

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación



“Sistema Web que Administra los
Productos Educativos de los
Profesores del Área de Programación”

Elaborado por: Uriel Caamaño Mejia

Tutor: Erica Vera Cervantes

Introducción.....	6
Capítulo. I Marco Teórico	1
1.1 Ingeniería de software.....	1
1.1.1 Definición.....	1
1.1.2 Proceso de Software.....	1
1.1.3 Modelo de Proceso de Software.	2
1.1.4 Métodos de Ingeniería de Software.	4
1.1.5 Atributos de un buen software.....	6
1.2 Internet.	7
1.2.1 El modelo OSI.	7
1.2.2 Arquitectura cliente-servidor.....	8
1.2.3 Protocolo TCP/IP.....	8
1.2.4 HTML.	9
1.2.5 ASP.	10
1.3 Bases de Datos.....	11
1.3.1 Modelo Entidad-Relación.	11
1.3.2 Diferentes Esquemas de Bases de Datos.....	12
1.3.2.1 Modelo Relacional.....	12
1.3.2.2 Modelo de Red.	13
1.3.2.3 Modelo Jerárquico.	13
1.3.3 Normalización de Bases de Datos.	14
1.4 Análisis y Diseño Orientado a Objetos.....	15
1.5 UML (Lenguaje Unificado de Modelado).	17
1.5.1 Definición.....	17
1.5.2 Diagramas de Casos de Uso	18
1.5.3 Diagramas de Clases	20
1.5.4 Diagramas de Secuencia.	21
1.5.5 Diagramas de Colaboración	22
Capítulo II Análisis.....	23
2.1 Análisis Conceptual.	23
2.1.1 Identificación de Entidades y Atributos.....	23
2.1.2 Modelo Entidad – Relación del sistema.....	26
2.2 Análisis Funcional.	27
2.2.1 Actores.	28
2.2.1.1 Descripción de actores.	29
2.2.2 Casos de uso.	30
2.2.3 Diagramas de Casos de Uso.	32
2.2.4 Especificación de Casos de Uso.	32
2.3 Identificación de clases.....	74
Capítulo III Diseño.....	76
3.1 Diseño Conceptual.....	76
3.1.1 Normalización.	76
3.1.2 Modelo Relacional.....	78
3.1.3 Instancias de la Base de Datos.....	79
3.1.4 Especificación de Atributos.	81
3.2 Diseño Funcional.	85
3.2.1 Diagramas de Secuencia.	85

3.2.2 Diagramas de Estado.....	88
Capítulo IV Implementación.	91
4.1 Interfaces.	91
Conclusiones y Recomendaciones.	103
Bibliografía.	103

Índice de Figuras

1.1.3 Modelo en Cascada	3
1.1.4 Métodos de Ingeniería	6
1.2.1 Capas del Modelo OSI	7
1.2.2 Arquitectura cliente-servidor	8
1.3.1 Modelo Entidad- Relación	12
1.3.1.1 Relación entre entidades	12
1.5.2 Diagrama de Caso de Uso	20
1.5.3 Diagrama de Clases Restaurante	22
1.5.4 Diagrama de Secuencia de Máquina de Café	23
1.5.5 Diagrama de Colaboración de un Servidor	23
Identificación de Atributos datos personales 1.1A	24
Identificación de Atributos datos laborales 2.1.1B	25
Identificación de Atributos materiales 2.1.1C	26
Identificación de Atributos horarios 2.1.1D	26
Identificación de Atributos materias 2.1.1E	27
Identificación de Atributos país 2.1.1D	27
Identificación de Atributos áreas 2.1.1E	27
Identificación de Atributos categorías 2.1.1F	27
Identificación de Atributos cuerpo académico 2.1.1G	27
2.1.2 Diagrama Modelo Entidad-Relación	28
2.2.1 Diagrama Funcional	28
2.2.1.1 Representación de actores en los diagramas	31
2.2.2 Tabla Caso de Uso	31
2.2.3 Diagrama de Caso de Uso	33
Especificación Diagrama de Caso de Uso Login (A)	34
Especificación Diagrama de Caso de Uso Profesor (B)	39
Especificación Diagrama de Caso de Uso Materia (C)	43
Especificación Diagrama de Caso de Uso Áreas (D)	47
Especificación Diagrama de Caso de Uso Categorías (E)	51
Especificación Diagrama de Caso de Uso Cuerpo Académico (F)	55
Especificación Diagrama de Caso de Uso País (G)	59
Especificación Diagrama de Caso de Uso Datos Personales (H)	64
Especificación Diagrama de Caso de Uso Datos Laborales (I)	68
Especificación Diagrama de Caso de Uso Horario (J)	72
Especificación Diagrama de Caso de Uso Material Didáctico (K)	75
2.2.5 Identificación de Clases	76
3.1 Base de Datos Normalizada	79
Modelo Relacional 3.1.2	80
Instancia área (A)	81
Instancia categoría (B)	81
Instancia cuerpo académico (C)	81
Instancia horario (D)	82
Instancia material didáctico (E)	82
Instancia materia (F)	82

Instancia país (G)	82
Especificación de Atributos áreas (A)	83
Especificación de Atributos categorías (B)	83
Especificación de Atributos cuerpo académico (C)	83
Especificación de Atributos datos personales (D)	83
Especificación de Atributos datos laborales (E)	84
Especificación de Atributos horarios (F)	85
Especificación de Atributos materiales (G)	85
Especificación de Atributos materias (H)	86
Especificación de Atributos país (I)	86
3.2.1 Ingreso de Login	87
3.2.2 Catálogos de Administrador	88
3.2.3 Ingreso al menú de Profesor	89
3.2.2 Ingreso del Login	89
3.2.3 Ingreso al Menú Principal de Profesores	90
3.2.4 Ingreso al Menú Principal de Administrador	91
4.1 Acceso de Administrador	92
4.2 Menú Principal del Administrador	93
4.3 Catalogo de País	93
4.4 Catalogo Cuerpo Académico	94
4.5 Catalogo de Área.	94
4.6 Catalogo de Categorías.	95
4.7 Acceso de Profesor.	96
4.8 Menú Principal Profesor	96
4.9 Ingreso de Datos Personales.	97
4.10 Alta de Fotografía.	98
4.11 Alta de Datos Laborales.	98
4.12 Catalogo de Material Didáctico.	99
4.13 Alta de Material Didáctico.	100
4.14 Horario del Profesor.	100
4.15 Alta de Horario.	101
4.16 Menú de Reportes.	101
4.17 Reporte de Descargas del Material.	102

Introducción.

En la actualidad no existe un sitio general donde el alumno pudiera consultar los materiales que cada profesor ha realizado y que se considera para poner en disposición para el alumno. Algunos profesores han construido una página donde el alumno pueda disponer de estos materiales, pero otros profesores envían los materiales a la copiadora y solo a los alumnos de la materia que están impartiendo.

De esta forma surge la necesidad de contar con un Sitio WEB que disponga de los materiales de todas las áreas, de todas las materias de los profesores. Que cuente con una sección de Administrador que permita administrar los materiales e información acerca de los temas de tesis que se proponen, proyectos en los que se participa, etc.

En el Capítulo I, se describe el marco Teórico, en donde se aborda los conceptos básicos necesarios para la realización de este trabajo, tales como Ingeniería de Software, Internet, Bases de Datos y Lenguaje de Modelado Unificado (UML).

En el Capítulo II, se aborda el Análisis funcional del sistema. Este capítulo, incluye Diagramas de Casos de Uso, Diagramas de Clases, además de la documentación pertinente a cada diagrama.

En el Capítulo III tenemos el Diseño del sistema. Partiendo del capítulo II, en el que se realiza el análisis del sistema, se realiza el diseño, esto se podrá hacer, ya que para realizar el diseño se toma en cuenta los Diagramas de Casos de Uso y los Diagramas de Clases, y partiendo de estos, se realizan los Diagramas de Secuencia, quedando listo el Diseño funcional del sistema. Al igual que en el capítulo II, además de los Diagramas, se incluye la documentación referente a cada Diagrama.

El Capítulo IV hace referencia a la implementación, se mostrarán las pantallas del sistema Web.

Capítulo. I Marco Teórico

1.1 Ingeniería de software.

El software es un ingrediente indispensable para el funcionamiento del computador. Está formado por una serie de instrucciones y datos, que permiten aprovechar todos los recursos que el computador tiene, de manera que pueda resolver gran cantidad de problemas. Un computador en si, es sólo un conglomerado de componentes electrónicos; el software le da vida al computador, haciendo que sus componentes funcionen de forma ordenada.

El software es un conjunto de instrucciones detalladas que controlan la operación de un sistema computacional. Funciones del software:

- Administrar los recursos de cómputo
- Proporcionar las herramientas para optimizar estos recursos.
- Actuar como intermediario entre el usuario y la información almacenada.

1.1.1 Definición.

La Ingeniería de software es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

Es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, que en palabras más llanas, se considera que la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos. [1]

1.1.2 Proceso de Software.

El proceso de ingeniería de software se define como un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad. El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.

Existen cuatro actividades fundamentales que son comunes para todos los procesos del software. Estas actividades son:

1. **Especificación del software:** En esta actividad se deben definir y entender tanto la funcionalidad del software como las restricciones sobre su operación.
2. **Desarrollo del Software:** Dentro de esta actividad se debe producir el software que cumpla con la especificación.
3. **Validación del software:** Dentro de esta actividad el software se debe validar para asegurarse de que eso es realmente lo que el cliente requiere.
4. **Evolución del software:** Esta actividad es la más larga, ya que dentro de ésta el software debe evolucionar para cumplir con los cambios requeridos por el cliente.

Existen distintos procesos del software y cada uno de ellos organiza estas actividades (especificación, desarrollo, validación y evolución) de diferentes formas y las describen con diferente nivel de detalle. El tiempo en el que se lleva acabo cada actividad varía, así como los resultados. Se pueden utilizar diferentes procesos para producir el mismo tipo de producto. Sin embargo, para algunos tipos de aplicación, algunos procesos son más convenientes que otros.

1.1.3 Modelo de Proceso de Software.

Ya se definió lo que es un proceso del software, ahora, un modelo de procesos del software, es una descripción de un proceso del software que se presenta desde una perspectiva particular. Los modelos son simplificaciones, por lo tanto un modelo de procesos del software es una abstracción de un proceso real. Estos modelos incluyen actividades que son parte de los procesos y productos de software y el papel de la gente involucrada en la ingeniería de software. Algunos ejemplos de estos tipos de modelos que se pueden producir son:

1. **Un modelo de flujo de trabajo.** Muestra la secuencia de actividades en el proceso en conjunción con sus entradas, salidas y dependencias. Las actividades en este modelo representan acciones humanas.
2. **Un modelo de flujo de datos o de actividad.** Representa el proceso como un conjunto de actividades, cada una de las cuales lleva a alguna transformación en los datos. Muestra cómo la entrada en el proceso, por ejemplo una especificación, se transforma en una salida como un diseño. Estas actividades son de menor nivel que las de un modelo de flujo de trabajo. Pueden representar transformaciones llevadas acabo por la gente o por las computadoras.
3. **Un modelo de rol / acción.** Representa los roles de la gente involucrada en el proceso del software y las actividades de las que son responsables.

Existe una gran variedad de modelos generales diferentes o paradigmas de desarrollo de software, tales como los que se mencionan a continuación:

1. **Modelo Cascada.** Este es el más básico de todos los modelos, y sirve como bloque de construcción para los demás modelos de ciclo de vida. La visión del modelo cascada del desarrollo de software es muy simple; dice que el desarrollo de software puede ser a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de una fase contribuyen a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la fase. Las flechas muestran el flujo de información entre las fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación (Véase en la figura 1.1.3).[1]

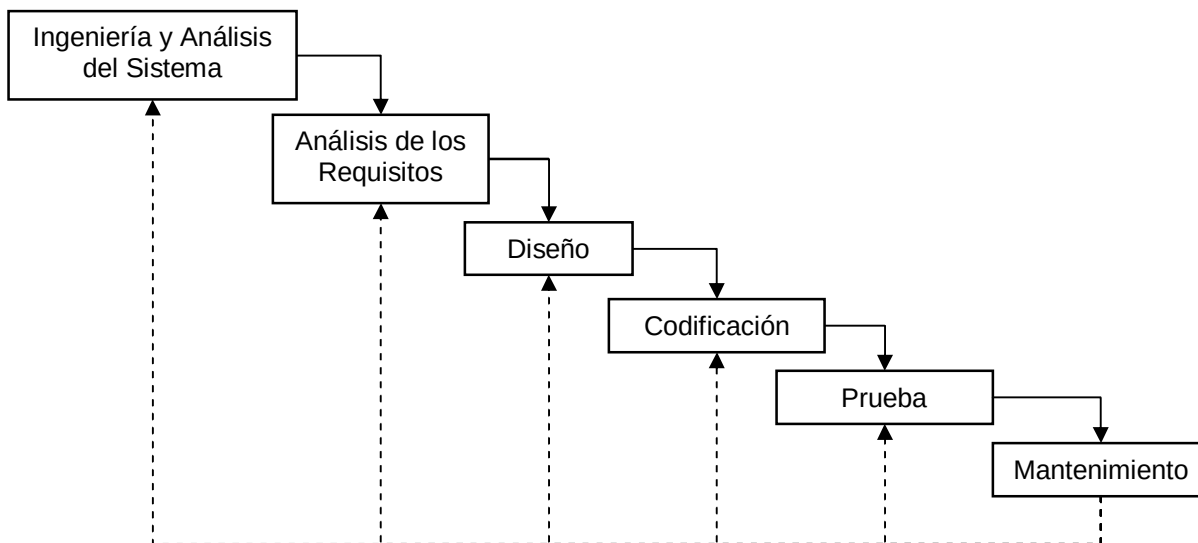


Figura Modelo en Cascada 1.1.3

2. **Modelo De Desarrollo Evolutivo.** Como el modelo de desarrollo incremental, el modelo de desarrollo evolutivo (algunas veces denominado como prototipado evolutivo) construye una serie de grandes versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al comenzar, el modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto.

En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.

El sistema es entonces desarrollado, los usuarios lo usan, y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada. El proceso se repite indefinidamente.

3. Modelo Espiral. El modelo espiral de los procesos de software es un modelo del ciclo de meta-vida. En este modelo, el esfuerzo de desarrollo es iterativo. Tan pronto como uno completa un esfuerzo de desarrollo, otro comienza. Además, en cada desarrollo ejecutado, puedes seguir estos cuatros pasos:

- Determinar qué quieres lograr.
- Determinar las rutas alternativas que puedes tomar para lograr estas metas. Por cada una, analizar los riesgos y resultados finales, y seleccionar la mejor.
- Seguir la alternativa seleccionada en el paso 2.
- Establecer qué tienes terminado.

Se observa un escenario particular. Digamos que en este proyecto, nosotros viajaremos a resolver un conjunto particular de problemas del cliente. Durante el primer viaje alrededor de la espiral, se analiza la situación y se determina que los mayores riesgos son la interfaz del usuario. Después de un cuidadoso análisis de las formas alternativas de direccionar ésto (por ejemplo, construir un sistema y esperar lo mejor, escribir una especificación de requerimientos y esperar que el cliente lo entienda, y construir un prototipo), se determina que el mejor curso de acción es construir un prototipo.

Luego se provee el prototipo al cliente quien nos provee con retroalimentación útil. Ahora, se comienza el segundo viaje alrededor de la espiral. Este tiempo decidimos que el mayor riesgo es ese miedo a que muchos nuevos requerimientos comiencen a aparecer sólo después de que el sistema sea desplegado. Después del despliegue, el cliente nos provee de retroalimentación que dirá si estamos correctos con esos requerimientos, pero 50 nuevos requerimientos ahora se originarán en las cabezas de los clientes. Y el tercer viaje alrededor de la espiral comienza.

El modelo espiral captura algunos principios básicos:

- Decidir qué problema se quiere resolver antes de viajar a resolverlo.
- Examinar tus múltiples alternativas de acción y elegir una de las más convenientes.
- Evaluar qué tienes hecho y qué tienes que haber aprendido después de hacer algo.
- No ser tan ingenuo para pensar que el sistema que estás construyendo será "EL" sistema que el cliente necesita, y
- Conocer (comprender) los niveles de riesgo, que tendrás que tolerar.
- El modelo espiral no es una alternativa del modelo cascada, ellos son completamente compatibles.

1.1.4 Métodos de Ingeniería de Software.

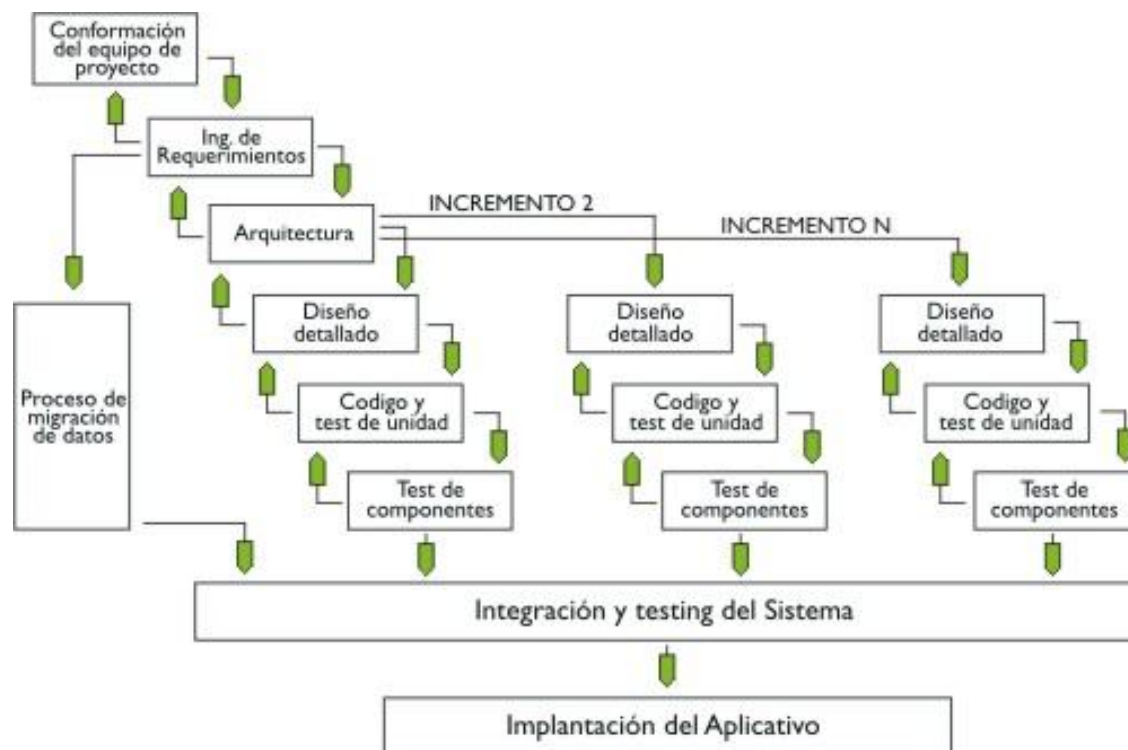
La ingeniería de Software fue definida por Bauer a finales de los 60s como el establecimiento y uso de principios de ingeniería para obtener software que fuera confiable y que funcionara eficientemente con las máquinas reales. A pesar de ser vieja, esta definición da el sentimiento correcto detrás de la disciplina.

Los desarrollos de ingeniería de Software comenzaron con la técnica de programación y después fueron utilizados en otras fases del ciclo vital del software. La programación estructurada fue seguida por otros métodos estructurados de análisis y también métodos

estructurados de diseño. Además, comenzaron tecnologías orientadas a objeto. En épocas tempranas la programación era la tarea de oro de ingeniería del Software, pero ahora la ingeniería y el diseño de requisitos son más populares.

Se han desarrollado muchos métodos para hacer modelos de sistemas de información. Sin embargo, los métodos orientados a objetos van a llegar a ser el Standard. Los métodos formales confían en las técnicas matemáticas que expresan y modelan los requisitos de cualquier producto en el ciclo vital del software. El uso de métodos formales es recomendado cuando sea posible en un ciclo vital del software. El Desarrollo de Software Orientado a Objetos comenzó en los 80s como una etapa natural de los métodos estructurados. UML (Unified Modeling Language) ha emergido como una unificación de los diversos métodos orientados a objeto y se está convirtiendo en un Standard de ISO.

Actualmente, la tecnología de componente es un método para desarrollar sistemas de información que esta creciendo. A diferencia de los métodos tradicionales, la tecnología de componente ensambla componentes para formar una solución de software. Actualmente, hay dos estándares de componentes en competencia: JavaBeans por Sun y DCOM por Microsoft. Los componentes de software son materiales reusables para construir sistemas de software. La tecnología Component-Base (Basada en Componentes) es un método poderoso por la empresa de la ingeniería de sistemas de información porque es una tecnología que reduce el conflicto entre sistemas de alta complejidad y de la búsqueda para la alta calidad y la productividad (Véase la siguiente figura 1.1.4). [1]



Métodos de Ingeniería 1.1.4

1.1.5 Atributos de un buen software.

Para mencionar atributos de un buen software, es inevitable arribar al concepto de calidad.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro o de un programa a otro.

La calidad del software puede medirse después de elaborado el producto, pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

Facilidad de mantenimiento

- Ha de poder evolucionar para adaptarse a las necesidades de cambio de los clientes.

Confiabilidad

- No debe causar daños físicos o económicos en el caso de fallo del sistema
- Fiabilidad, seguridad y protección.

Eficacia

- Hacer efectivo el propósito del software.

Usabilidad

- Fácil de utilizar
- Debe tener una interfaz de usuario apropiada y una documentación adecuada

Reusabilidad

- Capacidad de que un software pueda utilizarse en un contexto diferente al de su creación

Portabilidad

- Facilidad de transferir productos de software a diferentes plataformas [2]

1.2 Internet.

Lo que se conoce hoy como Internet es en realidad un conjunto de redes independientes (de área local y área extensa) que se encuentran conectadas entre sí, permitiendo el intercambio de datos y constituyendo por lo tanto una red mundial que resulta el medio idóneo para el intercambio de información, distribución de datos de todo tipo e interacción personal con otras personas.

Para ello es necesario el uso de un protocolo de comunicaciones común. El protocolo que proporciona la compatibilidad necesaria para la comunicación en Internet es el TCP/IP.

