



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



Facultad de Ciencias de la Computación

**Sistema de Administración del Personal de la
Vicerrectoría de Docencia**

TESIS PROFESIONAL PRESENTADA COMO REQUISITO
PARA OBTENER EL TÍTULO DE:

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN.

PRESENTA:

Nicolás Zirahuén Gutiérrez Rangel

ASESOR:

M.C. Yolanda Moyao Martínez

COASESOR:

Lic. José Javier Juárez Caballero

Puebla, Pue. Julio 2010



AGRADECIMIENTOS

A Dios, por su compañía y permitirme cumplir una de mis grandes metas.

A mi asesora de tesis M.C. Yolanda Moyao Martínez por tener la paciencia y apoyarme en la culminación de este proyecto.

A mi coasesor de tesis Lic. José Javier Juárez Caballero, por que sin su ayuda no hubiera podido realizar esta labor, por haber sido un amigo.

A mis amigos, que estuvieron presentes a lo largo de mi carrera, y que me apoyaron.

A todos los profesores, que con su dedicación y esfuerzo me ayudaron a mi formación Profesional.

A todos mis familiares, en especial a mis tíos Martha (Mi “amora”) y Mateo, porque a pesar de estar lejos han estado conmigo en todo momento.

A mis padres Nicolás Gutiérrez Rangel y María A. Rangel Ramírez, por ser para mí un ejemplo a seguir, además de ser las personas mas importantes y trascendentales en mi proyecto de vida y con los que comparto los retos que existen en ella, por apoyarme, comprenderme y tenerme paciencia.

A mis hermanos Yunuén y Mario, por el cariño que me han mostrado en todo momento, pero sobre todo por ser parte de mi vida.

A mi cuñado Uriel, por el apoyo que me ha brindado en los momentos difíciles.

DEDICATORIA

A Todos los integrantes de mi familia, pá, má, Yuni, Mario, Uriel, y en especial a mis enanos “Ari” y “Uri”

*Este trabajo es de ustedes, fruto de sus sacrificios y amor.
Los quiero.*



INDICE:

INIDICE	i
INTRODUCCION	iv
<u>CAPITULO 1</u>	
MARCO TEORICO	1
1.1. Biometría	1
1.1.1. Antecedentes	2
1.1.2. Patrones de reconocimiento.....	4
1.1.3. Alcance de la Biometría	6
1.1.4. Métodos de captura.....	7
1.1.4.1. Multiprocesado de la imagen	8
1.1.4.2. Seguimiento de crestas.....	9
1.1.4.3. Verificación de las huellas.....	10
1.2. Ingeniería de software	12
1.2.1. ¿Qué es un sistema?	12
1.2.2. Características del software.....	13
1.2.3. Proceso del software	13
1.2.4. Modelo de proceso del software	14
1.3. Bases de datos	16
1.3.1. Antecedentes	16
1.3.2. Modelo Entidad-Relación	17
1.3.3. Modelo Relacional	18
1.3.4. Bases de datos relacionales	19
1.3.5. Normalización	20
1.3.6. El lenguaje de consulta	21
1.4. Herramientas de desarrollo y diseño	22
1.4.1. Microsoft SQL server.....	22
1.4.2. AJAX.....	24
1.4.3. ASP.NET	26
<u>CAPITULO 2</u>	
ANALISIS Y DISEÑO DEL SISTEMA.....	27
2.1 Planteamiento del problema	27
2.1.1. Especificación de requerimientos.....	28
2.1.1.1. Definición conceptual del sistema.....	29



INDICE GENERAL



2.1.2. Modelo conceptual del sistema	30
2.1.2.1. Diagrama de flujo de datos	31
2.1.3. Casos de uso.....	35
2.1.3.1. Identificación de actores	35
2.1.3.2. Diagrama de casos de uso del sistema	37
2.1.3.2.1. Flujo de eventos del sistema	37
2.2. Diseño del sistema	44
2.2.1. Diseño conceptual	44
2.2.1.1. Entidades	44
2.2.1.2. Relaciones.....	45
2.2.1.3. Atributos.....	45
2.2.1.4. Diagrama Entidad-Relación	45
2.2.2. Diseño lógico	46
2.2.2.1. Modelo Relacional	46
2.2.2.2. Normalización de la base de dato	47
2.2.3. Diseño físico	49
2.2.3.1. Diccionario de dato	50

CAPITULO 3

IMPLEMENTACION DEL SISTEMA..... 53

3.1 Implementación del sistema	53
3.2. Implementación de la interfaz	58

CAPITULO 4

PRUEBAS DEL SISTEMA 63

4.1 Casos de prueba.....	63
4.1.1. Interfaz principal	66
4.1.2. Interfaz menú principal	67
4.1.3. Interfaz menú administrar	68
4.1.3.1. Agregar	68
4.1.3.2. Editar	72
4.1.3.3. Eliminar	77
4.1.4. Interfaz consultar reportes	79
4.1.5. Interfaz Consultar historial.....	83
4.1.6. Registro de asistencia	84



CONCLUSIONES, LIMITACIONES Y PERSPECTIVAS

CONCLUSIONES	88
LIMITACIONES	89
PERSPECTIVAS	89
BIBLIOGRAFIA.....	91



Introducción

A partir del momento en que el hombre desarrolla la necesidad de efectuar una actividad de manera colectiva y organizada para obtener un beneficio en común, se abre paso a la creación de organizaciones y empresas, cuya finalidad consiste en combinar elementos o recursos fundamentales, como por ejemplo, el ser humano, implementos técnicos, materias primas (factores productivos), para hacer real una idea de negocio, que tenga la finalidad de obtener beneficios, generar utilidades, aportar soluciones y contribuir al bien común.

Para que esta actividad se pueda desarrollar de una manera un poco más organizada es necesario contar con tecnología que permita administrar los factores productivos de la empresa y como se combinan entre sí, se trate de bienes o de los mismos empleados de que laboran en la institución.

Anteriormente la administración se hacía mediante sistemas de ficheros que más tarde fueron evolucionando a los sistemas de bases de datos.

Los sistemas de bases de datos nos permiten agilizar el proceso de administrar personal, dependiendo de los aspectos que se desean cuidar, ya que es posible desarrollarlos de acuerdo a las necesidades que sean requeridas, tales como, llevar un registro de sus trabajadores, identificar los tipos de empleados con los que se cuenta, si generan horas extras u otro tipo de retribuciones para la empresa, reportes de pago, solo por citar algunos.

Estos sistemas son efectivos, pero no del todo ya que en realidad no se sabe si en realidad el trabajador es el que checa en su horario establecido o alguien hace su tarea por él, es por lo que se ha pensado en actualizar este tipo de sistemas en uno que permita realizar una lectura biométrica de cada uno de los empleados para así corroborar que en realidad se trata de la persona correcta.

Los sistemas biométricos son métodos basados en la inteligencia artificial y el reconocimiento de patrones, que permite la identificación y/o verificación de la identidad de personas a partir de características morfológicas o de comportamiento, propias y únicas del individuo, conocidas como autenticadores.

Algunas de las ventajas que se obtienen con un sistema biométrico son:

- Control sobre el tiempo extra que representa un alto costo en las empresas



INTRODUCCION



- Obliga el cumplimiento de la jornada de trabajo del personal
- Desalienta artimañas en el registro de checado de los trabajadores
- Mejora desempeño y productividad del personal de Nóminas
- Introduce el concepto de innovación tecnológica a los trabajadores
- Realza la imagen de empresa moderna y actualizada

En nuestra universidad existe una dependencia llamada Vicerrectoría de docencia, la cual necesita llevar el control de las asistencias del personal que labora en ella.

Actualmente el registro de asistencia del personal se realiza a través del llenado de hojas, las cuales se van cambiando día a día, y se tiene que realizar el conteo de las horas manualmente, es decir, aun se tiene que firmar la hora de entrada y salida lo cual es muy tedioso e inconsistente.

El registro de las horas de los estudiantes de servicio social, práctica profesional y personal que labora por horas, es contabilizado por cada área de manera diferente, además de que se requiere llevar un historial (bitácora) de los estudiantes que han apoyado los distintos procesos, para su seguimiento en el aspecto profesional. Lo cual no se tiene en este momento.

A consecuencia de lo anterior es que surge el proyecto que se documenta en el presente trabajo de tesis.

Objetivo general

El objetivo general de nuestro proyecto es dar una solución tecnológica al problema del registro de asistencias del personal de la Vicerrectoría de Docencia y de los estudiantes que prestan servicios en la misma, para llevar un mejor control y evitar inconsistencias a la hora del checado de asistencia, además de estandarizar el conteo de horas laboradas para todas las áreas de la dependencia, todo esto mediante la implementación de un sistema administrativo capaz de obtener el registro de asistencia por medio de un lector biométrico.

Objetivos específicos

- Documentar y experimentar el desarrollo de un sistema funcional aplicando los conocimientos adquiridos en el transcurso de la formación profesional.



INTRODUCCION



- Analizar la problemática del almacenamiento de la información y proponer una solución.
- Aplicar el Proceso de desarrollo de Software para la construcción del sistema.
- Diseñar la base de datos siguiendo el Modelo Relacional.
- Implementar la base de datos en SQL server 2005.
- Implementar el sistema utilizando el lenguaje de programación Visual.net.
- Realizar las pruebas del Sistema.

Inicialmente, presentamos el marco teórico sobre el cual se fundamentará el sistema; en este se mencionan conceptos fundamentales acerca de lo que es la biometría, la ingeniería de software, el modelo relacional, las bases de datos así como las herramientas necesarias para el desarrollo del sistema.

Posteriormente, se establece el análisis y diseño del sistema, es decir el planteamiento del problema, la especificación de los requerimientos, la elaboración de casos de uso, el diagrama (Entidad-Relación), el Modelo Relacional, el Diseño Físico de la Base de Datos, la Normalización de la Base de datos y diseño de la interfaz del sistema.

Enseguida, se documenta la implementación del sistema, la cual consiste en la creación de nuestra base de datos, así como también las interfaces principales con las que el usuario tendrá interacción.

Finalmente, realizaremos pruebas con valores y datos reales de manera que se pueda observar de manera clara y detallada que el sistema cumple con los objetivos planteados al inicio.



1.-Marco teórico

1.1.- Biometría

El concepto biometría proviene de las palabras bios (vida) y metros (medida), por lo tanto con ello se infiere que todo equipo biométrico mide e identifica alguna característica propia de la persona. Biometría es "una característica única física medible y como tal, diferenciable".

La biometría es una técnica que consiste en identificar o verificar automáticamente la identidad de las personas, basándose en sus características físicas o sus pautas de comportamiento. Por ejemplo, el reconocimiento del iris, las características de la mano, las huellas dactilares o la voz. [1]

Los métodos tradicionales de autenticación por contraseña o tarjetas inteligentes trabajan en base a lo que se conoce o posee. Por lo tanto no identifican completamente a un usuario.

Los sistemas biométricos, en cambio, operan en base a características físicas o del comportamiento de cada usuario, es decir, parámetros prácticamente inseparables del individuo. [2]

En realidad, si entendemos este concepto en términos muy amplios, podemos decir que la biometría se practica desde el principio de los tiempos y, de hecho, nosotros mismos la practicamos muchas veces a lo largo del día sin casi darnos cuenta. Por ejemplo, cuando descolgamos el teléfono y escuchamos la voz de la persona que está llamando, nuestro cerebro trata de comprobar si esa voz se parece a cualquiera de las muestras que tiene almacenadas en su memoria y que ha ido recopilando a lo largo de nuestra vida. Si nuestro cerebro encuentra similitudes suficientes entre alguno de sus recuerdos y lo que está escuchando en ese momento, entonces reconocemos a la persona que nos ha llamado. Si no, asumimos que estamos ante alguien a quien no conocemos.

El reto de la tecnología biométrica radica en conseguir que las máquinas puedan llevar a cabo esas operaciones de reconocimiento que son, en apariencia, tan simples y que lo hagan de forma rápida y libre de errores. Igual que ocurre con otras capacidades humanas, tales como el lenguaje, dotar a las máquinas de la capacidad de llevar a cabo la identificación con efectividad, se ha revelado como una tarea muy compleja.[3]

Los sistemas biométricos incluyen un dispositivo receptor y un software que interpreta la muestra física y la transforma en una secuencia numérica única.



En el caso del reconocimiento de la huella digital, se ha de tener en cuenta que en ningún caso se extrae la imagen de la huella, sino una secuencia de números que la representan, esto además sirve para que nadie que opere con estos sistemas guarde un registro fotográfico de las huellas digitales con la posibilidad de reproducirlas para otros usos. Sus aplicaciones abarcan un gran número de sectores: desde el acceso seguro a la PC y redes, protección de ficheros electrónicos, hasta el control horario y control de acceso físico a una sala de acceso restringido. [4]

1.1.1.- Antecedentes

Los sistemas biométricos tienen como principal ventaja, una mayor seguridad y comodidad que los sistemas tradicionales basados en los passwords o tarjetas. El acceso a través de la biometría no depende de algo que sepamos o que tengamos y que nos puedan robar o copiar; depende de lo que somos, es decir nuestras características físicas.

Es por este motivo por el que la tecnología biométrica es el sistema de seguridad más fiable en la actualidad. Esta secuencia numérica, llamada patrón de registro, queda almacenada en una base de datos segura y servirá para las siguientes comparaciones cada vez que la persona autorizada desee acceder al sistema.

-Debido a que las huellas digitales son un rasgo distintivo entre los seres humanos, estas han sido utilizadas como medio de identificación.

Según B.C. Bridgest, especialista en la materia, las huellas digitales comenzaron a usarse en las antiguas civilizaciones. En 1953 los jóvenes científicos ingleses Francis Crick y James Watson, entraron al bar "The Eagle" anunciando que habían encontrado "el secreto de la vida". Su hallazgo: el ADN, información que hace posible identificar de forma exacta a los seres vivos y ha permitido los más grandes avances médicos de la historia.

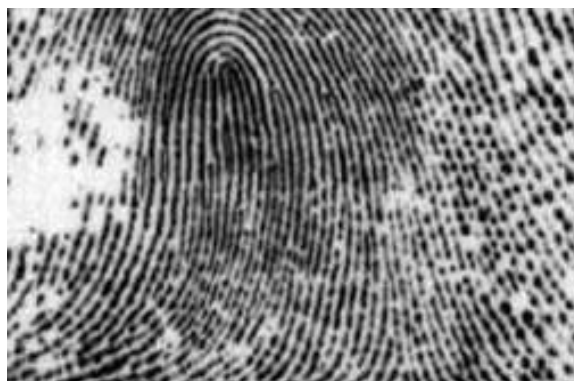


Figura 1.1.-Ejemplo de huella digital



Sin embargo, mucho antes de ese descubrimiento, durante siglos la huella digital fue la prueba más irrefutable para distinguir con exactitud la identidad de una persona, pues se trata de una característica física única en cada ser humano.

Formada por un patrón de crestas y valles, este es el sistema de reconocimiento biométrico más antiguo. Pese a ello, continúa implementándose en múltiples áreas que van desde la criminología hasta los documentos de identidad y ya comienza a integrarse a los sistemas de tecnología más avanzados, donde incluso se están desarrollando plataformas que permiten utilizar la huella digital electrónica para realizar complejos trámites a través de Internet.

Las huellas dactilares son una característica física que sólo se presenta en los primates, en los humanos se forman a partir de la sexta semana de vida intrauterina y sus características no varían con la edad de la persona. Además de crestas papilares y valles interpapilares, las rugosidades con formas arbitrarias que adopta la piel que cubre la yema de los dedos posee minucias, es decir, discontinuidades locales de las crestas.

Dado que el sudor se deposita en los surcos, al tocar alguna superficie, la marca única de cada persona queda impresa en el objeto, por lo que es posible obtener una copia del negativo de una huella digital, superponiendo un polvo blanco de consistencia similar al talco, pues éste absorberá la grasa.

Si bien en criminología se ha utilizado la huella digital como prueba de la presencia de una persona en un determinado lugar durante el último siglo, su aplicación identificadora en otras áreas viene de mucho antes.

Ya en la antigua Babilonia, muchas de las transacciones comerciales quedaban grabadas en tablas de arcilla que se firmaban con la huella digital, práctica que también existía en Persia durante el Siglo XIV, época en la que oficiales del gobierno notaron que no había dos huellas dactilares iguales. Asimismo, la antigua legislación china establecía que para divorciarse era necesario un documento que expusiera siete motivos y fuera firmado con la impresión digital.

En la cultura occidental en tanto, las huellas dactilares comenzaron a ser utilizadas para validar documentos y contratos a partir del Siglo XIX, época en la que se registran las primeras investigaciones sistemáticas sobre el tema en Europa, lo que permitió masificar el uso de la identificación dactiloscópica en el campo civil y criminal.



En 1880 el cirujano inglés Henry Faulds, presentó un estudio sobre los surcos de la piel, inspirado en la presencia de marcas de dedos en piezas de cerámica prehistórica. En su trabajo, el médico propuso la importancia de huellas digitales como una forma para identificar personas, estableciendo un método de clasificación. [5]

Actualmente en la mayoría de los países, la implementación de dispositivos biométricos se lleva a cabo para usos de identificación y seguridad, por instituciones bancarias, gobiernos, empresas y hasta compañías fabricantes de computadoras.

La investigación en la biometría tiene como propósito encontrar nuevos métodos de identificación de patrones que permitan el mejoramiento de las técnicas utilizadas actualmente para la identificación de las características fisiológicas y de comportamiento de las personas. El mejoramiento de estos métodos permitirá resolver problemas que existen actualmente con la identificación de patrones.

Siendo la anterior la meta del área de investigación, al mejorar estas técnicas la tecnología que implementa la biometría permitirá que existan dispositivos que mejoren los procesos de identificación para factores de seguridad en sistemas bancarios, de gobierno entre otros.

1.1.2.- Patrones de reconocimiento

La identificación por huella dactilar está basada principalmente en las *minucias*, o la ubicación y dirección de los finales y bifurcaciones (separaciones) de las crestas a lo largo su trayectoria.

Las figuras 1.2 y 1.3 presentan ejemplos de características de huellas dactilares: dos tipos de minucias y ejemplos de otras características algunas veces utilizadas durante la clasificación automática y procesos de extracción de minucias:

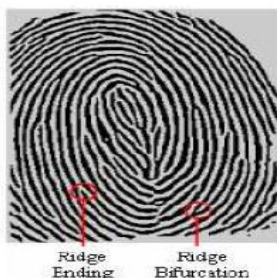


Figura1. 2.- Minucias



Figura 1.3.-Otras características



La huella digital aparece generalmente constituida por una serie de líneas oscuras que representan las crestas y una serie de espacios blancos que representan los valles, aunque existen muchas otras características de huellas digitales. Cada uno de estos puntos tiene una característica y una posición única, que puede ser medida. Comparando esta distribución es posible obtener la identidad de una persona que intenta acceder a un sistema en general, esta tecnología es conocida como AFIS (Sistema Automático de Identificación Biométrica).

Otra forma de distinguir las huellas digitales es por sus patrones, los cuales presentó Johannes Evangelista Purkinje en su tesis doctoral.

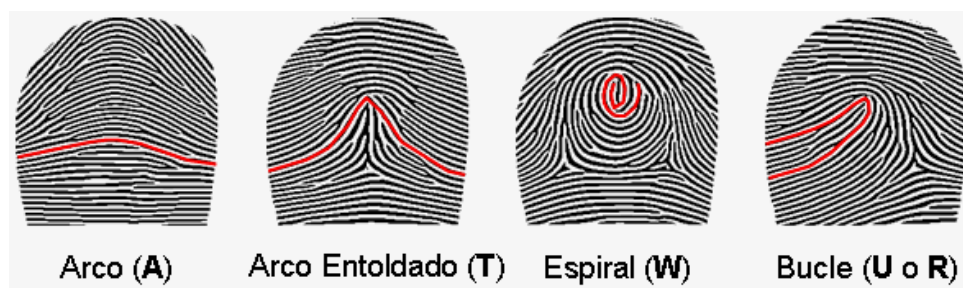


Figura 1.4.-Cuatro patrones principales

De manera general la forma de procesar una huella digital se muestra en la figura 1.4.

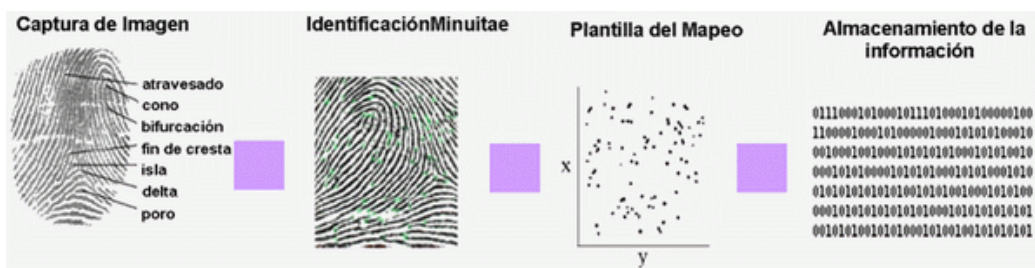


Figura 1.5.- Proceso común de escaneo de la huella digital

La tecnología AFIS explota algunas de estas características de huellas dactilares. Las crestas de fricción no siempre fluyen continuamente a través de un patrón y a menudo resultan en características específicas como finales de crestas, crestas divisorias y puntos, u otra información. [6]



1.1.3.- Alcance de la biometría

La biometría es un área de investigación que todavía tiene mucho campo para crecer. La madurez de esta tecnología se encuentra ligada a la investigación de nuevas técnicas para su mejoramiento. El uso de la biometría en un futuro será aun mayor al actual. Cada vez la tecnología de la seguridad y de la identificación de las personas por medio de tarjetas y claves es más vulnerable.

En la seguridad, la biometría juega y jugará un rol muy importante. Esto es porque cada vez se requiere que las contraseñas usadas para la identificación de una persona en un sistema sean más complejas; entre más complejo, más difícil de recordar. La tecnología biométrica permitirá que las contraseñas sean algo del pasado, puesto que las características que evalúa son individuales, únicas y no replicables.

Las herramientas biométricas son sistemas automáticos de chequeo de patrones, que en pocos segundos pueden obtener una muestra del individuo, información que comparan con una base de datos para determinar si los documentos concuerdan o no a la identidad de una persona. [7]

A diferencia de otros sistemas de seguridad, los datos biométricos no se pueden robar o perder, porque forman parte del individuo, por lo que nadie los puede memorizar, ni tampoco es necesario recordarlos, como sucede con las claves y contraseñas.

