



BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA
“FACULTAD DE CIENCIAS DE LA COMPUTACIÓN”

TESIS:

SISTEMA DE INFORMACIÓN PARA ADMINISTRAR
CASOS PRÁCTICOS PARA PERFILES PROFESIONALES
(SICP)

ASESOR:

M.C. MELIZA CONTRERAS GONZÁLEZ

AUTOR:

ELIZABETH CERÓN AQUINO

Puebla, Pue., martes 9 de diciembre de 2018



INDICE

ANTECEDENTES	4
Capítulo 1. FUNDAMENTOS TEÓRICOS.....	6
1. UML Lenguaje de Modelo Unificado.....	7
1.1 UML Modelos.....	8
1.1.1 Beneficios de esta Tecnología	8
1.1.2 Elementos de UML	9
1.1.3 Organización De Los Casos De Uso	11
1.2 Base De Datos.....	12
1.2.1 Estructura de una Base de Datos.....	12
1.3 Normalizando La Base De Datos	13
1.3.1 Primera Forma Normal (1fn).....	13
No se permiten campos nulos	13
1.3.2 Segunda Forma Normal (2fn)	14
1.3.3 Tercera Forma Normal (3fn)	14
1.3.4 Modelo Entidad Relación	14
1.4 MySQL.....	14
1.4.1 Orígenes	15
1.4.2 Evolución.....	15
1.4.3 Restricciones en MySQL.....	16
1.4.4 Claves Primarias	16
1.4.5 Claves Ajenas.....	16
1.5 PHP	18
1.6 HTML5.....	18
1.7 Macromedia Dreamweaver.....	20
1.8 Programas similares	20
1.8.1 EASYPLANEX	20
1.8.2 TEACH2000	21
Capítulo 2. ANÁLISIS Y DISEÑO DEL SISTEMA	22
2.1 Introducción al Análisis del Sistema	23
2.2 Análisis del sistema	23
2.2.1.1 Especificación de Casos de Uso	23



2.2.2	Diagrama de Casos de Uso.....	26
2.2.3	Diagrama de Clases	29
2.2.4	Diagrama de Secuencias	30
2.3	DISEÑO DE LA BASE DE DATOS.....	32
2.3.1	Modelo Entidad Relación	32
2.4	La Base de Datos Implementada	32
2.4.1	Tabla de Registro de Usuario	33
2.4.2	Tabla Registro de Caso Práctico.....	34
2.4.3	Tabla Catalogo Licenciatura	34
2.4.4	Tabla Catalogo Área.....	35
2.4.5	Tabla Catalogo Subárea	36
Capítulo 3. DISEÑO DE LA INTERFAZ DE USUARIO		37
3.1	Conceptos Generales	38
3.2	Balsamiq Mockups	38
BIBLIOGRAFIA		48

ANTECEDENTES

Los casos prácticos tienen el propósito de evaluar las habilidades en relación con el análisis e interpretación de información, así como la toma de decisiones y elaboración de juicios de valores, a partir de las aplicaciones de los conocimientos previos para encontrar la solución a una situación nueva. Se consideran un puente entre la teoría y la práctica, ya que se demuestre la comprensión total de la teoría a través de su aplicación [10].

Las ventajas de estos casos prácticos son claras, en el aspecto técnico resuelven la dificultad de elaboración de pruebas que miden los conocimientos de los estudiantes, también permite la comparación de resultados entre instituciones.

Considerando ampliamente las penurias que presenta el plan de trabajo para el proceso de elaboración de casos prácticos para perfiles profesionales se puede constatar diferentes problemas en la creación, desarrollo y búsqueda, en lo cual es complicado tener estructurado una base, ya que no se cuenta con un sistema o aplicación necesaria para consultar los casos prácticos bajo ciertos criterios (Titulo, Nombre del alumno, Asesor, bibliografía, palabras claves) y brindarles a alumnos, profesores, empleados que necesiten apoyo, orientación, ayuda, tramite, aclaraciones, entre otros.

El proyecto se vincula con los siguientes sistemas que se utilizan para calificar y establecer un orden de búsqueda por distintos parámetros (alumnos, tema, área, entre algunos más), se complementan en algunos puntos, pero no respaldan las necesidades marcadas para el uso de casos prácticos.

En la implementación del proyecto se toma en cuenta la tesis profesional: “Simulación de Pruebas para Preparación del Examen del CENEVAL” [9] el cual tiene como propósito que los alumnos mejoren su preparación para presentar tal prueba además de que practiquen y mejoren sus habilidades. La acreditación de esta prueba es muy importante para los alumnos, puesto que actualmente algunas universidades están optando por que los alumnos realicen este tipo de pruebas con el fin de que obtengan el título de la Licenciatura que estén cursando, y en la Benemérita Universidad Autónoma de Puebla, particularmente en la Facultad de Ciencias de la Computación, la acreditación de este examen es la pauta para obtener el Título tanto de Licenciatura como de Ingeniería.

Actualmente no se cuenta con una plataforma para la búsqueda de casos prácticos por lo que se realizara un proceso de búsqueda manual. Teniendo en cuenta que no se tiene un almacenamiento digital de casos prácticos para perfiles profesionales surge este proyecto, el cual se llevara a cabo mediante un sistema manejador de bases de datos para administrar de una manera más fácil y sencilla todos los casos dados y así ofrecer un mejor servicio a la comunidad estudiantil.



Por lo que se propone elaborar una solución automatizada de registro de casos prácticos para perfiles profesionales, que permita reducir al mínimo los errores y se encuentren disponibles para su revisión.

Se han considerado las siguientes etapas del proyecto:

- 1.- El análisis y diseño de sistema.
- 2.- Creación de la interfaz y desarrollo de aplicación web, utilizando multimedia y manejo digital de medios.
- 3.- Realización de pruebas para detección de fallas en el sistema.

La confección del sistema pretende proporcionar una herramienta de simple funcionamiento para la elaboración de casos prácticos, alcanzando mejores resultados y reduciendo el tiempo.



Capitulo1. FUNDAMENTOS TEÓRICOS

1. UML Lenguaje de Modelo Unificado

Desde tiempos pasados se han utilizado los casos de uso para describir un conjunto de secuencias, en las cuales cada secuencia representa la interacción de cosas fuera del sistema (actores) con el sistema (y sus abstracciones clave). Estos comportamientos son funciones a nivel del sistema que acostumbra visualizar, especificar, construir y documentar en la fase de obtención y análisis de los requerimientos. Un caso de uso representa un (o más) requerimiento funcional completo del sistema. Por ejemplo, un caso de uso central de un banco es procesar préstamos.

El UML, proporciona una representación gráfica de un caso de uso y un actor. Esta notación permite visualizar un caso de uso independiente de su implementación y en el contexto de otros casos de uso [2].

En el caso de la implementación usaremos un programa en este caso llamado Visual Paradigm (es un UML herramienta de diseño y una herramienta CASE UML diseñado para ayudar al desarrollo de software. VP-UML soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML)). El cuál nos permitirá hacer digitalmente la visualización de los casos de uso, obteniendo un resultado más legible y exacto a los requisitos en la construcción del sistema para la creación de casos prácticos.

En conceptos de creación de un sistema para la creación de casos prácticos tenemos el diseño de bases de datos para tener un diseño más claro y un almacenamiento más compacto y fácil de utilizar.

El Lenguaje Unificado de Modelado (UML) es una técnica para la especificación de sistemas en todas sus fases. Este ha sido desarrollado por los más importantes autores en materia de Análisis y Diseño de Sistemas y ha sido usada con éxito en sistemas hechos para toda clase de industrias alrededor del mundo: Salud, Bancos, Comunicaciones, Aeronáutica, Finanzas, etc.

Sin lugar a dudas OOAD (Object Oriented Analysis and Design), implementado con UML, es la metodología más avanzada en la actualidad. Esta metodología introduce los Casos de Uso, una poderosa herramienta para reducir los riesgos en la definición de requerimientos de sistemas nuevos. Los Casos de Uso sirven como columna vertebral del proceso de desarrollo de aplicaciones y tienen como objetivo garantizar que los resultados se ajusten completamente a las expectativas de los usuarios finales.

Analiza los diagramas que componen UML y ofrece acercamientos a casos de uso guiados sobre cómo estos diagramas se usan para modelar sistemas. También trata los mecanismos de extensibilidad de UML, los cuales permiten ampliar su notación y su semántica.

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos

concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado Racional) -pero no especifica en sí mismo qué metodología o proceso usar.

1.1 UML Modelos

Los modelos permiten:

- Visualizar como es un sistema o como queremos que sea.
- Especificar la estructura y/o comportamiento de un sistema.
- Hacer una plantilla que guíe la construcción de los sistemas
- Documentar las decisiones que hemos tomado.

El modelado sirve no solamente para los grandes sistemas; aún en aplicaciones de pequeño tamaño se obtienen beneficios de modelar, sin embargo, es un hecho que entre más grande y más complejo es el sistema, el modelado juega un papel más importante. Esto se debe a una razón simple: "Hacemos modelos de sistemas complejos porque no podemos entenderlos en su totalidad".

Hay límites para el entendimiento de la complejidad. A través del modelado reducimos el ámbito del problema de estudio al enfocar solo un aspecto a la vez.

UML puede ser usado extensivamente en: Recopilación de requerimientos, Análisis de aplicaciones, Diseño de sistemas, en pruebas, en implementación, en reingeniería y prácticamente en cualquier actividad de desarrollo que sea susceptible de ser modelada.

Cabe aclarar que aunque UML es orientado a objetos preferentemente, es útil en cualquier modelo tecnológico ya que es independiente de lenguajes de programación o tecnología determinada.

Esta consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

En otros términos, así como en la construcción de un edificio se realizan planos previo a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema, ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, mismas que facilitan a integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

1.1.1 Beneficios de esta Tecnología

Los beneficios son claros al ocupar este lenguaje de modelamiento:



Mejores tiempos totales de desarrollo (de 50% o más). En la mayoría de organizaciones hoy en día el tiempo que pasa desde que un proyecto arranca hasta que se estabiliza es más del doble de lo planeado originalmente. Con el uso de UML las fases de análisis y diseño consumirán mayor tiempo, pero el tiempo de construcción, implantación y estabilización se reducen drásticamente debido a que no hay correcciones mayores en las fases de mayor impacto de un proyecto.

Mejor calidad. El uso de UML hace indispensable la participación del usuario en la definición de requerimientos y por lo tanto mejora considerablemente el apego del sistema a las necesidades de sus usuarios. El mantenimiento correctivo se reduce drásticamente (hasta un 80% con respecto a un sistema hecho sin metodología). Algo similar ocurre en los proyectos de reingeniería.

Mejor soporte a la planeación y al control de proyectos. Al existir entregables definidos y estandarizados en las distintas fases de un proyecto y al ser éstos revisables y certificables por gente distinta del autor, tenemos que los planes de trabajo pueden ser fácilmente creados y corroborados en avance. Lo que permite tomar decisiones a tiempo.

Mayor independencia del personal de desarrollo. Al tener documentadas las aplicaciones en un lenguaje estándar, podemos mover al personal de una aplicación a otra sin correr altos riesgos y sin depender del conocimiento personal de las aplicaciones.