1.2.1 El modelo OSI.

El modelo OSI (Open System Interconnection) es utilizado por prácticamente la totalidad de las redes del mundo. Este modelo fue creado por el ISO (Organización Internacional de Normalización), y consiste en siete niveles o capas donde cada una de ellas define las funciones que deben proporcionar los protocolos con el propósito de intercambiar información entre varios sistemas. Esta clasificación permite que cada protocolo se desarrolle con una finalidad determinada, lo cual simplifica el proceso de desarrollo e implementación. Cada nivel depende de los que están por debajo de él, y a su vez proporciona alguna funcionalidad a los niveles superiores. Los siete niveles del modelo OSI se muestran en la tabla 1.2.2.

Aplicación	El nivel de aplicación es el destino final de los datos donde se proporcionan los servicios al usuario.
Presentación	Se convierten e interpretan los datos que se utilizarán en el nivel de aplicación.
Sesión	Encargado de ciertos aspectos de la comunicación como el control de los tiempos.
Transporte	Transporta la información de una manera fiable para que llegue correctamente a su destino.
Red	Nivel encargado de encaminar los datos hacia su destino eligiendo la ruta más efectiva.

Enlace	Enlace de datos. Controla el flujo de los mismos, la sincronización y los errores que puedan producirse.
Físico	Se encarga de los aspectos físicos de la conexión, tales como el medio de transmisión o el hardware.

Capas del Modelo OSI 1.2.1

1.2.2 Arquitectura cliente-servidor.

La arquitectura cliente-servidor es una forma específica de diseño de aplicaciones, también se conoce con este nombre a las computadoras en las que estas aplicaciones son ejecutadas. Por un lado, el cliente es la computadora que se encarga de efectuar una petición o solicitar un servicio. El cliente no posee control sobre los recursos, sino que es el servidor el encargado de manejarlos. Por otro lado se encuentra una computadora remota que actúa como servidor, esta se encarga de evaluar la petición del cliente y decidir si la acepta o la rechaza consecuentemente. Una vez que el servidor acepta la petición que realice el cliente, la información requerida es enviada al cliente que efectuó la petición, siendo este último (el cliente) el responsable de mostrar los datos al usuario con el formato adecuado. Finalmente se debe precisar que cliente y servidor no tienen que estar necesariamente en computadoras separadas, sino que pueden ser programas diferentes que se ejecuten en la misma computadora. [16]

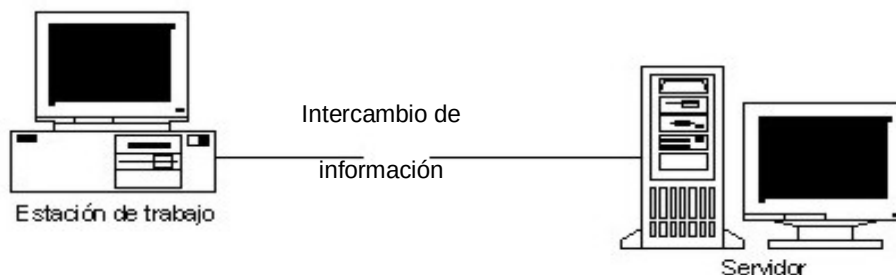


Figura 1.2.2 Arquitectura cliente-servidor

1.2.3 Protocolo TCP/IP.

TCP/IP es el protocolo común utilizado por todas las computadoras conectadas a Internet, de manera que éstos puedan comunicarse entre sí. Hay que tener en cuenta que en Internet se encuentran conectadas computadoras de clases muy diferentes y en muchos casos con hardware y software incompatibles, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP

(Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto

TCP (Transmission Control Protocol).

El protocolo de control de transmisión (TCP) pertenece al nivel de transporte, siendo el encargado de dividir el mensaje original en datagramas de menor tamaño, y por lo tanto, mucho más manejables. Los datagramas serán dirigidos a través del protocolo IP de forma individual. El protocolo TCP se encarga además de añadir cierta información necesaria a cada uno de los datagramas. Esta información se añade al inicio de los datos que componen el datagrama en forma de cabecera.

IP (Internet Protocol)

El IP es un protocolo que pertenece al nivel de red, por lo tanto, es utilizado por los protocolos del nivel de transporte como TCP para encaminar los datos hacia su destino. IP tiene únicamente la misión de encaminar el datagrama, sin comprobar la integridad de la información que contiene. Para ello se utiliza una nueva cabecera que se antepone al datagrama que se está tratando. [16]

1.2.4 HTML.

HTML (HyperText Markup Language) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...). Este lenguaje es el que se utiliza para presentar información en el World Wide Web.

La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Explorer o Netscape).

Para la creación de documentos HTML sólo es necesario:

a) Un procesador de texto: los documentos HTML están en formato de texto sencillo (también conocido como ASCII). El procesador de texto se utiliza para escribir el documento en lenguaje HTML, que será posteriormente interpretado por el programa navegador correspondiente, siempre que el documento esté guardado en formato: "sólo texto" con extensión .html.

El texto escrito tiene dos partes bien diferenciadas, el contenido de la información y el conjunto de etiquetas del lenguaje HTML, que permiten determinar el estilo y el tipo de letra que tendrá la presentación del documento final y que pueda ser leído por un programa cliente. Es importante mencionar, que el lenguaje HTML, no solo es capaz de manejar y

mostrar texto, sino que también se pueden manejar y mostrar tanto texto como imágenes, tablas, listas colgantes, etc.

Para escribir un párrafo sin estilo específico (por defecto) no es necesario poner etiqueta alguna. Lo único que hay que tener en cuenta es que al presentar el documento se hace caso omiso de los espacios, tabulaciones y retornos de carro que se encuentren en el texto fuente. Por esta razón se utilizan una serie de etiquetas que sustituyen a estos elementos.

b) Un navegador del WWW o lo que se denomina programa cliente que permite el acceso a páginas WWW de Internet. El programa cliente permite ver una página antes de introducirla en un servidor. De este modo, se pueden comprobar las modificaciones de las distintas páginas que se diseñan.

Con el procesador de textos se crea un fichero con un nombre al que necesariamente hay que añadirle la extensión html. Por ejemplo: prueba.html. Si se añade algo nuevo se deben guardar los cambios antes de visualizar el nuevo documento en el navegador. El documento se abre con el comando Open File del menú File del programa cliente. Los cambios son reflejados en la pantalla. Si se trabaja con el programa cliente y el procesador de textos al mismo tiempo, activando simplemente la opción Reload se pueden comprobar los cambios efectuados.

Las etiquetas del lenguaje HTML.- El lenguaje HTML usa etiquetas o directivas (tags) para indicarle al programa cliente de Web como mostrar el texto.

Las etiquetas están formadas por determinados caracteres metidos entre los signos <>, y con la barra </> cuando se trata de la segunda etiqueta de un efecto (la de cierre). Por ejemplo: <title> para abrir y </title> para cerrar.

1.2.5 ASP.

Las siglas ASP corresponden a las palabras Active Server Pages (Páginas Activas en el Servidor). Es una tecnología que impulsó Microsoft hace ya varios años, y que en la actualidad es uno de los lenguajes de programación Web más utilizados. Su funcionamiento se basa, principalmente, sobre servidores Microsoft con Internet Information Server para Windows NT o 2000, y en caso de contar con un sistema operativo Windows 95 o 98 se utiliza un servidor Web personal, especialmente el Personal Web Server.

Hay que destacar que las páginas ASP se ejecutan del lado del servidor, devolviendo al cliente los resultados, esto quiere decir que no importa el navegador o sistema operativo del usuario, ya que el mismo verá una simple página HTML. Para correr aplicaciones ASP bajo sistemas con Windows 95 o 98 es necesario contar con el Personal Web Server, el cual está incluido en el CD de Instalación de Windows 98. Si el sistema es Windows NT o 2000 hay que contar con el IIS (Internet Information Server).

Una de las características más importantes de las páginas ASP es la posibilidad de conectar con diferentes tipos de bases de datos, para extraer-agregar-eliminar datos de ellas, y generar páginas con esos datos. Estas páginas se generan en forma dinámica, dependen de las sentencias que se establezcan, para así obtener los resultados del proceso realizado. Pueden conectarse a motores de bases de datos SQL, Access, Oracle y a cualquier otro con soporte de conexión ODBC.

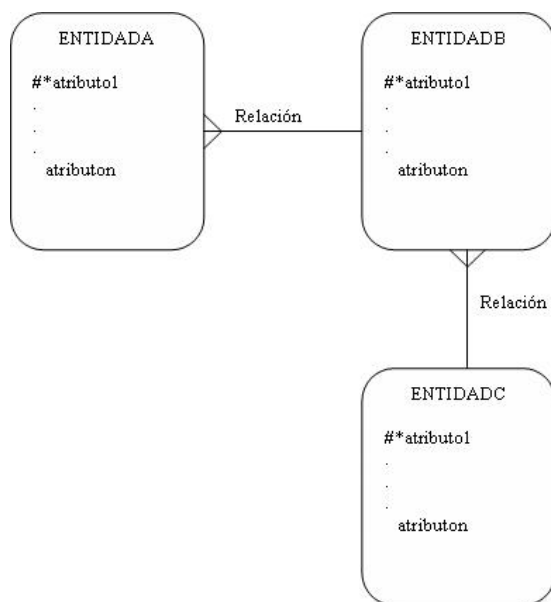
No hay un software específico para realizar páginas ASP, se puede utilizar desde el más sencillo editor de textos hasta software como el DreamWeaver de Macromedia. En caso de realizar la edición con un editor simple, es deseable que posea la opción de resaltado de sintaxis, la cual hará más simple la tarea de programación.

1.3 Bases de Datos.

Almacén de datos relacionados con diferentes modos de organización. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al diseñador. Se diseña y almacena datos con un propósito específico.

1.3.1 Modelo Entidad-Relación.

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases 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 (véase la figura 1.3.1). [3]



Modelo Entidad- Relación 1.3.1

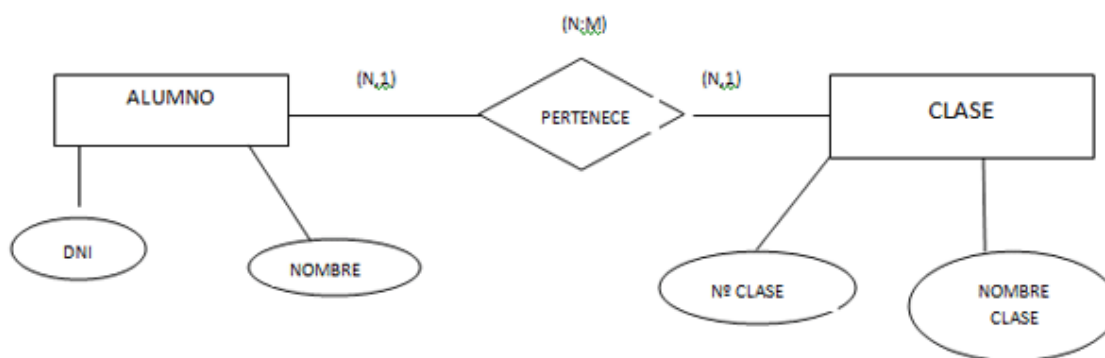
Entidad.

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, diseños de productos, conciertos, excursiones, etc. Cada ejemplar de una entidad se denomina instancia. Por ejemplo, Ignacio y Camila pueden ser dos instancias distintas de la entidad persona.[3]

Relación.

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en 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; etc.(véase la figura 1.3.1.1) [3]



Relación entre entidades 1.3.1.1

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. 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 pueden ser simples o compuestos. Un atributo simple es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. [5]

1.3.2 Diferentes Esquemas de Bases de Datos.

1.3.2.1 Modelo Relacional.

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está

compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, (Structured Query Language) Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

1.3.2.2 Modelo de Red.

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

1.3.2.3 Modelo Jerárquico.

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

1.3.3 Normalización de Bases de Datos.

El proceso de normalización de una base de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo E-R (entidad-relación) al modelo relacional. Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla bidimensional sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada columna debe tener su nombre único.
- No puede haber dos filas iguales. No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo.

Formas Normales

Las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos. El creador de estas 3 primeras formas normales fue Edgar F. Codd, éste introdujo la normalización en un artículo llamado A Relational Model of Data for Large Shared Data Banks Communications of the ACM, Vol. 13, No. 6, June 1970.

Primera Forma Normal (1NF). Una relación está en Primera Forma Normal si y sólo si todos los dominios son atómicos. Un dominio es atómico si los elementos del dominio son indivisibles.

Segunda Forma Normal (2NF). Dependencia completa. Esta en 2NF si esta en 1NF y si sus atributos no principales dependen de forma completa de la clave principal. Toda columna que no sea clave debe depender por completo de la clave primaria. Los atributos dependen de la clave. Varía la clave y varían los atributos. Dependencia completa. Sus atributos no principales dependen de forma completa de la clave principal.

Tercera Forma Normal (3NF). La definición de Codd indica que una tabla está en 3NF si y solo si las dos condiciones siguientes se mantienen:

- La tabla está en la segunda forma normal (2NF)
- Ningún atributo no-primario de la tabla es dependiente transitivamente de una clave candidata

Un atributo no-primario es un atributo que no pertenece a ninguna clave candidato. Una dependencia transitiva es una dependencia funcional $X \rightarrow Z$ en la cual Z no es inmediatamente dependiente de X , pero sí de un tercer conjunto de atributos Y , que a su vez depende de X . Es decir, $X \rightarrow Z$ por virtud de $X \rightarrow Y$ y $Y \rightarrow Z$.

1.4 Análisis y Diseño Orientado a Objetos.

La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos). A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

Dada esta propiedad de conjunto de una clase de objetos, que al contar con una serie de atributos definitorios, requiere de unos métodos para poder tratarlos (lo que hace que ambos conceptos están íntimamente entrelazados), el programador debe pensar indistintamente en ambos términos, ya que no debe nunca separar o dar mayor importancia a los atributos en favor de los métodos, ni viceversa.

Hacerlo puede llevar al programador a seguir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen esa información por otro (llegando a una programación estructurada camuflada en un lenguaje de programación orientado a objetos).

Esto difiere de la programación estructurada tradicional, en la que los datos y los procedimientos están separados y sin relación, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida. La programación estructurada anima al programador a pensar sobre todo en términos de procedimientos o funciones, y en segundo lugar en las estructuras de datos que esos procedimientos manejan.

En la programación estructurada se escriben funciones y después les pasan datos. Los programadores que emplean lenguajes orientados a objetos definen objetos con datos y métodos y después envían mensajes a los objetos diciendo que realicen esos métodos en sí mismos.

La Programación Orientada a Objetos (POO) como solución

La programación orientada a objetos es una nueva forma de programar que trata de encontrar solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad ("métodos"). Corresponden a los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa).

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instancia es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Método: algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Representación de un objeto: un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

Hay un cierto desacuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como "orientado a objetos", pero hay un consenso general en que las características siguientes son las más importantes (para más información, seguir los enlaces respectivos):

Abstracción: Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.

Encapsulamiento. Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes de los sistemas. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo.

Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el

encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que reimplementar su comportamiento.[7]

1.5 UML (Lenguaje Unificado de Modelado).

1.5.1 Definición

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group).

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.

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir -es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Diagramas de estructura enfatizan en los elementos que deben existir en el sistema modelado:

Diagrama de clases .Son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, donde se crea el diseño conceptual de la información que se maneja en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

- **Propiedades** también llamados atributos o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso.
- **Operaciones** son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una

palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.

- **Interfaz** es un conjunto de operaciones y/o propiedades que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto.
- **Herencia** se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre. Por ejemplo: Una persona puede subdividirse en Proveedores, Acreedores, Clientes, Accionistas, Empleados; todos comparten datos básicos como una persona, pero además tendrá información adicional que depende del tipo de persona, como saldo del cliente, total de inversión del accionista, salario del empleado, etc.

Diagrama de objetos. Son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML.

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase.

Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma, Nombre de objeto: Nombre de clase. Por ejemplo, Miguel: Persona.

Diagramas de comportamiento enfatizan en lo que debe suceder en el sistema modelado:

Diagrama de casos de uso. Es una especie de diagrama de comportamiento.

El Lenguaje de Modelado Unificado define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso.

Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

1.5.2 Diagramas de Casos de Uso

El estándar de Lenguaje de Modelado Unificado de OMG define una notación gráfica para realizar diagramas de casos de uso, pero no el formato para describir casos de

uso. Mucha gente sufre la equivocación pensando que un caso de uso es una notación gráfica (o es su descripción). Mientras la notación gráfica y las descripciones son importantes, ellas son documentación de un caso de uso un propósito para el que el actor puede usar el sistema.

El valor verdadero de un caso de uso reposa en dos áreas:

- La descripción escrita del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Esta descripción se enfoca en el valor suministrado por el sistema a entidades externas tales como usuarios humanos u otros sistemas.
- La posición o contexto del caso de uso entre otros casos de uso. Dado que es un mecanismo de organización, un conjunto de casos de uso coherente, consistente promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo.

Es práctica común crear especificaciones suplementarias para capturar detalles de requisitos que caen fuera del ámbito de las descripciones de los casos de uso. Ejemplos de esos temas incluyen rendimiento, temas de escalabilidad/gestión, o cumplimiento de estándares (Ejemplo en la figura 1.5.2).[8]

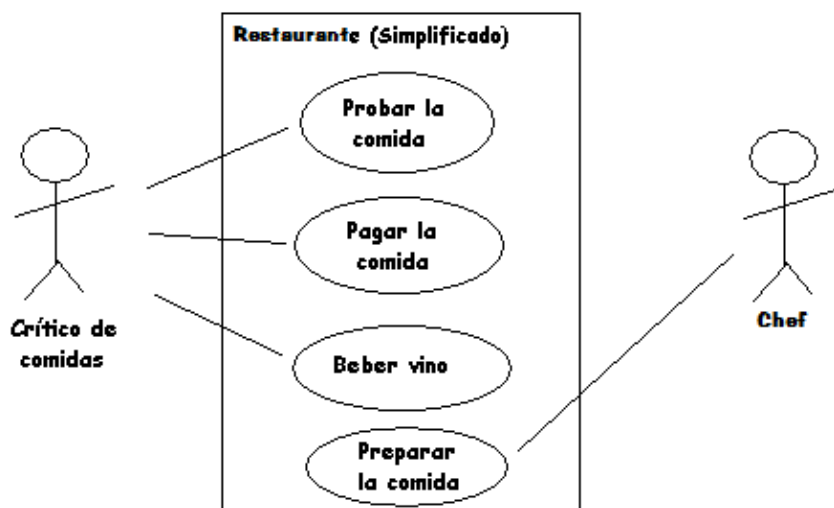


Diagrama de Caso de Uso 1.5.2

Relaciones de Casos de Uso

Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML, el cual describe notación gráfica para esas relaciones.

Incluye. En una forma de interacción, un caso de uso dado puede "incluir" otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido. Esto es útil

para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual.

La notación es una flecha rayada desde el caso de uso que lo incluye hasta el caso de uso incluido, con la etiqueta "«include»". Este uso se asemeja a una expansión de una macro donde el comportamiento del caso incluido es colocado dentro del comportamiento del caso de uso base. No hay parámetros o valores de retorno.

Extend. En otra forma de interacción, un caso de uso dado, (la extensión) puede extender a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación es una flecha rayada desde el caso de uso extensión al caso de uso extendido, con la etiqueta «extend». Esto puede ser útil para lidiar con casos especiales, o para acomodar nuevos requisitos durante el mantenimiento del sistema y su extensión.

Generalization. En la tercera forma de relación entre casos de uso, existe una relación generalización/especialización. Un caso de uso dado puede estar en una forma especializada de un caso de uso existente. La notación es una línea sólida terminada en un triángulo dibujado desde el caso de uso especializado al caso de uso general. Esto se asemeja al concepto orientado a objetos de sub.-clases, en la práctica puede ser útil factorizar comportamientos comunes, restricciones al caso de uso general, describelos una vez, y enfóntate a los detalles excepcionales en los casos de uso especializados.[8]

Diagramas de Interacción, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

1.5.3 Diagramas 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 (véase la figura 1.5.3).[8]

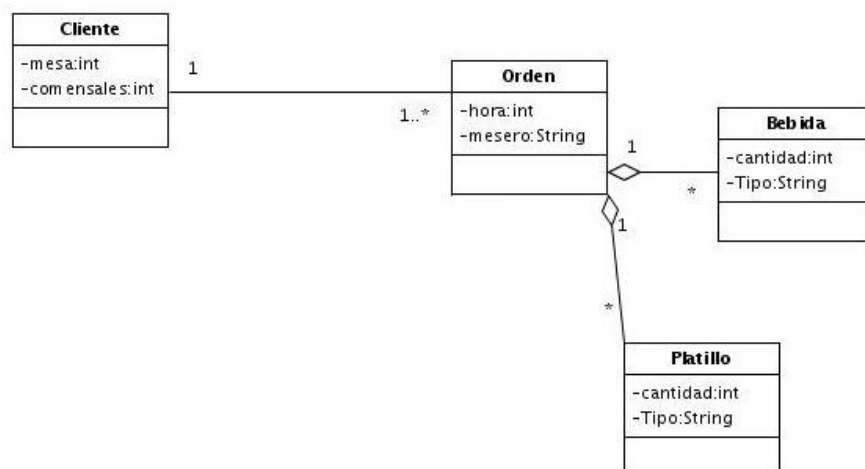


Diagrama de Clases Restaurante 1.5.3

1.5.4 Diagramas de Secuencia.

Un **diagrama de secuencia** muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

Durante el análisis inicial, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la case instanciada por el objeto en la recepción final del mensaje (véase el diagrama 1.5.4).[8]

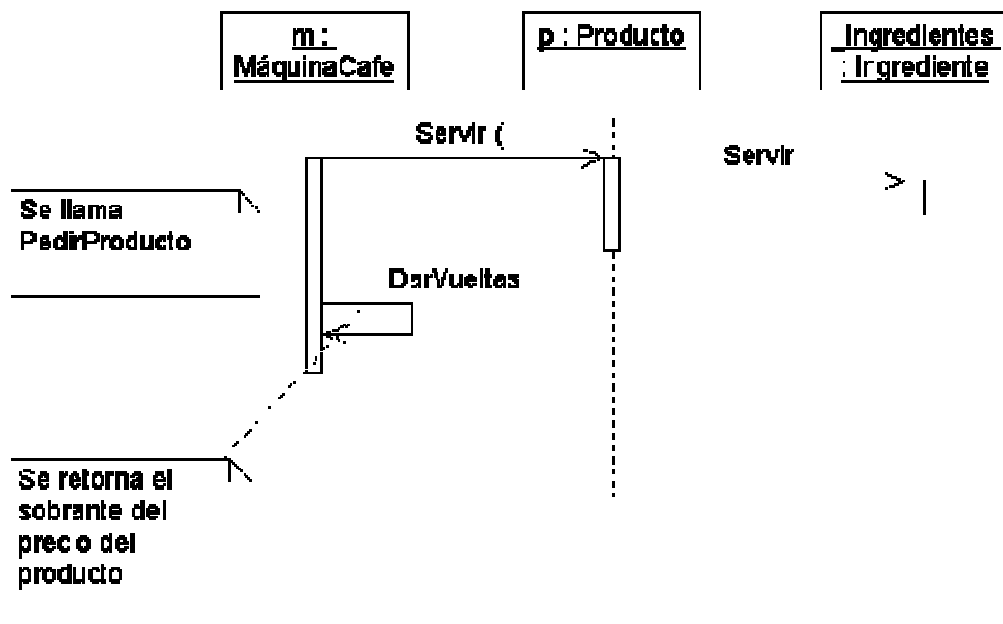


Diagrama de Secuencia de Máquina de Café 1.5.4

1.5.5 Diagramas de Colaboración

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizados entorno a los objetos y los enlaces entre ellos. Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal qué pasa primero, qué pasa después. Los clientes entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis. Por contra los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros (véase la figura 1.5.5). [8]

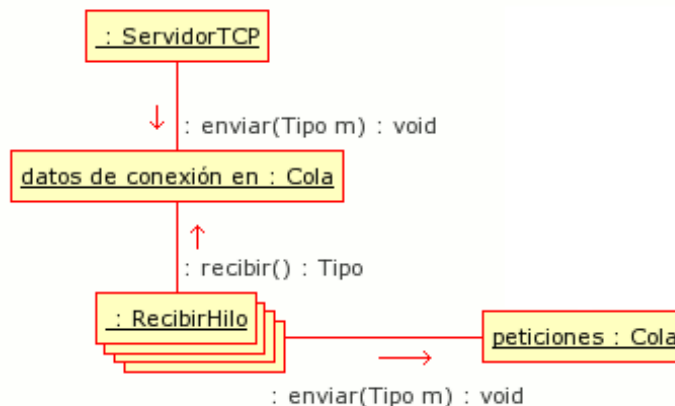


Diagrama de Colaboración de un Servidor 1.5.5

Capítulo II Análisis.

