

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 2

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

*Herramienta web de interacción con  
PostgreSQL de apoyo a las asignaturas  
Sistemas de Bases de Datos*

**Autores:** Osmany Machin Rodríguez

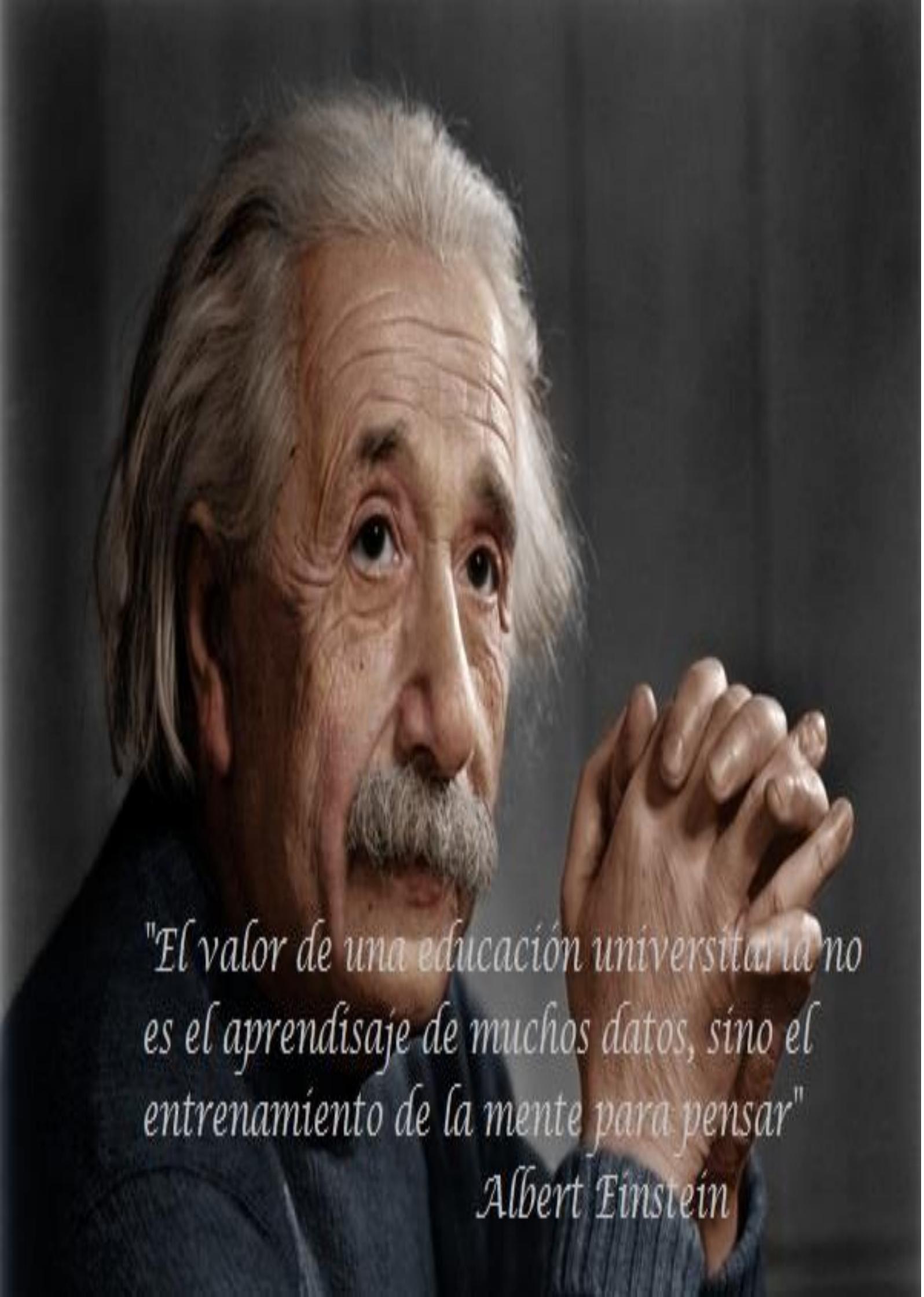
Marlon Oscar Cárdenas Carreras

**Tutor:** Lic. Lianne Guillén Pérez

**Co-Tutor:** Ing. Yubismel Perdomo Velázquez

**Consultante:** Ing. Raymon Weeden Gamboa

La Habana, Cuba  
junio, 2017  
“Año 59 de la Revolución”

A close-up portrait of Albert Einstein, showing his characteristic wild white hair and mustache. He is looking slightly to the right with a thoughtful expression. His hands are clasped together in front of him. The background is a dark, neutral color.

*"El valor de una educación universitaria no es el aprendizaje de muchos datos, sino el entrenamiento de la mente para pensar"*

*Albert Einstein*

## **Agradecimientos**

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a mi madre, mi padre y mi hermana, por la orientación, el seguimiento y la supervisión continúa, pero sobre todo por la motivación y el apoyo recibido a lo largo de estos años. Especial reconocimiento merece el interés mostrado por mi trabajo y las sugerencias recibidas de los profesores Lianne y Yubismel, con los que me encuentro en deuda por el ánimo infundido y la confianza en mí depositada. También me gustaría agradecer la ayuda recibida del profesor Raymond Weeden que fue como un tutor más. Quisiera hacer extensiva mi gratitud a mis compañeros del apartamento, de aula, profesores y amigos.

## **Declaración de Autoría**

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales, con carácter exclusivo.

Para que así conste, firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2017.

\_\_\_\_\_  
Osmany Machin Rodríguez

Firma del Autor

\_\_\_\_\_  
Marlon Oscar Cárdenas Carreras

Firma del Autor

\_\_\_\_\_  
Ing. Yubismel Perdomo Velázquez

Firma del Tutor

\_\_\_\_\_  
Lic. Lianne Guillen Pérez

Firma del Tutor

## **Resumen:**

En el camino hacia una sociedad de la información y del conocimiento se reconoce ampliamente la contribución fundamental de las tecnologías de información y comunicación (TIC). Las mismas presentan grandes capacidades para transformar la enseñanza y mejorar el aprendizaje. Para ello, las bases de datos poseen gran documentación y variedad de herramientas, pero las mismas carecen de funcionalidades integradas de gran importancia para la docencia. Es por ello que el siguiente trabajo tiene como objetivo elaborar una herramienta web que permita interactuar con el Sistema Gestor de Bases de Datos PostgreSQL, para apoyar el desarrollo de las clases prácticas de laboratorio en las asignaturas de Sistemas de Bases de Datos I y II. Partiendo de un análisis sobre los sistemas gestores de bases de datos, utilizando como lenguaje de programación del lado del servidor PHP y como framework de desarrollo Symfony en su versión 2.8. Además, se hizo uso de la metodología de desarrollo de software Agile Unified Process (AUP). La herramienta obtenida contiene las siguientes funcionalidades: Asignación de tareas a estudiantes, creación y eliminación de subgrupos de trabajo, asignación de evaluaciones y como funcionalidad principal un intérprete de consultas SQL, dando cumplimiento al objetivo y las tareas planteadas.

## Índice:

Introducción.....	9
Capítulo 1: Fundamentación teórica.....	15
1.1. Concepto general de Bases de Datos.....	15
1.2. Estudio del estado del arte.....	16
1.2.1. Estudio de herramientas que interactúan con un Sistema Gestor de Bases de Datos en el mundo .....	16
1.2.2. Estudio de herramientas que interactúan con un Sistema Gestor de Bases en Cuba.....	19
Valoración del estudio del estado del arte.....	19
1.3. Herramientas, tecnologías y las metodologías de desarrollo de software a utilizar .....	20
1.3.1. Herramientas y lenguaje de modelado .....	20
1.3.2. Paradigma de programación .....	21
Programación orientada a objetos (POO) .....	22
1.3.3. Tecnologías y lenguajes de programación del lado del servidor ..	23
1.3.4. Tecnologías y lenguajes de programación del lado del cliente.....	26
1.3.5. Servidor web.....	27
1.3.6. Entorno de Desarrollo Integrado (IDE) .....	28
1.3.7. Metodología de desarrollo .....	28
Otras metodologías ágiles.....	29
Extreme Programming.....	29
SCRUM .....	29
RUP.....	30

Valoración de las metodologías .....	30
Conclusiones del capítulo .....	30
Capítulo 2: Características del sistema .....	31
2.1.1. Información que se maneja.....	31
2.2. Modelo de dominio.....	31
Diagrama de Clases del Modelo del Dominio .....	33
2.3. Reglas del negocio.....	33
2.3.1. Trabajadores del negocio.....	34
2.3.2. Entidad del negocio .....	34
2.4. Especificación de los requisitos de software .....	34
2.4.1. Requerimientos Funcionales:.....	34
2.4.2. Requisitos no Funcionales .....	35
2.5. Definición de casos de uso y actores del sistema .....	39
2.5.1. Casos de Uso del Sistema.....	39
2.5.2. Diagrama de Casos de Uso .....	42
2.5.3. Casos de uso expandidos.....	42
2.6. Patrón de arquitectura.....	49
Modelo Vista Controlador .....	49
Descripción de capas .....	49
2.7. Patrones de diseño .....	50
2.7.1. Patrones GRASP .....	50
Conclusiones parciales .....	53
Capítulo 3: Análisis y diseño del sistema .....	54

3.1. Prototipos no funcionales .....	54
3.2. Estándar de codificación .....	56
3.3. Diagrama de clases de diseño .....	57
3.4. Diseño de la BD .....	58
Modelo de datos.....	58
3.5. Modelo de implementación .....	59
3.5.1. Modelo de Despliegue .....	60
3.5.2. Descripción de los nodos .....	61
3.6. Pruebas al sistema.....	61
Pruebas de caja negra .....	62
¿Qué es una prueba de carga?.....	63
¿Cuál es la diferencia con una prueba de estrés? .....	64
Conclusiones parciales .....	65
Conclusiones.....	66
Recomendaciones.....	67
Referencias bibliográficas .....	68
Bibliografía Consultada .....	71
Anexos .....	74

## Introducción

El mundo contemporáneo se caracteriza por el desarrollo de las ciencias y en particular de las Tecnologías de Información y Comunicación (TIC), razón por la que se alude a la existencia de la "Sociedad del Conocimiento", "Sociedad de la Información" o la "Era de la Información".

La Conferencia Mundial sobre Educación Superior, celebrada en París en 1998 bajo los auspicios de la UNESCO, reconoció ampliamente la contribución fundamental de las TIC a la calidad y pertinencia de la gestión de información y conocimientos que se realiza en el ámbito de la educación superior, en el camino hacia una sociedad de la información y del conocimiento.

La cantidad de información que debe manejar una entidad para sobrevivir es cada vez mayor. Para ello, deben existir métodos eficientes tanto para el almacenamiento rápido como para la consulta ágil. La tecnología que actualmente es utilizada para manejar grandes volúmenes de datos son las Bases de Datos.

En los inicios de la computación se elaboraban programas de computador a los cuales, siempre que se ejecutaban, se les proporcionaban los datos de entrada y no se veía la necesidad de guardar la información en memoria secundaria, tanto la resultante como la de entrada, para su uso posterior.

Con el tiempo, el computador adquiere un uso más comercial en las empresas para llevar la contabilidad, la nómina y otras actividades. Estas tareas, por lo general, necesitaban una serie de datos iguales para usarlos en las diferentes corridas de los programas e implicaban un gran esfuerzo porque había que entrarlos nuevamente, cada vez.

Ante este problema, aparecen los Sistemas de Archivos, donde los datos se almacenan de manera permanente para sobrevivir a los programas que los usan; característica conocida como la persistencia.

Aunque el ambiente de archivos representó un avance en su momento, posteriormente se enfrentaron con tres problemas básicos.

El primer problema, consiste en la alta redundancia de datos. El mismo dato aparece repetido en varios archivos. Las diferentes versiones de un mismo dato pueden estar con un grado de actualización distinto en cada lugar.

El segundo problema es la inflexibilidad porque cuando se quiere agrupar los datos de cierta manera no se puede hacer, debido a la organización dada en los archivos que no tienen ninguna clase de vínculos o presentan formatos diferentes. Esta inflexibilidad impide resolver rápidamente consultas espontáneas y aunque los datos existan, la información no puede proporcionarse relacionando datos.

El tercer problema que se presenta con el sistema de archivos es el costo de efectuar cambios en las estructuras de los datos porque al cambiar la representación de un dato, se necesita cambiar el programa para que lo reciba de la nueva manera. Además, es altamente probable que los mismos datos se encuentren en otros archivos; entonces los cambios se propagarán de una manera incontrolable. Esto aumenta el tiempo que el personal especializado debe invertir en el mantenimiento de los programas y, por ende, reduciendo el tiempo que le pudieran dedicar al desarrollo de nuevas aplicaciones. (1)

Las tecnologías referentes a las Bases de Datos proporcionan los medios a las entidades para que cumplan con su objetivo. Una de ellas es la Universidad de Ciencias Informáticas (UCI), dicha universidad cuenta con un vasto programa de estudio y la puesta en práctica de una amplia producción industrial de software de salud y educación. En el desarrollo de estos últimos se tienen buenos precedentes con plataformas que ayudan a la docencia, aunque siguen haciendo un poco engorroso el trabajo en algunas actividades.

Los estudiantes en la UCI en su formación como ingenieros reciben un gran número de asignaturas de vital importancia para su labor como futuros profesionales. De gran valor son las asignaturas de Sistemas de Bases de Datos I y Sistemas de Bases de Datos II. Estas asignaturas tienen un caudal de contenidos que se imparten por parte del profesor a partir de las diferentes formas de organización de la clase tales como: conferencias, talleres, seminarios, clases prácticas y clases prácticas de laboratorios.

A partir de un estudio exploratorio realizado a los docentes que imparten las clases prácticas de laboratorio en la asignatura de Bases de Datos I y Bases de Datos II, se pudo constatar que aun cuando se reconocen las particularidades del trabajo del profesor con los estudiantes, se presentan limitaciones:

- Las herramientas utilizadas en las clases prácticas de laboratorio de las asignaturas Sistemas de Bases de Datos I y II no permiten la asignación de tareas, por lo que el profesor debe auxiliarse de otros medios informáticos para orientar los ejercicios o divulgar los scripts que se utilizarán.
- Al finalizar cada turno de clases se hace necesario salvar los ejercicios resueltos. La herramienta utilizada en las asignaturas Sistemas de Bases de Datos I y II no le facilita el control al profesor de los mismos, pues esta salva se realiza por cada estudiante de forma individual.
- Los estudiantes deben auxiliarse de la realización de backup a la Bases de Datos donde se trabajó durante la clase para mantener la información de los ejercicios realizados.
- La herramienta utilizada en las asignaturas Sistemas de Bases de Datos I y II no permite al estudiante enviar al profesor los ejercicios desarrollados en el estudio individual, al ausentarse a una actividad de clase práctica de laboratorio. Para ello tiene que auxiliarse de otras vías informáticas.

Por parte de los profesores se constatan también dificultades tales como:

- Las herramientas utilizadas en las asignaturas Sistemas de Bases de Datos I y II, no permiten a los profesores agrupar sus estudiantes y asignar tareas atendiendo a la necesidad de la clase.
- Las herramientas utilizadas en las asignaturas Sistemas de Bases de Datos I y II, no facilita una sistematicidad del desarrollo de las habilidades alcanzadas por cada estudiante durante el curso.

- En caso de ausencia del profesor, quien pasa a trabajar con el grupo no tiene las respuestas a las actividades, ni las evidencias de las habilidades desarrolladas de cada estudiante durante todo el transcurso de la asignatura.

Dada la **situación problemática** anteriormente descrita surge como **problema a resolver**: ¿Cómo contribuir al flujo de trabajo de las clases prácticas de laboratorio en las asignaturas de Sistemas de Bases de Datos I y II?

Luego del análisis de la situación actual, la investigación enmarca su **objeto de estudio** en las herramientas que interactúan con sistemas gestor de bases de datos vinculados a la docencia, centrándose el **campo de acción** en las herramientas que interactúan con el sistema gestor de bases de datos PostgreSQL que se utilizan en las clases prácticas de laboratorio de las asignaturas Sistemas de Bases de Datos I y II en la UCI.