Mayor soporte al cambio organizacional, comercial y tecnológico. Un modelo permite cuantificar el impacto de un cambio antes de hacerlo y permite ensayar distintos enfoques de solución. Con UML un cambio se puede hacer primero en papel.

Alto reusó. Los productos de un desarrollo pueden ser usados en otro. Se pueden crear componentes reusables que con la difusión y administración adecuadas minimizarán costos y errores.

Minimización de costos. Los puntos antes mencionados tienen un impacto económico que generalmente tiende a ser proporcional al tamaño de la organización.

1.1.2 Elementos de UML

Diagrama de Casos de Uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de *casos de uso* y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen *qué* es lo que debe hacer el sistema, pero no cómo.

Diagrama de Clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Diagramas de Secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

Diagramas de Colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

Diagrama de Estado

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Los diagramas de estado ven a los objetos como máquinas de estado o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito.

Diagrama de Actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

1.1.3 Organización De Los Casos De Uso

Los casos de uso se pueden organizar agrupándolos en paquetes. También se pueden organizar al especificar generalizaciones, relaciones de inclusión y de extensión entre ellos. Estas relaciones se pueden aplicar para separar el comportamiento común o para factorizar las variaciones.

La generalización entre casos de uso es como la generalización entre clases. Significa que un caso de uso hijo hereda el comportamiento y significado del caso de uso padre. El hijo puede agregar o redefinir el comportamiento de su padre y el hijo puede ser sustituido en cualquier parte que el padre aparezca. Por ejemplo, en un sistema bancario, se puede tener el caso de uso validar usuario, el cual es responsable de verificar la identidad del usuario. Además, se podría tener dos hijos especializados para este caso de uso (revisa contraseña y escudriña retina), ambos llegan a ser como válida usuario y pueden ser aplicados en cualquier lugar en el cual valida usuario aparece, aunque cada uno de ellos usa su propio comportamiento. La generalización entre componentes se presenta como una línea sólida dirigida con una flecha abierta grande, del mismo modo que la generalización entre clases.

Una relación entre casos de uso del tipo incluye significa que el caso de uso base explícitamente incorpora el comportamiento de otro caso de uso en la localidad especificada en la base. El caso de uso incluido nunca permanece solo, sólo es instanciado como parte de algún caso de uso mayor que lo incluye. Puedes pensar en incluye como el caso de uso base que extrae el comportamiento de un caso de uso proveedor.

La relación de inclusión se utiliza para evitar describir el mismo flujo de eventos varias veces, al poner el comportamiento común en un caso de uso. La relación incluye es esencialmente un ejemplo de delegación –tomas un conjunto de responsabilidades del sistema y lo captura en un lugar (el caso de uso incluido), entonces le permites a otras partes del sistema (otros casos de uso) incluir la nueva agregación de responsabilidades en cualquier parte que requieras utilizar esa funcionalidad.

En UML, la relación incluye se presenta como una dependencia, estereotipada como include.

Para especificar la localidad en el flujo de eventos en el cual el caso de uso base incluye el comportamiento de otro, simplemente se escribe include seguido del nombre del caso de uso que deseas incluir.

Una relación de extiende entre casos de uso significa que el caso de uso base implícitamente incorpora el comportamiento de otros caso de uso en la localidad especificada indirectamente por el caso de uso extendido. El caso de uso base puede estar solo, pero bajo ciertas condiciones, su comportamiento puede ser extendido por el comportamiento de otro caso de uso. Este caso de uso base puede ser extendido sólo en ciertos puntos, en sus puntos de extensión. Puedes pensar en extiende como la extensión de un caso de uso que empuja el comportamiento de un caso de uso base.

En UML, la relación extiende se presenta como una dependencia, estereotipado como extend. Se pueden listar los puntos de extensión del caso de uso base en un compartimiento extra. Estos puntos de extensión son sólo etiquetas que pueden aparecer en el flujo del caso base.

Existen diferentes programas que apoyan a la creación de casos de uso y desarrollos de sistemas para satisfacer necesidades entre ellos contamos con Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML [3].

1.2 Base De Datos

Todas las empresas requieren almacenar información. Desde siempre lo han hecho. La información puede ser de todo tipo. Cada elemento informativo (nombre, dirección, sueldo, etc.) es lo que se conoce como dato (en inglés data).

Las soluciones utilizadas por las empresas para almacenar los datos son diversas. Antes de la aparición de la informática se almacenaban en ficheros con cajones y carpetas y fichas. Tras la aparición de la informática estos datos se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo, ...).

Además las empresas requieren utilizar aplicaciones informáticas para realizar tareas propias de la empresa a fin de mecanizar a las mismas. Estas aplicaciones requieren manejar los datos de la empresa.

1.2.1 Estructura de una Base de Datos

Las bases de datos están compuestas (como ya se han comentado), de datos y de metadatos. Los metadatos son datos (valga la redundancia) que sirven para especificar la estructura de la base de datos; por ejemplo qué tipo de datos se almacenan (si son texto o números o fechas ...), qué nombre se le da a cada dato (nombre, apellidos,...), cómo están agrupados, cómo se relacionan, de este modo se producen dos visiones de la base de datos:

Estructura lógica. Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando. Es una estructura que permite idealizar a la base de datos. Sus elementos son objetos, entidades, nodos, relaciones, enlaces, que realmente no tienen presencia real en la física del sistema. Por ello para acceder a los datos tiene que haber una posibilidad de traducir la estructura lógica en la estructura física.

Estructura física. Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos (en los metadatos).

1.3 Normalizando La Base De Datos

La normalización es un método de análisis de base de datos para conseguir una base de datos relacional, que respete la integridad referencial, y que no tenga redundancia de datos. Se divide en formas normales.

Hay que destacar que la normalización se puede hacer a nivel completo de la base de datos, o a nivel de tablas o esquemas. La técnica es la misma: analizar el conjunto de campos y en base a eso designar una clave inicial que identifique a un grupo de datos. Por ejemplo si estamos normalizando todo un esquema de facturación podemos partir de los datos del cliente añadiendo la clave del cliente, y según vayamos normalizando nos saldrán todas las tablas y les iremos dando claves primarias nuevas. Si lo que normalizamos es una tabla, el procedimiento es el mismo y ya irán saliendo otras tablas subordinadas si acaso.

1.3.1 Primera Forma Normal (1fn)

Se podría decir que al aplicarla hay que asegurarse de que:

No se permiten vectores de campos en una columna.

Es cuando en un campo de texto metemos varios valores del mismo dominio, Lo típico en estos casos es separar los datos por comas, espacios u otro carácter y después procesarlo mediante la aplicación. Para evitar esto hay que definir una nueva tabla que tendrá el identificador de la tabla de la que parte y el campo multivalorado, haciendo juntos de clave única compuesta (se puede definir otra incremental si se desea, pero el conjunto de los otros dos campos tiene que ser único). Además en esta tabla se puede agregar campos que ayuden a describir el tipo de registro.

No se permiten grupos repetidos en varias columnas.

Esto es una variante de lo anterior: separamos los campos de un mismo dominio en varias columnas, haciendo un grupo difícilmente procesable a la hora de consultarlo.

No se permiten campos nulos

Esta regla es algo discutible, pero tiene su lógica. Para empezar, si un campo va a tener valores nulos, ¿qué proporción de registros tendrán ese campo con valor nulo? En mi opinión esta regla nos ayuda a separar unas entidades de otras, porque si una cantidad de registros tienen unos atributos que otros no, ¿no será que pertenecen a otra clase? [6].

1.3.2 Segunda Forma Normal (2fn)

Una tabla está en segunda forma normal siempre que esté en primera forma normal y todos sus atributos (campos) dependan totalmente de la clave candidato sin ser parte de ella. Viene a ser que, si un campo de la tabla no depende totalmente de una clave única (que pueden ser compuestas), debe sacarse fuera con la parte de la clave principal de la que es dependiente.

1.3.3 Tercera Forma Normal (3fn)

Una tabla está en tercera forma normal siempre que esté en segunda forma normal (y por consiguiente en primera) y todos sus campos no primarios (campos que no forman parte de una clave candidata) dependen únicamente de la clave candidata. Suena como la segunda forma normal, pero es muy distinta: ningún campo que no sea parte de la clave candidata puede depender de otro campo que no sea la clave candidata.

1.3.4 Modelo Entidad Relación

Se desarrollan las tablas en workbench para facilitar la creación de la base de datos y la normalización de la misma base.

Al crear el modelo de la base de dato con la herramienta MySql Workbench, se puede crear de una manera muy fácil y sencilla la conjugación de los objetos y relacionarlos con sus respectivos componentes para obtener una mejor estructura y un almacenamiento más explícito.

1.4 MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU (General Public License). Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración [8].

Es un sistema gestor de bases de datos que se puede encuadrar dentro de la categoría de los programas open-source.

MySQL es un caso particular, pues se trata de un programa de licencia open-source y gratuito pero que, sin embargo, está mantenido por una empresa, MySQL AB, con sede en Suecia.

El código fuente de MySQL está sólo relativamente abierto y disponible para modificaciones, puesto que es la empresa MySQL AB la que contrata y coordina los trabajos de mantenimiento del producto. No obstante, los trabajadores contratados, procedentes de todo el mundo, son usuarios del producto que realizan sus encargos a través de Internet.

1.4.1 Orígenes

El origen de MySQL se remonta a la década de los ochenta. Michael Widenius, también conocido como Monty, un joven programador que realizaba complejas aplicaciones en lenguaje BASIC, al no encontrar un sistema de almacenamiento de archivos que le resultara satisfactorio, pensó en construir el suyo propio.

Años después, en 1995, y en colaboración con David Axmark, Widenius desarrolló un producto que básicamente era el resultado de sus investigaciones, más dos aportaciones nuevas: el uso del lenguaje SQL y la accesibilidad a través de Internet. Así nació MySQL y también la empresa MySQL AB.

1.4.2 Evolución

La evolución de MySQL se produce con las sugerencias de los usuarios, canalizadas por la empresa MySQL AB, que contrata a programadores de todo el mundo para que, a través de Internet, vayan ampliando y mejorando el producto.

Un programa open-source es aquel cuyo código fuente está disponible para los usuarios y abierto a modificaciones.

En el caso de que el programa open-source se pueda adquirir gratis, el usuario debe distribuir también gratis los cambios que pueda realizar, o los productos que pueda desarrollar basándose en el original. En caso contrario, si el usuario desea hacer negocio con el producto, entonces debe adquirir la licencia comercial de pago.

Las versiones, como es costumbre en este tipo de software libre, se van colgando en Internet para que los usuarios puedan disponer de ellas.

Sin embargo, también como es habitual, hay que distinguir entre versiones de prueba o beta y versiones estables o de producción.

Las versiones de prueba o beta están a disposición de los usuarios para ser testadas en busca de posibles mejoras. Por su parte, las versiones estables o de producción son las que ya están preparadas para ser utilizadas a pleno rendimiento.

En la actualidad, 4.1.14 es la última versión estable.

Esta versión dispone de características como gestión de transacciones, integridad referencial (bajo ciertas condiciones), uso de índices, incorporación de librerías, etcétera.

