BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

SISTEMA EN WEB CREADOR DE EXÁMENES PARA PRESENTAR CERTIFICACIONES EN BASES DE DATOS RELACIONALES

Tesis presentada como requisito para obtener el título de Licenciatura en:

Ingeniería en Ciencias de la Computación

Presenta:

JOSÉ DANIEL CORTEZANO MORA

Asesor:

M.I. CARLOS ARMANDO RÍOS ACEVEDO

Noviembre 2015

DEDICATORIA

La presente tesis se la dedico a toda mi familia que con todo su cariño y amor son mi más grande motivación y con nada en el mundo podría pagar todo lo que me han dado. Ustedes son parte fundamental de todas mis vivencias y siempre han sido, son y sé que serán un apoyo incondicional.

A mi papá, gracias por el ejemplo de esfuerzo, constancia y superación que eres para nosotros, gracias por la comprensión, la paciencia y el apoyo que me has dado, porque si hoy puedo ver un logro alcanzado, es en gran parte gracias a ti. Te amo papá

A mi mamá, siempre has estado con nosotros en todo momento y no dejas de hacerlo, gracias por tu paciencia, por ser la guerrera que eres, por tus sacrificios, por todo tu apoyo y todo tu amor. Te amo mamá.

A mi hermano Arturo, que toda la vida ha sido un ángel en mi camino, un amigo y un ejemplo a seguir, la vida no la concibo sin tu compañía, tus cuidados y tu gran amor. A mi hermano Mauricio, que a pesar de nuestras dificultades, siempre guardaré las cosas buenas que he pasado a tu lado y yo sé que si la vida te puso en mi camino es para algo bueno. Los amo hermanos.

A mi tía Luchita, que aunque no pueda leer esto, siempre te estaré agradecido por todos los cuidados y el amor que me has dado. Eres parte importante de esto y espero algún día poder retribuirte algo de lo que has hecho por mí. Te amo Luchita.

A mi Yuri, mil gracias por acompañarme durante todo este arduo camino, por la confianza y el amor que me brindas que me hace sentir que siempre puedo contar contigo. Gracias por tu ayuda y apoyo en todas las decisiones que he tomado, por compartir conmigo tristezas y alegrías y por hacer de todas las situaciones siempre aventuras inolvidables. Te amo Yuri.

Esto es por y para ustedes, admiro su fortaleza y agradezco todo lo que han hecho de mí.

AGRADECIMIENTOS

A la vida por permitirme cumplir un objetivo más y por poner en mi camino a las personas indicadas.

A mi papá Arturo, a mi mamá Rosa, a mis hermanos Arturo y Mauricio, a mi tía y a la vez segunda mamá Luchita, a mi linda Yuri y a todas las personas que directa o indirectamente hicieron posible este trabajo.

De manera especial agradezco a mi Docente Tutor, M.I. Carlos Armando Ríos Acevedo, quien con su gran calidad de persona, profesionalismo y experiencia acompaño el desarrollo de este trabajo, desde sus inicios hasta el final, aportando siempre con su conocimiento, observaciones y consejos oportunos.

ÍNDICE

RESUMEN **1**

CAPÍTULO 1: INTRODUCCIÓN **2**

1.1. Justificación 2

1.1.1. Definición del Problema 3

1.1.2. Objetivos 4

1.1.3. Objetivo General 4

1.1.4. Objetivos Específicos 4

1.1.5. Plan del Proyecto 4

1.1.6. Metodología y procedimiento 4

1.1.7. Plan del Proyecto 7

CAPÍTULO 2: MARCO TEÓRICO **8**

2.1. Base de Datos 8

2.1.1. Modelo de datos Entidad-Relación(E-R) 9

2.1.1.1. Entidad 10

2.1.1.2. Relación (interrelación) 10

2.1.1.3. Cardinalidad 10

2.1.1.4. Atributo 11

2.1.2. El Modelo Relacional 11

2.1.2.1. Estructura General 12

2.1.2.2. Concepto de tabla. Dominios y atributos 14

2.1.2.3. Restricciones 16

2.1.2.4. Reglas Fundamentales. Claves 17

2.1.2.5. Regla de identificación única 18

2.1.2.6. Regla de integridad referencial 18

2.1.3. Sistema de Gestión de Bases de Datos (DBMS Database Management System) 19
2.2. Base de Datos 19
2.2.1. Servidores WEB 20
2.2.2. Apache, PHP y MySQL..... 20
2.2.2.1. Apache 21
2.2.2.2. PHP..... 21
2.2.2.3. MySQL 22
2.3. MVC: Modelo-Vista-Controlador (Model-View-Controller) 25
2.3.1. Aplicación del patrón MVC en PHP 26
2.3.1.1. Características 26
2.4. Framework 28
2.4.1. Frameworks PHP 30
2.4.1.1. Zend Framework 31
2.4.1.2. CodeIgniter 31
2.5. Lenguaje CSS (Cascading Style Sheets - Hojas de Estilo en Cascada) 33
2.5.1. Bootstrap 35

CAPÍTULO 3: ANÁLISIS Y DISEÑO 36

3.1. Escenario 37
3.2. Alcance.....37
3.3. Roles dentro del Proyecto 38
3.4. Presentación del Sistema 38
3.5. El Cliente 39
3.6. Lista de funciones a implementar (Product Backlog) 39
3.7. Sprint Planning Meeting y Complete Spring Backlog (Reunión de planeación por Sprint y Lista de Tareas por Sprint completo) 43
3.8. Scrum Meeting (Reuniones con el Scrum Master) 45
3.9. Sprint Review y Sprint Retrospective 46

3.10. Resultados 50

3.10.1. Descripción de los elementos del MVC (Modelo-Vista-Controlador) 52

CAPÍTULO 4: PRUEBAS DEL SISTEMA 73

4.1. Tipos de pruebas aplicadas al sistema 73

4.2. Acceso al sistema 74

4.2.1. Página de inicio..... 74

4.2.2. Página de login 76

4.2.3. Página de registro..... 77

4.3. Interfaces según el tipo de usuario.....78

4.3.1. Opciones disponibles para el Administrador 78

4.3.1.1. Opción "Listar Usuarios"..... 78

4.3.1.2. Opción "Mostrar Certificaciones" 79

4.3.1.3. Opción "Mostrar Exámenes" 81

4.3.1.4. Opción "Mostrar Preguntas" 83

4.3.1.5. Opción "Ver Mis Resultados" 84

4.3.1.6. Opción "Busca Usuario"..... 85

4.3.1.7. Opción "Ver Perfil" 86

4.3.2. Pantalla de Bienvenida para Usuarios Registrados 88

4.3.2.1. Opción “Mostrar Exámenes” para Usuarios Registrados 88

4.3.2.2. Opción "Ver Mis Resultados" 90

4.3.2.3. Opción "Ver Perfil" 90

CONCLUSIONES Y TRABAJOS A FUTURO 93

BIBLIOGRAFÍA 95

ÍNDICE DE FIGURAS

1.1.4.1. Lista de Sprints a realizar 6

2.1.1.1. Símbolos básicos para diagramas de entidad-relación..... 9

2.1.2.1.1. Representación gráfica de una relación de tres conjuntos 13

2.1.2.1.2. Representación gráfica de la conexión de varios registros..... 13

2.1.2.2.1. Tabla platos con algunos atributos 15

2.3.1.1.1. Esquema básico de MVC (Modelo-Vista-Controlador)..... 27

2.3.1.1.1. Esquema básico de MVC (Modelo-Vista-Controlador)..... 27

2.4.1. Analogía de la última pieza que hay que agregarle a un Framework 29

3.7.1. Nomenclatura para identificar los Id's de las tareas a realizar 45

3.9.1. Modelo Relacional de la Base de Datos del Sistema 48

3.10.1. Representación de los componentes del MVC utilizados en el sistema 51

4.2.1.1. Imagen 1 del Slider en la página de Inicio..... 75

4.2.1.2. Imagen 2 del Slider en la página de Inicio..... 75

4.2.1.3. Imagen 3 del Slider en la página de Inicio..... 76

4.2.2.1. Página de login para todos los Usuarios 76

4.2.3.1. Página de registro para un usuario nuevo 77

4.3.1.1. Página principal para un Administrador..... 78

4.3.1.1.1. Lista de usuarios registrados..... 79

4.3.1.2.1. Captura de pantalla al seleccionar la opción "Mostrar Certificaciones" 79

4.3.1.2.2. Página para registrar una certificación 80

4.3.1.2.3. Administración de certificaciones utilizando la Librería Grocery Crud 80

4.3.1.3.1. Lista de exámenes creados 81

4.3.1.3.2. Lista de certificaciones en las que se puede crear un examen 82

4.3.1.3.3. Página mostrada al seleccionar una certificación para crear un examen 82

4.3.1.4.1. Lista de certificaciones en las que se puede crear una pregunta..... 83

4.3.1.4.2. Lista de preguntas agregadas en la certificación que fue seleccionada 83

4.3.1.4.3. Página mostrada al seleccionar una certificación para agregar pregunta..... 84

4.3.1.5.1. Calificación de los exámenes que ha respondido el usuario en sesión 84

4.3.1.6.1. Buscador de usuarios registrados en el Sistema 85

4.3.1.6.2. Ejemplo de búsqueda exitosa de un usuario 85

4.3.1.7.1. Página mostrada al seleccionar la opción: “Ver Perfil”..... 86

4.3.1.7.2. Página mostrada al seleccionar la opción: “Editar Perfil”..... 87

4.3.1.7.3. Pantalla que muestra la opción: “Cerrar Sesión” del menú principal..... 87

4.3.2.1. Pantalla de Bienvenida para usuarios registrados 88

4.3.2.1.1. Opción: “Mostrar Exámenes” para usuarios registrados 89

4.3.2.1.2. Ejemplo de Examen mostrado a los usuarios que lo elijan..... 89

4.3.2.2.1. Lista de calificaciones en los exámenes contestados 90

4.3.2.3.1. Pantalla mostrada al seleccionar la opción “Ver Perfil” 91

4.3.2.3.2. Pantalla mostrada al seleccionar el enlace “Editar Información” 91

4.3.2.3.3. Pantalla que muestra la opción: “Cerrar Sesión” del menú principal 92

RESUMEN

Existen diversas empresas e instituciones nacionales e internacionales encargadas de certificar personal con el fin de obtener un conocimiento específico, dicho conocimiento puede ser teórico o práctico dependiendo del área en la cual se necesite desarrollarse. El método más común para comprobar los conocimientos adquiridos es mediante un examen, donde se evalúa el nivel de preparación que se ha obtenido y dependiendo del resultado se puede conseguir un reconocimiento que valide la capacidad en el área seleccionada.

Este documento pretende contribuir en el diseño e implementación de un sistema Web, el cual será capaz de crear exámenes teóricos de opción múltiple con la finalidad de evaluar el conocimiento adquirido, de acuerdo a los resultados se podrá certificar o no en dichas aptitudes. El objetivo es producir un sistema estable, dinámico, web, gratuito, que sea útil y que facilite en tiempo y costos la planeación y realización de un examen tradicional, sin hacer a un lado el objetivo principal: *evaluar el conocimiento de cualquier aspirante a certificación que utilice dicho sistema*.

Se pretende llevar un control de los alcances obtenidos en las capacitaciones realizadas, donde las estadísticas nos permitirán inferir los puntos donde se debe poner más énfasis al enseñar un conocimiento, ya sea general o específico.

Para el desarrollo del proyecto se utilizará la metodología Ágil SCRUM,[1] que propone un modelo de proceso incremental, basado en iteraciones y revisiones continuas. SCRUM se puede definir como una metodología con herramientas que se enfocan en la gestión y desarrollo del software. La base fundamental de esta metodología consiste en la división del trabajo completo (Product Backlog) en distintos apartados o bloques que pueden ser abordados en periodos cortos de tiempo (1-4 semanas) que se denominan Sprints.

El sistema a realizar requiere de elementos como: APACHE Y PHP, el framework CODEIGNITER 2.1.4 como herramienta de desarrollo, MYSQL como manejador de base de datos y BOOTSTRAP como hoja de estilo.

Al finalizar este trabajo, presentamos un sistema de generación y evaluación de exámenes para la certificación de usuarios que han recibido una capacitación previa. Los exámenes se podrán crear de distintas preguntas apoyándose en una base de datos y de acuerdo a las necesidades que requiera el administrador. La evaluación puede ser realizada en un intervalo de tiempo definido y los resultados serán obtenidos en línea.

Para finalizar, podemos decir que los exámenes de certificación son una forma de mantenerse actualizado y sobre todo de especializarse en algún área. Con ellos podemos tener beneficios de distinto tipo, como lo son: remuneraciones económicas, obtención o aumento en un puesto laboral y la preparación obtenida que con el tiempo se reflejara en mejores oportunidades en general.

[1] La palabra “Scrum” no es un acrónimo, sino un término extraído del Rugby que significa melé, un tipo de jugada en la cual los jugadores se agrupan en una formación para obtener el balón. Referencia: https://iswugscrumaps2.wordpress.com/historia-3/

CAPÍTULO 1: INTRODUCCIÓN

En este capítulo se presentan el contexto y marco conceptual de la problemática a la cual se dará solución. De esta manera, el lector comprenderá el escenario real y entenderá la solución propuesta, además de que podrá identificar sus alcances y analizar otras soluciones existentes. Asimismo se presenta la planificación de tareas y actividades a ser realizadas, para terminar con la descripción de la solución implementada.

1.1. Justificación

En la actualidad, es necesario adquirir nuevas habilidades y conocimientos que te hagan ser competitivo en el mercado laboral. Para esto es necesario especializarse en las diferentes áreas del conocimiento, donde muchas veces es requerido que antes de desempeñar una acción específica se cuente con la certeza de que las personas son capaces de llevar a cabo las tareas de su especialidad. Algunos de estos conocimientos se pueden conseguir tomando un curso de certificación, mismo que nos puede ayudar a obtener nuevas destrezas y aptitudes. Es muy común que para evaluar un curso de certificación se aplique un examen y en el ámbito de la computación se utiliza como apoyo la tecnología, misma que nos ayudará a que la mayoría de estos exámenes sean realizados mediante un software.

Aunque existe una gran cantidad de software para la creación de exámenes de evaluación, muchos tienen un alto costo debido a que utilizan licencias, permisos o convenios necesarios para su funcionamiento, además de que están diseñados para un tema en específico y en algunos casos deben ser adquiridos directamente de una empresa productora de software.

Debido a la necesidad del departamento de Bases de Datos de la Facultad de Ciencias de la Computación en la Benemérita Universidad Autónoma de Puebla de poseer un *software gratuito* que permita la generación de exámenes de evaluación aplicados en certificaciones que se realicen en el futuro, surge la idea de este proyecto.

En esta documentación, se describe como tal el sistema generador de exámenes, que es capaz de crear reactivos de tipo “opción múltiple”. Estos reactivos pueden ser creados de manera manual y después ser seleccionados para crear uno o más exámenes, lo que hace más fácil y rápida la generación de exámenes. En particular, es una herramienta útil enfocada principalmente para usarse en la enseñanza de Bases de Datos, aunque también podría utilizarse en otras áreas.

El sistema está destinado a contribuir con el trabajo del instructor de un curso y puede ser muy útil para programas de certificación de competencias y en cursos que se imparten de manera regular en los programas de estudio de diferentes niveles de educación, además de que puede ser un producto provechoso como un instrumento para la evaluación.

1.2. Definición de Problema

El tema de investigación propuesto, surge de la necesidad por utilizar un software gratuito, que sea capaz de ayudar al instructor de un curso de Base de Datos para formular exámenes con reactivos de opción múltiple referentes a esta área de la computación. Existe una gran necesidad de tener este tipo de sistema en un ambiente Web, ya que existen pocos sistemas que sean gratuitos, y aunque es claro que se ha desarrollado una gran variedad de software de este tipo, en su mayoría se deben pagar altas cuotas para poder utilizarlos.

Sin duda hay un gran problema cuando se necesita crear un sistema que sea capaz de evaluar los conocimientos que un estudiante ha adquirido durante un curso, los principales involucrados son los instructores que imparten el curso y los estudiantes que lo eligen y que al final tienen que reflejar los conocimientos que han adquirido.

A partir de la solución de este problema con dicho programa de cómputo, se tienen beneficios para los profesores, por ejemplo la facilidad de diseñar de manera ágil las evaluaciones sin tener que pagar por el uso de un software, así como automatizar el proceso de generación de exámenes, mantener una gran base de datos con reactivos de opción múltiple y facilitar el control y seguimiento de un curso; en el caso de los estudiantes el sistema actúa como una guía de estudio, es decir, es una herramienta que les ayuda a medir el grado de dificultad de las preguntas en un examen, en el que obtendrán una calificación, así mismo se podrá interpretar en que área del conocimiento hay que poner más énfasis en el estudio.

Esta solución está diseñada especialmente para su utilización como una herramienta dinámica en el diseño de exámenes en cursos de certificación (en este caso en el área de Bases de Datos Relacionales).

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema que sea capaz de generar exámenes para certificaciones en Bases de Datos Relacionales, con reactivos de opción múltiple, donde las preguntas para cada examen deberán ser seleccionadas manualmente, además de evaluar las respuestas contestadas y así obtener una calificación.

1.3.2. Objetivos Específicos

- Reunir información y obtener los requerimientos necesarios para el desarrollo del sistema.
- Llevar a cabo una planificación con los objetivos propuestos.
- Hacer un análisis con la información y requerimientos obtenidos.
- Diseñar el modelado del sistema.
- Diseñar la interfaz de usuario.
- Realizar el modelo de la base de datos.
- Elaborar un prototipo funcional.
- Validar cada elemento del sistema.
- Redactar la información necesaria en cada fase del proyecto.
- Evaluar el sistema y entregarlo.

1.4. Plan del Proyecto

En esta sección se describe la metodología y procedimiento utilizado para llevar a cabo la administración del proyecto, así como del ciclo de desarrollo del software y la guía de trabajo a seguir.

1.4.1. Metodología y procedimiento

Para el desarrollo del proyecto se utilizará la metodología Ágil SCRUM, que propone un modelo de proceso incremental, basado en iteraciones y revisiones continuas.

SCRUM se puede definir como una metodología con herramientas que se enfocan en la gestión y desarrollo del software. Dado que es un proceso iterativo y que va incrementándose gradualmente, se basa en un desarrollo ágil del software a través de etapas que se complementan, además de que se asignan roles con trabajos definidos y se organizan las actividades diarias.

La base fundamental de esta metodología consiste en la división del trabajo completo (Product Backlog) en distintos apartados o bloques que pueden ser abordados en periodos cortos de tiempo (1 a 4 semanas) que se denominan Sprints.

Explicación del trabajo a realizar:

- Al inicio de este documento se explica la justificación del proyecto, sus antecedentes y objetivos.
- Después se procede a establecer claramente los requerimientos creando el “Product Backlog”.
- Al inicio de cada iteración se seleccionan aquellos requerimientos que tengan mayor valor.
- Se analizarán y definirán tareas con el fin de lograr un mejor desarrollo, esas tareas conforman el Sprint y se registran en el Sprint Backlog (en la reunión de Planificación).
- Cuando se llegue al final de cada Sprint debe haber una nueva funcionalidad para probar, la cual es presentada para revisión por parte de los encargados de la supervisión del proyecto.
- Eventualmente se revisará el estado de las tareas para detectar y remover obstáculos.

Con esto, podemos ver que en el Sprint 0 se planea tener 6 iteraciones (Sprints), cada una con una duración de 2 a 3 semanas. Cada Sprint (Del 1 al 6) contará con su propia lista de Backlogs (tareas a realizar) con el objetivo común de cumplir con los requerimientos necesarios del sistema.

Después de la finalización de cada Sprint se tendrá una parte del proyecto que será revisada y evaluada.

A continuación se muestra una Lista con los Sprints a realizar, también llamado SPRINT 0. Dicha lista la podemos ver gráficamente en Figura 1.1.4.1.

- Especificaciones
- Análisis
- Diseño
- Codificación
- Pruebas
- Mantenimiento

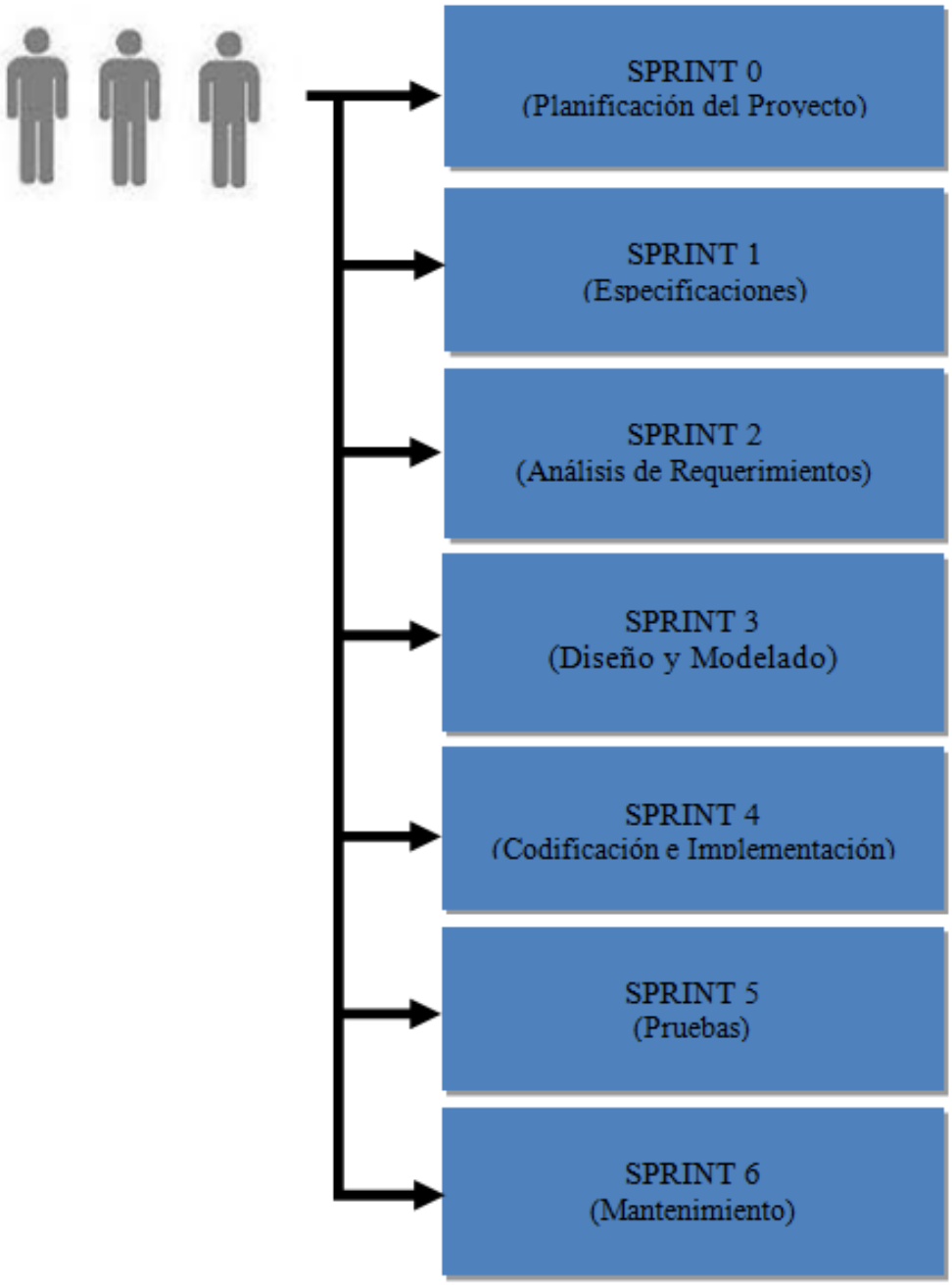

Figura 1.1.4.1: Lista de Sprints a realizar

1.5. Alcances y limitaciones

Una vez finalizado el proyecto, se tendrá un sistema en WEB, capaz de crear exámenes de certificación con reactivos de opción múltiple, que además será fácil de utilizar y que dará al usuario una manera práctica de automatizar algunas actividades en el proceso de evaluación de un curso de certificación en el área de Bases de Datos, aunque no se descarta que en un futuro esta aplicación pueda ser utilizada en otro u otros cursos de certificación.