Con todo esto, se estima que la tecnología biométrica comenzará a imponerse como una exigencia a escala mundial en las próximas décadas, siendo masificado su uso especialmente en los lugares de trabajo.

Sin embargo, dadas las respectivas realidades locales de los países en vías de desarrollo, es muy probable que lo primero que se implemente sean los elementos detectores de la biometría de las huellas digitales, puesto que dichos lectores son más fáciles de adquirir por su bajo costo.

Esta tecnología permite detectar modelos y formas presentes en la superficie de las yemas de los dedos y verificar la identidad de la persona a través de diversos enfoques: hay sistemas que funcionan como el proceso policial de coincidencia de huellas; otros utilizan dispositivos para describir los modelos y rasgos esquemáticos de las huellas, mientras que algunos manejan modelos



tridimensionales basados en ultrasonido, que permiten incluso detectar si un dedo está vivo o muerto.

Pero la tecnología biométrica digital también ha ingresado con fuerza en los últimos años en otros ámbitos, permitiendo la compra de bonos para atención médica, el control de acceso a algunas empresas e incluso ha comenzado a utilizarse en algunas casas comerciales, cuyas máquinas ofrecen el pago de cuentas en dinero efectivo, sin necesidad de utilizar sobres o formularios.

Mientras que los avances científicos con sus aspectos interactivos y digitales afectan nuestra vida en los ámbitos más diversos, paulatinamente los sistemas de seguridad están evolucionándolos hacia un área que es la más simple y, a la vez, la más compleja de todas: buscan reconocer las marcas distintivas de nuestro cuerpo, las mismas que en la antigüedad se imprimían en la arcilla. [5]

1.1.4.- Métodos De Captura

Cuando se quiere desarrollar un sistema automático de identificación de personas mediante huella dactilar, es preciso disponer de algún mecanismo para capturar una imagen de la huella del usuario de forma automática, fiable y sencilla. Este ha sido uno de los puntos donde la biometría ha encontrado, durante años, muchos inconvenientes. Éstos han venido tanto por el coste, como por el tamaño de los métodos ópticos de captura de la imagen, los cuales, además, requerían un excesivo mantenimiento.

Debido a que el resto de las partes del sistema de identificación por huella extracción de características y verificación), estaban ofreciendo resultados más que satisfactorios, la industria del sector impulsó investigaciones que conllevaran a nuevos métodos de captura.

La mayoría de las técnicas biométricas, llevan mucho tiempo en investigación, por lo tanto el número y variedad de algoritmos matemáticos para realizar cualquiera de los procesos involucrados en el sistema, es enorme. En este caso se van a citar dos de los que más éxito tiene dentro del mundo científico y de investigación.

Pero ambos procesos requieren de un pre procesado previo de la imagen de la huella dactilar. Este pre procesado, común para los dos algoritmos, depende en gran medida del resultado de la huella capturada por el dispositivo concreto, y



del grado de tolerancia al movimiento que se le quiera dar al usuario final. En concreto, este pre-procesado está compuesto de los siguientes pasos:

1. Localización y segmentación de la huella: en la imagen capturada, no sólo habrá huella, sino que también pueden haber partes de la imagen que no contengan información sobre la misma (ya sea porque el área designada por el dispositivo sea mayor que el tamaño del dedo, o por la incorrecta colocación del dedo en el sensor. Se analizará dónde se encuentra la huella localización), y se tomará de ella aquella área que sea fuente de estudio, eliminándose todo lo demás (segmentación). Si la zona segmentada no cumple unas determinadas características (como por ejemplo, tamaño inferior a uno dado), el sistema puede rechazar dicha huella, indicándoselo al usuario, y evitando de esta forma realizar todo el cálculo restante.

2. Enfatización de la huella: se hace una ecualización de los niveles de grises de la huella, de forma que se pueda aprovechar mucho mejor, la diferencia de tonalidades entre unos puntos y otros, y además, poder equiparar las zonas donde se ha realizado menor presión del dedo, con las zonas de más presión.

3. Cálculo del mapa de orientación: con la imagen resultante de los dos pasos anteriores, se hace un estudio de gradiente a baja escala, para poder obtener, de una forma aproximada, la orientación que tienen las crestas, en cada zona de la imagen. Para ello se hace el estudio en subimágenes, tomando porciones disjuntas de la imagen anteriormente obtenida. Esta información servirá de ayuda para los algoritmos de extracción de características. En algunos sistemas, este paso se hace al principio, para ser utilizado en la segmentación, y luego se vuelve a calcular con imagen obtenida tras la ecualización.

1.1.4.1.- Multiprocesado de la Imagen

En este algoritmo, se obtienen las minucias después de todo un procesado sucesivo de la imagen resultante del pre-procesado. Este proceso se divide en los siguientes pasos:

1. Detección de las crestas: una vez que se ha localizado y enfatizado la huella, se intentan localizar las crestas, para separarlas de los valles. En concreto se trata de un proceso de binarización, es decir, de reducir la imagen de 256 niveles de grises, a una imagen de sólo blanco y negro. Los métodos para poder conseguir esta binarización son muchos, desde el uso de umbrales globales o adaptativos,



hasta la utilización de la propiedad de que una cresta es aquel punto que obtiene un mínimo local en la dirección perpendicular a su campo de orientación.

2. Esqueletización: dependiendo del método de detección de crestas utilizado, será necesario realizar un “adelgazamiento” de los tramos que representan las crestas, es decir, hacer que estas crestas estén representadas por líneas de un único píxel de grosor. A este proceso se le denomina esqueletización, y se suele realizar fijándose en cada uno de los píxeles de la imagen, y dependiendo del número de píxeles puestos a 0 (si el 0 representa cresta) que le rodean (es decir, sus vecinos), eliminar parte de ellos. Este estudio se hace una y otra vez, hasta alcanzar el esqueleto de la huella.

3. Detección de minucias: con la imagen esqueletizada, se realiza un proceso de búsqueda de las “potenciales” minucias dentro de la imagen. Dicho proceso está basado en localizar aquellos píxeles puestos a 0 en la imagen, que tienen un solo vecino, o más de dos, siendo el primer caso el de una terminación, y el segundo, el de una bifurcación.

4. Eliminación de artefactos: en el paso anterior, se ha conseguido un conjunto de puntos que pueden ser minucias, pero muchos de ellos no lo serán, ya que estarán formados por artefactos de la imagen. Por lo tanto, se vuelve a procesar la imagen (post-procesado), para determinar si cada uno de los puntos anteriormente detectados, puede ser considerado realmente una minucia o no.

Como se puede ver, este algoritmo realiza una serie grande de procesados de la misma huella, por lo que su coste computacional es muy elevado.

1.1.4.2.- Seguimiento de Crestas

Para intentar evitar ese excesivo coste computacional, los profesores de la Universidad de Bolonia, Maio y Maltoni, analizaron la posibilidad de realizar todas las tareas anteriormente descritas, con un único procesado de la imagen, intentado equiparar el algoritmo a lo que la mente humana hace. La filosofía de este método es muy sencilla, y en lenguaje coloquial, el algoritmo se puede describir de la siguiente forma:

1. Determinación de puntos de inicio: la imagen de la huella se cuadrícula, estableciendo una serie de puntos, a partir de los cuales se iniciará el proceso. Simplemente es decidir cada cuántos píxeles se va a considerar que puede haber



una cresta nueva. Es un parámetro del algoritmo y, por tanto, una decisión del diseñador del mismo.

2. Búsqueda de una cresta: tomando un punto de inicio, se busca, en la normal al campo de orientación un mínimo local, que determine dónde se encuentra una cresta.

3. Seguimiento de la cresta hacia la derecha: partiendo de la cresta, se va siguiendo la trayectoria de la cresta. Esto se realiza dando pequeños saltos, en la dirección dada por el campo de orientación para ese punto, y posteriormente, volver a buscar el mínimo local en el nuevo punto, para volver a reubicar la cresta. Si en el punto al que se ha saltado no se localiza una huella, se ha detectado una minucia (terminación). Si, en lugar de eso, chocamos con una cresta que ya hemos estudiado anteriormente, tenemos otra minucia (bifurcación). Si no se produce ninguno de esos casos, seguimos dando saltos, hasta que ocurra uno de estos casos o se acabe la imagen.

4. Seguimiento de la cresta hacia la izquierda: desde el punto de inicio, se realiza el mismo proceso que en el paso anterior, pero en lugar de ir hacia la derecha, siguiendo la cresta, se hace en sentido contrario.

5. Determinación del siguiente punto de inicio: una vez terminada de procesar la cresta, tanto para la derecha como para la izquierda, se escoge un nuevo punto de inicio en la imagen, para seguir nuevas crestas. Evidentemente, este punto se escogerá como el siguiente, que no se encuentre dentro de una zona ya estudiada. Y una vez teniendo el nuevo punto, se vuelve al paso 2. Y así hasta que se acaben los puntos de inicio.

6. Eliminación de redundancias: una vez procesada toda la imagen, se obtendrá un conjunto de minucias, dentro de las cuales se pueden encontrar muchas idénticas. Por lo tanto se hace una depuración de las minucias extraídas, para eliminar las no interesantes.

Como se puede apreciar, este algoritmo es muy parecido a la forma que puede trabajar nuestro cerebro, si nosotros queremos hacer una búsqueda sistemática de las minucias en la imagen.

1.1.4.3.- Verificación De Las Huellas

Una vez que se han obtenido las minucias, se crea con ellas un vector, que servirá para identificar al usuario. De la extracción previa de las huellas del



usuario, se creará un vector de características que será utilizado como patrón del usuario, y será la referencia para su comparación con las muestras que se obtengan cada vez que el usuario quiera identificarse.

La comparación de dicho patrón con la muestra, es una de las tareas más críticas de todo el proceso de identificación.

La problemática viene dada por:

– En las huellas no se dispone de ningún punto de referencia válido universalmente, es decir, un punto a partir del cual se pueda llegar a tomar medidas, y que dicho punto sea válido para todas las huellas. En otras técnicas se tiene un punto claro. Esta falta de referencia, hace que muchos métodos de comparación propuestos, no sean factibles. Algunos autores intentan utilizar como punto de referencia el cono, pero su localización exacta no es sencilla, y además, en algunas huellas no se obtiene ese punto singular.

– La elasticidad del dedo hace que, dependiendo de la postura del dedo y de la presión realizada, se tenga una variabilidad en las medidas de un mismo usuario, que haga bastante difícil la comparación de las huellas obtenidas. De hecho, el problema puede llegar a ser más grave, ya que, dependiendo del grado de rotación que se haya dado al dedo, pueden llegar a capturarse un elevado número de minucias en la muestra, que no se encuentran consideradas en el patrón.

De todos los algoritmos que se han llegado a proponer existe una tendencia generalizada a utilizar el algoritmo de comparación elástica de A. K. Jain. Este algoritmo comienza con la búsqueda de una minucia de referencia. Esto se realiza analizando tanto el vector resultante de la muestra, como el del patrón, de forma que se pueda obtener una minucia cuya semejanza entre los dos vectores sea tan alta, que pueda ser considerada como idéntica.

Posteriormente se hace un cambio de coordenadas de los dos vectores, para pasarlo a polares (radio y ángulo), con el centro en esa minucia de referencia. Una vez realizado esto, se comparan, uno a uno, todos los pares de minucias susceptibles de estar en el mismo sitio. Para considerar que están en el mismo sitio, se crea un área de influencia de cada minucia del patrón, que permita la tolerancia dada por la elasticidad del dedo. Si la minucia de la muestra está en la zona de influencia de la minucia del patrón, y su tipo y orientación son compatibles, entonces se podrá determinar que la minucia es la misma.



Realizando esto para todos los pares de minucias, y aplicando unas fórmulas de medida de proximidad, con factores de penalización para los casos en los que la comparativa haya sido infructuosa, se consigue un resultado numérico que dará la probabilidad de que las huellas sean idénticas.

Como en todo sistema biométrico, dependiendo del umbral que ponga el diseñador del sistema, la probabilidad hará que la huella sí que se considere idéntica, o falsa. Se estudiarán entonces las posibles tasas de error: Falsa Aceptación (FAR) o Falso Rechazo (FRR), para indicar la calidad del sistema de identificación en la aplicación dada. [8]

1.2.- Ingeniería de software

La ingeniería del software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.

1.2.1.- ¿Qué es un sistema?

Un sistema es una caja negra que tiene un conjunto de una o más entradas y un conjunto de una o más salidas.

En el caso de sistemas de software, las entradas consisten en datos digitales provenientes de dispositivos de entrada u otros sistemas de software, y las salidas son datos digitales.

El tiempo de respuesta de un sistema de software es el que transcurre entre que un conjunto de entradas al sistema y que aparecen todas las salidas asociadas del mismo.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos.

En los pasados 50 años, el software ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por sí misma. Pero la temprana cultura e historia de la «programación» ha creado un conjunto de problemas que persisten todavía hoy. El software se ha convertido en un factor que limita la evolución de los sistemas informáticos. El intento de la ingeniería del software es proporcionar un marco de trabajo para construir software con mayor calidad.



El impacto del software en nuestra sociedad y en la cultura continúa siendo profundo. Al mismo tiempo que crece su importancia, la comunidad del software trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de programas de computadora de alta calidad. Hoy en día el software tiene un doble papel. Es un producto y, al mismo tiempo, el vehículo para entregarlo.

1.2.2.-Características del software

1. El software se desarrolla no se fabrica.
2. El software no se estropea.
3. Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida. [9]

1.2.3.- Proceso del software

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software.

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

1. *Especificación del software* donde los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.
2. *Desarrollo del software* donde el software se diseña y programa.
3. *Validación del software* donde el software se valida para asegurar que es lo que el cliente requiere.
4. *Evolución del software* donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

El uso de un proceso inadecuado del software puede reducir la calidad o la utilidad del producto de software que se va a desarrollar y/o incrementar los costes de desarrollo.



1.2.4.-Modelo de proceso del software

Un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del 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 junto 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 realiza alguna transformación en los datos.

Muestra cómo la entrada en el proceso, tal como una especificación, se transforma en una salida, tal como un diseño. Pueden representar transformaciones llevadas a cabo por las personas o por las computadoras.

3. *Un modelo de rollación.* Representa los roles de las personas involucrada en el proceso del software y las actividades de las que son responsables.

La mayor parte de los modelos de procesos del software se basan en uno de los tres modelos generales o paradigmas de desarrollo de software:

1. *Cascada.* Considera las actividades anteriores y las representa como fases de procesos separados, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera. Después de que cada etapa queda definida «se firma» y el desarrollo continúa con la siguiente etapa.

2. *Desarrollo iterativo.* Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga las necesidades de dicho cliente. El sistema puede entonces ser entregado. De forma alternativa, se puede re implementar utilizando un enfoque más estructurado para producir un sistema más sólido.



3. *Ingeniería del software basada en componentes (CBSE)*. Esta técnica supone que las partes del sistema existen. El proceso de desarrollo del sistema se enfoca en la integración de estas partes más que desarrollarlas desde el principio

En este caso utilizaremos el modelo de cascada por lo que ahondaremos más en esta metodología.

El primer modelo de proceso de desarrollo de software que se publicó se derivó de procesos de ingeniería de sistemas más generales (Royce, 1970). Este modelo se muestra en la Figura 6. Debido a la cascada de una fase a otra, dicho modelo se conoce como modelo en cascada o como ciclo de vida del software. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

1. *Análisis y definición de requerimientos*. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen y sirven como una especificación del sistema.

2. *Diseño del sistema y del software*. El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.

3. *Implementación y prueba de unidades*. Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.

4. *Integración y prueba del sistema*. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.

5. *Funcionamiento y mantenimiento*. Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos. [10]



CAPITULO 1.- MARCO TEÓRICO

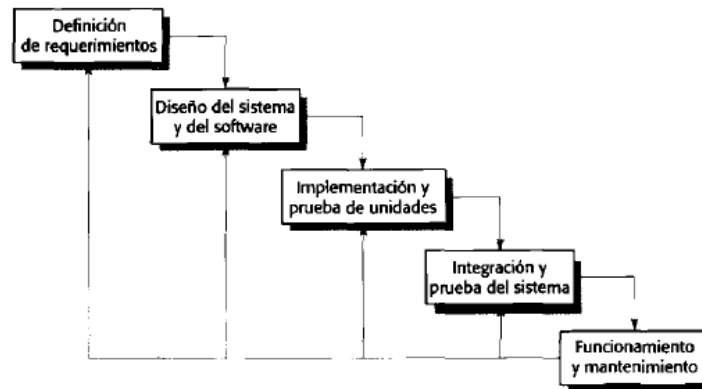


Figura 1. 6.- Ciclo de vida del software

1.3.- Bases de datos

Las bases de datos tienen un impacto decisivo sobre el creciente uso de las tecnologías. Las bases de datos desempeñan un papel crucial en casi todas las áreas de aplicación ya sea la ingeniería, la industria, los negocios, la medicina, solo por mencionar algunas. El término base de datos es muy común en la actualidad que es necesario dar nuestra definición aunque de manera muy general. [11]

Una base de datos es una colección de datos relacionados entre si, contiene información relevante para una empresa por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. [11,12]

Es necesario tener muy en claro algunos conceptos de para comprender el desarrollo e implementación de nuestra aplicación.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. [13]

1.3.1.- Antecedentes

Los predecesores de los sistemas de bases de datos son los sistemas de ficheros. Un sistema de ficheros está formado por un conjunto de programas que



dan servicio a los usuarios finales. Cada programa define y gestiona sus propios datos. Aunque los sistemas de ficheros supusieron un gran avance sobre los sistemas manuales, tienen inconvenientes bastante importantes, como la redundancia de datos y la dependencia entre programas y datos.

Los sistemas de bases de datos surgieron con el objetivo de resolver los problemas que planteaban los sistemas de ficheros. Una base de datos es un conjunto de datos relacionados que recogen las necesidades de información de una empresa u organización. Estos datos se comparten por todos los usuarios. El SGBD es un conjunto de programas que permiten a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a dicha base de datos. [13]

La base de datos contiene tanto los datos como su definición. Todos los accesos a la base de datos se realizan a través del SGBD. El SGBD proporciona un lenguaje de definición de datos que permite a los usuarios definir la base de datos, y un lenguaje de manejo de datos que permite a los usuarios la inserción, actualización, eliminación y consulta de datos de la base de datos.

El SGBD proporciona un acceso controlado a la base de datos. Proporciona seguridad, integridad, concurrencia y controla la recuperación ante fallos. Además, proporciona un mecanismo de vistas que permite mostrar a los usuarios sólo aquellos datos que les interesan.

Las personas involucradas en el entorno de una base de datos son: el administrador de la base de datos, los diseñadores de la base de datos, los programadores de aplicaciones y los usuarios finales.

El modelo relacional, propuesto por Codd en 1970, representa la segunda generación de los SGBD. Este modelo es el más extendido en la actualidad. La tercera generación de los SGBD se encuentra representada por el modelo relacional extendido y el modelo orientado a objetos.

1.3.2.- Modelo entidad-relación

El modelo de datos entidad-relación (E-R) se basa en una percepción del mundo real que consta de una colección de objetos básicos, llamados *entidades*, y de *relaciones* entre estos objetos. Una entidad es se puede interpretar como una cosa u objeto en el mundo real con características distintas de otros objetos.



Las entidades se describen en una base de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un *diagrama ER*, que consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con la entidad o relación que representa. [13]

En la Figura 1.7 se muestra un ejemplo de diagrama E-R.

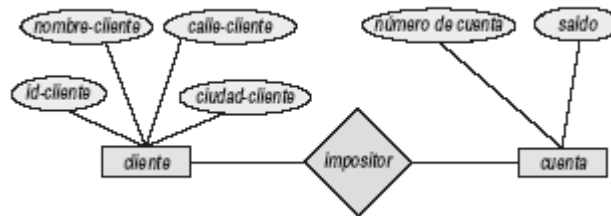


Figura 1.7. -Ejemplo de diagrama E-R

1.3.3.- Modelo Relacional

Bajo la estructura de la base de datos se encuentra el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

El modelo relacional se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos. Relacional se basa en el concepto matemático denominado “relación”, que gráficamente se puede representar como una tabla. En el modelo relacional, los datos y las relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una con un nombre que es único y con un conjunto de columnas.



El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro.

El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional. [12,13]

1.3.4.- Bases de datos relacionales

Una base de datos relacional es un repositorio compartido de datos. Para hacer disponibles los datos de una base de datos relacional a los usuarios hay que considerar varios aspectos.

Uno es la forma en que los usuarios solicitan los datos es decir los tipos y formas de consultar la información. Otro aspecto es la integridad de datos y la seguridad; las bases de datos necesitan proteger los datos del daño provocado por los usuarios, ya sean intencionados o no. El componente de mantenimiento de la integridad de una base de datos asegura que las actualizaciones no violan las restricciones de integridad que hayan especificado sobre los datos.

En una base de datos relacional, todos los datos se almacenan y se acceden a ellos por medio de relaciones. Las relaciones que almacenan datos son llamados "relaciones base" y su implementación es llamada "tabla". Otras relaciones no almacenan datos, pero que son calculadas al aplicar operaciones relacionales. Estas relaciones son llamadas "relaciones derivadas" y su implementación es llamada "vista" o "consulta". Las relaciones derivadas son convenientes ya que expresan información de varias relaciones actuando como si fuera una sola.

La base de datos se organiza en dos marcadas secciones; el esquema y los datos (o instancias).

El esquema es la definición de la estructura de la base de datos y principalmente almacena los siguientes datos:

- El nombre de cada tabla
- El nombre de cada campo



- El tipo de dato de cada campo
- La tabla a la que pertenece cada campo

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización, el resultado de dicho proceso es un esquema que permite que la base de datos sea usada de manera óptima.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver.

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

En el modelo relacional los atributos deben estar explícitamente relacionados a un nombre en todas las operaciones, en cambio, el estándar SQL permite usar columnas sin nombre en conjuntos de resultados, como el asterisco taquigráfico (*) como notación de consultas.

Al contrario del modelo relacional, el estándar SQL requiere que las columnas tengan un orden definido, lo cual es fácil de implementar en una computadora, ya que la memoria es lineal.

Sin embargo, en SQL el orden de las columnas y los registros devueltos en cierto conjunto de resultado nunca está garantizado, a no ser que explícitamente sea especificado por el usuario.

