



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE CÓMPUTO EN LÍNEA PARA
CONTROL DE ASISTENCIAS**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

**PRESENTAN:
EFRAIN CRUZ JIMÉNEZ
FERNANDO RUIZ SALAZAR**



**ASESOR
DRA. ANA MARÍA VÁZQUEZ VARGAS**

CIUDAD UNIVERSITARIA

2009



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

SISTEMA DE CÓMPUTO EN LÍNEA PARA CONTROL DE ASISTENCIAS

ÍNDICE

INTRODUCCIÓN	4	
CAPÍTULO 1	MARCO TEÓRICO	6
1.1. Ingeniería de Software.		7
1.1.1 Ciclo de vida del software		7
1.1.1.1 Análisis de Requisitos.		7
1.1.1.2 Especificación.		8
1.1.1.3 Diseño y arquitectura.		8
1.1.1.4 Programación.		8
1.1.1.5 Pruebas.		8
1.1.1.6 Documentación.		8
1.1.1.7 Implementación.		8
1.1.2 Objetivos de la Ingeniería de Software.		8
1.1.3 Tipos de metodologías.		9
1.1.3.1 Definición y objetivos de la metodología.		9
1.1.3.2 Clasificación de las Metodologías		10
1.1.3.2.1 Metodologías Estructuradas.		10
1.1.3.2.2 Metodologías Orientadas a Objetos		10
1.2. Bases de datos		10
1.2.1 Conceptos Básicos		10
1.2.2 Modelo relacional		10
1.2.2.1 Definición de modelo relacional		10
1.2.2.2 Operadores		11
1.2.2.3 Proceso de Normalización		13
1.2.3 Lenguaje de consulta estructurado (SQL)		16
1.2.3.1 Lenguaje de definición de datos		16
1.2.3.2 Lenguaje de manipulación de datos		17
1.3. Software de Distribución Libre		18
1.3.1 Definición		18
1.3.2 Diferencias con "Open Source"		18
1.3.3 PHP y JAVA como lenguajes de programación de distribución libre		19
1.4. Integración de Sistemas		19
1.4.1 Definición		19
1.5. Conexiones a Bases de Datos		20
1.5.1 Definición.		20
1.5.2 Conceptos relacionados.		20
1.6. Internet		21
1.6.1 Definición		21
1.6.2 WWW		21
1.6.3 Uso y funcionalidad		21
CAPÍTULO 2	ANÁLISIS PRELIMINAR	22
2.1. Análisis del Problema		23
2.1.1 Planteamiento del problema		23
2.1.2 Problemática actual		23
2.1.3 Levantamiento de requerimientos		23

2.1.4 Solución Propuesta	30
2.1.4.1 Implementación de la tecnología necesaria	30
2.1.4.2 Módulo de administración de alumnos y tutores	31
2.1.4.3 Credencialización	31
2.1.4.4 Módulo de reportes	31
2.1.4.5 Portal web de consulta para los tutores	31
2.1.4.6 Seguridad de la Solución	32
CAPÍTULO 3	FLUJO FÍSICO Y COMPONENTES
3.1. Generalidades	34
3.2. Servidor	35
3.2.1 Requerimientos mínimos	35
3.2.2 Instalación de servidor PHP	35
3.2.2.1 Instalación del sistema de cómputo en línea para control de asistencias	37
3.2.3 Instalación de SQL Server 2000 y Servidor SQL	38
3.2.4 Instalación de software para plataforma de SMS	40
3.2.4.1 Pruebas de instalación	45
3.2.5 Instalación de software de relojes checadores	46
3.2.6 Instalación de servidor ftp	47
3.3 Terminales o Clientes	49
3.3.1 Especificaciones (Requerimientos mínimos)	49
3.3.2 Instalación de software auxiliares	50
3.3.2.1 Clientes FTP	50
3.3.2.2 Software de impresión de credenciales	52
3.3.3 Acceso al sistema vía red local	53
3.3.4 Acceso al sistema vía Internet	54
3.4. Red	55
3.4.1 Componentes	55
3.4.1.1 RJ45	55
3.4.1.2 Cable UTP	56
3.4.1.3 Tarjeta de red	56
3.4.1.4 Switch	56
3.4.2 Configuración de equipos	56
3.4.2.1 Servidor	57
3.4.2.2 Clientes	57
3.4.2.3 Sensores	58
3.5. Sensores	59
3.5.1 Especificaciones	59
3.5.2 Pruebas de funcionamiento	60
3.6. SMS	61
3.6.1 Instalación de interfaz de conectividad	61
3.6.2 Componentes	62
3.6.2.1 Especificaciones de equipos (requerimientos mínimos)	62
3.6.2.2 Instalación de controladores de dispositivos	62
3.6.2.3 Pruebas de conectividad	62
3.6.3 Pruebas de envíos manuales	64
3.7. Impresora de Credenciales	65
3.7.1 Especificaciones y características	65