La versión de prueba actualmente disponible es la 5.0, que empieza a incorporar elementos hasta ahora no incluidos en MySQL, como el uso de procedimientos, cursores, triggers o vistas.

Existe una versión incluso más reciente, la 5.1, pero que aún es un proyecto. Esta versión promete mejoras en la seguridad y en la gestión de claves ajenas, aunque todavía no se conoce cómo se plasmarán dichas características en posteriores versiones estables.

Características de MySQL

Aparte de las características que definen MySQL como programa open-source, existen aspectos que lo diferencian de otros productos como, por citar uno conocido, Access. Los atributos a los que hacemos referencia son:

- a) Posibilidad de crear y configurar usuarios, asignando a cada uno de ellos permisos diferentes.
- b) Facilidad de exportación e importación de datos, incluso de la base de datos completa.
- c) Posibilidad de ejecutar conjuntos de instrucciones guardadas en ficheros externos a la base de datos.

Todas estas características, junto con las operaciones más frecuentes que se realizan sobre bases de datos, se estudiarán en los siguientes Apartados. De todas formas, veamos en principio la manera de obtener el producto y, posteriormente, cómo realizar su instalación.

1.4.3 Restricciones en MySQL

En MySQL, como en otros sistemas gestores de bases de datos, existe la posibilidad de incorporar ciertas restricciones y propiedades especiales en las definiciones de los campos de una tabla. Así pues, a continuación vamos a analizar las restricciones más frecuentes, empezando por aquellas de mayor importancia: las claves primarias y ajenas.

1.4.4 Claves Primarias

Para definir la clave primaria de una tabla se usa la expresión PRIMARY KEY, que puede introducirse en la misma línea de definición del campo que hará de clave, o bien al final de la declaración de campos. No obstante, cuando la clave primaria esté formada por más de un campo, será obligatorio definirla al final.

1.4.5 Claves Ajenas

La función de las claves ajenas consiste en garantizar la integridad referencial entre tablas. Supongamos que disponemos de una tabla provincia cuya clave primaria es idprov. Además, tenemos una tabla ciudad que queremos enlazar con la tabla provincia para que quede constancia de la provincia a la que pertenece cada ciudad. Así pues, añadiríamos a la tabla ciudad un campo, denominado también idprov, que haría referencia a la clave primaria de la tabla provincia y que en la tabla ciudad realizaría la función de clave ajena.



La integridad referencial entre ambas tablas se garantizaría si, además de saber a qué provincia corresponde cada ciudad, se asegura que no podrá asignarse un código de provincia a una ciudad si previamente dicha provincia no existe en la tabla provincias.

La clave ajena hace referencia a un campo que se añade a una tabla para hacer enlace con otra.

Lógicamente, para poder hacer referencia a la tabla provincia, ésta debe haber sido declarada previamente.

Como regla general, es conveniente declarar en primer lugar aquellas tablas que no contienen claves ajenas.

Es realmente importante tener en cuenta que los campos enlazados (la clave primaria y la clave ajena) deben ser exactamente del mismo tipo.

De la misma forma que ocurría con las claves primarias, las claves ajenas pueden definirse al final de la tabla.

Así mismo, el campo o campos que forman parte de la clave ajena deben estar indexados, para lo que usaremos la cláusula INDEX y el nombre del campo o campos que forman el índice entre paréntesis. De esa forma, la creación de la tabla ciudad quedaría, de manera definitiva.

Aún queda otra cuestión importante por resolver: en MySQL las tablas pueden ser de varios tipos, entre ellos MyISAM o InnoDB. Aunque más adelante se volverá sobre esta cuestión, es importante saber que las claves ajenas sólo funcionarán si las tablas son de tipo InnoDB, al menos en la versión de MySQL que estamos utilizando.

Es necesario realizar la comprobación, puesto que por defecto MySQL crea las tablas como MyISAM.

El diseño utilizaremos diferentes métodos para satisfacer las necesidades presentadas, teniendo de una manera sencilla los modelos que podemos llevar a cabo para así realizar un esquema de bajo nivel y llevarlo a un nivel programable el cual nos permita el almacenamiento de los datos necesarios para dar de alta casos prácticos. Ocupando así para esta implementación MySQL (sistema de gestión de bases de datos relacional).

Aportando MySQL diferentes programas o aplicaciones que nos facilitan la construcción de modelos, en este caso tenemos el modelo EER o bien llamado Modelo Entidad Relación Extendido, usaremos para diseñar este modelo MySQL Workbench para exportar de una manera simple y eficaz la base de datos la cual podemos visualizar por "queries" activándola con Wamp Server ya que nos permite las conexiones con diferentes servidores [5].

Para la complementación del sistema para la administración de casos prácticos se desarrollaran un interfaz que le permita al usuario ingresar de una manera fácil y segura datos personales para darse de alta y de igual manera sus o su caso práctico, en el desarrollo de estas interfaces ocuparemos aplicaciones que nos proporciona Adobe Systems Incorporated, emplearemos Dreamweaver CS6, Flash Builder 4.7 y Creativa Suite.

En una presentación más contemplada para el usuario, se necesita de una conexión entre la base de datos en MySQL y la interfaz creada el Flash Builder en este procedimiento se considera utilizar PHP, él cual es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Php se instala en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros [11].

1.5 PHP

PHP se centra principalmente en secuencias de comandos del lado del servidor, por lo que se puede hacer cualquier cosa en cualquier otro programa CGI puede hacer, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Pero PHP puede hacer mucho más.

Con PHP no está limitado a resultados en HTML. Habilidades de PHP se incluyen imágenes, archivos PDF y películas Flash (usando libswf y Ming). De igual manera puede enviar fácilmente cualquier texto, como XHTML y cualquier otro archivo XML. PHP puede autogenerar estos archivos y guardarlos en el sistema de archivos, en lugar de imprimirlo, formando una caché del lado del servidor para el contenido dinámico.

Una de las características más fuertes y destacable de PHP es su soporte para una amplia gama de bases de datos. Escribir una página web habilitada para la base de datos es muy sencillo con una de las extensiones de bases de datos específicas (por ejemplo MySQL).

PHP cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También puede abrir sockets de red e interactuar con cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos WDDX entre lenguajes de programación web. Hablando de interconexión, PHP tiene soporte para la instanciación de objetos Java y su utilización de forma transparente como objetos PHP [16].

1.6 HTML5

La palabra WEB es el término más usado para referirse al World Wide Web que es la Red Mundial de páginas o Documentos de texto entrelazados. Un documento entrelazado no es más que un documento que contiene enlaces a otros documentos o páginas de texto, donde se amplía o hay más información relacionada.

Las páginas web son archivos con extensión .asp o .aspl. Cuando el navegador lee este archivo interpreta el código escrito en él y nos muestra los resultados en la pantalla o en los parlantes.

Los sitios web están escritos en código HTML (Hyper Text Markup Language), o dinámicamente convertidos a éste, y se acceden aplicando un software conveniente llamado navegador web, también conocido como un cliente HTTP. Los sitios web pueden ser visualizados o accedidos desde

un amplio abanico de dispositivos con conexión a Internet, como computadoras personales, portátiles, PDAs, y teléfonos móviles.

Es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella.

HTML5 está siendo desarrollado por Ian Hickson de Google Inc. y David Hyatt de Apple Inc. junto con todas las personas que participan en Web Hypertext Application Technology Working Group.

HTML5 nos permite una mayor interacción entre nuestras páginas web y contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar nuestro diseño básico.

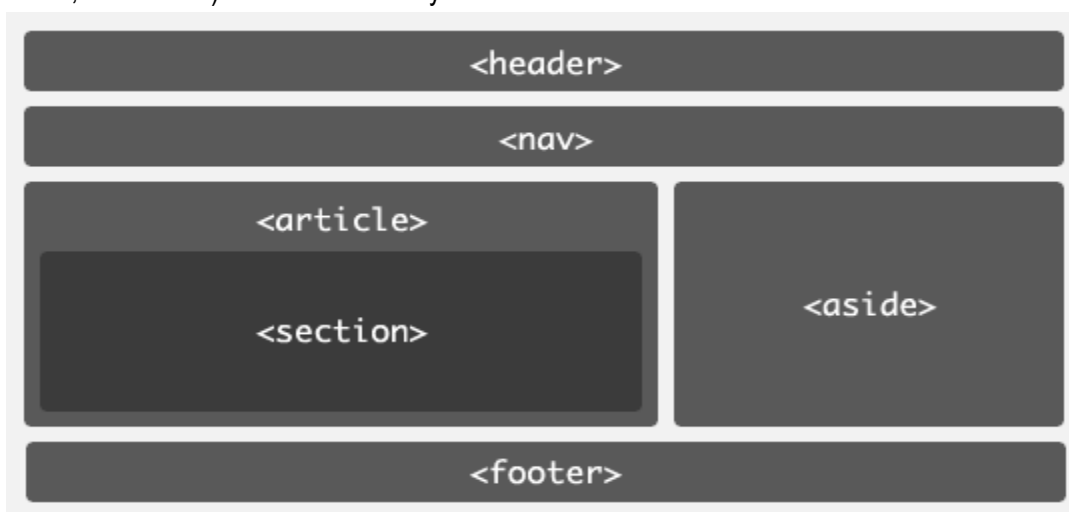


Ilustración 1.6.1.6.1 HTML

Esta nueva versión se basó en el diseño más común de las páginas web alrededor del mundo para llegar a un estándar de etiquetas que realicen las mismas tareas de manera más rápida y eficiente, he aquí algunos ejemplos:

Un nuevo diseño para páginas web, reflejado en las etiquetas `<header>`, `<footer>`, `<nav>`, `<section>`, `<article>` las cuales están destinadas a remplazar la necesidad de tener una `<div>` para cada parte de la página, y en cambio, tener etiquetas específicas para ello.

La nueva etiqueta `<video>` para insertar un reproductor de video, mejorando el reproductor antiguo utilizado por la etiqueta `<embed>` y evitándonos la pena de insertar el código de `<object>`, así como eliminar la necesidad del Flash Player para reproducir videos (lo que nos lleva a un ahorro en la cantidad de memoria utilizada).

Una nueva tag `<audio>` para insertar audio en nuestro sitio web, remplazando la vieja etiqueta `<embed>` con las mismas cualidades de la etiqueta anterior.

Una etiqueta <canvas> para manejo de gráficos en internet, sea para dibujar vectores o hacer animaciones.

Basta con volver a definir nuestras primeras líneas de un archivo HTML para comenzar a usar la nueva versión. De la siguiente manera:

```

<!DOCTYPE html>
<html>
<head>...</head>
<body>
<header><!-- El encabezado a utilizar --></header>
<nav><!-- Nuestro menú de navegación --></nav>
<article>
  <section>...</section>
</article>
<footer>...</footer>
</body>
</html>

```

Ilustración 1.61.6.2 CODIGO HTML

1.7 Macromedia Dreamweaver

Creador y editor de páginas web tipo WYSIWYG creado por Macromedia (actualmente de Adobe Systems)[15].

Es la aplicación de este tipo más usada en el sector de diseño y programación web. Posee, como toda la línea Macromedia/Adobe, excelentes funcionalidades e integración con otras herramientas.