Extraer los requisitos de un producto de software es la primera etapa para crearlo, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos se plasma en el documento. Especificación de Requerimientos del Sistema, cuya estructura puede venir definida por varios estándares. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aun no está formalizada, ya se habla de la Ingeniería de Requisitos.

2.1 Análisis Conceptual.

En este modelo se habla de una forma general la representación de los datos y la descripción de cada uno de ellos.

2.1.1 Identificación de Entidades y Atributos.

A continuación se explica brevemente los atributos de cada una de las Entidades datos_ personales, datos_laborales, materiales, horarios, materias, país, áreas, cuerpo_academico y categorías.

Tabla: datos personales

Atributos	Descripción
clave_profesor	Identificador Único
clave_pais	Identificador de Referencia del catalogo de país
grado	Grado
apellido_paterno	Apellido Paterno
apellido_materno	Apellido Materno
Nombre	Nombre
curp	Curp
sexo	Sexo
estado_civil	Estado Civil
telefono_trabajo	Teléfono de Trabajo
telefono_celular	Teléfono de Celular
telefono_casa	Teléfono de Casa
direccion	Dirección
colonia	Colonia
ciudad	Ciudad

codigo_postal	Código Postal
correo_electronico	Correo Electrónico
perfil_academico	Perfil Académico
tutor	Tutor
usuario	Usuario
password	Password
estado	Estado
texto_identificacion	Texto de Identificación
dia_nacimiento	Día de Nacimiento
mes_nacimiento	Mes de Nacimiento
ano_nacimiento	Año de Nacimiento
Edad	Edad
FileName	Nombre del archivo
FileSize	Tamaño del archivo
FileData	Contenedor de Imágenes o Documentos
ContentType	Tipo de documento

Identificación de Atributos datos personales 2.1.1

Tabla: datos laborales

Atributos	Descripción
clave_profesor	Identificador Único
no_area	Identificador de Referencia del catalogo de Área
no_categoria	Identificador de Referencia del catalogo de Categoría
no_cuerpo	Identificador de Referencia del catalogo de Cuerpo
unidad_academica	Unidad Académica
tipo_contrato	Tipo de Contrato
tiempo_contrato	Tiempo de Contrato
perfil_academico	Perfil Académico
dia_ingreso	Día de Ingreso
mes_ingreso	Mes de Ingreso
ano_ingreso	Año de Ingreso
dia_inicio_contrato	Día de Inicio de Contrato
mes_inicio_contrato	Mes de Inicio de Contrato
ano_inicio_contrato	Año de Inicio de Contrato
dia_fin_contrato	Día de Fin de Contrato
mes_fin_contrato	Mes de Fin de Contrato
ano_fin_contrato	Año de Fin de Contrato
perfil_promep	Perfil De Promep
ano_perfil_promep	Año de Perfil de Promep
padron	Padrón
ano_padron	Año de Padrón

sni	Sin
ano_sni	Año de Sni
num_cubiculo	Numero de Cubículo
equipo_computo	Equipo de Computo
linea1	Linea1
linea2	Linea2
anos_experiencia	Años de Experiencia

Identificación de Atributos datos laborales 2.1.2

Tabla: materiales

Atributos	Descripción
idmaterial	Identificador Único
clave_profesor	Identificador de Referencia del catalogo de datos personales
idmateria	Identificador de Referencia del catalogo de materia
tipo_material	Tipo de Material
titulo_material	Titulo de Material
FileName	Nombre del archivo
FileSize	Tamaño del archivo
FileData	Contenedor de Imágenes o Documentos
ContentType	Tipo de documento

Identificación de Atributos materiales 2.1.3

Tabla: horarios

Atributos	Descripción
clave_horario	Identificador Único
clave_profesor	Identificador de Referencia del catalogo de datos personales
actividad	Actividad
tipo_actividad	Tipo de Actividad
dia	Día
hora_inicio	Hora de Inicio
minutos_inicio	Minutos de Inicio
hora_final	Hora de Final
minutos_final	Minutos de Final
lugar	Lugar

Identificación de Atributos horarios 2.1.4

Tabla: materias

Atributos	Descripción
idmateria	Identificador Único
clave_materia	Clave de Materia

nombre_materia	Nombre de la Materia
----------------	----------------------

Identificación de Atributos materias 2.1.5

Tabla: país

Atributos	Descripción
clave	Clave
pais	País
clave_pais	Clave De País

Identificación de Atributos país 2.1.6

Tabla: áreas

Atributos	Descripción
no_area	Identificador Único
area	Área
descripcion	Descripción

Identificación de Atributos áreas 2.1.7

Tabla: categorías

Atributos	Descripción
no_categoria	Identificador Único
categoria	Categoría
descripcion_categoria	Descripción de la Categoría

Identificación de Atributos categorías 2.1.8

Tabla: cuerpo académico

Atributos	Descripción
no_cuerpo	Identificador Único
cuerpo_academico	Cuerpo Académico
descripcion_cuerpo	Descripción de Cuerpo
siglas	Siglas
estado	Estado

Identificación de Atributos cuerpo académico 2.1.9

2.1.2 Modelo Entidad – Relación del sistema.

En la siguiente figura se muestra el diagrama entidad-relación, se explica brevemente las entidades y las relaciones encontradas para la realización de esta base de datos.

La entidad principales datos personales, ya que desde esta se desprenden las entidades secundarias, como lo es datos laborales, horarios, materiales y la que complementa a la tabla datos personales es país. La entidad datos laborales a su vez contiene a las entidades áreas, categorías y cuerpo académico. Por último, la entidad materia se relaciona con la entidad material. Este diagrama describe la manera en que se relacionan las tablas con cada

uno de sus atributos de la base de datos de Productos Educativos (véase en la siguiente figura).

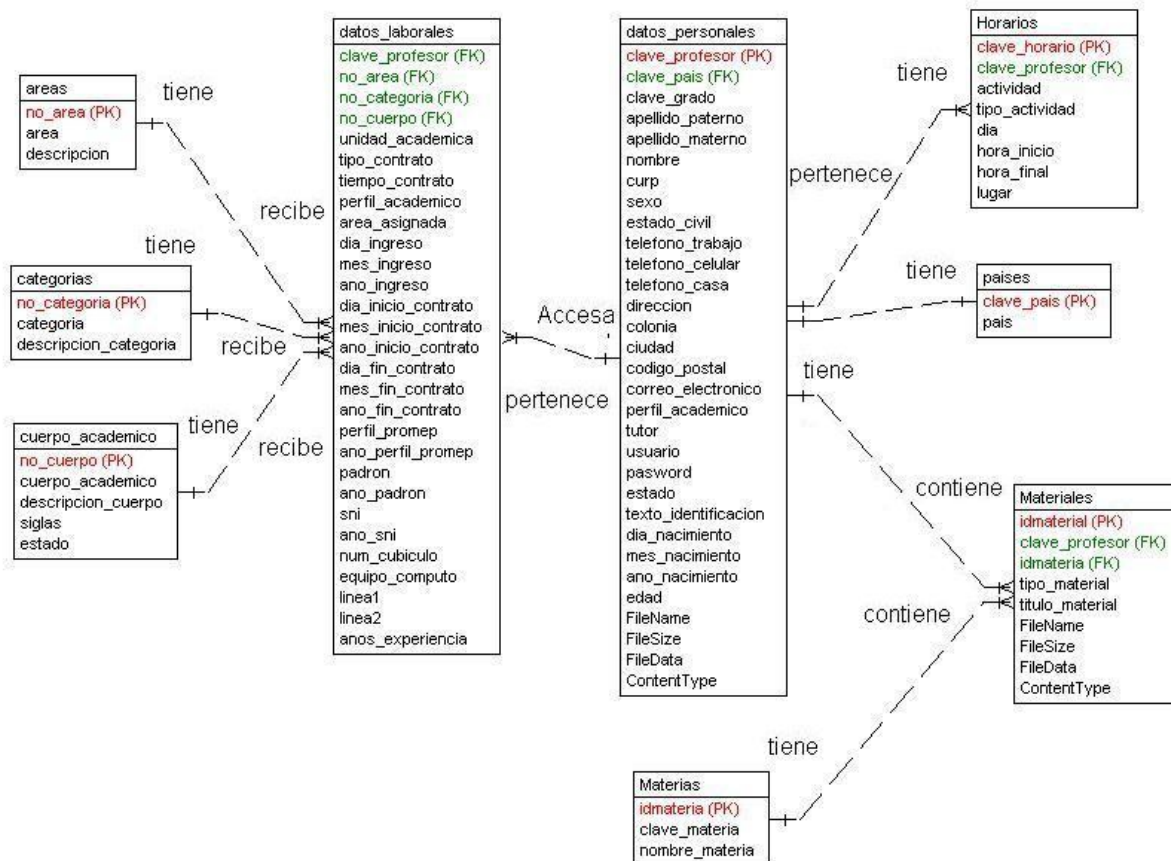


Diagrama Modelo Entidad-Relación 2.1.2

2.2 Análisis Funcional.

Por lo anterior, la facultad requiere de profesores por lo que la facultad requiere de un sitio WEB que apoye a los profesores a informar acerca de los Productos Educativos con los que cuentan para el apoyo e impartición de su materia. En la actualidad los alumnos no saben con que material cuentan de las diferentes materias

Se ve la necesidad de contar con un sistema WEB que permita al alumno bajar cualquier material que esté disponible, que haya realizado el profesor y que le sirva en cualquier materia que esté cursando de la carrera. Los profesores tienen programas, notas, audiovisuales de las diferentes materias que imparten, pero el material con el que cuentan los profesores lo tienen resguardado en su equipo de cómputo y muchas veces el alumno o el profesor por falta de tiempo no comparte el recurso, con esto se pretende mejorar la problemática para brindar un mejor servicio con los alumnos.

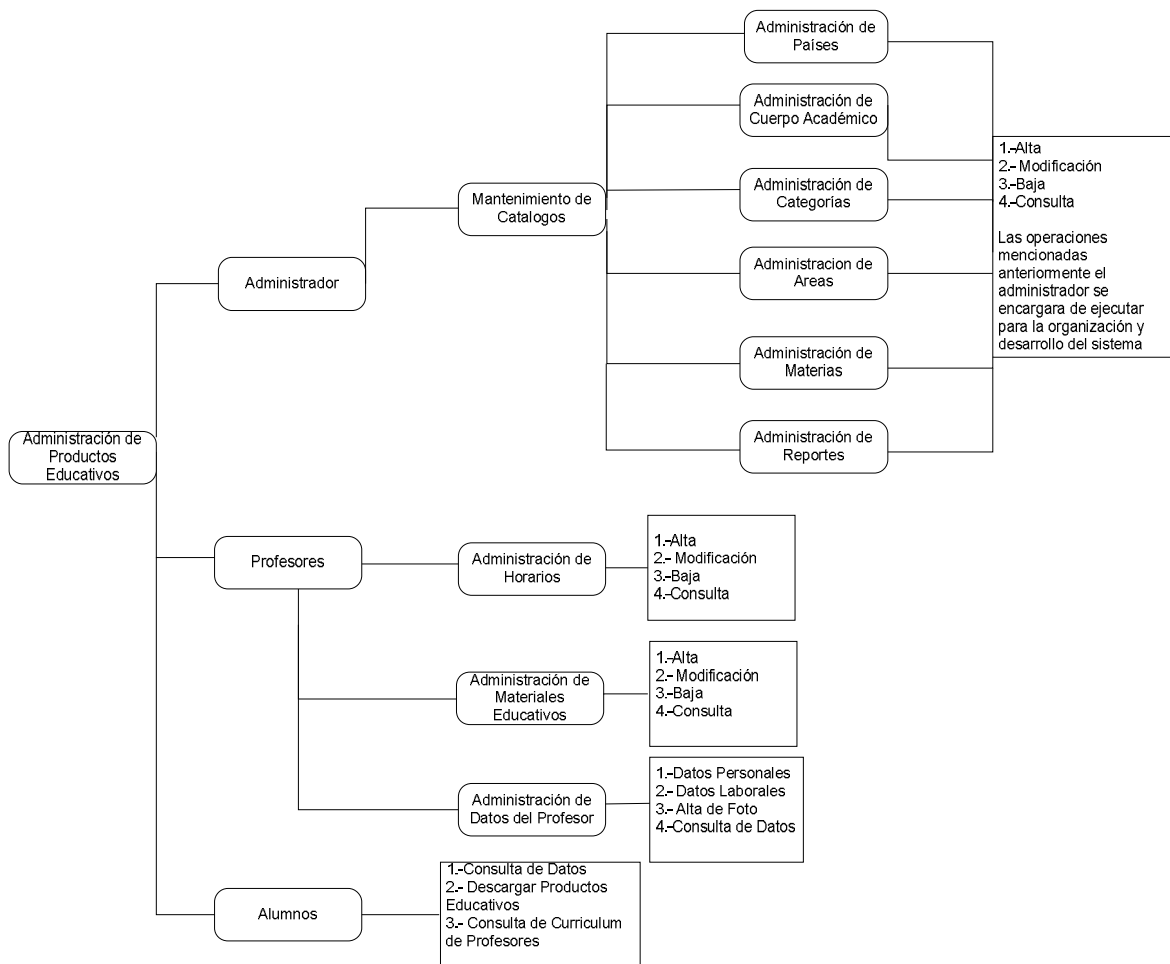


Diagrama Funcional 2.2.1

2.2.1 Actores.

Después de analizar cada una de las funciones que debe desempeñar el sistema, se puede ver que hay tres tipos de usuarios que van a interactuar en él, los cuales son:

- ❑ Administrador
- ❑ Profesor
- ❑ Alumno

Cada uno de ellos tendrá un desempeño diferente, lo cual se explica a continuación en la descripción de los actores

2.2.1.1 Descripción de actores.

Administrador

El actor Administrador, ingresa a la página introduciendo un login y password que lo identifique, una vez que haya pasado la verificación de sus datos, puede realizar las consultas necesarias. Además puede realizar operaciones de inserción, modificación o eliminación de profesores, el cual tendrá el control absoluto del sistema.

Profesor

El actor administrador, al igual que el Administrador, ingresa a la página introduciendo un login y password que lo identificará como profesor, puede realizar las mismas operaciones de inserción, modificación o eliminación de toda su administración de sus Productos Educativos, Materias, Horarios, Proyectos, Publicaciones y datos personales del docente.

Alumno

Este actor puede ingresar a la página sin introducir un login y password y sólo tendrá derecho a la consulta de los Materiales Didácticos, Horarios y el Currículo del Profesor.

Después de saber quiénes van a interactuar con el sistema, se debe analizar el problema para saber de qué forma van a interactuar dichos actores, esto nos lleva a la descripción de lo que van a solicitar del sistema o que van a requerir del sistema, para lo cual se identificaron los siguientes casos de uso. Para representar a los actores en nuestros diagramas utilizaremos las siguientes figuras.

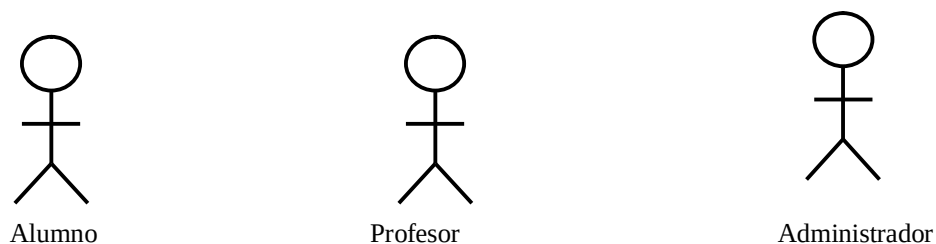


Figura Representación de actores en los diagramas 2.2.1.1

Esta es la representación gráfica en la notación de UML de cada actor identificado en el sistema y que ya se han descrito con anterioridad.

2.2.2 Casos de uso.

Actor	Caso de Uso
Administrador	Login
	Catalogo de Profesor
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Materias
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Áreas
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Categorías
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Cuerpo Académico
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Países
	Alta
	Actualización
	Eliminación
	Consulta

Profesor	Login
	Datos Personales
	Alta
	Actualización
	Eliminación
	Consulta
	Datos Laborales
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Horarios
	Alta
	Actualización
	Eliminación
	Consulta
	Catalogo de Material Didáctico
	Alta
	Eliminación
	Consulta
Alumno	Consulta Material Didáctico
	Consulta Horario
	Consulta Datos Profesor

Tabla Caso de Uso 2.2.2

2.2.3 Diagramas de Casos de Uso.

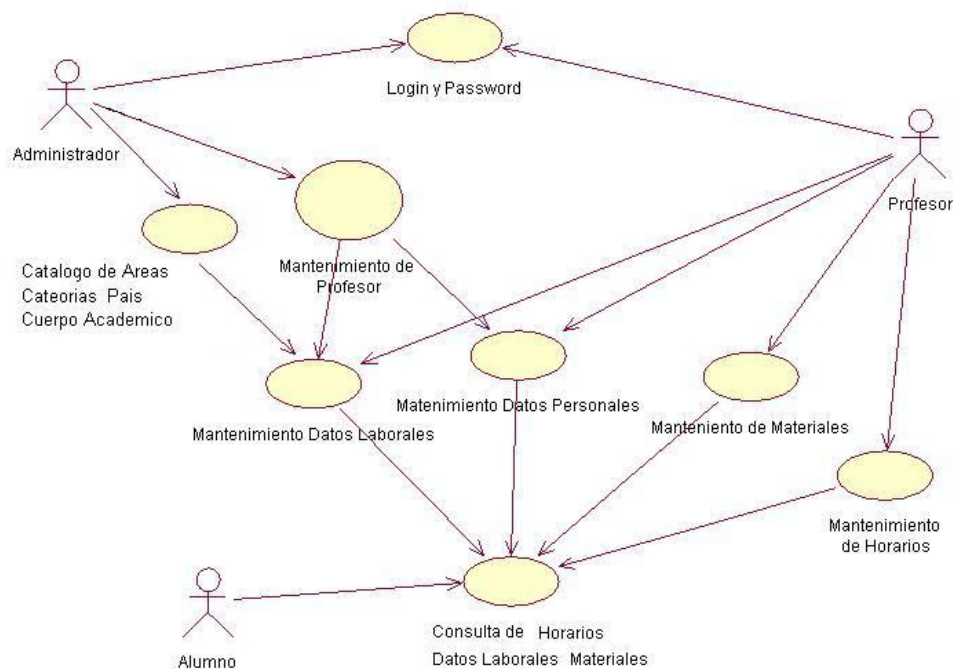


Figura 2.2.3 Diagrama de Caso de Uso

2.2.4 Especificación de Casos de Uso.

Esos son los actores y los casos de uso que interactuarán con el sistema, ahora se mostrará gráficamente la forma en la que interactúan entre sí.

Login.

Actores: Usuarios.

Propósito:

Realizar la correcta autenticación de un usuario en la base de datos.

Visión General:

Un usuario ingresa a la página tecleando su nombre de usuario y contraseña, para visualizar, ingresar, actualizar o generar algún reporte referente a una estimación determinada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ El usuario debe de estar dado del alta correctamente en la base de datos principal.

Poscondiciones:

- ✓ El Usuario fue validado con éxito en la base de datos e ingresa a la página principal.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.- El caso de uso inicia cuando un usuario ingresa a la página principal para que se autentifique. 2.- El usuario teclea su nombre y contraseña. 4.-El Usuario ingresa correctamente a la página principal, en donde visualiza las obras.	3.-El sistema verifica que el usuario y contraseña sean válidos

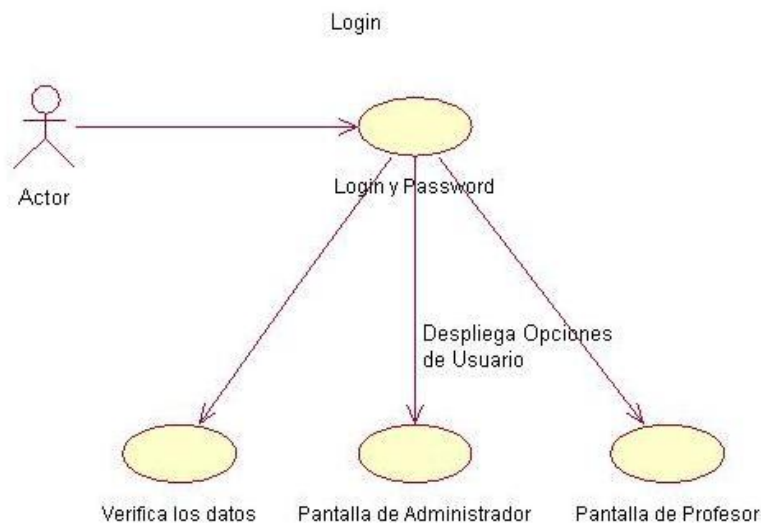


Figura 2.2.4 Especificación Diagrama de Caso de Uso Login

2.2.5 Caso de Uso de Profesor

Alta de Profesor.