Adicionalmente, el software podrá ser utilizado bajo licencia gratuita, ya que se plantea el hecho de trabajar en la filosofía Open Source. Con esto se permitirá a cualquier entidad universitaria utilizarlo para el propósito para el cual ha sido creado.

Con el sistema se podrá asegurar la calidad de un curso impartido, es decir se podrá corroborar si la base de datos de reactivos y los exámenes que se generan con el sistema, cumplen con los objetivos establecidos por el personal especializado en el tema. Se tratará de generar un ambiente similar al de un examen en una certificación comercial de Bases de Datos.

La Base de Datos únicamente almacenará la información necesaria para el funcionamiento, pero no se descarta la posibilidad de agregar otras funcionalidades, pues al ser un sistema “open Source” se podrá modificar de acuerdo a las necesidades del usuario. Dependiendo de las modificaciones que se le realicen, este sistema puede ser usado en laboratorios, bibliotecas, empresas, entre otros.

El programa en lo que respecta al entorno de trabajo, consta de las siguientes funciones: registrar usuario en el sistema, actualizar datos personales, registrar una certificación, registrar una pregunta, crear exámenes y resolver un examen creado por un usuario administrador.

CAPÍTULO 2: MARCO TEÓRICO

En este capítulo se dan a conocer algunos conceptos básicos del contexto de este trabajo, esto con la finalidad de situar al problema en un conjunto de conocimientos. Dentro de esta información se explican algunos conceptos de Bases de Datos, del patrón de diseño MVC (Modelo-Vista-Controlador), del framework Codeigniter y sus características, de algunos certificadores comerciales, entre otros elementos.

2.1. Base de Datos

Una base de datos es una colección de archivos relacionados que permite el manejo de información de un lugar. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de algún atributo de una entidad del mundo real. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar.

Ejemplo 1:

- Para una UNIVERSIDAD, las entidades de interés podrían ser: ALUMNOS, PROFESORES, SALONES, etc.
- Para la entidad ALUMNO, los atributos importantes podrían ser: NOMBRE, DIRECCION, TELEFONO, CARRERA, MATRICULA, etc.

Ejemplo 2:

- Para una INDUSTRIA, las entidades de interés podrían ser: DEPARTAMENTOS, EMPLEADOS, PRODUCTOS, etc.
- Para la entidad DEPARTAMENTO, los atributos importantes podrían ser: NOMBRE DEL DEPARTAMENTO, NOMBRE DEL DIRECTOR DEL DEPARTAMENTO, FUNCION DEL DEPARTAMENTO, etc.

Un archivo de una base de datos también puede ser pensado como una tabla en la que tenemos renglones y columnas, cada renglón correspondiendo a un registro del archivo y cada columna correspondiendo a un campo.

2.1.1 Modelo de datos Entidad-Relación(E-R)

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de base de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. En la figura 2.1.1.1 podemos ver los símbolos más utilizados al elaborar diagramas entidad-relación.

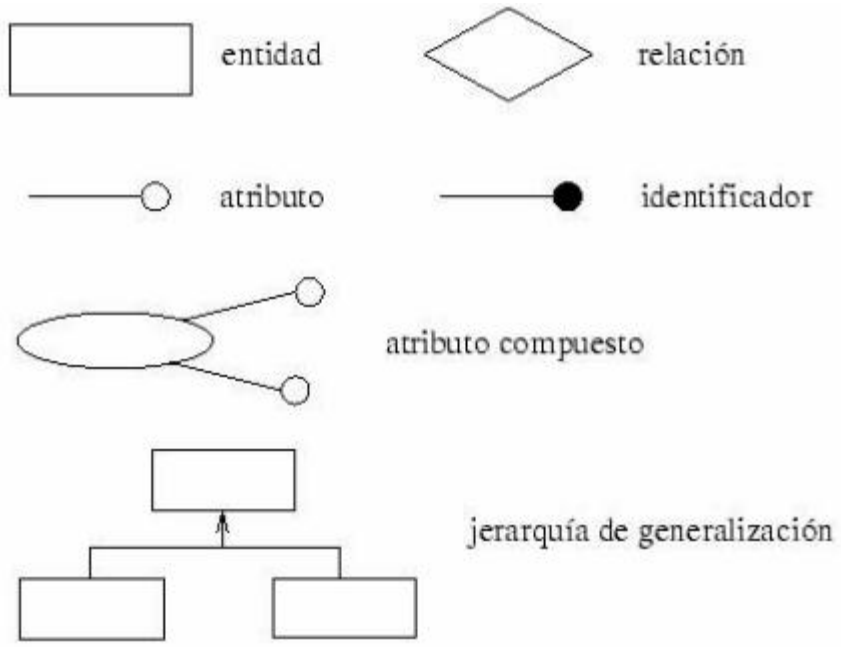

Figura 2.1.1.1: Simbología básica para diagramas de entidad-relación

2.1.1.1. Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, productos, viajes, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles.

- Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad.
- Una entidad fuerte es una entidad que no es débil.

2.1.1.2. Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes de una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria y así sucesivamente.

2.1.1.3. Cardinalidad

La Cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante.

2.1.1.4. Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina multivaluados, y pueden tener un número máximo y un número mínimo de valores. La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece, el valor por omisión es (1,1).

Por último, los atributos pueden ser derivados. Un atributo derivado es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

2.1.2. El modelo relacional

El modelo relacional fue propuesto por E.F. Codd en los laboratorios de IBM en California. Se trata de un modelo lógico, que establece una estructura sobre los datos, aunque posteriormente éstos puedan ser almacenados de múltiples formas para aprovechar características físicas concretas de la máquina sobre la que se implante la base de datos realmente. El modelo relacional constituye una

alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que además de proporcionarnos los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de un álgebra relacional) para su manipulación, sin ambigüedad posible.

El carácter formal del modelo relacional hace relativamente sencilla su representación y gestión por medio de herramientas informáticas. No es casual, pues, que haya sido elegido como referencia para la construcción de la gran mayoría de los Sistemas de Gestión de Bases de Datos comerciales disponibles en el mercado; ni tampoco que sea también habitualmente seleccionado como modelo de referencia para la elaboración del esquema lógico de una base de datos, como tercer paso de la habitual metodología de diseño de Bases de Datos (después del análisis de requerimientos y la elaboración del esquema conceptual).

En el modelo relacional se basa en el concepto matemático de relación. En este modelo, la información se representa en forma de “tablas” o relaciones, donde cada fila de la tabla se interpreta como una relación ordenada de valores (un conjunto de valores relacionados entre sí).

2.1.2.1. Estructura general

El nombre de modelo relacional viene de la estrecha relación que existe entre el elemento básico de este modelo, y el concepto matemático de relación. Podemos decir que una relación R está definida sobre los conjuntos D1, D2,.., Dn. Si nos fijamos en la Figura 2.1.2.1.1, podemos ver que una de estas relaciones no es más que una lista de líneas, donde cada línea está dividida en trozos.

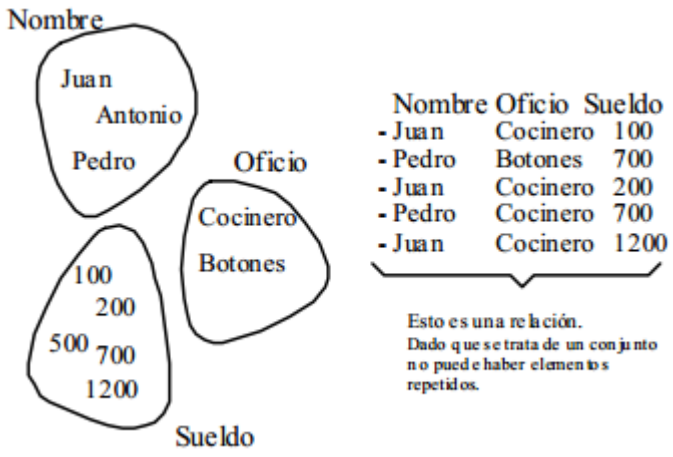

Figura 2.1.2.1.1: Representación gráfica de una relación de tres conjuntos

El modelo relacional representa las listas de líneas mediante registros o fichas (como se puede observar en la figura 2.1.2.1.2), cada una de las cuales puede ser manejada individualmente y con independencia de las demás. No obstante, a efectos de facilitar la visualización, puede ser posible ver todas las líneas juntas como si de una lista se tratase.

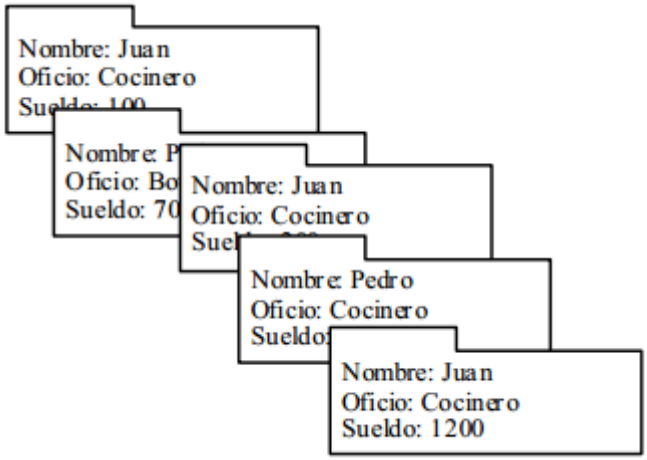

Figura 2.1.2.1.2: Representación gráfica de la conexión de varios registros

De esta manera, tendremos varios tipos de fichas: fichas de clientes, de proveedores, de facturas, de albaranes, de reservas, de empleados, de apuntes contables, etc., cada una de las cuales podemos almacenar en un cajón o en un fichero independiente. De cada tipo de ficha tendremos un montón de fichas

rellenas: 100 ó 200 clientes, 4 ó 5 proveedores, miles de facturas, etc. Por tanto, es interesante distinguir entre un tipo de ficha, que no hace referencia a ninguna ficha rellena en concreto (p.ej. una ficha de clientes), y una ficha concreta, rellena con unos datos concretos (la ficha de Juan el Cocinero).

Pues bien, el modelo relacional plasma en un ordenador este mismo esquema, aprovechando las enormes características de computación y almacenamiento de las máquinas actuales.

2.1.2.2. Concepto de tabla. Dominios y atributos.

Una tabla en el modelo relacional viene a ser como una de las listas descritas anteriormente. Una tabla o relación es una matriz rectangular que almacena líneas con una estructura concreta:

- La primera línea de una tabla, es una cabecera que indica el nombre de cada columna. O sea, cada columna tiene asignado un nombre único, e indica que los artículos almacenados en esa columna deben pertenecer a un conjunto de valores concreto: Números, Letras, Frases, etc.
- Cada línea (excepto la primera) recibe el nombre de tupla, y almacena artículos concretos para cada columna.
- Todas las filas deben ser diferentes entre sí.
- El orden de las filas y de las columnas carece de importancia a efectos del S.G.B.D. Este hecho es el que verdaderamente diferencia las tablas relacionales del concepto matemático de relación, en el que el orden de las columnas es fundamental.

En la figura 2.1.2.2.1 aparece una tabla con el nombre de varios Platos. Con este ejemplo podemos ver que toda tabla debe poseer al menos el primer elemento, que identifica el nombre de la columna.

PLATOS

Código	Nombre	Precio	Menú
15	Solomillo a la pimienta	1000	No
23	Fondue Neûchatel	1500	No
17	Migas con chocolate	850	Sí
34	Ajo blanco con uvas	850	Sí
12	Paella valenciana	1100	Sí
21	Mono a la cantonesa	7500	No
07	Sopa de aleta de tiburón	525	Sí

Figura 2.1.2.2.1: Tabla platos con sus algunos atributos

En este caso, nuestra tabla o relación contiene las columnas Código, Nombre, Precio y Menú (que indica si ese plato pertenece o no a las opciones del menú del día).

El grado de esta tabla es el número de campos que posee, en nuestro caso 4. La Cardinalidad es el número de tuplas concretas que almacena: 7. Nótese que el grado de una tabla es independiente del momento en que observemos la tabla (el número de columnas es invariable, y queda definido en el momento en que se crea la tabla), mientras que la Cardinalidad depende de la situación real que represente la tabla y puede variar en el tiempo; p.ej., la lista de platos puede cambiar todos los meses, o según la estación del año, para adecuarse a los gustos propios de cada una de ellas.

La primera fila de una tabla es la más importante, ya que nos da su estructura. Esta columna identifica los nombres de campo o atributos de que consta cada tabla. En otras palabras, cada tupla está formada por un conjunto de información estructurada en elementos más simples llamados atributos. El nombre del atributo debe describir el significado de la información que representa.

Los dominios a que puede pertenecer un atributo, suelen depender de los que proporcione el S.G.B.D. que empleemos. Suelen ser comunes dominios como: Texto, Número entero, Número decimal, Fecha, Hora, Sí/No decimal, Fecha, Hora, Sí/No, etc.

Independientemente del dominio a que pertenezca un atributo, cualquier atributo puede tomar un valor especial que designa la ausencia de dato; es el valor NULO (en inglés NULL o NIL). Cuando, por cualquier circunstancia, se desconoce el valor de un atributo, como p.ej. la dirección de un cliente, o bien ese atributo carece de sentido (el atributo Teléfono de un empleado que no tiene teléfono en su casa), podemos asociar a dicho atributo este valor especial, común a cualquier dominio. Hay que hacer notar la diferencia entre el valor NULO, y el valor «espacios en blanco»; el ordenador considera un espacio en blanco como cualquier otro carácter, por lo que a efectos computacionales, la introducción de caracteres «espacios en blanco» es distinta al concepto de «ausencia de información». Los espacios en blanco sí son datos, y pertenecen al dominio Texto.

2.1.2.3. Restricciones.

En el apartado anterior observamos que cada atributo está obligado a tomar un valor perteneciente a un dominio concreto, siendo imposible el que guarde otro distinto. Esto supone una restricción sobre los atributos.

Otras restricciones ya comentadas son:

- La imposibilidad de repetir tuplas en una misma tabla.
- La imposibilidad de establecer un orden en las tuplas, aunque algunos S.G.B.D. relajen un poco esta restricción.

En este apartado vamos a introducir otras restricciones más importantes que posee el modelo relacional, así como los conceptos implicados. Por último veremos las posibilidades que tiene el usuario para definir restricciones en función de las características propias de su trabajo.

2.1.2.4. Reglas fundamentales. Claves

El modelo relacional intenta representar con una tabla a un tipo de objetos de la vida real, como puedan ser Empleados, Clientes, etc., e incluso considera las relaciones entre estos objetos como objetos en sí mismos. Para ello, designa cada tipo de objetos por un conjunto de atributos que son los que darán la «forma particular» a cada objeto. Volvamos al caso de nuestra tabla de Platos.

Para representar a un Plato, hemos escogido los atributos: Código, Nombre, Precio y Menú. Un plato concreto puede ser p.ej., (17, «Migas con chocolate», 850, Sí). Este plato concreto, como cualquier otro objeto distinguible del mundo real posee unas características que lo distinguen de los demás de su mismo tipo, tal y como lo hace el ADN de una persona. El ADN conforma la esencia o clave de toda persona. Es más, todo objeto posee algo concreto que lo hace diferente; incluso puede poseer más de una cosa por la que se le puede diferenciar: una persona puede distinguirse también por su nacionalidad y su D.N.I., lo que conforma otra clave de esa persona.

Como en una tabla, las tuplas pueden estar en cualquier orden, no podemos referenciar una tupla concreta mediante su posición entre las demás, y necesitamos alguna forma de seleccionar una tupla concreta. La forma de hacerlo es mediante una clave.

Una clave es un atributo o conjunto de atributos cuyo valor es único y diferente para cada tupla.

Cada tabla puede poseer más de una clave. P.ej., en nuestro caso de la tabla Platos, la clave puede ser tanto el atributo Código, como el atributo Nombre. Tenemos dos claves potenciales, también llamadas claves candidatas. De entre todas las claves candidatas, el administrador, cuando define la tabla, debe decidir cuál de ellas va a ser la clave principal o clave primaria, mientras que el resto de las claves pasan a denominarse claves alternativas o claves alternas. La distinción entre clave principal y claves alternativas, es sólo a efectos de acceso interno a los

datos, y para que el S.G.B.D. adopte ciertas decisiones sobre cómo almacenar esos datos en los sistemas de almacenamiento masivos.

El concepto de clave es tan importante, que da lugar a una serie de restricciones fundamentales sobre la base de datos. Son la regla de identificación única y la regla de integridad referencial.

2.1.2.5. Regla de identificación única

Esta restricción ya fue estudiada en el tema de los diagramas E-R. En esencia, los conceptos de clave de una entidad en el diagrama E-R, y de clave de una tabla coinciden plenamente. Así pues, al igual que en aquel momento, podemos enunciar esta regla de la misma forma:

«En ninguna tupla de una tabla, ninguno de los atributos que formen parte de la clave primaria de una relación podrá tomar un valor nulo. El valor de la clave será único para cada tupla de la tabla».

Esta regla nos dice que una vez escogida la clave primaria de una tabla, y dado que ninguno de los atributos que la componen es superfluo, no podemos dejar de lado el valor de ninguno de ellos para identificar unívocamente a una tupla. O sea, el que todos los atributos de la tabla sean necesarios está en contradicción con que alguno de ellos pueda carecer de valor. Ningún atributo de la clave puede ser vacío porque en tal caso no sería una característica identificativa propia del objeto a que pertenece.

2.1.2.6. Regla de integridad referencial.

Esta otra regla, aunque también tiene una relación directa con el concepto de clave, está quizás más vinculada a la forma en que un diagrama E-R se convierte en un esquema relacional, por lo que puede no ser bien comprendida en este punto. No obstante, por completitud se da aquí su enunciado:

«Si una tupla de una tabla A posee atributos (A1...An) que hacen referencia a la clave primaria de otra tupla de una tabla B, dichos atributos poseen, o bien valores nulos, o bien valores (V1...Vn) que se corresponden con la clave de una tupla concreta de B».

2.1.3. Sistema de Gestión de Bases de Datos (DBMS Database Management System)

Un sistema de gestión de Bases de Datos (DBMS Database Management System) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos, es básicamente un sistema computarizado para llevar registros, es decir, un sistema cuya finalidad general es almacenar información y permitir a los usuarios realizar una variedad de operaciones sobre dichos registros, por ejemplo: insertar, borrar, modificar, agregar, consultar. El objetivo primordial de un DBMS es proporcionar todas las facilidades que sean posibles para que la extracción o almacenamiento de información en la Base de Datos sea eficiente.

Los DBMS están diseñados para gestionar grandes bloques de información, ya sea para la definición de estructuras para el almacenamiento o para los mecanismos necesarios para el manejo de la información.

Además los DBMS deben mantener la seguridad de la información almacenada pese a la caída del sistema o accesos no autorizados.

2.2. Bases de Datos en la WEB

La Web es un medio que en la mayoría de los casos trabaja con bases de datos, para localizar, enviar, recibir información de diversos tipos, entre otras funciones. En el ámbito competitivo, es esencial ver las ventajas que esta vía electrónica proporciona para presentar la información, reduciendo costos y el almacenamiento de la información, además de aumentar la rapidez de difusión de la misma.

En el pasado, las Bases de Datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero en la actualidad la Web permite acceder a una Base de Datos desde cualquier parte del mundo. Estas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de la información, con ventajas que no podrían obtenerse de otro medio.

Con estos propósitos, los usuarios de Internet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo y recursos mínimos. Asimismo, las Bases de Datos serán usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

2.2.1. Servidores WEB

El servidor de Páginas es la parte primordial de cualquier sitio de Internet, ya que es el encargado de generar y enviar la información a los usuarios finales.

Sin embargo, conforme la demanda de los Servidores de páginas Web incrementó, fue necesario agilizar este proceso, el llamar a un intérprete para que ejecutara otro programa, ponía una demanda muy fuerte sobre el equipo que mantenía el Servidor de Páginas Web.

Hoy en día, se puede decir que todos los Servidores de Páginas Web son Servidores de Aplicaciones, ya que se les ha desarrollado alguna funcionalidad especial que les permite realizar, valga la redundancia, aplicaciones de Servidor.

2.2.2. Apache, PHP y MySQL

Sin duda, al hablar de desarrollo de aplicaciones con software libre, es necesario mencionar tres elementos muy importantes, como lo son: Apache, PHP y MySQL. Estas herramientas proporcionan el soporte prefecto para el desarrollo las aplicaciones Web ya antes mencionadas. Veamos cómo funcionan las aplicaciones basadas en ellos.

Por separado son programas potentes, de código abierto y que nos pueden dar respuestas a muchos problemas que necesitamos resolver. En conjunto, forman de cierto modo, el estándar sobre el que se realizan muchos de los proyectos pensados para operar en la nube o en una red de área local.

2.2.2.1. Apache

Es un servidor web que nos permite visualizar páginas a través de nuestro navegador, ya estén alojadas en un servidor remoto, en nuestro servidor o en nuestro ordenador. Es el servidor web más popular y entre sus ventajas destaca la modularidad, ser multiplataforma y la gran comunidad que tiene detrás con una buena documentación y soporte para resolver dudas y problemas. Sin duda una herramienta aprobada y fiable.

2.2.2.2. PHP

Hipertext Preprocesor (PHP) fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Es un lenguaje de programación interpretado del lado del servidor, gratuito, multiplataforma, e independiente de la plataforma, es rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al navegador Web. Las páginas que se ejecutan en el servidor pueden realizar accesos a Bases de Datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de PHP.