Encaminado a resolver el problema se determinó como **objetivo general de la investigación**: elaborar una herramienta web que permita interactuar con el Sistema Gestor de Bases de Datos PostgreSQL, para apoyar el desarrollo de las clases prácticas de laboratorio en las asignaturas de Sistemas de Bases de Datos I y II.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- Realización de un análisis sobre el funcionamiento de las clases en laboratorios de la asignatura de Sistemas de Bases de Datos, la existencia de herramientas web de apoyo a la docencia a nivel internacional y nacional, así como las tendencias tecnológicas actuales.
- Definición de herramientas, tecnologías y metodologías a utilizar en el desarrollo de la herramienta de apoyo a la docencia para perfiles de usuarios en las clases prácticas de laboratorio en la asignatura de Sistemas de BD en la UCI.
- Realización del análisis y el diseño de la herramienta web.

- Implementación de la herramienta informática propuesta utilizando las herramientas y tecnologías definidas teniendo en cuenta los requisitos expuestos por el cliente.
- Realización de pruebas al sistema para verificar el funcionamiento de la herramienta.

### **Métodos investigativos:**

Para llevar a cabo la investigación, se emplean los métodos de nivel teórico y empírico.

### **Métodos teóricos:**

- ✓ **Análisis Histórico-Lógico:** se utiliza este método para realizar el estudio del estado del arte, o sea para investigar acerca de otras aplicaciones o soluciones similares y de los lenguajes y metodologías de desarrollo de software existentes, describir la metodología, herramientas y lenguaje a utilizar en el análisis, diseño e implementación del sistema.
- ✓ **Método Analítico-Sintético:** se utiliza para analizar los elementos esenciales de la bibliografía consultada durante la descripción de los conceptos asociados al dominio del problema.
- ✓ **Modelación:** se utiliza para la creación de diagramas. Mediante su utilización se podrán elaborar diferentes tipos de diagramas que brindarán información clara sobre el tema de estudio.

### **Métodos empíricos:**

- ✓ **Observación:** se utiliza para diagnosticar el uso de los sistemas gestor de bases de datos durante las clases.

Para una mejor comprensión de la investigación, cuyo diseño metodológico se acaba de describir, se decidió definir una estructura capitular que aporte cierto grado de organización y facilite el estudio del documento. Los capítulos que la conforman, son los siguientes:

### **Capítulo 1: Fundamentación teórica**

En este capítulo se abordan las tendencias tecnológicas, metodologías de desarrollo de software y aplicaciones informáticas necesarias para dar solución al problema planteado.

### **Capítulo 2: Características del sistema**

En este capítulo se definen los actores del sistema, así como, los requisitos funcionales y no funcionales y las funcionalidades principales.

### **Capítulo 3: Diseño del sistema**

Este capítulo aborda la modelación y construcción detallada de la estructura de la herramienta. Además, se exponen pruebas realizadas a la aplicación una vez finalizado el desarrollo y el resultado de las mismas.

## Capítulo 1: Fundamentación teórica

En el presente capítulo se expone el marco teórico que sustenta la investigación realizada. Se relacionan los conceptos para el entendimiento de la situación problemática. Se analizan las herramientas, metodología y tecnologías necesarias para la implementación de la solución, justificando a su vez la selección y utilización de cada una de ellas.

### 1.1. Concepto general de Bases de Datos

Desde el surgimiento de la actividad humana, el hombre dedicó espacio para organizar las formas de control de sus recursos. El sostenido desarrollo de la información durante siglos ha impuesto el perfeccionamiento de ese control.

Con la ayuda de las (TIC), como parte de la tecnología de avanzada que el progreso social genera, se accede de forma privilegiada a la información. Por esta razón resulta imposible permanecer al margen del uso crítico de estas herramientas, máxime si se trata de las clases prácticas de laboratorio en las asignaturas Sistemas de Bases de Datos I y II.

Se hace indispensable el control de datos e informaciones y agilizar su procesamiento. La capacidad de los ordenadores personales para almacenar, controlar y procesar datos abrió el campo de uso de las Bases de Datos, así se estableció el vínculo entre los datos, la información y el control, cimiento de las BD.

A partir de la interrogante **¿Qué es una BD?** se le da respuesta a la conceptualización de BD teniendo en cuenta el criterio de diferentes autores.

- *Según el Departamento de Ciencias de la Computación de la Universidad de Granada: Fondo común de información almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independientemente de su lugar de procedencia y del uso que haga de ella.*
- *Según la Universidad Abierta de Cataluña en el artículo Bases de datos en su primera edición del año 2005 de los autores Rafael Campos Pare,*

*Luis Alberto Casillas Santillán, Dolores Costal Costa, Marc Giber Ginesta, Carmen Martin Escofet, Oscar Pérez Mora: es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.*

- *Para Gener (2005) las Bases de Datos “son un conjunto de datos organizados, que tienen una estructura lógica y están relacionados entre sí, a los cuales se le puede acceder con facilidad con diferentes propósitos”.*

Estos autores coinciden en plantear que las Bases de Datos son conjuntos de datos organizados almacenados en una computadora, que pueden ser utilizadas por cualquier persona y programa, que acceden con facilidad con diferentes propósitos.

## **1.2. Estudio del estado del arte**

La gestión de BD en el mundo es cada vez más imprescindible para las nuevas aplicaciones y necesidades del mercado. Para lograr un buen desempeño y mejor funcionamiento de la aplicación a desarrollar, se realizó un estudio de los diferentes sistemas que se han desarrollado en el mundo y en Cuba con el mismo fin, con el objetivo de compararlos y tomar experiencias de cada uno.

### **1.2.1. Estudio de herramientas que interactúan con un Sistema Gestor de Bases de Datos en el mundo**

#### **PhpPgAdmin**

Es una aplicación web que provee una manera conveniente a los usuarios para crear bases de datos, tablas, alterarlas y consultar sus datos usando el lenguaje estándar SQL. phpPgAdmin estuvo basado en PhpMyAdmin, pero hoy día ya no comparte código con él; incluso provee las mismas funcionalidades y más a los usuarios del servidor de base de datos PostgreSQL. (2)

## **SQLPro for Postgres**

Es un avanzado, pero ligero cliente de base de datos PostgreSQL, que permite un acceso rápido y sencillo a servidores Postgres.

Las características incluyen:

- Resaltado de sintaxis.
- Interfaz basada en pestañas.
- Soporte para ejecutar múltiples consultas a la vez.
- Acceso rápido a tablas, columnas o filas.
- Requiere: 64 bits - MacOS 10.9 o superior. (3)

## **Adminer.php**

Adminer.php es un administrador de bases de datos muy útil y sencillo de utilizar, sobre todo si no tiene acceso a PhpMyAdmin o necesita hacer algún cambio o consulta en la base de datos de una tienda on-line o página web, sea cual sea la plataforma CMS que está utilizando, tanto en WordPress, Joomla, Prestashop o Magento. (4)

## **Navicat for PostgreSQL**

Navicat for PostgreSQL es una herramienta de fácil uso para la administración gráfica de la base de datos PostgreSQL. Permite crear y ejecutar consultas y funciones, además de gestionar datos. Navicat for PostgreSQL conecta a cualquier servidor local / remoto de base de datos PostgreSQL desde la versión 7.3 o superior. Es compatible con bases de datos Cloud como Amazon RDS y Amazon Redshift, y con todos los objetos de bases de datos PostgreSQL.

Navicat for PostgreSQL está diseñada escribir consultas SQL sencillas o complejas de bases de datos a desarrollar. Sus características principales incluyen SQL Constructor / Editor, herramienta de Formato de Datos, Transferencia de Datos, Importación / Exportación, Sincronización de Datos / Estructura, Informes y mucho más. (5)

## **PhpMyAdmin**

Es una herramienta que puede administrar un servidor MySQL entero (necesita un superusuario) o una base de datos sencilla. Para realizar esto último, es necesario tener un usuario MySQL debidamente configurado que pueda leer o escribir solamente en la base de datos deseada.

En la actualidad, PhpMyAdmin tiene algunas funcionalidades como:

- Visualizar y borrar bases de datos, tablas, vistas, campos e índices
- Mostrar múltiples resultados a través de procedimientos almacenados o consultas
- Crear, copiar, borrar, renombrar y alterar bases de datos, tablas, campos e índices
- Realizar labores de mantenimiento de servidor, bases de datos y tablas, dando consejos acerca de la configuración del servidor
- Ejecutar, editar y marcar cualquier expresión SQL, incluyendo consultas en lote
- Carga tablas con el contenido de ficheros de texto (6)

## **Microsoft® SQL Server™**

Es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. Microsoft SQL Server 2014 se basa en las funciones críticas ofrecidas en la versión anterior, proporcionando un rendimiento, una disponibilidad y una facilidad de uso innovadores para las aplicaciones más importantes. Microsoft SQL Server 2014 ofrece nuevas capacidades en memoria en la base de datos principal para el procesamiento de transacciones en línea (OLTP) y el almacenamiento de datos, que complementan nuestras capacidades de almacenamiento de datos en memoria para lograr la solución de base de datos en memoria más completa del mercado.

SQL Server 2014 también proporciona nuevas soluciones de copia de seguridad y de recuperación ante desastres, así como de arquitectura híbrida

con Windows Azure, lo que permite a los clientes utilizar sus actuales conocimientos con características locales que aprovechan los centros de datos globales de Microsoft. Además, SQL Server 2014 aprovecha las nuevas capacidades de Windows Server 2012 y Windows Server 2012 R2 para ofrecer una escalabilidad sin parangón a las aplicaciones de base de datos en un entorno físico o virtual. (7)

### **1.2.2. Estudio de herramientas que interactúan con un Sistema Gestor de Bases en Cuba**

**Aplicación educativa bajo ambiente web, como apoyo para la enseñanza de la asignatura BD orientados a objetos:** Este fue el título de un trabajo investigativo desarrollado en la Universidad de Oriente en el año 2009 para la carrera de licenciatura en informática. Dicho estudio no se ha puesto en práctica hasta el momento.

#### **PgAdmin III**

Es un completo sistema de gestión y diseño de bases de datos PostgreSQL para sistemas Unix y Windows. Está libremente disponible bajo los términos de la licencia de PostgreSQL y puede ser redistribuida siempre que se cumplan los términos de la licencia. El proyecto es administrado por el equipo de desarrollo PgAdmin.

Este software fue escrito como un sucesor de los productos originales PgAdmin y PgAdmin II, que, aunque popular, tenía limitaciones en el diseño que les impidió ser llevado al 'siguiente nivel'. (8)

Esta herramienta se utiliza en las asignaturas Sistemas de Bases de Datos I y Sistemas de Bases de Datos II.

#### **Valoración del estudio del estado del arte**

Se entiende que estas herramientas existentes en el ámbito nacional e internacional brindan grandes facilidades para la interacción con las Bases de Datos. Pero las mismas no cuentan con funcionalidades que permitan asignar tareas o gestionar usuarios, tampoco almacenan información referente a los

usuarios que la utilicen. Por lo que se concluye no asimilar ninguna de las herramientas anteriormente mencionadas ya que se requeriría mucho esfuerzo en interpretar el código e insertarle nuevas funcionalidades. Además, muchas de ellas son de software libre pero no son de código abierto.

### **1.3. Herramientas, tecnologías y las metodologías de desarrollo de software a utilizar**

#### **1.3.1. Herramientas y lenguaje de modelado**

**Visual Paradigm v8.0:** es una herramienta para desarrollo de aplicaciones utilizando modelado UML\* ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.

Ventajas de Visual Paradigm:

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática Ad-hoc.
- Ambiente visualmente superior de modelado.
- Sofisticado diagramador automáticamente de layout. (9)

**Enterprise Architect v7.1:** Es la herramienta de modelado de procesos que permite visualizar, entender, analizar, mejorar y documentar los procesos de negocio, el flujo de documentos y la información dentro de la organización. Soporta el modelado de procesos de negocio con la notación BPMN (Business Process Modeling Notation), el modelado del Flujo de datos o documentos con la notación DFD tradicional (Data Flow Diagram), así como el modelado de datos mediante los requisitos de ERD (Entity Relationship Diagram). Esta herramienta reduce significativamente el esfuerzo a realizar en el modelado mientras la documentación ordenada muestra en qué forma se ejecutan los procesos dentro de su empresa. (10)

**Lenguaje Unificado de Modelado:** fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, una de ellas es en el flujo de procesos en la fabricación. (11)

**Notación de Modelado de Procesos de Negocios:** (BPMN por sus siglas en inglés es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD). (12)

Resultando que los estudiados sobre estas herramientas permiten el modelado BPMN (dado sus siglas en inglés, Business Process Modeling Notation). Por lo que se determinó utilizar el Visual Paradigm para el modelado de los procesos del negocio a informatizar.

### 1.3.2. Paradigma de programación

Un paradigma de programación es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.

¿Cuáles son los principales paradigmas de programación?

- **Imperativo.** Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.

- **Declarativo.** Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.
- **Lógico.** El problema se modela con enunciados de lógica de primer orden.
- **Funcional.** Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.
- **Orientado a objetos.** El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

Otros son de aparición relativamente reciente y no forman parte del grupo principal:

- **Dirigido por eventos.** El flujo del programa está determinado por sucesos externos (por ejemplo, una acción del usuario).
- **Orientado a aspectos.** Apunta a dividir el programa en módulos independientes, cada uno con un comportamiento bien definido.

Cada paradigma es ideal para la resolución de un conjunto de problemas en particular, por lo que no puede decirse que uno sea necesariamente mejor que otro. (13)

### **Programación orientada a objetos (POO)**

La orientación a objetos es uno de los muchos paradigmas de programación que existen. Un paradigma de programación es un modelo que guía la resolución de problemas computacionales, y que en muchos casos define la sintaxis de un lenguaje. Es decir, a través de la sintaxis que impone, un lenguaje de programación puede orientarse hacia algún estilo de resolución de problemas particular. Si bien la mayoría de los lenguajes combina características de varios paradigmas, el más popular en la actualidad es el orientado a objetos, uno de los cuatro principales junto con el lógico, el imperativo y el funcional.

Tradicionalmente, los programas consistían en una sucesión de tareas a ejecutarse. Pero la programación orientada a objetos propone un modelo diferente: el programa se compone de un conjunto de entidades, llamadas objetos, que se comunican y realizan pedidos entre sí a través de mensajes. Cada objeto cuenta con características propias (atributos) y es capaz de realizar ciertas acciones (métodos). Si un objeto requiere obtener el resultado de cierta operación sobre un conjunto de datos, no tiene más que pedirselo a un objeto que pueda realizar esa operación. El objeto remitente no tiene por qué conocer la forma en que el objeto receptor desempeña la acción. Los objetos actúan como un equipo de trabajo con una estricta división de tareas.

El paradigma orientado a objetos permite modelar un problema de una forma amena, que reproduce con mayor fidelidad los mecanismos del mundo real. Todos los elementos que componen el Universo tienen características y comportamiento propios, y se relacionan con otros elementos de múltiples formas; por lo tanto, pueden ser fácilmente representados como objetos. Crear un conjunto de objetos que refleje adecuadamente el problema real, y que sepan organizarse entre sí para resolverlo, depende en buena medida de la imaginación del programador. (14)

### **1.3.3. Tecnologías y lenguajes de programación del lado del servidor**

**Python:** es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (15)

**ASP.NET:** es un modelo de desarrollo web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common

Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, entre otros. (16)

**PHP:** El lenguaje de programación PHP (cuyo nombre proviene de *PHP: Hypertext Preprocessor*) es un lenguaje interpretado con una sintaxis similar a la de otros lenguajes de programación como C++ o Java.

Aunque este lenguaje de programación se podría utilizar para realizar cualquier tipo de programa, es en la generación dinámica de páginas web donde ha alcanzado su máxima popularidad. En concreto, suele incluirse código PHP incrustado en las páginas HTML, siendo el servidor web el encargado de ejecutar este código antes de servir la página. (17)

Atendiendo a lo anteriormente descrito de cada uno de los lenguajes de programación, se puede hacer uso indistintamente de los mismos en el desarrollo de la herramienta. Debido a las políticas de seguridad de la UCI ningún servidor hosteará una aplicación desarrollada en ASP.NET, por lo que el mismo queda descartado. A partir de los anterior, se decide utilizar el lenguaje PHP.