Sin embargo, todo lo dicho es dicho y una base de datos relacional es utilizada para la formación del ingreso de datos de forma sistematizada, "fácil", y ordenada

1.3.5. – Normalización

En general, el objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que nos permita almacenar la información sin redundancias innecesarias, pero que también nos permita recuperar fácilmente esa información. Un enfoque es el diseño de esquemas que se encuentren en una *forma normal* adecuada.

Primera forma normal (1FN)



La primera forma normal, impone un requisito muy elemental a las relaciones; a diferencia de las demás formas normales, no exige información adicional como las dependencias funcionales.

Un dominio es atómico si se considera que los elementos del dominio son unidades indivisibles. Se dice que el esquema de una relación R está en la primera forma normal (1FN) si los dominios de todos los atributos de R son atómicos, también prohíbe atributos compuestos que por si mismos son multivaluados, estos se denominan relaciones anidadas ya que cada tupla puede tener una relación dentro de sí.

Segunda forma normal (2FN)

Una relación esta en 2FN si cumple con 1FN y que todo atributo no primo en la relación depende funcionalmente de manera total de la clave primaria.

Si un esquema no se encuentra en 2FN se le puede normalizar en varias relaciones en las que los atributos no primos estén asociados solo a la parte de la llave primaria de la que dependen funcionalmente de manera total

Tercera forma normal (3FN)

La tercera forma normal se basa en el concepto de dependencia transitiva. De acuerdo con la definición original de Codd, un esquema se encuentra en 3FN si se encuentra en 2FN y ningún atributo no primo de la relación depende transitivamente de la llave primaria.

1.3.6.- El lenguaje de consulta

Un lenguaje de consulta es un lenguaje en el que un usuario solicita información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consulta pueden clasificarse como procedimentales o no procedimentales. En los lenguajes procedimentales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos para calcular el resultado deseado. En los lenguajes no procedimentales el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

La mayor parte de los sistemas comerciales de bases de datos relacionales ofrecen un lenguaje de consulta que incluye elementos de los enfoques procedimental y no procedimental.



En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes separados; en su lugar simplemente forman partes de un único lenguaje de bases de datos, tal como SQL.

SQL permite el uso de valores nulos para indicar que el valor o bien es desconocido, o no existe. Se fijan criterios que permiten al usuario especificar a qué atributos no se puede asignar valor nulo.

La estructura básica de una expresión SQL consiste en tres cláusulas: select, from y where.

- La cláusula select corresponde a la operación proyección del álgebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.
- La cláusula from corresponde a la operación producto cartesiano del álgebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión.
- La cláusula where corresponde al predicado selección del álgebra relacional. Es un predicado que engloba a los atributos de las relaciones que aparecen en la cláusula from.[13]

Toda la información anterior está dada de una forma muy general, solo para que tengamos un concepto más claro con respecto al tipo de base de datos que se desea aplicar a nuestro proyecto.

1.4.-Herramientas de desarrollo y diseño

1.4.1.- Microsoft SQL server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico.

Con la aparición de SQL Server 2005 el mundo de las Bases de datos está cambiando. Los desarrolladores ahora pueden ubicar su código apropiadamente en relación a su funcionalidad, acceder a datos nativos como XML, y construir sistemas complejos que sean manejados por el servidor de Bases de Datos. Estos



puntos hacen que el desarrollo de Bases de Datos esté encaminado hacia una integración.

SQL Server 2005 es más que un sistema gestor de Bases de Datos ya que incluye múltiples componentes y servicios que la convierten en una plataforma de aplicaciones corporativas.

Componentes de SQL Server 2005:

En la figura 1.8 se ilustran las relaciones existentes entre los componentes de SQL Server 2005 y se indica la interoperabilidad entre ellos.

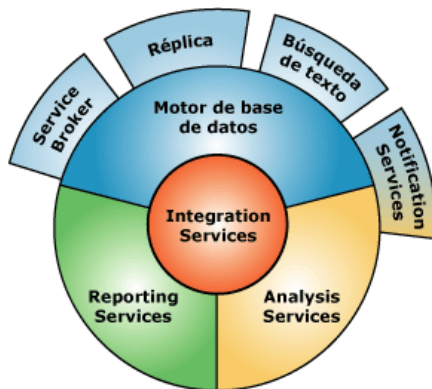


Figura 1. 8.- Relaciones entre componentes de SQL Server

SQL Server 2005 incluye varias características de seguridad configurables y de gran precisión. Estas características permiten a los administradores implementar una defensa optimizada para los riesgos de seguridad específicos de su entorno.

Características

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones



2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma *gratuita*. [14]

1.4.2. - AJAX

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

En realidad AJAX no es una tecnología, sino la unión de varias tecnologías que juntas pueden lograr cosas realmente impresionantes como GoogleMaps, Gmail, el Outlook Web Access o algunas otras aplicaciones muy conocidas.

En realidad, el término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML" y el concepto es: Cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en *background*, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma.

Definiendo Ajax

Ajax no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- presentación basada en estándares usando XHTML y CSS;
- exhibición e interacción dinámicas usando el Document Object Model ;
- Intercambio y manipulación de datos usando XML and XSLT ;
- Recuperación de datos asincrónica usando XMLHttpRequest ;
- y JavaScript poniendo todo junto.

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una página HTML al cliente.



CAPITULO 1.- MARCO TEÓRICO



Figura1. 9.- El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der.).

Este acercamiento tiene mucho sentido a nivel técnico, pero no lo tiene para una gran experiencia de usuario. Mientras el servidor está haciendo lo suyo el usuario está esperando. Y, en cada paso de la tarea, el usuario espera por más.

Una aplicación AJAX elimina la naturaleza “arrancar-frenar- arrancar-frenar” de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario.

En vez de cargar un pagina Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript y usualmente “sacado” en un frame oculto). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario.

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

Cada acción de un usuario que normalmente generaría un requerimiento HTTP toma la forma de un llamado JavaScript al motor AJAX en vez de ese requerimiento. Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor es manejado por su cuenta.

Si el motor necesita algo del servidor para responder hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación. [15]



1.4.3.- ASP.NET

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Cualquier persona que está familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de *script* y plataformas de servidor. Desafortunadamente para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones web tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.[16]



2.- Análisis Y Diseño Del Sistema

En este capítulo estableceremos la problemática a resolver en este trabajo de tesis, analizando los principales puntos a desarrollar para el sistema, especificando los requerimientos proporcionados por el usuario, desarrollaremos los casos de uso detectados, con sus respectivos diagramas, también identificaremos las entidades con sus atributos correspondientes para poder realizar posteriormente el diagrama entidad-relación, el modelo relacional y el diseño físico y lógico de la base de datos, así como el diseño de la interfaz de nuestro sistema.

2.1.- Planteamiento del problema

Actualmente en la Vicerrectoría de Docencia se necesita llevar el registro de asistencias de su personal, lo cual se realiza mediante el llenado de hojas de papel, las cuales se van cambiando día a día, y se tiene que realizar el conteo de las horas manualmente, es decir, aún se tiene que firmar la hora de entrada y salida lo cual es muy tedioso e inconsistente.

El conteo de las asistencias registradas, en el caso de los estudiantes de servicio social, práctica profesional y personal que labora por horas, es realizado por el encargado de cada área de manera diferente, esto tiene limitantes, cuando se desean realizar tareas como generar informes, consultar reportes, hacer búsquedas, extraer datos concretos que reflejen el comportamiento de la información, además de que no se tiene la certeza de que en realidad los datos registrados sean los correctos.

Se requiere llevar un historial (bitácora) de los estudiantes que han apoyado los distintos procesos, para su seguimiento en el aspecto profesional, así como también la elaboración de reportes de las asistencias de los trabajadores de la dependencia, el cual contendrá la cantidad de horas laboradas, inasistencias, y el monto del pago que le corresponde, de acuerdo al tipo de empleado y área en el que labora.

Es por ello que se plantea una solución tecnológica al problema por medio de la implementación de un sistema administrativo capaz de automatizar el registro de asistencia del personal por medio de un lector biométrico, mismo que será almacenado en una base de datos para después generar los reportes y consultas necesarias.



Para la solución de este problema nos apoyaremos en conceptos y estrategias aplicados actualmente al desarrollo de sistemas administrativos con bases de datos. La base de datos que se pretende diseñar almacenará información específica, acerca de los diferentes tipos de reportes que se desean obtener, así como los registros de asistencia de los empleados.

2.1.1.- Especificación de Requerimientos

Específicamente es necesario desarrollar un sistema funcional capaz de obtener el registro de asistencias del personal, y a su vez permita automatizar dichos registros para agilizar la emisión de los reportes y consultas necesarios para cada área en específico.

Para el sistema a desarrollar es necesario considerar las siguientes operaciones:

1. Registrar por medio de un lector biométrico el número de horas trabajadas, inasistencias y retardos de los empleados de nómina institucional, interna y por honorarios.
2. Registrar por medio de un lector biométrico el número de horas trabajadas, inasistencias y retardos de los empleados temporales.
3. Registrar por medio de un lector biométrico el número de horas cubiertas por los estudiantes de Servicio Social y Prácticas Profesionales
4. Informar el número de horas trabajadas, inasistencias y retardos de los empleados de nómina institucional, interna y por honorarios conforme sean requeridos.
5. Informar el número de horas trabajadas, inasistencias y retardos de los empleados conforme sean requeridos.
6. Informar el número de horas cubiertas por los estudiantes de Servicio Social y Prácticas Profesionales conforme sean requeridos.
7. Generar reportes del pago a personal temporal.
8. Buscar registros.
9. Modificar registros.



10. Eliminar registros.

Los usuarios deben proporcionar su nombre y su ID de trabajador para poder acceder al sistema.

El sistema permite a los empleados pasar lista por medio de un checador biométrico el cual enviara los datos a la base de datos donde serán almacenados.

El sistema generara los siguientes reportes:

- Un informe que incluirá el registro de las horas cubiertas, por los estudiantes de Servicio Social y Prácticas Profesionales conforme sean requeridos.
- Un informe que incluirá el registro de las horas laboradas, inasistencias y retardos de los empleados de nómina institucional, interna y por honorarios conforme sean requeridos.
- Un informe que incluirá el registro de las horas laboradas, inasistencias y retardos que nos permite calcular el monto a pagar al personal que labora por hora.

El usuario podrá consultar los reportes:

- Por área
- Por tipo de empleado
- Por tipo de estudiante
- Por Fecha

2.1.1.1.- Definición conceptual del sistema

El sistema al iniciar nos mostrará una pantalla en la cual el usuario se debe autenticar para poder acceder a las funciones del sistema, de acuerdo al tipo de usuario del que se trate, ya sea administrador, coordinador o empleado.

El sistema nos mostrara una pantalla con las siguientes opciones:

- Administrar (solo tiene acceso el administrador): el sistema visualizará otra pantalla donde se podrán gestionar empleados, estudiantes y áreas.



- Cada uno de estas opciones tendrá las opciones de registrar, editar y eliminar.
- Consultar Reportes (solo tienen acceso el administrador y coordinador): el sistema muestra una pantalla para la consulta de los reportes correspondientes la cual podrá generar 2 tipos de reportes:
 - Reporte General: Muestra de manera resumida las estadísticas de asistencia registradas en el sistema.
 - Reporte Detallado: Muestra a detalle las asistencias registradas en el sistema.
- Ver historial personal: el sistema nos mostrará una pantalla en la cual el empleado puede ver su evolución.

Además de un apartado en el cual se puede registrar asistencia del empleado, en caso de que exista alguna contingencia en el checado por medio del lector biométrico, dicho apartado solo estará disponible para un administrador o coordinador de área.

A continuación las funciones más importante del sistema:

- Agregar Empleados, Estudiantes y Áreas.
- Editar Empleados, Estudiantes y Áreas.
- Eliminar Empleados, Estudiantes y Áreas.
- Registrar asistencias
- Consultar Asistencias.
- Emitir Reportes.
- Imprimir Reportes.

2.1.2.-Modelo conceptual del sistema

La función que debe cumplir el modelo conceptual es la de mostrar a grandes rasgos el funcionamiento del sistema, en el cual se deben identificar los servicios principales que se proporcionan al usuario, esto se muestra en la figura 2.1.



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA

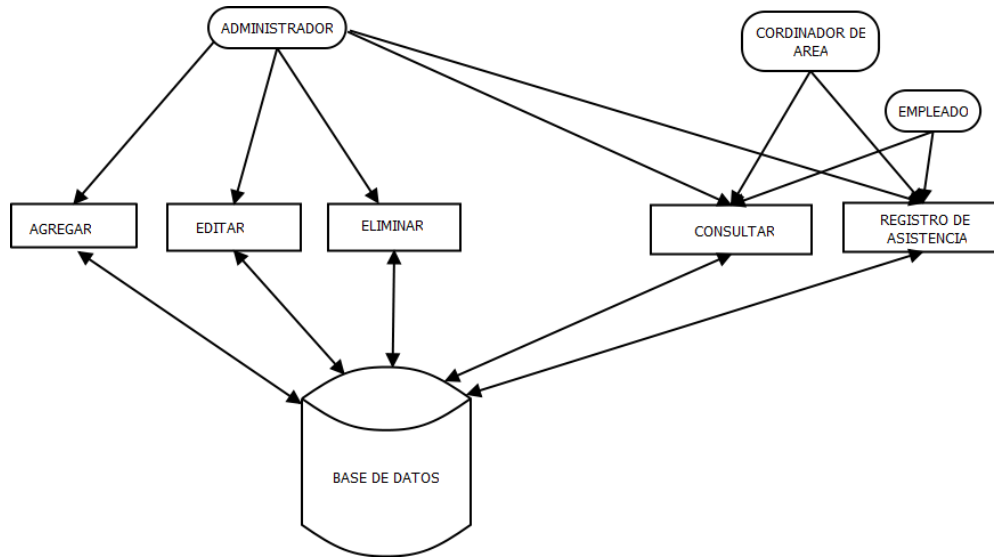


Figura 2.1.-Modelo conceptual del sistema

2.1.2.1.-Diagrama de flujo de datos

El objetivo del diagrama de flujo de datos (DFD'S), es mostrarnos una representación grafica de las interrelaciones de sus funciones, ya que nos muestra los flujos de entradas y salidas de datos a través del sistema, y finalmente nos servirá como una plataforma para fundamentar el mismo, como ayuda en el diseño de la base de datos y de las pantallas.

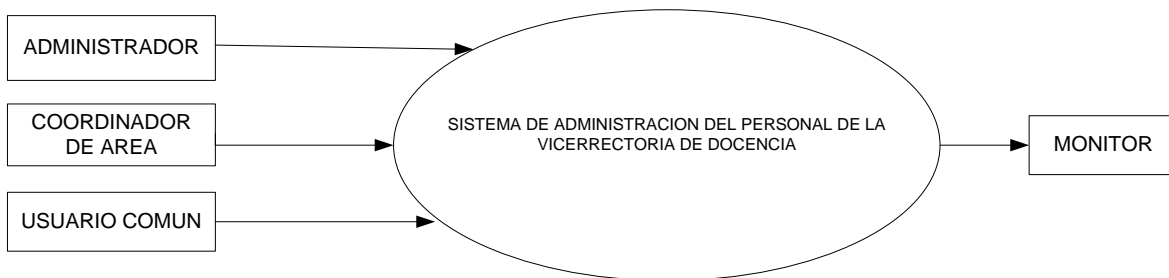


Figura 2.2.-Diagrama de Flujo de Datos (DFD'S) nivel 0

A partir del diagrama de la figura 2.2 podemos observar el tipo de salida que tendrá nuestro sistema.



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA

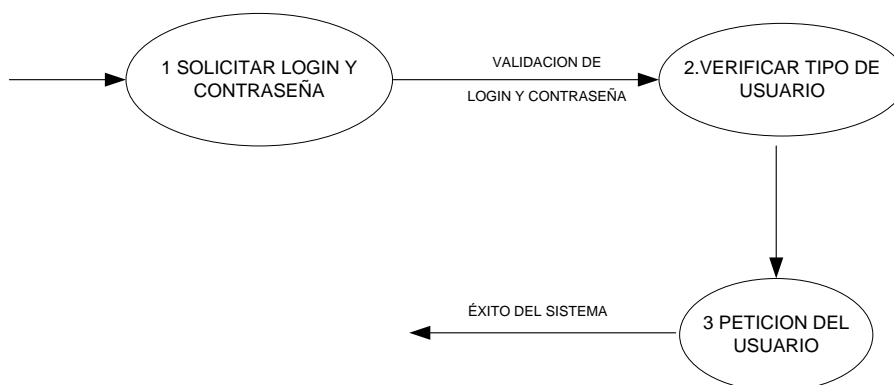


Figura 2.3.-DFD'S nivel 1, Sistema de administración del personal

En este diagrama de la figura 2.3 se solicita el login y contraseña, se validan los datos en la base de datos para verificar el tipo de usuario, después se presenta otra interface de acuerdo al tipo de usuario para que pueda realizar la tarea específica que se haya solicitado.

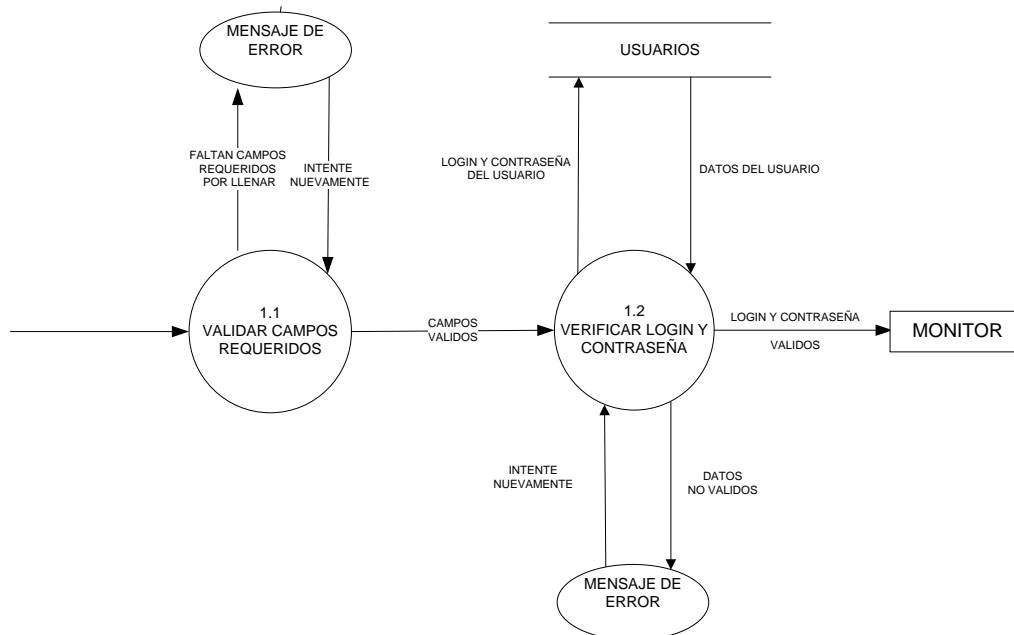


Figura 2.4.-DFD'S Nivel 2, **Solicitar login y contraseña 1**

En el diagrama de la figura 2.4 se solicita el login y contraseña de un usuario, posteriormente se verifica que los campos requeridos no estén vacíos, de lo contrario se emitirá un mensaje de error, finalmente se verifica que los datos existan en la base de datos, de lo contrario no se permitirá el acceso al sistema.

La figura 2.5 muestra como al obtener login y contraseña validos, se muestra una interface de acuerdo al tipo de usuario que se haya identificado.



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA

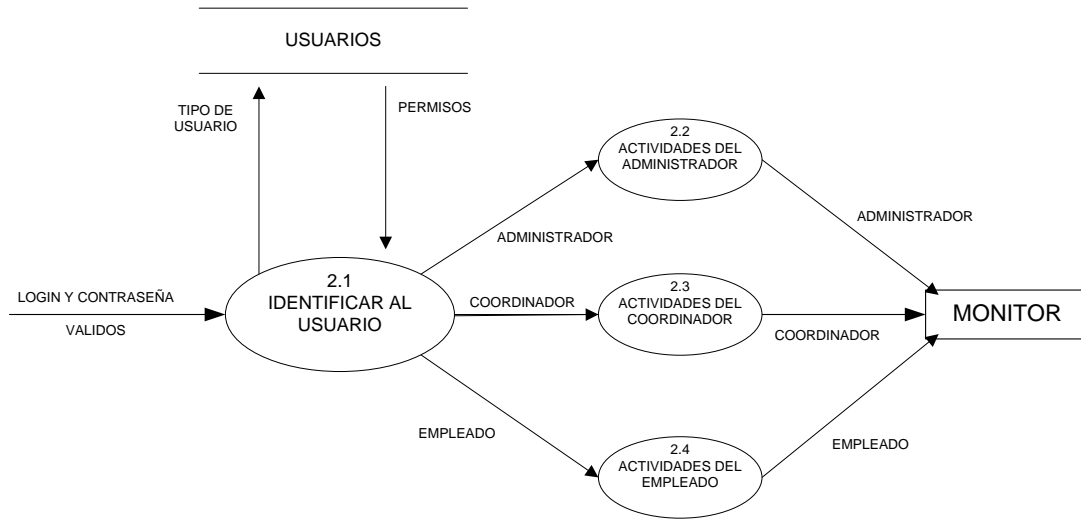


Figura 2.5.-DFD'S Nivel 2, Verificar tipo de usuario 2

En la figura 2.6 se presentan los módulos que comparten todos los usuarios.