Es independiente de cualquier plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser

compatible con el lenguaje y significa una ventaja importante, ya que permite visualizar el sitio desarrollado en PHP en un sistema o en otro y prácticamente sin ningún trabajo. Este lenguaje es utilizado muchas veces para conexiones con bases de datos como MySQL.

2.2.2.3. MySQL

El software MySQL® proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License (http://www.fsf.org/licenses/) o pueden adquirir una licencia comercial estándar de MySQL AB.

MySQL, es el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.

El sitio web MySQL (http://www.mysql.com/) proporciona la última información sobre MySQL y MySQL AB.

- MySQL es un sistema de gestión de bases de datos

Una base de datos es una colección estruturada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar

los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en computación, como aplicaciones autónomas o como parte de otras aplicaciones.

- MySQL es un sistema de gestión de bases de datos relacionales

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL"se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones.

- MySQL software es Open Source.

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adapatarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License), http://www.fsf.org/licenses/, para definir lo que puede y no puede hacer con el software en diferentes situaciones. Si no se encuentra cómodo con la GPL o necesita añadir código MySQL en una aplicación comercial, puede comprarnos una licencia comercial. Consulte la Introducción a las Licencias MySQL para más información (http://www.mysql.com/company/legal/licensing/).

- El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.

Si esto es lo que está buscando, debería probarlo. El servidor MySQL también tiene una serie de características prácticas desarrolladas en cooperación con los usuarios. Puede encontrar comparaciones de rendimiento de MySLQL Server con otros sistemas de gestión de bases de datos en nuestra página de comparativas de rendimiento.

MySQL Server se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL Server ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.

- Una gran cantidad de software de contribuciones está disponible para MySQL

Es muy posible que nuestras aplicaciones o lenguajes favoritos tengan soporte por el servidor de base de datos MySQL.

Existen varias interfaces de programación de aplicaciones que permiten acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, entre otros; cada uno de estos utiliza una interfaz de programación de aplicaciones específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

2.3. MVC: Modelo-Vista-Controlador (Model-View-Controller).

MVC es un patrón de diseño de software que se ubica en la categoría de patrones arquitectónicos; este se especifica bajo la proposición de dividir la aplicación en tres tipos de elementos, el modelo, las vistas y controladores, es decir, separar el código de los programas por sus diferentes responsabilidades. MVC 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. La manera en que los elementos dentro de MVC se comunican, difieren dependiendo del tipo de aplicación que se está describiendo.

Sin embargo, en el centro de cada arquitectura MVC se encuentran presentaciones separadas, lo que declara una división especifica entre los objetos de dominio que modelan nuestra percepción del mundo real (modelo de objetos), y la presentación de dichos objetos que son los elementos que vemos en la pantalla (objetos de vista).

MVC define la separación de estos tres tipos de elementos:

- Modelo: Elementos (objetos) que contienen los datos y definen la lógica para manipular dichos datos. A menudo los objetos tienen una naturaleza reutilizable, distribuida, persistente y portátil para una variedad de plataformas.

- Vista: Hace referencia a los elementos que representan algo visible en la interfaz de usuario, por ejemplo, un panel o botones. Con el fin de mostrar datos de los objetos de modelo es posible que desee crear sus propios objetos personalizados, como un gráfico, por ejemplo.

- Controlador: Actúa como un mediador entre los objetos del modelo y la vista. Un objeto Controller comunica datos de ida y vuelta entre los objetos del modelo y de la vista. Un controlador también realiza todas las tareas específicas de la aplicación, tales como la entrada del usuario o la carga de procesamiento de datos de configuración. Por lo general se puede ver un

controlador por aplicación o ventana, en muchas aplicaciones, el controlador está estrechamente acoplado a la vista. Dado que los controladores son específicos de aplicaciones, por lo general es muy difícil encontrar reutilización en otras aplicaciones.

El campo de desarrollo de este patrón es tan amplio que se puede utilizar casi en cualquier aplicación. Por supuesto, dependiendo de la aplicación puede que algunas clases varíen y deban acoplarse a otras, sin embargo, en general, siempre es una buena idea estructurar la aplicación en orden a MVC.

2.3.1. Aplicación del patrón MVC en PHP

Hasta la aparición de los diferentes Frameworks dentro del mundo del desarrollo de PHP, existían dudas sobre el éxito de la aplicación de este patrón a los desarrollos de PHP. Actualmente, gracias a estos Frameworks, su aplicación es más sencilla y los beneficios producidos son reconocibles.

2.3.1.1. Características

El patrón Modelo-Vista-Controlador para el diseño de aplicaciones Web es un estándar de la industria a nivel mundial. Hay muchos libros y recursos excelentes disponibles sobre el tema que ayudan a acelerar el proceso de aprendizaje para el equipo de desarrollo. Recordemos que, MVC se refiere a Model, View, Controller, donde la idea básica de este patrón es separar nuestros sistemas en tres capas, el modelo, la vista y el controlador.

El modelo se encarga de todo lo que tiene que ver con la obtención de datos, guarda y recupera la información de los elementos que utilicemos, ya sea una base de datos, ficheros de texto, XML, etc.

La vista presenta la información obtenida en el modelo de manera que el usuario la pueda visualizar.

El Controlador, dependiendo de la acción solicitada por el usuario, es el que pide al modelo la información necesaria y llama a la plantilla (la vista) que corresponda para que la información sea presentada.

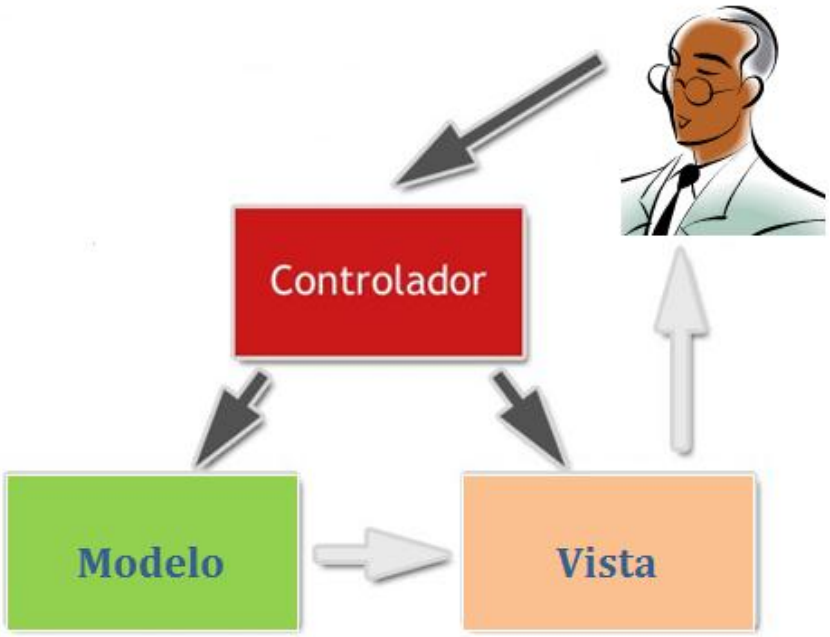

Figura 2.3.1.1.1: Esquema básico de MVC (Modelo-Vista-Controlador)

En la figura 2.3.1.1.1 podemos ver gráficamente cómo sería el flujo de trabajo característico en un esquema MVC, que explicado paso a paso es el siguiente:

- El usuario realiza una solicitud a nuestro sitio web. Generalmente estará desencadenada por acceder a una página de nuestro sitio. Esa solicitud le llega al controlador.
- El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.

- Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha es de otro color.
- Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente, se distingue la flecha en otro color.

Es necesario dedicar un poco de tiempo para aprender a utilizar un marco MVC en PHP. Sin embargo, para el desarrollador de aplicaciones Web grandes, este esfuerzo debe ser recompensado por los numerosos beneficios de utilizar un patrón de diseño MVC, tales como:

- Aplicar la modularidad y la partición de aplicación.
- Aumentar la creación de roles específicos en el desarrollo.
- Aumentar la capacidad de gestión de código.
- Aumentar la posibilidad de extender el código tanto como se requiera (Capacidad de adaptación a cambios).

2.4. Framework

La palabra "framework" o esquema de trabajo, define en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para resolver un tipo de problemática particular, que sirve como referencia para enfrentar nuevos problemas de índole similar.

En el desarrollo de software, un framework, es una estructura conceptual y tecnológica de soporte definido, normalmente con módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Los Frameworks tienen como objetivo ofrecer una funcionalidad definida, auto-contenida, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento.

En general, con el término framework, nos estamos refiriendo a una estructura de software integrada por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta, parecido al rompecabezas de la figura 2.4.1.

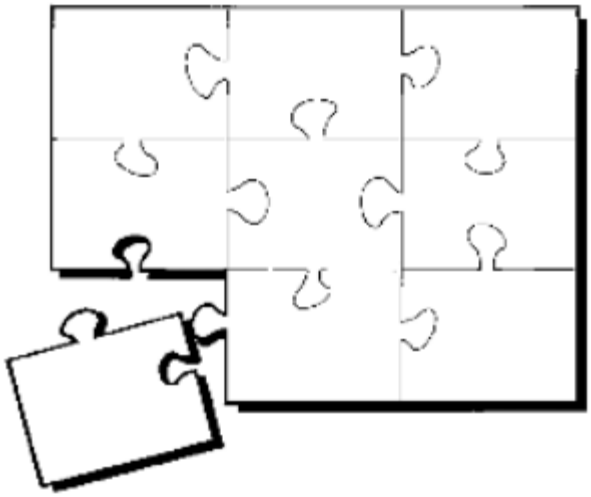

Figura 2.4.1: Analogía de la última pieza que hay que agregarle a un Framework

2.4.1. Frameworks PHP

Un Framework para PHP es una herramienta utilizada para desarrollar aplicaciones web con el lenguaje. Un conjunto de librerías que una vez aprendidas y utilizadas de la mejor manera, nos facilitan el trabajo, ya que nos brindan una cantidad de códigos que nos evitan el reescribir una y otra vez los mismos script.

Actualmente los Frameworks vienen desarrollados bajo una serie de patrones de diseño como lo son: Modelo-Vista-Controlador (El famoso MVC), Active Record, entre otros patrones, además de contar con plantillas fácil de manejar, soporte por parte de la comunidad, librerías, y algunas características más.

Entre los framework para PHP que podemos encontrarnos en la actualidad, están los siguientes.

2.4.1.1. Zend Framework

Zend Framework es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

ZF ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

2.4.2. CodeIgniter

Probablemente ya sepamos que un framework es un programa para desarrollar otros programas, CodeIgniter, por tanto, 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.

CodeIgniter y otros Frameworks PHP pueden ayudarte al trabajar como desarrollador PHP, creando aplicaciones web más profesionales y con código más reutilizable, con la diferencia que CodeIgniter está creado para que sea fácil de instalar en cualquier servidor y de empezar a usar, esta es una ventaja sobre cualquier otro framework. Además muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que goces de mayor libertad a la hora de desarrollar sitios web.

Algunos de los puntos más interesantes de este framework, sobre todo en comparación con otros productos similares, son los siguientes:

- Versatilidad: Quizás la característica principal de CodeIgniter, en comparación con otros Frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.

- Compatibilidad: CodeIgniter es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.

- Actualización: Desde la versión 2 de CodeIgniter ya solo es compatible con la versión 5 de PHP. Para los que todavía usen PHP 4 pueden descargar una versión antigua del framework, como CodeIgniter V 1.7.3, que todavía era compatible. Estas versiones están en la página de descargas de CodeIgniter.

- Facilidad de instalación: No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.

- Flexibilidad: CodeIgniter es bastante menos rígido que otros Frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

- Ligereza: El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

- Documentación tutorializada: La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida cuando ya sabemos acerca del framework y queremos consultar

sobre una función o un método en concreto, pero para iniciarnos sin duda es de mucha ayuda.

Sin duda, lo más destacable de CodeIgniter es su accesibilidad, ya que podemos utilizarlo en la mayor gama de entornos.

2.5. Lenguaje CSS (Cascading Style Sheets - Hojas de Estilo en Cascada)

(Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación. CSS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc. A pesar de que la recomendación oficial del grupo de trabajo de la W3C ya había alcanzado la estabilidad requerida para que fuera soportada por los principales navegadores comerciales, como Netscape e Internet Explorer, en el año de 1998, surgió la situación comúnmente conocida como la “guerra de los navegadores”, que hacía que los intereses comerciales de las dos compañías en lucha por el mercado de usuarios de Internet se interpusieran en el camino de las CSS.

Netscape 4 e Internet Explorer 4 incorporaron parcialmente un soporte a esta recomendación, pero éste dejaba mucho que desear, especialmente en Netscape 4, por lo que no era plausible la incorporación de las CSS en el diseño de sitios, salvo en un muy mínimo número de características, esto aún con reservas, puesto que la manera de tratar los estándares era muy diferente y ni con eso se podía asegurar una visualización correcta de la misma hoja de estilos.

El uso más amplio de las posibilidades de las CSS en Netscape 4, por ejemplo, podía llegar a causar un "crash" en el browser sin explicación alguna. Internet

Explorer 4 tenía una mejor implementación de la recomendación CSS nivel 1, y aunque era parcial, permitía el uso de un mayor número de características, como aquéllas que exceden a la tipografía (la parte de las reglas de las CSS pensada para reemplazar al tradicional uso de tablas en el diseño de "layouts").

A diferencia de entonces, hoy CSS ya no es una novedad, pero ahora ya no hay pretextos para no usarla. Y es posible utilizar ciertas características más amplias de las CSS, como el control de otros aspectos gráficos tales como imágenes y colores de fondo, márgenes exactos y bordes, todo esto para evitar el trabajoso y a veces poco gratificante diseño de tablas complejas para un "layout", que incluye frecuentemente tablas anidadas y complicados algoritmos de combinación de celdas (me refiero a la forma en la que los navegadores resuelven esto para su graficación), características que hacen al archivo muy pesado para descargar, porque inundan el código con la extensa serie de etiquetas requeridas. Hoy la "guerra de los navegadores" ha terminado, la W3C y los estándares empiezan a liderar el camino, por lo que ya no hay tantas trabas para el uso más extendido, aunque medido, de las CSS.

Podríamos decir que los beneficios de usar CSS son dobles. Por un lado, evitamos hacer a los archivos demasiado pesados (excluyendo el largo código requerido para las tablas anidadas y el añadido de características gráficas), y definimos el "estilo visual" de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta, para cada una de las páginas. Por otro, trabajamos con estándares, y separamos hasta cierto punto la estructura (el código) de la presentación, logrando una manera más clara de trabajar, todo esto en un sencillo documento CSS, al que podemos llamar "plantilla gráfica" para un sitio. Cabe señalar, que cualquier cambio hecho a un estilo CSS, se reflejará en todos los elementos que sean referidos a éste, automáticamente, con sólo editar un sencillo documento CSS.

2.5.1. Bootstrap

Bootstrap es el framework de Twitter que permite crear interfaces web con CSS y Javascript, además de crear diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales y adicionales.

Bootstrap fue desarrollado por Mark Otto y Jacbod Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia a través de herramientas internas. Antes de Bootstrap, se usaban varias librerías para el desarrollo de interfaces de usuario, las cuales guiaban a inconsistencias y a una carga de trabajo alta en su mantenimiento.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles), esto se denomina diseño adaptativo o Responsive Design.

Bootstrap es de código abierto y está disponible en GitHub. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

CAPÍTULO 3: ANÁLISIS Y DISEÑO

En este capítulo se muestra el análisis y diseño del sistema propuesto para este trabajo de tesis, el cual hace uso principalmente de las siguientes tecnologías: una Base de Datos, un servicio Web, un Framework, una metodología ágil, un CSS.

Dicho sistema, es un Sistema Web creador de exámenes para presentar certificaciones en Bases de Datos Relacionales, la propuesta para este trabajo de tesis, es crear un software que brinde un servicio educativo virtual enfocado principalmente al apoyo de los usuarios que requieran presentar un examen de certificación en Bases de Datos Relacionales.

A continuación se presenta cada etapa de desarrollo del sistema: Desde el listado de funciones a implementar, el diseño, creación y gestión de la Base de Datos, hasta la descripción del proceso evolutivo de dicho programa.

Para el presente trabajo se utilizará SCRUM, ya que tiene un desarrollo que requiere un trabajo duro pero no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto, además de que utiliza un modelo de proceso incremental.

Al comienzo de cada iteración (“sprint”) se determina que es lo que se va a elaborar, tomando como criterio la prioridad de las actividades y la cantidad de trabajo que se podrá abordar durante la iteración.

3.1. Escenario

En esta sección describiremos el entorno de los usuarios, el modo de trabajo, el equipo de trabajo, la estructura organizativa y el objetivo del producto final.

El Sistema debe ser terminado con un equipo de trabajo de dos personas. El desarrollo del proyecto se realiza con metodología ágil adaptable. Esto quiere decir que no siempre tenemos una línea a seguir.

Teniendo en cuenta limitaciones y costos, debemos decidir entre tomar unas decisiones u otras, y en caso de ser necesario resolver cualquier duda que surja en la evolución del proyecto.

En este esquema de trabajo es necesario establecer prioridades y proceder conforme a ellas, para saber cuánto tiempo se ha empleado en realizar una tarea, un sprint, una iteración, un proyecto, etcétera, donde lo más importante es conocer cuánto falta para llegar al objetivo.

3.2. Alcance

Para iniciar, debemos detectar los objetivos de negocio, es decir, objetivos necesarios para el cliente una vez que se le entregue el software. Estos vienen determinados por cada elemento que interviene en el proyecto, y siguiendo la definición de SCRUM, el cliente forma parte del esquema de trabajo, pero por ahora sólo nos interesa saber qué requisitos necesita el usuario para poder utilizar la aplicación.

También es necesario agrupar las tareas por historia de usuario, es decir, asignar las tareas adecuadas a los miembros del equipo de trabajo. Para que se pueda asignar una tarea tiene que estar completa, con tiempo estimado de realización. Debe haber una restricción de productividad, en donde se prohíbe tener varias tareas en ejecución. A medida de que las tareas se vayan completando, al integrante en turno se le van asignando otras tareas nuevas. Con esto debe ser posible obtener la duración real del proyecto.

Con los errores o tareas clasificadas con prioridad “urgente”, se decide que hay que hacer, la tarea se le asigna a un integrante del proyecto, este valorará el nivel de dificultad y tratará de abordarla de la mejor manera. En algunos casos será necesaria la intervención de más de un compañero.

Como ya hemos mencionado anteriormente, debemos conocer el avance y el tiempo de desarrollo del proyecto, incluyendo las tareas que están en ejecución, tomando en cuenta que lo más importante para el equipo de desarrollo es conocer qué falta por hacer.

Ante esta situación es importante documentar la información del proyecto. Conociendo estos datos podremos conocer el origen, el tiempo y toda la información relacionada con las tareas ejecutadas por el equipo de desarrollo. Se tiene claro que el cálculo del tiempo de trabajo es un requisito prioritario y aunque podemos interpretar el tiempo en horas, no para todo el mundo significa lo mismo una hora de su tiempo. En esta ocasión solo expresaremos el tiempo como horas de trabajo en general.

3.3. Roles dentro del Proyecto

- Product Owner: El encargado es el M.C. Carlos Armando Ríos Acevedo.

- Scrum Master: José Daniel Cortezano Mora (Responsable de moderar reuniones y de que las ayudas/problemas que se plantean se resuelvan a lo largo del periodo de trabajo).

- Team Member: compuesto por:

    - José Daniel Cortezano Mora

3.4. Presentación del Sistema

El proyecto tiene por objetivo crear un sistema que sea capaz de generar exámenes para certificaciones en Bases de Datos Relacionales, con la capacidad de poder crear reactivos de opción múltiple de manera manual, además de evaluar las respuestas de un examen realizado en dicho sistema y así obtener una calificación.

3.5. El Cliente

El cliente en este caso es cualquier Usuario que requiera utilizar el sistema Web y busque tener un seguimiento personal de su conocimiento.

3.6. Lista de funciones a implementar (Product Backlog)

A continuación detallamos el *Product Backlog,* conforme se vayan completando las historias de usuarios este podrá sufrir modificaciones en cada iteración.

El backlog es un inventario o una lista priorizada de requerimientos funcionales, tareas realizadas y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones. Siempre está en continuo crecimiento y evolución. Representa todo aquello que esperan los clientes, usuarios, y en general los interesados en el producto.

Todo lo que suponga un trabajo que debe realizar el equipo tiene que estar reflejado en el backlog.

Primeramente vamos a definir los requerimientos en bruto. Veremos que luego de la primera reunión surgirá la lista con ciertas características que la identifican.

Enunciamos entonces la lista (llamada Sprint 0) preparada por el Product Owner:

SPRINT 0

Requerimientos
El Sistema debe ser Web.
El Sistema debe tener un diseño accesible.
El Sistema debe ser realizado con software libre.
El acceso al sistema será a través de usuario y contraseña.
Diseñar la Base de Datos.
Registrar el ingreso de un nuevo usuario.
Registrar el ingreso de una nueva certificación, asignándole su número de curso.
Registrar para cada certificación la fecha de realización.
Registrar el ingreso de una nueva pregunta.
Registrar el ingreso de un nuevo examen.
Registrar para cada examen la fecha de realización.
Permitir Buscar un usuario que ya haya contestado un examen.
Permitir la eliminación de un usuario.
Permitir la eliminación de una certificación.
Permitir la modificación de datos de un usuario.
Permitir la modificación de datos de una certificación.
Permitir la modificación de datos de una pregunta
Registrar las respuestas de un examen contestado
Evaluar las respuestas contestadas en un examen.
Registrar una calificación con respecto a la evaluación del examen contestado.

Recordemos que el product backlog no es un documento de requisitos, sino una herramienta de referencia para el equipo. Sabemos que nuestro formato de lista debe incluir al menos la siguiente información para cada línea:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad.
- Campo o sistema de priorización.
- Estimación (En Horas de Trabajo).

Con esta información el Product Owner puede empezar a darle forma al listado.