### **¿Utilizar PHP puro o un *framework* de desarrollo?**

Primero debemos analizar que es un *framework*, la palabra inglesa *framework* (infraestructura, armazón, marco) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Otra definición:” es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Sí, es una definición muy genérica, pero también puede serlo un *framework*: sin ir más lejos, el paradigma MVC (Model-View-Controller) dice poco más que “separa en tu aplicación la gestión de los datos, las operaciones, y la presentación”. En el otro extremo, otros *framework*

pueden llegar al detalle de definir los nombres de ficheros, su estructura, las convenciones de programación”

### ¿Que ventajas trae utilizar un *framework*?

- El programador no necesita plantearse una estructura global de la aplicación, sino que el *framework* le proporciona un esqueleto que hay que “rellenar”.
- Facilita la colaboración. Cualquiera que haya tenido que “pelearse” con el código fuente de otro programador (¡o incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al *framework* concreto para facilitar el desarrollo.
- Hacer posible el desarrollo de la velocidad.
- Proporcionar código bien organizado, reutilizable y mantenible.
- Promover prácticas modernas de desarrollo web tales como herramientas de programación orientadas a objetos.

Existen muchos *framework* de desarrollo en PHP algunos son mejores por sus funcionalidades otros por su rápido aprendizaje o por la rapidez con la que se desarrollan las aplicaciones. A continuación, se nombran algunos de ellos:

**Symfony:** Es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado con numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de los sistemas gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Sever de Microsoft. Se puede ejecutar en plataformas \*nix (Unix, Linux, etc.) (18)

**Zend Framework:** Zend Framework 2 es un framework de código abierto para desarrollar aplicaciones y servicios web utilizando PHP 5.3+. Zend Framework 2 utiliza código 100% orientado a objetos y utiliza la mayoría de las nuevas características de PHP 5.3, es decir, espacios de nombres, enlace estático tardío, funciones lambda y cierres. (19)

**CodeIgniter:** Es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro *framework*, CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales.

Se decidió utilizar Symfony debido a la gran comunidad existente en la UCI que trabaja con el *framework*. También se escogió la versión 2.8 aunque ya existen versiones superiores a la 3.0 por ser la versión más estable hasta el momento.

#### **1.3.4. Tecnologías y lenguajes de programación del lado del cliente**

**HTML5:** Quinta versión de HTML que reemplaza a HTML4 corrigiendo problemas con los que se encontraban los desarrolladores. Con la aparición del mismo se introducen cambios en la semántica que hacen que la estructura de la web sea más coherente y de fácil entendimiento. Se utiliza para describir

texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones de multimedia (gráfico, imágenes, sonido, video). Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, es decir, se utiliza para crear las páginas web y les indica a los navegadores cómo deben mostrar el contenido.

**JavaScript:** Lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario. Hoy es el motor de las aplicaciones más conocidas en el ámbito de Internet: Google, Facebook, Twitter. La Web 2.0 se basa en el uso de JavaScript para implementar aplicaciones enriquecidas que son capaces de realizar todo tipo de efectos, interfaces de usuario y comunicación asíncrona con el servidor.

### 1.3.5. Servidor web

Para el despliegue de la aplicación es importante montar un servidor web que cumpla con las necesidades de la aplicación. Para ello analizaremos los siguientes servidores web:

**WampServer:** Permite pre visualizar sitios web localmente e instalar un servidor web en Windows, así como administrar configuraciones de servidores, crear aplicaciones web, gestionar BD y utilizar lenguaje PHP.

**Xampp:** Es un servidor de plataforma libre, es un software que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de BD MySQL, un servidor de FTP FileZilla, el popular administrador de BD escrito en PHP, MySQL, entre otros módulos.

Atendiendo a las características de cada uno se definió la utilización del servidor web WampServer. También porque el Xampp al iniciarse tiene muchas funcionalidades innecesarias para el equipo de desarrollo, lo que disminuye el rendimiento de la máquina.

### 1.3.6. Entorno de Desarrollo Integrado (IDE)

**NetBeans:** Es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

**PhpStorm:** Es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación php como lo indica su nombre.

El PhpStorm brinda un buen autocompletado de código y permite una gran navegabilidad dentro de su interfaz, cosa que no se puede hacer en el NetBeans, por lo que se decidió escoger el PhpStorm.

### 1.3.7. Metodología de desarrollo

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos. (20)

## **Otras metodologías ágiles**

### **Extreme Programming**

La programación extrema o Extreme Programming (XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (21)

### **SCRUM**

Define un marco para la gestión de proyecto, que se ha utilizado con éxito durante los últimos 10 años. Esta especialmente indicada para proyectos con un rápido cambio en los requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprints, con una duración de 30 días. El resultado de cada Sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (22)

## **RUP**

El Rational Unified Process o Proceso Unificado de Racional. Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo que es enfocada hacia “diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal. (23)

### **Valoración de las metodologías**

Se decidió utilizar una metodología ágil ya que se necesita rapidez para un producto viable que pueda ser probado por el usuario en el menor tiempo posible. Flexibilidad, las fases tradicionales de análisis, implementación, pruebas e implantación se entremezclan para programar el proyecto y el equipo de desarrollo es de dos integrantes. Entre estas se utilizó AUP porque aplica técnicas ágiles incluyendo desarrollo dirigido por pruebas, modelado ágil, gestión de cambios ágil, y refactorización de Base de Datos para mejorar la productividad.

### **Conclusiones del capítulo**

En este capítulo, se hizo un estudio de las herramientas que interactúan con los sistemas gestores de Bases de Datos y algunos que se utilizan específicamente en la docencia. Además, se hizo un análisis de los lenguajes y otras tecnologías que se utilizarán en el diseño, modelado e implementación de la solución informática. También se concluye con la utilización del lenguaje de programación Php y como framework de desarrollo Symfony 2.8. Además, se seleccionó el IDE PhpStorm y los lenguajes HTML5, CSS3 y la metodología de desarrollo de software AUP.

## Capítulo 2: Características del sistema.

Las herramientas de apoyo a la docencia son plataformas que buscan dar un apoyo a las actividades académicas que se realizan. El objeto de estas herramientas se basa en que puedan operar de acuerdo con la necesidad impuesta por el centro y la enseñanza en que se establece. Las actividades principales a realizar son: entrada, almacenamiento, control, procesamiento y evaluación de la información. Con estas se realiza una estructura de soporte para la administración del contenido por parte de los participantes. Además, permite la práctica y evolución de manera independiente.

### 2.1. Objeto de informatización

El sistema de clases de laboratorio de la asignatura de BD, es el mecanismo a informatizar, con la finalidad de que cada estudiante pueda tener un seguimiento de los ejercicios y actividades desarrolladas durante las clases de este tipo, así como también, los profesores les den seguimiento a sus estudiantes durante el semestre o el curso.

#### 2.1.1. Información que se maneja

**Estudiantes:** mantiene al estudiante informado sobre su desempeño académico en la asignatura de BD. Además, visualiza la respuesta a los ejercicios realizados y los errores cometidos.

**Profesor:** cuenta con la información de todos los estudiantes para la gestión de grupos, subgrupos y asignación de tareas a los estudiantes acorde a su desempeño.

### 2.2. Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés (24). Es utilizado por el analista como un medio para comprender el negocio al cual el sistema va a servir.

El modelo de dominio que se representa, describe cómo se desarrolla el proceso de clases en laboratorio de la asignatura de BD. Este mecanismo no

es más que la técnica ejercida por el profesor para llevar a cabo un mejor desempeño de los estudiantes durante las clases de laboratorio. Tiene como finalidad tener una buena información docente de los estudiantes respecto a la asignatura. Los parámetros del mismo, son explicados al inicio del curso. (Ver Figura 1)

**Estudiante:** persona que se dedica a la lectura, puesta en práctica y aprehensión de conocimientos sobre alguna materia o arte.

**Profesor:** persona que se dedica a la enseñanza y máximo responsable de la asignación de tareas a los estudiantes y dar seguimiento a los mismos.

**Facultad:** entidad que forma parte de una universidad y está conformada por estudiantes y profesores.

**Grupo:** pluralidad de individuos que se relacionan entre sí, con un cierto grado de interdependencia, que dirigen su esfuerzo a la consecución de un objetivo común, con la convicción de que juntos pueden alcanzar este objetivo mejor que en forma individual. En este caso se refiere a un grupo de estudiantes.

**Tareas:** ejercicio o compendio de ejercicios que los profesores asignan a los estudiantes o grupos de estudiantes.

**Subgrupo de trabajo:** conjunto de estudiantes definido por un profesor según las necesidades académicas.

## Diagrama de Clases del Modelo del Dominio

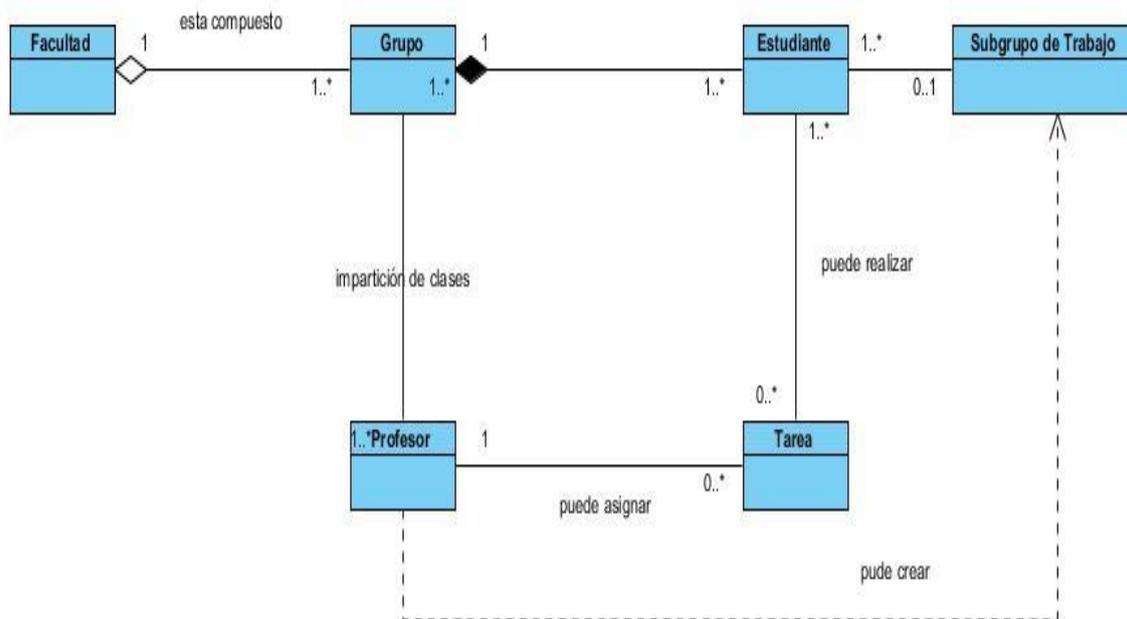


Figura 1. Diagrama de Clases del Modelo del Dominio.

### 2.3. Reglas del negocio

Durante el desarrollo del negocio se determinó establecer un conjunto de reglas, estas deben tomarse en cuenta al llevar a cabo la informatización de los procesos del negocio, esto permite que el sistema elaborado se ajuste y garantice el cumplimiento de las necesidades del cliente.

1. De los estudiantes, se debe tener toda la información relacionada a su persona, ya sea: grupo, facultad, solapín, nombre y apellidos.
2. De los profesores, se debe tener grupo o grupos a los que le imparte clases, facultad o área de trabajo, solapín, nombre y apellidos.
3. Tener una restricción de acceso a información según rol que quiera acceder a la misma.
4. Un profesor puede gestionar una evaluación durante el transcurso de la clase o durante el transcurso de la asignatura en el semestre.
5. Se consideran tareas a un ejercicio o compendio de ejercicios asignados por el profesor a un estudiante o un subgrupo de estudiantes durante el transcurso de la asignatura.

6. Se considera subgrupo a un conjunto de estudiantes designados por el profesor para el cumplimiento de tareas específicas.
7. Para que el profesor pueda orientar una tarea a un estudiante o subgrupo o crear subgrupos de trabajo, los estudiantes deben haberse autenticado al menos una vez en el sistema

### **2.3.1. Trabajadores del negocio**

Los trabajadores del negocio (Profesor y Estudiante) son las personas que serán los futuros usuarios dentro de la Herramienta Web Sistema de Ayuda a la Asignatura de BD.

### **2.3.2. Entidad del negocio**

Documento o sistema del cual se consulta o se toma información significativa, que es manipulada por los actores y trabajadores del negocio. Las entidades del negocio de la Herramienta de apoyo a la docencia, representa las entidades reales: Tareas. Estas entidades son la base para compartir documentos entre los profesores y estudiantes. (Ver Figura2)

<b>Trabajadores</b>	<b>Justificación</b>
Profesor	Es el encargado de asignar las tareas pertinentes a los estudiantes, así como crear los subgrupos de trabajo.
Estudiante	Es el encargado de dar respuesta a las tareas asignadas por el profesor durante el transcurso de la clase o una vez concluida la misma.

**Tabla 1: Descripción de los Trabajadores del negocio.**

## **2.4. Especificación de los requisitos de software**

### **2.4.1. Requerimientos Funcionales:**

**RF 1:** Controlar el acceso a la información correspondiente a cada usuario.

1.1 Definir niveles de acceso.

**RF 2:** Gestionar subgrupos.

2.1 Insertar estudiantes a un grupo.

2.2 Eliminar estudiantes de un grupo.

2.3 Mostrar estudiantes del grupo.

**RF 3:** Asignar tareas.

3.1 Asignar tarea a un estudiante.

3.2 Asignar tarea a un grupo.

3.3 Asignar tarea a un subgrupo.

**RF 4:** Gestionar evaluación

4.1 Asignar una evaluación.

4.3 eliminar una evaluación

**RF 5:** Mostrar tareas.

**RF 6:** Mostrar evaluación.

**RF 7:** Inserta tarea.

**RF 8:** Insertar respuesta.

#### **2.4.2. Requisitos no Funcionales**

**RNF 1:** Usabilidad.

1.1 Documentar adecuadamente la aplicación. Esto proporciona que cada usuario pueda satisfactoriamente utilizar las funcionalidades que brinda la misma.

1.2: Adiestrar al personal. El personal que hace uso del sistema, debe poseer la capacitación requerida al nivel que se establece.

1.3: Definir nivel de acceso. El sistema sólo puede ser usado por los estudiantes y profesores que reciban o impartan la asignatura de BD.

1.4: Tener en cuenta la experiencia del usuario, la aplicación debe ser fácil de usar.

## **RNF 2: Fiabilidad**

2.1: Acceder para dar mantenimiento al sistema en horarios de menos usabilidad.

2.2: Brindar prestaciones cuando esté en modo de funcionamiento con problemas.

2.3: Especificar la salida del sistema, el usuario debe dirigirse a la parte superior de la interfaz que se observa, a la opción **Salir** y confirmar la salida.

2.4: Categorizar los errores. Categoría definida: menores, significativos y críticos. La primera categoría es asignada cuando se produzca un error producto al funcionamiento en la red o la electricidad del local donde se imparte la clase. La segunda se asigna cuando existan problemas en el nivel de acceso a la información y la tercera cuando haya pérdida total de la información a registrar o esté inhabilitado alguna funcionalidad del sistema.

## **RNF 3: Eficiencia.**

3.1: Implementar sobre la tecnología Web las prestaciones, para facilitar su uso a través de la red.

## **RNF 4: Soporte.**

4.1 Utilizar como medio de almacenamiento de la información, una Base de Datos soportada sobre el Sistema Gestor de Bases de Datos (SGBD) PostgreSQL.

4.2: Integrar el sistema de manera que se mantenga la colaboración entre todos los subsistemas que lo componen.

## **RNF 5:** Restricciones de diseño e implementación.

5.1: Definir la lógica de presentación que constituye una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de los datos de entrada.

5.2: Validar el proceso de la captación de datos para evitar entradas inadecuadas.

5.3: Utilizar PostgreSQL como Sistema Gestor de BD.

5.4: Utilizar como *framework* Symfony.

## **RNF 6:** Interfaz.

### 6.1: Interfaces de usuario

6.1.1: Realizar interfaces con un diseño sencillo que contenga pocos gráficos, para así acelerar las respuestas del sistema al usuario y brindarle mayores posibilidades de adaptación al sistema.