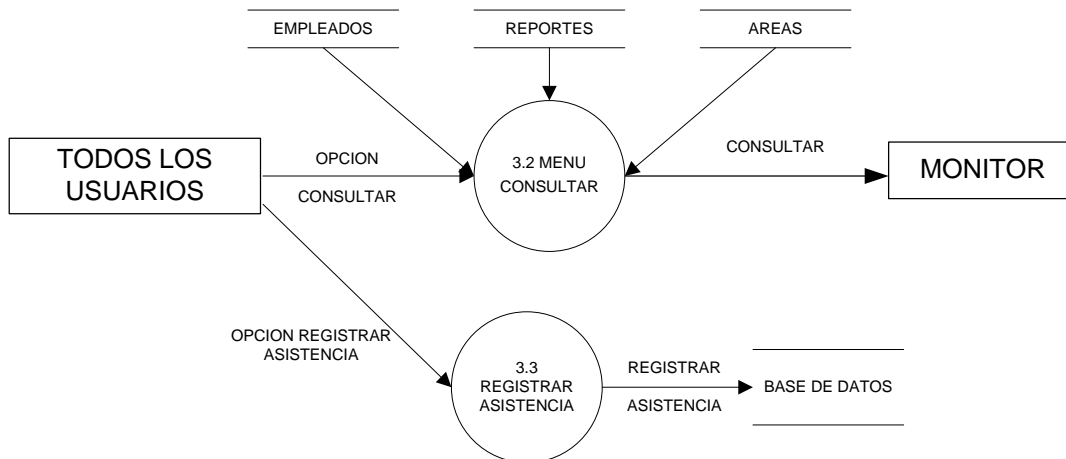


Figura 2.6.-DFD'S Nivel 2, Petición del usuario 3

Ahora podemos observar en la figura 2.7 solo el modulo que le corresponde al administrador del sistema.

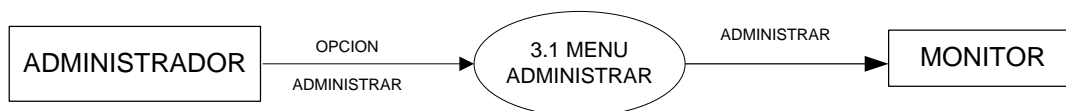


Figura 2.7.-DFD'S Nivel 2, Petición del administrador 3



En la figura 2.8 podemos ver los módulos concernientes al menú consultar, identificando las opciones con respecto al tipo de usuario.

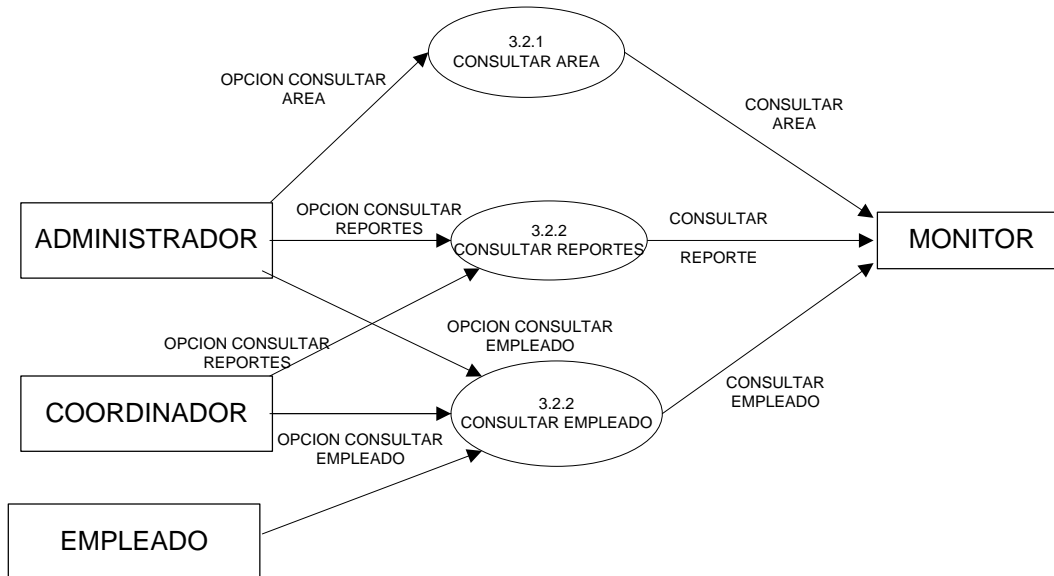


Figura 2.8.-DFD'S Nivel 3, **Menú consultar 3.2**

En el diagrama de la figura 2.9 se observan los diferentes módulos de la opción administrar.

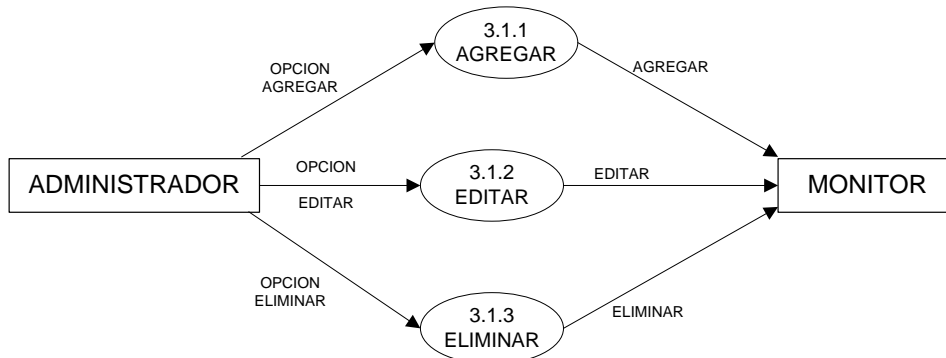


Figura 2.9.-DFD'S Nivel 3, **Menú administrar 3.1**

Podemos ver en el diagrama de la figura 2.10 la descripción del módulo agregar, en el cual el administrador de alta los datos de los empleados, estudiantes o áreas, las cuales son validados antes de enviarlos a la base datos.



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA

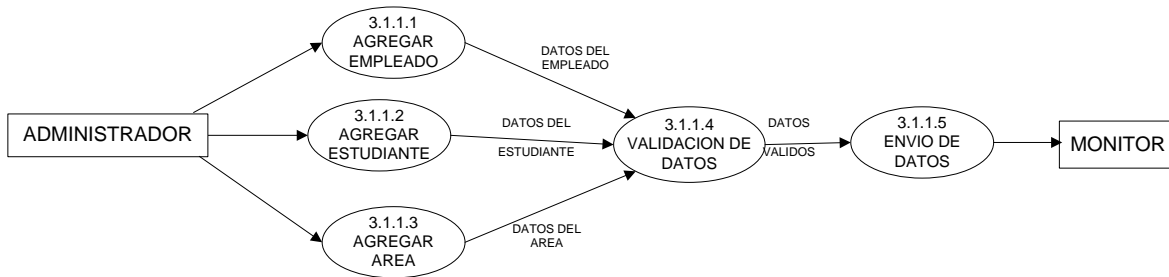


Figura 2.10.-DFD'S Nivel 4, Agregar 3.1.1

En la figuras 2.11 y 2.12 se observa el funcionamiento de los módulos editar y eliminar respectivamente, en los cuales antes de enviar los datos al servidor se validan dichos datos solicitados.

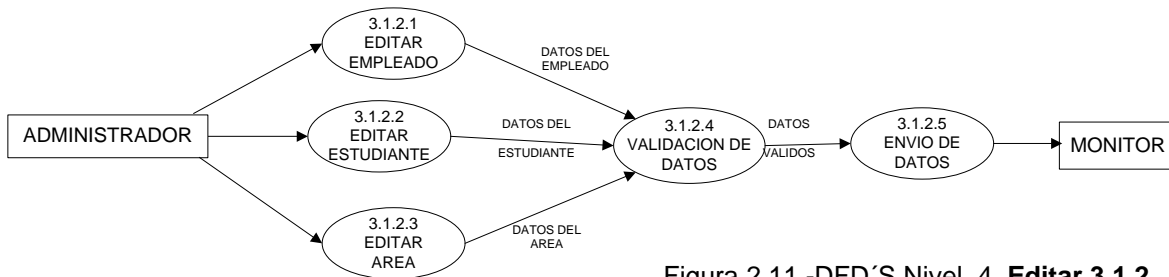


Figura 2.11.-DFD'S Nivel 4, Editar 3.1.2

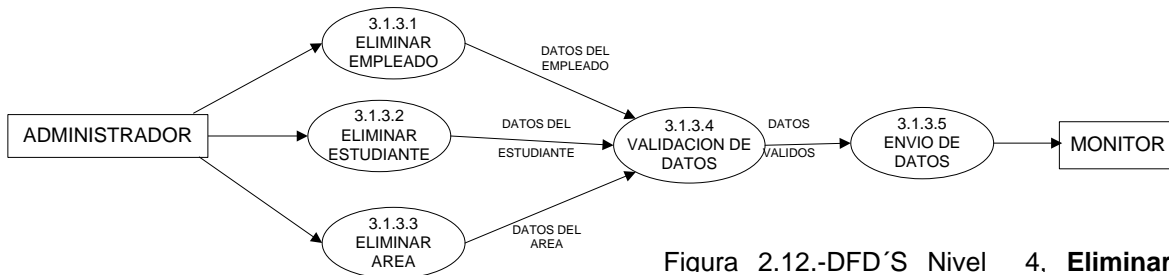


Figura 2.12.-DFD'S Nivel 4, Eliminar

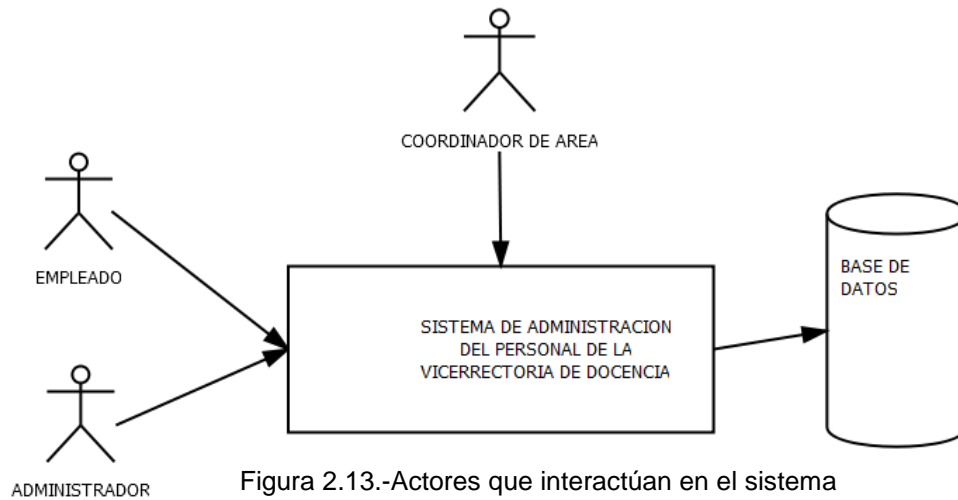
2.1.3.-Casos de uso.

La función primordial de los casos de uso es dar una idea o bosquejo en el cual se deben reflejar las necesidades del usuario y su interacción con el sistema.

A continuación se describen los actores que interactúan en el sistema y sus respectivos casos de uso.

2.1.3.1.- Identificación de actores:

En la figura 2.13 se observa una aproximación de lo que será el sistema identificando los principales actores que interactúan en dicho proyecto.



Los actores que participan en el sistema son: Administrador, coordinador de área, empleado y el sistema de base de datos. Las actividades de cada uno de los actores dentro del sistema se describen a continuación:

Administrador:

- Gestión de empleados.
- Gestión de estudiantes.
- Gestión de áreas.
- Registrar asistencias.
- Consultar Reportes.

Coordinador de área:

- Registrar asistencias.
- Consultar Reportes.

Empleado o Estudiante:

- Registrar asistencia.
- Consultar su historial.

Sistema de base de datos:

- Autenticar los usuarios.
- Emisión de reportes.



2.1.3.2.- Diagrama de casos de uso del sistema

En la figura 2.14 se muestra una aproximación del funcionamiento del sistema de forma general, en este diagrama, se presentan los actores que interactúan de manera directa con el sistema y sus casos de usos sobre los cuales establece una relación:

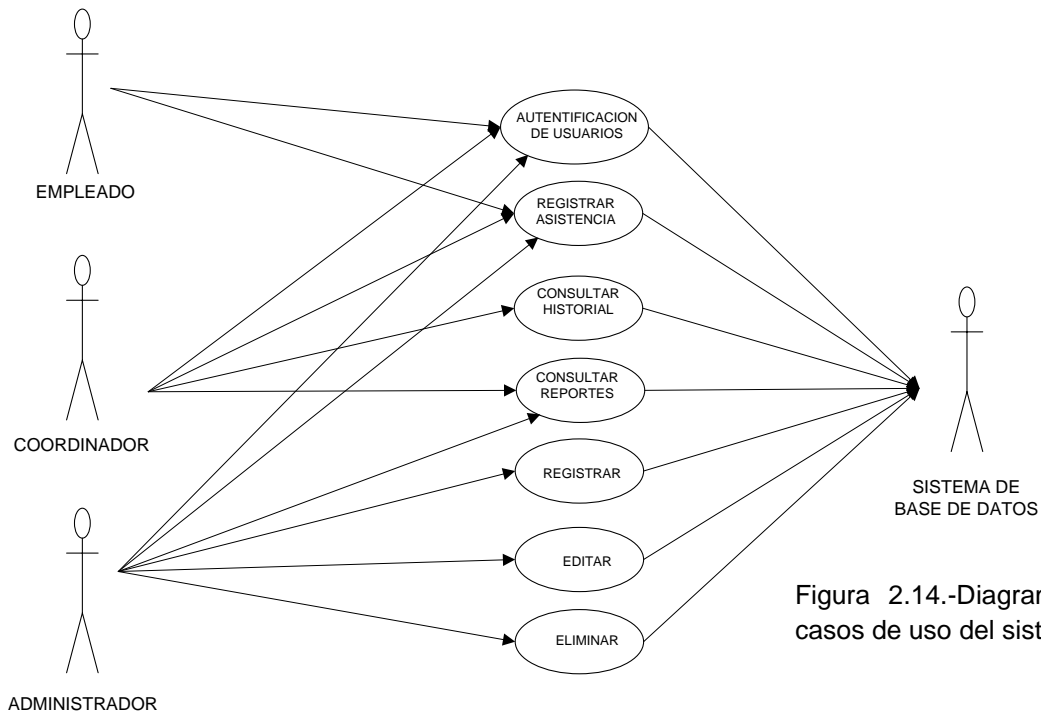


Figura 2.14.-Diagrama general de casos de uso del sistema.

2.1.3.2.1.- Flujo de eventos del sistema

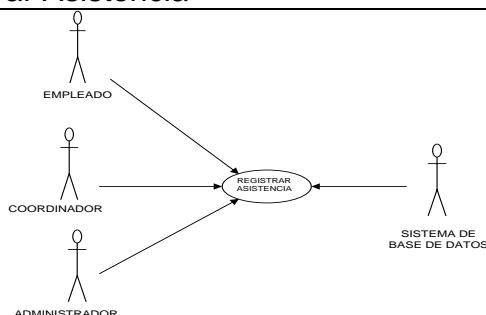
El propósito del flujo de eventos es documentar el flujo que seguirán los casos de uso. En el flujo de eventos describe a detalle lo que realizan los usuarios del sistema y como responde éste a las solicitudes que le son requeridas.

CASO DE USO: Autenticación de Usuarios	
Descripción:	El usuario proporciona sus datos de acceso los cuales son validados por el sistema, si los datos son correctos se le permite el acceso de acuerdo al tipo de



	privilegio otorgado.
Actores:	Administrador, coordinador de área, y Empleado
Precondiciones:	Los usuarios deben de estar registrados en el sistema y contar con sus claves de acceso.
Flujo Normal:	<p>1.- Inicia el caso de uso: El sistema solicita nombre de usuario y contraseña del usuario (Empleado, coordinador ó administrador).</p> <p>2.-El usuario proporciona su nombre de usuario y clave.</p> <p>3.-El sistema valida con el sistema de base de datos los datos del empleado, coordinador ó administrador según sea el caso.</p> <p>4.-Fin del caso de uso: El sistema otorga acceso a la página del menú principal al usuario.</p>
Flujos Alternativos o excepciones:	3.- El sistema comprueba la validez de los datos, si los datos no son correctos, envía un mensaje al actor permitiéndole proporcionarlos nuevamente.
Postcondiciones:	El actor ha ingresado correctamente al Sistema.

CASO DE USO: Registrar Asistencia



Descripción:	Los usuarios registran su asistencia por medio un lector biométrico, las cuales se almacenan en la base de datos.
Actores:	Empleado, Coordinador de área y Administrador
Precondiciones:	El usuario debe de estar registrado en el sistema y en el lector biométrico.
Flujo Normal:	1.- Inicia el caso de uso:



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA



	El usuario pasa su huella dactilar por el lector biométrico. 2.- Fin del caso de uso: Los datos son enviados al servidor.
Flujos Alternativos o excepciones:	2.-Si el usuario no existe en los registros los datos son descartados.
Postcondiciones:	3.-Los datos son almacenados

CASO DE USO: Consultar historial	
<pre> graph TD EMPLEADO --> U1((CONSULTAR HISTORIAL)) COORDINADOR --> U1 ADMINISTRADOR --> U1 U1 --- S1[SISTEMA DE BASE DE DATOS] </pre>	
Descripción:	Los usuarios podrán consultar sus horas laboradas registradas en el sistema.
Actores:	Empleado, Coordinador de área, Administrador
Precondiciones:	Los usuarios deben de ser autenticados para poder acceder. El sistema debe de tener registros de las horas laboradas de los usuarios.
Flujo Normal:	1.- Inicia el caso de uso: El usuario elige la opción ver historial en el menú principal. 2.-El sistema despliega las diferentes opciones para visualizar su progreso. 3.-El usuario elige la opción que desea. 4.-Fin del caso de uso: El sistema despliega los datos almacenados según la opción elegida.
Flujos Alternativos o excepciones:	4.-El sistema desplegara el mensaje “no se tienen registros almacenados actualmente” en caso de no contar con los registros solicitados.
Postcondiciones:	El usuario habrá visto todos sus datos almacenados.

CASO DE USO: Consultar Reportes

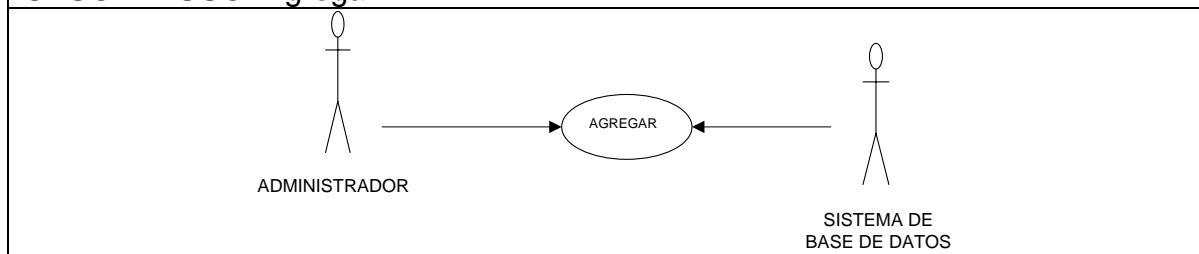


CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA



Descripción:	Los usuarios pueden consultar los diferentes tipos de reportes que están almacenados en la base de datos.
Actores:	Administrador o Coordinador de Área
Precondiciones:	Los usuarios deben de ser autenticados en el sistema. Se deben de tener datos registrados en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1.- Inicia el caso de uso: El usuario debe seleccionar la opción Consultar Reportes en el menú principal. 2.-El sistema mostrará otra pantalla con los diferentes filtros para generar los reportes a consultar. 3.-El actor debe de elegir la opción que deseé pulsando en el vínculo correspondiente a su búsqueda. 4.- Fin del caso de uso: El sistema mostrará el reporte seleccionado.
Flujo Alternativo o excepciones:	4.-El sistema desplegara el mensaje "no se tienen registros almacenados actualmente" en caso de no contar con los registros solicitados.
Postcondiciones:	El actor podrá imprimir, o guardar los reportes solicitados.

CASO DE USO: Agregar





Descripción:	El administrador una vez autenticado como tal podrá dar de alta un empleado, Estudiante o Área con sus datos correspondientes para que este pueda interactuar con el sistema
Actores:	Administrador
Precondiciones:	Los usuarios deben de ser autenticados en el sistema.
Flujo Normal:	<p>1.- Inicia el caso de uso: El usuario deberá elegir el vinculo Administrar en el menú principal.</p> <p>2.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para administrar.</p> <p>3.- El usuario ahora elegirá la opción agregar.</p> <p>4.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para agregar.</p> <p>5.- El usuario ahora elegirá la opción registrar empleado, estudiante o área según sea el caso.</p> <p>6.- El sistema mostrará un formulario el cual tendrá que ser llenado con todos los datos correspondientes.</p> <p>7.- El usuario deberá llenar todos los campos solicitados para guardar los datos y después deberá pulsar el botón guardar.</p> <p>8.-El sistema informará que los datos han sido guardados con éxito.</p> <p>9.- Fin del caso de uso: Los datos han sido almacenados.</p>
Flujos Alternativos o excepciones:	<p>8.- El sistema verificara que los datos obligatorios estén llenos, de no ser así el sistema te indicara cuales datos tienes que llenar para poder guardar el registro.</p> <p>8.- El sistema verificará que los datos no existan de ser así mostrará el mensaje "error al almacenar el registro".</p>
Postcondiciones:	Los datos están ahora almacenados en la base de datos.



CASO DE USO: Eliminar	
<pre> graph LR A[ADMINISTRADOR] --> UC((ELIMINAR)) B[SISTEMA DE BASE DE DATOS] --> UC </pre>	
Descripción:	Permite al administrador eliminar un empleado, estudiante o área, registrados en el sistema.
Actores:	Administrador
Precondiciones:	Los usuarios deben de ser autenticados en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1.- Inicia el caso de uso: El usuario deberá elegir el vinculo Administrar en el menú principal. 2.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para administrar. 3.- El usuario ahora elegirá la opción eliminar. 4.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para eliminar. 5.- El usuario ahora elegirá la opción eliminar empleado, estudiante o área según sea el caso. 6.- El sistema mostrará las diferentes opciones de búsqueda. 7.- El sistema mostrará todos los resultados de la búsqueda. 8.-El usuario seleccionará un registro del resultado. 9.- El sistema mostrará las opciones aceptar y cancelar. 10.-el usuario pulsará la opción aceptar. 11.-El sistema preguntará ¿Estás seguro de eliminar este registro? 12.- Fin del caso de uso: El actor elegirá eliminar o no el registro.
Flujos Alternativos o excepciones:	7.-El sistema desplegara el mensaje



	“no se tienen registros almacenados actualmente” en caso de no contar con los registros solicitados.
Postcondiciones:	Los datos ya no existen en la base de datos.