ID	Descripción	Prioridad	Estimación (Horas)
	El Sistema debe ser Web.	Alta	24
	Investigar sobre tecnologías de desarrollo Web.		
	El Sistema debe tener un diseño accesible.	Alta	24
	El Sistema debe ser realizado con software libre.	Alta	24
	El acceso al sistema será a través de usuario y contraseña.	Alta	24
	Generar la Base de Datos.	Alta	56
	Registrar el ingreso de un nuevo usuario.	Alta	12
	Registrar el ingreso de una nueva certificación, asignándole su número de curso.	Media	6
	Registrar para cada certificación la fecha de realización.	Media	6
	Registrar el ingreso de una nueva pregunta.	Alta	12
	Registrar el ingreso de un nuevo examen.	Alta	24
	Registrar para cada examen la fecha de realización.	Media	6
	Permitir Buscar un usuario que ya haya contestado un examen.	Media	6
	Permitir la eliminación de un usuario.	Media	6
	Permitir la eliminación de una certificación.	Media	6
	Permitir la modificación de datos de un usuario.	Media	6
	Permitir la modificación de datos de una certificación.	Media	6

	Permitir la modificación de datos de una pregunta	Media	6
	Registrar las respuestas de un examen contestado	Alta	12
	Evaluar las respuestas contestadas en un examen.	Alta	12
	Registrar una calificación con respecto a la evaluación del examen contestado.	Alta	12

Con ayuda del Product Owner se le colocarán prioridades al listado y al finalizar será revisado nuevamente por el equipo de trabajo. El ID de los requerimientos (que luego se convertirán en tareas) surgirá en base a los objetivos del primer Sprint, es decir en la primera reunión (etapa de planeación), así se podrá decidir qué requisitos irán primero, cuales irán después y así sucesivamente, con esto ya podremos colocar el ID del mismo. La prioridad surge de la importancia de los requisitos, que como ya mencionamos antes, se determinará con la colaboración del Product Owner y con la ayuda del equipo. Luego la estimación surge en base a las apreciaciones y a la experiencia de los elementos del equipo de trabajo.

En lo que se refiere a la asignación de tareas, es muy importante identificar las fortalezas del equipo de desarrollo, que puede hacer y que no puede hacer una persona, es muy importante aclarar estos puntos en la primera reunión.

3.7. Sprint Planning Meeting y Complete Spring Backlog (Reunión de planeación por Sprint y Lista de Tareas por Sprint completo).

Desde el primer día que empezamos a trabajar en el proyecto, se hizo una reunión en la que estuvieron presentes el Product Owner y el desarrollador (Scrum Team), elementos que intervienen en el proyecto.

En esta reunión se eligieron Sprints con un plazo de tiempo de 15 (quince) días, todo esto en función del proyecto, tomando en cuenta necesidades y tiempos, aunque bien podría elegirse otro plazo de tiempo que se acomode a cada una de nuestras necesidades, por ejemplo una semana o dos semanas. El plazo de tiempo no debe ser muy largo, preferentemente menor a un mes. Una vez elegido este periodo de tiempo, se revisa el Product Backlog y analizamos los requisitos (llamados tareas) que se establecieron, empezando desde el primero.

Se realizan una serie de preguntas al Scrum Team: ¿La primera tarea puede estar hecha dentro de quince días?

Al examinar la pregunta, vemos si hace falta descomponer la tarea en subtareas, en caso de no ser así, se estima el tiempo que tardaríamos en resolverla y determinamos que "sí". En caso de que nuestra respuesta hubiera sido “no”, tendríamos que descomponerla en tareas más sencillas hasta que al menos una de ellas si pudiéramos llevarla a cabo.

Se toma la segunda tarea y se pregunta al Scrum Team: ¿Puede estar la primera y la segunda tarea en quince días? Analizamos la respuesta y respondemos que "sí".

Se repite el proceso con las siguientes tareas hasta que comencemos a dudar si se va a terminar con todo lo previsto. Si el Product Owner quiere que esté alguna tarea que no va a estar, la podemos cambiar por otra que sí esté, o "reducir" el alcance de una de las tareas que hemos registrado antes para que la otra tarea que queremos se pueda lograr. Podemos decir que este es un proceso de estimación de tiempos entre la programación y el análisis de las tareas, en donde buscamos responder si las tareas seleccionadas “Quedarán terminadas o no en 15 días”. El encargado del proyecto puede decidir el orden, intercambiar tareas,

modificarlas o dividirlas, pero el o los encargados de programar tienen la última palabra de cuánto tiempo necesitan (estimación) para cada tarea. Recordemos que el tiempo necesario para las tareas seleccionadas no puede ser mayor un mes.

Una vez llegado a un acuerdo, las tareas las pasamos a una nueva lista, donde quedará completo nuestro Sprint Backlog. Al final llegamos al acuerdo que dentro de 15 días vamos a entregar al Product Owner un avance significativo del proyecto que estamos realizando y nos aseguraremos que las tareas del Sprint Backlog que hayamos seleccionado estén funcionando.

Finalmente tenemos para el primer período (Sprint1) la siguiente lista de tarea:

SPRINT 1

ID	Descripción	Prioridad	Estimación (Horas)
S1.1	El Sistema debe ser Web.	Alta	24
S1.1.1	Investigar sobre tecnologías de desarrollo Web.		
S1.2	El Sistema debe tener un diseño accesible.	Alta	24
S1.2.1	Investigar sobre plantillas de diseño y hojas de estilo para aplicaciones Web.		
S1.3	El Sistema debe ser realizado con software libre.	Alta	24
S1.3.1	Investigar sobre tecnologías de software libre para aplicaciones Web.		
S1.4	El acceso al sistema será a través de usuario y contraseña.	Alta	24
S1.4.1	Implementar un método de Validación para entrada de datos.		

Definimos los Id's para las tareas, donde las tareas son subdivididas según la apreciación del equipo de trabajo, cada tarea tiene su prioridad y establecimos la estimación en horas de cada una de las tareas.

El Id lo identificamos de la siguiente manera:

SX.X1.X2.X3

SX= *Representa al término Sprint del cual estamos haciendo referencia (En nuestro caso es el Sprint 1, es decir, S1). En caso de que fuera el Sprint 2, el término sería S2 y así sucesivamente.*

X1 = *Representa al Nº de Tarea.*

X2 = *Representa al Nº de una Subtarea.*

X3 = *Representaría al Nº de una Subdivisión de una Subtarea (En caso de que existiera).*

Figura 3.7.1: Nomenclatura para identificar los Id's de las tareas a realizar

3.8. Scrum Meeting (Reuniones con el Scrum Master).

Después de la reunión en la que se decidió el Spring Backlog, es momento de trabajar. A partir de ese día, el Scrum Team se reunirá cada 15 días y cada uno contestará algunas preguntas, por ejemplo:

· ¿Qué has hecho hasta el momento?

· ¿Qué estás haciendo?

. ¿Qué vas a hacer?

· ¿Necesitas ayuda?

Estas reuniones no deben durar más de 15 minutos, ya que las cuestiones como los problemas y la ayuda que se necesite, se tienen que resolver a lo largo del

periodo establecido para trabajar. El Product Owner también deberá colaborar en lo que respecta a quitar obstáculos, uno de ellos debe ser resolver las dudas que surjan en el desarrollo del proyecto.

En estas reuniones aprovecharemos para volver a estimar el tiempo de las tareas en curso. Por ejemplo, si una tarea desde el primer día se dijo que se iba a tardar ocho horas, pero hoy, después de haber trabajado las ocho horas estimadas, sale un problema inesperado, debemos de calcular cuánto tiempo más tardaremos en resolverla y ajustar nuestro Spring Backlog con los nuevos tiempos. Lo único que debemos hacer es precisar el motivo del cambio en los tiempos y seguir trabajando para llegar a las metas establecidas.

Lo importante es mostrar al final del periodo de trabajo una versión con funcionalidades implementadas y no adelantarse ni retrasarse demasiado en las tareas de ese periodo.

3.9. Sprint Review y Sprint Retrospective

Al concluir el plazo de tiempo establecido, vemos que nuestra estimación ha sido buena, tenemos nuestra versión del programa con todas las funcionalidades del Sprint Backlog. Si bien hay algunas cosas por modificar, la primera versión cumple con todas las funcionalidades establecidas en el Spring Backlog.

Con esto, debemos hacer otra reunión de aproximadamente un par de horas, llamada Sprint Review, en esta reunión estará presente el Scrum Team, principalmente el Scrum Master y el Product Owner. Es el momento de que el Scrum Master enseñe la primera versión del sistema.

El objetivo primordial es que se recolecten opiniones, posibles mejoras o aprobar que se ha realizado bien el trabajo. Inmediatamente después, haremos otra reunión, llamada Sprint Retrospective, en el que analizaremos qué cosas pueden mejorarse a la hora de trabajar para el siguiente Sprint, si la comunicación ha sido buena o debe mejorarse y si hay algún problema que deba arreglarse. En general,

Sprint Retrospective nos ayuda a ver cómo se puede mejorar la forma de trabajo en el Sprint Backlog del siguiente periodo.

Con esto, volvemos a empezar; realizaremos otro Sprint Planning Meeting para ver qué tareas son las siguientes o cuáles hacen falta para el Sprint que viene, se platicará con el equipo de trabajo y se realizará un nuevo Sprint Backlog.

Vemos a continuación como quedo nuestro segundo Sprint:

SPRINT 2

ID	Descripción	Prioridad	Estimación (Horas)
S2.1	Crear la Base de Datos.	Alta	56
S2.1.1	Diseñar la Base de Datos.		40
S2.1.2	Implementar la Base de Datos.		16
S2.2	Registrar el ingreso de un nuevo usuario.	Alta	14
S2.2.1	Permitir la eliminación de un usuario.		
S2.2.2	Permitir la modificación de datos de un usuario.		

Seguimos el procedimiento ya mencionado antes y realizaremos otro Sprint Planning Meeting, identificaremos que tareas nos hace falta terminar, y en caso de que hayas dudas, se resolverán, así en el siguiente Sprint no habrá retrasos en los tiempos estimados.

Este es el modelo Relacional de la Base de Datos que fue creada:

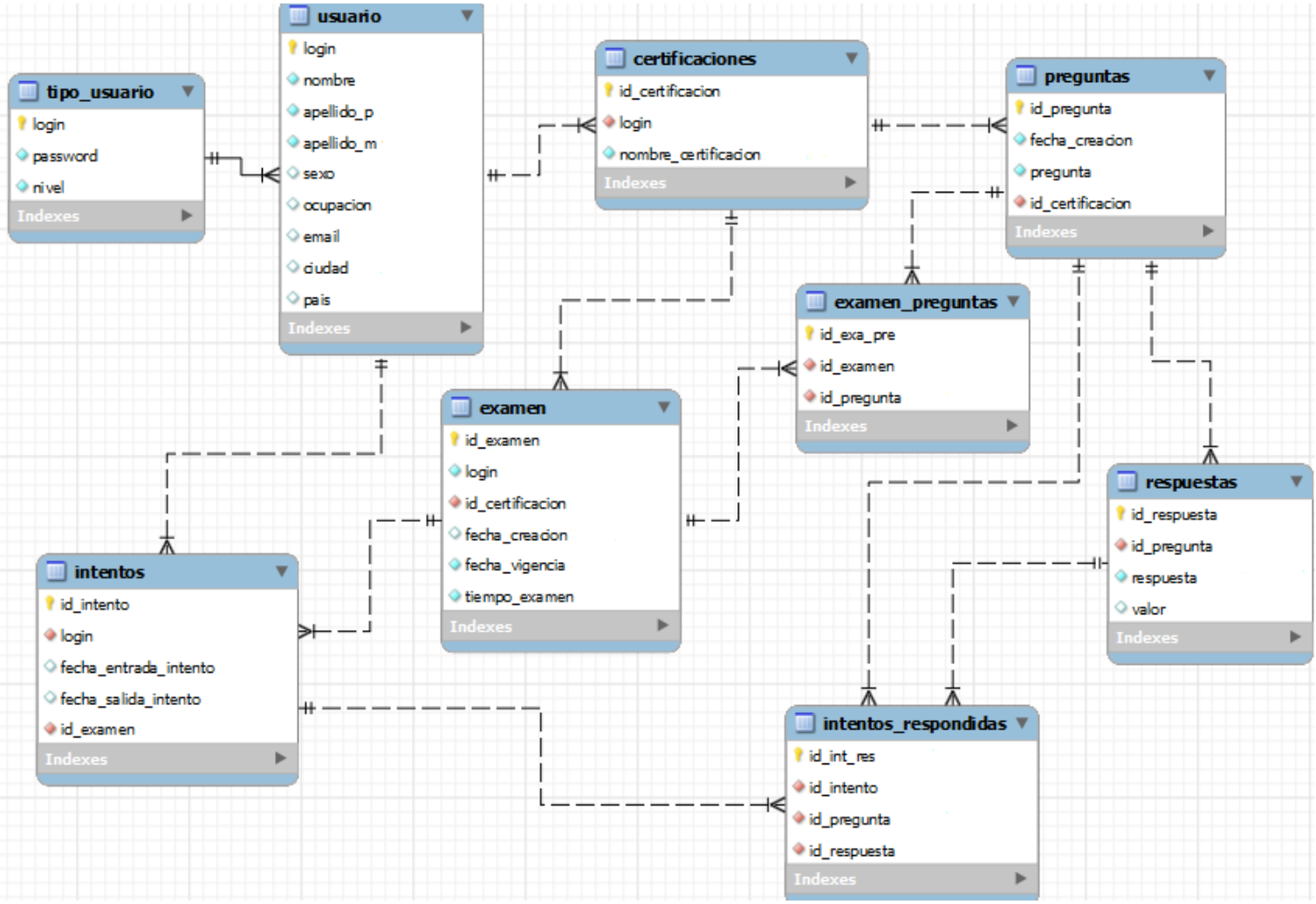

Figura 3.9.1: Modelo Relacional de la Base de Datos del Sistema

Después de haber platicado con el equipo, es momento de realizar nuestro último Sprint Backlog, que queda de la siguiente manera:

SPRINT 3

ID	Descripción	Prioridad	Estimación (Horas)
S3.1	Registrar el ingreso de una nueva certificación, asignándole su número de curso.	Alta	12
S3.1.1	Registrar para cada certificación la fecha de realización.		
S3.1.2	Permitir la modificación de datos de una certificación.		
S3.1.3	Permitir la eliminación de una certificación.		
S3.2	Registrar el ingreso de una nueva pregunta.	Media	8
S3.2.1	Permitir la modificación de datos de una pregunta		
S3.3	Registrar el ingreso de un nuevo examen.	Alta	24
S3.3.1	Registrar para cada examen la fecha de realización.		
S3.4	Registrar las respuestas de un examen contestado	Alta	12
S3.5	Evaluar las respuestas contestadas en un examen.	Alta	14
S3.5.1	Registrar una calificación con respecto a la evaluación del examen contestado.		
S3.6	Permitir Buscar un usuario que ya haya contestado un examen.	Media	8

En cada Sprint el equipo va mostrando los resultados que se van consiguiendo. No está permitido trabajar sin exponer los avances que se van obteniendo.

Estas iteraciones pueden ocurrir tantas veces sea necesario, el objetivo es que el Sistema quede terminado cumpliendo con los requisitos que se pidieron. En caso de que haya tiempo de sobra, puede existir la posibilidad de realizar algunas mejorías.

3.10. Resultados

Como resultado de una correcta utilización de la metodología ágil SCRUM pudimos desarrollar exitosamente el proyecto establecido.

El producto final es un Sistema Web que es capaz de generar exámenes para certificaciones, en él se pueden crear reactivos de opción múltiple de manera manual, donde además se pueden evaluar las respuestas de un examen contestado y así obtener una calificación.

En el “Sprint 0” se levantó una tabla de requerimientos de acuerdo a lo que el Product Owner esperaba obtener del software a desarrollar, analizando estos requerimientos, se planificó utilizar MVC (Modelo, Vista, Controlador), que es una forma de trabajar que separa los datos de una aplicación, la interfaz de usuario y la lógica de programación, es decir, los dividimos en tres módulos que fueron resueltos en los Sprints antes mencionados.

El siguiente gráfico (Figura 3.10.1) muestra la agrupación de los tres elementos del MVC, aplicados en nuestro sistema:

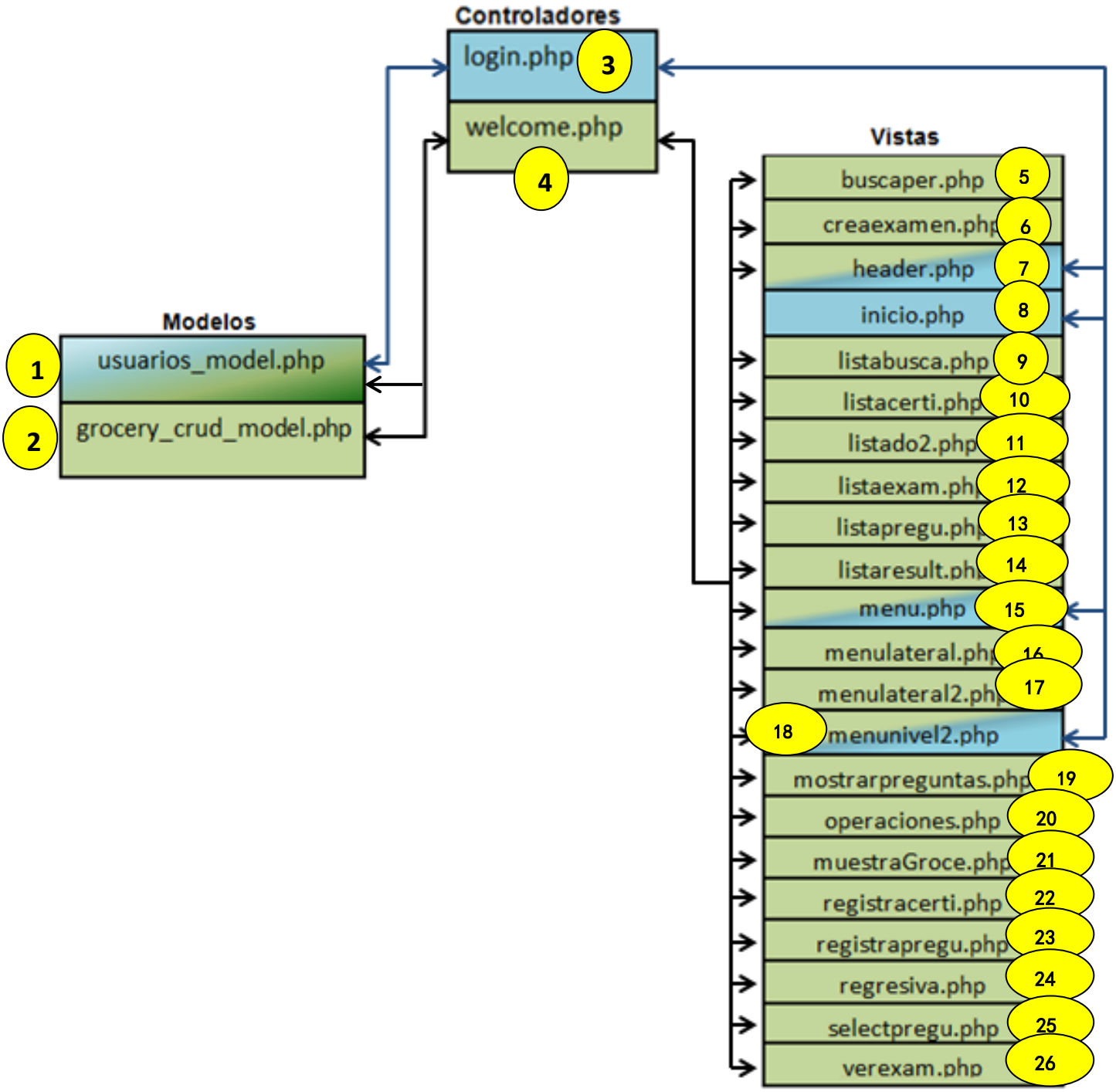

Figura 3.10.1: Representación de los componentes del MVC utilizados en el sistema

3.10.1. Descripción de los elementos del MVC (Modelo-Vista-Controlador).

<table>
<tr><td>Elemento Número:</td><td>1</td><td>Nombre de archivo:</td><td>usuarios_model.php</td></tr>
<tr><td>Tipo de elemento:</td><td><u>MODELO</u></td><td>VISTA</td><td>CONTROLADOR</td></tr>
<tr><td>Descripción:</td><td colspan="3">Es uno de nuestros dos Modelos, se encarga de la manipulación de los datos de nuestro sistema. Guarda y recupera la información de los elementos que utilizamos. Es el Modelo principal, el funcionamiento correcto del Sistema es en mayor parte gracias a él.</td></tr>
<tr><td>Funciones Principales:</td><td colspan="3">Entre las funciones principales de este modelo se encuentran las siguientes:
Verificar que un registro exista la Base de Datos (solo para validar información, como lo es el nombre de usuario o login y su contraseña) y que devuelva verdadero o falso de acuerdo al resultado.
Guardar los datos de un usuario que se registre por primera vez o que actualice su información personal.
Guardar información de nuevos registros, como pueden ser certificaciones, preguntas, respuestas y exámenes, principalmente.
Recuperar un solo registro de una tabla, por ejemplo un id, un nombre o un nivel de usuario).
Recuperar todos los elementos existentes en una tabla, es decir, todos los registros que se tengan. Por ejemplo todos los exámenes existentes en el sistema o todos los usuarios que han contestado algún examen sobre determinada certificación.
Recuperar solo algunos registros de varias tablas, por ejemplo, el nombre de usuario, el examen que ha contestado y la calificación que ha obtenido.
Eliminar la información de un registro existente, por ejemplo el nombre de una certificación o una pregunta.</td></tr>
</table>

Elemento Número:	**2**	**Nombre de archivo:**	grocery_crud_model.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Es el segundo de nuestros dos modelos, es un archivo necesario para el funcionamiento de Grocery CRUD. Grocery CRUD es una librería que nos hace más sencilla la manera de programar, unas líneas de código indicando las tablas de nuestra Base de Datos, las relaciones entre ellas y si todo funciona correctamente tendremos acceso a la interfaz adecuada para administrar esos datos. Este modelo junto a la librería Grocery CRUD, proporcionan todos los elementos necesarios para añadir, mostrar y editar los datos de una tabla, en nuestro caso solo lo utilizamos para la tabla certificaciones.		
Funciones Principales:	Las funciones principales de este modelo son las siguientes: • Recuperar todos los registros disponibles de la tabla certificaciones. • Añadir registros nuevos en la tabla certificaciones. • Guardar los registros actualizados de la tabla certificaciones. • Eliminar un registro existente, por ejemplo el nombre de una certificación • Ordenar por nombre los registros disponibles en la tabla certificaciones. • Verificar puntos importantes en la conexión de la librería Grocery CRUD con la Base de Datos, por ejemplo, que la Base de Datos exista, que exista la llave primaria y que sea única, que existan registros en la tabla certificaciones, entre otros.		

Elemento Número:	3	**Nombre de archivo:**	login.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Es uno de nuestros dos controladores, este controlador comunica datos de ida y vuelta entre el modelo usuarios_model y algunas vistas. Cabe señalar que dicho controlador solamente se encarga de dos cosas, una de ellas es la validación de la información de los usuarios que ingresen al sistema, es decir, únicamente determina si un usuario puede o no ingresar al sistema. La otra es cerrar la sesión de un usuario que salga del sistema. Aunque este no es nuestro controlador principal, es muy importante para el funcionamiento del sistema. En general este controlador solicita al modelo la información de un usuario que desea "loguearse". Si todo sale bien recibirá una respuesta, ya sea positiva o negativa, con esto el controlador podrá mandar la información que desea mostrar en pantalla a una vista y ésta mostrará la salida correspondiente.		
Funciones Principales:	Las funciones principales de este controlador son: • Validar la información de los usuarios que ingresen al sistema. • Transmitir información entre el modelo usuarios_model y algunas vistas, esto con el fin de mostrar los resultados de dicha validación. • Mostrar las vistas adecuadas para cualquier resultado obtenido. • Cerrar la sesión de un usuario que salga del sistema.		

