerdo a los nodos específicos en el modelo de despliegue. Se muestran las dependencias existentes entre las partes del código y la estructura del sistema en ejecución mediante los diagramas de componentes y despliegue respectivamente, estos diagramas son artefactos generados durante el flujo de trabajo de Implementación y conforman lo que se conoce como modelo de implementación, este modelo describe los componentes a construir, su organización y dependencia entre los nodos físicos en los que funcionará la aplicación la que es implementada en términos de componentes. Además se realiza una breve descripción de algunas de las pruebas que se le realizan al sistema una vez concluida totalmente la etapa de desarrollo.

4.2 Diagrama de Componentes

El diagrama de componentes ilustra los componentes del software que serán usados para conformar el sistema. Un componente puede ser siempre considerado como una unidad autónoma dentro de un sistema o subsistema. UML define cinco estereotipos estándar que se aplican a los componentes: ejecutable, biblioteca, tabla, archivo y documento. Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación.

En la figura que se muestra a continuación se representa la distribución de los componentes de la aplicación Virtual_Know, para ello se han agrupado estos componentes en dos paquetes principales, el paquete FRM_Visual en el cual están contenidos todos los elementos del sistema los que representan las interfaces o

Capítulo 4: Construcción de la solución propuesta

ventanas de trabajo a través de las cuales el usuario interactúa con la aplicación, así como el paquete denominado virtual_know donde están agrupados los componentes que representan las clases del sistema o la parte lógica del mismo, ya que a través de los mecanismos de este paquete es donde se procesa el intercambio de información entre el usuario y la base de datos, se representan también las relaciones que existen entre los distintos componentes del paquete FRM_Visual y los de virtual_know así como la de este último con la base de datos con el mismo nombre.