La aplicación permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado. Permite la conexión a un servidor, a base de datos, soporte para programación en ASP, PHP, Javascript, cliente FTP integrado, etc. [7].

1.8 Programas similares

Se han encontrado diferentes sistemas para la creación de exámenes o registros

1.8.1 EASYPLANEX

Similares para casos prácticos para perfiles profesionales, mencionando algunos de ellos se encuentra también software para el proceso de evaluación de proyectos o casos prácticos, por ejemplo EASYPLANEX es una aplicación de software que proporciona una solución integral para la evaluación y optimización de proyectos de capital, la herramienta es bilingüe Inglés / Español,



incluso existe una versión disponible - EasyPlanEx Free - que incluye casos de muestra que pueden consultarse en el portal [12].

1.8.2 TEACH2000

Así también se cuenta con el software TEACH2000, que es un programa diseñado para ayudar a los profesores de lenguas extranjeras a crear evaluaciones para sus alumnos, permite añadir nuevas preguntas a la evaluación o crear una nueva partiendo desde cero, puede crear un archivo HTML con los cuestionarios creados de forma que tan sólo subiéndolo al servidor, los alumnos tengan acceso a éste [13].



Capítulo 2. ANÁLISIS Y DISEÑO DEL SISTEMA

2.1 Introducción al Análisis del Sistema

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de manejarla con métodos y procedimientos más adecuados. Se puede dividir en dos: el análisis de sistemas que comprende la planificación, el levantamiento inicial de información y el estudio en detalle del sistema actual para luego recomendar o estructurar las especificaciones necesarias para el nuevo sistema; y el diseño que consiste en llevar a cabo el sistema por medio de la clasificación y empleo de la información de manera que se pueda ofrecer una alternativa mucho más viable [1].

En pocas palabras; “El análisis especifica qué es lo que el sistema debe hacer. El diseño establece cómo alcanzar el objetivo”. Se deben utilizar metodologías que permiten ver los sistemas en base a sus procesos, por lo menos en sistemas de procesado por lotes o secuencial. Un ejemplo de ello es la metodología estructurada. Existen otras metodologías como la orientada a objetos.

2.2 Análisis del sistema

Un proceso que debemos considerar en el desarrollo del sistema SICP es el análisis, llevando investigaciones y contemplando los diferentes puntos de acción que se puedan presentar se ha completado, se logra utilizar las Especificaciones de Casos de Uso para una mejor comprensión de información.

2.2.1.1 Especificación de Casos de Uso

2.2.1.1 Registro Usuario

1. **Nombre:** Registrar Usuario
2. **Autor:** Elizabeth Cerón Aquino
3. **Fecha:** 10-enero-2015
4. **Descripción:** El Usuario introducirá sus datos para ser registrado en el sistema.
5. **Actores:** Usuario
6. **Pre-condiciones:** El usuario debe tener una matrícula escolar y correo electrónico.
7. **Flujo básico:**
 - 7.1 Entrar a la página de Registro de Usuario.
 - 7.2 Proporcionar la información a conforme la pide el formulario.
 - 7.3 Guardar los datos.
8. **Flujo alternativo:** El sistema valida los datos si son correctos de lo contrario emitirá un mensaje de error.
9. **Flujo de excepción:**

- 9.1 El sistema no puede guardar datos si no se ha completado el formulario teniendo en cuenta las opciones que son obligatorias.
- 9.2 El sistema no guardara datos por que la Matricula ya existe.
- 9.3 El sistema no guardara datos porque todo está en blanco.
- 10. **Post-condiciones:** Condiciones y operaciones concluidas exitosamente.

2.2.1.2 Registro Caso Práctico

- 1. **Nombre:** Registrar Caso
- 2. **Autor:** Elizabeth Cerón Aquino
- 3. **Fecha:** 10-enero-2015
- 4. **Descripción:** El usuario proporcionara la información del Caso Práctico.
- 5. **Actores:** Usuario
- 6. **Pre-condiciones:** El Usuario debe de tener implementado ya el Caso Práctico.
- 7. **Flujo básico:**
 - 7.1 Iniciar sesión proporcionando Usuario (Matricula) y contraseña.
 - 7.2 Entrar a la página registro de Caso Practico.
 - 7.3 Proporcionar la información del Caso Practico conforme esta creado el formulario.
 - 7.4 Guardar los datos.
- 8. **Flujo alternativo:** El sistema valida los datos si son corrector de lo contrario emitirá un mensaje de error.
- 9. **Flujo excepción:**
 - 9.1 El sistema no guardara los datos del Caso Practico si no ha completado el formulario teniendo en cuenta las opciones que son obligatorias.
 - 9.2 El sistema no guardara datos porque todo está en blanco.
- 10. **Post-Condicion:** Condiciones y operaciones concluidas exitosamente.

2.2.1.3 Administrar Usuario

- 1. **Nombre:** Administrar Usuario
- 2. **Autor:** Elizabeth Cerón Aquino
- 3. **Fecha:** 10-enero-2015
- 4. **Descripción:** Se le permitirá Insertar y Modificar usuario el cual usara esa cuenta.
- 5. **Actores:** Usuario
- 6. **Pre-condiciones:**
 - 6.1 El usuario deberá a ver proporcionado los datos del Registro Usuario correctamente.
- 7. **Flujo básico:**
 - 7.1 El usuario deberá de ingresar los datos correctos en el Registro de usuario para Insertar se realizara en el punto 8.1.
 - 7.2 El sistema muestra en el Perfil de Usuario la opción de modificar se realizará en el punto 8.2.

8. Flujo Alternativo:

8.1 Insertar

- 8.1.1 El formulario de Registro de Usuario se debe encontrar abierto con los datos correctamente.
- 8.1.2 El usuario dará click en el botón Guardar.

8.2 Modificar

- 8.2.1 Se mostrara en pantalla la página de Modificar Usuario.
- 8.2.2 El usuario podrá cambiar datos respetando los criterios en el formulario.
- 8.2.3 El usuario deberá dar click en el botón de Actualizar.

9. Flujo de excepción:

- 9.1 El sistema no guardara la información en la base de datos por que algún campo obligatorio haya quedado vacío.
- 9.2 El sistema no guardara la información en la base de datos por que los datos no concuerden con los estructura del formulario.
- 9.3 El sistema no mandara datos a la base de datos porque no hay conexión a la base de datos.

10. Post-condiciones: Las condiciones y las operaciones sean concluidas correctamente.

2.2.1.4 Registro Caso Práctico

- 1. **Nombre:** Administrar Caso Practico
- 2. **Autor:** Elizabeth Cerón Aquino
- 3. **Fecha:** 10-enero-2015
- 4. **Descripción:** Se le permitirá Insertar, Modificar, Buscar o Eliminar Casos Prácticos.
- 5. **Actores:** Usuario
- 6. **Pre-condiciones:**
 - 6.1 El usuario deberá a ver proporcionado los datos del Registro Usuario correctamente.
- 7. **Flujo básico:**
 - 7.1 El sistema muestra en el Perfil de Usuario las siguientes opciones:
 - 7.1.1 **Insertar** desarrollo 8.1.
 - 7.1.2 **Modificar** desarrollo 8.2.
 - 7.1.3 **Buscar** desarrollo 8.3.
 - 7.1.4 **Eliminar** desarrollo 8.4.

8 Flujo Alternativo:

8.1 Insertar

- 8.1.1 El usuario deberá de haber Iniciado Sesión.
- 8.1.2 Abrir Perfil de Usuario.
- 8.1.3 Dar click en Insertar Nuevo Caso Práctico
- 8.1.4 Se abrirá la página de Registro de Caso Práctico punto 2.2.1.2
- 8.1.5 Dar click en el botón de Guardar datos.

8.2 Modificar



- 8.2.1 El usuario deberá de haber Iniciado Sesión.
- 8.2.2 Abrir Perfil de Usuario.
- 8.2.3 Dar click en el Caso Práctico a Modificar
- 8.2.4 Dar click en el apartado de Modificar.
- 8.2.5 Se abrirá la página de Modificar Caso Práctico
- 8.2.6 Se ingresaran los cambios de datos del Caso Practico en los apartados correspondientes en el formulario.
- 8.2.7 Dar click en el botón de Actualizar datos.

8.3 Buscar

- 8.3.1 El usuario tendrá la opción de buscar cualquier Caso Practico solo con colocarse en el apartado de Buscar.
- 8.3.2 Escribir palabra clave.
- 8.3.3 Dar click en el botón de Buscar.
- 8.3.4 Se mostraran los resultados en pantalla.
- 8.3.5 Seleccionar un Caso Práctico.
- 8.3.6 Se mostrara en pantalla el Caso Practico.

8.4 Eliminar

- 8.4.1 El usuario deberá de haber Iniciado Sesión.
- 8.4.2 Abrir Perfil de Usuario.
- 8.4.3 Dar click en el Caso Práctico a Eliminar
- 8.4.4 Seleccionar del menú el apartado de Eliminar
- 8.4.5 Se abrirá un mensaje "Seguro de Eliminar el Registro"
- 8.4.6 Aceptar para su eliminación.
- 8.4.7 Dar click en el apartado de Perfil de Usuario o Página Principal.

9 Flujo de excepción:

- 9.1 El sistema no guardara la información en la base de datos por que algún campo obligatorio haya quedado vacío.
- 9.2 El sistema no guardara la información en la base de datos por que los datos no concuerden con los estructura del formulario.
- 9.3 El sistema no mandara datos a la base de datos porque no hay conexión a la base de datos.

10 **Post-condiciones:** Las condiciones y las operaciones sean concluidas correctamente.

2.2.2 Diagrama de Casos de Uso

Considerando para el análisis del sistema la participación de Visual Paradigm los casos de usos para un mejor entendimiento y una estructuración más eficiente para los usuarios.