Elemento Número:	**4**	**Nombre de archivo:**	welcome.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Es el controlador principal del sistema, este controlador se encarga de la comunicación de datos entre los dos modelos (usuarios_model y grocery_crud_model) y la mayoría de las vistas. Decimos que es nuestro controlador principal porque la mayoría de las funciones que puede realizar el sistema, están implementadas y organizadas en el. Es necesario declarar las librerías necesarias para que todas las funciones trabajen correctamente, los helpers (que nos ayudan a la hora de programar para escribir menos líneas de código) y muy importante, el modelo usuarios_model, si no hubiéramos declarado alguno de los elementos mencionados, es muy probable que el funcionamiento no hubiera sido el adecuado. En general, casi todo el funcionamiento del sistema depende en gran parte a este controlador, este controlador llama a las vistas que se desean mostrar, según sea el caso; si es necesaria alguna información que ingrese el usuario en alguna de las pantallas, el controlador la obtiene y la manipula de acuerdo a lo que se requiera. En algunos casos esta información será enviada al modelo usuarios_model y de ahí se devolverá una respuesta positiva o negativa (por ejemplo una lista de usuarios, o el id de una certificación) que podría volver a mostrarse en otra vista.		
Funciones Principales:	Al ser el controlador principal, tiene muchas funciones, pero entre las principales se encuentran las siguientes: • Llamar a las vistas necesarias para se puedan mostrar encabezados, la hoja de estilo de la página, el menú principal y los menús laterales de cada una de las pantallas que aparecen en el sistema.		

	• Llamar a las vistas adecuadas en cada función declarada. • Enlazarse con la librería Grocery Crud, con el modelo Grocery_crud_model y llamar una vista que muestre la información de la tabla certificaciones pero al estilo de la librería Grocery Crud. • Llamar a una vista que registre un usuario nuevo y mande dicha información a un modelo para que sea guardada. • Llamar a una vista que liste a todos los usuarios registrados en la Base de Datos. • Llamar a una vista que modifique los datos personales de un usuario. • Llamar a una vista que registre una Certificación y mande la información a un modelo para que sea guardada. • Llamar a una vista que liste todas las certificaciones disponibles. • Llamar a una vista que registre una Pregunta y mande la información a un modelo para que sea guardada. • Llamar a una vista que cree un Examen y mande toda la información introducida en la pantalla a un modelo para que sea guardada. • Llamar a una vista que liste todos los exámenes disponibles. • Llamar a una vista que muestre un examen (Las Preguntas con sus respuestas). • Llamar a una vista que muestre los resultados de un examen que haya sido resuelto por un usuario. • Llamar a una vista que haga la búsqueda de un usuario por nombre y apellidos.

Elemento Número:	**5**	**Nombre de archivo:**	buscaper.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Es una de nuestras vistas que muestra al usuario una pantalla de búsqueda de un usuario. De esta pantalla se va de vuelta al controlador y éste hace el llamado a una vista final para mostrar los resultados obtenidos.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar una pantalla de búsqueda de un usuario por medio de su nombre y sus apellidos, donde los datos deben ser ingresados en un cuadro de texto.		

Elemento Número:	**6**	**Nombre de archivo:**	creaexamen.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En esta vista se muestra al usuario una pantalla con todas las certificaciones disponibles, con la opción de seleccionar una de ellas para crear un examen. De esta vista se devuelve al controlador un id de la certificación seleccionada y éste a su vez hace el llamado a otra vista para poder cumplir con la función de crear un examen.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar una pantalla con las certificaciones disponibles y a cada una de ellas asignarles un link de direccionamiento hacia la pantalla que crea un examen.		

Elemento Número:	**7**	**Nombre de archivo:**	header.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Es en este archivo donde declaramos la cabecera útil para todo el sistema. Es decir, agregamos encabezados que utilizaran otros archivos para que puedan funcionar correctamente. Como primer lugar incluimos en la vista los archivos “css” y los “js”, aunque si quisiéramos, también podríamos definir un título o dar información sobre fechas importantes, versión del sistema, o algún enlace de navegación dentro del sistema. Recordemos que los archivos “css” y los “js” en conjunto, nos ayudan a organizar la presentación y a mejorar el aspecto de una página web. Gracias a esta vista podemos definir el "estilo visual" de nuestro sitio Web sin necesidad de hacerlo etiqueta por etiqueta. Además si necesitamos agregar más hojas de estilos o archivos JavaScript, tendremos que modificar un solo archivo.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Declarar todos los archivos “css” y los “js” que se utilizaran en todo el sistema para que pueda funcionar correctamente. En caso de ser necesario se declarará un título o algunos datos informativos, pero eso ya dependerá de las preferencias de los usuarios que administren el sistema.		

Elemento Número:	**8**	**Nombre de archivo:**	inicio.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista es muy importante, ya que muestra al usuario la pantalla de entrada al sistema. Con solo dos cajas de texto, el usuario ingresa en una de ellas su login, en la otra caja de texto ingresa su contraseña y con ayuda de la Clase Form Validation que CodeIgniter provee, se evalúa que se introduzcan los dos campos necesarios para acceder al sistema. En caso de no ingresar alguno de ellos, la clase muestra un mensaje de error. Suponiendo que una persona accede por primera vez al sistema, esta pantalla muestra un link con el cual se podrá dirigir a otra vista para realizar el registro de un usuario nuevo.		
Funciones Principales:	La función principal de esta vista es: • Mostrar una pantalla de acceso de un usuario al sistema, en donde se ingresa el nombre de usuario y su contraseña. • Mostrar un link de direccionamiento para el registro de un usuario que ha ingresado por primera vez al sistema.		

Elemento Número:	**9**	**Nombre de archivo:**	listabusca.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En caso de hacer la búsqueda de un usuario y dicha búsqueda sea exitosa, solo en ese caso, esta vista imprimirá en pantalla el resultado obtenido, es decir, mostrará una lista del usuario que encontró en la Base de Datos con las certificaciones en las que ha hecho uno o más exámenes.		
Funciones Principales:	La función principal de esta vista es: • Después de realizar la búsqueda exitosa de un usuario, imprimir en pantalla el nombre de la persona encontrada (nombres y apellidos) con las certificaciones en las que ha hecho uno o más exámenes.		

Elemento Número:	**10**	**Nombre de archivo:**	listacerti.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En esta vista se imprime en pantalla una lista con las certificaciones disponibles y cada una de ellas muestra: un id que identifica a la certificación, el login del usuario que creó dicha certificación, el nombre de la certificación, un link para eliminarla y otro para modificar su nombre.		
Funciones Principales:	La función principal de esta vista es: • Después de realizar la búsqueda exitosa de un usuario, imprimir en pantalla el nombre de la persona encontrada (nombres y apellidos) con las certificaciones en las que ha hecho uno o más exámenes.		

Elemento Número:	11	Nombre de archivo:	listado2.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista imprime en pantalla una lista de los usuarios que han hecho un examen para certificación. Dicha lista incluye: Su nombre, apellidos paterno y materno y las certificaciones en la que ha hecho examen.		
Funciones Principales:	La función principal de esta vista es: • Imprimir en pantalla una lista de los usuarios que han hecho un examen para certificación con su nombre, apellido paterno, apellido materno y las certificaciones en la que ha hecho examen		

Elemento Número:	12	Nombre de archivo:	listaexam.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista imprime en pantalla una lista de todos los exámenes que han sido creados por un administrador. Dicha lista incluye: El código de identificación del examen, el login de la persona que creó el examen, el nombre de la certificación a la que pertenece el examen y la fecha de creación del examen, además de que cada examen cuenta con un link que direcciona a otra vista, misma que permite ver el examen con sus preguntas, respuestas y el tiempo disponible para responderlo.		
Funciones Principales:	La función principal de esta vista es: • Imprimir en pantalla una lista de todos los exámenes que han sido creados por un administrador con el		

	código de identificación del examen, el login de la persona que creó el examen, el nombre de la certificación a la que pertenece el examen y la fecha de creación del examen, además de que cada examen tiene un link que al presionarlo permite ver el examen con sus preguntas, respuestas y tiempo disponible para resolverlo.

Elemento Número:	**13**	**Nombre de archivo:**	listapregu.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista imprime en pantalla una lista de todas las preguntas disponibles de cada certificación. Es necesario enfatizar que las preguntas que muestra están catalogadas por el nombre de certificación a la que pertenece. Esta vista precede de otra que muestra todas las certificaciones disponibles y que permite seleccionar alguna de ellas para que muestre las preguntas que han sido registradas en dicha certificación.		
Funciones Principales:	La función principal de esta vista es: • Imprimir en pantalla una lista de las preguntas existentes en cada certificación, donde en una vista anterior ya ha sido seleccionada una certificación para que puedan ser mostradas dichas preguntas.		

Elemento Número:	**14**	**Nombre de archivo:**	listaresult.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista imprime en pantalla una lista con los resultados de el (los) examen (exámenes) contestado (s) por un usuario. Cabe señalar que los resultados solo los puede ver un administrador del sistema o el propio usuario que esté utilizando el sistema en ese momento y tenga su sesión activa. Dicha lista contiene los siguientes elementos: el login del usuario que respondió el examen, el código de identificación del examen, la certificación en la que contesto el examen y la calificación que obtuvo al ser evaluadas sus respuestas por el propio sistema.		
Funciones Principales:	La función principal de esta vista es: • Imprimir en pantalla una lista con los resultados de el (los) examen (exámenes) contestado (s) por un usuario, dicha lista contiene: El login del usuario que respondió el examen, el código de identificación del examen, la certificación en la que contesto el examen y la calificación que obtuvo.		

Elemento Número:	**15**	**Nombre de archivo:**	menu.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista muestra en pantalla el menú principal de nuestro sistema. En cada opción del menú, nos podemos dirigir hacia otra vista, es decir, cada opción nos dirige a las funciones principales del sistema. Es importante aclarar que este menú solo es visible para los usuarios que son administradores del		

	sistema, ya que cuenta con opciones como: mostrar exámenes, mostrar preguntas, buscar un usuario, listar personas que estén registradas, entre otras.
Funciones Principales:	La función principal de esta vista es: • Mostrar en pantalla el menú principal del sistema, que cuenta con opciones como son: mostrar exámenes, mostrar preguntas, buscar un usuario y listar personas que estén registradas, las más importantes.

Elemento Número:	**16**	**Nombre de archivo:**	menulateral.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista muestra en pantalla uno de los dos menús laterales del sistema. Además del menú principal que aparece en la parte alta del sistema, podemos ver este otro que se sitúa en una sección lateral de nuestra izquierda. Dicho menú solo es visible después de seleccionar la opción “mostrar certificaciones” del menú principal. Dentro de las opciones con las que cuenta podemos distinguir las siguientes: Añadir una certificación (Ingresar el nombre de la certificación) y mostrar las certificaciones haciendo uso de la librería Grocery Crud. Cada una de ellas nos dirige a otra vista que nos permite visualizar los resultados a los que hace referencia cada opción.		
Funciones Principales:	La función principal de esta vista es: • Mostrar en pantalla un menú lateral que es visible después de seleccionar la opción “mostrar certificaciones” del menú principal del sistema. En el podemos ver las siguientes opciones: Añadir una certificación y mostrar certificaciones haciendo uso de la librería Grocery Crud.		

Elemento Número:	17	Nombre de archivo:	menulateral2.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista muestra en pantalla el segundo de los dos menús laterales. Este también se sitúa en un espacio lateral de nuestra izquierda, con la diferencia de que es visible después de seleccionar la opción “Mostrar exámenes” del menú principal, además de que solo se utiliza cuando los usuarios que ingresan al sistema son del tipo administrador. Las opciones que tiene este menú son las siguientes: Añadir un examen (Que nos permite ver las certificaciones en las que podemos crear un examen) y ver los resultados de los exámenes que se han resuelto. Cada una de ellas nos dirige a otra vista que nos permite visualizar los resultados a los que hace referencia cada opción.		
Funciones Principales:	La función principal de esta vista es: • Mostrar en pantalla un menú lateral que es visible después de seleccionar la opción “Mostrar exámenes” del menú principal del sistema. En el que tenemos las opciones: Añadir un examen y ver los resultados de mis exámenes.		

Elemento Número:	18	Nombre de archivo:	menunivel2.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista muestra el menú principal para los usuarios que no son administradores, es decir, los usuarios de nivel “2”. Cada opción del menú, nos dirige hacia otra vista, esto con el fin de cumplir con las funciones que se indican. Al ser una vista para		

	usuarios de nivel 2, el menú se limita a tener las siguientes opciones: mostrar exámenes disponibles, ver mis resultados y cerrar sesión.
Funciones Principales:	La función principal de esta vista es: • Mostrar el menú principal del sistema para los usuarios que no son administradores, es decir, de nivel 2. Este tiene las siguientes opciones: Mostrar exámenes disponibles, ver mis resultados y cerrar sesión.

Elemento Número:	**19**	**Nombre de archivo:**	mostrarpreguntas.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En esta vista se muestra al usuario una pantalla con todas las certificaciones disponibles, con la opción de seleccionar una de ellas para ver las preguntas que han sido registradas en dicha certificación. De esta vista se devuelve al controlador un id de la certificación seleccionada y éste a su vez hace el llamado a otra vista para poder cumplir con la función de ver las preguntas registradas.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar una pantalla con las certificaciones disponibles y a cada una de ellas asignarles un link de direccionamiento hacia la pantalla que muestra las preguntas registradas en dicha certificación.		

Elemento Número:	**20**	**Nombre de archivo:**	operaciones.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista es una de las más importantes de nuestro programa, ya que muestra la pantalla que permite el registro de un usuario nuevo en el sistema. Después de haber entrado a la pantalla principal, donde debemos ingresar el login y password, tenemos algunos casos, por ejemplo cuando podemos acceder a los servicios que ofrece el sistema. Otro sería cuando no estamos registrados y necesitamos elaborar dicho registro para tener acceso a los ya mencionados servicios del Sistema. Es en este caso cuando podemos ver la pantalla a la que hacemos referencia. La vista "operaciones.php" nos muestra todos los elementos necesarios para el registro de un usuario nuevo en el sistema. Con una combinación de cajas para introducir texto, menús desplegables de selección y botones, entre otros, un usuario nuevo puede llevar a cabo su registro para acceder y utilizar los recursos del Sistema. Dicha información ingresada y seleccionada en la vista es guardada en la Base de Datos para que pueda ser manipulada por el Sistema según se requiera.		
Funciones Principales:	La función principal de esta vista es: • Mostrar todos los elementos necesarios para el registro de un usuario nuevo en el sistema, por ejemplo, cajas para introducir texto, menús desplegables de selección y botones, entre otros. • Obtener el login del Usuario, su contraseña, nombres, apellidos, ocupación, ciudad y algunos datos más que serán guardados en la Base de Datos del programa.		

Elemento Número:	**21**	**Nombre de archivo:**	muestraGroce.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista recibe un archivo con la lista de todas las certificaciones disponibles en la Base de Datos. Teniendo este archivo y apoyado con un estilo para el texto de la información obtenida, podemos imprimir en pantalla una lista con las certificaciones disponibles y cada una de ellas con los siguientes links asociados: ver información de la certificación, editar dicha información o eliminarla. Además de que apoyándonos en la librería de Grocery Crud, podemos ver en esta vista enlaces como: Exportar la lista de certificaciones a una hoja de Excel, imprimir dicha lista o si se desea agregar el nombre de una certificación nueva.		
Funciones Principales:	La función principal de esta vista es: • Después de recibir un archivo con la lista de todas las certificaciones registradas en la Base de Datos y apoyado con un estilo para el texto de la información obtenida, imprimir una lista con las certificaciones disponibles y cada una de ellas con los siguientes link: Ver información de la certificación, editar dicha información o eliminarla. Y con apoyo de la librería de Grocery Crud, podemos ver enlaces como: Exportar la lista de certificaciones a una hoja de Excel, imprimirla o agregar una nueva certificación.		

Elemento Número:	22	Nombre de archivo:	registracerti.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En esta vista se muestra al usuario una pantalla que tiene solo un cuadro para ingresar texto y un botón. En él cuadro de entrada de texto se introduce el nombre de la certificación que se desea registrar y con el botón se envía dicha información a la Base de Datos donde es guardada.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar una pantalla con un cuadro para ingresar texto y un botón, mismo que sirve para enviar y guardar en la Base de Datos la información introducida.		

Elemento Número:	23	Nombre de archivo:	registrapregu.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En esta vista se muestra una pantalla que permite al usuario ingresar en cuadros de texto, la pregunta que desea registrar en el sistema, con sus respectivas respuestas (Correctas e incorrectas). Después de esto, con ayuda de un botón se envía dicha información a la Base de Datos donde es guardada.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar una pantalla con cuadros para ingresar texto (Donde se escribe la pregunta con sus respuestas) y un botón que sirve para enviar y guardar en la Base de Datos la información escrita.		

Elemento Número:	24	Nombre de archivo:	regresiva.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	En una combinación de PHP con JavaScript, este archivo recibe tres variables: “horas”, “minutos” y “segundos”, estas nos sirven para determinar el tiempo con el que se cuenta para resolver un examen. Ya que tenemos definido el tiempo, es en esta vista en donde se imprime en pantalla el tiempo que transcurre después de abrir un examen, o lo que es lo mismo, se imprime en pantalla la cuenta regresiva del tiempo que ha sido asignado para ser resuelto cada uno de los exámenes.		
Funciones Principales:	La función principal de esta vista es la siguiente: • Mostrar un reloj con la cuenta regresiva del tiempo que ha transcurrido según lo asignado para cada examen e imprimir un mensaje cuando el tiempo haya finalizado.		

Elemento Número:	25	Nombre de archivo:	selectpregu.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista es indispensable para conseguir el óptimo funcionamiento del Sistema. La importancia de esta radica en el cumplimiento del objetivo principal del Proyecto a desarrollar, hablamos de crear un Sistema que sea capaz de elaborar exámenes para certificaciones. Si esta vista no hubiese sido creada, el objetivo no estaría cumplido. Es en este archivo donde un usuario de tipo administrador puede		

	crear un examen para que sea resuelto por los usuarios de nivel “2”. En esta pantalla se muestran todos los requerimientos necesarios para poder elaborar un examen, con opciones para poder elegir el número de preguntas que tendrá, se puede seleccionar que preguntas deseo agregar al examen, la vigencia o fecha límite en la que se podrá contestar y el tiempo para responderlo. Dicha información ingresada y seleccionada en la vista es guardada en la Base de Datos para que pueda ser utilizada por el Sistema al mostrar los exámenes que serán contestados.
Funciones Principales:	La función principal de esta vista es: • Mostrar una pantalla con todos los requerimientos necesarios para poder elaborar un examen, con opciones para poder elegir el número de preguntas que tendrá, con la capacidad de seleccionar que preguntas deseo agregar al examen, la vigencia o fecha límite en la que se podrá contestar y el tiempo para responderlo. Al enviar la información ya habremos creado nuestro examen que podrá ser contestado por usuarios de nivel “2”.

Elemento Número:	**26**	**Nombre de archivo:**	verexam.php
Tipo de elemento:	MODELO	VISTA	CONTROLADOR
Descripción:	Esta vista nos permite ver en pantalla el examen que se va a contestar. Es de vital importancia el funcionamiento de la misma. Así como decíamos que la vista que crea un examen es muy importante, no podemos dejar atrás a esta que nos permite ver el examen a responder. Este archivo muestra en pantalla todos los elementos de un examen de opción múltiple. Independientemente de la		

	selección que haya hecho el usuario en la vista que aparece antes que esta, se podrá ver en pantalla: El nombre de la Certificación a la que pertenece el examen seleccionado, el login del usuario que lo creó, las preguntas que fueron escogidas al momento de crearlo con sus respectivas respuestas (verdadera y falsas) y un botón para enviar los resultados. Con esto, el sistema podrá evaluar las respuestas contestadas e imprimir como resultado una calificación.
Funciones Principales:	La función principal de esta vista es: • Mostrar en pantalla todos los elementos de un examen de opción múltiple. El nombre de la Certificación a la que pertenece el examen seleccionado, el login del usuario que lo creó, las preguntas que fueron escogidas al momento de crearlo con sus respectivas respuestas (verdadera y falsas) y un botón para enviar los resultados.

En resumen, con la división en los tres módulos antes mencionados (Modelos, vistas y controladores), el proyecto consta de:

- 2 modelos que manejan los datos de la aplicación, que recuperan y guardan la información que utilizamos.

- 22 vistas que podemos visualizar en las interfaces de usuario del sistema, es decir, que muestran la información obtenida en el modelo.

- 2 controladores que comunican los datos entre el modelo y la vista, ya que toda la información solicitada al modelo, la procesará en caso de ser necesario y la mostrará en la o las vistas necesarias.

CAPÍTULO 4: PRUEBAS DEL SISTEMA

En el presente capítulo se explican las pruebas aplicadas al sistema y se muestran los aspectos principales de la implementación del diseño antes especificado, revisando que el Sitio Web desarrollado incluya todos los contenidos que se han definido en el plan de desarrollo y mostrando las interfaces más importantes del sistema.

En general, se llevará a cabo una revisión de algunos aspectos importantes como son:

- Verificación de ortografía y redacción.
- Verificación de enlaces principales.
- Verificación de imágenes en páginas.
- Detección de enlaces rotos.

Con esto se buscará que el sistema proporcione una interfaz accesible a los usuarios, que podrán tener una idea clara del funcionamiento del sitio.

4.1. Tipos de pruebas aplicadas al sistema

Las pruebas aplicadas al sistema son necesarias para encontrar errores en determinadas situaciones, en donde algo que debe suceder no sucede y viceversa.

Las pruebas que se aplicaron al software fueron las siguientes:

- **Prueba de la caja negra.** Se consideró aplicar este tipo de prueba ya que esta no considera a la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Haciendo uso de la prueba explicada anteriormente, se definieron tres rubros en base a los cuales se definen los puntos a evaluar. Estos rubros son los siguientes:

- **Prueba de contenido.** Con esta prueba como su nombre lo indica, buscamos verificar que el contenido del sistema sea coherente y consistente a la vez. También se buscó verificar que las palabras usadas para transmitir una idea al usuario fueran las adecuadas.

- **Pruebas de Funcionalidad:** Con este tipo de pruebas examinamos si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas verificamos si el sistema lleva a cabo correctamente todas las funciones requeridas, un ejemplo sería la validación de datos al ingresar al sistema. Estas pruebas están enfocadas a condiciones planeadas de error y de exploración. Aquí es donde decimos que son pruebas de caja negra, ya que nos interesa saber si funciona bien o no, independientemente de la codificación del programa.