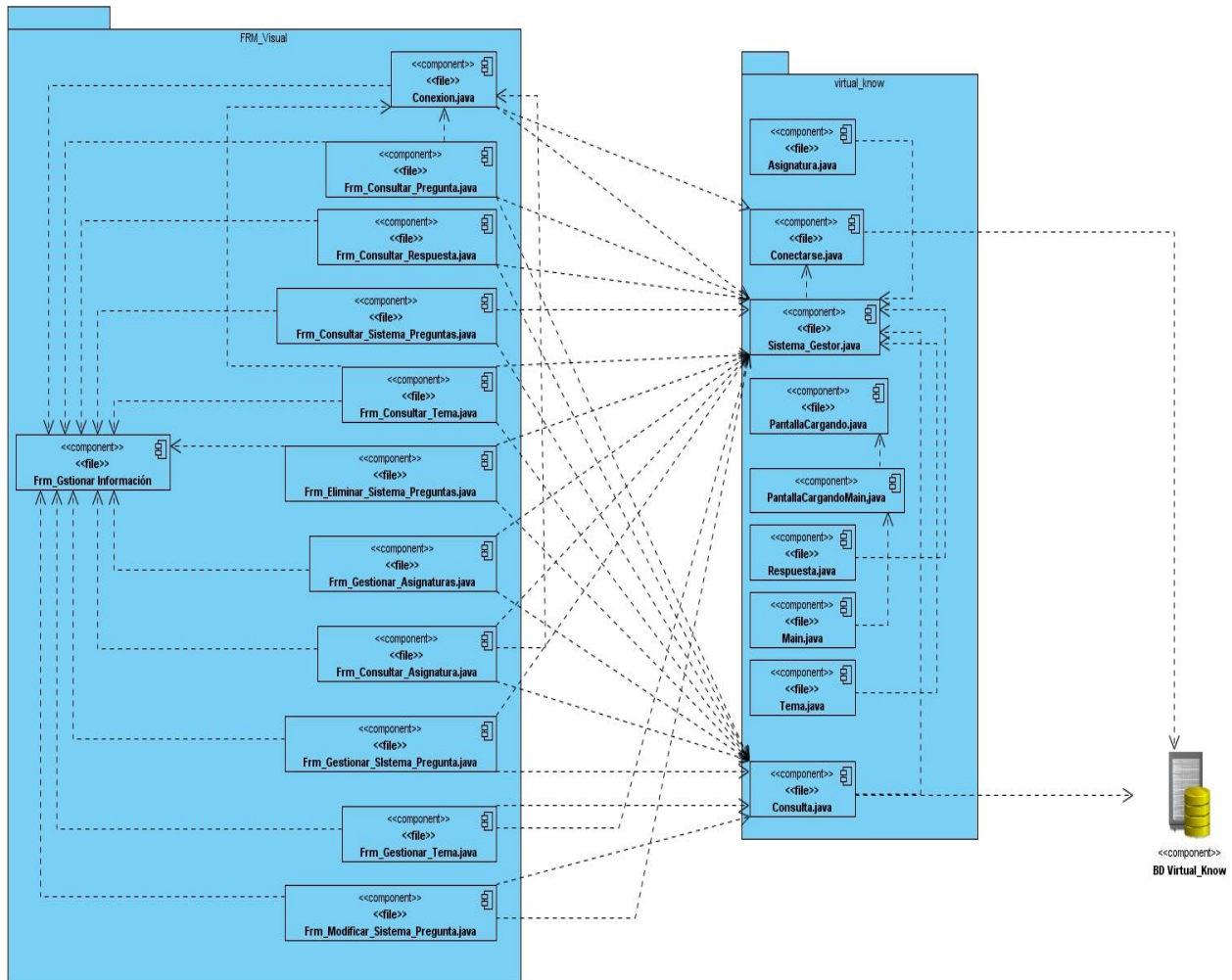


Figura No. 17: Diagrama de Componentes de la aplicación Virtual_Know

4.1 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Representa la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones antes planteadas y está compuesto por nodos, dispositivos y conectores donde los nodos son elementos de procesamiento con al menos un procesador, memoria, etc.- Ej: Servidor, PC cliente-; los dispositivos son nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela –Ej: Dispositivo USB- y los conectores expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo- Ej: USB, HTTP, HTTPS, TCP/IP-. Con el diagrama de despliegue se captura la configuración de los elementos de procesamiento y sus conexiones y se visualiza la distribución de los componentes de software en los nodos físicos.

En las figuras que se muestran a continuación se hace una representación del diagrama de despliegue del sistema, ilustrándose en ellas las relaciones software-hardware necesarias para el correcto funcionamiento de la aplicación en un ambiente de trabajo real. La primera imagen ilustra el despliegue óptimo o necesario para el uso de la aplicación, donde para ello es necesario tener los Servidores Web y Base de Datos separados de la PC Cliente o máquina donde se ejecute dicha aplicación, todo con el objetivo de evitar sobrecarga en los servidores, ya que una vez que cierta cantidad de usuarios interactúen con el sistema es necesario evitar las demoras en las transacciones que se realicen. La segunda imagen representa el despliegue actual o entorno real en el que se ejecuta la aplicación durante la etapa de desarrollo y pruebas.

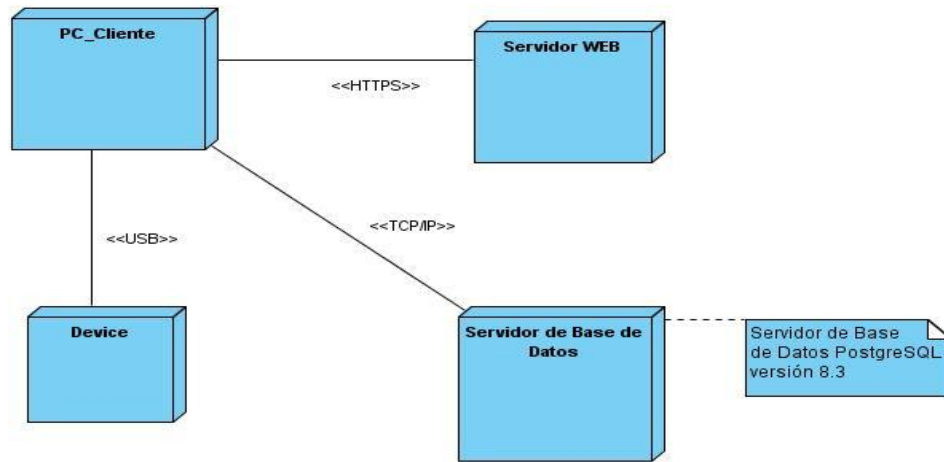


Figura No. 18: Diagrama de Despliegue óptimo.

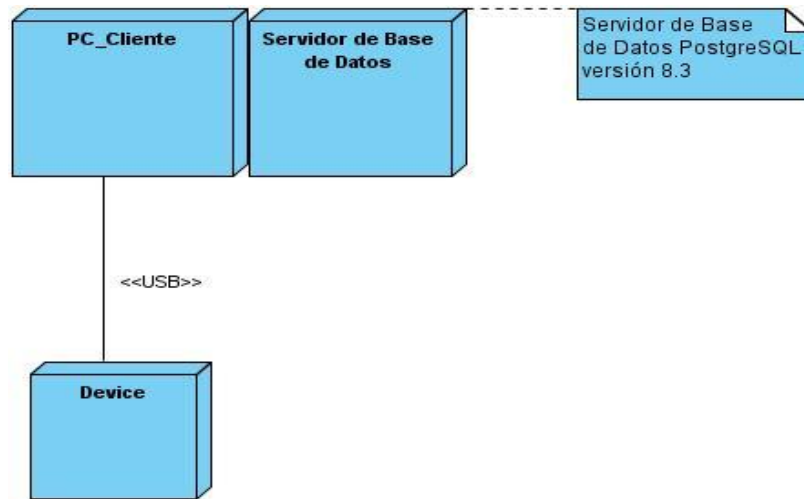


Figura No. 19: Diagrama de Despliegue actual.