Actores:

Administrador.

Propósito:

Realizar la alta del profesor.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El profesor debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor ingresará al sistema Web.
- ✓ El profesor ingresará datos personales, materiales didácticos y horarios.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va ingresa la alta del profesor. 2.- El usuario llena el formulario y elige la opción aceptar. 4.- El usuario visualiza al profesor ingresado.	 3.- El sistema muestra la lista de profesores dados de alta.

Consulta de Profesor.

Actores: Administrador.

Propósito:

Realizar una consulta de un profesor para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del profesor deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos del profesor satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Actualizar Profesor.

Actores:

Administrador.

Propósito:

Realizar la actualización del profesor.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ El profesor se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a actualizar un dato del profesor. 2.- El usuario selecciona el profesor. 3.- El usuario ingresa los nuevos datos a actualizar o modificar. 5.- El usuario visualiza los datos del profesor actualizado. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.

Eliminar Profesor.

Actores:

Administrador.

Propósito:

Realizar la baja del profesor.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ El profesor se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja a un profesor.	
2.- El usuario selecciona el profesor.	
3.- El usuario selecciona la opción baja del profesor.	
5.- El usuario visualiza la lista actualizada.	
6.-El usuario cierra la sesión iniciada.	
	4.- El sistema actualiza correctamente la base de datos.

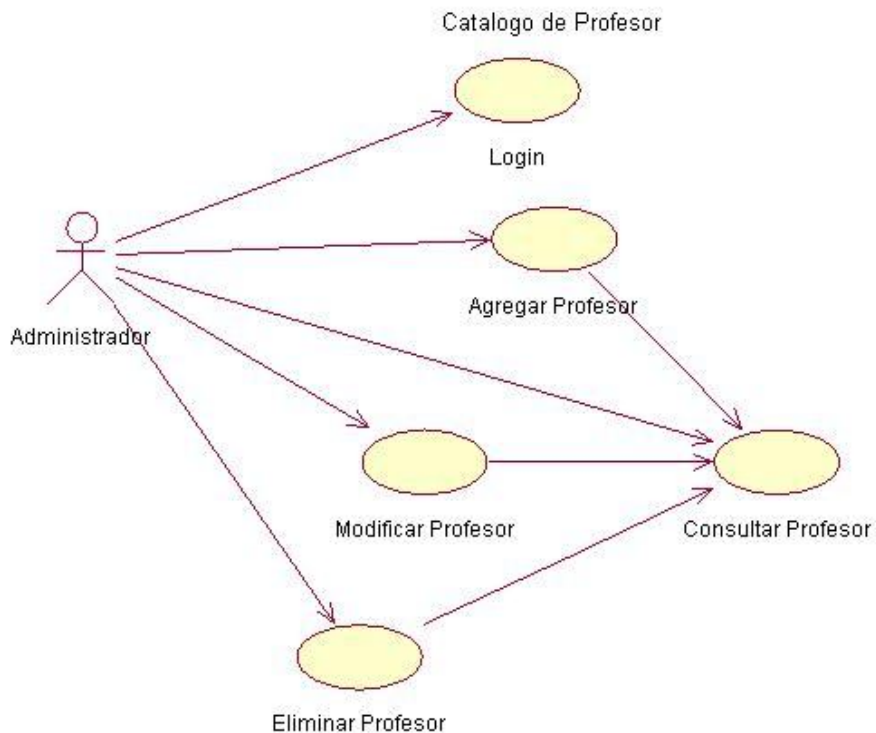


Figura 2.2.5 Especificación Diagrama de Caso de Uso Profesor (B)

2.2.6 Caso de Uso de Materia

Alta de Materia.

Actores:

Administrador.

Propósito:

Realizar el alta de la materia.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ La materia debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor seleccionara la materia para dar de alta el material didáctico.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa la alta de la materia. 2.- El usuario a de llenar el formulario de alta y elige la opción aceptar. 4.- El usuario visualiza la materia ingresada.	3.- El sistema muestra la lista de las materias dadas de alta.

Consulta de Materia.

Actores: Administrador.

Propósito:

Realizar una consulta de las materias para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos de la materia deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos de la materia satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
------------------	-----------------------

1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta. 5.-El usuario visualiza la información solicitada. 6.-El usuario cierra su sesión	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario. 4.-El sistema muestra los datos de la consulta solicitada.
---	--

Actualizar Materias.

Actores:

Administrador.

Propósito:

Realizar la actualización de la materia.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ La materia debe existir en la base de datos.

Poscondiciones:

- ✓ La materia se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza la materia. 2.- El usuario selecciona la materia. 3.- El usuario ingresa los nuevos datos a	

actualizar. 5.- El usuario visualiza los datos de la materia actualizados. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.
---	--

Eliminar Materia.

Actores:

Administrador.

Propósito:

Realizar la baja de la materia.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ La materia debe existir en la base de datos.

Poscondiciones:

- ✓ La materia se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja la materia. 2.- El usuario selecciona la materia. 3.- El usuario selecciona la opción baja de materia. 5.- El usuario visualiza la lista actualizada. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.

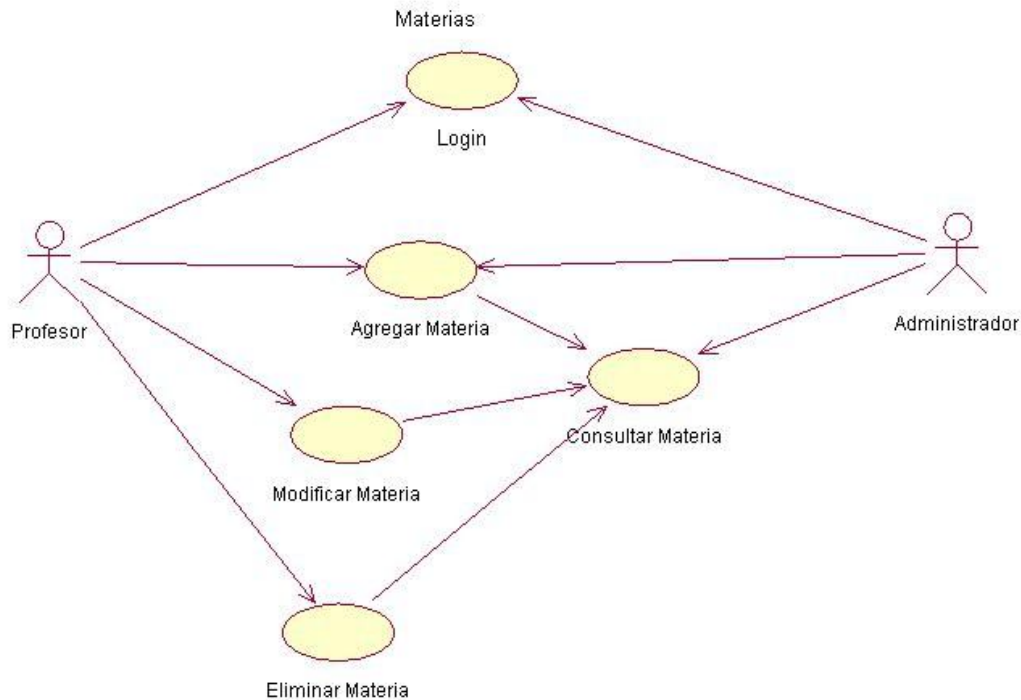


Figura 2.2.6 Especificación Diagrama de Caso de Uso Materia

2.2.7 Caso de Uso de Área

Alta de Áreas.

Actores:

Administrador.

Propósito:

Realizar el alta del área.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El área debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor seleccionara el área para dar de alta los datos laborales.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta del área. 2.- El usuario llena el formulario y elige la opción aceptar. 4.- El usuario visualiza el área ingresada.	3.- El sistema muestra la lista de las áreas dadas de alta.

Consulta del Área.

Actores: Administrador.

Propósito:

Realizar una consulta de las áreas para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del área deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos del área satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	2.- El sistema hace una petición al

5.-El usuario visualiza la información solicitada. 6.-El usuario cierra su sesión	servidor para ejecutar la consulta del usuario. 4.-El sistema muestra los datos de la consulta solicitada.
---	--

Actualizar Área.

Actores:

Administrador.

Propósito:

Realizar la actualización del área.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El área debe existir en la base de datos.

Poscondiciones:

- ✓ El área se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza el área. 2.- El usuario selecciona el área. 3.- El usuario ingresa los nuevos datos a actualizar. 5.- El usuario visualiza los datos de la materia actualizados.	4.- El sistema actualiza correctamente la base de datos.

6.-El usuario cierra la sesión iniciada.	
--	--

Eliminar Área.

Actores:

Administrador.

Propósito:

Realizar la baja del área.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El área debe existir en la base de datos.

Poscondiciones:

- ✓ El área se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja el área.	
2.- El usuario selecciona el área.	
3.- El usuario selecciona la opción baja de área.	
5.- El usuario visualiza la lista actualizada.	
6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.

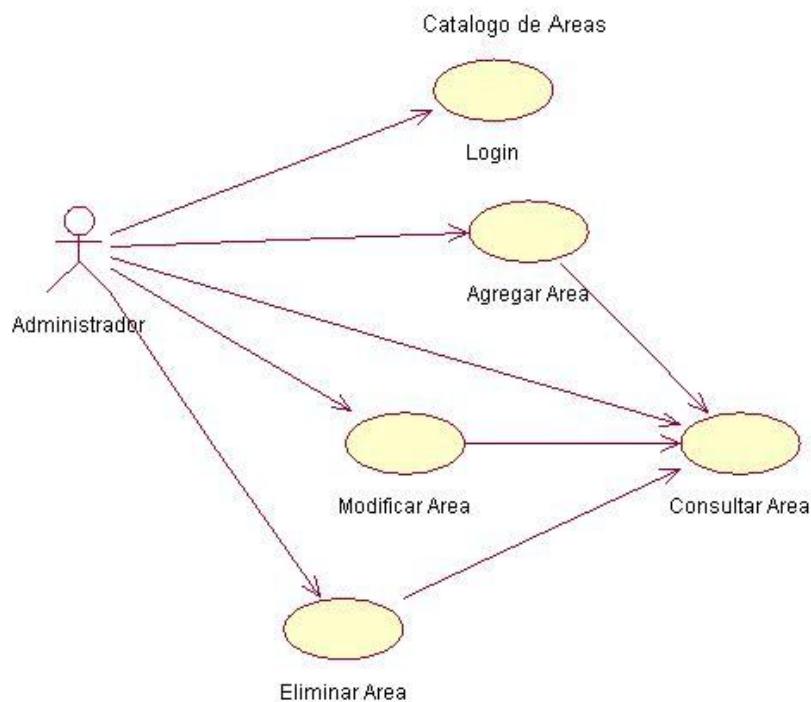


Figura 2.2.7 Especificación Diagrama de Caso de Uso Áreas

2.2.8 Caso de Uso de Categoría

Alta de Categoría.

Actores:

Administrador.

Propósito:

Realizar el alta de la categoría.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ La categoría debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor seleccionara la categoría para dar de alta los datos laborales.

Sistema Web que Administra los Productos Educativos de los Profesores del Área de Programación

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta a la categoría. 2.- El usuario llena el formulario y elige la opción aceptar. 4.- El usuario visualiza la categoría ingresada.	3.- El sistema muestra la lista de las categorías dadas de alta.

Consulta de la Categoría.

Actores: Administrador.

Propósito:

Realizar una consulta de las categorías para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos de la categoría deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos de la categoría satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	

5.-El usuario visualiza la información solicitada. 6.-El usuario cierra su sesión	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario. 4.-El sistema muestra los datos de la consulta solicitada.
---	--

Actualizar Categoría.

Actores:

Administrador.

Propósito:

Realizar la actualización de la categoría.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ La categoría debe existir en la base de datos.

Poscondiciones:

- ✓ La categoría se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza la categoría. 2.- El usuario selecciona la categoría. 3.- El usuario ingresa los nuevos datos a actualizar.	4.- El sistema actualiza correctamente la base de datos.

5.- El usuario visualiza los datos de la categoría actualizados.	
6.-El usuario cierra la sesión iniciada.	

Eliminar Categoría.

Actores:

Administrador.

Propósito:

Realizar la baja de la categoría.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ La categoría debe existir en la base de datos.

Poscondiciones:

- ✓ La categoría se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja la categoría.	
2.- El usuario selecciona la categoría.	
3.- El usuario selecciona la opción baja de categoría.	
5.- El usuario visualiza la lista actualizada.	
6.-El usuario cierra la sesión iniciada.	
	4.- El sistema actualiza correctamente la base de datos.

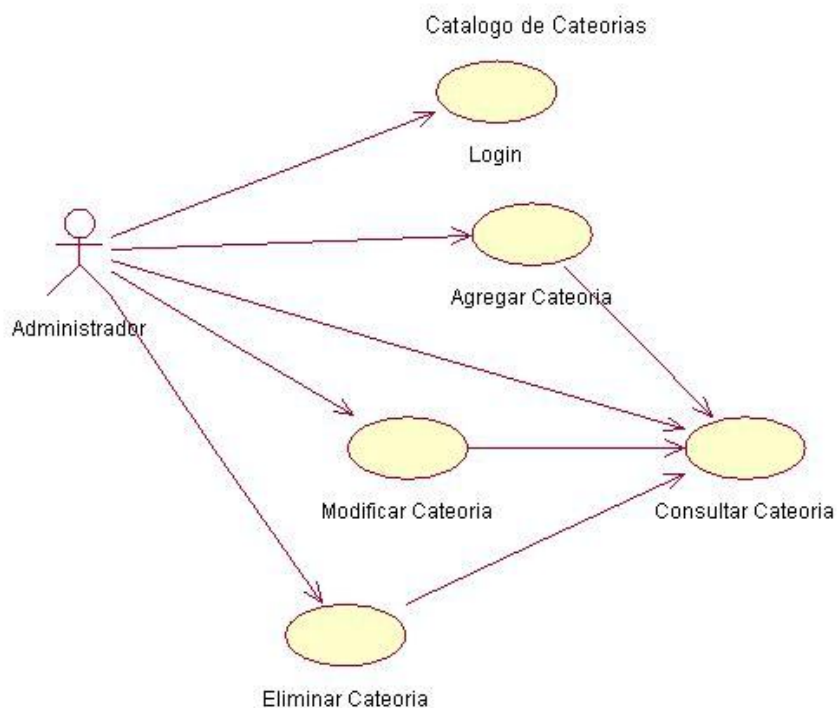


Figura 2.2.8 Especificación Diagrama de Caso de Uso Categorías

2.2.9 Caso de Uso de Cuerpo Académico

Alta de Cuerpo Académico.

Actores:

Administrador.

Propósito:

Realizar el alta del cuerpo académico.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El cuerpo académico debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor seleccionará el cuerpo académico dando de alta los datos laborales.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta el cuerpo académico.	
2.- El usuario llena el formulario y elige la opción aceptar.	
4.- El usuario visualiza el cuerpo académico ingresado.	3.- El sistema muestra la lista del cuerpo académico dado de alta.

Consulta del Cuerpo Académico.

Actores: Administrador.

Propósito:

Realizar una consulta del cuerpo académico para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del cuerpo académico deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos del cuerpo académico satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
5.-El usuario visualiza la información solicitada.	4.-El sistema muestra los datos de la consulta solicitada.
6.-El usuario cierra su sesión	

Actualizar Cuerpo Académico.

Actores:

Administrador.

Propósito:

Realizar la actualización del cuerpo académico.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El cuerpo académico debe existir en la base de datos.

Poscondiciones:

- ✓ El cuerpo académico se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza los datos al cuerpo académico.	

2.- El usuario selecciona el cuerpo académico. 3.- El usuario ingresa los nuevos datos a actualizar. 5.- El usuario visualiza los datos del cuerpo académico actualizados. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.
--	--

Eliminar Cuerpo Académico.

Actores:

Administrador.

Propósito:

Realizar la baja del cuerpo académico.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El cuerpo académico debe existir en la base de datos.

Poscondiciones:

- ✓ El cuerpo académico se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja el cuerpo académico. 2.- El usuario selecciona el cuerpo académico. 3.- El usuario selecciona la opción baja	

del cuerpo académico.	
5.- El usuario visualiza la lista actualizada.	4.- El sistema actualiza correctamente la base de datos.
6.-El usuario cierra la sesión iniciada.	

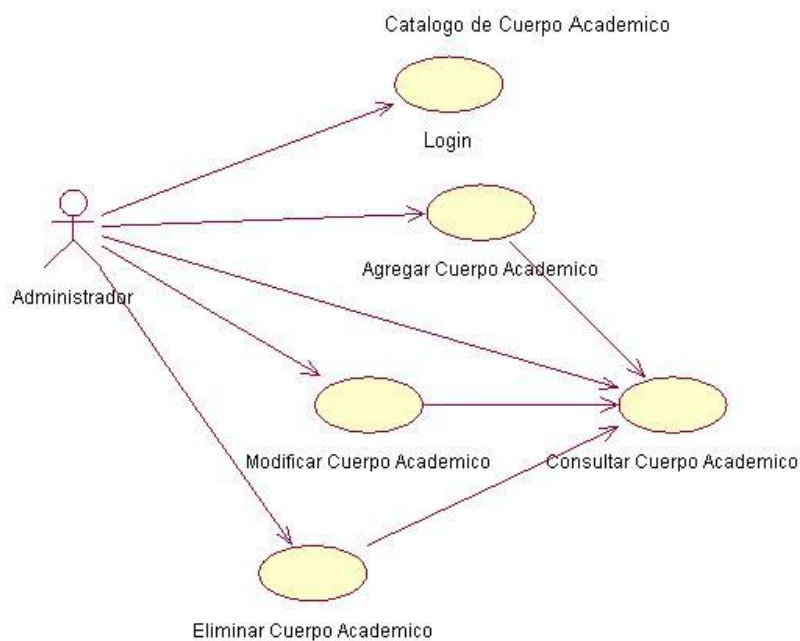


Figura 2.2.9 Especificación Diagrama de Caso de Uso Cuerpo Académico

2.2.10 Caso de Uso de País

Alta de País.

Actores:

Administrador.

Propósito:

Realizar la alta de país.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El país debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor seleccionará el país al dar de alta los datos personales.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta el país.	
2.- El usuario llena el formulario y elige la opción aceptar.	
	3.- El sistema actualiza la lista del país dado de alta.
4.- El usuario visualiza el país ingresado.	

Consulta del País.

Actores: Administrador.

Propósito:

Realizar una consulta del país para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del país deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos del país satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Actualizar País.

Actores:

Administrador.

Propósito:

Realizar la actualización del país.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El país debe existir en la base de datos.

Poscondiciones:

- ✓ El país se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza el país.	

2.- El usuario selecciona el país 3.- El usuario ingresa los nuevos datos a actualizar. 5.- El usuario visualiza los datos del país actualizado. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.
--	--

Eliminar País.

Actores:

Administrador.

Propósito:

Realizar la baja del país.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El país debe existir en la base de datos.

Poscondiciones:

- ✓ El país se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja el país. 2.- El usuario selecciona el país. 3.- El usuario selecciona la opción baja del país.	

5.- El usuario visualiza la lista actualizada.	4.- El sistema actualiza correctamente la base de datos.
6.-El usuario cierra la sesión iniciada.	

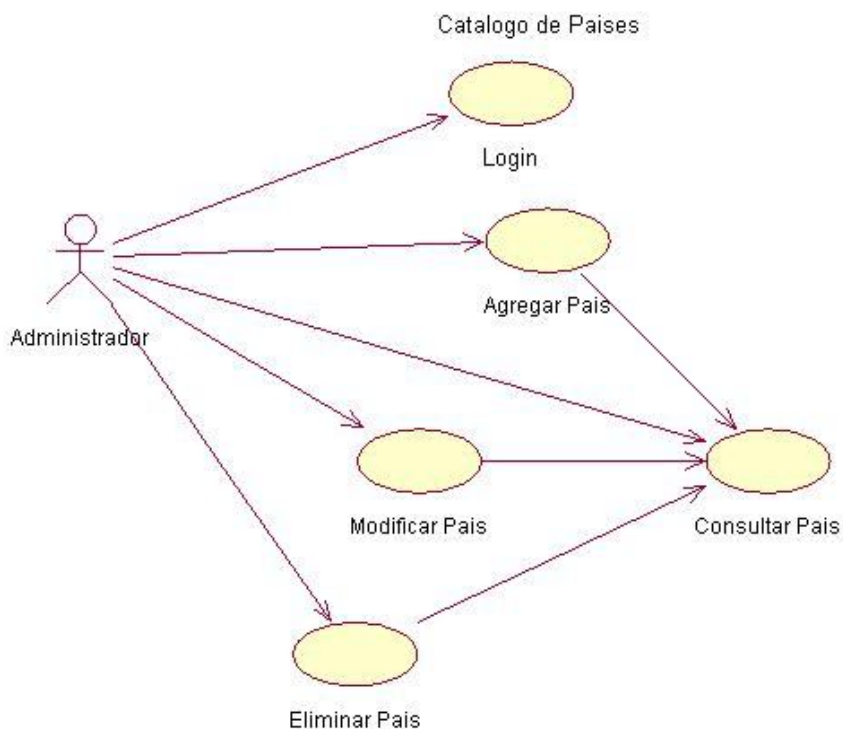


Figura 2.2.10 Especificación Diagrama de Caso de Uso País

2.2.11 Caso de Datos Personales

Alta de Datos Personales.

Actores:

Profesor.

Propósito:

Realizar el alta de sus datos personales.

Visión General:

Tipo: Primario y esencial.