6.1.2: Los estilos CSS usados fueron definidos por los desarrolladores.

### 6.2: Interfaces Hardware.

6.2.1: El usuario requiere una máquina con un procesador 600 mega Hertz (MHz) como mínimo y se recomienda 800 (MHz).

6.2.2: Memoria RAM como mínimo 256 megabytes (MB) y se recomienda 512 MB.

6.2.3: Requiere una capacidad en el disco duro de 20GB como mínimo.

6.2.4: Monitor VGA o superior.

6.2.5: Ratón Microsoft o compatible.

### 6.3: Interfaces Software.

6.3.1: El sistema debe ser multi plataforma.

6.3.2: Navegador Web Mozilla Firefox, Internet Explorer o Chrome.

6.3.3: Se utiliza el LDAP como protocolo de ligero acceso a directorios.

**RNF 7:** Requisitos Legales, de Derecho de Autor y otros.

7.1 Establecer derechos. Esta herramienta de ayuda para la docencia en la asignatura de BD pertenece a la UCI, específicamente al Departamento de Ingeniería de Software de la Facultad 2.

**RNF 8:** Estándares Aplicables.

8.1 Definir un estándar de codificación para lograr la reutilización de código en el equipo de trabajo y un satisfactorio entendimiento del código desarrollado.

**RNF 9:** Portabilidad.

9.1: Utilizar componentes que puedan tener funcionamiento en sistemas operativos como Microsoft® Windows® XP o superior y/o GNU-Linux.

**RNF 10:** Seguridad.

10.1: Garantizar que la información sea editada únicamente por el personal autorizado. Que contenga verificación sobre acciones irreversibles (eliminaciones).

10.2: Emplear métodos para asegurar la integridad de los datos.

10.3: Contar con varios niveles de acceso para permitir el trabajo organizado con el sistema.

10.4: Guardar de modo confidencial la información de los estudiantes, por lo que sólo podrá ser consultada por el personal autorizado y está prohibida su divulgación.

10.5: Realizar la comunicación usando un protocolo seguro. (Https).

**RNF 11:** Requisitos de Disponibilidad.

11.1: La herramienta debe estar disponible 24 horas y para dar mantenimiento a la misma utilizar horarios que no sean los establecidos para las clases

**RNF 12:** Rendimiento.

12.1: El tiempo de respuesta de cada una de las páginas debe ser menor de un minuto, excepto aquellas que requieren más tiempo.

12.2: Se implementa sobre tecnología web, facilitando su uso en la red.

## 2.5. Definición de casos de uso y actores del sistema

- Definición de los actores

Actores	Justificación
Profesor	Puede ver el estado de las actividades, emitir una evaluación de los grupos a los que le imparte clase.
Usuario	Generaliza el rol de autenticación al sistema y puede obtener acceso según los permisos que tenga, dígase estudiante, profesor, jefe de asignatura, jefe de disciplina, jefe de departamento.
Registro de Identidad UCI	Brinda un grupo de servicios que pueden ser reutilizados. Mediante él se pueden obtener una persona dado el usuario y contraseña.
Estudiantes	Puede ver sus actividades asignadas, sus evaluaciones.

**Tabla 2: Definición de actores del sistema**

### 2.5.1. Casos de Uso del Sistema

CUS-1	Autenticar Usuario
-------	--------------------

Actor	Usuario
Descripción	Permite, una vez que el usuario introduzca los datos que se le piden para acceder a la aplicación, que éstos sean verificados y admitan que se le den los permisos al usuario, según su rol y se le habilite la entrada a la aplicación con las operaciones que puede realizar.
Referencia	RF1

**Tabla 3: Descripción del caso de uso Autenticar Usuario.**

CUS-2	Gestionar Grupo
Actor	Profesor
Descripción	Permite que una vez que el profesor este autenticado, atendiendo a la necesidad de la clase pueda crear, eliminar o modificar subgrupos de trabajo para el desarrollo de las tareas.
Referencia	RF2

**Tabla 4: Descripción del caso de uso Gestionar grupo**

CUS-3	Asignar Tareas
Actor	Profesor
Descripción	Permite que el profesor asigne las tareas pertinentes a los estudiantes en específico o a los subgrupos de trabajo
Referencia	RF3

**Tabla 5: Descripción del Caso de Uso Asignar Tarea.**

CUS-4	Gestionar evaluación
Actor	Profesor
Descripción	Permite que una vez autenticado el profesor asigne, modifique o elimine una evolución de acuerdo con su criterio
Referencia	

**Tabla 6: descripción del caso de Uso Gestionar evolución.**

CUS-5	Revisar tareas
Actor	Estudiantes
Descripción	Permite al estudiante observar la tarea q le fue asignada durante la clase
Referencia	

**Tabla 7: descripción del caso de uso Revisar tareas**

CUS-6	Revisar evaluación
Actor	Estudiantes
Descripción	Permite al estudiante observar cuáles son sus evaluaciones durante el curso
Referencia	

**Tabla 8: descripción del caso de uso Revisar evaluación**

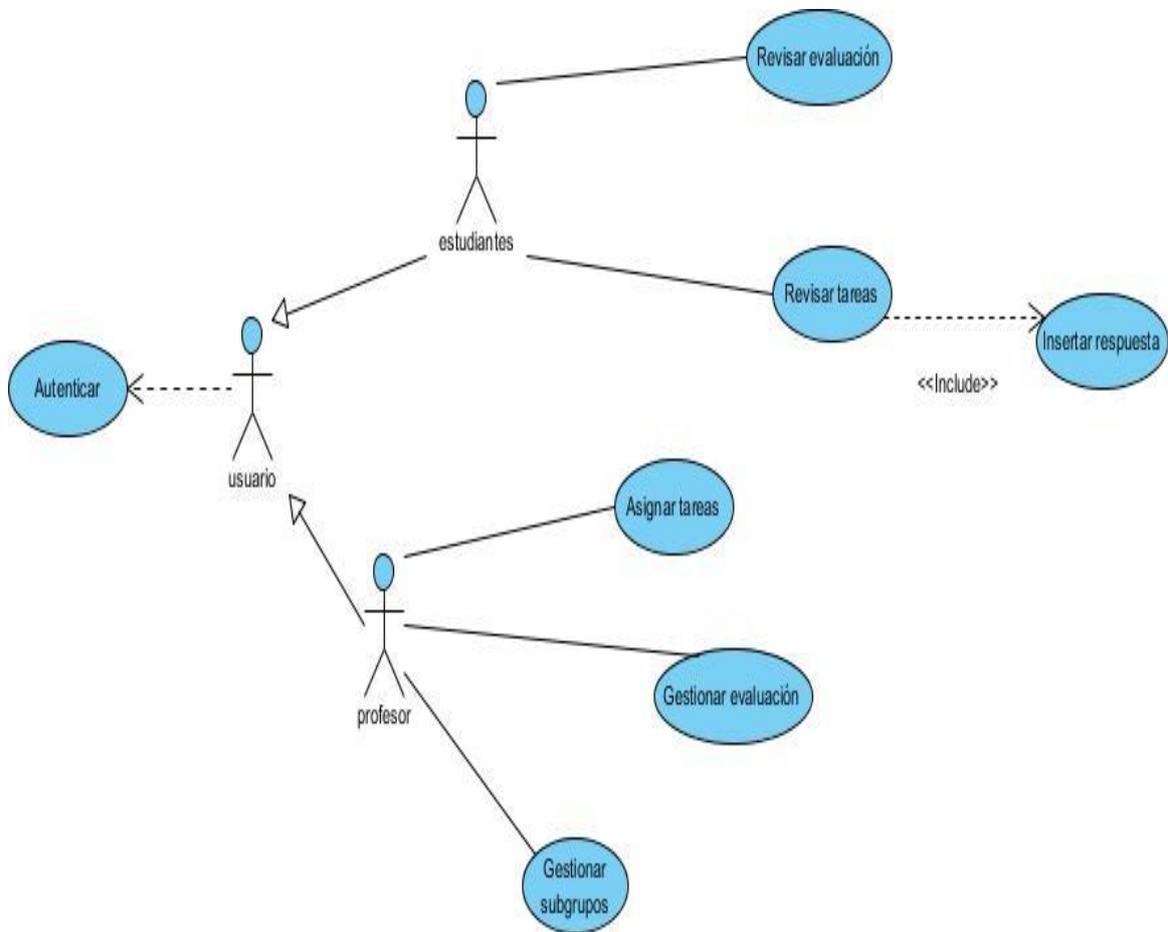
CUS-8	Insertar respuesta
Actor	Estudiantes
Descripción	Permite a los estudiantes dar

	respuesta a las tareas orientadas
Referencia	

**Tabla 9: Descripción del Caso de Uso Insertar respuesta.**

### 2.5.2. Diagrama de Casos de Uso

Los Casos de Uso describen como se ve el sistema del punto de vista del usuario. Es decir, como interactúa con el usuario. Al aumentar el número de casos de uso no resulta fácil relacionarlos. De ahí la necesidad de tener una vista más general, de donde viene la gran utilidad de diagramas de casos de uso.



**Figura 2: Diagrama de Caso de Uso del Sistema.**

### 2.5.3. Casos de uso expandidos

<b>Caso de uso</b>
--------------------

<b>CU-1</b>	Autenticar usuario	
<b>Propósito</b>	Permite autenticarse	
<b>Actor:</b> Usuario		
<b>Resumen:</b> El caso de uso se inicia cuando el usuario desea acceder a la aplicación. Para autorizar su entrada a la aplicación el sistema le solicita los datos de usuario y contraseña. Estos son verificados para darle los permisos según el rol que desempeña cada usuario. El caso de uso finaliza cuando se le habilita la entrada y los permisos al usuario a la aplicación.		
<b>Referencias</b>	RF1	
<b>Flujo de sucesos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. El usuario entra Usuario y Contraseña.	2. Es sistema encripta la contraseña. 3. Busca el usuario y compara la contraseña. 4. En caso de ser correcto se verifica el cargo que ocupa el usuario en la facultad. 5. Una vez que se confirma el cargo se le asignan los permisos. 6. Muestra la interfaz del usuario y con las operaciones a las que tiene acceso.	
<b>Flujo alternativo</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
	5. En caso de no existir se envía un mensaje de error.	

**Tabla 10: Descripción del CU Expandido Autenticar Usuario**

<b>Caso de uso</b>	
<b>CU-2</b>	Gestionar subgrupos
<b>Propósito</b>	Permite reagrupar los estudiantes según la necesidad del profesor o de la clase
<b>Actor:</b> Profesor	
<b>Resumen:</b> El caso de uso inicia una vez que el profesor esta autenticado en el sistema, el mismo escoge los estudiantes pertenecientes al subgrupo de trabajo antes o durante la clase. Después de la creación del grupo el profesor lo puede eliminar o modificar. El caso de uso finaliza cuando el profesor asigne las tareas a los estudiantes.	
<b>Referencias</b>	RF2
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El profesor selecciona la opción Crear Subgrupo de Trabajo que le brinda la aplicación en el menú.	2. El sistema permite crear un grupo, modificarlo o eliminarlo.  3. Al asignar un estudiante a un subgrupo de trabajo este debe realizar solamente las tareas asignadas al mismo.  4. El sistema brinda la posibilidad de que un estudiante este asignado a varios subgrupos.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

**Tabla 11: Descripción del CU Expandido Gestionar Subgrupos de Trabajo**

<b>Caso de uso</b>	
<b>CU-1</b>	Asignar Tareas
<b>Propósito</b>	Asignar una tarea a un grupo o un estudiante específico
<b>Actor:</b> Profesor	
<b>Resumen:</b> El caso de uso se inicia una vez que el profesor a definido la estructura de la clase, ya sea, por subgrupos de trabajo o individual, se asignan las tareas pertinentes y finaliza cuando se emite una evaluación de la misma.	
<b>Referencias</b>	RF3
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1.El profesor asigna una o varias tareas presionando el botón asignar tareas en el menú.</p> <p>4. El profesor una vez cumplido el tiempo para resolver las tareas realiza una evaluación de la misma.</p>	<p>2.Los estudiantes pueden revisar las tareas asignadas.</p> <p>3.Después de la revisión y aclaración de lo que se debe hacer con las tareas se da paso a resolverlas durante la clase o después de la misma.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

**Tabla 12: Descripción del CU Expandido Asignar tareas a Subgrupos de Trabajo.**

<b>Caso de uso</b>
--------------------

<b>CU-6</b>	Gestionar evaluación	
<b>Propósito</b>	Permite al asignar , eliminar , modificar una evaluación de acuerdo a lo observado durante el transcurso del curso	
<b>Actor :</b> Profesor		
<b>Resumen:</b> el caso de uso comienza una vez q el profesor se encuentra autenticado en el sistema y tiene un criterio para emitir sobre el estudiante. El mismo escoge al estudiante que va a evaluar y puede eliminar, modificar o asignar la evolución. Finaliza cuando el profesor cierra la opción.		
<b>Referencia</b>		
<b>Flujo de sucesos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. el profesor selecciona la opción evaluaciones	2. el sistema permite asignar, modificar y eliminar una evaluación  3. las evaluaciones quedan registradas en el sistema	
<b>Flujo alternativo</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	

**Tabla 13: Descripción del CU expandido gestionar evaluación.**

<b>Caso de uso</b>	
<b>CU-7</b>	Revisar evaluación
<b>Propósito</b>	Permite observar la evolución emitida

<b>Actor:</b> estudiantes	
<b>Resumen:</b> El caso de uso comienza una vez que el estudiante se autentica y escoge la opción evaluaciones, observa las evaluaciones. Finaliza cuando el estudiante cierra la opción.	
<b>Referencia</b>	
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. el estudiante selecciona la opción evaluaciones	2. el sistema muestra todas las evaluaciones que tiene registradas el estudiante.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

**Tabla 14: Descripción del CU expandido revisar evaluación.**

<b>Caso de uso</b>	
<b>CU-</b>	Revisar tareas
<b>Propósito</b>	Permite observar la evaluación asignada por el profesor
<b>Actor:</b> estudiantes	
<b>Resumen:</b> el caso de uso comienza después que el profesor inserta la tarea, el estudiante, presiona la opción tareas y puede observar la tarea que le ha sido asignada.	

<b>Referencia</b>	
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. el estudiante presiona la opción tareas	2. el sistema muestra las tareas asignadas
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

**Tabla 15: Descripción del CU expandido Revisar tareas.**

<b>Caso de uso</b>	
<b>CU-6</b>	Insertar respuesta
<b>Propósito</b>	Permite dar respuesta a las tareas asignadas.
<b>Actor</b> : Profesor	
<b>Resumen:</b> El caso de uso comienza cuando se le asigna una tarea a un estudiante o a un subgrupo, los mismos revisa la tarea y dan respuesta a las orientaciones.	
<b>Referencia</b>	
<b>Flujo de sucesos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El profesor signa una tarea	2. Los estudiantes revisan la tarea asignada  3. Los estudiantes responden las tareas orientadas
<b>Flujo alternativo</b>	

Acción del actor	Respuesta del sistema

**Tabla 16: Descripción del CU expandido Insertar respuestas.**

## 2.6. Patrón de arquitectura

### Modelo Vista Controlador

Es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

### Descripción de capas

#### Modelo

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes *selects*, *updates*, *inserts*, etc.

No obstante, cabe mencionar que cuando se trabaja con MCV lo habitual también es utilizar otras librerías como PDO o algún ORM como Doctrine, que nos permiten trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de usar directamente sentencias SQL, que suelen depender del motor de base de datos con el que se esté trabajando, se utiliza un dialecto de acceso a datos basado en clases y objetos.

#### Vista

Las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera.

## **Controlador**

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo. (25)

## **2.7. Patrones de diseño**

Symfony sigue la mayoría de mejores prácticas y patrones de diseño para la Web. Para dar solución a la implementación de los casos de uso planteados anteriormente se aplican los siguientes patrones de diseño que utiliza el *framework*.

### **2.7.1. Patrones GRASP**

- **Experto:** Expresa que los objetos hacen cosas de acuerdo a la información con que cuentan. Esto trae como beneficio que se conserve el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Además, favorece el bajo

acoplamiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida alentando con ello definiciones de clases sencillas que son fáciles de comprender y mantener.