4.3 Pruebas

Las pruebas son actividades a través de las cuales un sistema o componente es ejecutado bajo ciertas condiciones o requisitos los que permiten determinar de forma parcial o completa si el funcionamiento es el esperado o si existen errores que atenten

Capítulo 4: Construcción de la solución propuesta

contra el buen desempeño de la aplicación. Los resultados una vez obtenidos son observados y registrados para realizar una evaluación de la condición actual del sistema. Las pruebas constituyen un elemento crítico a la hora de hablar en términos de calidad y se clasifican en dos tipos principales, las de caja blanca (que son realizadas sobre el código de la aplicación) y las de caja negra (que son realizadas sobre la interfaz del software con el objetivo fundamental de descubrir si la entrada de datos es validada así como si los resultados obtenidos son los esperados). A continuación se representan algunas de las pruebas de caja negra realizadas a las principales funcionalidades de la herramienta Virtual_Know:

Tabla No. 30: Pruebas de caja negra del caso de uso "Autenticar Usuario"

Condición de Entrada	Casos válidos	Casos no válidos
Dirección IP servidor	Cadena de caracteres de la forma 192.168.x.x	Campo vacío o cadena de otra forma que de la especificada
Puerto de Conexión BD	Numero entero	Campo vacío o cadena de caracteres
Nombre BD	Nombre de la Base de datos	Campo vacío
Usuario	Cadena de caracteres	Campo vacío
Contraseña	Cadena de caracteres	Campo vacío

Caso de Uso	Autenticar Usuario
Caso de Prueba	Establecer la conexión con el usuario que introduzca correctamente los datos de autenticación.
Entrada	EL usuario introduce los datos de conexión correctamente de la forma: Host: 192.168.23.260 Puerto: 5432 Nombre de la BD: sistemapregunta Usuario: profesor Pass: profesor
Resultado	El sistema establece la conexión con la base de datos y proporciona el

Capítulo 4: Construcción de la solución propuesta

	acceso al usuario autenticado.
Condiciones	No debe existir ningún campo vacío y los datos deben ser del tipo especificado.

Caso de Uso	Autenticar Usuario
Caso de Prueba	Establecer la conexión con el usuario que introduzca incorrectamente los datos de autenticación.
Entrada	Cualquiera de los campos está vacío o simplemente no son los correctos
Resultado	El sistema muestra un mensaje de error especificando que los datos introducidos son incorrectos.
Condiciones	Deben existir campos vacíos o con datos incorrectos

De acuerdo a los resultados obtenidos producto de haber realizado las pruebas a la funcionalidad Autenticar Usuario se pudo comprobar que los mismos fueron satisfactorios al no presentar dicha funcionalidad ninguna inconsistencia en cuanto a la entrada y manejo de datos.

Tabla No. 31: Pruebas de caja negra del caso de uso "Gestionar Asignaturas" escenario "Insertar Asignatura"

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Cadena de caracteres	Dejar vacío el campo nombre.

Caso de Uso	Gestionar Asignaturas escenario "Insertar Asignatura"
Caso de Prueba	Permitir insertar una nueva asignatura en la base de datos introduciendo correctamente los datos.
Entrada	El usuario introduce un nombre para la nueva asignatura
Resultado	El sistema inserta una nueva asignatura en la base de datos.
Condiciones	Los datos introducidos por el usuario son correctos y no está vacío el

Capítulo 4: Construcción de la solución propuesta

	campo Nombre.
--	---------------

Caso de Uso	Gestionar Asignaturas escenario “Insertar Asignatura”
Caso de Prueba	Insertar una nueva asignatura dejando vacío el campo Nombre
Entrada	El usuario deja el campo Nombre vacío.
Resultado	El sistema muestra un mensaje especificando que no se admiten campos en blanco.
Condiciones	Algún campo vacío.

Tabla No. 32: Pruebas de caja negra del caso de uso “Gestionar Asignaturas” escenario “Modificar Asignatura”.

Condición de Entrada	Casos válidos	Casos no válidos
Identificador	Número entero	Dejar vacío el campo identificador
Nombre	Cadena de caracteres	Dejar vacío el campo Nombre

Caso de Uso	Gestionar Asignaturas escenario “Modificar Asignatura”
Caso de Prueba	Modificar una asignatura existente introduciendo correctamente sus datos
Entrada	El usuario selecciona el identificador de la pregunta haciendo clic en la tabla que se muestra en la interfaz e inserta un nuevo nombre para la asignatura seleccionada.
Resultado	El sistema modifica el nombre de la asignatura en la base de datos.
Condiciones	Los datos introducidos deben estar correctos y no deben existir campos en blanco.
Caso de Uso	Gestionar Asignaturas escenario “Modificar Asignatura”
Caso de Prueba	Modificar una asignatura introduciendo incorrectamente los datos o dejando campos vacíos.
Entrada	El usuario deja alguno de los campos requeridos vacíos.
Resultado	El sistema muestra un mensaje especificando que no se permiten campos en blanco.