Sistema Web que Administra los Productos Educativos de los Profesores del Área de Programación

Precondiciones:

- ✓ El profesor debe de existir en la base de datos.

Poscondiciones:

- ✓ El administrador podrá visualizar los datos para el control del sistema.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa la alta de sus datos personales.	
2.- El usuario llena el formulario y elige la opción aceptar.	
	3.- El sistema actualiza la base de datos.
4.- El usuario visualiza los datos ingresados.	

Consulta del Datos Personales.

Actores:

Profesor, Alumno.

Propósito:

Realiza una consulta del la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos personales deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos personales satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Actualizar Datos Personales.

Actores:

Profesor.

Propósito:

Realizar la actualización de los datos personales.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ Los datos personales se actualizaron correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza los datos personales. 2.- El usuario ingresa los nuevos datos a actualizar. 4.- El usuario visualiza los datos actualizados. 6.-El usuario cierra la sesión iniciada.	3.- El sistema actualiza correctamente la base de datos.

Eliminar Datos Personales.

Actores:

Profesor.

Propósito:

Realizar la baja de los datos personales.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ Los datos se dan de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja sus datos personales. 2.- El usuario selecciona los datos.	

3.- El usuario selecciona la opción borrar. 5.- El usuario visualiza los datos actualizados. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.
--	---

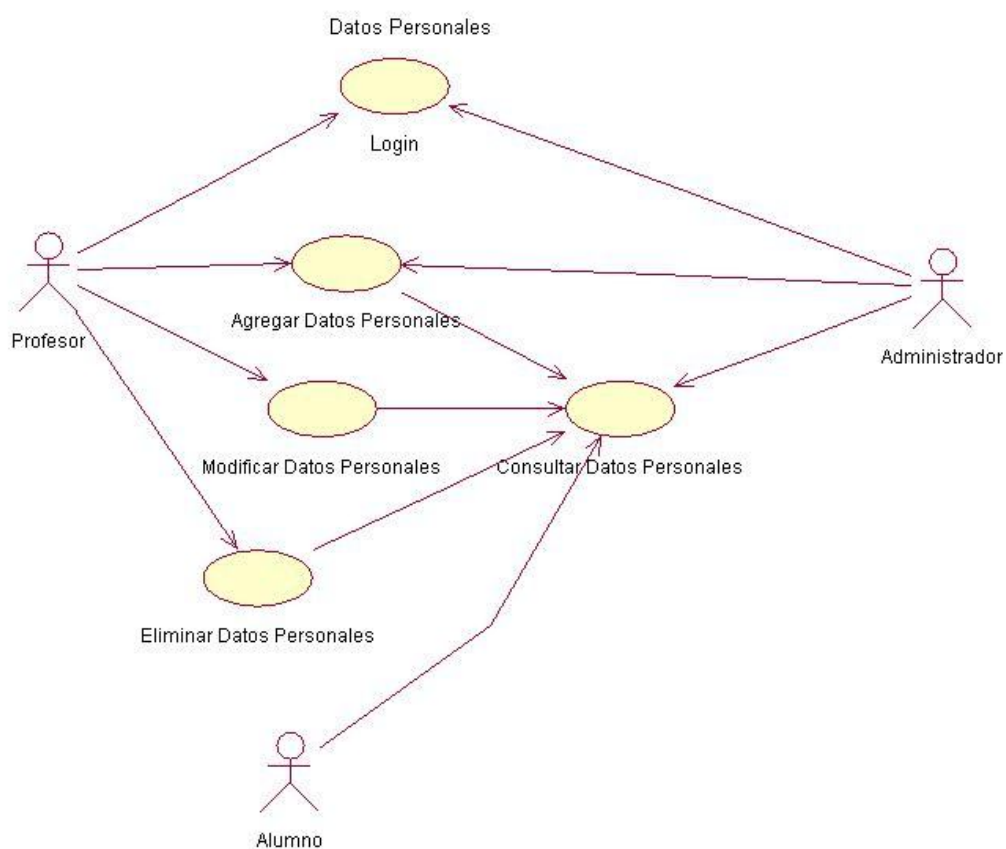


Figura 2.2.11 Especificación Diagrama de Caso de Uso Datos Personales

2.2.12 Caso de Datos Laborales

Alta de Datos Laborales.

Actores:

Profesor, Alumno.

Propósito:

Realizar la alta sus datos laborales.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El profesor debe de existir en la base de datos.

Poscondiciones:

- ✓ El alumno tendrá acceso a ciertos datos del profesor.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta sus datos laborales.	
2.- El usuario llena el formulario y elige la opción aceptar.	
	3.- El sistema actualiza la base de datos.
4.- El usuario visualiza los datos ingresados.	

Consulta del Datos Laborales.

Actores:

Profesor, Alumno.

Propósito:

Realiza una consulta de la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos laborales deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos laborales satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Actualizar Datos Laborales.

Actores:

Profesor.

Propósito:

Realizar la actualización de los datos laborales.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ Los datos laborales se actualizaron correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza los datos laborales. 2.- El usuario ingresa los nuevos datos a actualizar. 4.- El usuario visualiza los datos actualizados. 5.-El usuario cierra la sesión iniciada.	3.- El sistema actualiza correctamente la base de datos.

Eliminar Datos Laborales.

Actores:

Profesor.

Propósito:

Realizar la baja de los datos laborales.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El profesor debe existir en la base de datos.

Poscondiciones:

- ✓ Los datos se dan de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja sus datos laborales. 2.- El usuario selecciona los datos.	

3.- El usuario selecciona la opción borrar.	
5.- El usuario visualiza los datos actualizados.	
6.-El usuario cierra la sesión iniciada.	
	4.- El sistema actualiza correctamente la base de datos.

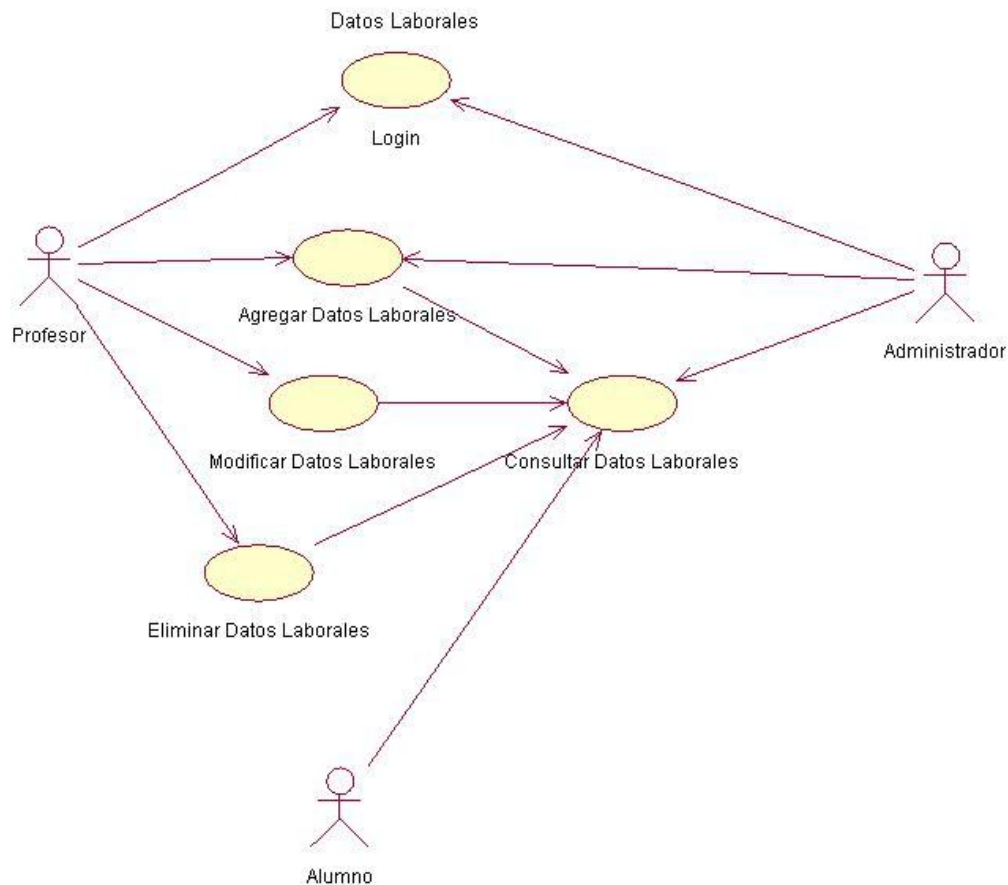


Figura 2.2.12 Especificación Diagrama de Caso de Uso Datos Laborales

2.2.13 Caso de Uso de Horario

Alta de Horario.

Actores:

Profesor, Alumno, Administrador.

Propósito:

Realizar la alta del horario.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El profesor debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor podrá dar de alta su horario.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta el horario.	
2.- El usuario llena el formulario y elige la opción aceptar.	
	3.- El sistema muestra la lista del horario dado de alta.
4.- El usuario visualiza el horario ingresado.	

Consulta del Horario.

Actores:

Profesor, Alumno.

Propósito:

Realizar una consulta del horario para visualizar la información deseada.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del área deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los datos del área satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Actualizar Horario.

Actores:

Profesor.

Propósito:

Realizar la actualización del horario.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El horario debe existir en la base de datos.

Poscondiciones:

- ✓ El horario se actualizó correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario actualiza el horario. 2.- El usuario selecciona el horario. 3.- El usuario ingresa los nuevos datos a actualizar. 5.- El usuario visualiza los datos actualizados. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.

Eliminar Horario.

Actores:

Profesor.

Propósito:

Realizar la baja del horario.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El horario debe existir en la base de datos.

Poscondiciones:

- ✓ El horario se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja el horario. 2.- El usuario selecciona el horario.	

3.- El usuario selecciona la opción baja de horario. 5.- El usuario visualiza la lista actualizada. 6.-El usuario cierra la sesión iniciada.	4.- El sistema actualiza correctamente la base de datos.
---	---

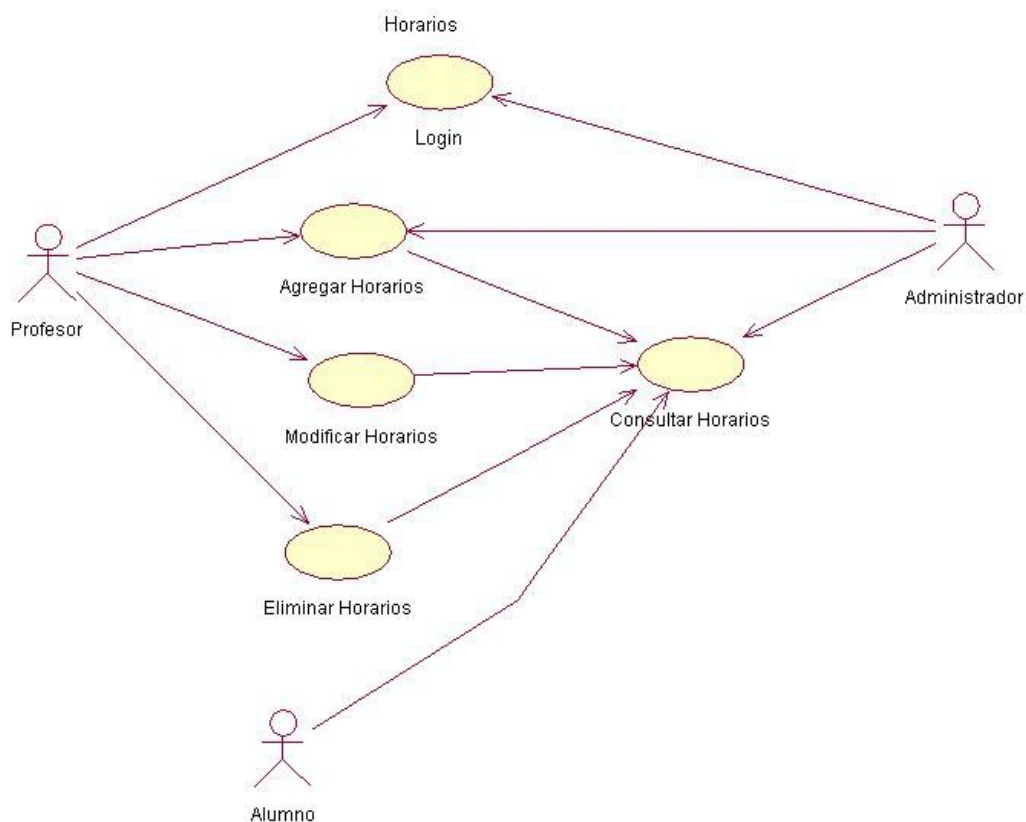


Figura 2.2.13 Especificación Diagrama de Caso de Uso Horario

2.2.14 Caso de Uso de Material Didáctico.

Alta de Material Didáctico.

Actores:

Profesor.

Propósito:

Realizar el alta del material didáctico.

Visión General:

Tipo: Primario y esencial.

Precondiciones:

- ✓ El profesor debe de existir en la base de datos.

Poscondiciones:

- ✓ El profesor podrá dar de alta el material didáctico.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario ingresa a dar de alta el material didáctico.	
2.- El usuario llena el formulario y elige el material que dará de alta y elige opción aceptar.	
4.- El usuario visualiza el material ingresado.	3.- El sistema muestra la lista del material didáctico.

Consulta del Material Didáctico.

Actores:

Profesor, Alumno.

Propósito:

Realizar una consulta del material didáctico.

Visión General:

El usuario del sistema selecciona la opción de consulta del menú principal de la página y visualiza la información solicitada.

Tipo: Primario y esencial.

Precondiciones:

- ✓ Los datos del material didáctico deben estar correctamente actualizados en la base de datos.

Poscondiciones:

- ✓ El usuario pudo consultar y visualizar los materiales didácticos satisfactoriamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario selecciona la opción de consulta.	
	2.- El sistema hace una petición al servidor para ejecutar la consulta del usuario.
	4.-El sistema muestra los datos de la consulta solicitada.
5.-El usuario visualiza la información solicitada.	
6.-El usuario cierra su sesión	

Eliminar Material Didáctico.

Actores:

Profesor.

Propósito:

Realizar la baja del material didáctico.

Visión General:

Tipo:

Primario y esencial.

Precondiciones:

- ✓ El material debe existir en la base de datos.

Poscondiciones:

- ✓ El material se da de baja correctamente.

Curso Típico de Eventos.

Acción del Actor	Respuesta del sistema
1.-El caso de uso inicia cuando un usuario va a dar de baja el material didáctico.	
2.- El usuario selecciona el material didáctico.	
3.- El usuario selecciona la opción baja de material didáctico.	
5.- El usuario visualiza la lista actualizada.	4.- El sistema actualiza correctamente la base de datos.
6.-El usuario cierra la sesión iniciada.	

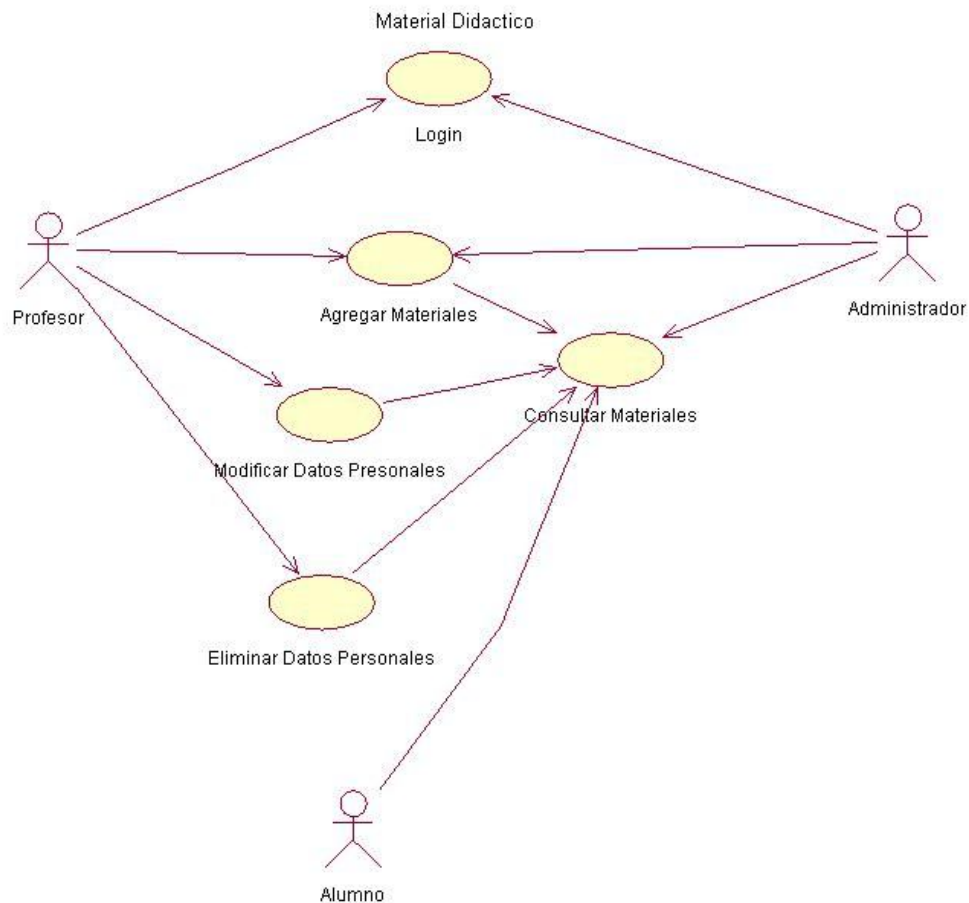


Figura 2.2.14 Especificación Diagrama de Caso de Uso Material Didáctico

2.3 Identificación de clases.

En la siguiente figura 2.3 mostraré las clases descritas anteriormente en los casos de uso, se hace la identificación mas específica de las Entidades y Atributos.

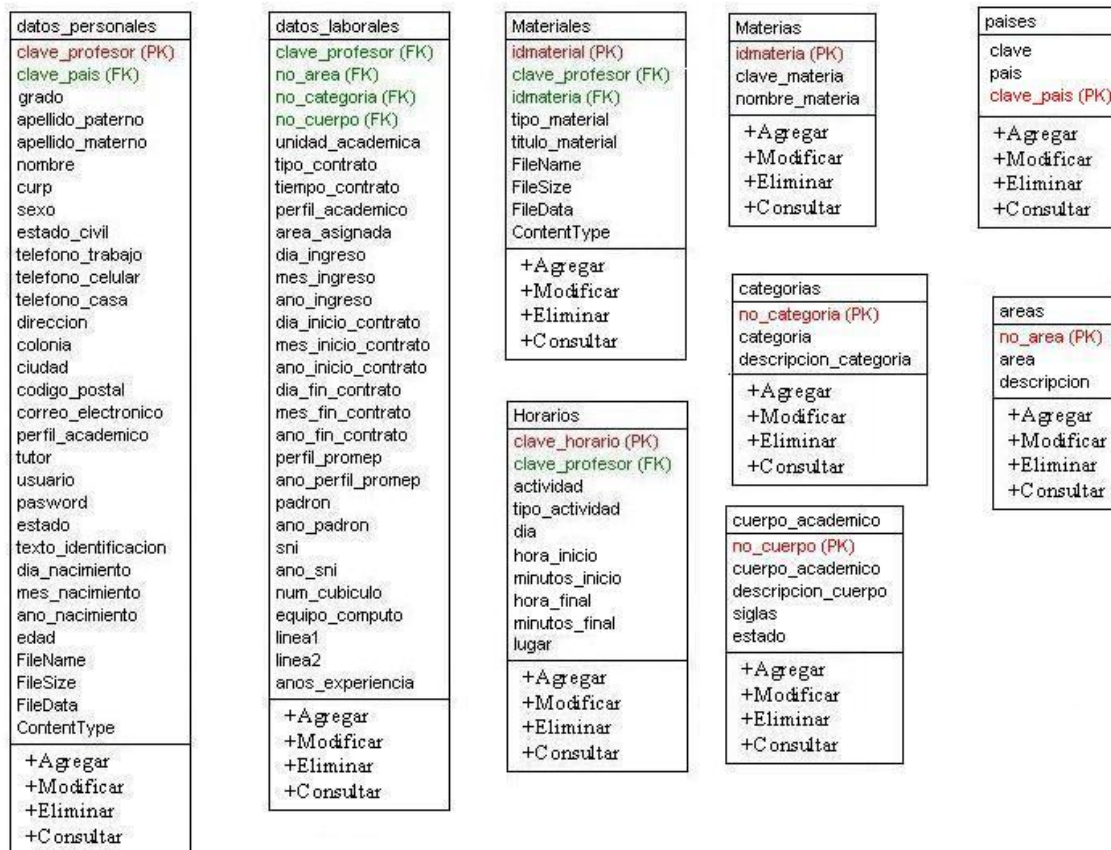


Figura 2.3 Identificación de Clases

Capítulo III Diseño.

El diseño de un sistema de información responde a la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis.

Es común que los diseñadores hagan un esquema del formato o pantalla que esperan que aparezca cuando el sistema está terminado, se realiza en papel o en la pantalla de una terminal utilizando algunas de las herramientas automatizadas disponibles para el desarrollo de sistemas.

También se indican los datos de entrada, los que serán calculados y los que deben ser almacenados. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento. Los procedimientos que se escriben indican cómo procesar los datos y producir salidas.

3.1 Diseño Conceptual.

Cuando se trabaja bajo el análisis conceptual de una situación, nos referimos a la abstracción de hechos reales de los cuales se emite un concepto o es posible hacer una idea de ello. Estos requerimientos contienen el conjunto de hechos y reglas que dan pauta a la creación del esquema conceptual, donde por medio de éste se podrá realizar una descripción de alto nivel de la futura base de datos.

3.1.1 Normalización.

Normalización es un conjunto de reglas que nos ayudará para desarrollar un esquema que minimice los problemas de lógica. Consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Primera Forma Normal (1FN)

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una clave primaria.
- La tabla no contiene atributos nulos.
- Eliminar grupos repetitivos y formar una nueva tabla.

Segunda Forma Normal (2FN)

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. Los atributos dependen de la clave. Varía la clave y varían los atributos.

Tercera Forma Normal (3FN)

Una relación R está en 3FN si y solo si esta en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria.

Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación esta en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación.