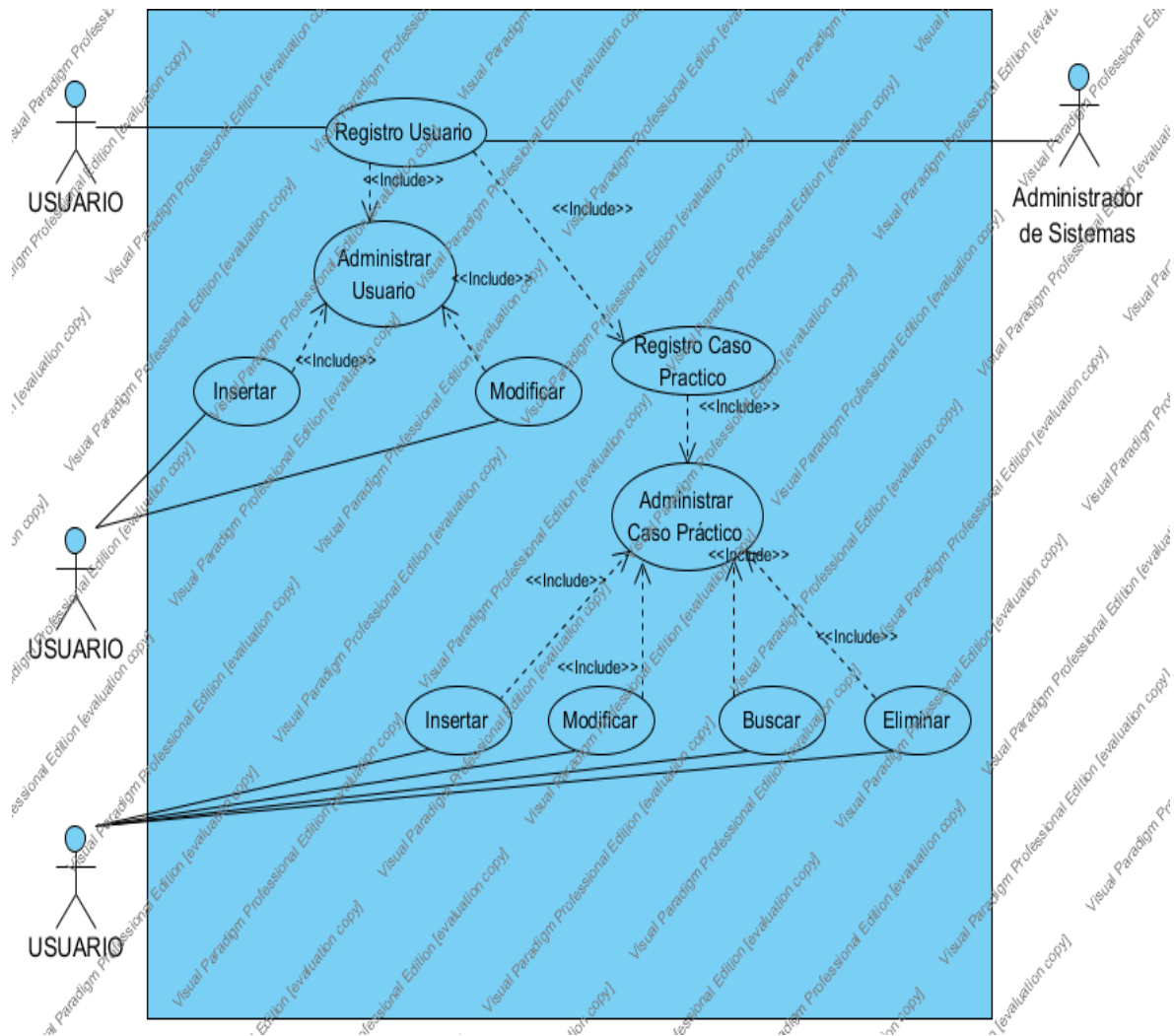


Ilustración 2.2.2.1 Caso de Uso

El Caso de Uso muestra el funcionamiento general del cómo debe de ser el Registro de Usuario, Registro del Caso Practico, cuando se validan los datos si se Inserta, Modifica, Elimina o se Consulta, siempre y cuando el Usuario o el Caso Practico sean del mismo Usuario.

El usuario la primera interacción que tendrá al entrar a la herramienta de Casos Prácticos será en la primera página y en la cual tendrá que iniciar sesión, si es la primera ocasión que ingresara necesitara registrarse primero antes de iniciar sesión

El usuario tendrá que proporcionar datos personales para Iniciar Sesión y posteriormente Registrar el Caso Practico, los datos consisten en datos que el usuario conoce prácticamente de memoria y son fáciles de recordar.



Para poder Iniciar Sesión, se va a presentar un formulario donde el Usuario tendrá que escribir su matrícula en el usuario y la contraseña que proporcione en el Registro de Usuario. Registro de Caso Practico consiste en que el usuario ingrese los datos que complementan el inicio del registro del Caso Practico.

El Usuario podrá consultar los Casos Prácticos ya registrados pero no tendrá ningún derecho de Modificarlo o Eliminarlo, esta consulta se implementa para auxiliar a los usuarios que necesiten información respecto algunos temas.

El Usuario solo podrá modificar el o los Casos Prácticos que el mismo registró en el sistema o podrá modificar los datos de usuario, con motivo de tener una base de datos actualizada y a la vez mantener la privacidad del Usuario y del Caso Practico.

El Usuario solo tendrá derecho de eliminar el o los Casos Prácticos que registra o la cuenta de usuario que el creo siempre y cuando tenga en cuenta la acción que conlleva la eliminación de estos objetos.

Utilizando la herramienta UML Visual Paradigm, construimos los casos de uso, diagrama de clases y diagrama de secuencias:

El modelo de casos de uso se refleja cómo es la interacción de los usuarios y los objetos que son relacionados con la administración de Casos Práctico.

2.2.3 Diagrama de Clases

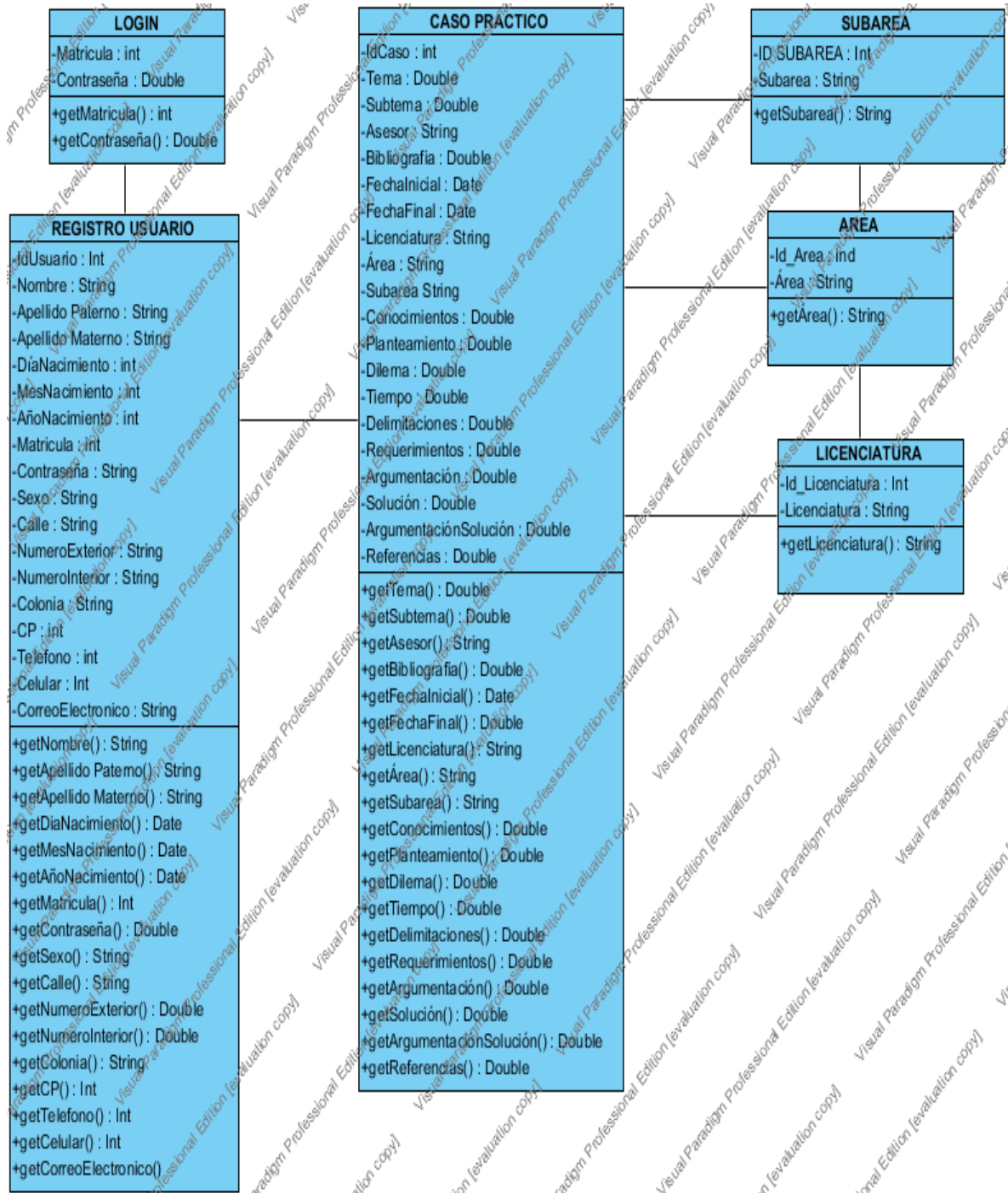


Ilustración 2.2.3.1 Diagrama de Clases

En UML Visual Paradigm completamos el Diagrama de Clases que está compuesto por las clases básicas que contienen en resumen los objetos que se necesita para tener una herramienta que administre Casos Prácticos.

2.2.4 Diagrama de Secuencias

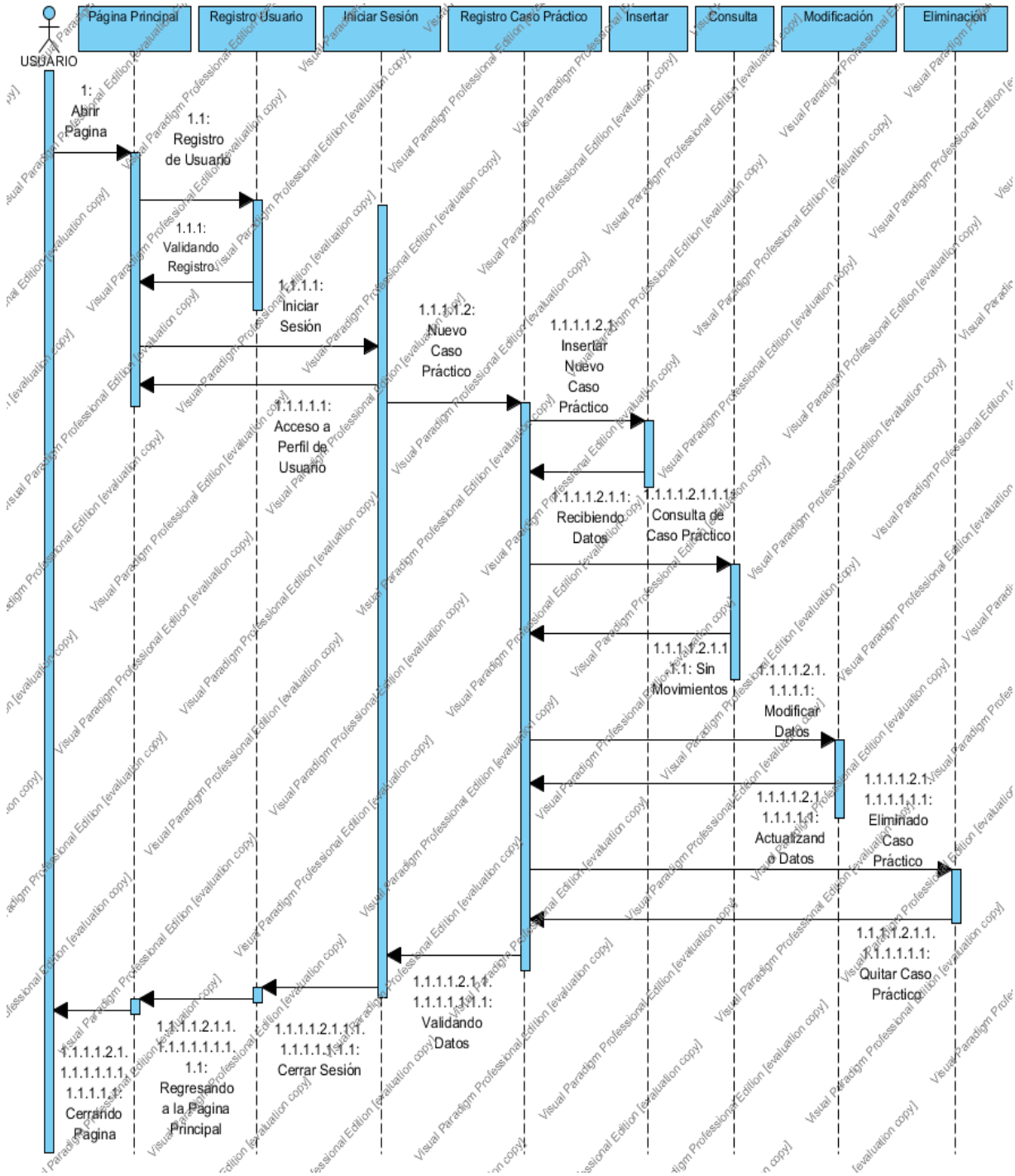


Ilustración 2.2.4.1 Diagrama de Secuencias

Consideramos el Diagrama de Secuencias de la forma como se medirían las secuencias del uso de cada paso para llevar a cabo desde el Registro de Usuario hasta la Inserción, Modificación, Consulta, y Eliminación que se requiera para el manejo del Casos Practico.