Capítulo 4: Construcción de la solución propuesta

Condiciones	Algún campo vacío.
-------------	--------------------

Tabla No. 33: Pruebas de caja negra del caso de uso “Gestionar Asignaturas” escenario “Eliminar Asignatura”.

Condición de Entrada	Casos válidos	Casos no válidos
Identificador	Número entero	Dejar vacío el campo identificador
Caso de Uso	Gestionar Asignaturas escenario “Eliminar Asignatura”	
Caso de Prueba	Eliminar una asignatura existente introduciendo correctamente sus datos	
Entrada	El usuario selecciona el identificador de la pregunta que desea eliminar haciendo clic en la tabla que se muestra en la interfaz.	
Resultado	El sistema elimina la asignatura de la base de datos.	
Condiciones	Los datos introducidos deben estar correctos y no deben existir campos en blanco.	

Caso de Uso	Gestionar Asignaturas escenario “Modificar Asignatura”
Caso de Prueba	Eliminar una asignatura existente introduciendo incorrectamente sus datos.
Entrada	El usuario no selecciona el identificador de la pregunta de la tabla que se muestra en la interfaz.
Resultado	El sistema muestra un mensaje especificando que no se admiten campos en blanco.
Condiciones	Existen campos vacíos.

Después de haberle realizado las pruebas correspondientes a la funcionalidad Gestionar Asignaturas se arribó a la conclusión de que los resultados obtenidos se corresponden con lo esperado de acuerdo al funcionamiento correcto de esta funcionalidad.

Capítulo 4: Construcción de la solución propuesta

Tabla No. 34: Pruebas de caja negra del caso de uso “Gestionar Temas” escenario “Insertar Tema”.

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Cadena de caracteres	Dejar vacío el campo Nombre
Asignatura	Cadena de caracteres	Dejar vacío el campo Asignatura
Id_asignatura	Numero entero	Dejar vacío el campo Id_Asignatura

Caso de Uso	Gestionar Temas escenario “Insertar Tema”
Caso de Prueba	Permitir insertar un nuevo tema a la base de datos una vez que se hayan introducido los datos correctamente,
Entrada	El usuario inserta el nombre del nuevo tema, selecciona el nombre de la asignatura a la cual pertenece y se carga automáticamente el identificador de esta en el campo correspondiente.
Resultado	Se inserta un nuevo tema en la base de datos.
Condiciones	Los datos introducidos deben estar correctos y no deben existir campos en blanco.

Caso de Uso	Gestionar Temas escenario “Insertar Tema”
Caso de Prueba	Insertar un nuevo tema en la base de datos introduciendo datos incorrectos.
Entrada	El usuario deja campos en blanco o no realiza la selección de la asignatura.
Resultado	El sistema muestra un mensaje especificando que no se admiten campos en blanco.
Condiciones	Existen campos vacíos.

Tabla No. 35: Pruebas de caja negra del caso de uso “Gestionar Temas” escenario “Modificar Tema”.

Capítulo 4: Construcción de la solución propuesta

Condición de Entrada	Casos válidos	Casos no válidos
Nombre	Cadena de caracteres	Dejar vacío el campo Nombre
Identificador	Número entero	Dejar vacío el campo Identificador
Asignatura	Cadena de caracteres	Dejar vacío el campo Asignatura
Id_asignatura	Número entero	Dejar vacío el campo Id_Asignatura

Caso de Uso	Gestionar Temas escenario “Modificar Tema”
Caso de Prueba	Permitir modificar un tema existente en la base de datos una vez que se hayan introducido los datos correctamente,
Entrada	El usuario selecciona de la tabla el tema que desea modificar, inserta el nombre del tema, selecciona el nombre de la asignatura a la cual pertenece y se carga automáticamente el identificador de esta así como el identificador del tema en los campos correspondientes.
Resultado	Se modifica el tema en la base de datos.
Condiciones	Los datos introducidos deben estar correctos y no deben existir campos en blanco.

Caso de Uso	Gestionar Temas escenario “Modificar Tema”
Caso de Prueba	Modificar un tema existente en la base de datos introduciendo datos incorrectos.
Entrada	El usuario deja campos en blanco o no selecciona los correspondientes.
Resultado	El sistema muestra un mensaje especificando que no se admiten campos en blanco.
Condiciones	Existen campos vacíos o no se realizó ninguna selección.

Capítulo 4: Construcción de la solución propuesta

Tabla No. 36: Pruebas de caja negra del caso de uso "Gestionar Temas" escenario "Eliminar Tema".