- **Creador:** Permite asignar quien debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito general de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

```
/**
 * Creates a new AreaDeTrabajo entity.
 *
 * @Route("/", name="AreaDeTrabajo_create")
 * @Method("POST")
 * @Template("AppBundle:Respuestas:new.html.twig")
 */
public function createAction(Request $request)
{
    $entity = new Respuestas();

    $form = $this->createCreateForm($entity);
    $form->handleRequest($request);

    if ($form->isValid()) {
        $em = $this->getDoctrine()->getManager();
        $em->persist($entity);
        $em->flush();
    }
}
```

Figura 3: Evidencia del patrón creador en la clase AreaDeTrabajoController.php

- **Alta Cohesión:** El objetivo de este patrón es asignar responsabilidades de tal forma que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

```

<?php

namespace AppBundle\Controller;

use AppBundle\Entity\Evaluaciones;
use AppBundle\Entity\Respuestas;
use AppBundle\Entity\Subgrupos;
use AppBundle\Entity\Tareas;
use AppBundle\Form\EvaluacionesType;
use AppBundle\Form\RespuestasType;
use AppBundle\Form\TareasType;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Request;

class DefaultController extends Controller
{
    /**
     * @Route("/", name="homepage")
     */
    public function inicioAction(Request $request)
    {

```

Figura 4: Evidencia del patrón alta cohesión en la clase DefaultController.php

- **Bajo Acoplamiento:** El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios. Este patrón especifica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización.

```

/**
 * AreaDeTrabajo controller.
 *
 * @Route("/AreaDeTrabajo")
 */

class AreaDeTrabajoController extends Controller
{
    /**
     * @Route("/mostrarareadetrabajo", name="mostrarareadetrabajo")
     */
    public function indexAction() {

        $em = $this->getDoctrine()->getManager();

```

Figura 5: Evidencia del patrón bajo acoplamiento en la clase AreaDeTrabajoController.php

## **Conclusiones parciales**

El desarrollo de este capítulo permitió, un mejor entendimiento del sistema propuesto y sus características. Obteniéndose el Modelo de Dominio donde se especifica de una forma más detallada los mecanismos que se manejan durante una clase de laboratorio. Llevando a cabo una minuciosa descripción de los casos de uso y de los trabajadores del sistema, se logró un mayor entendimiento de la solución propuesta para el desarrollo de la herramienta web.

## Capítulo 3: Análisis y diseño del sistema

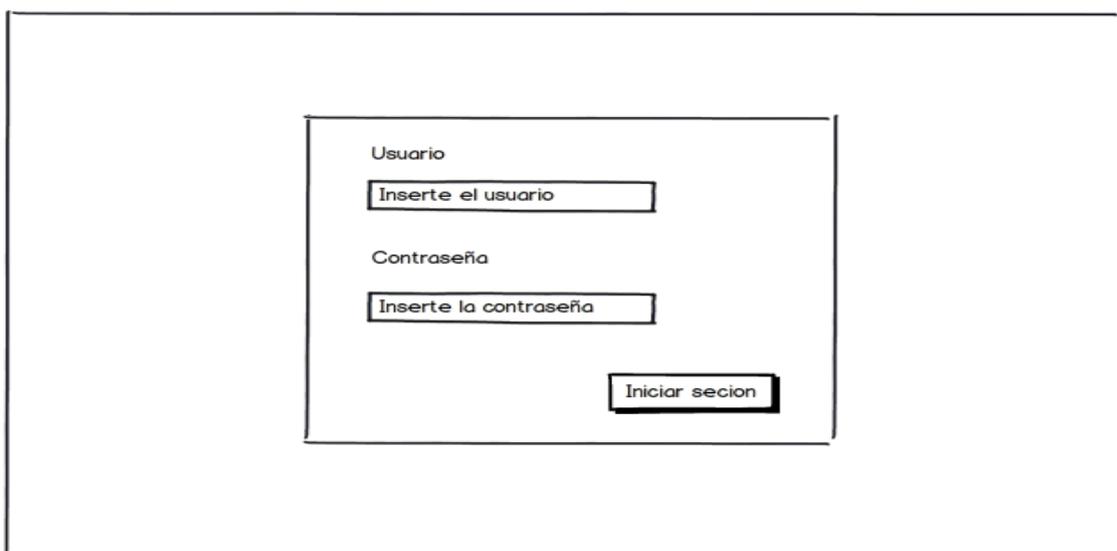
Este capítulo presenta el diseño, la implementación y la prueba del sistema realizado, así como el diagrama de despliegue, describiéndose en la construcción de la aplicación.

### 3.1. Prototipos no funcionales

#### ¿Qué es un Prototipo?

Un Prototipo es un objeto que sirve como referencia para futuros modelos en una misma cadena de producción. Un Prototipo es el primer dispositivo que se fabrica y del que se toman las ideas más relevantes para la construcción de otros diseños y representa todas las ideas en cuanto a diseño, soporte y tecnología que se les puedan ocurrir a sus creadores. Por lo general un prototipo no sale a la venta a menos que sea menos que sea un terminal orientado para que otros desarrolladores de tecnología trabajen con él para insertar nuevas funciones o especificaciones a este para que funcione de una manera más eficiente. (26)

Para la implementación de la herramienta de apoyo a la docencia el equipo de desarrollo se basó en varios prototipos no funcionales, los mismos se visualizarán a continuación.

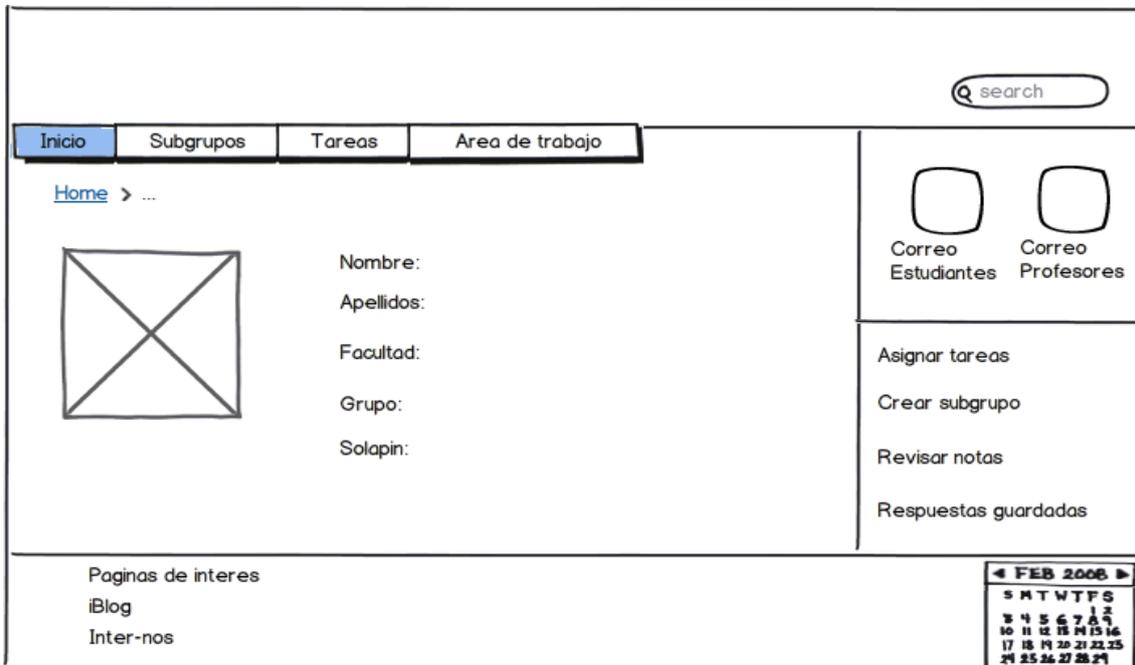


El diagrama muestra un formulario de autenticación dentro de un recuadro rectangular. El formulario contiene tres elementos de entrada:

- Un campo de texto etiquetado "Usuario" con el placeholder "Inserte el usuario".
- Un campo de texto etiquetado "Contraseña" con el placeholder "Inserte la contraseña".
- Un botón etiquetado "Iniciar seccion" (con un error ortográfico).

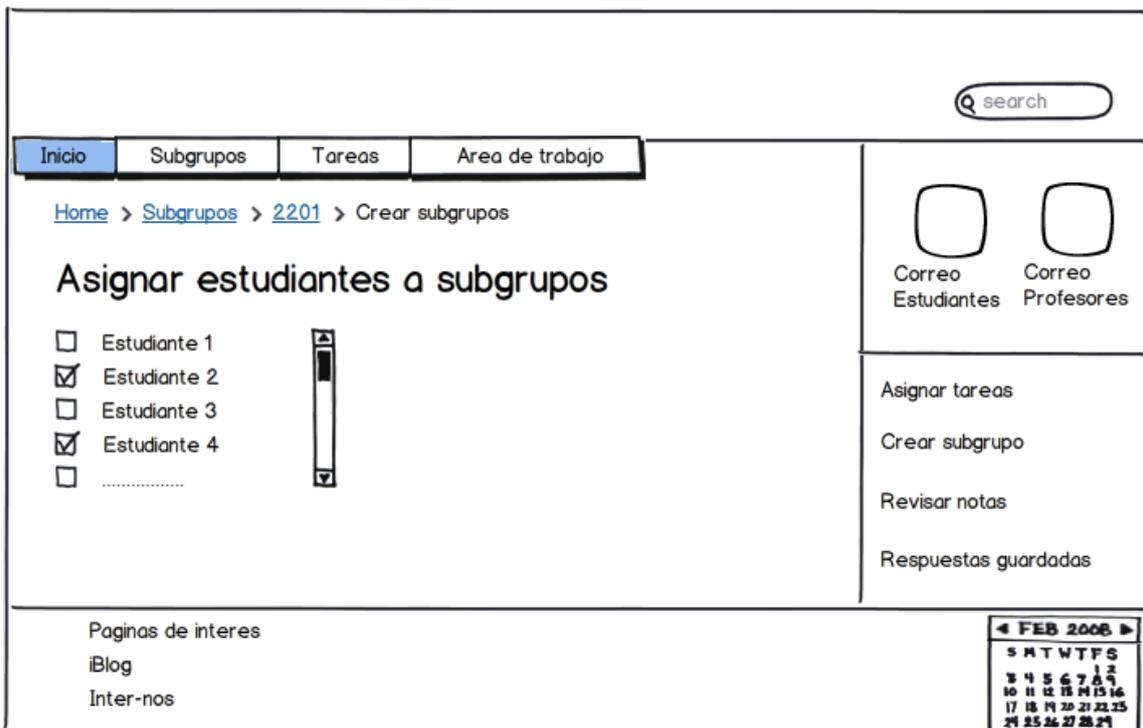
Figura 6: Prototipo no funcional Autenticar.

En este prototipo los usuarios del sistema deben introducir su usuario y contraseña del dominio UCI, el mismo toma los datos del servidor LDAP (Protocolo Ligerero de Acceso a Datos) para verificarlos, si los datos introducidos no son correctos el usuario no tendrá acceso a la herramienta.



**Figura 7: Prototipo no funcional: página de inicio**

Este prototipo muestra los datos de la persona que se autentica en el sistema, los datos son obtenidos del consumo de servicios web, en este caso con el wsdl <http://identificacion.uci.cu/servicios/v5/servicios.php?wsdl> y <https://autenticacion2.uci.cu/v6/PasarelaAutenticacionWS.wsdl>.



**Figura 8: Prototipo no funcional: Crear Subgrupos.**

Los prototipos no funcionales referentes a edición y eliminación de subgrupos tienen el funcionamiento básicamente parecido al de crear subgrupo. El profesor debe seleccionar de una lista de estudiantes solo a los que desea agregar a un nuevo subgrupo y posteriormente agregar el nombre del mismo. El sistema almacena el nombre del estudiante y el subgrupo al que pertenece una vez realizada la acción.

### **3.2. Estándar de codificación**

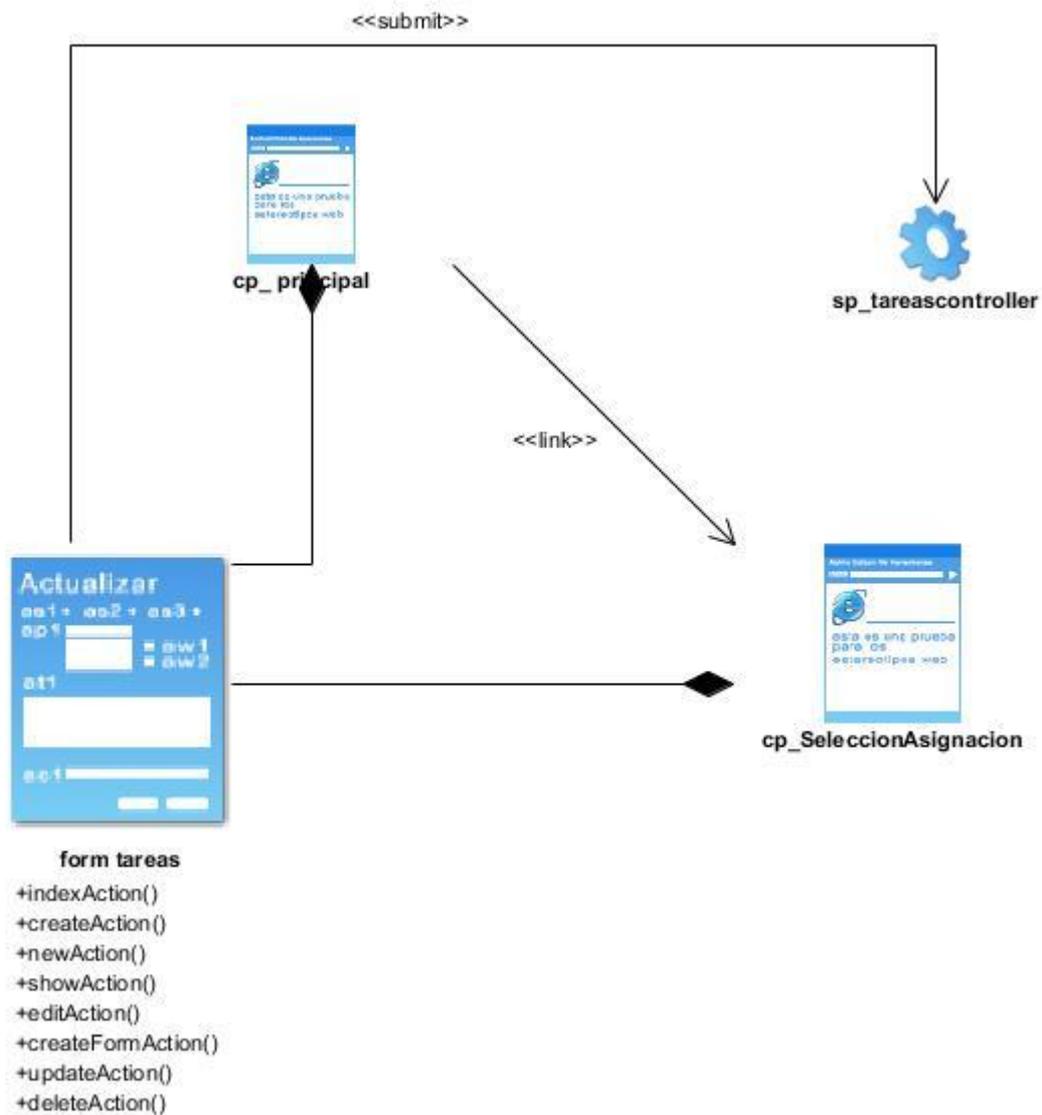
Actualmente se hallan estándares de codificación para la mayoría de los lenguajes de programación existentes. El uso de ellos partiendo de las convenciones definidas permite una mejor comunicación entre los programadores creando condiciones para la reusabilidad y mantenimiento de los sistemas. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para aplicaciones desarrolladas en PHP (PHP Coding Standard), que mayormente está basada en el estándar de código para aplicaciones en C++ (C++ Coding Standard) [COD03]. Las etiquetas de apertura y cierre del lenguaje serán de la forma `<? php?>`, ya que siempre están disponible en cualquier configuración. Para nombrar las

variables se seguirá la regla de escribir los identificadores con letras minúsculas y en español, no se usará separador entre palabras, sino que las siguientes palabras se escribirán con letra inicial mayúscula, tratando de usar nombres sugerentes a la acción de la variable.