CASO DE USO: Editar	
<pre> graph LR A[ADMINISTRADOR] --> UC((EDITAR)) B[SISTEMA DE BASE DE DATOS] --> UC </pre>	
Descripción:	Permite al administrador editar los datos de un empleado, estudiante o área registrados en el sistema.
Actores:	Administrador
Precondiciones:	Los usuarios deben de ser autenticados en el sistema. Deben de existir registros en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1.- Inicia el caso de uso: El usuario deberá elegir el vinculo Administrar en el menú principal. 2.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para administrar. 3.- El usuario ahora elegirá la opción editar. 4.- El sistema mostrará otra pantalla en la cual se verán las diferentes opciones para editar. 5.- El usuario ahora elegirá la opción editar empleado, estudiante o área según sea el caso. 6.- El sistema mostrará las diferentes opciones de búsqueda. 7.- El sistema mostrará todos los resultados de la búsqueda. 8.-El usuario podrá seleccionar el resultado para ver todas sus características. 9.- El sistema mostrará el formulario con todos los datos correspondientes. 10.- El usuario podrá editar todos los



	campos que requiera, después deberá pulsar el botón guardar. 11.-El sistema informará que los datos han sido guardados con éxito. 12.- Fin del caso de uso: Los datos han sido almacenados
Flujos Alternativos o excepciones:	6.-El sistema desplegara el mensaje “no se tienen registros almacenados actualmente” en caso de no contar con los registros solicitados. 11.- El sistema verificara que los datos obligatorios estén llenos, de no ser así el sistema te indicara cuales datos tienes que llenar para poder guardar el registro.
Postcondiciones:	Los datos han sido editados correctamente.

2.2.- Diseño del sistema.

En este apartado nos enfocaremos al diseño de nuestra base de datos, el cual estará dividido en tres fases que son: el diseño conceptual, diseño lógico y diseño físico.

2.2.1.- Diseño conceptual.

En esta etapa se trata de identificar las entidades, relaciones y asociarlas con sus respectivos atributos, para posteriormente realizar el diagrama entidad relación del sistema.

2.2.1.1.-Entidades

A continuación se muestran en la tabla 2.1 las entidades identificadas en el sistema.

ENTIDAD	DESCRIPCION
EMPLEADO	Esta entidad se refiere al conjunto de trabajadores que laboran en la Vicerrectoría de Docencia, de los cuales se almacenarán todos sus datos personales, así como sus datos de acceso al sistema.



ESTUDIANTE	Esta entidad se refiere al conjunto de estudiantes que prestan Servicio Social y Prácticas Profesionales en la Vicerrectoría de Docencia, de los cuales se almacenarán todos sus datos personales, así como sus datos de acceso al sistema.
AREA	Esta entidad se refiere al conjunto de departamentos que conforman la Vicerrectoría de docencia.
CATEGORIA	Se refiere al conjunto de categorías que puede tener un empleado en la Vicerrectoría de docencia.
BITACORA	Se refiere a los registros de los horarios de entrada y salida de los usuarios del sistema.

Tabla 2.1.- Identificación de Entidades

2.2.1.2.-Relaciones

A continuación se observan en la tabla 2.2 las relaciones identificadas con respecto a las entidades que se muestran en la tabla 2.1.

ENTIDAD	RELACION	CONECTIVIDAD	ENTIDAD
EMPLEADO	PERTENECE	M:1	AREA
ESTUDIANTE	PERTENECE2	M:1	AREA
EMPLEADO	TIENE	M:1	CATEGORIA
EMPLEADO	REGISTRA	M:1	BITACORA
ESTUDIANTE	REGISTRA2	M:1	BITACORA

Tabla 2.2.- Identificación de Relaciones

2.2.1.3.-Atributos

Después de analizar las entidades y sus respectivas relaciones se define a continuación sus atributos en la tabla 2.3

1.	Empleado (ID_Emp , Tipo_Emp, Tipo_contrato, Nom foto, E-mail, huella, estado, Password)
2.	Estudiante (Matricula , Nom, Fac, foto, E-mail, huella, Tipo , Tipo_cont ,estado, Password)
3.	Area (ID_Area , Nom_Area, edificio, estado)
4.	Categoría (ID_Cat , Nom_Cat)
5.	Bitacora (ID_Bitacora , hora, bandera, Clave)

Tabla 2.3.- Identificación de atributos

2.2.1.4.-Diagrama Entidad-Relación



Planteado todo lo anterior pasamos al desarrollo del diagrama Entidad-Relación (Figura 2.15) de nuestro sistema.

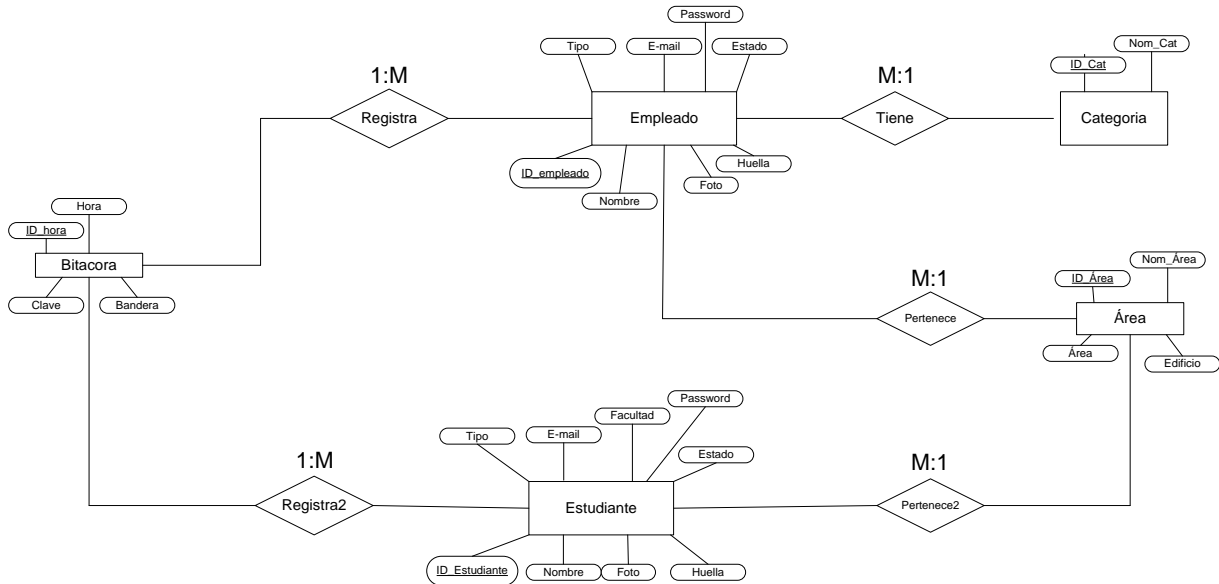


Figura 2.15.- Diagrama Entidad-Relación

2.2.2.-Diseño lógico

En esta etapa transformaremos el esquema conceptual a un esquema lógico, básicamente lo que haremos será convertir el diagrama entidad relación al modelo relacional, después aplicaremos las reglas de normalización de base de datos para evitar inconsistencias en nuestra base de datos.

Para llevar a cabo la transformación del modelo entidad-relación al modelo relacional se siguen las siguientes reglas:

1. Toda entidad se convierte en una tabla.
2. Toda relación **N: M** se genera una nueva tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación.
3. Toda relación **1: M** se incluye en la tabla **M** la llave primaria de la tabla **1** como llave foránea.

2.2.2.1.-Modelo Relacional

Con las reglas definidas anteriormente podemos definir nuestro esquema relacional, que como su nombre lo indica su estructura fundamental es una



relación es decir una tabla bidimensional formada por líneas (tuplas) y columnas (atributos). Las relaciones representan nuestras entidades, cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad. A continuación se muestra el modelo relacional del sistema (fig. 2.16).

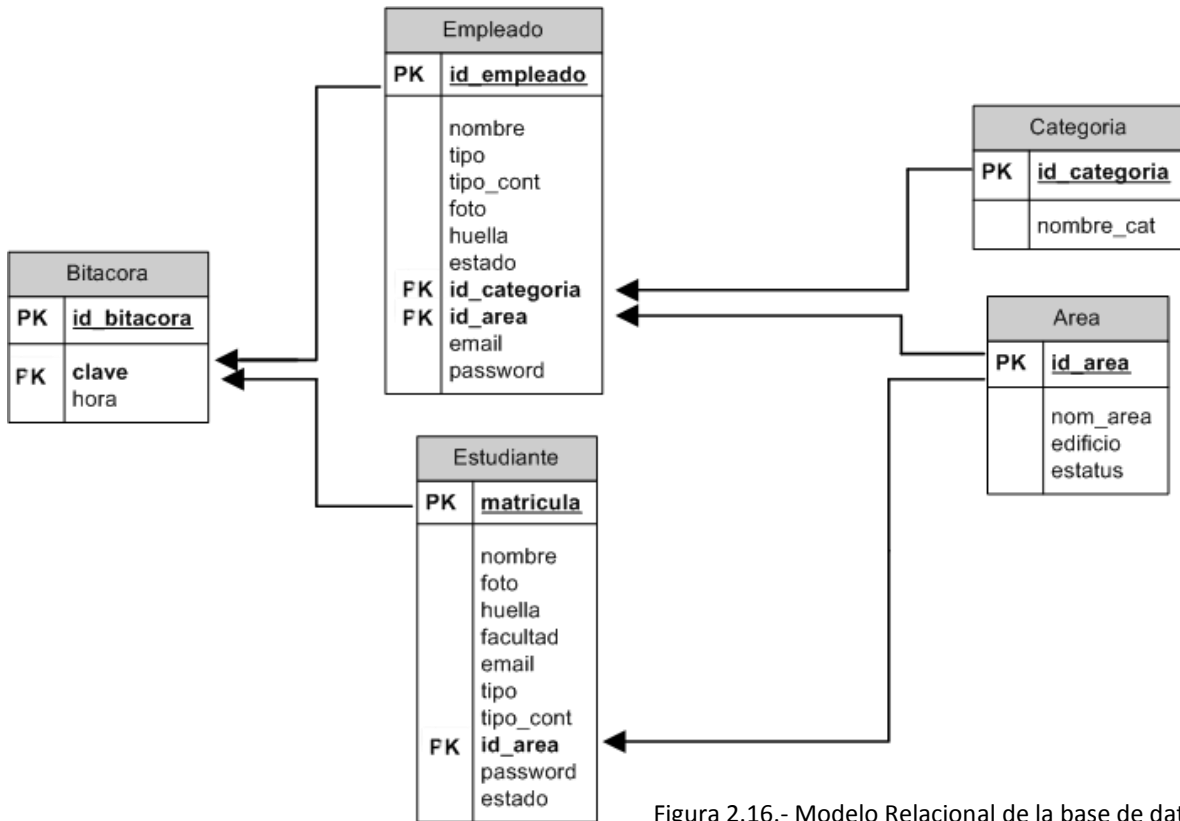


Figura 2.16.- Modelo Relacional de la base de datos

2.2.2.2.-Normalización de la Base de datos

Una vez que hemos obtenido el modelo relacional de nuestra base de datos es necesario que se le aplique la normalización. Esta consiste en mantener la estructura lógica de los datos en una forma limpia asegurando que las relaciones no tengan datos redundantes, para evitar los problemas que se puedan encontrar al momento de manipular la información contenida en la base de datos.

Primera forma normal. (1FN):

Se puede decir que una relación esta en primera forma normal, cuando en los identificadores o llaves primarias no se pueden tomar valores nulos, existan tuplas que se repitan, además de que sus valores deben de ser atómicos y del mismo tipo.



Por lo tanto aplicando la primera forma normal tenemos las siguientes tablas:

Tabla Área			
ID_Area	Nom_Area	Edificio	Estado

Tabla Bitácora			
ID_Bitacora	Hora	Bandera	Clave

Tabla Categoría	
ID_Cat	Nom_Cat

Tabla Empleado										
ID_Emp	Tipo	Tipo_Cont	Nombre	Foto	Huella	e_mail	Estado	Password	ID_cat	ID_area

Tabla Estudiante											
Matricula	Tipo	Tipo_cont	Nombre	Id_fac	Foto	Huella	Estado	Password	ID_area	e_mail	

Como en las tablas empleado y estudiante tenemos elementos multivaluados, como son los casos de Tipo_contrato, y facultad lo cual nos genera 2 nuevas tablas.

Tabla Contrato	
Tipo_cont	Nom_Cont

Tabla Facultad	
Id_fac	Facultad

Segunda forma normal. (2FN):

Para que una relación se encuentre en segunda forma normal es necesario que se cumpla con la 1FN y que todos los atributos que no son llaves dependan únicamente de la llave primaria, ya que cada tabla tiene un único identificador como llave primaria, lo cual se cumple para cada entidad que tenemos a continuación:

Bitacora (ID Bitacora PK, Hora, bandera, Clave FK)

ID Bitacora → Hora, bandera, Clave

Area (ID_Area PK, Nom_Area, edificio, estado)

ID Area → Nom_Area, edificio, estado



Categoría (ID_Cat **PK**, Nom_Cat, estado)

ID_Cat →, Nom_Cat, estado

Empleado (ID_Emp **PK**, Tipo, tipo_cont **FK**, Nombre, foto, E-mail, huella, estado, Password, ID_Cat **FK**, ID_Area **FK**)

ID_Emp → Tipo, Tipo_cont, Nombre, E-mail, huella, estado, Password, ID_Cat, ID_Area

Estudiante (Matricula **PK** Tipo, tipo_cont **FK**, estado, Nombre, id_fac **FK**, foto, E-mail, huella, Password, ID_Area **FK**)

Matricula → Tipo, tipo_cont, estado, Nombre, id_fac, foto, E-mail, huella, Password, ID_Area

Contrato (tipo_cont **PK**, Nom_cont)

Contrato → tipo_cont **PK**, Nom_cont

Facultad (id_fac **PK**, facultad)

Facultad → id_fac **PK**, facultad

Tercera forma normal. (3FN):

Una relación esta en tercera forma normal si, y sólo si, está en 2FN y, además no existen dependencias transitivas entre los atributos, se refiere a dependencias transitivas cuando existe más de una forma de llegar a referenciar a un atributo de una relación.

Como se pudo apreciar anteriormente todos y cada uno de los atributos son dependientes solo de la llave primaria, por tanto ya se encuentra en tercera forma normal.

2.2.3.- Diseño Físico

El propósito principal de esta etapa es el de producir una descripción de la implementación física de nuestra base de datos garantizando que se tenga un acceso eficiente a los datos, basándonos en el diseño lógico que anteriormente definimos.

Mientras que en el diseño lógico se especifica qué se guarda, en el diseño físico se especifica cómo se guarda. El diseño físico no es una etapa aislada, ya que algunas decisiones que se tomen durante su desarrollo pueden provocar una reestructuración del esquema lógico.



Para realizar esta etapa se debe de tener muy claro que sistema gestor de bases de datos es el adecuado para utilizar, ya que el esquema físico se debe de adaptar a este, para nuestro caso utilizaremos SQL Server 2005, ya que se adecua a nuestras necesidades.

En la figura 2.17 se muestra la construcción física de la base de datos, donde podemos observar las diferentes entidades con sus atributos y claves primarias.

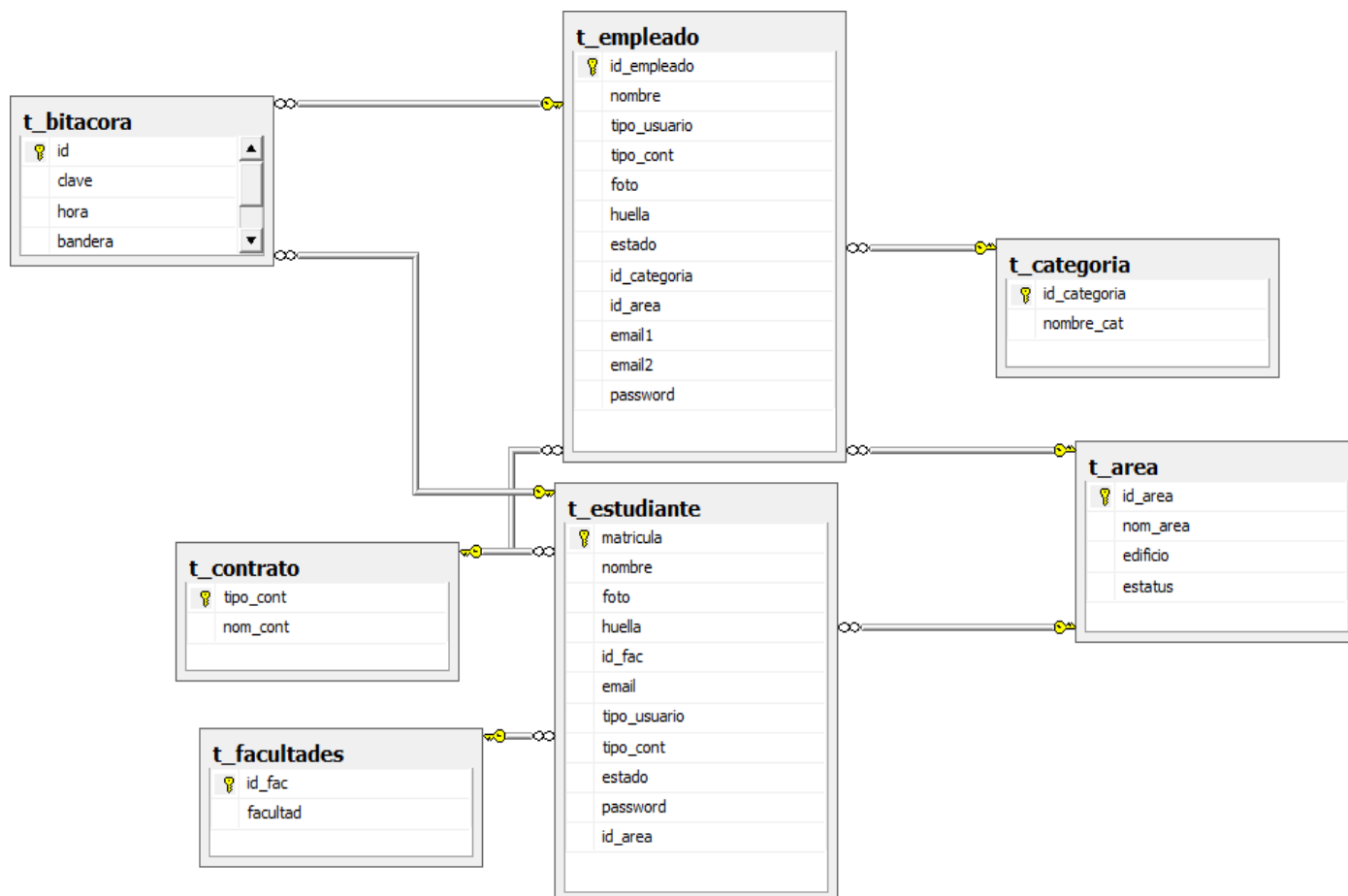


Figura 2.17.- Diseño físico de la base de datos de nuestro sistema

2.2.3.1.- Diccionario de datos

En el diccionario de datos se deben de establecer las tablas que conforman el sistema a desarrollar con sus respectivos atributos, haciendo una descripción



detallada de cada uno de ellos, el tipo de dato que los conforman, su tamaño, si es un atributo normal, llave primaria o foránea.

En la tabla 2.4 se muestra el diccionario de datos de nuestro sistema.

ATRIBUTO	DESCRIPCION	TIPO DE DATO	LONGITUD	LLAVE
1.-ENTIDAD EMPLEADO				
ID_Empleado	Especificamos el identificador del empleado.	Varchar	9	PK
Tipo	Especificamos el tipo de empleado.	Int		
Tipo_Cont	Especificamos el tipo de contratacion del emp.	int		FK
Nombre	Especificamos el nombre del empleado.	Varchar	50	
Foto	Especificamos la dirección donde se almacena la fotografía.	Varchar	50	
Huella	Especificamos la dirección donde se almacena la huella digital.	Varchar	50	
Estado	Especificamos el estado del empleado.	int		
Password	Especificamos el password del empleado.	Varchar	8	
ID_Cat	Especificamos el identificador de la categoría.	Varchar	3	FK
ID_Area	Especificamos el identificador del área.	Int		FK
e-mail	Especificamos la dirección de e-mail.	Varchar	50	
2.-ENTIDAD Estudiante				
Matricula	Especificamos el identificador del estudiante.	Varchar	9	PK
Tipo	Especificamos el tipo de estudiante.	Int		
Tipo_Cont	Especificamos el tipo de servicio del estudiante	int		FK
Id_fac	Especificamos el identificador de la facultad a la que pertenece.	Int		FK
Nombre	Especificamos el nombre del estudiante.	Varchar	50	
Foto	Especificamos la dirección donde se almacena la fotografía.	Varchar	50	
Huella	Especificamos la dirección donde se almacena la huella digital.	Varchar	50	
Estado	Especificamos el estado del empleado.	int		
Password	Especificamos el password del empleado.	Varchar	8	FK
ID_Area	Especificamos el identificador del área.	Int		
e-mail	Especificamos la dirección de e-mail..	Varchar	50	
3.-ENTIDAD Área				
ID_Area	Especificamos el identificador del área.	int		PK
Nom_Area	Especificamos el nombre del área.	Varchar	50	
Edificio	Especificamos el nombre del edificio.	Varchar	50	
Estado	Especificamos el estado del área	int		
4.-ENTIDAD Bitácora				
ID_Bitacora	Especificamos el identificador de la bitácora.	int		PK
Hora	Especificamos la hora de registro	date		
Bandera	Se utiliza para indicar si la hora registrada fue de entrada o salida.	int		
Clave	Especifica la clave del empleado o estudiante.	int		FK
5.-ENTIDAD Categoría				



CAPITULO 2.- ANÁLISIS Y DISEÑO DEL SISTEMA



ID_Cat	Especificamos el identificador de la categoría.	Varchar	3	PK
Nom_Cat	Especificamos el nombre de la categoría.	Varchar	20	
6.-ENTIDAD Contrato				
Tipo_cont	Especificamos el identificador del contrato.	int		PK
Contrato	Especificamos el tipo de contrato	Varchar	40	
6.-ENTIDAD Facultad				
Id_fac	Especificamos el identificador de la facultad.	int		PK
Facultad	Especificamos el nombre de la facultad	Varchar	40	

Tabla 2.4.- Diccionario de datos



3.- Implementación del sistema

En el presente capítulo es donde implementaremos el sistema apoyándonos con lo analizado en el capítulo anterior de esta tesis, utilizando las herramientas necesarias para ello.