Condición de Entrada	Casos válidos	Casos no válidos
Identificador	Número entero	Dejar vacío el campo Identificador

Caso de Uso	Gestionar Temas escenario "Eliminar Tema"
Caso de Prueba	Modificar un tema existente en la base de datos introduciendo datos correctos.
Entrada	El usuario selecciona de la tabla existente en la interfaz el tema que desea eliminar y se carga el identificador en el campo correspondiente.
Resultado	Se elimina el tema de la base de datos.
Condiciones	Debe haberse seleccionado previamente el tema a eliminar.

Caso de Uso	Gestionar Temas escenario "Eliminar Tema"
Caso de Prueba	Modificar un tema existente en la base de datos introduciendo datos incorrectos.
Entrada	El usuario no selecciona de la tabla existente en la interfaz el tema que desea eliminar.
Resultado	El sistema muestra un mensaje especificando que no se admiten campos en blanco.
Condiciones	No se realizó la selección del tema.

Después de obtenidos los resultados de las pruebas aplicadas a la funcionalidad Gestionar Tema se concluye diciendo que dichos resultados arrojan que la misma presenta un buen funcionamiento así como un correcto manejo de posibles errores.

Capítulo 4: Construcción de la solución propuesta

En el presente capítulo se puede apreciar el diagrama de despliegue óptimo de la aplicación Virtual_Know así como el actual, se incluyó también el diagrama de componentes del sistema y el resultado de algunas de las pruebas de caja negra realizada a las principales funcionalidades de la aplicación, arrojando que dichas funcionalidades no presentan anomalías en su funcionamiento y que las mismas realizan un correcto tratamiento de los datos introducidos por el usuario.

Conclusiones

Tomando como referencia los objetivos planteados en la investigación así como el trabajo realizado para darle cumplimiento por parte del equipo de desarrollo, se arribó a las siguientes conclusiones.

- Se realizó un detallado estudio de las herramientas que serían usadas en la implementación de la herramienta Virtual_Know así como de los sistemas similares a esta existentes en el mundo, todo ello en busca de soluciones factibles y características funcionales similares a tener en cuenta para el desarrollo de dicha herramienta.
- Se efectuó el diseño de la herramienta Virtual_Know de acuerdo a lo especificado en el levantamiento de requisitos, lo que brindó al equipo una visión para su posterior implementación.
- Se llevó a cabo la implementación del módulo para la gestión de asignaturas de la herramienta educativa SMProg, el cual permite manejar el contenido que será evaluado en los juegos didácticos.
- Se puso a disposición de los profesores el módulo de gestión de asignaturas (Virtual_Know), el cual facilita la gestión de asignaturas, temas y sistemas de preguntas para sus asignaturas, así como el adecuado funcionamiento del módulo SMProg para el EVA - UCI.

Por todo lo antes planteado es posible concluir que los objetivos trazados en el presente trabajo de diploma fueron cumplidos de forma satisfactoria y los resultados alcanzados fueron los esperados.

Recomendaciones

Una vez concluido el desarrollo del módulo de gestión de asignaturas, cumplidos los objetivos trazados y teniendo en cuenta las experiencias y conocimientos adquiridos a todo lo largo del desarrollo del mismo se recomienda:

- Realizar la confección del Manual de Usuario el cual sirva de guía y de apoyo a los profesores que interactuarán con el sistema.
- Realizar una investigación así como un análisis para determinar nuevas funcionalidades que se puedan agregar a la aplicación.
- Realizar la integración con la herramienta educativa SMProg.
- Implementar la versión web de la herramienta Virtual_Know.
- Refinar aún más las funcionalidades ya implementadas.
- Realizar mejoras a la apariencia visual del sistema.
- Extender el uso de la aplicación a profesores de otra facultad siempre con el objetivo de mejorar el proceso enseñanza-aprendizaje.