- **Pruebas de Usabilidad:** Las pruebas realizadas en este rubro tuvieron la finalidad de verificar que tan fácil de usar es el sistema. Estas pruebas verificaron: *Aprendizaje* (Qué tan fácil es para los usuarios realizar tareas básicas la primera vez que tienen contacto con el sistema), *eficiencia* (una vez que los usuarios han aprendido algo del sistema, que tan rápido pueden llevar a cabo las tareas), *manejo de errores* (cuántos errores comete el usuario, que tan graves son éstos y que tan fácil es para el usuario recuperarse de ellos) y *grado de satisfacción* (que tan satisfactorio es usar el sistema). Para la obtención de resultados realistas en este tipo de pruebas, fue importante dejar que las personas que probaron el sistema resolvieran los problemas que se les presentaron por sí mismos, ya que si uno les ayudaba, las pruebas no hubieran arrojado resultados verídicos.

4.2. Acceso al sistema

En esta sección se muestran las interfaces de todos los elementos que intervienen en el procedimiento para acceder al sistema.

4.2.1. Página de inicio

Las figuras 4.2.1.1, 4.2.1.2 y 4.2.1.3 forman parte de la pantalla de inicio que se muestra a todos los usuarios. En conjunto, las tres figuras conforman un slider (rotador o carrusel de imágenes) que presenta una ilustración por cada una de las diapositivas con las que cuenta dicha página de inicio. Esta es la pantalla de entrada a la que tiene acceso cualquier persona, sin importar el rol de usuario que tenga.

Como se puede observar en la parte inferior de cada una de las imágenes, hay un link (“Ir al sistema”) que nos dirige a la página de login. En la siguiente sección explicaremos las características de dicha página.

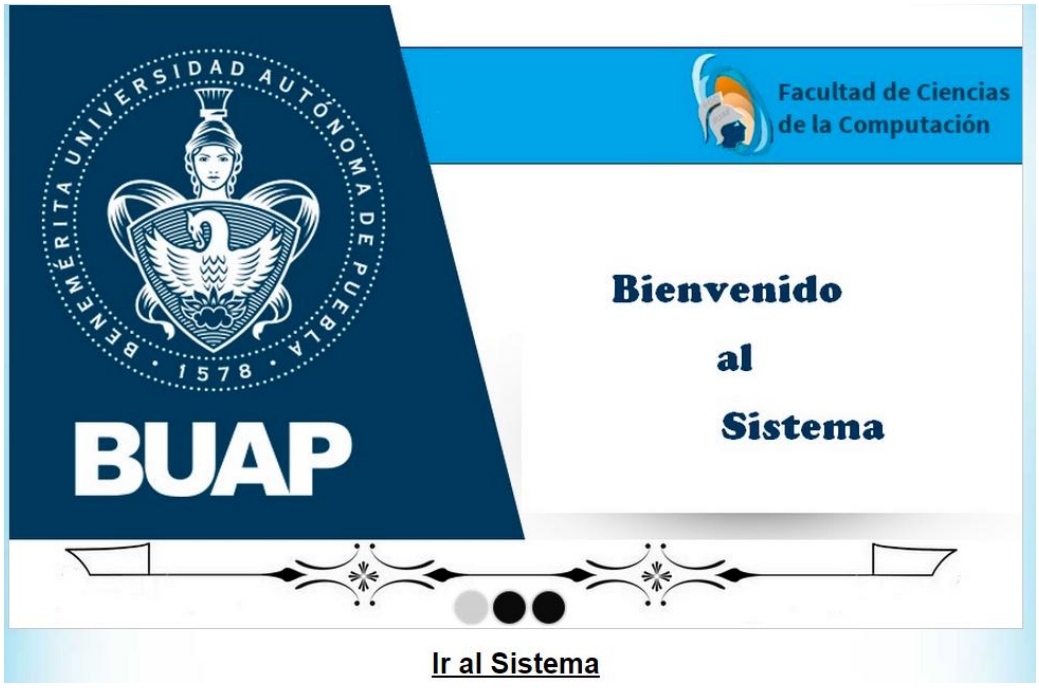

Figura 4.2.1.1: Imagen 1 del Slider en la página de Inicio

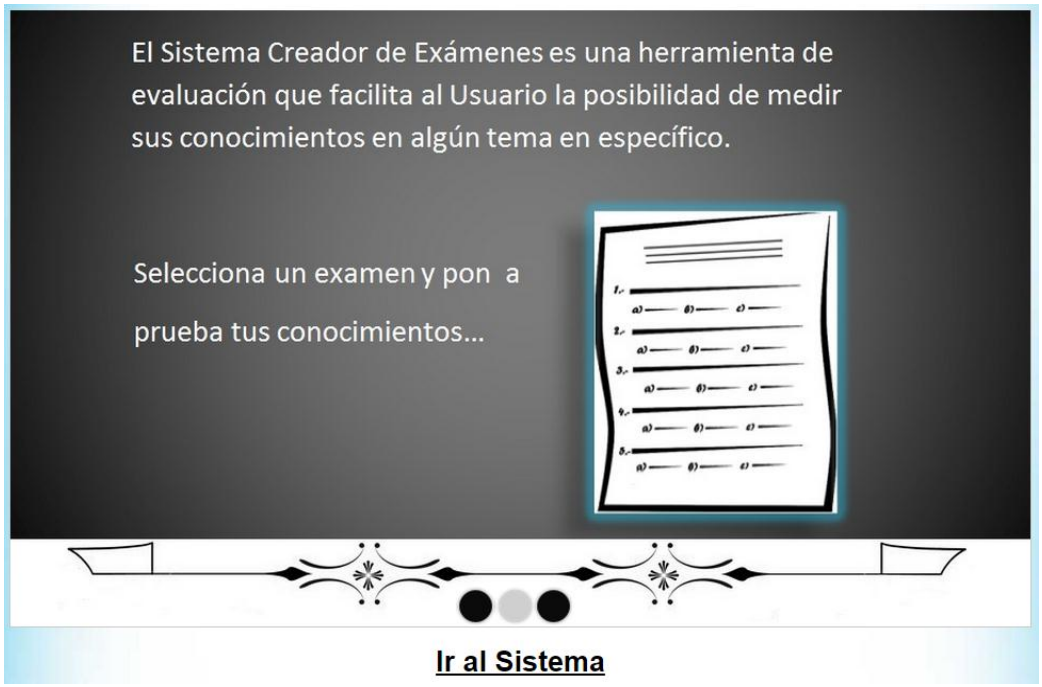

Figura 4.2.1.2: Imagen 2 del Slider en la página de Inicio

Figura 4.2.1.3: Imagen 3 del Slider en la página de Inicio

4.2.2. Página de login

En esta pantalla (Figura 4.2.2.1) podemos ver dos cuadros de texto, en donde el usuario que desee acceder al sistema deberá ingresar su login (id de identificación) y su contraseña. En caso de no estar registrado, en la parte de abajo podemos distinguir un enlace con letras rojas que al ser seleccionado nos muestra la pantalla de registro.

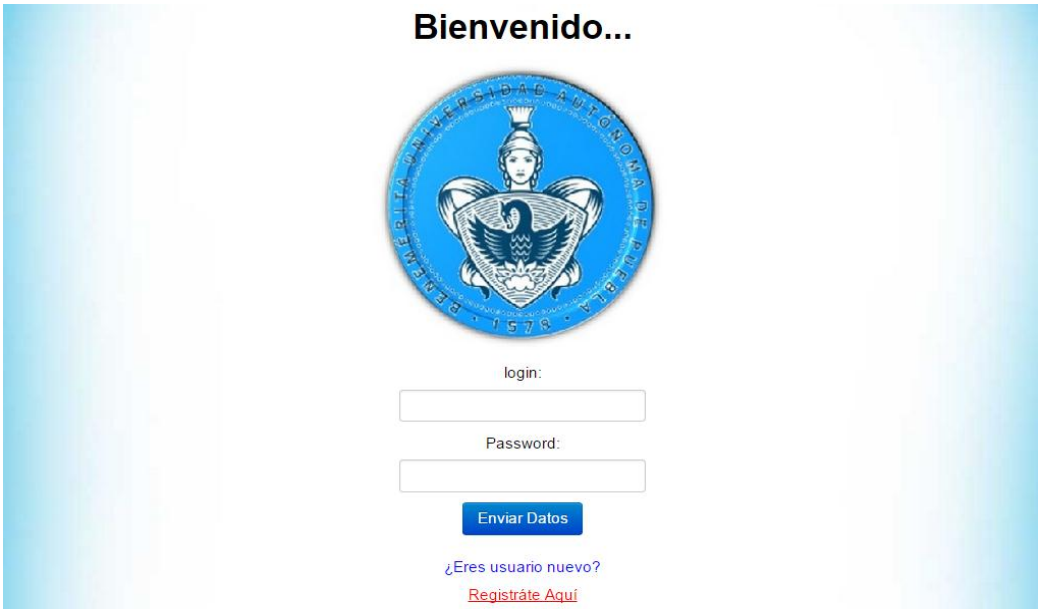

Figura 4.2.2.1: Página de login para todos los Usuarios

4.2.3. Página de registro

En la pantalla de registro (Figura 4.2.3.1), el usuario deberá ingresar algunos datos personales para su identificación, dicha información será almacenada en caso de ser requerida más adelante. Esta pantalla soló se utilizará para las personas que deseen acceder por primera vez al sistema.

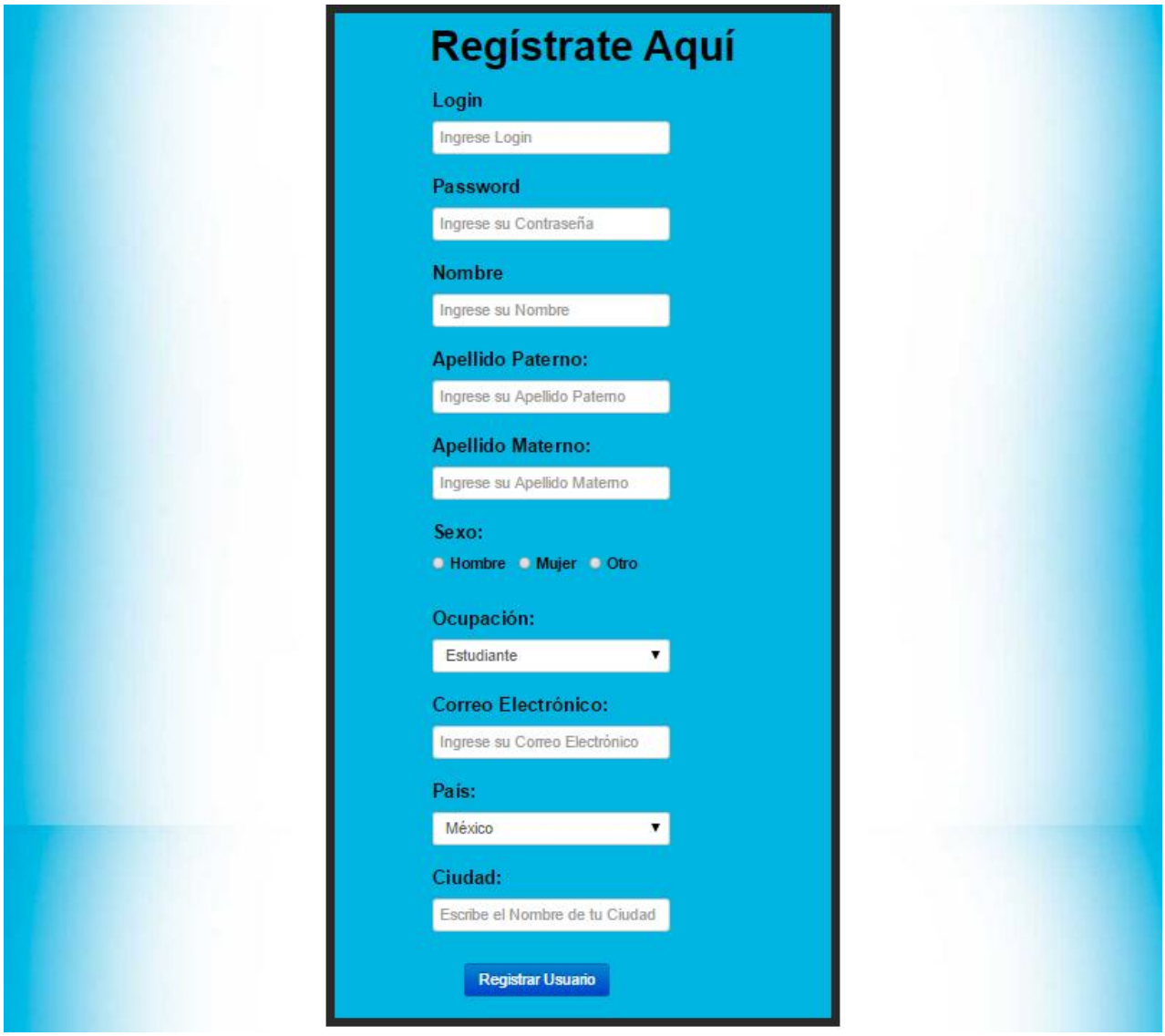

Figura 4.2.3.1: Página de registro para un usuario nuevo

4.3. Interfaces según el tipo de usuario

Después de haber ingresado al Sistema, el Usuario podrá ver un tipo de menú según el rol que tenga. El menú de un Administrador es diferente al de un Usuario Registrado. En este capítulo se muestran las capturas de pantalla de cada uno de los menús, según el usuario que haya iniciado sesión.

4.3.1. Opciones disponibles para un Administrador

En la figura 4.3.1.1 podemos ver que el menú de un Administrador cuenta con diferentes opciones para la creación de exámenes y una mejor organización del trabajo.

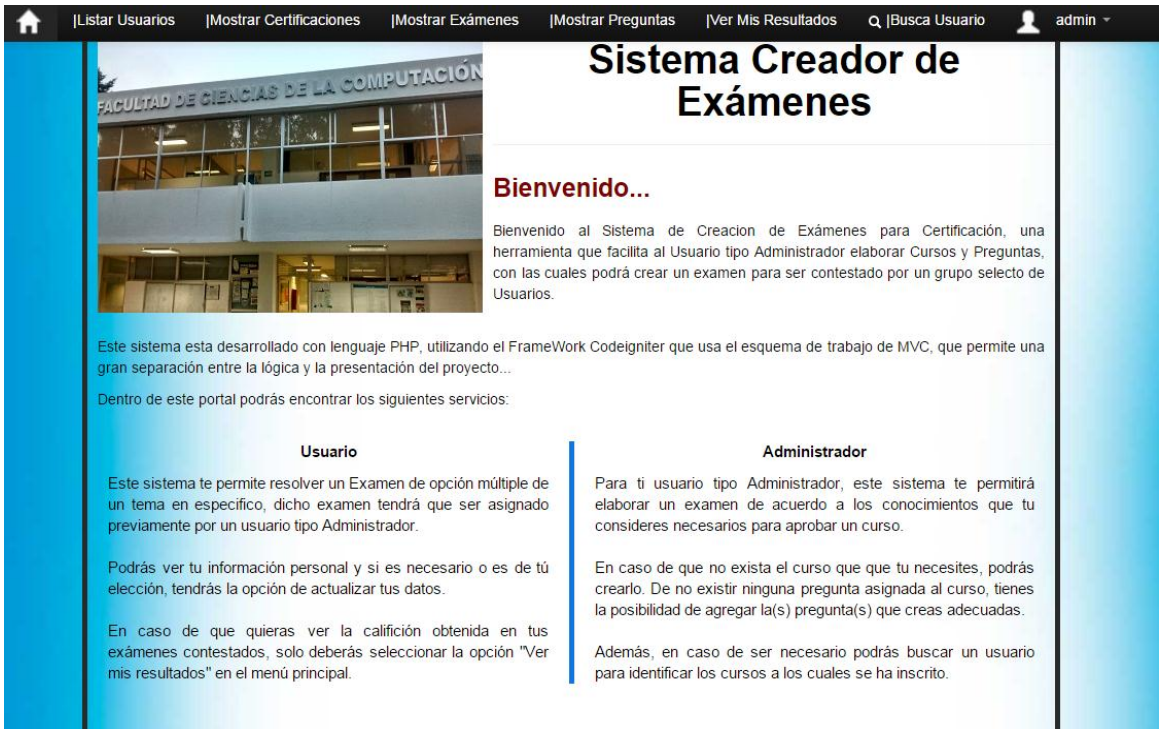

Figura 4.3.1.1: Página principal para un Administrador

4.3.1.1. Opción “Listar Usuarios”

Al seleccionar esta opción, el sistema desplegará una lista con el nombre de los usuarios registrados y las certificaciones (cursos) en las que han respondido uno o más exámenes, esto lo podemos observar en la figura 4.3.1.1.1.

Figura 4.3.1.1.1: Lista de usuarios registrados

4.3.1.2. Opción "Mostrar Certificaciones"

Si el administrador elige esta opción, podrá ver la página que muestra la lista de certificaciones (cursos) que han sido registradas (Figura 4.3.1.2.1). Esta lista tiene el login del usuario que registró la certificación y el nombre de la misma. Además, en un menú lateral se encuentran disponibles las opciones: "Añadir Certificaciones" (Figura 4.3.1.2.2) y "Certificaciones con Grocery" (Figura 4.3.1.2.3).

Figura 4.3.1.2.1: Captura de pantalla al seleccionar la opción "Mostrar Certificaciones"

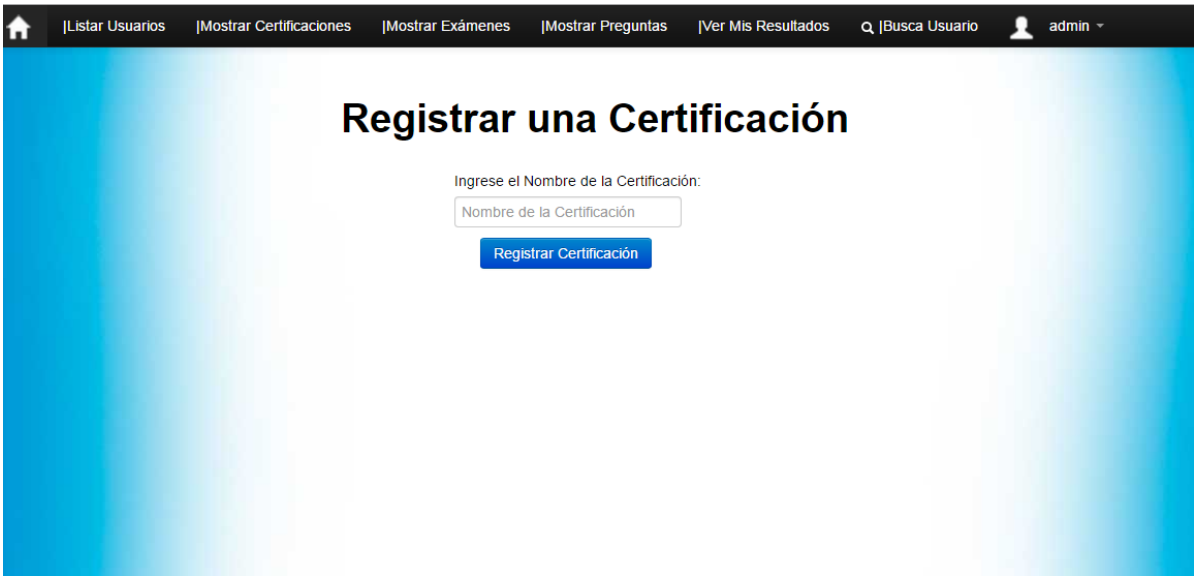

Figura 4.3.1.2.2: Página para registrar una certificación

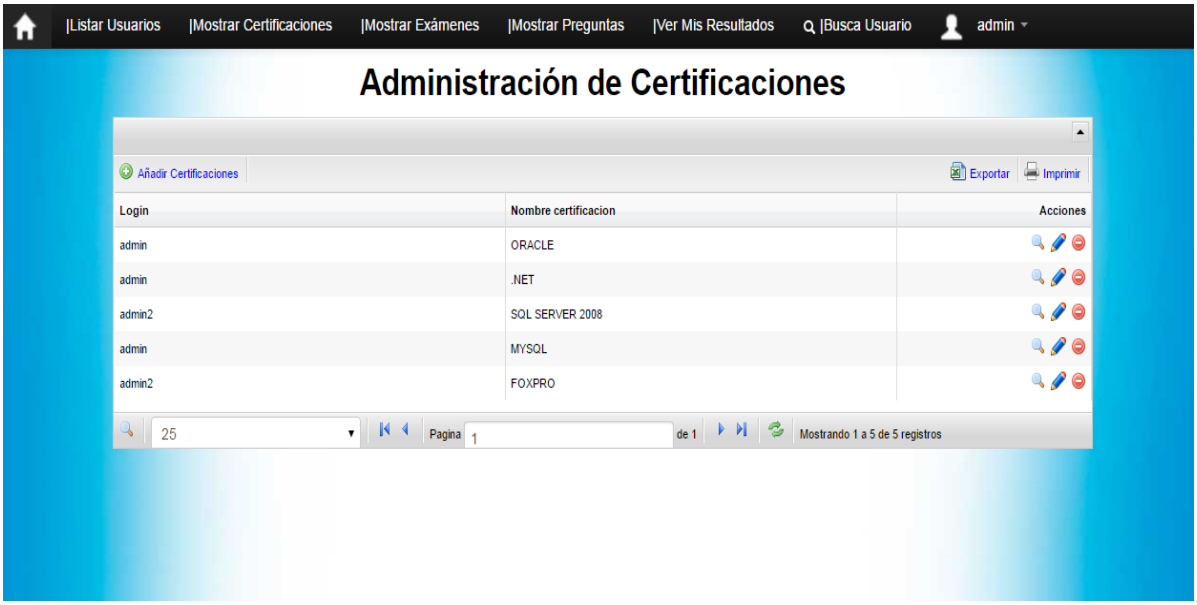

Figura 4.3.1.2.3: Administración de certificaciones utilizando la Librería Grocery Crud

4.3.1.3. Opción “Mostrar Exámenes”

Cuando el administrador seleccione esta opción, podrá ver en pantalla una lista con todos los exámenes que han sido creados (Figura 4.3.1.3.1). Dicha lista tiene el identificador (código) del examen, el login de la persona que lo creó, la certificación (curso) a la que pertenece y la vigencia o hasta cuando esta disponible. Además, cada uno de ellos cuenta con un enlace que nos dirige al examen en caso de querer contestarlo y obtener una calificación.

El Administrador que desee evaluar al grupo del cual es responsable, deberá crear un examen con las características que él considere adecuadas (Numero de preguntas, tiempo para contestar, fecha de vigencia del examen). Esto lo podrá hacer seleccionando el enlace con la leyenda “Añadir un Examen”, ubicado en un menú lateral de la misma página.