Todos los campos id van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo id\_noEscala. Los arreglos empezarán con el identificador array y las palabras no se separarán. Ejemplo: arrayBuscar.

### **3.3. Diagrama de clases de diseño**

Los diagramas de clases describen la estructura del sistema, muestra las relaciones entre las clases y sus atributos. Al crearse el diseño conceptual de la información que se manejará en el sistema, conjuntamente con los componentes encargados del funcionamiento es cuando se utilizan los diagramas de clases, o sea en el proceso de diseño y análisis del sistema. A continuación, se muestra el Diagrama de Clases de Diseño Asignar tareas, los demás diagramas se muestran en el anexo 1.



**Figura 9: Diagrama de Clases de Diseño Asignar Tareas.**

### 3.4. Diseño de la BD

#### Modelo de datos

Es un conjunto de conceptos que pueden servir para describir la estructura de una Base de Datos; tipo de datos, las relaciones y que deben cumplirse para esos datos. Por lo general los modelos de datos contienen además un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la Base de Datos. (27)

A continuación, se muestra como queda la estructura de la Base de Datos de la herramienta de apoyo a la docencia, la misma permite la representación lógica

de los datos que se procesarán. Este modelo muestra las entidades de datos, los atributos asociados a cada una de ellas y las relaciones entre las mismas.

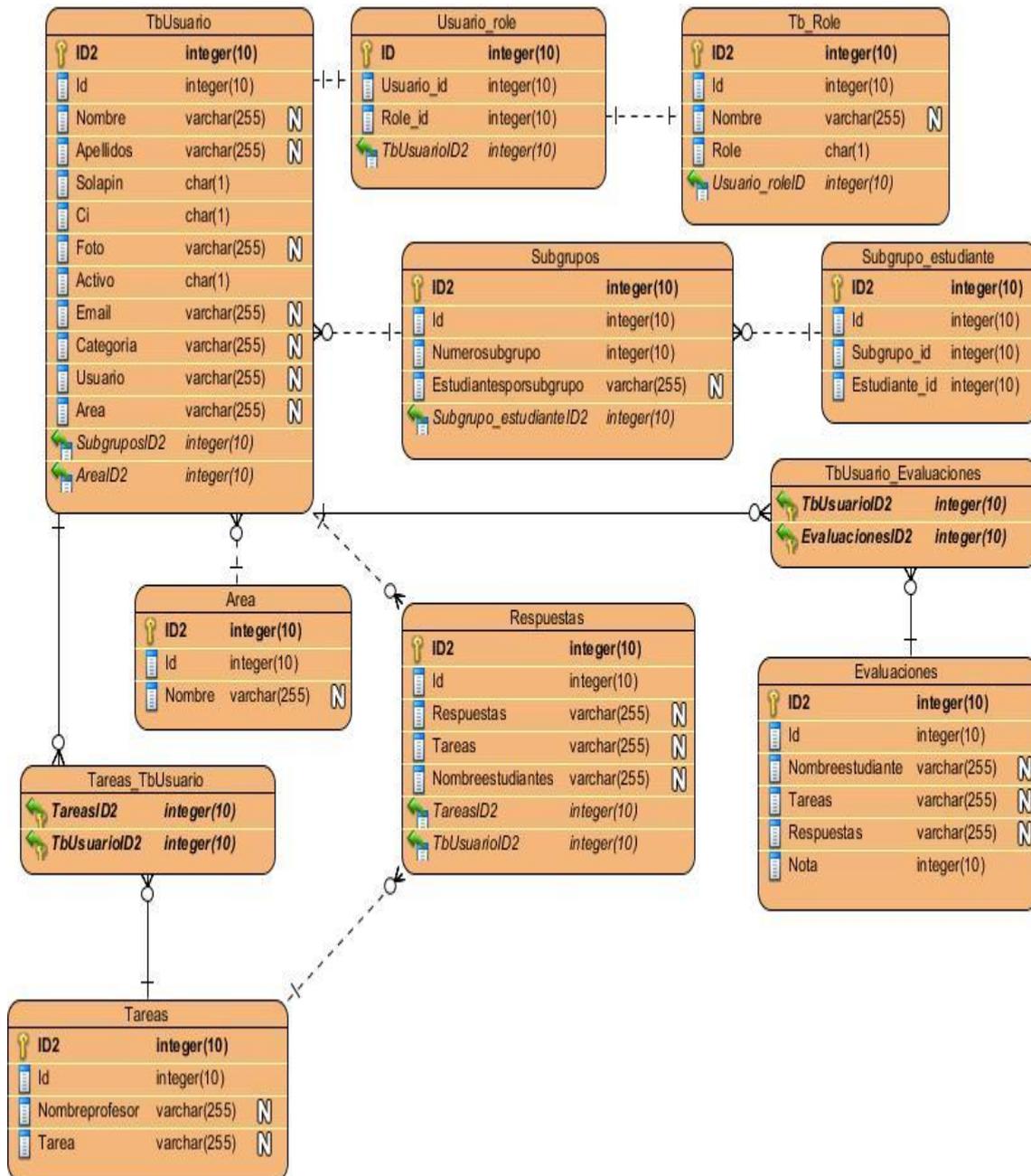


Figura 10: Modelo de datos de la Herramienta de Apoyo a la Docencia

### 3.5. Modelo de implementación

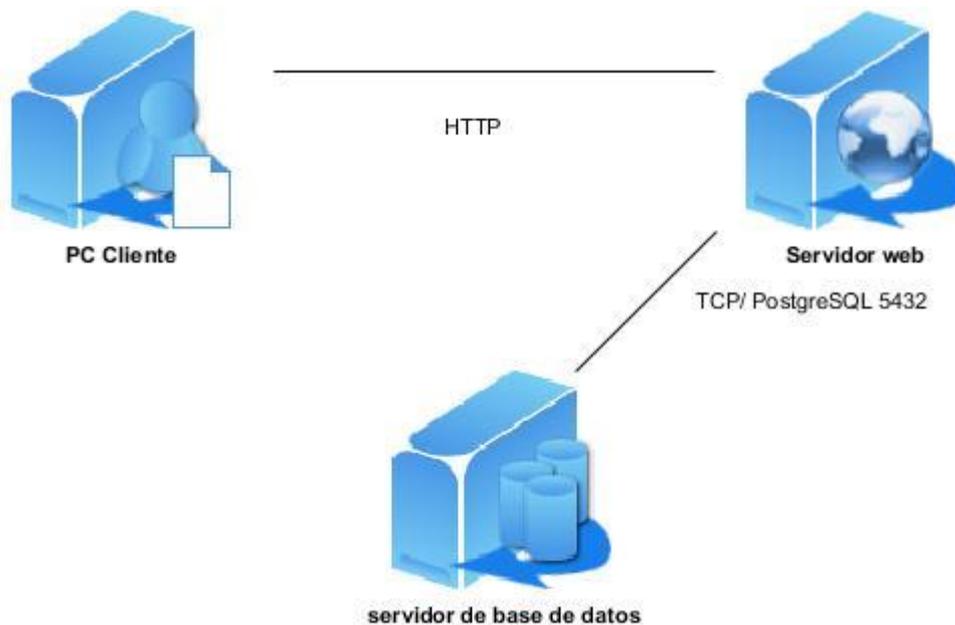
Los diagramas de implementación ofrecen una ilustración de la arquitectura física del hardware, del software y de los artefactos del sistema. Los diagramas

de implementación pueden entenderse como lo contrario de los casos de uso, porque ilustran la forma física del sistema, en lugar de representar conceptualmente los usuarios y dispositivos que interactúan con el sistema. (28)

El modelo de despliegue de la Herramienta de apoyo a la docencia está compuesto por el diagrama de despliegue y el diagrama de componente como se muestra a continuación.

### **3.5.1. Modelo de Despliegue**

Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico. Un diagrama de despliegue consta de la interconexión de nodos a partir de relaciones de asociación (Figura 1.). En este contexto, las relaciones representan enlaces físicos (normalmente bidireccionales), como es el caso de una conexión directa mediante cables o indirecta por vía satélite. (29). A continuación, se realiza una representación gráfica (Ver Figura 6) de los nodos físicos en los que estará desplegado el sistema propuesto y la comunicación entre ellos.



**Figura 11: Diagrama de Despliegue.**

### 3.5.2. Descripción de los nodos

- **Servidor Web:** Aquí es donde se ejecutarán todas funcionalidades del servidor Web, entre ellas está el procesamiento de datos y el control de flujo.
- **Servidor de Bases de Datos:** es en este nodo donde se ejecutará el servidor PostgreSQL, además de contar con la Base de Datos de la herramienta. También en este servidor estará desplegada la Base de Datos para la realización de consultas que darán respuesta a las tareas orientadas.
- **PC Cliente:** Es el equipo donde exista un navegador para acceder a la herramienta.

### 3.6. Pruebas al sistema

Las pruebas es un proceso que acompaña al desarrollo de cualquier software para medir la calidad del producto, considerándose un proceso crítico diseñado para detectar errores no detectados con antelación sin demostrar la ausencia de los mismos, pero si los defectos del software. Cuando estás desarrollando un aplicativo web que podría estar bajo altos niveles de tráfico, luego de

configurarla y tener todo listo para hacerla disponible al mundo, es una buena práctica ponerla bajo pruebas de estrés y pruebas de carga y concurrencia para determinar su capacidad y velocidad de respuesta. Esto permite saber los límites bajo los cuales la aplicación se comportará de manera apropiada y te permitirá saber cuándo implementar diferentes estrategias de optimización, como a niveles de configuración del servidor o a veces implementar soluciones de balanceo de carga.

### Pruebas de caja negra

Las pruebas de caja negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos únicamente en los requerimientos de software y especificaciones funcionales. (30)

A continuación, se muestra el Diseño de Caso de Prueba del CU Gestionar Evaluación, el resto de los Diseños de Casos de Prueba pueden encontrarse en el Anexo 2 del presente trabajo.

Nombre de la sección	Escenarios de la descripción	Descripción de la sección	Flujo central	Resultados esperados	Resultados obtenidos
SC1: Gestionar evaluación	EC1.1: Insertar evaluación	El profesor selecciona la opción Evaluar Tarea. La herramienta muestra un página con un formulario para	Evaluaciones / Evaluar tarea	El sistema debe guardar en la BD los campos llenados por el	El sistema guarda en la BD los campos llenados por el profesor.

		insertar la nota del estudiante, el nombre y el número de la tarea, después debe dar clic en el botón "evaluar tarea".		profesor.	
	EC1.2: Campos vacíos	El profesor selecciona la opción Evaluar Tarea. La herramienta muestra un página con un formulario para insertar la nota del estudiante, el nombre y el número de la tarea, pero deja uno de estos campos vacíos y después da clic en el botón "evaluar tarea".	Evaluaciones / Evaluar tarea	El sistema debe mostrar un mensaje indicando que debe llenar el campo	El sistema muestra un mensaje indicando que debe llenar el campo

**Tabla 17: Diseño de Caso de Prueba del CU Gestionar evaluación.**

### **¿Qué es una prueba de carga?**

Una prueba de carga se define como el proceso que se le impone a un sistema basado en una cantidad predefinida de peticiones o procedimientos con la finalidad de determinar su comportamiento esperado para dicha situación.

## ¿Cuál es la diferencia con una prueba de estrés?

Mucha gente los define como sinónimos y no hacen ninguna distinción; sin embargo, el concepto de pruebas de estrés refiere al caso particular de determinar el comportamiento del sistema bajo un nivel de exigencia mayor al que es capaz de manejar, de ahí el origen de su nombre.

Bajo este tipo de pruebas el sistema se satura y los escenarios más comunes suelen ser:

- El sistema va respondiendo lo que puede y desecha algunas peticiones.
- El sistema responde todas las peticiones, pero con un retraso considerable.
- El sistema colapsa y queda fuera de línea. (31)

Normalmente este tipo de pruebas suelen ser usados en servidores webs, pero también puede usar el mismo principio para pruebas de software (Desktop) o cualquier sistema de información.

Todos estos servidores web en su configuración por defecto trabajan excelentes pero los sitios web albergados pueden convertirse en una fuente de carga estresante para el servidor y forzando a colapsar los servicios. Existen diversas formas tanto manuales, y herramientas automatizadas para realizar este tipo de pruebas, algunos automatizados son Jmeter. (32)

Las pruebas se realizaron en una computadora con las siguientes características:

Memoria RAM: 4.00GB (3.89 GB utilizable).

Procesador: Intel(R) Pentium(R) CPU N3540 @ 2.16GHz.

Sistema operativo: Windows 10, sistema operativo de 64bits, procesador x64.

El resultado arrojado por las pruebas realizadas en el Jmeter son las siguientes:

Para al interfaz de Insertar respuestas:

Con 30 usuarios:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/CorregirNue...	30	1250	1286	1545	789	1555	100.00%	19.2/sec	45.2
Total	30	1250	1286	1545	789	1555	100.00%	19.2/sec	45.2

**Figura 12:Prueba de carga y estrez, interfaz insertar respuesta con 30 usuarios.**

Con 60 usuarios:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/CorregirNue...	60	2335	2315	2734	1862	2815	100.00%	20.6/sec	48.6
Total	60	2335	2315	2734	1862	2815	100.00%	20.6/sec	48.6

**Figura 13:Prueba de carga y estrez, interfaz insertar respuesta con 60 usuarios.**

Con 100 usuarios:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/CorregirNue...	100	4466	4475	4902	3913	5071	100.00%	19.7/sec	46.5
Total	100	4466	4475	4902	3913	5071	100.00%	19.7/sec	46.5

**Figura 14:Prueba de carga y estrez, interfaz insertar respuesta con 100 usuarios.**

Con 150 usuarios:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/CorregirNue...	150	9923	9909	10381	9249	10729	100.00%	13.9/sec	32.9
Total	150	9923	9909	10381	9249	10729	100.00%	13.9/sec	32.9

**Figura 15:Prueba de carga y estrez, interfaz insertar respuesta con 150 usuarios.**

Por lo que se puede concluir que la herramienta tendrá un buen funcionamiento hasta un total de 100 usuarios conectados en el mismo momento.

### Conclusiones parciales

En este capítulo se logra el diseño, análisis e implementación y prueba del sistema realizado, se muestra la estructura de la Base de Datos, el modelo de despliegue de la Herramienta de apoyo a la docencia compuesto por el diagrama de despliegue y el diagrama de componente. Se describe la construcción de la aplicación.

## Conclusiones

La fundamentación teórica realizada evidencia la repercusión que ha tenido a nivel mundial como nacional, el desarrollo y evolución de las herramientas web de ayuda a la docencia específicamente en la asignatura de Sistema de Bases de Datos, lo cual exige que los docentes adopten una nueva posición en el proceso de enseñanza aprendizaje, a partir de la utilización de estas.

Se seleccionaron las tecnologías y la metodología de desarrollo de software. Atendiendo a las necesidades de la herramienta se fueron utilizando y poniendo en práctica los conocimientos adquiridos de su estudio.

Para satisfacer las necesidades del cliente se realizó el análisis y diseño de la aplicación, basada. Con lo que se logró un enfoque centralizado en la arquitectura, requisitos funcionales y no funcionales del sistema, casos de uso, escenarios y patrones de diseño.

Se elaboró una herramienta web que permite la creación y configuración de perfiles de usuarios, para interactuar con el SGBD PostgreSQL, la cual facilita el desempeño de las clases prácticas de laboratorio en las asignaturas de BD I y II en la UCI.

Las pruebas de software permitieron comprobar las funcionalidades de la herramienta, utilizando las pruebas de caja negra y las pruebas de carga y estrés. De esta forma se obtuvo una solución que se ajusta a los requisitos definidos.

## Recomendaciones

- Desarrollar módulos de integración con otras asignaturas de la carrera.
- Realización del despliegue de la herramienta.