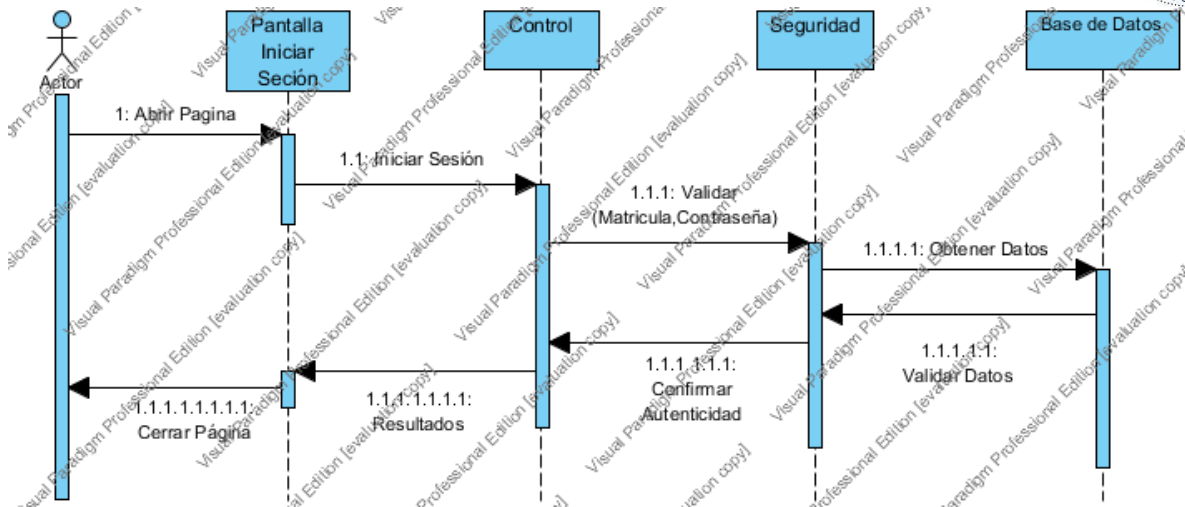


Ilustración 2.2.4.2 Iniciar Sesión

El Diagrama de Secuencia para Iniciar Sesión es la validación de los datos en el Registro de Usuario para poder Iniciar Sesión e ingresar a su Perfil de Usuarios, la interfaz tal como la base de datos proceden una secuencia de pasos que tienen que realizar para poder Ingresar al Perfil de Usuario.

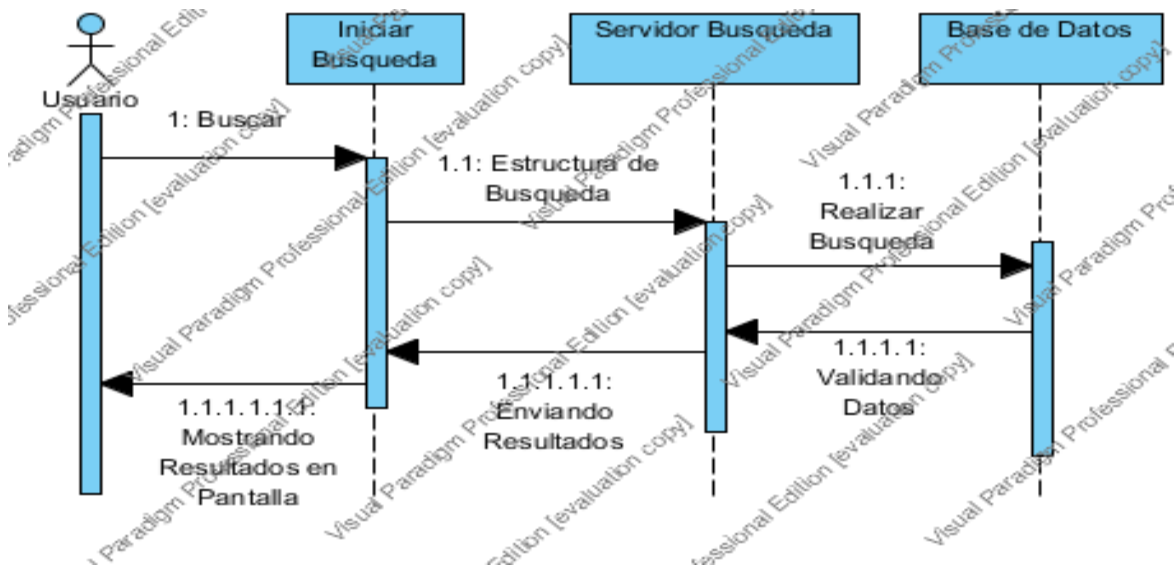


Ilustración 2.2.4.3 Búsqueda

Para el usuario debe de ser una forma muy fácil de Buscar los Casos Prácticos y sea por una Palabra Clave o por Categorías, la secuencia de pasos para poder obtener una búsqueda exitosa es con los pasos que se muestran en el Diagrama de Secuencias Búsqueda, los cuales muestran desde la interacción del Usuario con la Interfaz hasta la base de datos donde se encuentra almacenado los Casos Prácticos.

2.3 DISEÑO DE LA BASE DE DATOS

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos, quizá las más fuertes sean: [3]

La velocidad de acceso

El tamaño de la información

El tipo de la información

Facilidad de acceso a la información

Facilidad para extraer la información requerida

El comportamiento del manejador de bases de datos con cada tipo de información.

2.3.1 Modelo Entidad Relación

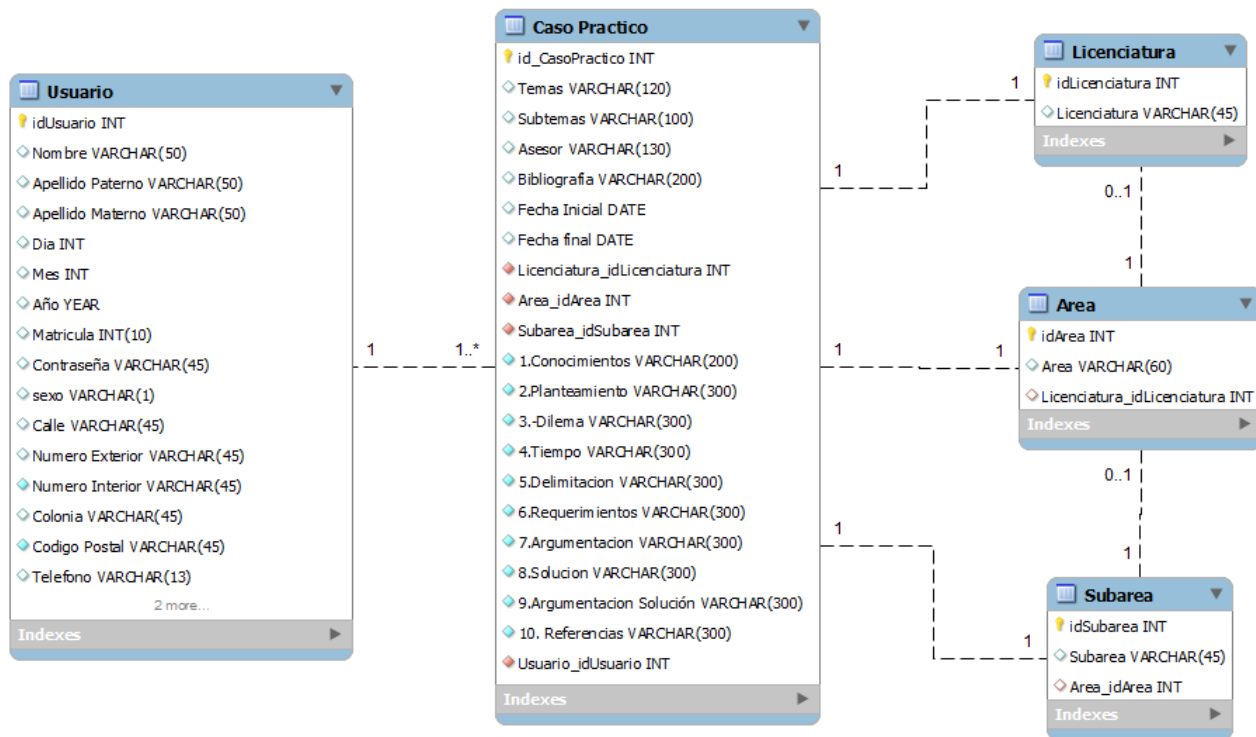


Ilustración 2.3.1.1 Modelo Relacional

Existen muchas formas de clasificar a las bases de datos. Una de ellas es según el contenido:

2.4 La Base de Datos Implementada

Obteniendo la Base de Datos en Workbench se exporta en sql a PhpMyAdmin con tal herramienta se crean las pruebas de Insertar diferentes registros y asegurarse de que el proceso se esté llevando correctamente.