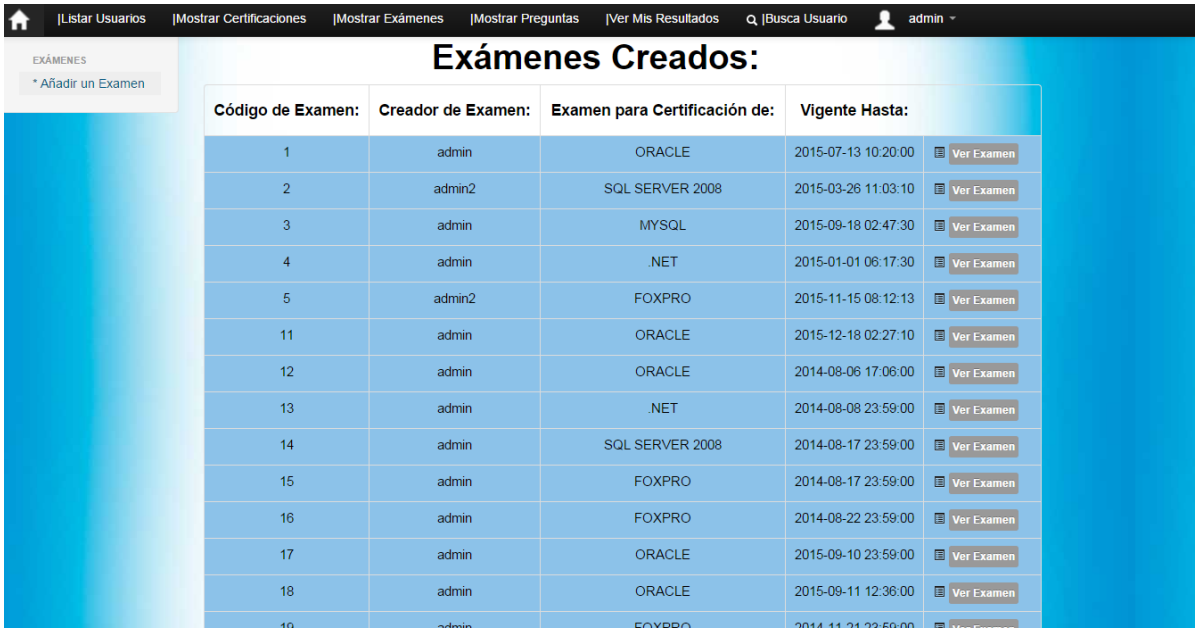

Figura 4.3.1.3.1: Lista de exámenes creados

Una vez que haya sido elegida la opción “Añadir examen”, el sistema mostrará una lista con las certificaciones en las que se puede crear un examen (Figura 4.3.1.3.2). Si alguna de ellas fue seleccionada, podremos ver en pantalla la interfaz de creación de exámenes, misma que puede ser observada en la figura 4.3.1.3.3.

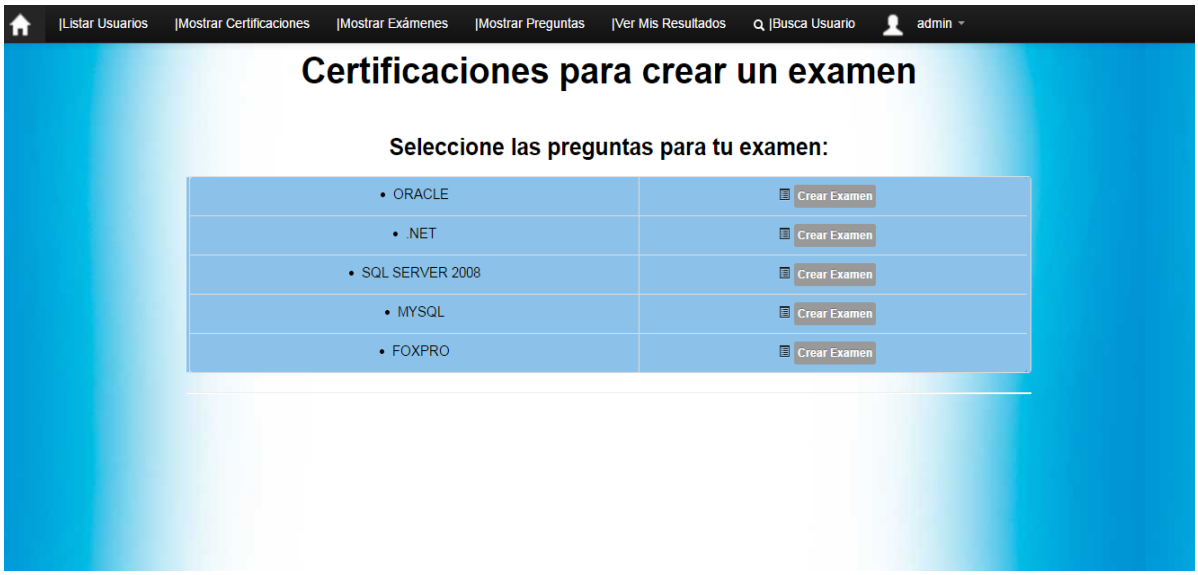

Figura 4.3.1.3.2: Lista de certificaciones en las que se puede crear un examen

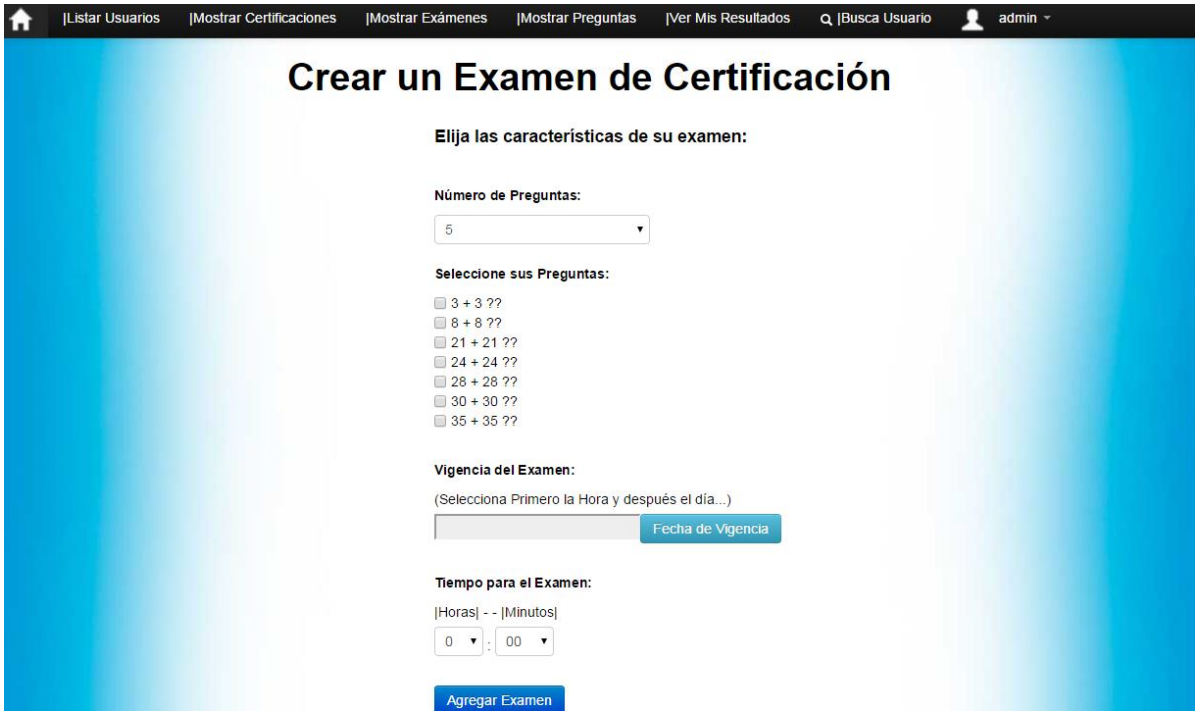

Figura 4.3.1.3.3: Página mostrada al seleccionar una certificación para crear un examen

4.3.1.4. Opción “Mostrar Preguntas”

Al ser seleccionada, se podrá ver en pantalla una lista con las certificaciones (cursos) disponibles (Figura 4.3.1.4.1). En cada una de ellas se encuentra un enlace que al ser seleccionado mostrará las preguntas registradas en dicha certificación (Figura 4.3.1.4.2). Además, tendrá la opción de agregar una pregunta nueva en el curso que desee (En la figura 4.3.1.4.3 podemos ver la interfaz que nos permite agregarla), esto será posible escogiendo la opción “Añadir una pregunta”, del menú lateral.

Figura 4.3.1.4.1: Lista de certificaciones en las que se puede crear una pregunta

Figura 4.3.1.4.2: Lista de preguntas agregadas en la certificación que fue seleccionada

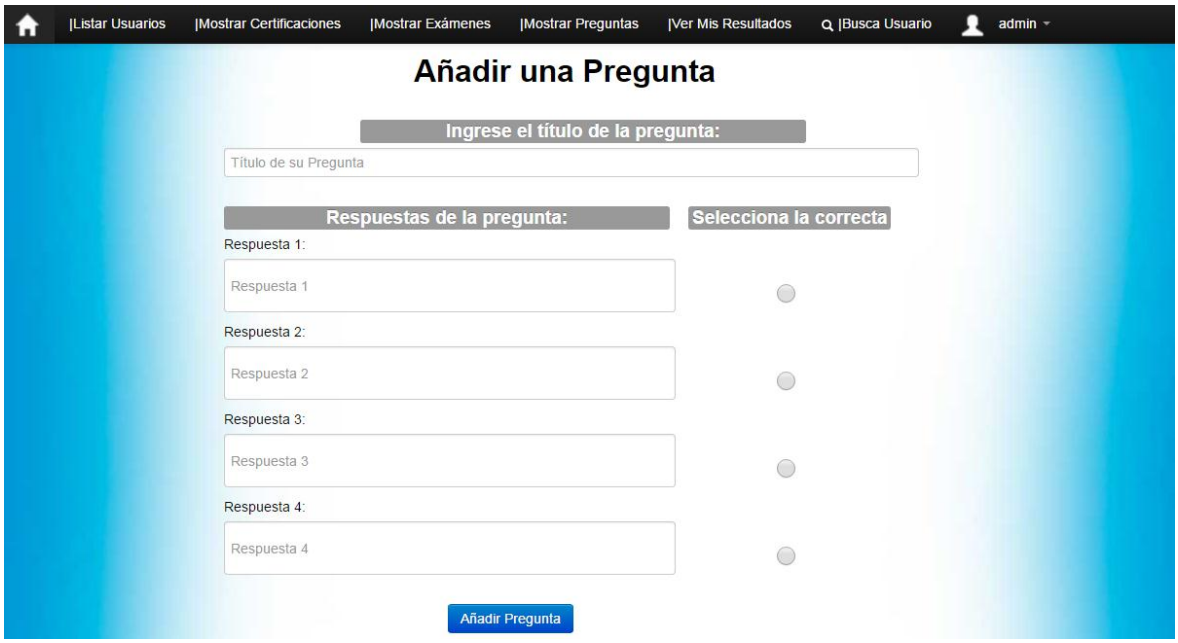

Figura 4.3.1.4.3: Página mostrada al seleccionar una certificación para agregar pregunta

4.3.1.5. Opción "Ver Mis Resultados"

Cuando sea elegida esta opción, se podrá ver en pantalla una lista con la calificación de los exámenes que ha respondido el Administrador (Figura 4.3.1.5.1), si es que ha resuelto alguno. Cada elemento de la lista tiene la calificación obtenida al contestar un examen y la certificación a la que pertenece.

Login	Certificación	Código de Examen	Calificación del Examen
admin	ORACLE	18	0.00
admin	ORACLE	18	10.00
admin	ORACLE	30	0.00
admin	ORACLE	30	0.00
admin	FOXPRO	31	4.00
admin	FOXPRO	5	0.00
admin	FOXPRO	5	10.00
admin	FOXPRO	31	0.00
admin	MYSQL	29	0.00
admin	SQL SERVER 2008	32	0.00
admin	.NET	33	10.00

Figura 4.3.1.5.1: Calificación de los exámenes que ha respondido el usuario en sesión

4.3.1.6. Opción “Busca Usuario”

El administrador podrá elegirla en caso de que desee buscar a un usuario que esté registrado en el sistema (En la figura 4.3.1.6.1 se puede observar la interfaz de búsqueda de un usuario). Solo bastará con que ingrese el nombre y apellidos (Paterno y materno) del usuario a buscar. En caso de que se encuentre la persona buscada, el sistema mostrará el nombre del usuario encontrado y el nombre de las certificaciones en las cuales ha contestado un examen, como se puede ver en el ejemplo de la figura 4.3.1.6.2.

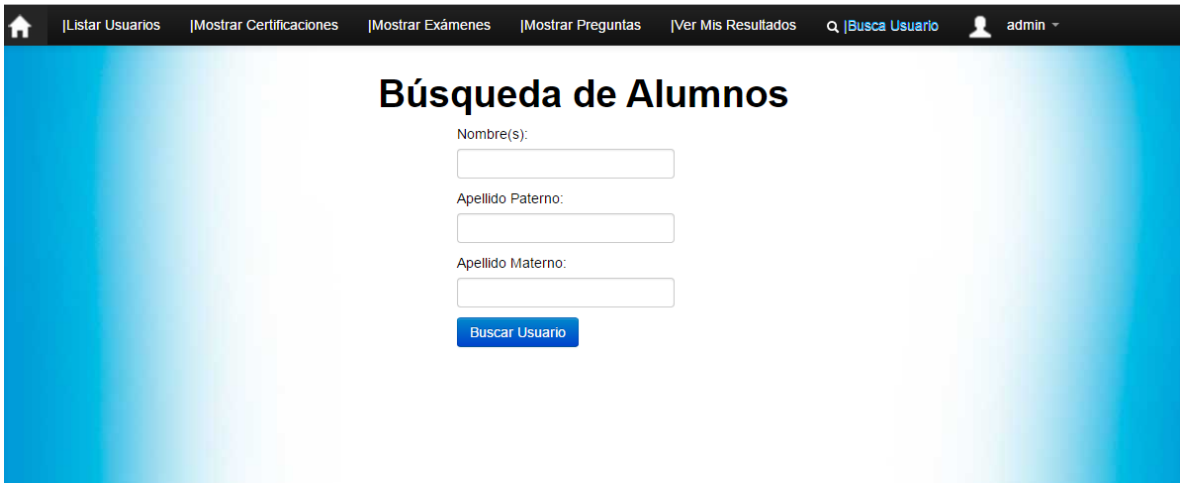

Figura 4.3.1.6.1: Buscador de usuarios registrados en el sistema

Figura 4.3.1.6.2: Ejemplo de búsqueda exitosa de un usuario

4.3.1.7. Opción “Ver Perfil”

En la figura 4.3.1.7.1, de izquierda a derecha, al final del menú principal, podemos ver una etiqueta con el login (identificador) del usuario que haya iniciado sesión, a un lado se encuentra una pestaña que al ser seleccionada desplegará un menú con dos opciones: “Ver Perfil” y “Cerrar Sesión”. La primera nos permite visualizar la información personal que fue introducida al momento de registrarnos (Ver figura 4.3.1.7.2). En caso de querer modificar dicha información, podremos hacerlo seleccionando el enlace Editar Información” del menú lateral en la misma página. En la figura 4.3.1.7.3 podemos ver un ejemplo de la segunda opción, que solo tiene la función de cerrar la sesión del usuario activo.

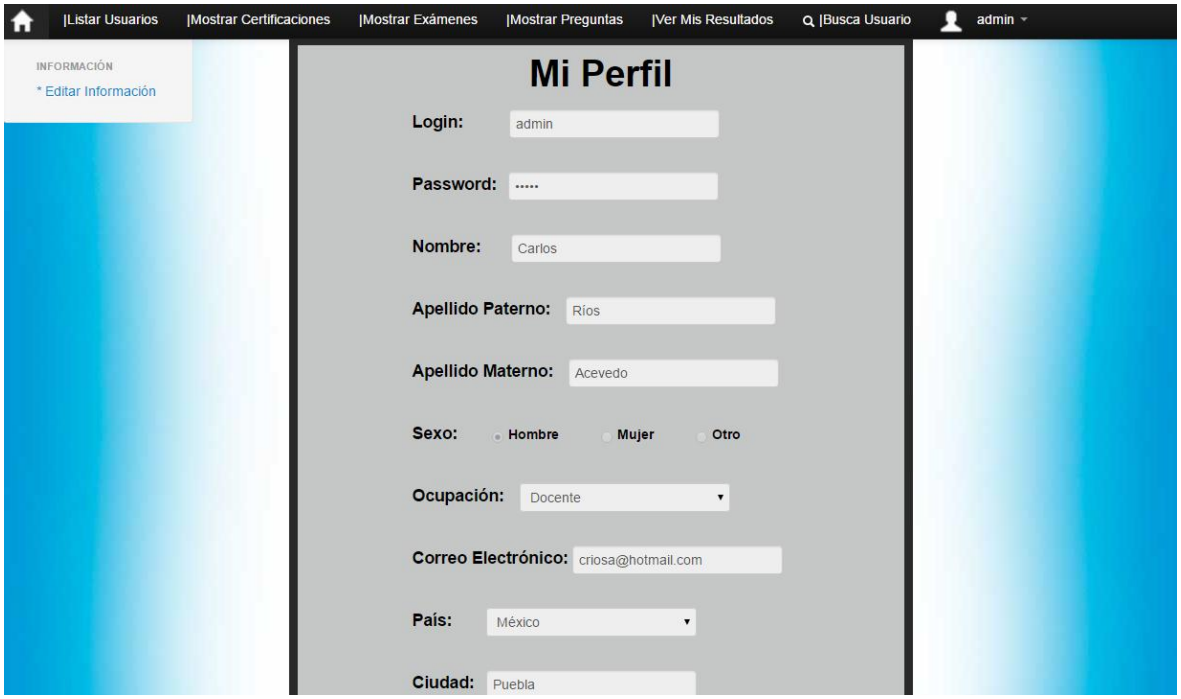

Figura 4.3.1.7.1: Página mostrada al seleccionar la opción: “Ver Perfil”

Figura 4.3.1.7.2: Página mostrada al seleccionar la opción: "Editar Perfil"

Figura 4.3.1.7.3: Pantalla que muestra la opción: "Cerrar Sesión" del menú principal

4.3.2. Pantalla de Bienvenida para Usuarios Registrados

Un usuario registrado sólo tiene disponibles las opciones de “Mostrar Exámenes”, “Ver Mis Resultados” y la opción que permite ver y modificar la información del perfil de usuario (Figura 4.3.2.1).

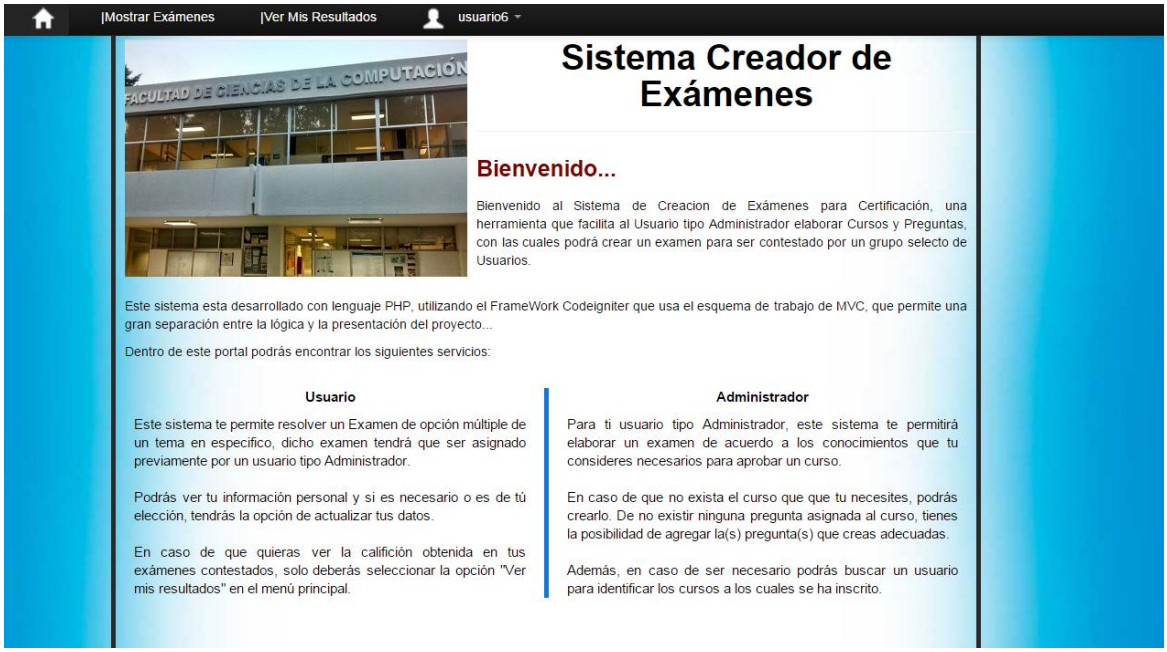

Figura 4.3.2.1: Pantalla de Bienvenida para usuarios registrados

4.3.2.1. Opción “Mostrar Exámenes” para Usuarios Registrados

Los usuarios registrados podrán ver en pantalla una lista con todos los exámenes que han sido creados, tal y como se muestra en la figura 4.3.2.1.1. Esta lista como ya mencionamos antes, tiene el identificador (código) del examen, el login de la persona que lo creó, la certificación (curso) a la que pertenece y la vigencia o hasta cuando está disponible. En cada elemento de la lista se puede distinguir un enlace que nos lleva a otra página que muestra el examen seleccionado. Una persona registrada que desee contestar cualquiera de los exámenes de la lista, solo deberá escoger el indicado (el que haya sido señalado por el administrador) y si aún no completa sus dos intentos, podrá acceder a él. En la figura 4.3.2.1.2 podemos ver el ejemplo de un examen seleccionado.

Al terminar de contestar y enviar los resultados, el usuario regresará a la pantalla que muestra la lista total de exámenes.