Después de aplicar la Normalización en la Base de datos queda de la siguiente forma (véase en la siguiente figura 3.1)

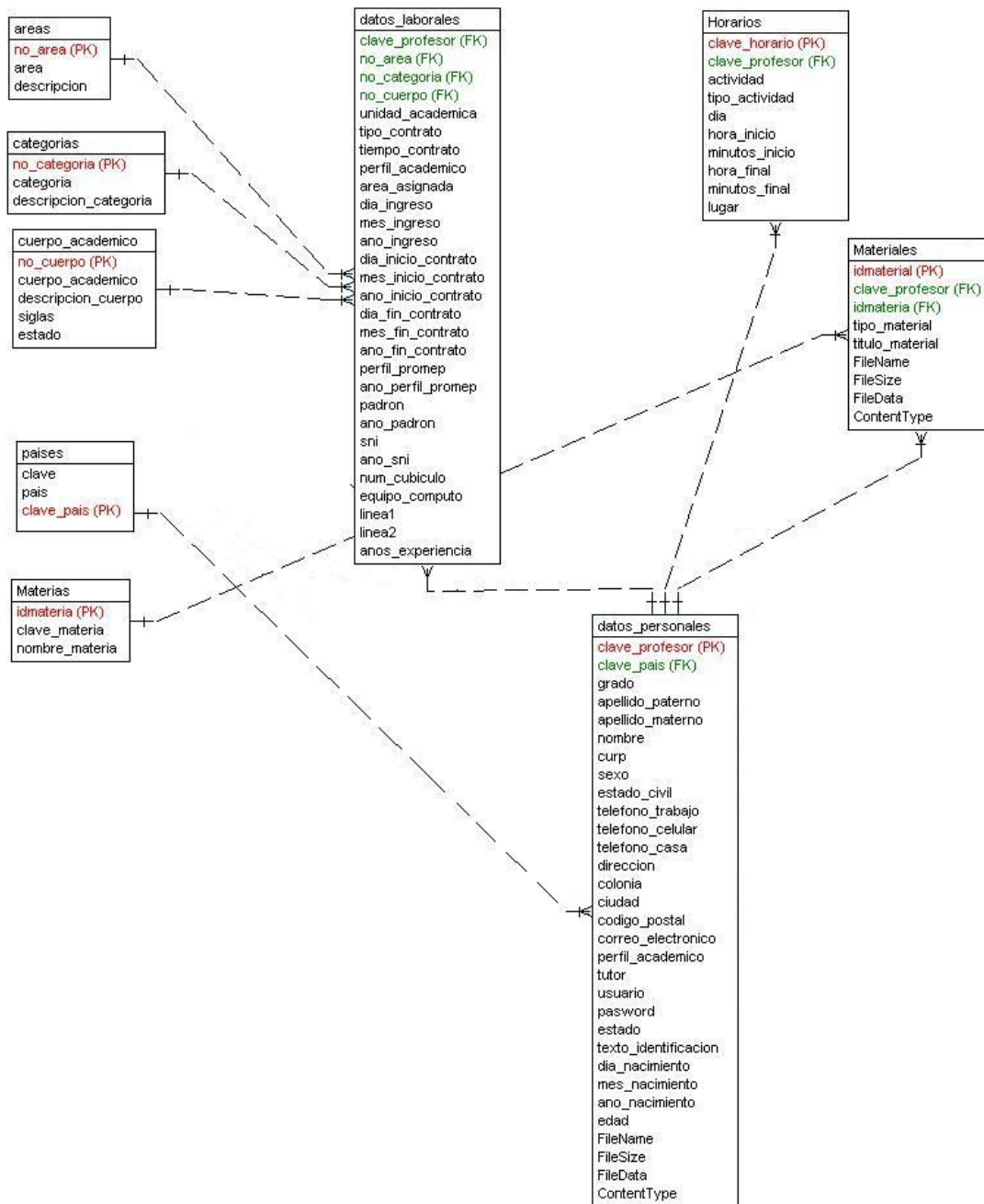


Figura 3.1 Base de Datos Normalizada

3.1.2 Modelo Relacional.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene mayor relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de «consultas» que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real (PK hace referencia a la llave primaria y FK a la llave foránea) Figura Modelo Relacional 3.1.2.

areas	<u>PK(no_area)</u> , area, descripcion
categorias	<u>PK(no_categoria)</u> , categoria, descripcion_categoria
cuerpo_academico	<u>PK(no_cuerpo)</u> , cuerpo_academico, descripcion_cuerpo
datos_laborales	<u>PK(clave_profesor)</u> ,FK(<u>no_area</u> , <u>no_cuerpo</u> , <u>no_categoria</u>), unidad_academica, tipo_contrato, tiempo_contrato, perfil_academico,area_asignada, dia_ingreso, mes_ingreso, ano_ingreso, dia_inicio_contrato, mes_inicio_contrato, ano_inicio_contrato, dia_fin_contrato, mes_fin_contrato, ano_fin_contrato, perfil_promep, ano_perfil_promep, padron, ano_padron, sin, ano_sni, num_cubiculo, equipo_computo, linea,linea2, anos_experiencia
datos_personales	<u>PK(clave_profesor)</u> ,FK(<u>clave_pais</u>), clave_grado, grado, apellido_paterno, apellido_materno, nombre, estado, sexo, tutor, telefono_trabajo, usuario, password, curp, estado_civil, telefono_celular, telefono_casa, direccion, colonia, ciudad, codigo_postal,correo_electronico, perfil_academico, texto_identificacion, FileName, FileSize, FileData, ContentType, dia_nacimiento, mes_nacimiento, ano_nacimiento, edad
horarios	<u>PK(clave_horario)</u> ,FK(<u>clave_profesor</u>),actividad,tipo_actividad, dia, hora_inicio,minutos_inicio, hora_final, minutos_final, lugar
materiales	<u>PK(inmaterial)</u> ,FK(<u>clave_profesor</u> , <u>idmateria</u>), tipo_material, titulo_material, FileName, FileSize, FileData, ContentType
materias	<u>PK(idmateria)</u> , clave_materia, nombre_materia
países	clave, pais, clave_pais

Figura Modelo Relacional 3.1.2

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia, dependencias incoherentes y proteger la integridad de los datos.

3.1.3 Instancias de la Base de Datos.

Una instancia de manera formal es la aplicación de un esquema a un conjunto finito de datos. En palabras no tan técnicas, se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla:

Instancia de Área

No_area	Area	descripcion
1	Entorno Social	Entorno Social
2	Programación Básica	Programación Básica
3	Base de Datos	Base de Datos
4	Software de Base	Software de Base

Figura Instancia área (A)

Instancia de Categoría

no_categoria	categoria	descripcion_categoria
1	PI ASOCIADO A	PI ASOCIADO A
2	PI ASOCIADO B	PI ASOCIADO B
3	PI ASOCIADO C	PI ASOCIADO C
4	PI TITULAR A	PI TITULAR A

Figura Instancia categoría (B)

Instancia de Cuerpo Académico.

no_cuerpo	cuerpo_academico	descripcion_cuerpo	siglas	estado
1	Sistemas de Información	Sistemas de Información	SI	EN FORMACIÓN
2	Cómputo Distribuido	Sistemas Distribuidos	CD	EN CONSOLIDACIÓN
3	Ingeniería Computacional	Ingeniería Computacional	IC	EN FORMACIÓN
4	Computación Matemática	Computación Matemática	CM	EN FORMACIÓN

Figura Instancia cuerpo académico (C)

Instancia de Horario

clave_horario	clave_profesor	actividad	tipo_actividad	dia	hora_inicio	minutos_inicio
1	100210511			3	0	
2	100238799			2	0	
3	100238799			4	0	
8	100203199	YYYY	2	3	4	15

Figura Instancia horario (D)

Instancia de Material Didáctico.

idmaterial	clave_profesor	idmateria	tipo_material	titulo_material	FileName	Fi
1	100408933	2	1	prado	quickstart.pdf	13
2	100195599	6	2	GHJGJHJ	convocatoria_mcts.pdf	83
3	100203199	7	3	CCCCC	ultimos_articulos.doc	38
4	100203199	5	2	bb	ultimos_articulos.doc	38

Figura Instancia material didáctico (E)

Instancia de Materia.

idmateria	clave_materia	nombre_materia
2	PO	Programacion
3	MAT	Matematicas
4	ESP	Español
5	GEO	Geografia

Figura Instancia materia (F)

Instancia de País.

clave	pais	clave_pais
1	Afganistan	1
2	Africa del Sur	2
3	Albania	3
4	Alemania	4

Figura Instancia país (G)

3.1.4 Especificación de Atributos.

En cada modelo de datos a los atributos se les puede asignar un grado de pertenencia que tiene para una entidad y por ende al modelo, estos grados de pertenencia se deben asignar a partir de la especificación de requerimientos del usuario. La importancia que pueden adquirir estos esquemas es que en algún instante se requiera refinar el modelo

de datos a partir de cierto grado de pertenencia dado a los atributos, es así como se puede representar en el esquema aquel atributo que nunca puede ser nulo en la base de datos, o bien, aquellos atributos que se puedan descartar o considerar como nulos.

áreas			
Key	Name	DataType	Length
PK	no_area	Autonumerico	Entero Largo
	area	Text	50
	descripcion	Text	50

Figura Especificación de Atributos áreas (A)

categorias			
Key	Name	DataType	Length
PK	no_categoria	Autonumerico	Entero Largo
	Categoría	Text	50
	descripcion_categoria	Text	50

Figura Especificación de Atributos categorías (B)

Cuerpo academico			
Key	Name	DataType	Length
PK	no_cuerpo	Autonumerico	Entero Largo
	cuerpo_academico	Text	50
	descripcion_cuerpo	Text	50
	Siglas	Text	50
	estado	Text	50

Figura Especificación de Atributos cuerpo académico (C)

datos personales			
Key	Name	DataType	Length
PK	clave_profesor	Integer	Entero Largo
FK	pais	Integer	Entero Largo
	grado	Integer	Entero Largo
	apellido_paterno	Text	100
	apellido_materno	Text	100
	nombre	Text	100
	curp	Text	50
	sexo	Text	2
	estado_civil	Integer	Entero Largo
	telefono_trabajo	Text	50
	telefono_celular	Text	50
	telefono_casa	Text	50

	direccion	Text	50
	colonia	Text	50
	ciudad	Text	50
	codigo_postal	Text	50
	correo_electronico	Text	50
	perfil_academico	Text	50
	tutor	Si/No	
	usuario	Text	50
	password	Text	50
	estado	Text	50
	Text_identificacion	Memo	
	dia_nacimiento	Integer	Entero Largo
	mes_nacimiento	Integer	Entero Largo
	ano_nacimiento	Integer	Entero Largo
	edad	Integer	Entero Largo
	FileName	Text	
	FileSize	Integer	Entero Largo
	FileData	Objeto OLE	
	ContentType	Text	50

Figura Especificación de Atributos datos personales (D)

datos laborales			
Key	Name	DataType	Length
PK	clave_profesor	Integer	Entero Largo
FK	no de area	Integer	Entero Largo
FK	no de categoria	Integer	Entero Largo
FK	no de cuerpo	Integer	Entero Largo
	unidad_academica	Text	50
	tipo_contrato	Integer	Entero Largo
	tiempo_contrato	Integer	Entero Largo
	perfil_academico	Text	50
	dia_ingreso	Integer	Entero Largo
	mes_ingreso	Integer	Entero Largo
	ano_ingreso	Integer	Entero Largo
	dia_inicio_contrato	Integer	Entero Largo
	mes_inicio_contrato	Integer	Entero Largo
	ano_inicio_contrato	Integer	Entero Largo
	dia_fin_contrato	Integer	Entero Largo
	mes_fin_contrato	Integer	Entero Largo
	ano_fin_contrato	Integer	Entero Largo
	perfil_promep	Si/No	
	ano_perfil_promep	Integer	Entero Largo

	padron	Si/No	
	ano_padron	Integer	Entero Largo
	sni	Si/No	
	ano_sni	Integer	Entero Largo
	num_cubiculo	Text	50
	equipo_computo	Si/No	
	linea1	Text	50
	linea2	Text	50
	anos_experiencia	Integer	Entero Largo

Figura Especificación de Atributos datos laborales (E)

horarios			
Key	Name	DataType	Length
PK	clave_horario	Autonumerico	Entero Largo
FK	clave_profesor	Integer	Entero Largo
	actividad	Text	50
	tipo_actividad	Integer	Entero Largo
	dia	Integer	Entero Largo
	hora_inicio	Integer	Entero Largo
	minutos_inicio	Integer	Entero Largo
	hora_final	Integer	Entero Largo
	minutos_final	Integer	Entero Largo
	lugar	Text	50

Figura Especificación de Atributos horarios (F)

materiales			
Key	Name	DataType	Length
PK	idmaterial	Autonumerico	Entero Largo
FK	clave_profesor	Integer	Entero Largo
FK	idmateria	Integer	Entero Largo
	tipo_material	Integer	Entero Largo
	titulo_material	Text	50
	FileName	Text	50
	FileSize	Integer	Entero Largo
	FileData	Objeto OLE	
	ContentType	Text	50

Figura Especificación de Atributos materiales (G)

materias			
Key	Name	DataType	Length
PK	idmateria	Autonumerico	Entero Largo
	clave_materia	Text	50
	nombre_materia	Text	50

Figura Especificación de Atributos materias (H)

Pais			
Key	Name	DataType	Length
PK	clave	Autonumerico	Entero Largo
	pais	Text	100
	clave_pais	Integer	Entero Largo

Figura Especificación de Atributos país (I)

3.2 Diseño Funcional.

El diseño implementará todos los requisitos contenidos en el modelo de análisis.

3.2.1 Diagramas de Secuencia.

Los diagramas de secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los casos de uso. El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema, modela para cada método de la clase.

Diagrama de Secuencia de Login el proceso se lleva acabo cuando el usuario ingresa el login y la contraseña, se hace una verificación en la base de datos, si se ingresó de manera incorrecta, manda un mensaje de error y le pide que lo vuelva a intentar, pero si lo ingresó de manera correcta lo mandara al menú principal dando la bienvenida (Véase en la figura 3.2.1).

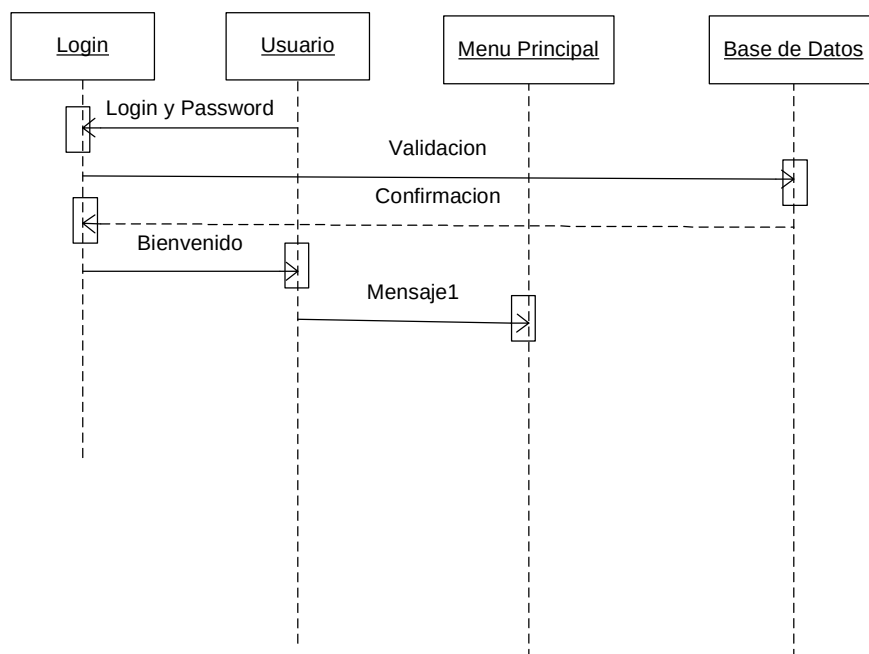


Figura3.2.1 Ingreso de Login

En el siguiente Diagrama se explica cómo trabaja un catalogo del administrador, primero ingresa su login y se valida en la base de datos y le envía la confirmación al usuario enviándolo al menú principal. El usuario selecciona la opción de dar de alta, envía los datos y si lo hizo de manera incorrecta le envía un mensaje de error, pero si lo hizo de manera correcta envía un mensaje de confirmación. Al hacer una modificación, se enviarán los datos a la base, pero si se hace de manera incorrecta, le enviara un mensaje de error, pero si lo hace de manera correcta le mandara un mensaje de confirmación, al seleccionar la opción baja, se selecciona un registro, si llega a tener problemas al eliminar le envía un mensaje de error, pero si se hace de manera correcta le envía un mensaje de confirmación, así trabajan los catálogos (áreas, categorías, país, cuerpo académico y materias), del administrador de manera similar.

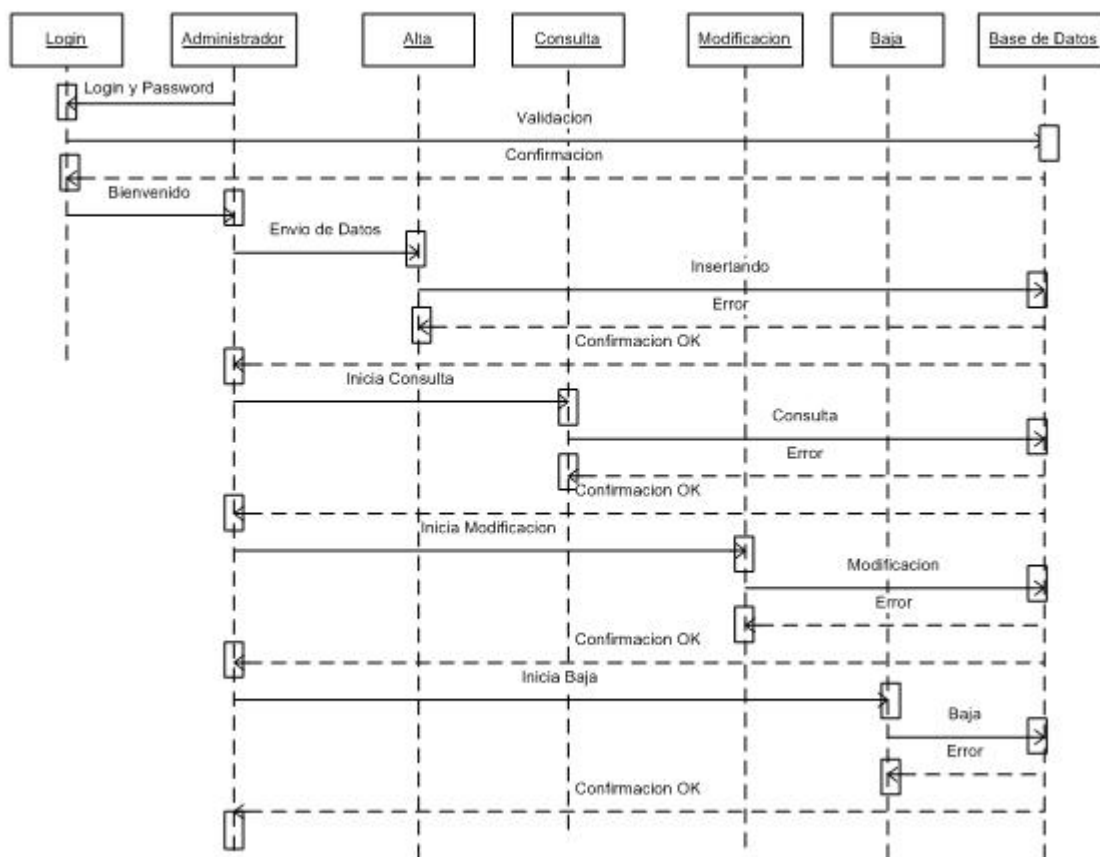


Figura3.2.2 Catálogos de Administrador

En el siguiente diagrama el profesor ingresa su login y si lo hizo de manera incorrecta se le envía un mensaje de error, pero sí lo hizo de manera correcta, se le envía un mensaje de confirmación, e ingresa al menú principal, donde ingresara a las opciones datos personales, datos laborales y horarios. Cabe mencionar que funcionan de la misma que los catálogos del administrador. En la opción de materiales didácticos solo se podrá dar de alta, consultar y eliminar.

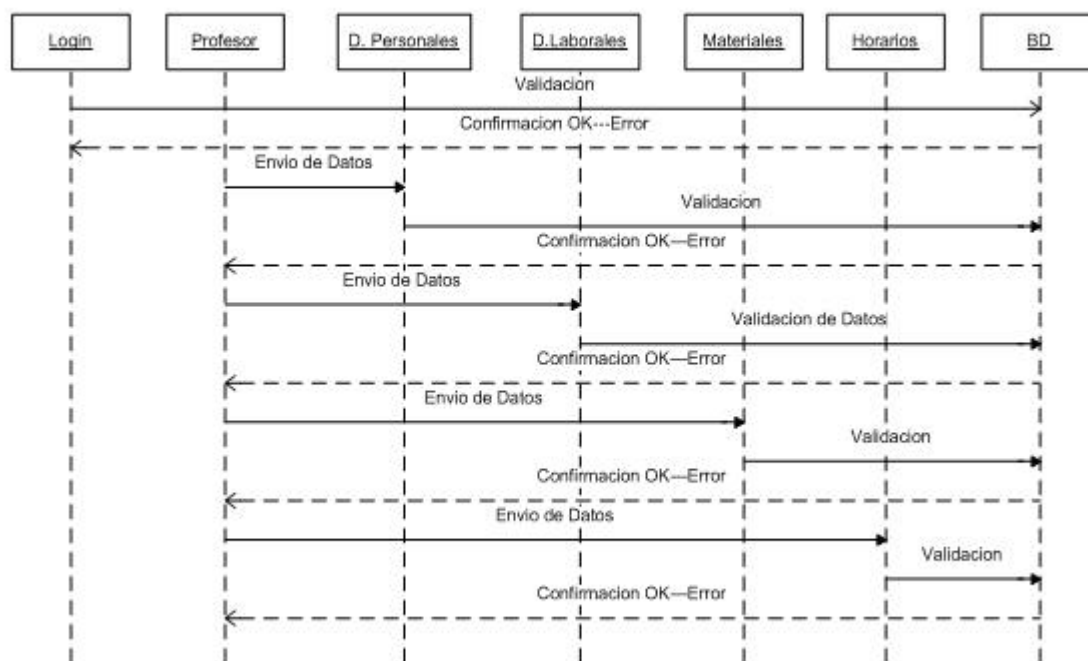


Figura3.2.3 Ingreso al menú de Profesor

3.2.2 Diagramas de Estado.

Un estado es una condición durante la vida de un objeto, de forma que cuando dicha condición se satisface se lleva a cabo alguna acción o se espera por un evento. El estado de un objeto se puede caracterizar por el valor de uno o varios de los atributos de su clase, además, el estado de un objeto también se puede caracterizar por la existencia de un enlace con otro objeto.