2.4.1 Tabla de Registro de Usuario

The screenshot shows the phpMyAdmin interface for the 'usuario' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	idUsuario	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
2	Nombre	varchar(50)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
3	Apellido Paterno	varchar(50)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
4	Apellido Materno	varchar(50)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
5	Dia	int(11)			Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
6	Mes	int(11)			Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
7	Año	year(4)			Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
8	Matricula	int(10)			Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
9	Contraseña	varchar(45)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
10	sexo	varchar(1)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
11	Calle	varchar(45)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
12	Numero Exterior	varchar(45)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
13	Numero Interior	varchar(45)	latin1_swedish_ci		No	Ninguna		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
14	Colonia	varchar(45)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
15	Codigo Postal	varchar(45)	latin1_swedish_ci		No	Ninguna		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
16	Telefono	varchar(13)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
17	Celular	varchar(13)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más
18	Correo_Electronico	varchar(50)	latin1_swedish_ci		Sí	NULL		Cambiar, Eliminar, Primaria, Único, Índice, Espacial, Más

Ilustración 2.4.1.1 Tabla Registro de Usuarios

La tabla de usuario ya implementada bajo MySql Workbench esta complementada con todos los objetos que se necesitan de los usuarios para obtener una información clara y correcta, en la cual empezamos con su nombre seguido de sus apellidos así mantendremos que el usuario es único, ya que tiene un ID que lo identifica, lo sigue su fecha de nacimiento con la cual es más claro para que no se multipliquen los usuarios, matricula la cual es identificada de la Universidad y una contraseña que es con la que van a acceder, posteriormente para llenar el registro, la dirección se compone por calle, numero exterior e interior, colonia, CP y para localizarlo teléfono, celular y correo electrónico.

2.4.2 Tabla Registro de Caso Práctico

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predefinido	Extra	Acción
1	id_CasoPractico	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Único Índice Más
2	Temas	varchar(120)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
3	Subtemas	varchar(100)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
4	Asesor	varchar(130)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
5	Bibliografia	varchar(200)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
6	Fecha Inicial	date			Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
7	Fecha final	date			Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
8	Licenciatura_idLicenciatura	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
9	Area_idArea	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
10	Subarea_idSubarea	int(11)			Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
11	1.Conocimientos	varchar(200)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
12	2.Planteamiento	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
13	3.-Dilema	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
14	4.Tiempo	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
15	5.Delimitacion	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
16	6.Requerimientos	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
17	7.Argumentacion	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
18	8.Solucion	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
19	9.Argumentacion Solución	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
20	10. Referencias	varchar(300)	latin1_swedish_ci		Sí	NULL		Cambiar Eliminar Primaria Único Índice Más
21	Usuario_idUsuario	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más

Ilustración 2.4.2.1 Tabla Registro Caso Practico

Tabla donde se almacenaran los datos del Registro del Caso Practico mostrada desde phpMyAdmin, donde el Usuario posteriormente de llenar en la interfaz se guardaran en el orden, para que este registro este completo y se almacene necesita los siguientes datos, Tema, subtema, Asesor, Bibliografía, Fecha Inicial y Fecha Final la cual es donde se inicia el Caso Práctico y se finaliza, heredara el id_Usuario ya que el Caso Practico solo pertenece a un solo usuario, Id_Licenciatura, el Id_Area y el Id_Subarea los cuales complementan el Registro.

2.4.3 Tabla Catalogo Licenciatura



Ilustración 2.4.3.1 Tabla Licenciatura

La tabla Licenciatura será llenada por el Administrador ya que será un catálogo para facilitar el llenado del registro de Caso Practico, donde se proporcionaran las Licenciaturas e Ingenierías que se encuentren activas.

2.4.4 Tabla Catalogo Área

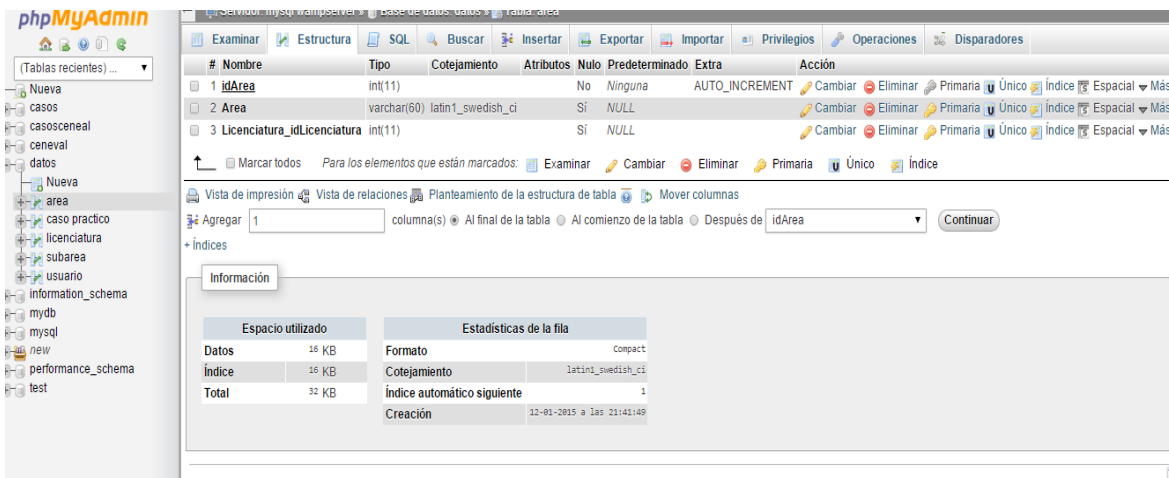


Ilustración 2.4.4.1 Tabla Área

La Tabla Áreas al igual que la tabla Licenciatura se convierte en un catálogo solo que tendrá las áreas que complementa a la licenciatura.

2.4.5 Tabla Catalogo Subárea

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	idSubarea	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Único Índice Espacial Más
2	Subarea	varchar(45)	latin1_swedish_ci		Si	NULL		Cambiar Eliminar Primaria Único Índice Espacial Más
3	Area_idArea	int(11)			Si	NULL		Cambiar Eliminar Primaria Único Índice Espacial Más

Espacio utilizado		Estadísticas de la fila	
Datos	16 KB	Formato	Compact
Índice	16 KB	Cotejamiento	latin1_swedish_ci
Total	32 KB	Índice automático siguiente	1
		Creación	12-01-2015 a las 21:41:49

Ilustración 2.4.5.1 Tabla Sub Áreas

La Tabla Sub Áreas al igual que la tabla Licenciatura y la tabla Áreas se convierte en un catálogo solo que tendrá las sub Áreas que complementa a las Áreas.



Capítulo 3. DISEÑO DE LA INTERFAZ DE USUARIO

3.1 Conceptos Generales

La Interfaz de Usuario, es implementada para un sistema en un conjunto de elementos hardware y software de una computadora que presentan información al usuario y le permiten interactuar con la información y con la computadora. También se puede considerar parte de la interfaz de usuario la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software.

Si la interfaz de usuario está bien diseñada, el usuario encontrará la respuesta que espera a su acción. Si no es así puede ser frustrante su operación, ya que el usuario habitualmente tiende a culparse a sí mismo por no saber usar el objeto.

Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos. Es por ello que no existe una interfaz válida para todos los usuarios y todas las tareas. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecúe a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

El modelo permite explicar o predecir comportamientos del sistema y tomar las decisiones adecuadas para modificar el mismo. Los modelos subyacen en la interacción con las computadoras, de ahí su importancia.

3.2 Balsamiq Mockups

Balsamiq maquetas es una herramienta wireframing rápida que le ayuda a trabajar más rápido y más inteligente. Se reproduce la experiencia de dibujar en una pizarra, pero el uso de un ordenador, son una forma más fácil de visualizar las funciones para construcción de los formularios para la Interfaz.[17]



Ilustración 3.2.1 Balsamiq Mockups

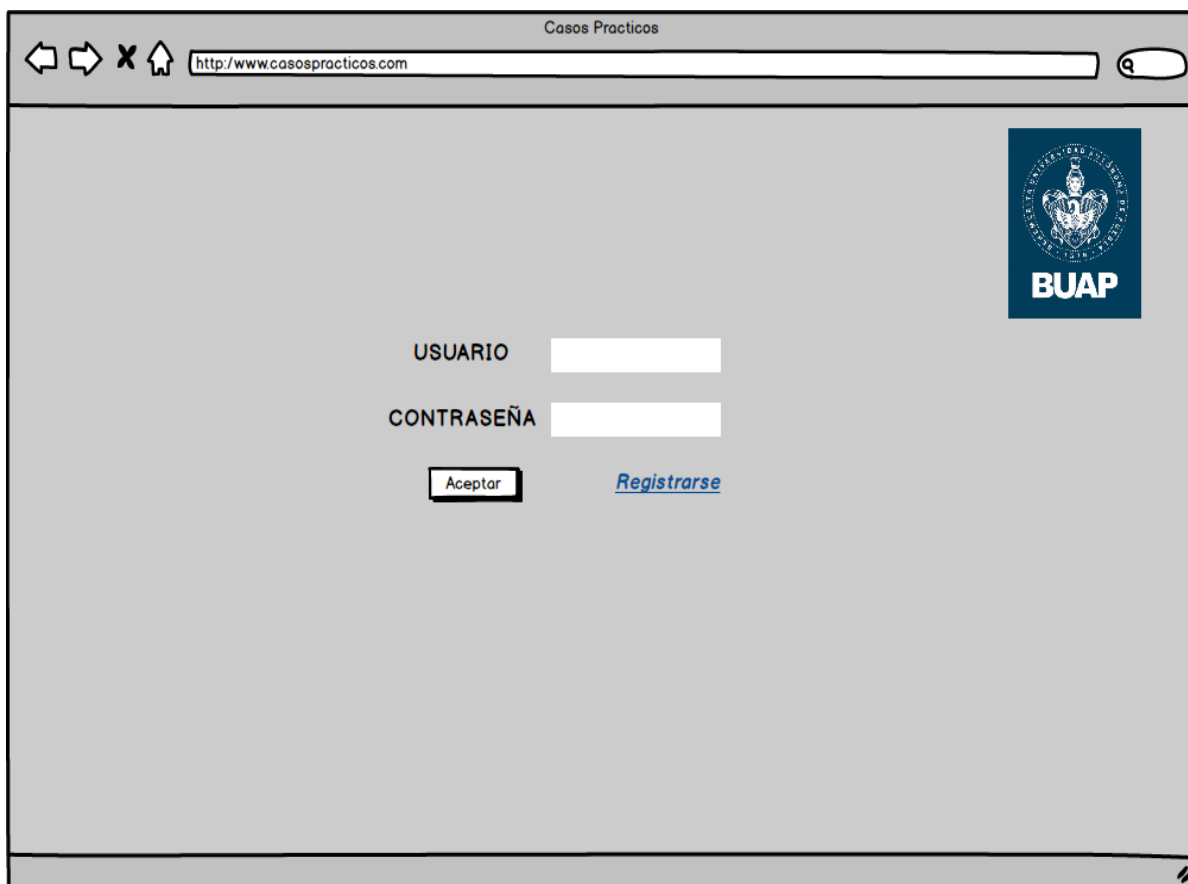


Ilustración 3.1.2 Login

La presentación del Login es una página simple en la cual podemos visualizar los simples componentes como Usuario y Contraseña, esta información la proporcionara el usuario respecto a un registro, si no cuenta con el registro, tendrá que dar click en la opción Registrarse para poder acceder a la página de Registro de Usuario y proceder con el proceso que sería proporcionar el Usuario y Contraseña para luego dar Aceptar, este prototipo fue de gran ayuda por lo cual se ve mucho espacio vacío y se vieron diferentes opciones para tener una mejor visualización de la página de Login.