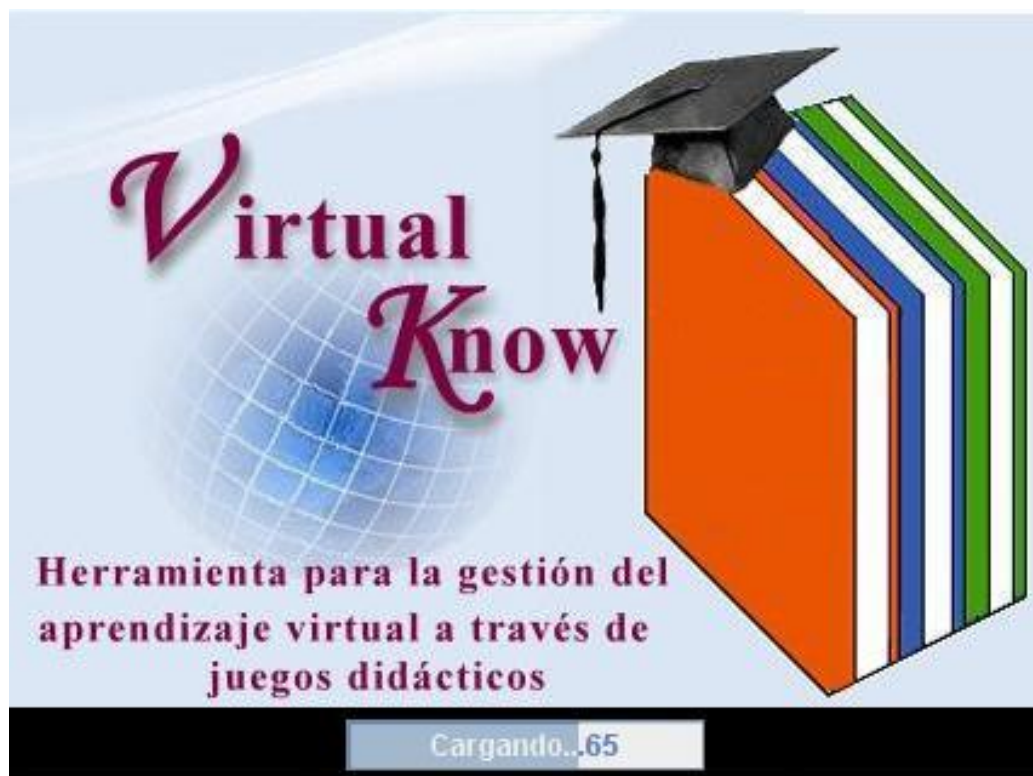
Bibliografía

1. **C. Batini, S. C.** *Diseño Conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones.* 1994. Addison-Wesley / Díaz de Santos.
2. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso Unificado de Desarrollo de Software.* La Habana: Félix Varela, 2004. Vol.I.
3. **Jacobson, I.; Booch, G. y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de software. 2000.* Addison-Wesley.
4. **J. Gosling.; B. Joy.; G. Steele; G. Bracha.** *The authoritative reference on the Java programming language, The Java Language Specification.* 2nd ed. Boston: Addison Wesley, 2000. [En línea] [citado el 23 de 11 de 2009]. Disponible en <http://java.sun.com/docs/books/jls/>.
5. **Larman, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ciudad de la Habana: s.n., 2004. Vol.I.*
6. **Lenguajes de Programación.** [En línea] [citado el: 16 de 10 de 2009]. Disponible en: <http://es.kioskea.net/contents/langages/langages.php3>.
7. **Modelo Entidad-Relación** [En línea] [Citado el: 29 de 1 de 2010]. Disponible en <http://www.calasanz-pereira.edu.co/index.php/Informatica/entidadrelacion.html>.
8. **Paradigm, Visual.** *Increase productivity and enhance communication and collaboration efficiency by using UML.* [En línea] [Citado el: 20 de 11 de 2009]. Disponible en <http://www.visual-paradigm.com>.
9. **PostgreSQL.** *Guía Ubuntu.* [En línea] [citado el 3 de 2 de 2010]. Disponible en <http://www.guia-ubuntu.org/index.php?title=PostgreSQL>.
10. **PostgreSQL, E. D. D.D.** *Manual de usuario de PostgreSQL.* [En línea] [Citado el 15 de 1 de 2010]. Disponible en <http://www.librosdeluz.net/tag/Libros-de-Bases-de-Datos>.
11. **PostgreSQL.org.** *PostgreSQL.* [En línea] [Citado el 7 de 12 de 2009]. Disponible en http://postgresql.org.pe/articles/pgsql_alta_disponibilidad.pdf.
12. **S. Pressman, R.** *Ingeniería del Software: Un Enfoque Práctico.* 5ta Edición. 2001. Madrid Carchelejo.

Referencias Bibliográficas

1. **Types of Education.** El surgimiento de la gestión del conocimiento [En línea] [citado el: 13 de 10 de 2009]. Disponible en: <http://education.nireblog.com/post/2007/07/11/el-surgimiento-de-la-gestion-del-conocimiento>.
2. **Olivares Abel, López William.** Sistema Informático para la Gestión de la Información de la Unión de Jóvenes Comunistas de la Facultad 2. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, 2009.
3. **Definición.** *Que significa asignaturas* [En línea] [citado el: 14 de 10 de 2009]. Disponible en: <http://definicion.de/asignaturas/>.
4. **Herramientas de Gestión del Conocimiento.** [En línea] [citado el: 14 de 10 de 2009]. Disponible en: <http://www.gestiondelconocimiento.com/documentos2/america/herramientas.pdf>
5. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso Unificado de Desarrollo de Software.* La Habana: Félix Varela, 2004. Vol.I.
6. **Jacobson, I.; Booch, G. y Rumbaugh, J.;** *El Proceso Unificado de Desarrollo de software. 2000.* Addison-Wesley.
7. **Larman, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ciudad de la Habana: s.n., 2004. Vol.I.*
8. **Lenguajes de Programación.** [En línea] [citado el: 16 de 10 de 2009]. Disponible en: <http://es.kioskea.net/contents/langages/langages.php3>.
9. **Larman, C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Ciudad de la Habana: s.n., 2004. Vol. II.*
10. **Visconti, Marcello.** *Fundamentos de Ingeniería de Software* [En línea] [citado el 24 de enero de 2010]. Disponible en: <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/03-AnDisOO.pdf>
11. **Modelo Entidad-Relación** [En línea] [citado el: 29 de enero de 2010]. Disponible en <http://www.calasanz-pereira.edu.co/index.php/Informatica/entidadrelacion.html>.