En el siguiente diagrama nos permite observar como el usuario ingresa el login e ingresa al menú principal (véase en la figura 3.2.2).

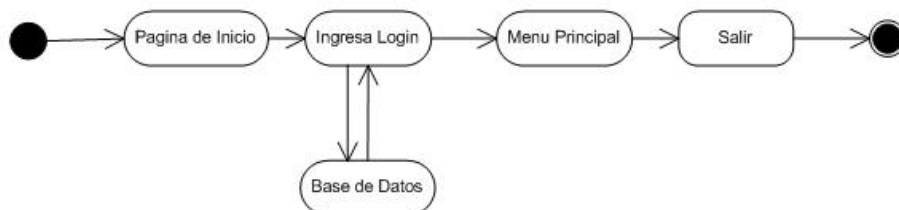


Figura 3.2.2 Ingreso del Login

Ingreso del profesor al menú principal tiene la opción de escoger datos personales, datos laborales, horario o materiales.

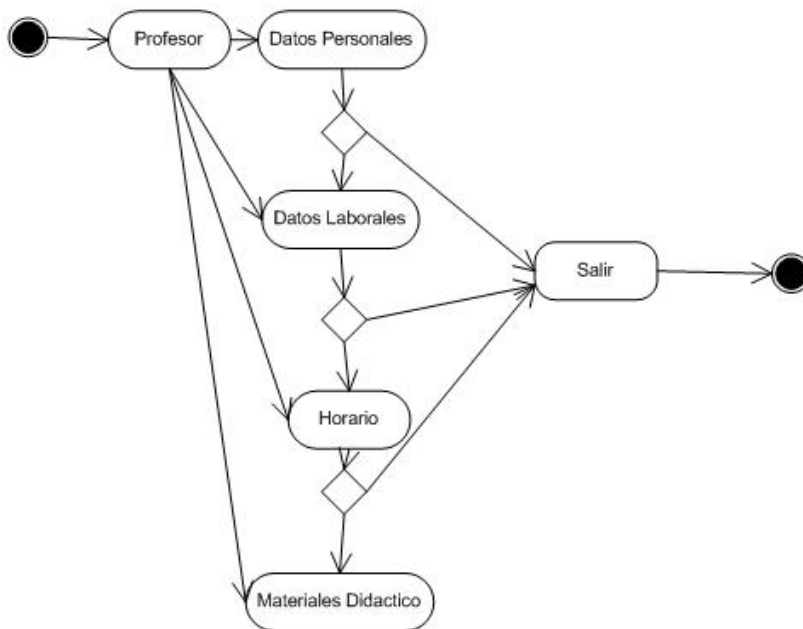


Figura 3.2.3 Ingreso al Menú Principal de Profesores

En el siguiente diagrama se observa que administrador ingresa al menú principal y tiene acceso a los catálogos de país, materias, áreas, categorías o cuerpo académico (véase en la figura 3.2.4).

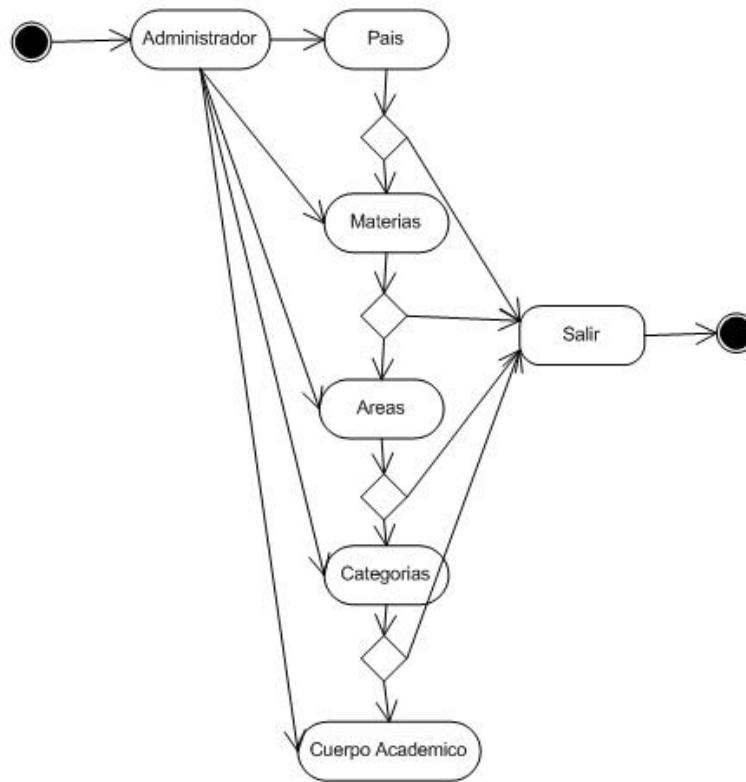


Figura 3.2.4 Ingreso al Menú Principal de Administrador

Capítulo IV Implementación.

4.1 Interfaces.

Se mostrarán a continuación las principales pantallas del sistema Web de Productos Educativos.

Primero mostraremos la página principal del administrador el cual tiene que autenticarse, para tener acceso al sistema (véase en la figura 4.1).



Figura 4.1 Acceso de Administrador

Una vez ingresado, tiene acceso a los catálogos que tendrá a su cargo.

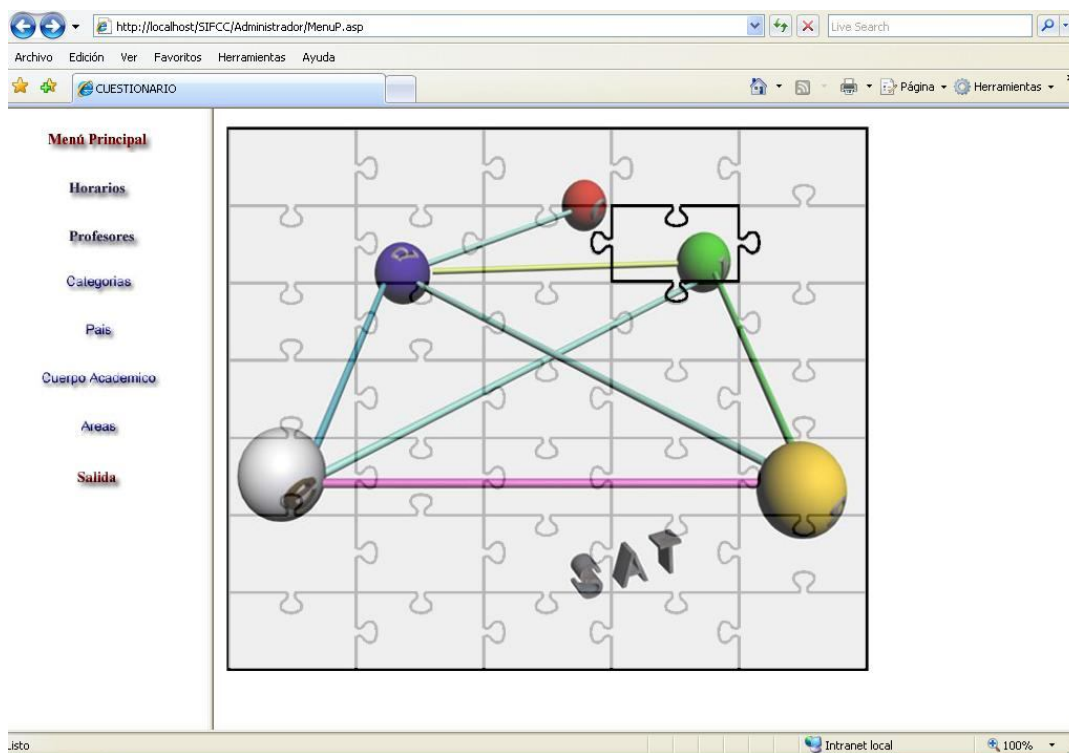


Figura 4.2 Menú Principal del Administrador

En la siguiente pantalla se muestra el catalogo de País, en el cual podrá dar de alta, modificar, consultar y eliminar los países solicitados.

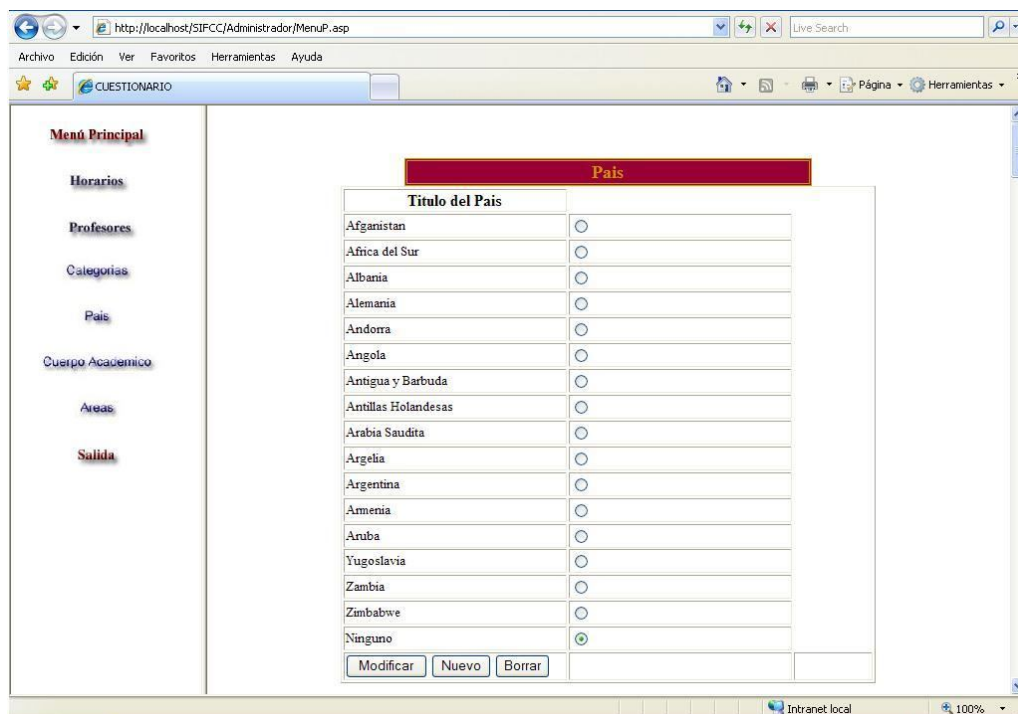


Figura 4.3 Catalogo de País

En la Figura 4.4 se muestra el catalogo de Cuerpo Académico en el cual podrá dar de alta, modificar, consultar y eliminar.

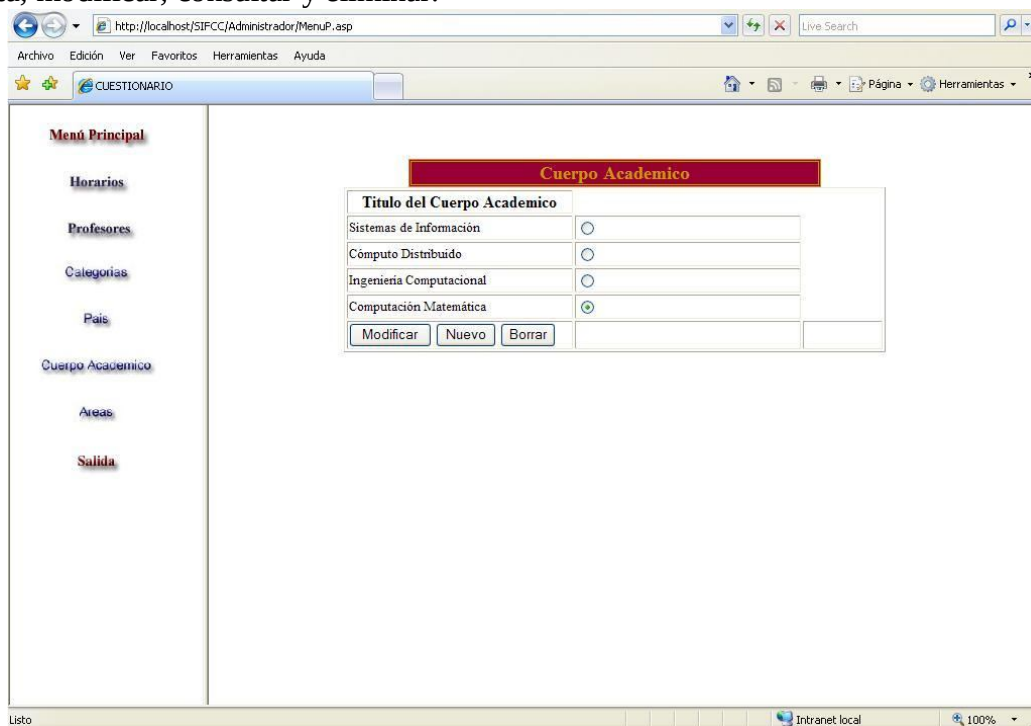


Figura 4.4 Catalogo Cuerpo Académico.

En la siguiente pantalla (véase en la figura 4.5) se muestra el catalogo de Áreas en el cual podrá dar de alta, modificar, consultar y eliminar.

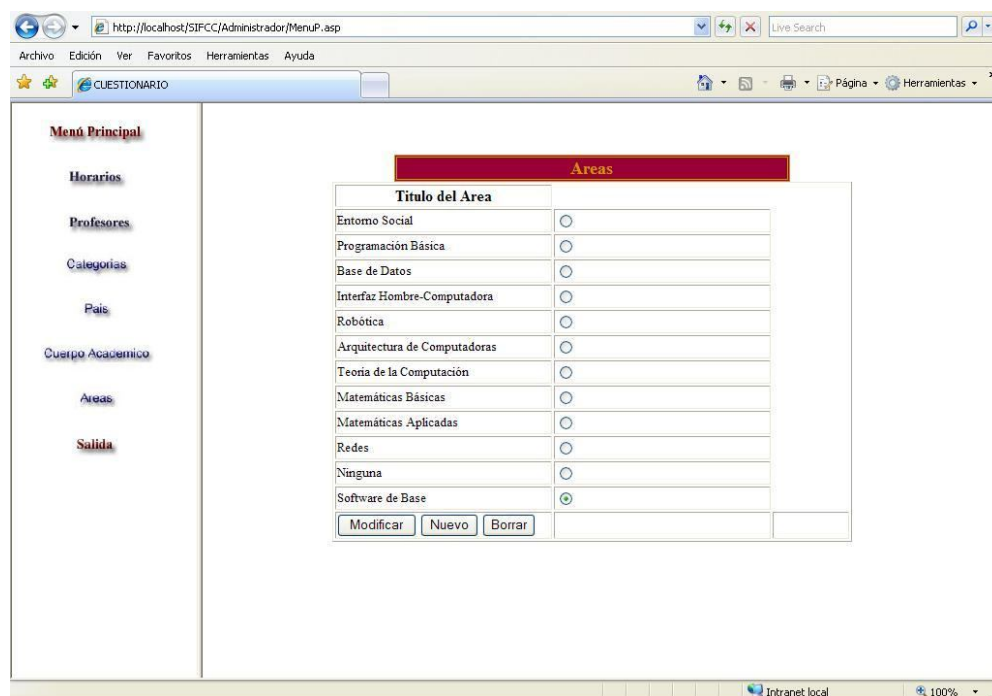


Figura 4.5 Catalogo de Área.

En la siguiente figura se muestra el catálogo de Categorías en el cual podrá dar de alta, modificar, consultar y eliminar.

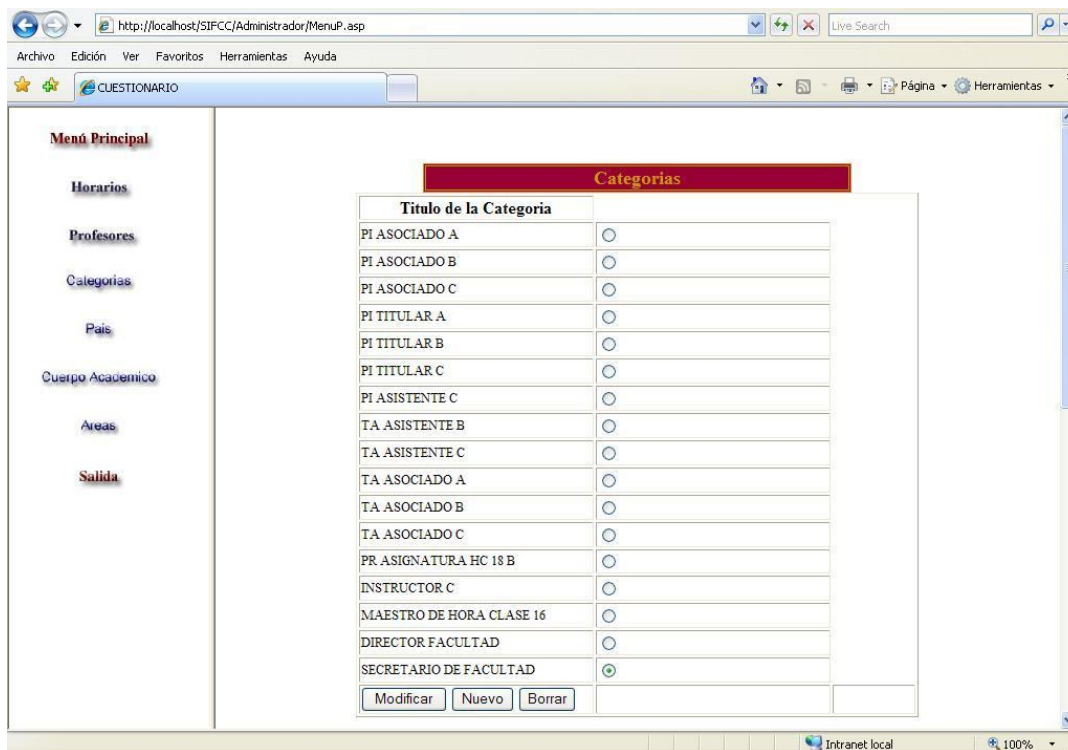


Figura 4.6 Catálogo de Categorías.

La siguiente pantalla es del actor profesor el cual, tendrá que autenticarse, le mostrará una lista de los profesores y seleccionará el que le corresponda e ingresar la contraseña.

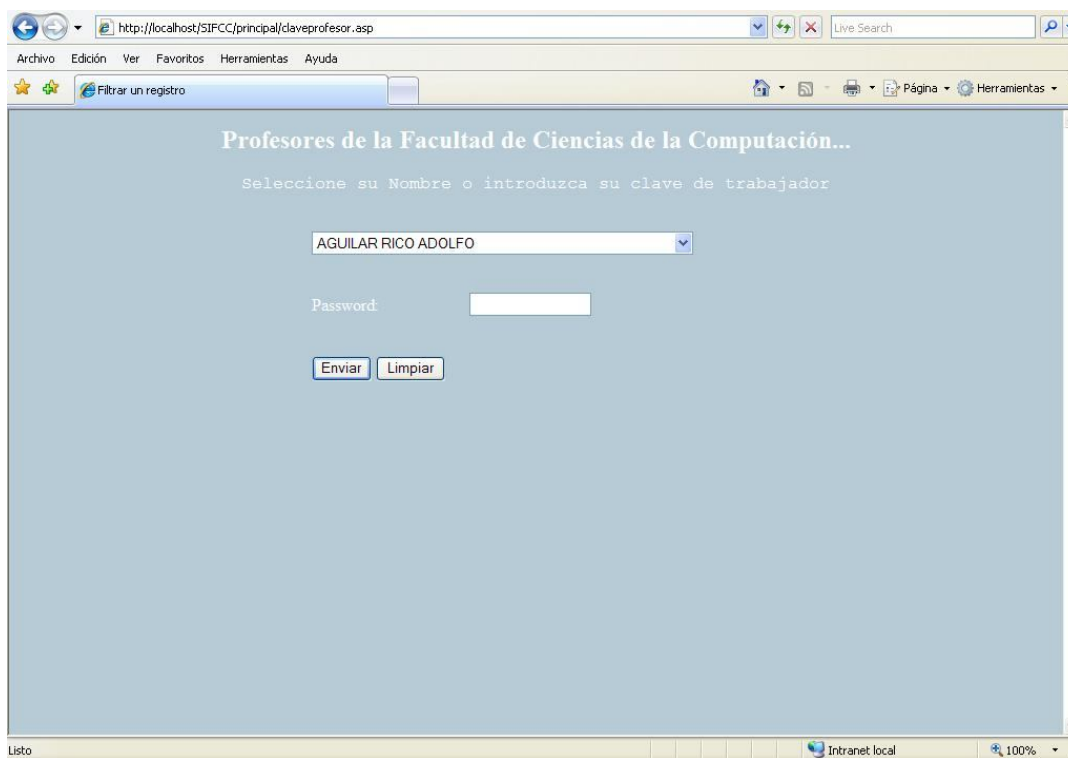


Figura 4.7 Acceso de Profesor.

Una vez autenticado el profesor ingresará a la siguiente pantalla (véase en la figura 4.8) donde se encuentra el menú principal.