Registro Usuario

http://www.registrousuariio.com

REGISTRO DE USUARIO

CASOS PRACTICOS

BUAP

Matricula

Contraseña

Nombre

Apellido Paterno

Apellido Materno

Fecha de Nacimiento Día Mes Año

Sexo Femenino Masculino

Calle

Numero Ext. Numero Int.

Colonia Codigo postal

Telefono Local Celular

Correo Electronico

Ilustración 3.1.3 Registro de Usuario

Cada Usuario tendrá que proporcionar una serie de datos para que de esa manera se pueda generar el Registro de Usuario y posteriormente puedan ir al Login para poder acceder, como podemos visualizar la serie de datos es información solo del usuario la cual tiene que ser verdadera, con Balsamiq se crea una forma fácil para poder ver los espacios, la organización de información y los colores.



Ilustración 3.1.4 Registro Casos Prácticos

En el caso de Registro de Casos prácticos de igual manera con los formularios anteriores se nota que falta mejor visibilidad y como apoyo se obtiene una idea básica que se prolongó hasta concluir un estilo moderno y sencillo para que el Usuario se sienta cómodo y satisfecho. La combinación de datos es más saturada que en los otros formularios y contienen menos espacios entre cada objeto, por lo cual este boceto es de gran ayuda.

Evaluación de Casos
 http://www.evaluacioncasos.com

EVALUACION DEL REGISTRO CASO PRACTICO

1.- Conocimientos y Habilidades que se Pretende Evaluar
 2.- Planteamiento del Problema
 3.- Dilema a Resolver
 4.- Tiempo de Resolución
 5.- Delimitación de la Orientación Metodología para Resolver el Caso
 6.- Requerimiento para la Propuesta de Solución
 7.- Argumentación del Caso Practico
 8.- Solución del Caso Practico
 9.- Argumentación de la Solución de Caso
 10.- Referencia Consultadas

Ilustración 3.1.5 Complemento del Registro del Caso Práctico

Con la Evaluación de Registro de Casos prácticos este formulario muestra de una manera simple el ordenamiento que debe de tener cada pregunta y que complementan el registro del Caso Practico simplemente son complementos pero pueden ser extensos o cortos por lo cual se necesitan cuadro de textos amplios para un mejor llenado de información.

CASOS PRÁCTICOS PARA PERFILES PROFESIONALES

Registro Casos Practicos

Consulta Casos Practicos

Iniciar Sesión

REGISTRO USUARIOS

Nombre:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Día:	<input type="text" value="1"/>
Mes:	<input type="text" value="1"/>
Año:	<input type="text" value="1960"/>
Matrícula:	<input type="text"/>
Contraseña:	<input type="text"/>
Sexo:	<input type="text" value="Masculino"/>
Calle:	<input type="text"/>
Número Exterior:	<input type="text"/>
Número Interior:	<input type="text"/>
Colonia:	<input type="text"/>
Código Postal:	<input type="text"/>
Teléfono:	<input type="text"/>
Celular:	<input type="text"/>
Correo Electrónico:	<input type="text"/>

Contacto:

Telefonos: 53-22-92-00 Ext. 7023 ó 7027

Email: acuerdo286lic@ceneval.edu.mx

El registro de Usuario es el más importante ya que será la función principal para poder dar de alta los Casos Prácticos y este registro está compuesto por los datos personales del usuario desde su nombre hasta su correo electrónico. Posteriormente se dará guardar para validar que los datos estén completados de una manera correcta y a su vez se dé el inicio de sesión.

CASOS PRÁCTICOS PARA PERFILES PROFESIONALES

Registro Casos Practicos

Consulta Casos Practicos

Iniciar Sesión

REGISTRO DE CASO PRACTICO

Área	<input type="text" value="Selecciona"/>
Subárea	<input type="text" value="Selecciona"/>
Licenciatura	<input type="text" value="Selecciona"/>
Temas:	<input type="text"/>
Subtemas:	<input type="text"/>
Asesor:	<input type="text"/>
Bibliografía:	<input type="text"/>
Fecha Inicial:	<input type="text" value="dd/mm/aaaa"/>
Fecha Final:	<input type="text" value="dd/mm/aaaa"/>

1.Conocimientos:

2.Planteamiento:

3.Dilema:



Registro del Caso Práctico se inicia con la información que se relaciona con las diferentes áreas, se va llenando conforme se solicita, los campos de Licenciatura, Área y Subárea están conformados por campos de opciones para que al usuario seleccione de una manera más rápida, el Tema y Subtema son datos que debe de tener el Caso Practico, el Autor es el que creador del mismo, se debe de tener la fecha inicial, más los 10 números consecuentes.

3.Dilema:
4.Tiempo:
5.Delimitación:
6.Requerimientos:
7.Argumentación:
8.Solución:
9.Argumentación Solución:
10.Referencias:
Cancelar
Guardar

Contacto:

Telefono: 53-22-92-00 Ext. 7023 ó 7027

Email: acuerdo286lic@ceneval.edu.mx

Continuando con el llenado ya finalizando tendremos un botón de guardar el cual validara la información para guardarla en la base de datos.



CASOS PRÁCTICOS PARA PERFILES PROFESIONALES

Registro Casos Practicos

Consulta Casos Practicos

Iniciar Sesión

INICIO DE SESIÓN

Matricula:

Contraseña:

Contacto:

Telefono: 53-22-92-00 Ext. 7023 ó 7027

Email: acuerdo286lc@ceneval.edu.mx

Diseño de la interfaz de iniciar sesión, tomando la matrícula y la contraseña ya registrada en los registros anteriores, los datos tienen que coincidir para poder ingresar a la cuenta de usuario.

CASOS PRACTICOS PARA PERFILES PROFESIONALES

Registro Casos Practicos

Consulta Casos Practicos

Cerrar Sesión

CONSULTAS

Licenciatura_idLicenciatura	Usuario_idUsuario	Area_idArea	Subarea_idSubarea	Temas	Subtemas
15	2	9	11	ESPECIALISTA EN MEDICINA FAMILIAR	EVALUACION DE LA AUTOESTIMA

Contacto:

Telefono: 53-22-92-00 Ext. 7023 ó 7027

Email: acuerdo286lc@ceneval.edu.mx

Se crea la Interfaz para las consultas dependiendo del tema o licenciatura.



Capítulo 4. CONCLUSIONES Y TRABAJO A FUTURO

4.1 Conclusiones

El objetivo de esta investigación es abordar el problema de la administración de los casos prácticos para perfiles profesionales, el cual ahora ya se cuenta una herramienta vía web la cual está diseñada para el almacenamiento de esta información y a su vez cualquier persona que necesite podrá consultar los casos Prácticos de una forma fácil y segura.

SICP (Sistema de Información para Administrar Casos Prácticos para Perfiles Profesionales), cuenta con un registro de usuario para poder acceder a la página y poder subir los casos prácticos que deseen compartir para brindar apoyo de asesorarse sobre diferentes temas o áreas, teniendo la responsabilidad que la información sea verídica.

Utilizando herramientas como las mencionadas anteriormente se puede construir aplicaciones para diferentes tipos de necesidades, haciéndolo de una manera eficazmente funcional, con formas físicas muy simple para cualquier usuario, con toques estéticos y novedosos. Las herramientas utilizadas crean cada paso para llegar a un proyecto por ejemplo SICP, el cual es una página web pero primero fueron prototipos en el cual se van saliendo las necesidades de los usuarios posteriormente fue la base de datos la cual se va creando de una manera funcionalmente bajo los términos de la Normalización y Estructuras, el diseño de la Interfaz de manera web es un concepto más estético para que al usuario no se le haga tedioso, los accesos deben de ser fáciles y visuales para que se detecten de una manera rápida.

4.2 Trabajo a futuro

SICP continuara una innovación de sus conceptos desde una mejor visibilidad hasta herramientas de mejor uso para cualquier usuario.

Utilizando los diferentes navegadores, esta página se pueda traducir en diferentes idiomas, para que así más usuarios puedan hacer uso de ella sin tener que estar traduciendo por partes.

Que puedan tener la opción de descargar en PDF para un mejor manejo de la información y sin tener que estar imprimiendo tantas hojas y cuidamos el medio ambiente.

Tener la opción de poder opinar sobre los casos prácticos para mantener actualizada la información y compartir opiniones entre usuarios.

Crear una aplicación APP para el mejor uso, así podrán tener siempre la información y más cuando la necesiten, una plataforma moderna y con el manejo simple para utilizarla, que no cueste y pueda usarse en cualquier teléfono móvil.



Que este Sistema se pueda utilizar en diferentes partes del mundo, que las opiniones e ideas sean diferentes para tener un mejor estatus de todos los casos prácticos.

BIBLIOGRAFIA

[1] Kendall & Kendall

Análisis y diseño de sistemas, Pearson Educación
México, 2005.

[2] Simón Bennett, Steve McRobb, Ray Farmer

Análisis y diseño orientado a objetos de sistemas usando UML
Madrid: McGraw-Hill, cop. 2007.
Universidad Yacambú

[3] Lennis Zamudio , Luís Ramírez , Manuel Álvarez, Anaíz Rodríguez

Curso: Teoría y Administración de Base de Datos
http://www.oocities.org/es/avrrinf/tabd/Foro/Foro_UML.htm

[4] García Rincón, Luis Francisco

Bases de datos: un enfoque práctico
México, D.F.: Trillas, 2007.

[5] M. Kroenke; traducción Ana Elizabeth García Hernández

Procesamiento de base de datos: fundamentos, diseño, e instrumentación
México: Person Educación, c200³

[6] Pavón Puertas Jacobo. 5

Creación de un portal con PHP y MySQL, 4a ed.
México: Alfaomega, Ra-Ma, 2011

[7] Spona, Helma.6

Programación de bases de datos con MySQL y PHP
México: Alfaomega Grupo Edito, 2010.

[8] Karlins, David; traducción: Eloy Pineda Rojas. 7

Adobe dreamweaver CS4: técnicas esenciales
México, D.F.: McGraw-Hill, 2010

[9] DuBois, Paul, Gómez Pastor José Manuel.8

MySQL



Madrid: Anaya Multimedia, c2005.

TESIS PROFESIONAL 9

[10] Carmona Luna Aida

Simulación de Pruebas para Preparación del Examen del CENEVAL,

México: Ciencia de la Computación, Buap.

Manual de Elaboración de Casos Prácticos 10

[11] ACREL,

Proceso de Evaluación para la Acreditación de Conocimientos a Nivel Licenciatura por Acuerdo

Secretarial 286,

CENEVAL

[12] Ian Gilfillan 11

"La biblia MySQL"

[13] Anaya multimedia

EASYPLANEX 12

Software de Evaluación y Optimización de Proyectos de Capital

<http://easyplanex.com/>

[14] TEACH2000 13

Programa de ayuda para evaluar a alumnos de Lenguas Extranjeras

<http://www.formacionyrecursos.com/teach2000-mete-al-profesor-en-el-pc.html>

[15] Dreamweaver CS6

Programa creador de interfaces y páginas web

<http://www.adobe.com/mx/products/dreamweaver.html>

[16] MANUAL DE PHP

Una explicación sencilla

<http://php.net/manual/en/tutorial.php>

[17] Balsamiq

Balsamiq Mockups

<https://balsamiq.com/>