## Referencias bibliográficas

1. Jiménez Ramírez, Claudia . [En línea] [Citado el: 19 de junio de 2017.] <http://wikifoundryattachments.com/YIJNuvlOmar+SwgXvN7F+A==448853>.
2. <http://phppgadmin.blogspot.com>. [En línea] [Citado el: 19 de junio de 2017.] <http://phppgadmin.blogspot.com/2015/07/que-es-phppgadmin.html>.
3. <http://www.macapps.es>. [En línea] [Citado el: 19 de junio de 2017.] <http://www.macapps.es/index.php/31-aplicaciones-tecnicas/6369-sqlpro-for-postgres>.
4. <https://www.argar.cat>. [En línea] [Citado el: 18 de junio de 2017.] <https://www.argar.cat/es/adminier-administrador-de-bases-de-datos/>.
5. <https://www2.navicat.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://www2.navicat.com/es/products/navicat-for-postgresql>.
6. <https://docs.phpmyadmin.net>. [En línea] [Citado el: 20 de junio de 2017.] <https://docs.phpmyadmin.net/es/latest/intro.html>.
7. <https://msdn.microsoft.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://msdn.microsoft.com/es-es/library/bb545450.aspx>.
8. [www.pgadmin.org](http://www.pgadmin.org). <https://www.pgadmin.org>. [En línea] [Citado el: 19 de junio de 2017.] <https://www.pgadmin.org/docs/pgadmin3/1.22/introduction.html>.
9. <http://www.software.com.ar>. [En línea] [Citado el: 17 de junio de 2017.] <http://www.software.com.ar/p/visual-paradigm-para-uml>.
10. <http://san-salvador-ss.all.biz>. [En línea] [Citado el: 19 de junio de 2017.] <http://san-salvador-ss.all.biz/business-process-visual-architect-g4268>.
11. <https://www.lucidchart.com>. [En línea] [Citado el: 17 de junio de 2017.] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>.
12. Soto, Daniel. <http://nextech.pe>. [En línea] 16 de septiembre de 2016. [Citado el: 19 de junio de 2017.] <http://nextech.pe/que-es-bpmn-y-para-que-sirve/>.
13. <http://www.4rsoluciones.com>. [En línea] [Citado el: 16 de mayo de 2017.] <http://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>.
14. <http://www.4rsoluciones.com>. [En línea] [Citado el: 16 de mayo de 2017.] <http://www.4rsoluciones.com/blog/que-es-la-programacion-orientada-a-objetos-2/>.

15. <http://docs.python.org.ar>. [En línea] <http://docs.python.org.ar/tutorial/3/real-index.html>.
16. <https://msdn.microsoft.com>. [En línea] <https://msdn.microsoft.com/es-es/library/4w3ex9c2%28v=vs.100%29.aspx>.
17. Liarte, Julio. <http://escueladeinformaticos.com>. [En línea] 8 de diciembre de 2015. [Citado el: 19 de junio de 2017.] <http://escueladeinformaticos.com/que-es-php-y-para-que-nos-sirve/>.
18. Potencier, Fabien y Zaninotto, François . *Symfony la guía definitiva*.
19. <https://framework.zend.com>. [En línea] [Citado el: 19 de junio de 2017.] <https://framework.zend.com/manual/2.4/en/ref/overview.html>.
20. <http://ingenieriadesoftware.mex.tl>. [En línea] [Citado el: 20 de junio de 2017.] <http://ingenieriadesoftware.mex.tl/images/18149/METODOLOGIAS%20AGILES.pdf>.
21. Bustamante , Dayana y Rodríguez , Jean C. <http://blogs.unellez.edu.ve>. [En línea] [Citado el: 20 de junio de 2017.] <http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>.
22. *Metodologías ágiles*. Gutierrez, Joaquin. Medellín : s.n., 2008, Vol. 5.
23. LinkedIn Corporation. <https://es.slideshare.net>. [En línea] [Citado el: 20 de junio de 2017.] <https://es.slideshare.net/BiingeSof/metodologias-rup>.
24. Larman, Craig y Hall, Pentice. *UML y Patrones*. 2003.
25. Alvarez, Miguel Angel . <https://desarrolloweb.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://desarrolloweb.com/articulos/que-es-mvc.html>.
26. <http://conceptodefinicion.de>. [En línea] [Citado el: 19 de junio de 2017.] <http://conceptodefinicion.de/prototipo/>.
27. Elmasir y Navathe. *Sistemas de Bases de Datos*.
28. <https://www.altova.com>. [En línea] <https://www.altova.com/es/umodel/uml-deployment-diagrams.html>.
29. *Análisis y diseño de sistemas*. Cossio Taboada, Humberto Miguel y Apaza Floren, Ovidio Huanca. La Paz : s.n., 2009.
30. <http://www.pmoinformatica.com>. [En línea] 20 de febrero de 2017. [Citado el: 18 de junio de 2017.] <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>.

31. Wiesel, Jonathan . *http://codehero.co*. [En línea] [Citado el: 16 de mayo de 2017.] <http://codehero.co/como-hacer-pruebas-de-carga-servidores-web/>.
32. Orozco, Antonio. *http://antoniojorz.blogspot.com*. [En línea] [Citado el: 16 de mayo de 2017.] <http://antoniojorz.blogspot.com/2015/05/prueba-de-carga-y-estres-jmeter.html>.

## Bibliografía Consultada

1. García Serrano, M. (2011). Estrategia de superación en informática para contribuir al uso de la educación a distancia en los docentes de la Universidad de Ciencias Pedagógicas “Rafael María de Medie”. Tesis en opción al título de Máster en Educación, Universidad de Ciencias Pedagógicas “Rafael María de Medie”, Pinar del Río.
2. Genera Navarro, E. J. (2005). Temas de Informática Básica. La Habana: Editorial Pueblo y Educación.
3. Hernández Sampieri, Roberto / Fernández Collado, Carlos / Baptista Lucio, Pilar. Metodología de la investigación, 6ta edición.
4. Jiménez Ramírez, Claudia . [En línea] [Citado el: 19 de junio de 2017.] <http://wikifoundryattachments.com/YIJNuvlOmar+SwgXvN7F+A==448853>.
5. <http://phppgadmin.blogspot.com>. [En línea] [Citado el: 19 de junio de 2017.] <http://phppgadmin.blogspot.com/2015/07/que-es-phppgadmin.html>.
6. <http://www.macapps.es>. [En línea] [Citado el: 19 de junio de 2017.] <http://www.macapps.es/index.php/31-aplicaciones-tecnicas/6369-sqlpro-for-postgres>.
7. <https://www.argar.cat>. [En línea] [Citado el: 18 de junio de 2017.] <https://www.argar.cat/es/adminier-administrador-de-bases-de-datos/>.
8. <https://www2.navicat.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://www2.navicat.com/es/products/navicat-for-postgresql>.
9. <https://docs.phpmyadmin.net>. [En línea] [Citado el: 20 de junio de 2017.] <https://docs.phpmyadmin.net/es/latest/intro.html>.
10. <https://msdn.microsoft.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://msdn.microsoft.com/es-es/library/bb545450.aspx>.
11. [www.pgadmin.org](http://www.pgadmin.org). <https://www.pgadmin.org>. [En línea] [Citado el: 19 de junio de 2017.] <https://www.pgadmin.org/docs/pgadmin3/1.22/introduction.html>.
12. <http://www.software.com.ar>. [En línea] [Citado el: 17 de junio de 2017.] <http://www.software.com.ar/p/visual-paradigm-para-uml>.
13. <http://san-salvador-ss.all.biz>. [En línea] [Citado el: 19 de junio de 2017.] <http://san-salvador-ss.all.biz/business-process-visual-architect-g4268>.

14. <https://www.lucidchart.com>. [En línea] [Citado el: 17 de junio de 2017.] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>.
15. Soto, Daniel. <http://nextech.pe>. [En línea] 16 de septiembre de 2016. [Citado el: 19 de junio de 2017.] <http://nextech.pe/que-es-bpmn-y-para-que-sirve/>.
16. <http://www.4rsoluciones.com>. [En línea] [Citado el: 16 de mayo de 2017.] <http://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>.
17. <http://www.4rsoluciones.com>. [En línea] [Citado el: 16 de mayo de 2017.] <http://www.4rsoluciones.com/blog/que-es-la-programacion-orientada-a-objetos-2/>.
18. <http://docs.python.org.ar>. [En línea] <http://docs.python.org.ar/tutorial/3/real-index.html>.
19. <https://msdn.microsoft.com>. [En línea] <https://msdn.microsoft.com/es-es/library/4w3ex9c2%28v=vs.100%29.aspx>.
20. Liarte, Julio. <http://escueladeinformaticos.com>. [En línea] 8 de diciembre de 2015. [Citado el: 19 de junio de 2017.] <http://escueladeinformaticos.com/que-es-php-y-para-que-nos-sirve/>.
21. Potencier, Fabien y Zaninotto, François . *Symfony la guía definitiva*.
22. <https://framework.zend.com>. [En línea] [Citado el: 19 de junio de 2017.] <https://framework.zend.com/manual/2.4/en/ref/overview.html>.
23. <http://ingenieriadesoftware.mex.tl>. [En línea] [Citado el: 20 de junio de 2017.] <http://ingenieriadesoftware.mex.tl/images/18149/METODOLOGIAS%20AGILES.pdf>.
24. Bustamante , Dayana y Rodríguez , Jean C. <http://blogs.unellez.edu.ve>. [En línea] [Citado el: 20 de junio de 2017.] <http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>.
25. *Metodologías ágiles*. Gutierrez, Joaquin. Medellín : s.n., 2008, Vol. 5.
26. LinkedIn Corporation. <https://es.slideshare.net>. [En línea] [Citado el: 20 de junio de 2017.] <https://es.slideshare.net/BiingeSof/metodologias-rup>.
27. Larman, Craig y Hall, Pentice. *UML y Patrones*. 2003.
28. Alvarez, Miguel Angel . <https://desarrolloweb.com>. [En línea] [Citado el: 20 de junio de 2017.] <https://desarrolloweb.com/articulos/que-es-mvc.html>.

29. <http://conceptodefinicion.de>. [En línea] [Citado el: 19 de junio de 2017.] <http://conceptodefinicion.de/prototipo/>.
30. Elmasir y Navathe. *Sistemas de Bases de Datos*.
31. <https://www.altova.com>. [En línea] <https://www.altova.com/es/umodel/uml-deployment-diagrams.html>.
32. *Análisis y diseño de sistemas*. Cossio Taboada, Humberto Miguel y Apaza Floren, Ovidio Huanca. La Paz : s.n., 2009.
33. <http://www.pmoinformatica.com>. [En línea] 20 de febrero de 2017. [Citado el: 18 de junio de 2017.] <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>.
34. Wiesel, Jonathan . <http://codehero.co>. [En línea] [Citado el: 16 de mayo de 2017.] <http://codehero.co/como-hacer-pruebas-de-carga-servidores-web/>.
35. Orozco, Antonio. <http://antoniojorz.blogspot.com>. [En línea] [Citado el: 16 de mayo de 2017.] <http://antoniojorz.blogspot.com/2015/05/prueba-de-carga-y-estres-jmeter.html>.

# Anexos

## Anexo1:

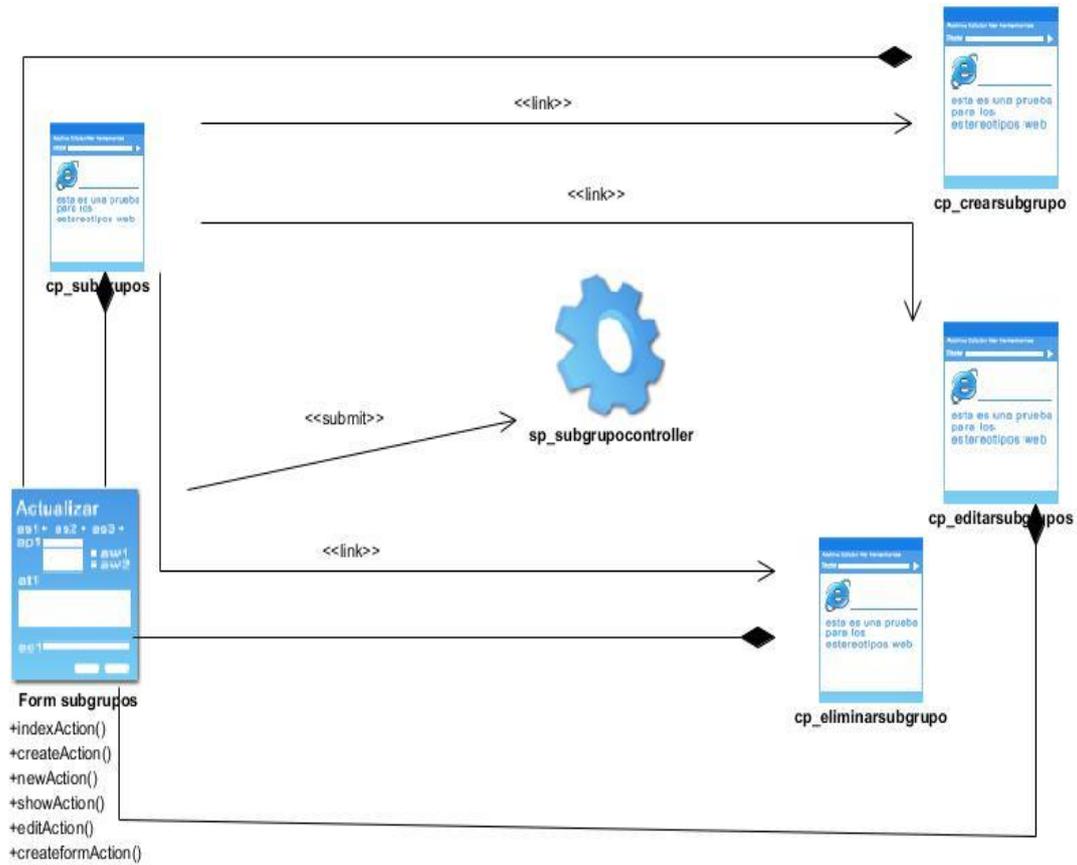


Figura 16: Diagrama de clases de diseño Gestionar Subgrupos

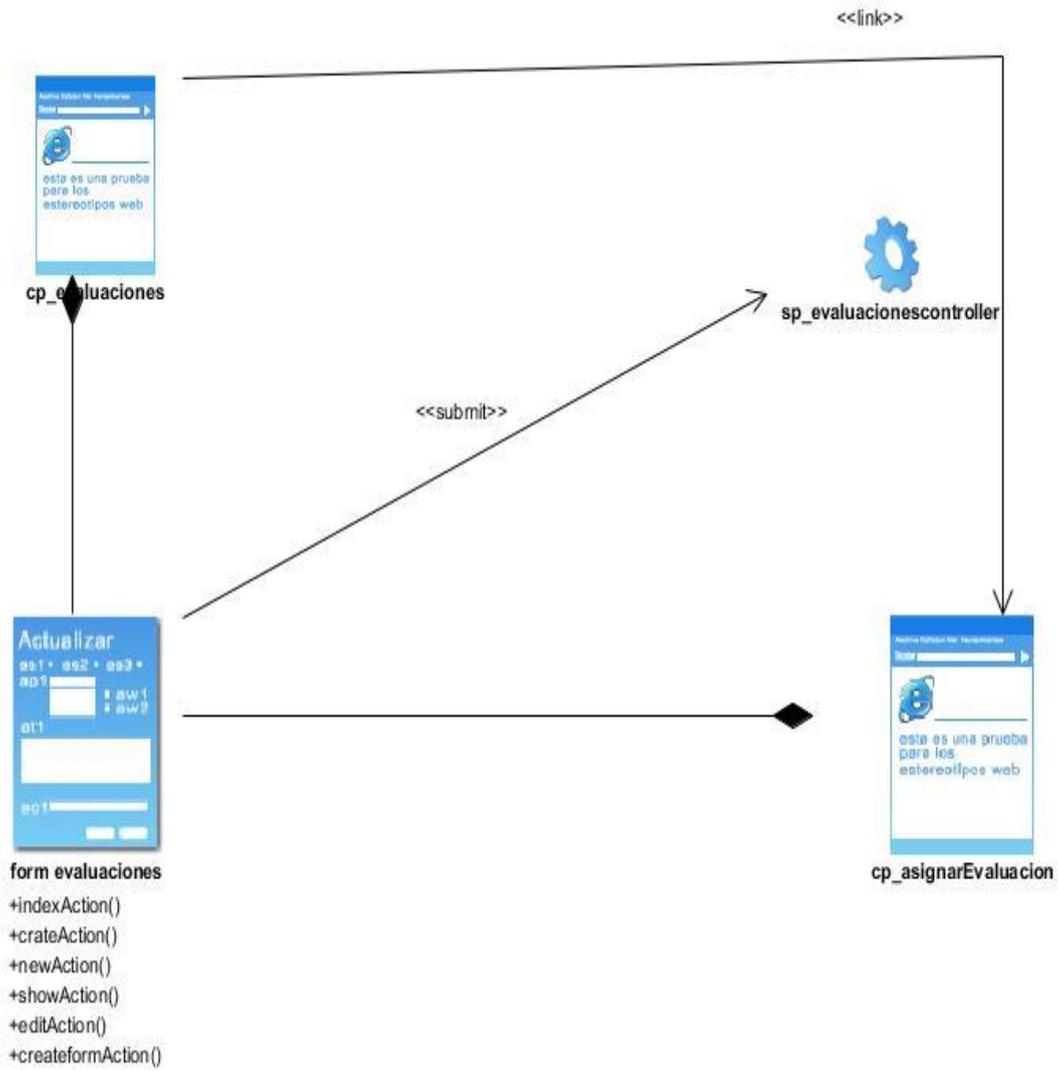


Figura 17: Diagrama de clases de diseño Gestionar Evaluación.