Anexos



Anexo No 1: Pantalla de bienvenida de la herramienta Virtual_Know.



Anexo No 2: Pantalla principal de la herramienta Virtual_Know.

The image shows a screenshot of a software dialog box. The title bar at the top reads "Datos de la conexión" and includes a close button (X). The main title of the dialog is "Autenticar Usuario". The dialog contains several input fields: "Dirección IP del servidor" and "Puerto" are on the top line; "Nombre de la base de datos" is on the second line; "Usuario" and "Contraseña" are on the third and fourth lines respectively. At the bottom right, there is a button with a right-pointing arrow and the text "Conectar".

Anexo No 3: Interfaz de conexión con la base de datos.

The image shows a software window titled "Gestionar Información" with a close button in the top right corner. The window is divided into three distinct sections, each with a title and a set of controls.

- Operaciones en Asignaturas y Temas:** This section contains the text "Seleccione la operación que desea realizar" followed by a dropdown menu currently showing "Gestionar Asignaturas" and a "Siguiente" button with a green arrow icon.
- Operaciones en Sistema de Preguntas:** This section contains the text "Seleccione la operación que desea realizar" followed by a dropdown menu currently showing "Insertar" and a "Siguiente" button with a green arrow icon.
- Consultar Información:** This section contains the text "Mostrar datos acerca de" followed by a dropdown menu currently showing "Asignatura" and a "Mostrar" button with a green arrow icon.

Anexo No 4: Interfaz de Gestión de Información.

Gestionar Asignaturas

Insertar Asignaturas Modificar Asignaturas Eliminar Asignaturas

Inserte el nombre de la nueva asignatura

Nombre

Insertar Guardar Información

Inserte el nombre de la asignatura que desea buscar

Nombre

Buscar

Asignaturas

Nombre de la Asignatura
Asignatura de Prueba

Anexo No 5: Interfaz de Gestión de Asignaturas.

The screenshot shows a window titled "Gestionar Temas" with three tabs: "Insertar Temas", "Modificar Temas", and "Eliminar Temas". The "Insertar Temas" tab is active. It contains two main sections. The first section, titled "Inserte el nombre del nuevo tema", has a text input field labeled "Nombre", a dropdown menu labeled "Seleccione la Asignatura a la que pertenece", and two buttons: "Insertar" (with a green checkmark icon) and "Guardar Información" (with a floppy disk icon). The second section, titled "Inserte el nombre del tema que desea buscar", has a text input field labeled "Nombre" and a "Buscar" button (with a magnifying glass icon). To the right of these sections is a table titled "Temas" with a header "Nombre del Tema" and one row containing the text "Tema de prueba".

Anexo No 6: Interfaz de Gestión de Temas.

The screenshot shows a software window titled "Insertar Información en el Sistema de Preguntas". The window has a tab labeled "Insertar Sistema de Preguntas".

Insertar Pregunta

Contenido de la Pregunta: [Empty text area]

Complejidad: [Bajo ▼]

Tema: [Dropdown menu] [Empty text field]

[Insertar]

Insertar Respuesta

Contenido de la respuesta: [Empty text field]

Tipo de Respuesta: [Correcta ▼]

[Insertar]

[Guardar Información]

Insertar Sistema de Preguntas

Contenido de la Pregunta: [Empty text field]

Tema: [Dropdown menu] [Empty text field]

[Insertar]

[Insertar Nueva Pregunta]

[Cerrar]

Anexo No 7: Interfaz de Gestión de Sistema de Preguntas.

Consultar Asignaturas

Identificador

Nombre

Consultar

Asignaturas

ID	Nombre
23	Introduccion a la Programacion
24	Programacion I
25	Programacion II
26	Programacion III
27	Programacion IV
28	Inteligencia Artificial
29	Matematica I
30	Matematica II
31	Matematica III
32	Matematica IV
33	Fisica I
34	Fisica II

Cerrar

Anexo No. 8: Interfaz de Consultar Asignaturas.

The screenshot shows a window titled 'Consultar Tema' with a search interface. It includes three input fields: 'ID del Tema', 'Nombre del Tema' (with a dropdown arrow), and 'ID de la Asignatura'. A 'Buscar' button with a magnifying glass icon is located to the right of these fields. Below the search area is a table with the title 'Temas'. The table has four columns: 'ID_Tema', 'Tema', 'ID_Asignatura', and 'Asignatura'. The table is currently empty.