Se ha decidido que como herramientas de desarrollo utilizaremos SQL server 2005 express para gestionar la base de datos, ya que el servidor donde quedará alojado el sistema tiene instalado un gestor de base de datos SQL Server 2005, además de C#, ASP.net y AJAX para desarrollo de nuestra interfaz, ya que nuestro sistema será implementado vía web, dichas herramientas nos otorgan la facilidad de trabajar con objetos permitiendo aprovechar al máximo las ventajas que esto además utilizaremos un lector biométrico marca ZKSoftware, modelo i680 para registrar la asistencias del personal, ya que nos ofrece las mejores características en comparación a otros lectores.

3.1.-Implementación de la base de datos.

Una vez mencionado lo anterior creamos nuestra base de datos mediante el siguiente script.

```
USE [master]
GO
/***** Objeto: Database [SAPVD] Fecha de la secuencia de comandos: 10/07/2009 12:06:48
*****/
CREATE DATABASE [SAPVD] ON PRIMARY
( NAME = N'SAPVD', FILENAME = N'c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\SAPVD.mdf' , SIZE = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH =
1024KB )

LOG ON
( NAME = N'SAPVD_log', FILENAME = N'c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\SAPVD_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH =
10%)

GO
EXEC dbo.sp_dbcmtlevel @dbname=N'SAPVD', @new_cmtlevel=90

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [SAPVD].[dbo].[sp_fulltext_database] @action = 'enable'
end
```

Como ya hemos creado la base de datos procedemos a la implementación de las tablas necesarias para el sistema.



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA



Para la creación de las tablas ejecutamos el siguiente script:

```

/*****Base de datos SAPVD *****/
/**Diseño Nicolás Zirahuén Gutiérrez Rangel***/
/*****Tablas SAPVD*****/

/***** SCRIPT TABLA T_AREA *****/
USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_area] Fecha de la secuencia de comandos: 12/14/2009
12:59:30 *****/
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[t_area]') AND
type in (N'U'))
DROP TABLE [dbo].[t_area]

USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_area] Fecha de la secuencia de comandos: 12/14/2009
12:59:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[t_area](
    [id_area] [int] IDENTITY(1,1) NOT NULL,
    [nom_area] [nvarchar](20) NOT NULL,
    [edificio] [nvarchar](20) NOT NULL,
    [estatus] [int] NOT NULL,
    CONSTRAINT [PK_t_area] PRIMARY KEY CLUSTERED
(
    [id_area] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

/***** SCRIPT TABLA T_AREA *****/

/***** SCRIPT TABLA T_CATEGORIA *****/
USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_categoria] Fecha de la secuencia de comandos: 12/14/2009
13:06:31 *****/
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[t_categoria]')
AND type in (N'U'))
DROP TABLE [dbo].[t_categoria]

USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_categoria] Fecha de la secuencia de comandos: 12/14/2009
13:07:11 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[t_categoria](
    [id_categoria] [int] IDENTITY(1,1) NOT NULL,
    [nombre_cat] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_t_categoria] PRIMARY KEY CLUSTERED
(
    [id_categoria] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

/***** SCRIPT TABLA T_CATEGORIA *****/

/***** SCRIPT TABLA T_EMPLEADO *****/
USE [SAPVD]
```



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA



```
GO
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_t_empleado_t_area]') AND parent_object_id =
OBJECT_ID(N'[dbo].[t_empleado]'))
ALTER TABLE [dbo].[t_empleado] DROP CONSTRAINT [FK_t_empleado_t_area]
GO
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_t_empleado_t_categoria]') AND parent_object_id =
OBJECT_ID(N'[dbo].[t_empleado]'))
ALTER TABLE [dbo].[t_empleado] DROP CONSTRAINT [FK_t_empleado_t_categoria]
GO
USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_empleado] Fecha de la secuencia de comandos: 12/14/2009
13:07:44 *****/
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[t_empleado]') AND
type in (N'U'))
DROP TABLE [dbo].[t_empleado]

USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_empleado] Fecha de la secuencia de comandos: 12/14/2009
13:08:06 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[t_empleado](
    [id_empleado] [int] NOT NULL,
    [nombre] [nvarchar](40) NOT NULL,
    [foto] [nvarchar](50) NULL,
    [huella] [nvarchar](50) NULL,
    [email1] [nvarchar](20) NULL,
    [email2] [nvarchar](20) NULL,
    [tipo_usuario] [int] NOT NULL,
    [tipo_cont] [int] NOT NULL,
    [estado] [int] NULL,
    [password] [nvarchar](8) NULL,
    [id_categoria] [int] NOT NULL,
    [id_area] [int] NOT NULL,
    CONSTRAINT [PK_t_empleado] PRIMARY KEY CLUSTERED
(
    [id_empleado] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[t_empleado] WITH CHECK ADD CONSTRAINT [FK_t_empleado_t_area] FOREIGN
KEY([id_area])
REFERENCES [dbo].[t_area] ([id_area])
GO
ALTER TABLE [dbo].[t_empleado] CHECK CONSTRAINT [FK_t_empleado_t_area]
GO
ALTER TABLE [dbo].[t_empleado] WITH CHECK ADD CONSTRAINT [FK_t_empleado_t_categoria]
FOREIGN KEY([id_categoria])
REFERENCES [dbo].[t_categoria] ([id_categoria])
GO
ALTER TABLE [dbo].[t_empleado] CHECK CONSTRAINT [FK_t_empleado_t_categoria]

/***** SCRIPT TABLA T_EMPLEADO *****/

/***** SCRIPT TABLA T_ESTUDIANTE *****/
USE [SAPVD]
GO
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_t_estudiante_t_area]') AND parent_object_id =
OBJECT_ID(N'[dbo].[t_estudiante]'))
ALTER TABLE [dbo].[t_estudiante] DROP CONSTRAINT [FK_t_estudiante_t_area]
```



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA



```
GO
USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_estudiante] Fecha de la secuencia de comandos: 12/14/2009
13:09:12 *****/
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[t_estudiante]')
AND type in (N'U'))
DROP TABLE [dbo].[t_estudiante]
USE [SAPVD]
GO

/***** Objeto: Table [dbo].[t_estudiante] Fecha de la secuencia de comandos: 12/14/2009
13:09:28 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[t_estudiante](
    [matricula] [int] NOT NULL,
    [nombre] [nvarchar](40) NOT NULL,
    [foto] [nvarchar](50) NULL,
    [huella] [nvarchar](50) NULL,
    [facultad] [nvarchar](30) NOT NULL,
    [email] [nvarchar](20) NULL,
    [tipo_usuario] [int] NOT NULL,
    [tipo_servicio] [int] NOT NULL,
    [estado] [int] NOT NULL,
    [password] [nvarchar](8) NOT NULL,
    [id_area] [int] NOT NULL,
    CONSTRAINT [PK_t_estudiante] PRIMARY KEY CLUSTERED
(
    [matricula] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[t_estudiante] WITH CHECK ADD CONSTRAINT [FK_t_estudiante_t_area]
FOREIGN KEY([id_area])
REFERENCES [dbo].[t_area] ([id_area])
GO
ALTER TABLE [dbo].[t_estudiante] CHECK CONSTRAINT [FK_t_estudiante_t_area]

/***** SCRIPT TABLA T_ESTUDIANTE *****/
/***** SCRIPT TABLA T_BITACORA *****/
USE [SAPVD]
GO
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_t_bitacora_t_employado]') AND parent_object_id =
OBJECT_ID(N'[dbo].[t_bitacora]'))
ALTER TABLE [dbo].[t_bitacora] DROP CONSTRAINT [FK_t_bitacora_t_employado]
GO
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_t_bitacora_t_estudiante]') AND parent_object_id =
OBJECT_ID(N'[dbo].[t_bitacora]'))
ALTER TABLE [dbo].[t_bitacora] DROP CONSTRAINT [FK_t_bitacora_t_estudiante]
GO
USE [SAPVD]
GO
/***** Objeto: Table [dbo].[t_bitacora] Fecha de la secuencia de comandos: 12/14/2009
13:10:18 *****/
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[t_bitacora]') AND
type in (N'U'))
DROP TABLE [dbo].[t_bitacora]

USE [SAPVD]
GO
```



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA

```

/***** Objeto: Table [dbo].[t_bitacora] Fecha de la secuencia de comandos: 12/14/2009
13:10:34 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[t_bitacora](
    [id_bitacora] [int] NULL,
    [hora] [datetime] NULL,
    [bandera] [int] NOT NULL,
    [clave] [nvarchar](9) NULL
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[t_bitacora] WITH CHECK ADD CONSTRAINT [FK_t_bitacora_t_empleado]
FOREIGN KEY([id_empleado])
REFERENCES [dbo].[t_empleado] ([id_empleado])
GO
ALTER TABLE [dbo].[t_bitacora] CHECK CONSTRAINT [FK_t_bitacora_t_empleado]
GO
ALTER TABLE [dbo].[t_bitacora] WITH CHECK ADD CONSTRAINT [FK_t_bitacora_t_estudiante]
FOREIGN KEY([matricula])
REFERENCES [dbo].[t_estudiante] ([matricula])
GO
ALTER TABLE [dbo].[t_bitacora] CHECK CONSTRAINT [FK_t_bitacora_t_estudiante]

/***** SCRIPT TABLA T_BITACORA *****/

```

En la figura 3.1 se puede observar un ejemplo de tabla ya implementada.

Nombre de columna	Tipo de datos	Permitir v...
id_empleado	nvarchar(9)	<input type="checkbox"/>
nombre	nvarchar(50)	<input type="checkbox"/>
tipo_usuario	int	<input type="checkbox"/>
tipo_cont	int	<input type="checkbox"/>
foto	nvarchar(50)	<input checked="" type="checkbox"/>
huella	nvarchar(50)	<input checked="" type="checkbox"/>
estado	int	<input type="checkbox"/>
id_categoria	nvarchar(3)	<input type="checkbox"/>
id_area	int	<input type="checkbox"/>
email1	nvarchar(40)	<input checked="" type="checkbox"/>
email2	nvarchar(40)	<input checked="" type="checkbox"/>
password	nvarchar(8)	<input type="checkbox"/>

Figura 3.1 Tabla Empleado implementada en SQL Server 2005

Como nuestro sistema será accedido a través de Internet es necesario programar una interfaz que sea visible desde la Web para que los usuarios interactúen con el sistema de base de datos por medio de un navegador, en la figura 3.2 podemos observar se realiza la interacción del sistema con nuestra base de datos.



```
namespace AccesoDatos
{
    public class historial : AccesoDatos.conexion
    {
        //Da recupera los datos del empleado
        public DataSet estadisticas(string user, string fi, string ff)
        {
            DataSet ds = new DataSet();
            DataSet resultado = new DataSet();
            try
            {
                System.Data.SqlClient.SqlConnection cn = new System.Data.SqlClient.SqlConnection();
                cn = obtenconexion();

                string SQL = "select p.clave ,nombre,area" +
                    " ,count(entrada) as Asistencias,sum(H.retardo) as retardos ,(dbo.DifDias('" + fi + " ', DATEADD (dd,1,'" + ff + "'))-count" +
                    " from vw_prueba p left join vw_Horas H on H.fecha2 =p.fecha2 " +
                    " where entrada='Entrada'and p.clave ='" + user + "' and H.hora >'" + fi + "' and H.hora < DATEADD (dd,1,'" + ff + " ')" +
                    " group by p.clave,nombre,area";

                SqlDataAdapter da = new SqlDataAdapter(SQL, cn);
                cn.Open();
                da.Fill(ds, "estadisticas");
                cn.Close();
                resultado = ds;
            }
            catch (Exception e)
            {
                Console.WriteLine("****se ha producido una excepcion del tipo:\n {0}", e.Message);
            }
            return resultado;
        }
        //Recupera datos
        public DataSet bitacora(string user, string fi, string ff)[...]
        public DataSet obtenestadisticas(string fi, string ff, string rep, string por, string opc)[...]
        public DataSet rangral(string fi, string ff, string rep, string por, string opc)[...]
    }
}
```

Figura 3.2.- Implementación de la interacción con la base de datos

3.2.-Implementación de la interfaz.

Como nuestra base de datos se ha implementado correctamente ahora estableceremos los parámetros para desarrollar nuestra interfaz.

El desarrollo de la interfaz del sistema está basado en los estándares Web 2.0 y es por medio de ella que los usuarios pueden comunicarse e interactuar. Las principales funciones de la interfaz Web son las siguientes:

- Control de acceso.
- Envío y almacenamiento de Información al servidor.
- Búsqueda, actualización de Información alojada en el servidor.

La Pantalla Inicial (Figura 3.3) permitirá a los usuarios autenticarse para poder acceder al sistema, el usuario deberá ingresar su nombre de usuario y password en los campos establecidos y después dar clic en el botón ingresar.



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA

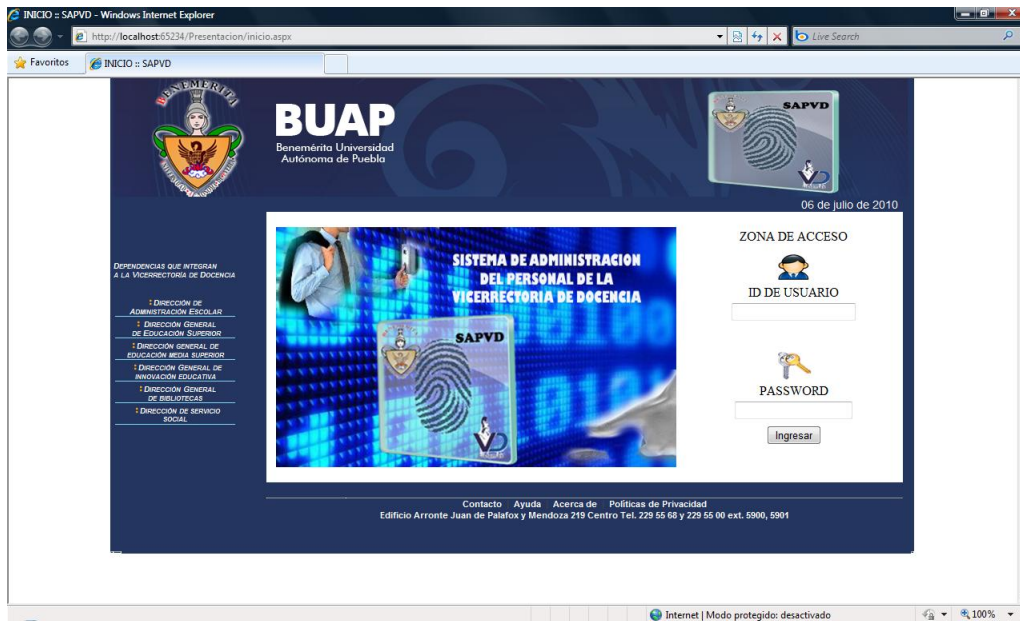


Figura 3.3 Página Principal

Una vez autenticado el usuario se accede a una nueva página (figura 3.4), en la parte izquierda se mostrarán, el nombre, fotografía e id del usuario logueado.



Figura 3.4 Página del Menú Principal

En la parte central se muestra un menú en el cual accederemos a las opciones principales de nuestro sistema que son:



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA

Consultar el historial; En este módulo (figura 3.5) el usuario logueado puede observar su progreso de asistencia.



Figura 3.5 Página Historial del usuario logueado.

Consultar Reportes; Módulo (figura 3.6) que permite obtener los registros necesarios para la emisión de reportes de asistencia, podrán acceder a este modulo los usuarios logueados como administrador y coordinador.



Figura 3.6 Página Reportes.



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA



Administrar; (figura 3.7) corresponde a los módulos donde se podrán agregar, modificar y eliminar datos de usuarios y de las diversas áreas de la dependencia, a los cuales solo un administrador tendrá acceso.



Figura 3.7 Página del Menú administrar

En esta pantalla podremos acceder a los diferentes formularios para agregar, editar o eliminar áreas, empleados o estudiantes. Cuyos formularios se muestran a continuación en las figuras 3.8-3.10.



Figura 3.8 Formulario para registrar o editar un área



CAPITULO 3.- IMPLEMENTACION DEL SISTEMA



ALTA EMPLEADO : SAPVD - Windows Internet Explorer
http://localhost:65234/Presntacion/Empleado.aspx

Benemérita Universidad Autónoma de Puebla
06 de julio de 2010

DEPENDENCIAS QUE INTEGRAN A LA VICERRECTORÍA DE DOCENCIA

- DIRECCIÓN DE ADMINISTRACIÓN ESCOLAR
- DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
- DIRECCIÓN GENERAL DE EDUCACIÓN MEDIA SUPERIOR
- DIRECCIÓN GENERAL DE INNOVACIÓN EDUCATIVA
- DIRECCIÓN GENERAL DE BIBLIOTECAS
- DIRECCIÓN DE SERVICIO SOCIAL

ID de Empleado:

FOTO:

Nombre del Empleado:

Tipo de Empleado: ADMINISTRADOR Tipo de Contrato: NOMINA INSTITUCIONAL

Categoría: VICE-RECTOR Área: ÁREA DE ESTADÍSTICA INSTITUCIONAL

Password: Verifica Password:

E-mail 1: E-mail 2:

Estatus: Activo

Figura 3.9 Formulario para registrar o editar un empleado

ALTA EMPLEADO : SAPVD - Windows Internet Explorer
http://localhost:65234/Presntacion/Empleado.aspx

Benemérita Universidad Autónoma de Puebla
06 de julio de 2010

DEPENDENCIAS QUE INTEGRAN A LA VICERRECTORÍA DE DOCENCIA

- DIRECCIÓN DE ADMINISTRACIÓN ESCOLAR
- DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
- DIRECCIÓN GENERAL DE EDUCACIÓN MEDIA SUPERIOR
- DIRECCIÓN GENERAL DE INNOVACIÓN EDUCATIVA
- DIRECCIÓN GENERAL DE BIBLIOTECAS
- DIRECCIÓN DE SERVICIO SOCIAL

ID de Empleado:

FOTO:

Nombre del Empleado:

Tipo de Empleado: ADMINISTRADOR Tipo de Contrato: NOMINA INSTITUCIONAL

Categoría: VICE-RECTOR Área: ÁREA DE ESTADÍSTICA INSTITUCIONAL

Password: Verifica Password:

E-mail 1: E-mail 2:

Estatus: Activo

Figura 3.10 Formulario para registrar o editar un estudiante



4.- PRUEBAS DEL SISTEMA

En este capítulo realizaremos las pruebas correspondientes para asegurarnos de que el sistema opera correctamente y cumple con las expectativas que se plantearon inicialmente.

4.1.- CASOS DE PRUEBA

A continuación se muestra una tabla con las pruebas realizadas al sistema:

# CASO	DESCRIPCIÓN	RELACION CON EL CASO DE USO	SALIDA ESPERADA	¿EL RESULTADO ES EL ESPERADO?
Caso #1	Verifica que el sistema no de acceso, si los campos necesarios para ingresar no son completados de manera correcta.	Autenticación de usuarios	Mensaje de Advertencia sobre campos que faltan por llenar y no se permite acceso al sistema.	Si.
Caso #2	Comprueba que el sistema no dará acceso a un usuario que no esté registrado en la base de datos del sistema.	Autenticación de usuarios	Mensaje de que el usuario no está registrado o la contraseña es incorrecta y no se permite acceso al sistema.	Si.
CASO #3	Corroborar que el sistema valida de manera correcta las claves de acceso y otorga los privilegios correspondientes a los usuarios logueados.	Autenticación de usuarios.	Acceso a la página del menú principal del sistema con los privilegios correspondientes al tipo de usuario logueado.	Si.
Caso #4	Verifica que el sistema no permite insertar registros si estos no han sido llenados en su totalidad.	Agregar	Advertencia de que hace falta llenar algún campo obligatorio.	Si.



CAPITULO 4.- PRUEBAS DEL SISTEMA



Caso #5	Verificar que los datos ingresados son almacenados en la base de datos.	Agregar.	Mensaje de que los datos fueron almacenados correctamente.	Si.
Caso #6	Verifica que los registros seleccionados son editados de manera correcta.	Editar.	Mensaje de que los registros se editaron correctamente.	Si.
Caso #7	Verifica que los registros se pueden eliminar de manera correcta.	Eliminar.	Mensaje de que los registros fueron eliminados correctamente.	Si.
Caso #8	Verifica que el sistema es capaz de emitir los reportes que le sean requeridos mediante los diferentes criterios de búsqueda que el sistema ofrece.	Consultar Reportes	Reportes con los datos almacenados en la base de datos debidamente filtrados por los criterios de búsqueda.	
Caso #9	Verifica que el sistema muestra el historial de los registros de un usuario en particular en un intervalo de fechas seleccionado.	Consultar historial.	Historial de las asistencias del usuario en el intervalo de fecha seleccionado.	Si.
Caso #10	Verifica que las asistencias son registradas correctamente en el lector biométrico.	Registrar asistencia	Mensaje en el display del lector biométrico que el registro ha sido correcto.	Si.
Caso #11	Verifica que las asistencias registradas por medio de la web son registradas correctamente.	Registrar asistencia	Mensaje de confirmación que el registros se almacenó satisfactoriamente.	Si.



CAPITULO 4.- PRUEBAS DEL SISTEMA



A continuación podemos observar el funcionamiento del sistema y su comportamiento al aplicarse los casos de prueba contenidos en la tabla anterior.

4.1.1.- INTERFAZ PRINCIPAL

En la pantalla principal el usuario se debe de autentificar para acceder al sistema, llenando los campos correspondientes, si el usuario deja algún campo en blanco, el sistema le indicara que debe de llenar el campo faltante como se puede observar en las figuras 4.1 y 4.2 (Caso de prueba #1).



Figura 4.1 Falta introducir id de usuario



Figura 4.2 Falta introducir password

El sistema revisará los datos proporcionados y los comparará con los que se encuentran almacenados en la base de datos, en caso de teclear un nombre de usuario no valido, el sistema mostrara un mensaje informando que el usuario no



está registrado en la base de datos o su contraseña es incorrecta como se muestra en la figura. 4.3 (Caso de prueba #2).