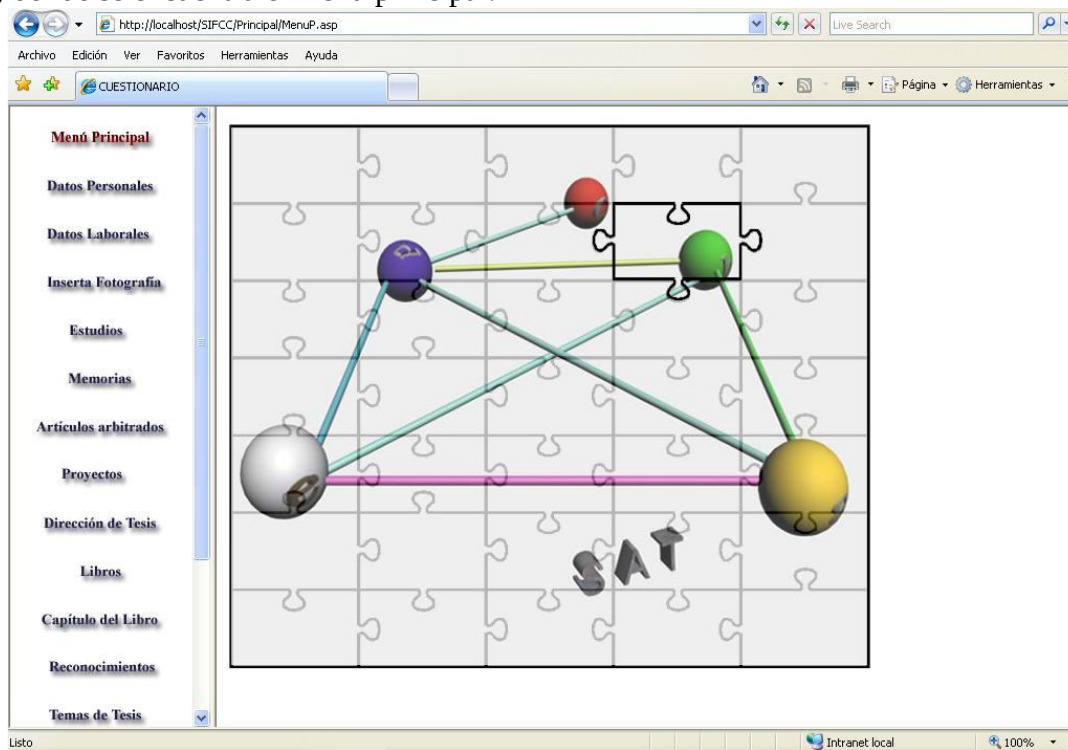


Figura 4.8 Menú Principal Profesor

En la siguiente pantalla el profesor ingresa los datos personales para el control interno del sistema.

Menú Principal

Datos Personales

Datos Laborales

Inserta Fotografía

Estudios

Publicaciones

Proyectos

Reconocimientos

Temas de Tesis

Material Didáctico

Horarios

Impresión Curriculum

Salida

Datos Personales
EUGENIA ERICA VERA CERVANTES

Clave Trabajador: 100203199

Nombre: EUGENIA ERICA

Apellido Paterno: VERA

Apellido Materno: CERVANTES

Último grado de Estudios: Maestría

Siglas de Estudios: M.C.

Institución Otorgante: BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

El último grado de Estudios ¿lo realizó en la BUAP?: SI

Fecha de Titulación: Día: 0 Año: 0 Mes: 0

Fecha de Nacimiento: Día: 0 Año: 0 Mes: 0

Género: Masculino Femenino

Estado Civil: SOLTERO (A)

Tutor: SI

CURP:

Correo Electrónico:

Dirección:

Dirección:

Colonia:

Ciudad:

País: Mexico

Código Postal:

Teléfono de trabajo: 2295500 EXT 7200

Teléfono de casa:

Teléfono celular: 123456

Enviar Borrar Ayuda Imprimir Datos

Figura 4.9 Ingreso de Datos Personales.

En la siguiente pantalla tendrá la opción el profesor de poder subir su foto (véase en la figura 4.10).

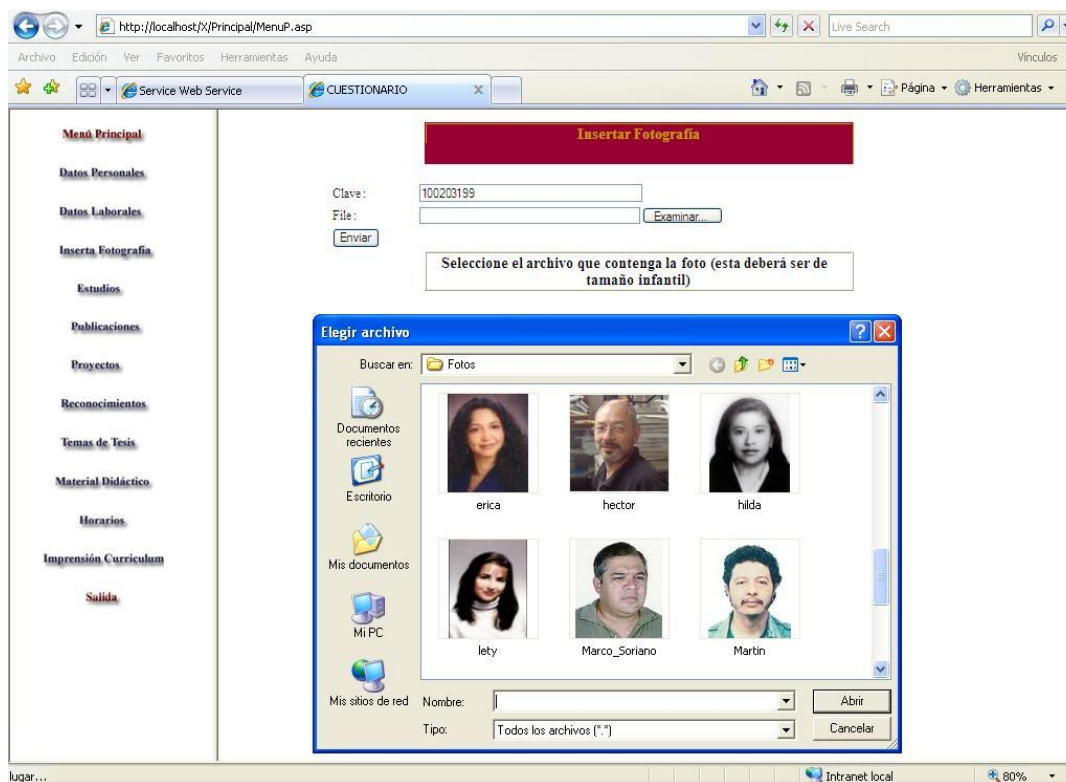


Figura 4.10 Alta de Fotografía.

En la siguiente pantalla el profesor dará de alta los datos laborales.

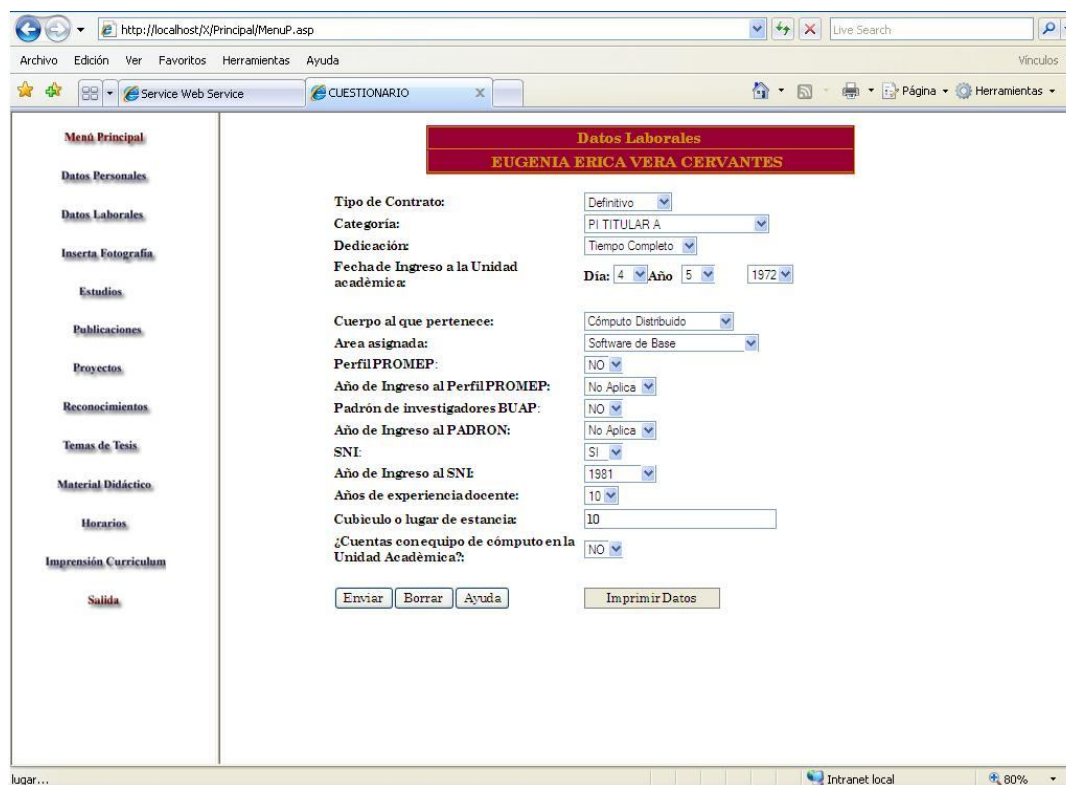


Figura 4.11 Alta de Datos Laborales.

Sistema Web que Administra los Productos Educativos de los Profesores del Área de Programación

En la siguiente figura 4.12 se muestra el catálogo del material didáctico ya ingresados por el profesor.

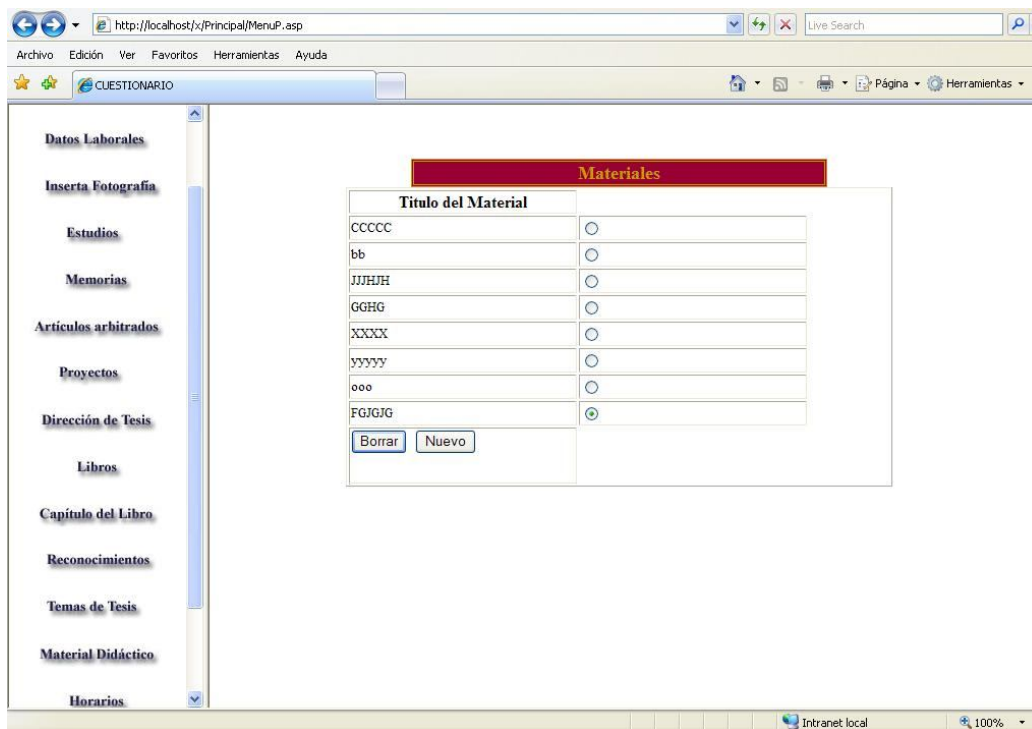


Figura 4.12 Catalogo de Material Didáctico.

Esta pantalla muestra cuando el profesor da de alta un material didáctico (véase en la figura 4.13).

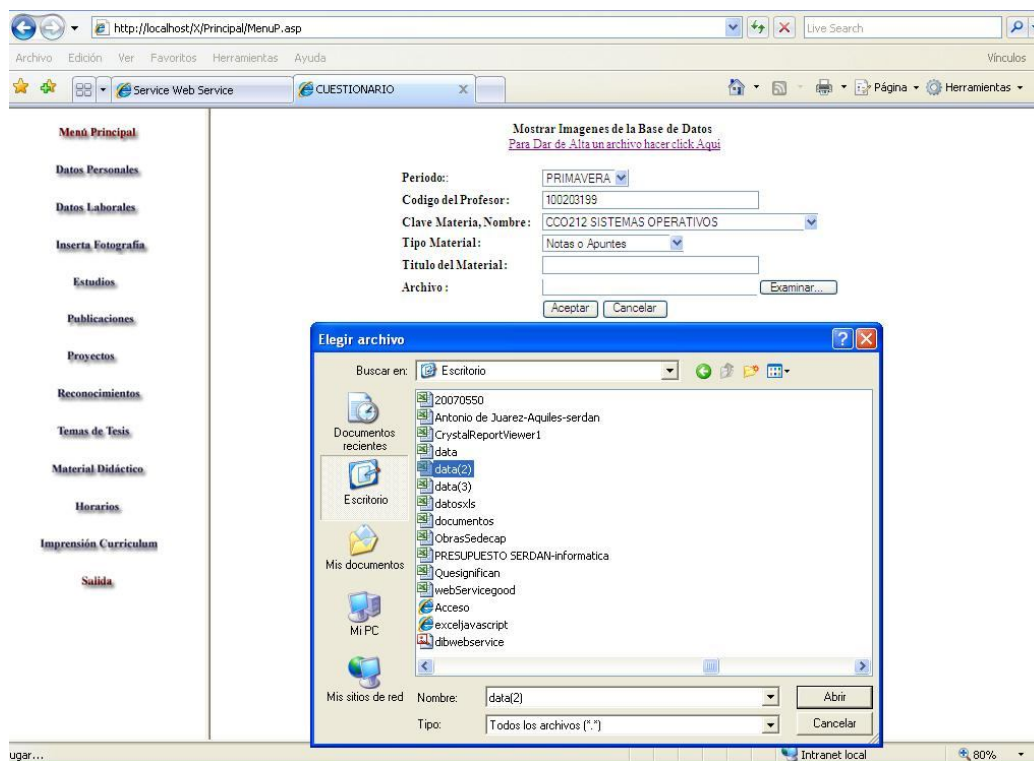


Figura 4.13 Alta de Material Didáctico.

En la figura siguiente (véase la figura 4.14) nos muestra el horario del profesor que ya tiene registrado.

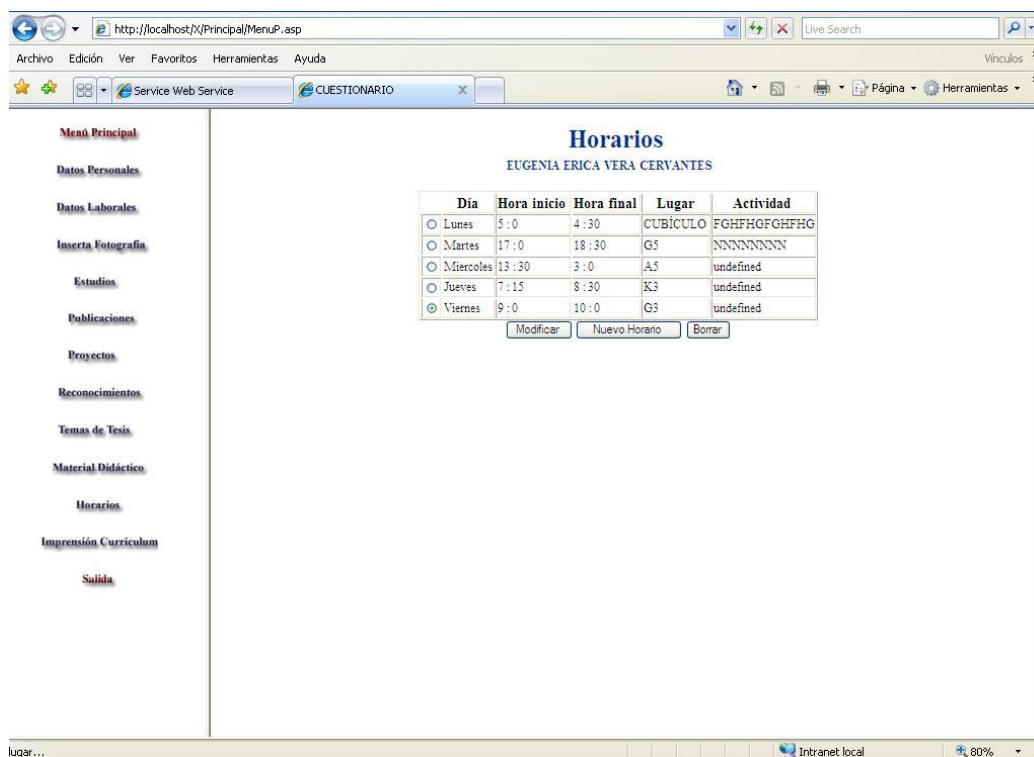


Figura 4.14 Horario del Profesor.

En la siguiente pantalla muestra el alta del horario del profesor (véase en la figura 4.15).

Figura 4.15 Alta de Horario.

En la siguiente pantalla se muestra el menú de reportes del profesor.

Figura 4.16 Menú de Reportes.

En la siguiente Figura se muestra un listado de los archivos descargados y se dice cuantas veces, sirve al profesor esta contabilidad para saber si los alumnos están descargando y si les esta sirviendo el material.

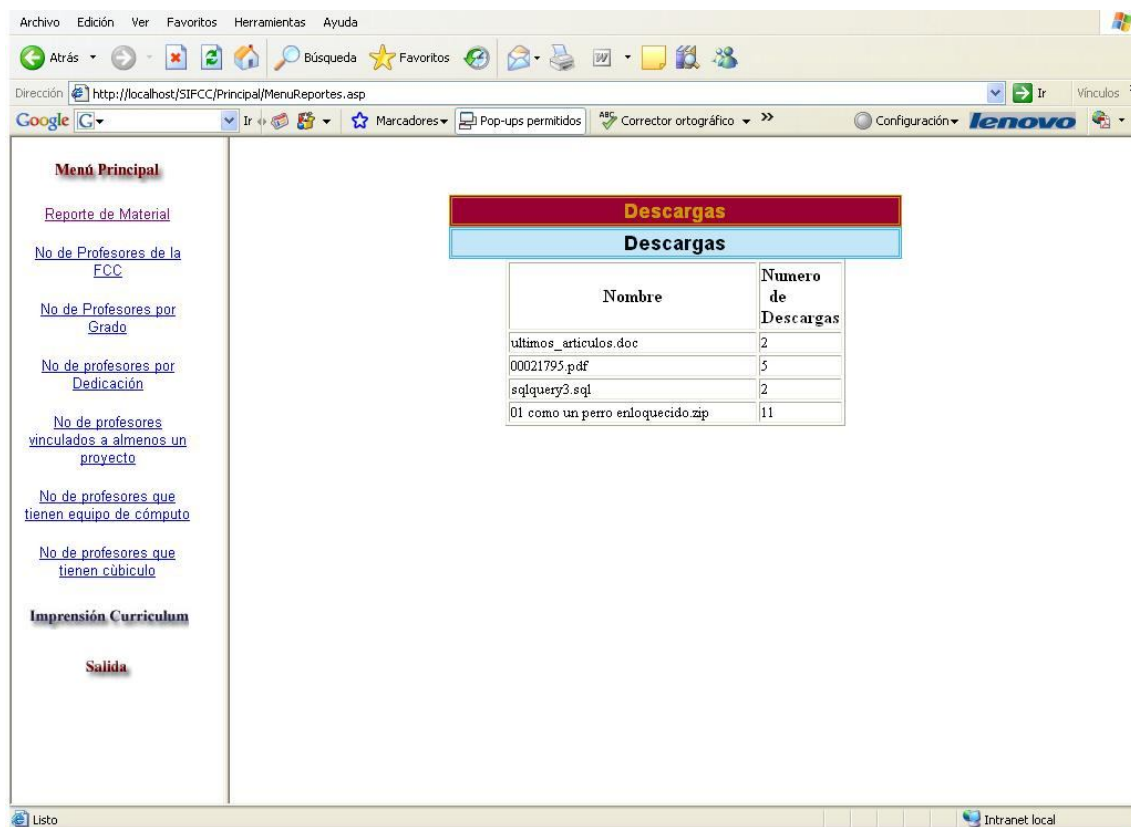


Figura 4.17 Reporte de Descargas del Material.

Conclusiones y Recomendaciones.

En el proyecto se presenta una solución a la administración de los productos educativos, como necesidad primordial de los profesores, los alumnos podrán descargar en línea los productos proporcionados por los profesores ya que es una necesidad para su aprovechamiento.

También se lleva a cabo la administración de los horarios de los profesores, este horario proporciona información acerca de las actividades que el profesor tiene, como dar asesorías, clases, etc.

El sistema se piensa en migrar a la plataforma .NET ya que proporciona ventajas sobre asp como: separar el código del lado de cliente y del lado del servidor, manejo de eventos, archivos de configuración independiente, soporte completo del paradigma de programación orientado a objetos.

También se piensa migrar la base de Access a SQL Server 2005 ya que proporciona ventajas como: manejo de grandes cantidades de información, seguridad, procedimientos almacenados, desencadenadores, etc.

Bibliografía.

1. Ingenieria de software/ Richard Fairley, Antonio Sanchez Aguilar, McGraw-Hill
2. http://pvgdata.com/Mauricio/?p=23&akst_action=share-this Atributos de un buen software
3. Fundamentos de bases de datos / Abraham Silberschaltz, Henry F. Korth, S. Sudarshan, McGraw-Hill
4. Salemi Joe, "Client/Server Computing with Oracle", Ziff-Davis Press, Emeryville, California, 1993.
5. Date C.J., "Introducción a los Sistemas de Bases de Datos", Addison Wesley Iberoamericana, 1993.
6. Elmasri Ramez, "Sistemas de Bases de Datos conceptos fundamentales", Addison Wesley Iberoamericana.
7. Analisis y diseño orientado a objetos con aplicaciones / Grady Booch
8. Aprendiendo UML en 24 Horas/Joseph Schumuller, Pearson Educación.
9. ¿Cómo construir una tienda virtual con Visual Basic y ASP?, Edgar D'Andrea , Infor Book's
10. ASP Active Server Pages 3.0, Jesús Bobadilla, Alejandro Alcocer, Luis Rodríguez – Manzanique, Mc Graw Hill
11. Programación para el Web , Kris Jamás, Suleiman "Sam" Lalani, Steve Weakley
12. Revista Mensual RED varios números
13. Curso programado de ortografía, Gonzalo Godinez de la Barrera, Trillas

14. Creación de aplicaciones Web en Windows NT, Active Server Pages, Jesús Bobadilla, Alejandro Alcocer, Alfaomega rama
15. Fundamentos de Bases de Datos, Abraham Silberschatz, Henry F. Korth, Mc Graw Hill
16. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf
Cliente Servidor