ID_Tema	Tema	ID_Asignatura	Asignatura

Anexo No. 9: Interfaz de Consultar Temas.


Design Preview [Frm_Consultar_SistemaPregunta]

Consultar Sistema de Preguntas

Identificador de la pregunta Tema

Pregunta Complejidad

Identificador del Tema




Sistema de Preguntas

Identificador	Pregunta	ID_Tema	Tema	ID_Complejidad	Complejidad

Seleccione una pregunta para ver su contenido

Respuestas asociadas a la pregunta

Identificador	Contenido	Corrección	ID_Pregunta




Anexo No. 10: Interfaz de Consultar Sistema de Preguntas.

Consultar Pregunta

ID_Pregunta ID_Complejidad


Pregunta Tema

ID_Tema



Preguntas

ID_Pregunta	Pregunta	ID_Tema	Tema	ID_Complejidad	Complejidad



Anexo No. 11: Interfaz de Consultar Preguntas.

Consultar Respuesta

Consultar Respuestas

Seleccione la pregunta de la cuál desea consultar sus respuestas

Identificador de la pregunta

Buscar

Respuestas

ID_Respuesta	Contenido	Correcta

Cerrar

Anexo No. 12: Interfaz de Consultar Respuestas.

The screenshot shows a software window titled "Eliminar Sistema de Preguntas" with a close button in the top right corner. The window contains a tabbed interface with two tabs: "Tablas de Datos" and "Eliminar Sistema de Preguntas". The "Eliminar Sistema de Preguntas" tab is active and contains three distinct sections for deleting data:

- Eliminar las Respuestas asociadas a una Pregunta:** This section includes a label "Identificador de Pregunta" with a subtext "Seleccione de la tabla Pregunta" and an empty text input field. To the right of the input field is a blue button with a red 'X' icon and the text "Eliminar".
- Eliminar Respuesta:** This section includes a label "Identificador de la Respuesta" with a subtext "Seleccione de la tabla Respuestas" and an empty text input field. To the right of the input field is a blue button with a red 'X' icon and the text "Eliminar".
- Eliminar Pregunta:** This section includes a label "Identificador de la Pregunta" with a subtext "Seleccione de la tabla Pregunta" and an empty text input field. To the right of the input field is a blue button with a red 'X' icon and the text "Eliminar".

Anexo No. 13: Interfaz de Eliminar Sistema de Preguntas.

Modificar Sistema de Preguntas

Tablas de Datos Modificar Sistema de Pregunta

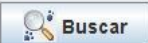
Pregunta

ID_Pregunta	Pregunta	ID_Tema	Tema	ID_Complejidad	Complejidad
28	Si la lista L inicialmente ti...	9	Estructuras de Datos Line...	2	Medio
29	Si la cola C inicialmente ti...	9	Estructuras de Datos Line...	1	Bajo
31	Dadas las siguientes afir...	10	Estructuras de Datos No ...	2	Medio

Respuestas

ID_Respuesta	Contenido	Correcta

Inserte el identificador de la pregunta



Anexo No. 14: Interfaz de Modificar Sistema de Preguntas.

Glosario de Términos

1. **UCI** (Universidad de las Ciencias Informáticas). Proyecto surgido como idea del Comandante en Jefe Fidel Castro Ruz al calor de la batalla de ideas con el objetivo de formar profesionales altamente calificados en la rama de la informática, este ha sido un sueño que se realiza día a día en el empeño de convertir la universidad en un centro de excelencia. Desde los primeros pasos el líder de la Revolución ha guiado con su experiencia y visión de futuro la edificación de esta institución.
2. **EVA** (Entorno Virtual de Aprendizaje): Un entorno virtual de aprendizaje es un espacio con accesos restringidos, concebido y diseñado para que las personas que acceden a él desarrollen procesos de incorporación de habilidades y saberes, mediante sistemas telemáticos.
3. **SMProg**: Software de Motivación a la Programación. Proyecto creado en la Facultad No 10 a raíz de un proyecto de innovación pedagógica con el objetivo de motivar el estudio de las asignaturas de Introducción a la Programación, Programación I, Programación II, Programación III, Programación IV por parte de los estudiantes.
4. **Moodle** (Modular Object-Oriented Dynamic Learning Environment). Moodle es un CMS o paquete de software para la creación de cursos y sitios web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista.
5. **CMS** (Content Management System o Sistema de Gestión de Contenido). Es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web.
6. **PostgreSQL**. Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Además constituye el exponente más avanzado de Bases de Datos OpenSource que existe en la actualidad.

7. **IDE** (Integrated Development Environment). Un IDE es un entorno de desarrollo integrado o programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.
8. **POO**. (Programación Orientada a Objetos). Es un paradigma de programación que usa objetos y sus interacciones para desarrollar aplicaciones de computadoras además de ser una forma especial de programar más cercana a como se expresarían las cosas en la vida real.