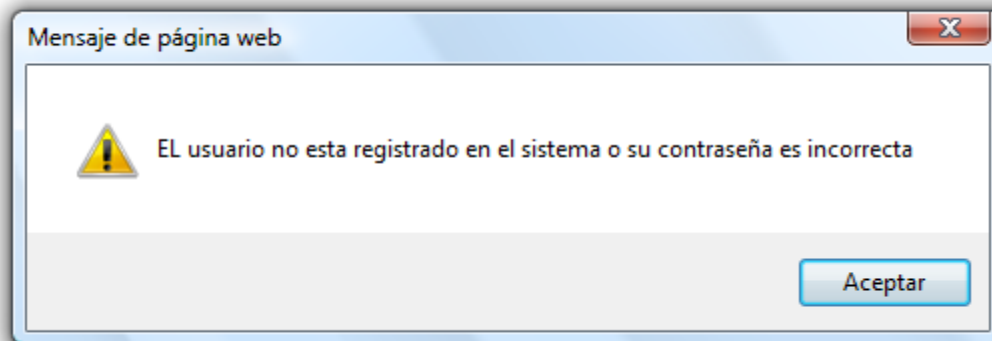


Figura 4.3 Mensaje de error al acceder al sistema.

4.1.2.- INTERFAZ MENÚ PRINCIPAL (Caso de prueba #3)

Si los datos que se proporcionaron en el paso anterior son correctos se podrá acceder al menú principal al que de acuerdo al tipo de usuario logueado se habilitarán los botones para realizar las operaciones correspondientes de acuerdo a los privilegios asignados al usuario.

En la figura 4.4 podemos observar el menú para un usuario logueado como administrador.



Figura 4.4 Menú principal del sistema para un Usuario con privilegios de Administrador.



CAPITULO 4.- PRUEBAS DEL SISTEMA



En la figura 4.5 podemos ver el menú para un usuario logueado como coordinador de área.



Figura 4.5 Menú principal del sistema para un Usuario con privilegios de Coordinador de área.

Y por ultimo si un usuario es logueado como un usuario común se mostrará el menú de la figura 4.6.

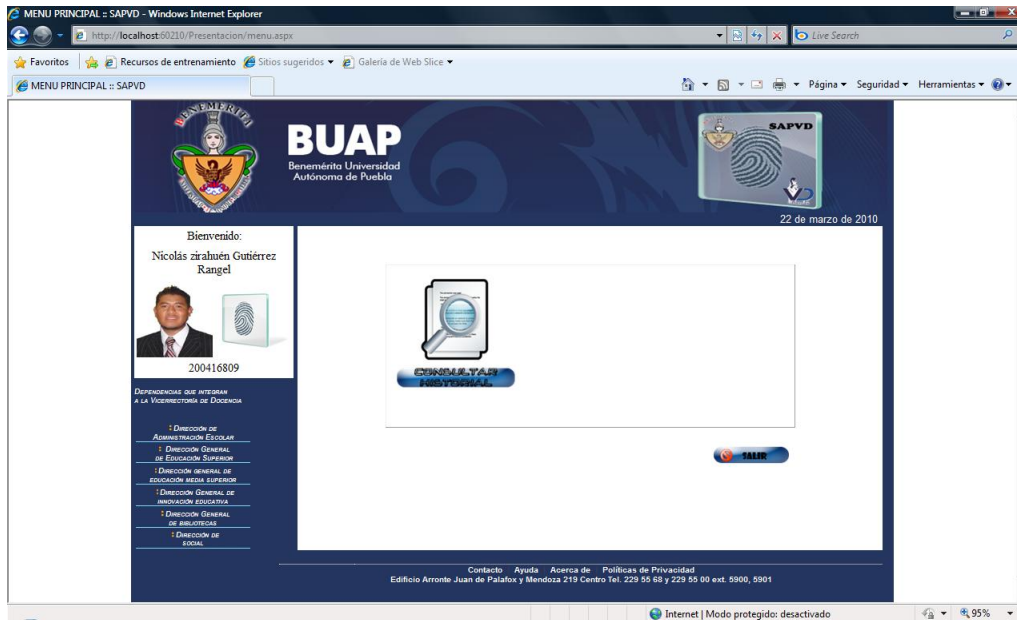


Figura 4.6 Menú principal del sistema para un Usuario común.



4.1.3.- INTERFAZ MENÚ ADMINISTRAR (solo para el administrador)

Esta opción solo es válida para un usuario logueado con privilegios de administrador, a través de esta pantalla el administrador será capaz de alimentar la base de datos, mediante las diferentes opciones que tiene permitido realizar dicho usuario, las cuales son agregar, editar, y eliminar registros ya sean de empleados, estudiantes o áreas.



Figura 4.7 Menú administrar del sistema.

4.1.3.1.- AGREGAR (Caso de prueba #4)

En la figura 4.7 se puede observar que el botón agregar esta instanciado tres veces ya que depende del tipo de registro que se quiera dar de alta, cada uno de ellos nos redirige a su formulario correspondiente el cual debe ser llenado de acuerdo a la elección hecha por el usuario.

En la figura 4.8 se muestra el formulario para dar de alta un área nueva.



CAPITULO 4.- PRUEBAS DEL SISTEMA

Figura 4.8 Formulario para dar de alta un área.

Ahora en la figura 4.9 podemos observar el formulario para dar de alta un empleado.

Figura 4.9 Formulario para dar de alta un Empleado.



CAPITULO 4.- PRUEBAS DEL SISTEMA



En la figura 4.10 podemos ver el formulario para dar de alta un estudiante nuevo.

Figura 4.10 Formulario para dar de alta un Estudiante.

El usuario deberá llenar los formularios, en caso de que el usuario olvide llenar algún campo el sistema le informará mostrándole una advertencia indicando que campo hace falta por cubrir (caso de prueba #5), a continuación se muestran algunos ejemplos (fig.4.11-4.13).

Figura 4.11 Advertencia indicando que hace falta llenar un campo en el formulario alta de área



CAPITULO 4.- PRUEBAS DEL SISTEMA



ALTA EMPLEADO - SAPVD - Windows Internet Explorer
http://localhost:60210/Presentacion/Empleado.aspx

ALTA EMPLEADO :: SAPVD

13 de abril de 2010

DEPENDENCIAS QUE INTERVIENEN A LA VICERRECTORÍA DE DOCENCIA

- 1 Dirección de ADMINISTRACIÓN ESCOLAR
- 1 Dirección General de Educación Superior
- 1 Dirección General de EDUCACIÓN MEDIA SUPERIOR
- 1 Dirección General de INNOVACIÓN EDUCATIVA
- 1 Dirección General de BIBLIOTECAS
- 1 Dirección de SOCIAL

ID de Empleado: 100497322

FOTO:

Nombre del Empleado: Javier Juarez Caballero

Tipo de Empleado: Tipo de Contrato:

Categoría: Área:

Password: *** Verifica Password: **

E-mail 1: 12312 E-mail 2: 21312312

Estatus:

Contacto Ayuda Acerca de Políticas de Privacidad
Edificio Arrote Juan de Palafox y Mendoza 219 Centro Tel. 229 55 68 y 229 55 00 ext. 5900, 5901

Internet | Modo protegido: desactivado

Figura 4.12 Advertencia indicando que hace falta llenar un campo en el formulario alta de Empleado

ALTA ESTUDIANTE - SAPVD - Windows Internet Explorer
http://localhost:60210/Presentacion/Estudiante.aspx

ALTA ESTUDIANTE :: SAPVD

13 de abril de 2010

DEPENDENCIAS QUE INTERVIENEN A LA VICERRECTORÍA DE DOCENCIA

- 1 Dirección de ADMINISTRACIÓN ESCOLAR
- 1 Dirección General de Educación Superior
- 1 Dirección General de EDUCACIÓN MEDIA SUPERIOR
- 1 Dirección General de INNOVACIÓN EDUCATIVA
- 1 Dirección General de BIBLIOTECAS
- 1 Dirección de SOCIAL

Matrícula: 200310077

Foto:

Nombre del Estudiante: Jose De Jesus Martinez Gonzalez

Tipo de Usuario: Tipo de Estudiante:

Facultad: Área:

Password: Verifica Password:

E-mail 1: Jesus_Martinez@hotmail.com

Estatus:

Contacto Ayuda Acerca de Políticas de Privacidad
Edificio Arrote Juan de Palafox y Mendoza 219 Centro Tel. 229 55 68 y 229 55 00 ext. 5900, 5901

Internet | Modo protegido: desactivado

Figura 4.13 Advertencia indicando que hace falta llenar un campo en el formulario alta de Estudiante

Al llenar el formulario correctamente en cualquiera de los casos de agregado antes mencionados y dar clic en el botón guardar, el sistema enviará los registros a la base de datos para que sean almacenados, si no existe ningún problema nos mostrará un mensaje el cual indica si el registro fue almacenado de manera correcta (fig.4.14) o no (fig. 4.15)



CAPITULO 4.- PRUEBAS DEL SISTEMA

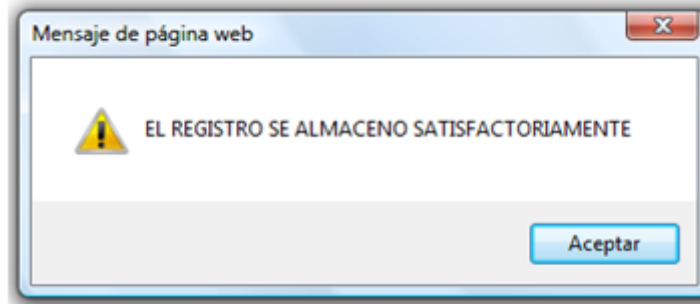


Figura 4.14 Alerta indicando que se almacenó el registro en la base de datos

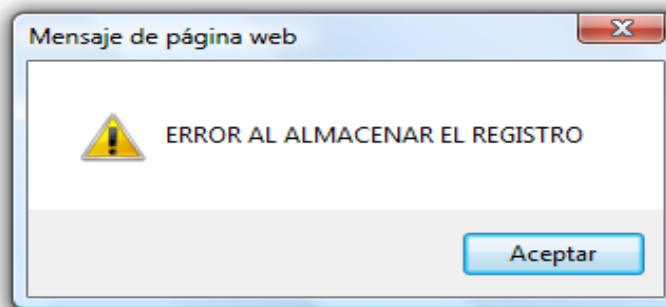


Figura 4.15 Alerta indicando que no se almacenó el registro en la base de datos

4.1.3.2.-EDITAR (Caso de prueba #6)

Al igual que la opción anterior, el botón editar se encuentra para los tres diferentes casos y cada uno redirige a una página diferente dependiendo del tipo de selección que el usuario realice (fig. 4.16-4.18), las cuales mostrarán un listado de todos los registros almacenados en la base de datos.



Figura 4.16 Página de búsqueda de empleados



Si seleccionamos editar empleado desplegará la página de búsqueda de empleados, en la que podremos realizar una búsqueda mas especifica del registro indicando lo parámetros a filtrar.

Si queremos realizar una búsqueda por id de empleado es necesario seleccionar en el combo además de proporcionar el id en el campo de texto y luego dar clic en el botón buscar.

FILTRAR POR: ID EMPLEADO:

- ID DE EMPLEADO
- NOMBRE
- ÁREA
- CATEGORIA

En el caso de que se conozca el nombre del empleado se puede buscar haciendo selección en filtrar por nombre y proporcionar el nombre en el campo de texto.

FILTRAR POR: NOMBRE :

- NOMBRE
- ID DE EMPLEADO
- NOMBRE
- ÁREA
- CATEGORIA

Si deseamos consultar a los empleados de una determinada área o categoría solo basta con seleccionar en el combo de filtrar por área o por categoría y el área o categoría deseada, y si se quiere una búsqueda mas especifica se puede insertar el id del empleado en el campo de texto.

FILTRAR POR: ID EMPLEADO:

ÁREA:

FILTRAR POR: ID EMPLEADO:

CATEGORIA:

Si seleccionamos editar Estudiante veremos la página de búsqueda de estudiantes (figura 4.17).



CAPITULO 4.- PRUEBAS DEL SISTEMA

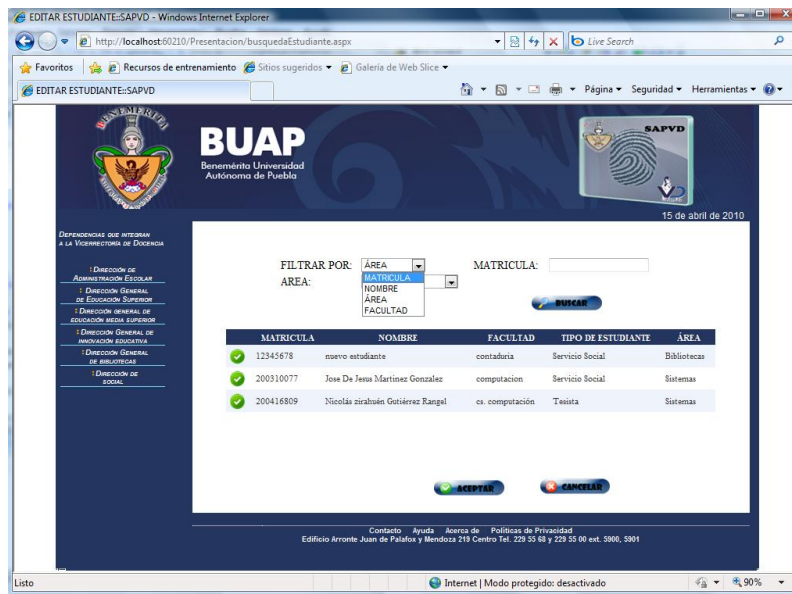


Figura 4.17 Página de búsqueda de Estudiantes

Para este caso también podremos realizar una búsqueda mas especifica del registro indicando lo parámetros a filtrar, los cuales pueden ser los siguientes:

Por matricula del estudiante.

FILTRAR POR: MATRICULA:

Por nombre del estudiante.

FILTRAR POR: NOMBRE:

Por área.

FILTRAR POR: MATRICULA:

ÁREA:



CAPITULO 4.- PRUEBAS DEL SISTEMA



Y por facultad.

FILTRAR POR: MATRICULA:

FACULTAD :

Si seleccionamos editar Área veremos la página de búsqueda de áreas (figura 4.18).

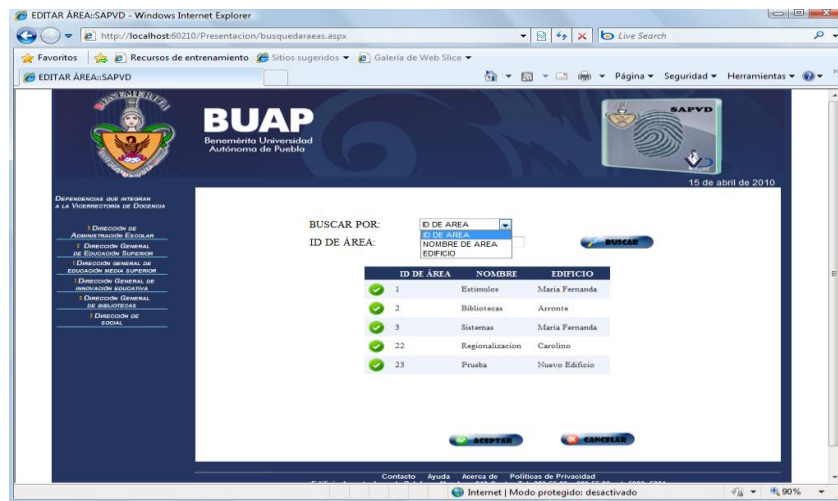


Figura 4.18 Página de búsqueda de áreas

Como en los casos anteriores también podremos realizar una búsqueda mas especifica del registro indicando lo parámetros a filtrar, los cuales pueden ser los siguientes:

BUSCAR POR:

ID DE ÁREA:

BUSCAR POR:

NOMBRE DE ÁREA:

BUSCAR POR:

EDIFICIO:



CAPITULO 4.- PRUEBAS DEL SISTEMA



Para todas las páginas de búsqueda el funcionamiento es el mismo, una vez obtenido el listado, el usuario deberá de seleccionar el registro que desea editar (fig. 4.19) y este se mostrará en el formulario que le corresponde, con todos los campos llenos con respecto a los datos almacenados en la base de datos para que el usuario edite o solo corrobore los registros (fig. 4.20).

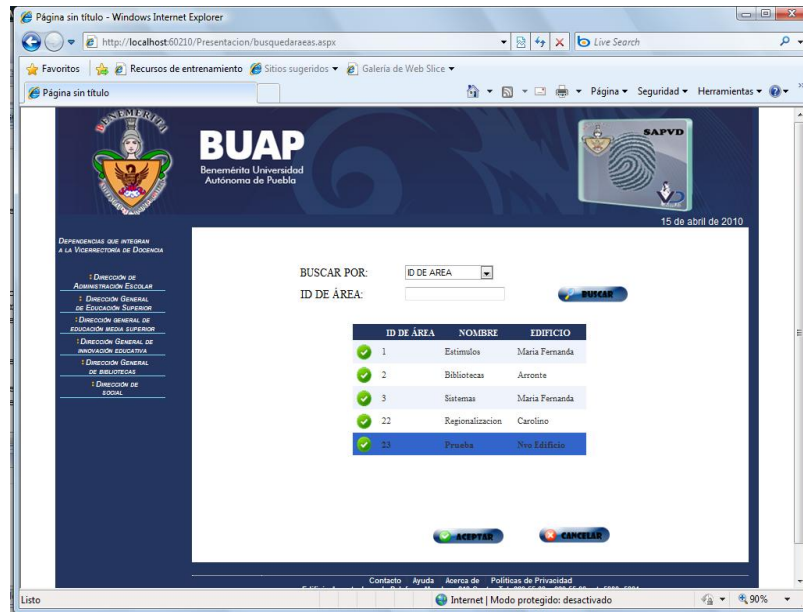


Figura 4.19 Registro seleccionado

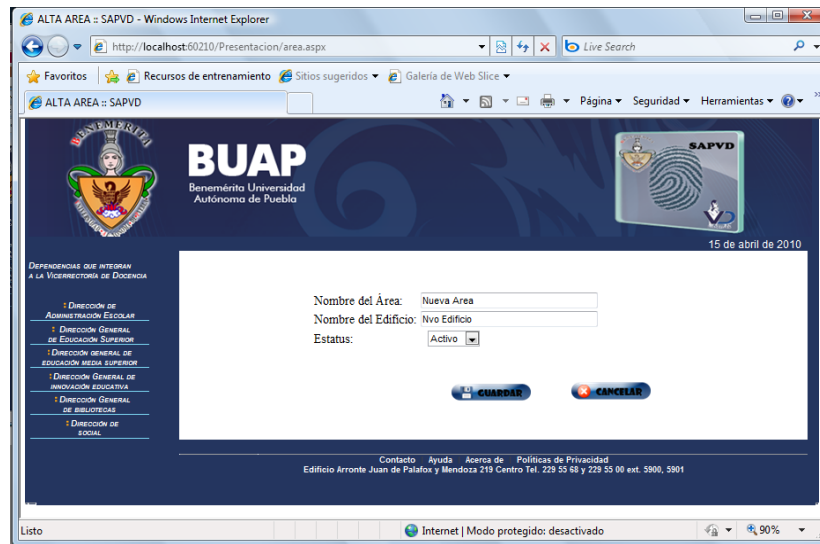


Figura 4.20 Formulario con los campos a editar del campo

Si la edición se realiza de manera correcta se mostrará una advertencia indicando que el registro fue editado satisfactoriamente (fig.4.21).



CAPITULO 4.- PRUEBAS DEL SISTEMA

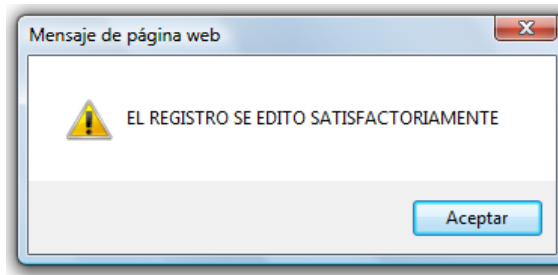


Figura 4.21 Alerta de edicion de registro

En caso de existir un error al editar el registro el sistema lo informará por medio de una advertencia (fig. 4.22).

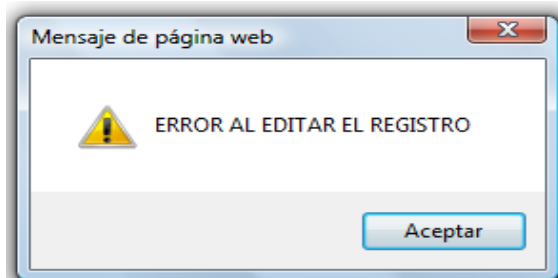


Figura 4.22 Alerta de error al editar el registro

4.1.3.3.-ELIMINAR (Caso de prueba #7)

Como en las opciones anteriores ahora tenemos tres instancias del botón eliminar y cada uno redirige a una página diferente dependiendo del tipo de selección que el usuario realice (figs. 4.16-4.18) las cuales mostrarán un listado de todos los registros almacenados en la base de datos, con los mismos criterios de filtrado mencionados en el apartado anterior dependiendo del tipo de registro que se desee buscar.

Una vez seleccionado el registro deseado se mostrará una alerta para confirmar si se desea eliminar el registro (fig. 4.23).



CAPITULO 4.- PRUEBAS DEL SISTEMA

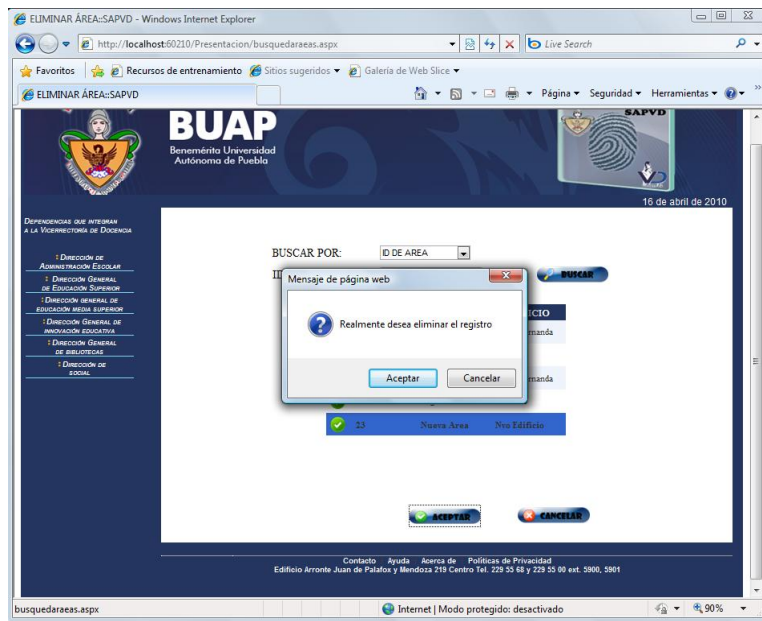


Figura 4.23 confirmación de eliminación de registro

Si se procede a la eliminación del registro se muestra de nuevo el listado donde se observa que el registro ha sido eliminado con éxito (fig.4.24).