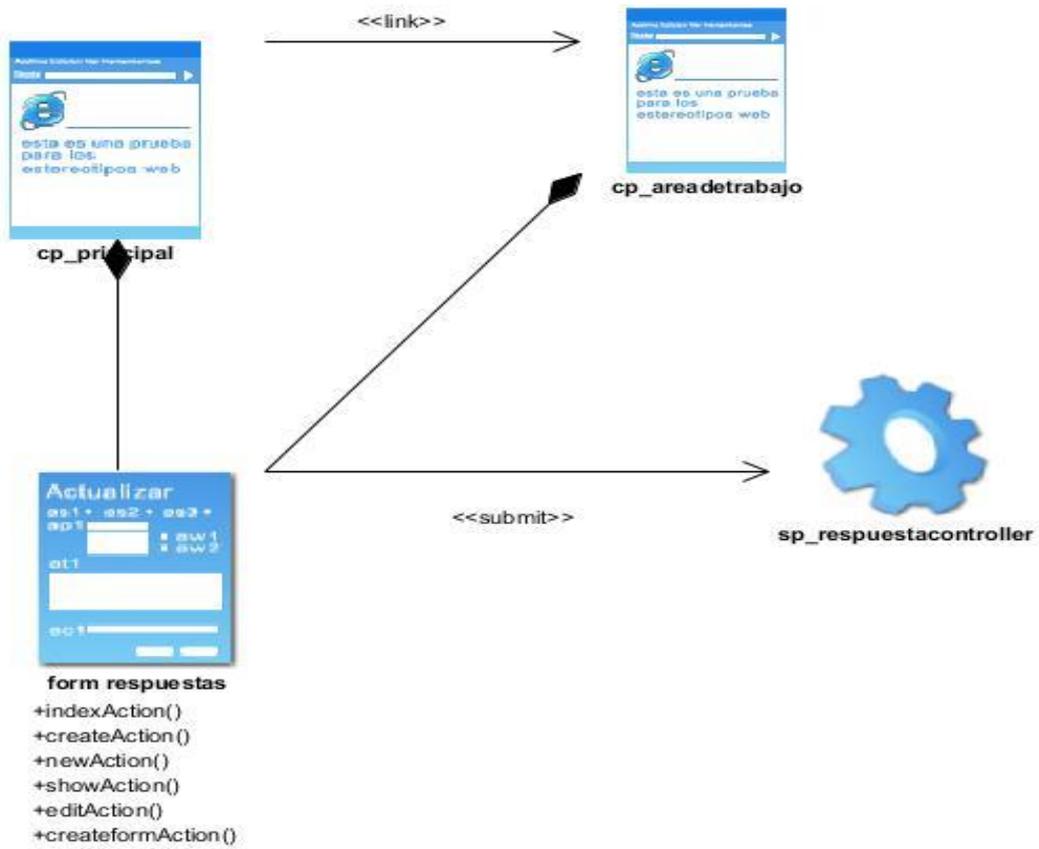


Figura 18: Diagrama de clases de diseño Insertar Respuestas.

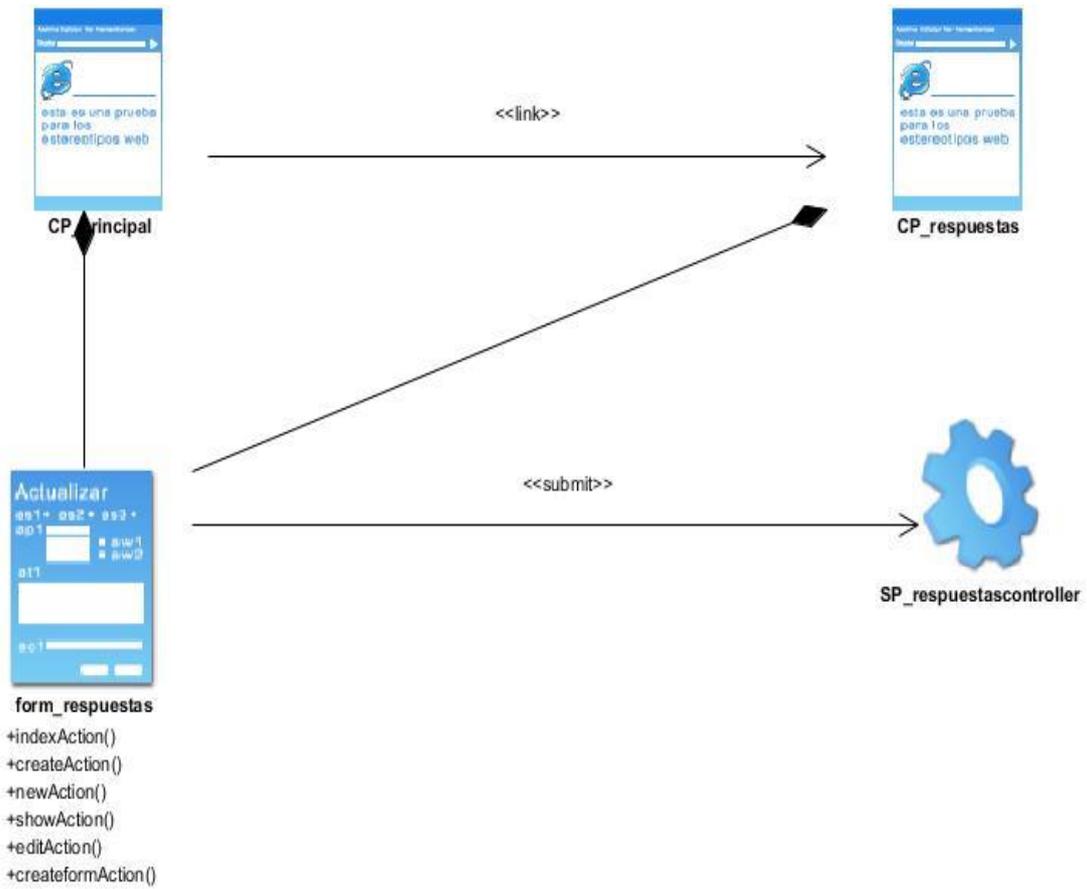


Figura 19: Diagrama de clases de diseño Revisar Respuestas.

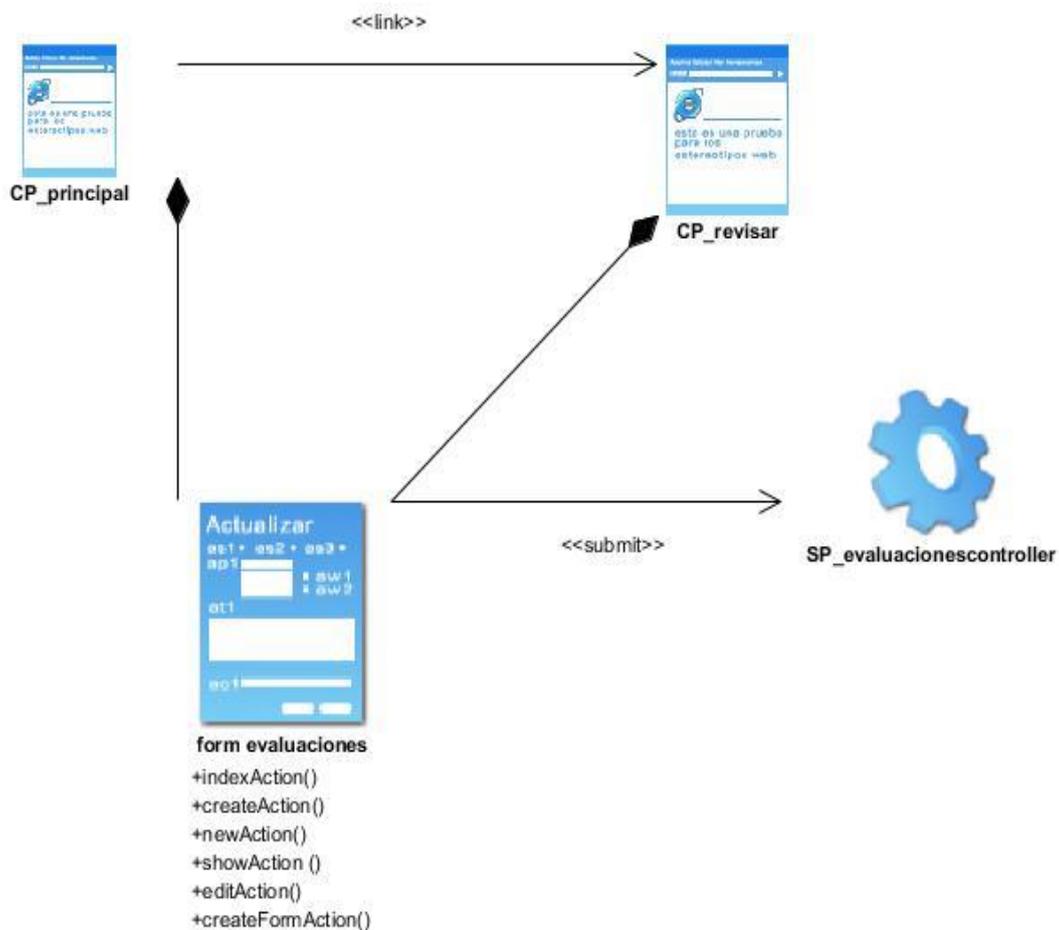


Figura 20: Diagrama de clases de diseño Revisar Tarea.

**Anexo2:**

Nombre de la sección	Escenarios de la descripción	Descripción de la sección	Flujo central	Resultados esperados	Resultados obtenidos
SC1: Autenticar Usuario	EC1.1: Insertar usuario y contraseña del dominio UCI	Una vez abierta la pestaña de la herramienta aparece la página de autenticar en la cual los usuarios deben llenar los campos con el	Login	La herramienta debe guarda en la BD la información referente a usuario autenticado si este lo hace por primera vez y accede a la página de inicio, si no es la primera vez	La herramienta guarda en la BD la información referente a cada usuario si es su primera vez y accede a la página de usuario y si no es la primera vez del usuario solo accede a

		usuario y la contraseña del dominio UCI y después da en el botón acceder		del usuario solo accede a la página de inicio	la página de inicio
	EC1.2: Campos vacíos	Una vez abierta la pestaña de la herramienta aparece la página de autenticar en la cual los usuarios deben llenar los campos con el usuario y la contraseña del dominio UCI y después da en el botón acceder pero los campos se dejan vacíos	Login	La herramienta debe mostrar un mensaje que diga "No se encontró el nombre de usuario"	La herramienta muestra un mensaje que dice "No se encontró el nombre de usuario"
	EC1.3: Campos incorrectos	Una vez abierta la pestaña de la herramienta aparece la página de autenticar en la cual los usuarios deben llenar los campos	Login	La herramienta debe mostrar un mensaje que diga "Credenciales no válidas"	La herramienta muestra un mensaje que dice "Credenciales no válidas"

		con el usuario y la contraseña del dominio UCI y da en el botón acceder pero uno de los campos es incorrecto			
--	--	--	--	--	--

**Tabla 18: Diseño de Caso de Prueba del CU Autenticar.**

<b>Nombre de la sección</b>	<b>Escenarios de la descripción</b>	<b>Descripción de la sección</b>	<b>Flujo central</b>	<b>Resultados esperados</b>	<b>Resultados obtenidos</b>
SC1:Asignar tarea	EC 1.1:Inserta la nueva tarea y el nombre del usuario que la orienta	Una vez en la opción asignar tarea. Aparecen los campos para ser llenados donde el usuario debe escribir la tarea a realizar y su nombre y pulsa asignar tarea	En la página principal pulsar la opción (Insertar una nueva tarea)	Guarda la nueva tarea y muestra una lista con todas las tareas orientadas	Guarda la nueva tarea y muestra una lista con todas las tareas orientadas
	EC1.2: cuando los campos están vacíos	Una vez en la opción asignar tarea. Aparecen	En la página principal pulsar la opción	Se muestra una mensaje "rellene estos	Se muestra una mensaje "rellene estos campos"

		los campos para ser llenados donde el usuario debe escribir la tarea a realizar y su nombre y pulsa asignar tarea	(Insertar una nueva tarea)	campos”	
--	--	---	----------------------------	---------	--

**Tabla 19: diseño de caso de prueba del CU asignar tareas**

<b>Nombre de la sección</b>	<b>Escenarios de la descripción</b>	<b>Descripción de la sección</b>	<b>Flujo central</b>	<b>Resultados esperados</b>	<b>Resultados obtenidos</b>
SC1: revisar tareas	EC1.1: tareas orientadas	Una vez seleccionada la opción tareas orientadas en la pestaña evaluación	Página principal /evaluaciones /tareas orientadas	Se muestran todas las tareas que fueron orientadas durante el curso	Se muestran todas las tareas que fueron orientadas durante el curso

**Tabla 20: diseño de caso de prueba del CU revisar tareas**

<b>Nombre de la sección</b>	<b>Escenarios de la descripción</b>	<b>Descripción de la sección</b>	<b>Flujo central</b>	<b>Resultados esperados</b>	<b>Resultados obtenidos</b>
SC1: revisar evaluación	EC1.1: evaluaciones	Pulsar la opción evaluaciones	Página principal /evaluaciones	Muestra todas las evaluaciones que se le han hecho	Muestra todas las evaluaciones que se le han hecho

				durante el curso	durante el curso
--	--	--	--	------------------	------------------

**Tabla 21: diseño de caso de prueba del CU revisar evaluación**

<b>Nombre de la sección</b>	<b>Escenarios de la descripción</b>	<b>Descripción de la sección</b>	<b>Flujo central</b>	<b>Resultados esperados</b>	<b>Resultados obtenidos</b>
SC1: insertar respuesta	EC1.1: Área de trabajo, el usuario llena los campos vacíos con las respuestas	Una vez que se pulsa la opción área de trabajo se llenan los campos con las respuestas correctas y se pulsa el botón comprobar consulta	Página principal  /área de trabajo  /comprobar consulta	Guarda la consulta y muestra todas las respuestas anteriores	Guarda la consulta y muestra todas las respuestas anteriores
	EC1.2: Área de trabajo, el usuario llena los campos vacíos con respuesta incorrectas	Una vez que se pulsa la opción área de trabajo se llenan los campos con las respuestas incorrectas y pulsa el botón comprobar consulta	Página principal  /área de trabajo  /comprobar consulta	Comprueba la consulta y muestra los errores dice el tipo de error	Comprueba la consulta y muestra los errores dice el tipo de error
	EC1.3 Área de trabajo el usuario deja los campos en blanco	Una vez que se pulsa la opción área de trabajo y deja algún	Página principal  /área de trabajo  /comprobar	Muestra un mensaje "rellene los campos"	Muestra un mensaje "rellene los campos"

		campo vacío	consulta		
--	--	-------------	----------	--	--

**Tabla 22: Diseño de caso de prueba del CU insertar respuesta**