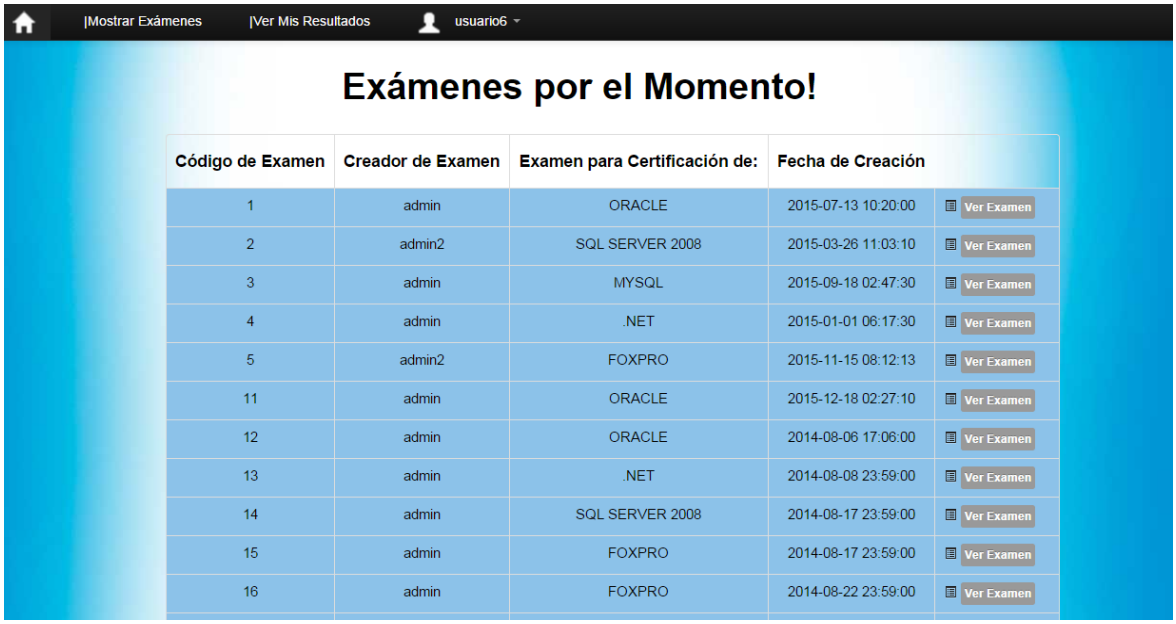

Figura 4.3.2.1.1: Opción: "Mostrar Exámenes" para usuarios registrados

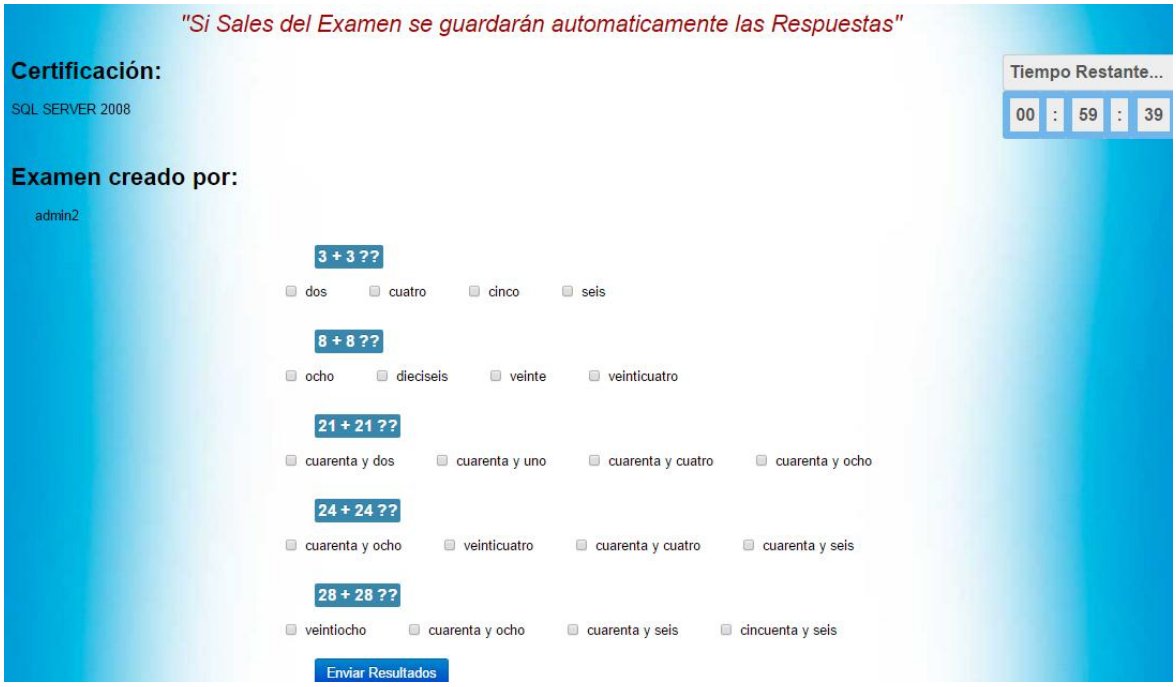

Figura 4.3.2.1.2: Ejemplo de Examen mostrado a los usuarios que lo elijan

4.3.2.2. Opción “Ver Mis Resultados”

Cuando sea seleccionada esta opción, el usuario podrá ver en pantalla una lista con la calificación de los exámenes que ha respondido, en caso de haber resuelto alguno. Dicha lista tiene el nombre de la certificación a la que pertenece cada examen contestado y la calificación obtenida después de ser evaluado por el sistema (Figura 4.3.2.2.1).

Figura 4.3.2.2.1: Lista de calificaciones en los exámenes contestados

4.3.2.3. Opción “Ver Perfil”

Se puede observar en el menú principal una etiqueta con el login (identificador) del usuario que ha iniciado sesión, esta tiene a un lado una pestaña que al ser seleccionada desplegará un menú con dos opciones: “Ver Perfil” y “Cerrar Sesión”, tal y como se muestra en la figura 4.3.2.3.1. Con la primera opción se podrá ver en pantalla la información personal del usuario que haya iniciado sesión. Recordemos que estos datos fueron registrados la primera vez que se accedió al sistema.

Si un usuario desea modificar dicha información, podrá hacerlo seleccionando el enlace “Editar Información” del menú lateral en la misma página. Podemos ver un ejemplo en la figura 4.3.2.3.2.

La segunda opción la podrá elegir el usuario activo en caso de querer cerrar su sesión (Figura 4.3.2.3.3).

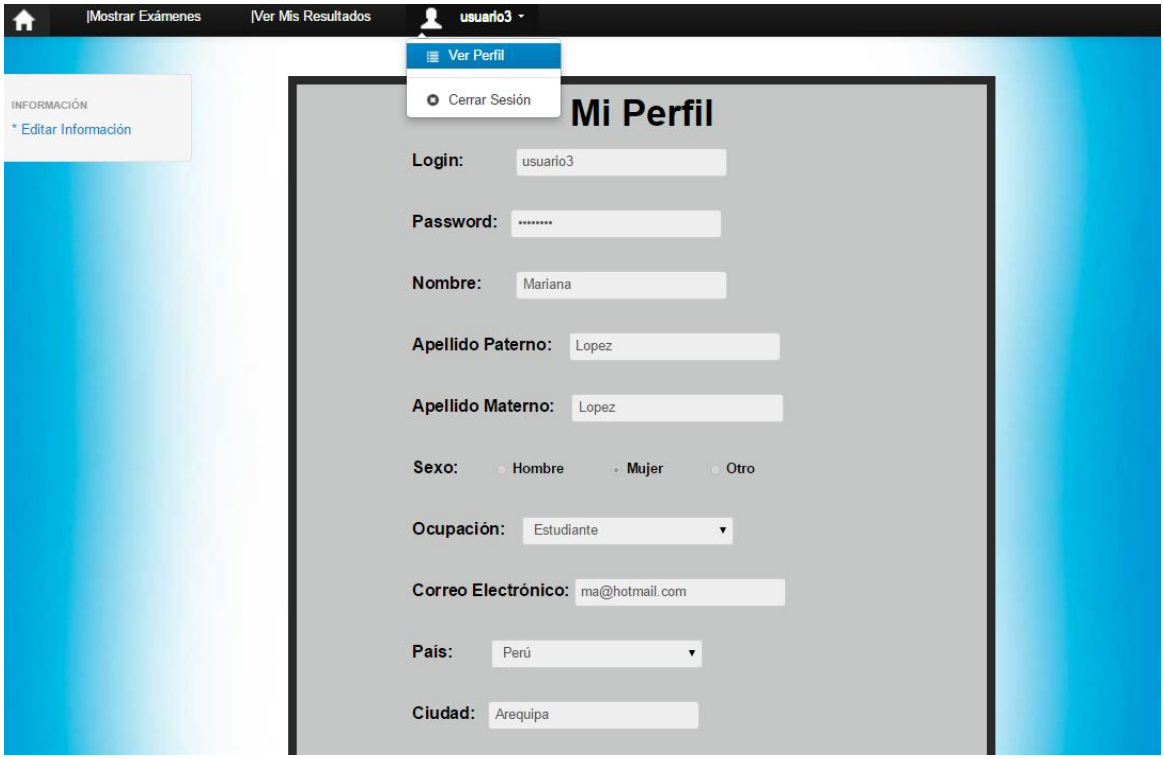

Figura 4.3.2.3.1: Pantalla mostrada al seleccionar la opción "Ver Perfil"

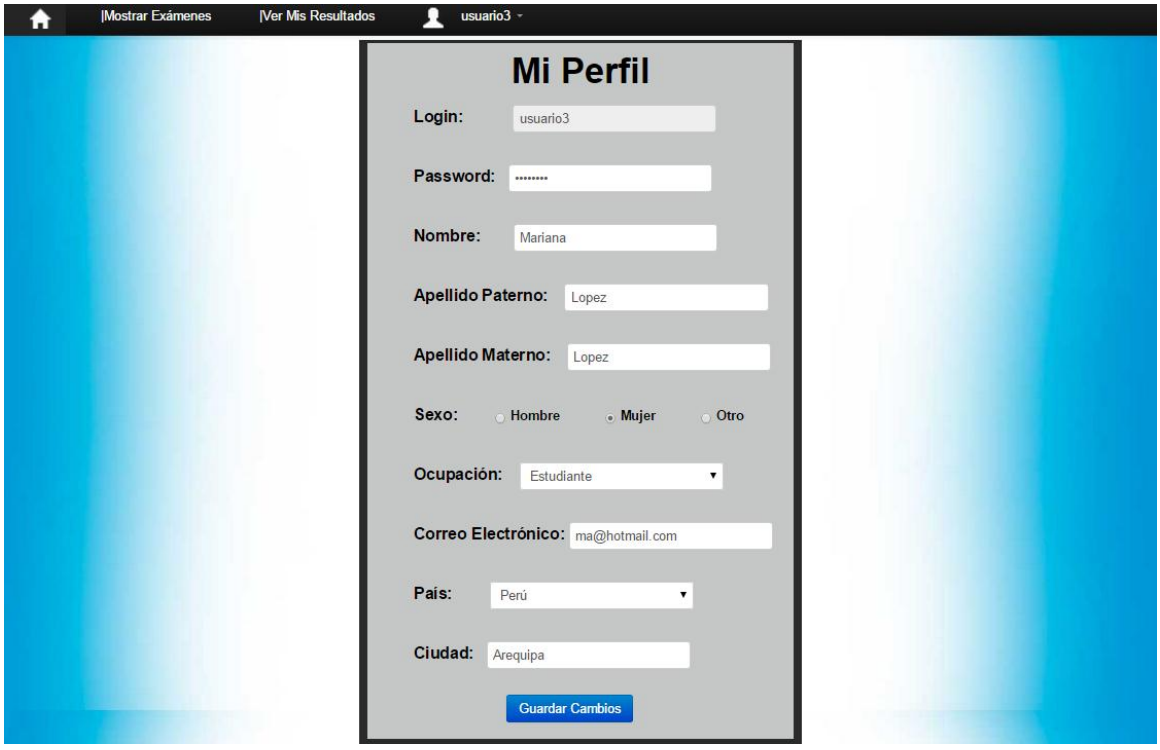

Figura 4.3.2.3.2: Pantalla mostrada al seleccionar el enlace "Editar Información"

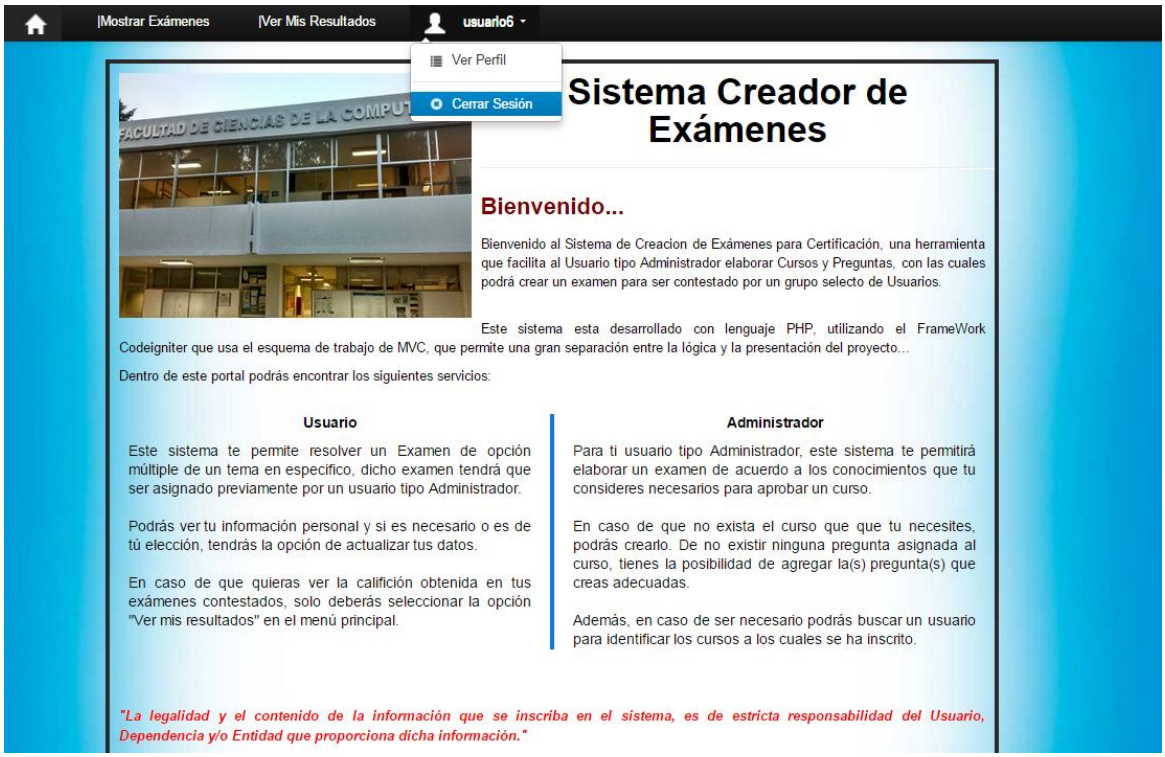

Figura 4.3.2.3.3: Pantalla que muestra la opción: "Cerrar Sesión" del menú principal

CONCLUSIONES Y TRABAJOS A FUTURO

Luego de revisar y verificar los objetivos planteados al comienzo de este proyecto, se llega a la conclusión de que todos los objetivos fueron alcanzados con éxito. Podemos decir que se ha realizado un Sistema Web que proporciona una herramienta de evaluación dirigida principalmente a capacitadores que tienen a su cargo un grupo de personas que participan en un curso. Este programa nos puede ayudar a tener un conocimiento global del desempeño de dichos usuarios (previamente registrados en el Sistema), con lo cual se podrá interpretar los resultados y tomar las decisiones que sean necesarias.

Gran parte de ese logro fue posible gracias a la metodología ágil SCRUM, utilizada en este trabajo. Se comprobó que una metodología ágil es eficiente, ya que es capaz de ajustarse a las cambiantes necesidades en el desarrollo del proyecto.

La importancia de este trabajo radica en poder cumplir con los requerimientos del Sistema, para esto fue fundamental el proceso de análisis, ya que permitió llevar a cabo una correcta priorización y dirección del trabajo. Fue posible conocer en todo momento el grado de evolución del proyecto, se pudieron identificar los avances y determinar en qué etapa del plan de trabajo había retrasos, con lo cual se redirigió el curso de los planes para alinearse a las nuevas necesidades.

La utilización del Framework “Codeigniter” fue determinante para poder cumplir con los objetivos planteados. Esta herramienta para el lenguaje PHP permitió la correcta organización y desarrollo del Software. La utilización de algunos de sus componentes nos facilitó el trabajo, ya que se pudo evitar reescribir varias líneas de código.

Asimismo, el uso del Framework “Bootstrap” permitió el desarrollo de un software con una interfaz amigable, con lo cual se satisface uno de los requerimientos especificados al inicio del proyecto.

El resultado de este trabajo se pone a disposición del público, como ya se mencionó con anterioridad, el Sistema fue realizado con código abierto, esto quiere decir que el usuario que desee hacer alguna modificación o mejora pueda llevarla a cabo, aunque se recomienda solo hacer cambios que puedan mejorar el desempeño del Software.

En todo el desarrollo del proyecto siempre se pensó en la interacción de los usuarios con el Sistema y se espera que cuando llegué el momento de poner en marcha el Software creado, los resultados sean positivos.

El impacto que pueda tener la implementación del sistema dependerá de la frecuencia con la cual sea utilizado el mismo y de la interacción que exista entre este software y el usuario, ya que será el usuario el que tenga la última palabra sobre el funcionamiento y la efectividad del sitio web.

Se considera en un futuro cercano poder darle continuidad al presente trabajo y agregar la sección de preguntas cerradas, con esto se podrá darle más funcionalidad al Software y será más eficiente la manera de evaluar el conocimiento de un usuario. Además, se planea hacer que cada una de las opciones disponibles al crear un examen puedan adaptarse al gusto del administrador (el que crea un examen), por ejemplo: que el número de preguntas que desee agregar pueda ser un número distinto al del menú desplegable que tiene el Sistema, así como el tiempo que va a tener el usuario para contestar cada examen.

Otra sugerencia de trabajo a futuro es la creación de un módulo de pruebas o ensayos para practicar el o los exámenes que el usuario vaya a responder. De esta manera se tendrían los elementos necesarios para obtener una buena calificación cuando llegue el momento de contestar la prueba final.

Cabe mencionar que para el completo funcionamiento del Sistema será necesario crear un banco de preguntas aprobado por especialistas en el tema, así posteriormente se tendrá la seguridad de que se ha creado un examen fiable, que medirá con un alto grado de precisión el conocimiento que un usuario ha adquirido en determinado curso, todo esto servirá para establecer si la persona ha obtenido una calificación aprobatoria o no, con lo cual se podrán tomar las decisiones correspondientes.

BIBLIOGRAFÍA

[1] Wesley Longman. (Septima Edición). (2001). Introducción a los Sistemas de Bases de Datos. México: Pearson Educación de México, S.A. de C.V.

[2] D.Ullman Jeffrey, Widom Jennifer. (Primera Edición). (1999). Introducción a los Sistemas de Bases de Datos. México: Prentice Hall Hispanoamericana, S.A.

[3] M. Kroenke David. (Octava Edición). (2003). Procesamiento de Bases de Datos. México: Pearson Educación de México, S.A.

[4] De Miguel Castaño Adoración, G. Piattini Velthius. (Segunda Edición). (1999). Fundamentos y modelos de Bases de Datos. México: Alfaomega Grupo Editor, S.A. de C.V.

[5] Celma Giménez Matilde, Casamayor Ródenas Juan Carlos, Mota Herranz Laura. (2003). Bases de Datos Relacionales. México: Pearson Educación, S.A.

[6] Spona Helma. (Primera Edición). (2010). Programación de Bases de Datos con MySQL y PHP. México: Alfaomega Grupo Editor, S.A. de C.V.

[7] Cuadra Dolores, Castro Elena, Mª Iglesias Ana, Martínez Paloma, Calle Fco. Javier, De Pablo César, Al-Jumaily Harith, Moreno Lourdes. (Primera Edición). (2008). Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación. México: Alfaomega Grupo Editor, S.A. de C.V.

[8] Qué es SCRUM. [En Línea]
Disponible: http://www.proyectosagiles.org/que-es-scrum

[9] La Guía de Scrum. [En Línea]
Disponible: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf

[10] Cómo utilizar la metodología Scrum para acometer proyectos. [En Línea]
Disponible: http://comunidad.iebschool.com/iebs/general/metodologia-scrum/

[11] Modelo SCRUM. [En Línea]
Disponible: http://www.navegapolis.net/files/s/NST-010_01.pdf

[12] Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? [En Línea]
Disponible: http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/

[13] Una Introducción a APACHE. [En Línea]
Disponible: http://linux.ciberaula.com/articulo/linux_apache_intro

[14] MySQL con Clase. [En Línea]
Disponible: http://mysql.conclase.net/curso/?cap=007a

[15] MySQL :: MySQL 5.5 Reference Manual. [En Línea]
Disponible: http://dev.mysql.com/doc/refman/5.5/en/tutorial.html

[16] Relaciones, claves primarias y foráneas. [En Línea]
Disponible: http://deletesql.com/viewtopic.php?f=5&t=20

[17] Llaves Primarias y Foraneas. [En Línea]
Disponible:
http://ingsystemas.webnode.es/bases-de-datos/llaves-primarias-y-foraneas/

[18] ¿Qué es un Framework Web? [En Línea]
Disponible: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

[19] Los Frameworks de PHP agilizan tu trabajo. [En Línea]
Disponible:
http://www.maestrosdelweb.com/los-frameworks-de-php-agilizan-tu-trabajo/

[20] Qué es MVC. [En Línea]
Disponible: http://www.desarrolloweb.com/articulos/que-es-mvc.html

[21] MVC (Model, View, Controller) explicado. [En Línea]
Disponible:
https://codigofacilito.com/articulos/mvc-model-view-controller-explicado

[22] Aplicación del patrón MVC en PHP. [En Línea]
Disponible:
http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/257

[23] MVC: Modelo, Vista y Controlador en PHP. [En Línea]
Disponible:
http://www.jc-mouse.net/proyectos/mvc-modelo-vista-y-controlador-en-php

[24] Tutorial MVC con PHP. [En Línea]
Disponible: http://blog.tednologia.com/mvc-en-php/

[25] Introducción al framework CodeIgniter. [En Línea]
Disponible: http://www.cristalab.com/tutoriales/introduccion-al-framework-codeigniter-c65436l/

[26] Bienvenido a CodeIgniter: Guía de Usuario de CodeIgniter. [En Línea]
Disponible: http://www.escodeigniter.com/guia_usuario/

[27] Manual de Codeigniter. [En Línea]
Disponible: http://www.desarrolloweb.com/manuales/manual-codeigniter.html

[28] CodeIgniter Guía del Usuario en Español. [En Línea]
Disponible:
http://www.areadepruebas.com.ar/downloads/CodeIgniter_Guia_Usuario_2.1.3.pdf

[29] Información general de MySQL para PHP. [En Línea]
Disponible: http://php.net/manual/es/mysql.php

[30] Introducción a CSS. [En Línea]
Disponible: https://librosweb.es/libro/css/

[31] Guía Breve de CSS. [En Línea]
Disponible: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo

[32] ¿Que es Bootstrap? [En Línea]
Disponible: https://raiolanetworks.es/blog/que-es-bootstrap/

[33] Bootstrap 3, el manual oficial. [En Línea]
Disponible: https://librosweb.es/libro/bootstrap_3

[34] Bootstrap 3 Tutorial. [En Línea]
Disponible: http://www.w3schools.com/bootstrap