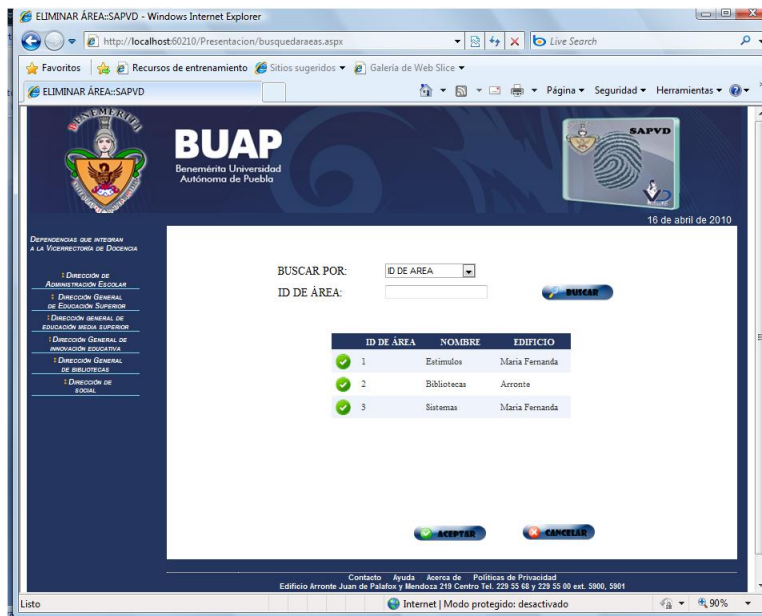


Figura 4.24 Listado con registro eliminado.



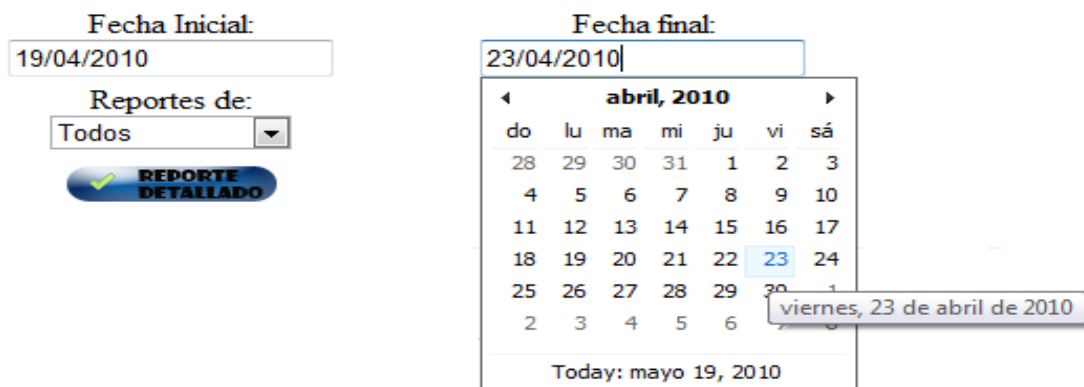
4.1.4.- INTERFAZ CONSULTAR REPORTES (solo para el administrador y coordinador de área) (Caso de prueba #8)

Otra de las opciones del menú (figuras 4.4 y 4.5 según los privilegios del usuario) principal es la de consultar reportes, si el usuario decide esta elección se mostrará una página en la que podremos generar reportes de asistencia según sean requeridos (figura 4.25).



Figura 4.25 Página consultar Reportes.

Para generar los reportes antes que nada se debe de seleccionar el intervalo de fecha en el cual se desea obtener el reporte esto se realiza al dar clic en los campos de texto etiquetados como “Fecha Inicial” o “Fecha Final”, y se desplegara un calendario en el cual se debe seleccionar el rango de días que se desean mostrar en el informe.





Una vez elegido el intervalo de fecha debemos elegir sobre que personal será el informe, por ejemplo todos,

Fecha Inicial: 19/04/2010 Fecha final: 23/04/2010

Reportes de:
 Todos
 Empleados
 Estudiantes

Por:
 Todo

REPORTE GENERAL

Por empleados,

Fecha Inicial: 19/04/2010 Fecha final: 23/04/2010

Reportes de:
 Empleados

Por:
 Tipo
 Todo
 Area
 Tipo
 Categoría

Tipo:
 Nomina

REPORTE DETALLADO

O por estudiantes.

Fecha Inicial: 19/04/2010 Fecha final: 23/04/2010

Reportes de:
 Estudiantes

Por:
 Tipo
 Todo
 Area
 Tipo

Tipo:
 Servicio social

REPORTE DETALLADO

Una vez definidos los parámetros de búsqueda tenemos 2 opciones de reportes: Reporte general o Reporte detallado.



Al elegir un reporte General se estará generando un informe en el cual se muestra a grandes rasgos el número de asistencias, inasistencias, horas laboradas y retardos empleados y estudiantes que estén dentro de los parámetros de búsqueda seleccionados anteriormente, así como la cantidad en efectivo que se les tendrá que pagar en caso de ser personal temporal de la institución, un ejemplo de reporte general se muestra en la figura 4.26.



1 / 1 Informe principal 100% Business Objects

BENEMERITA 19/05/2010

Reporte General

Registros

Del día: Lunes 19/04/2010 Hasta el día: Sábado 24/04/2010

Clave: 100497322						
Nombre	Área	Categoría	Asistencias	Retardos	Faltas	Horas laboradas
Javier Juarez Caballero	Sistemas	Tiempo completo	5	1	0	45
Total a pagar						\$ 0.00
Clave: 12345678						
Nombre	Área	Categoría	Asistencias	Retardos	Faltas	Horas laboradas
Manuel Perez gomez	Bibliotecas	Estudiante	5	2	0	43
Total a pagar						\$ 860.00
Clave: 200310077						
Nombre	Área	Categoría	Asistencias	Retardos	Faltas	Horas laboradas
Jose De Jesus Martinez Gonzalez	Sistemas	Estudiante	4	2	1	34
Total a pagar						\$ 680.00
Clave: 200416809						
Nombre	Área	Categoría	Asistencias	Retardos	Faltas	Horas laboradas
Nicolás zirahuén Gutiérrez Rangel	Sistemas	Estudiante	5	1	0	42
Total a pagar						\$ 840.00

Figura 4.26 Ejemplo de un Reporte General.



En el caso de elegir un reporte detallado , se genera un reporte en el cual además de incluir los datos del reporte general, se incluyen las fechas y horarios de los registros de entrada y salida del trabajador como se puede apreciar en la figura 4.27.



CAPITULO 4.- PRUEBAS DEL SISTEMA



1 / 1+ Informe principal

19/05/2010



Reporte Detallado

Registros					
Del día:		Lunes 19/04/2010		Hasta el día: Sábado 24/04/2010	
Clave:	100457322				
Nombre:	Javier Juarez Caballero				
Área:	Sistemas				
Categoría:	Tiempo completo				
Tipo:	Por Honorarios				
Fecha	Hora de Entrada	Hora de Salida	Retardos	Horas Laboradas	
Lunes 19-04-10	08:57:01	17:50:04	0	9	
Martes 20-04-10	08:57:01	17:57:01	0	9	
Miércoles 21-04-10	09:57:01	15:41:05	1	6	
Jueves 22-04-10	08:55:01	19:55:01	0	11	
Viernes 23-04-10	09:00:01	18:50:01	0	10	
Asistencias:	Faltas:	Retardos:	Horas Totales	Total a pagar	
5	0	1	45		
Clave:	12345678				
Nombre:	Manuel Perez gomez				
Área:	Bibliotecas				
Categoría:	Estudiante				
Tipo:	Servicio Social				
Fecha	Hora de Entrada	Hora de Salida	Retardos	Horas Laboradas	
Lunes 19-04-10	08:59:01	18:00:01	0	9	
Martes 20-04-10	09:17:01	18:17:01	1	9	
Miércoles 21-04-10	08:56:14	17:56:14	0	9	
Jueves 22-04-10	09:55:01	16:55:01	1	7	
Viernes 23-04-10	09:02:01	17:50:01	0	9	
Asistencias:	Faltas:	Retardos:	Horas Totales	Total a pagar	
5	0	2	43	\$ 960.00	
Clave:	200310077				
Nombre:	Jose De Jesus Martinez Gonzalez				
Área:	Sistemas				
Categoría:	Estudiante				
Tipo:	Servicio Social				
Fecha	Hora de Entrada	Hora de Salida	Retardos	Horas Laboradas	
Lunes 19-04-10	08:57:01	17:50:04	0	9	
Martes 20-04-10	09:07:01	18:07:01	0	9	
Miércoles 21-04-10	09:16:14	18:16:14	1	9	
Jueves 22-04-10	09:15:01	16:15:01	1	7	
Asistencias:	Faltas:	Retardos:	Horas Totales	Total a pagar	
4	1	2	34	\$ 680.00	

Figura 4.27 Ejemplo de un Reporte Detallado.



CAPITULO 4.- PRUEBAS DEL SISTEMA

4.1.5.- INTERFAZ CONSULTAR HISTORIAL (todos los usuarios) (Caso de prueba #9)

Al dar clic en consultar historial podremos observar el progreso en su asistencia y puntualidad del usuario logueado, por defecto se mostrará solo los registros del día en que se ha logueado (fig. 4.28).

Historial de asistencia para el usuario Nicolás zirahuén Gutiérrez Rangel (ID: 200416809) el día 19 de mayo de 2010.

Fecha	Entrada	Salida	Retardo	Horas Laboradas
Lunes 19-05-10	08:39:01	18:00:04	0	9

Clave	Nombre	Área	Retardos	Asistencias	Faltas	Horas Laboradas
200416809	Nicolás zirahuén Gutiérrez Rangel	Sistemas	0	1	0	9

Figura 4.28 Historial de usuario logueado.

Si el usuario desea consultar sus registros anteriores solo debe hacer clic en los campos de texto y se desplegará un calendario donde debe seleccionar el intervalo de fechas de los registros que desea consultar (Caso de prueba #13).

Historial de asistencia para el usuario Nicolás zirahuén Gutiérrez Rangel (ID: 200416809) el día 18/04/2010.

Intervalo de fechas: DE: 18/04/2010 HASTA: 24/04/2010

Una vez elegido el intervalo damos clic en buscar y se desplegará los registros en nuestra página como se muestra en la figura 4.29.



Figura 4.29 Historial de usuario logueado.

4.1.6.- REGISTRO DE ASISTENCIA (Caso de prueba #10)

Para el registro de asistencia es necesario que los usuarios estén registrados previamente en el sistema y en el lector.

Cuando el usuario desee registrar su entrada o salida el usuario debe seleccionar en el menú del dispositivo si su registro es de entrada o salida (figs. 4.31-4.32) y después solo tiene que colocar su dedo (que deberá ser registrado previamente en el dispositivo por un administrador, por lo regular el dedo índice) sobre el sensor del aparato para que este haga la digitalización (fig. 4.33) y se realice la autenticación del usuario en cuestión.



Figura 4.30 Lector en estado normal.



Figura 4.31 selección de registro de entrada.



CAPITULO 4.- PRUEBAS DEL SISTEMA



Figura 4.32 selección de registro de salida.



Figura 4.33 Registro de asistencia.

Si la huella no se encuentra registrada o se digitalizó de manera errónea el dispositivo emite un mensaje en su pantalla solicitando que intente nuevamente la verificación, tal y como se observa en la figura 4.34.



Figura 4.34 Mensaje de verificación incorrecta en el dispositivo .

Si la verificación es correcta entonces en el display del reloj se mostrarán los datos del usuario autenticado indicando que el registro es correcto, como se observa en la figura 4.35.



Figura 4.35 Verificación correcta.



El almacenamiento de los registros a la base de datos se hace por medio de una aplicación intermedia, la cual se ejecuta cada determinado tiempo, esto para evitar la pérdida de información o que se produzca un retraso al momento de registrar la entrada o salida del usuario en caso de que existan problemas de conexión a la red.

En caso de que exista alguna contingencia al momento de ingresar asistencia por medio del dispositivo biométrico, existe una alternativa dentro del sistema a la que solo un usuario con privilegios de administrador o coordinador de área tendrá acceso, que es la de registrar asistencia por medio de la web mediante un apartado en las opciones del menú principal del sistema (figura 4.36, Caso de prueba #11)

Figura 4.36 Apartado para registrar asistencia por medio de la web.

En donde el usuario introducirá el id de usuario que se desea registrar su asistencia y en el combo deberá elegir si el registro es de entrada o salida (figura 4.37).

Figura 4.37 Llenado del apartado

Y al dar clic en registrar asistencia se mostrara una ventana de confirmación (figura 4.38).



CAPITULO 4.- PRUEBAS DEL SISTEMA

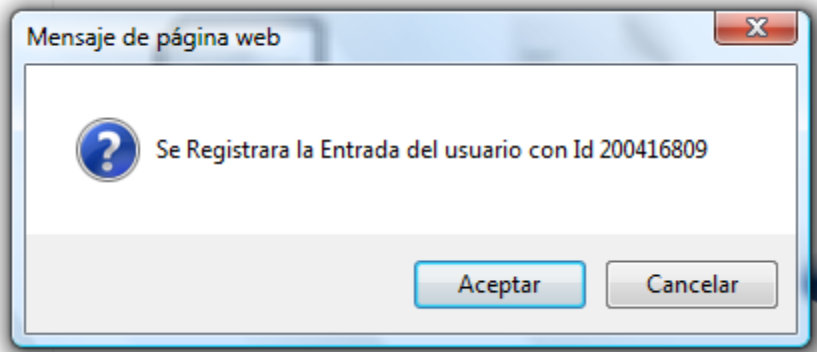


Figura 4.38 ventana de confirmación

Si se confirma que el id introducido es válido mostrara un mensaje con lo siguiente (figura 4.39):

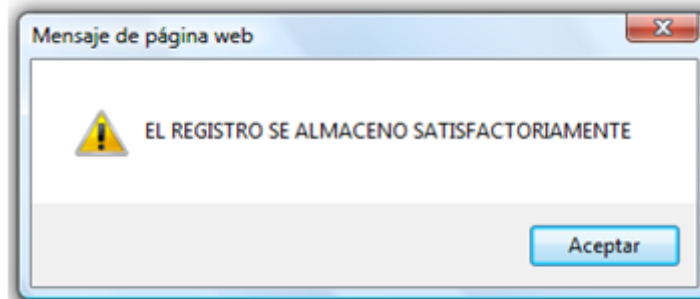


Figura 4.39 ventana de confirmación exitosa

En caso contrario se mostrará (figura 4.40):

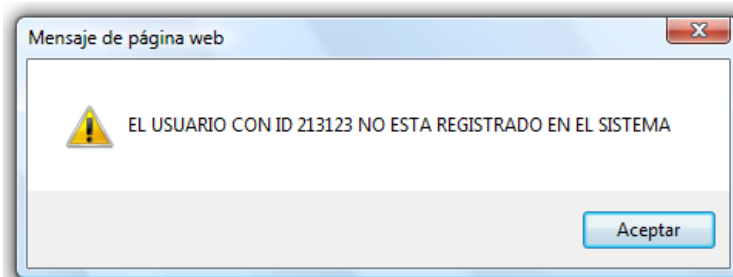


Figura 4.40 ventana de confirmación fallida



CONCLUSIONES.-

El objetivo principal del presente trabajo se cumple satisfactoriamente gracias a las metodologías y herramientas utilizadas que nos han permitido desarrollar el Sistema De Administración Del Personal de La Vicerrectoría de Docencia (SAPVD).

Para el desarrollo del sistema se usaron conceptos y metodologías de la ingeniería de software, empleando específicamente la metodología de cascada, ya que nos permitió ir desarrollando específicamente cada una de las etapas para la culminación del proyecto.

Durante el proceso de análisis se usaron diagramas de flujo de datos y de casos de usos que nos permitieron tener una perspectiva más amplia de lo que era requerido para el correcto funcionamiento del sistema, además de permitirnos identificar los actores que interactúan en el sistema y los módulos que se debían de implementar.

Dentro del análisis también se obtuvieron las entidades para diseñar el diagrama entidad-relación y así implementar la base de datos con el modelo relacional, lo cual facilita que en un futuro se le puedan agregar módulos para que el sistema pueda crecer.

El sistema fue implementado con un manejador de bases de datos Sql server 2005 Express, un servidor web internet information services 7, y con lenguaje asp.net y c#.

Tratando de que la interfaz sea amigable y entendible para cualquier usuario y que le facilite sus actividades.

El sistema fue sometido a verificación y validación mediante casos de prueba con la finalidad de evaluar su funcionalidad y usabilidad.

A causa de lo anterior hemos desarrollado un sistema capaz de automatizar el proceso de registro de asistencias, y la manipulación de los mismos mediante un explorador web, además de emitir los informes correspondientes que sean necesarios.

Se puede decir que el sistema en cuanto al registro de asistencias, asegura que los empleados de la dependencia son en realidad los que registran su entrada o salida y no otra persona, además de impulsarlos a cumplir con más responsabilidad sus jornadas de trabajo.



Nos permite estandarizar el conteo de las horas laboradas para todas las áreas, y al ser el sistema por medio de la web el usuario puede ver su progreso en cuanto a asistencia, administrar los datos de usuarios o áreas, así como la emisión de reportes, dependiendo de sus privilegios de acuerdo al tipo de usuario, todo esto desde cualquier computadora con conexión a internet, haciendo más fácil las actividades de los encargados de cada área.

Es por ello que se concluye que el objetivo primordial del presente trabajo ha sido cumplido.

El sistema está siendo implementado de manera experimental, ya que en realidad solo es utilizado para llevar un control informal de las asistencias de los empleados y estudiantes, es decir no implica ninguna sanción para los trabajadores, cumpliendo con los objetivos pensados para la primera fase de un proyecto institucional.

LIMITACIONES.-

El sistema inicialmente estaba pensado para que todo fuera controlado en el explorador web, pero a causa de que las librerías para manipular el lector biométrico no son compatibles para la web, fue necesario utilizar una aplicación intermedia de escritorio para hacer la recolección de los registros del lector e insertarlos a la base de datos.

Otra limitante es que no se abarcan varios aspectos como por ejemplo el manejo de días feriados, justificaciones, vacaciones y horas de comida, ya que la primera fase del proyecto institucional solo contempla el registro de asistencias y emisión de reportes de las mismas.

El sistema puede no estar exento de errores, por esto los encargados del área de sistemas de la Vicerrectoría de docencia llevarán un seguimiento del comportamiento del sistema con el fin de corregir los que se presenten.

PERSPECTIVAS.-

Cabe mencionar que el proyecto realizado en un principio surgió como un proyecto experimental, es decir solo sería implementado en la Vicerrectoría de docencia y las dependencias que la conforman, una de las perspectivas del sistema es que en un futuro se pueda implementar el proyecto a nivel institucional.

La segunda fase del proyecto consta de lo siguiente:



CONCLUSIONES, LIMITACIONES Y PERSPECTIVAS



- Definición de horarios vinculados con el reglamento institucional.
- Definición de vacaciones y días feriados.
- Definición de horas de comida.
- Justificación de asistencias y retardos.
- Vincular con salarios de nomina institucional.
- Vincular a todas las dependencias de la institución.
- Debido a su futuro crecimiento poder migrar a un sistema de base de datos como Oracle.
- Revisión del diseño de la base de datos para optimizar el rendimiento del sistema.

**BIBLIOGRAFIA**

- [1] RAND Corporation, “Biometrics offers advantages and controversy”, RAND Corporation. [En Línea]
Disponible: http://www.rand.org/natsec_area/products/biometrics.html
- [2] SisAFIS, Huella, Biometría, CCTV, Control de Acceso y reconocimiento de huellas digitales”, 2008 [En línea]. Disponible: <http://www.ccc2003rl.com/SisAfis.htm>
- [3] Galbally Herrero, Javier, “Interoperabilidad, Usabilidad Y Privacidad En Sistemas Biométricos Multimodales”, Memoria Trabajo Tutelado
- [4] Biometrics catalog “Biometrics glossary,” Biometrics Catalog, 2006. [En línea]. Disponible: <http://www.biometriccatalog.org/biometrics/GlossaryDec2005.pdf>
- [5] TECNOBIO, Tecnologías Biométricas de Identificación, TLF:981 360 512-tecnobio-id@wanadoo.es
- [6] Flores Armenta, Luis Alberto, “Reconocimiento De Huellas Dactilares”, Instituto Tecnológico De Nogales
- [7] Rojas Campos Liliana “Avances de IHC: Biometría,” IHC125412, 2007. [En línea]. Disponible: <http://ihc125412.blogspot.com/2007/09/avances-de-ihc-biometra.html>
- [8] *Raúl Sánchez Reíl, “Verificación automática de personas mediante huella dactilar”, Grupo Universitario de Tarjeta Inteligente, Departamento de Tecnología, Electrónica – Grupo de Microelectrónica, UNIVERSIDAD CARLOS III DE MADRID, junio 2002*
- [9] S. Pressman, Roger, “Ingeniería del software, un enfoque”, (5ª edición), McGraw-Hill, España, 2002
- [10] Sommerville, Ian, “Ingeniería del software”, (7ª edición), Pearson Educación, S.A., Madrid, 2005.
- [11] M. Riodan, Rebeca, “Diseño de bases de datos relacionales con Access y SQL Server”, McGraw-Hill, España, 2000.
- [12] Ramez A. Elmasri, Shamkant B. Navathe, “Sistemas De Bases de Datos, conceptos fundamentales”, 3ª edición, Addison Wesley, Madrid, 2002.
- [13] Silberschatz Abraham, F. Korth Henry, Sudarshan S., “Fundamentos de bases de datos” 4ª edición, McGRAW-HILL, 2002.



BIBLIOGRAFIA



[14] Información general del producto SQL, [En línea]. Disponible: <http://www.microsoft.com/spain/sql/productinfo/overview/default.msp>

[15] Eguíluz Pérez Javier, “Introducción a AJAX”, <http://www.librosweb.es/ajax>, 2008