3.8. Credenciales	66
3.8.1 Especificaciones y características	66
CAPÍTULO 4	ARQUITECTURA DEL SISTEMA
	67
4.1 Proceso General	68
4.1.1 Descripción	68
4.2 Base de Datos	68
4.2.1 Diccionario de Datos	68
4.2.1.1 Prefijos	68
4.2.1.2 Diccionario de datos	69
4.2.2 Diagrama Entidad Relación	74
4.2.2.1 Nomenclatura	74
4.2.2.2 Diagrama Entidad Relación	75
4.2.3 Diagrama Físico	76
4.2.4 Script SQL de la Base de Datos modelada	76
CAPÍTULO 5	MANUAL TÉCNICO
	81
5.1. Sección Administrador	82
5.1.1 Explicación de pestañas	83
5.1.2 Pantallas con código y explicación de las mismas	99
5.1.3 Especificación de cada lenguaje de programación	105
5.1.4 Sesiones y conectividad con la Base de Datos	108
CAPÍTULO 6	MANUAL DE USUARIO
	110
6.1. Servicio a usuarios	111
6.1.1 Portal para padres de familia	111
6.1.2 Pantallas con código y explicación de las mismas	114
CONCLUSIONES	120
GLOSARIO	122
BIBLIOGRAFÍA	125

INTRODUCCIÓN

INTRODUCCIÓN

El importante crecimiento que ha tenido en épocas recientes el uso de la telefonía móvil así como el internet ha logrado hacer que estos dos servicios sean parte de la vida cotidiana, tanto laboral como personal, de millones de personas alrededor del mundo.

Hoy en día principalmente dentro de organizaciones industriales y académicas el control de asistencia, control en los accesos, horarios de entrada y horarios de salida se ha convertido en una actividad cotidiana necesaria y esencial en las labores dentro de estos rubros, siendo cada vez mayor la importancia en la automatización de esta tarea, ya que es a través de este control es como se pueden determinar acciones a realizar tales como disciplina laboral, incentivos, sanciones, notificaciones, etc.

Basados en la importante presencia de estos dos servicios y en la necesidad de la automatización en el control de las asistencias, en la presente tesis documentamos el análisis, diseño, desarrollo e implementación de un sistema que integra ambos a fin de poder lograr un producto único que proporcione los beneficios que de manera independiente cada uno de ellos brinda.

El presente documento describe el proceso seguido para desarrollar el “Sistema de Cómputo en Línea para el Control de Asistencias”. El análisis de este proyecto ha sido desglosado en un total de 6 capítulos, descritos brevemente a continuación:

En el primer capítulo se plantean todos los conocimientos teóricos necesarios y fundamentales para poder dar la seguridad que el desarrollo del proyecto en curso llegará a buen término y tendrá una vida útil tanto duradera como productiva. Estos conocimientos no son más que las bases establecidas por la formación ingenieril, así como dentro de un marco de desarrollo común, como en uno más específico que es el área de la computación.

En el segundo capítulo se plantea la problemática actual de la institución en que será implementado el sistema por desarrollar, llevándose a cabo un levantamiento de requerimientos, en base a los cuales se propone una solución.

En el capítulo ulterior, se plantean y describen todos los componentes físicos necesarios para implementar la solución propuesta y se describe la forma en la que estarán interactuando mutuamente cada uno de ellos.

En el cuarto capítulo se documenta la arquitectura del sistema. La documentación generada para esta etapa de la tesis corresponde al diseño, desarrollo e implementación de la Base de Datos que el sistema utilizará, así como la interfaz a desarrollar.

El penúltimo capítulo hace referencia al manual técnico que brinda las bases y procedimientos seguidos para el desarrollo del módulo de administración del Sistema. En el mismo capítulo se ilustra la forma en que el administrador del sistema debe operar dicho módulo.

El último capítulo ha sido desarrollado para brindar al usuario final del sistema un manual que le indica cómo utilizar y explotar todas las herramientas que a su disposición se encuentran, ya que es en esta sección del sistema donde se puede observar el resultado perseguido desde un inicio, que es, brindar a los padres de familia o tutores una herramienta que les permita tener conocimiento de las asistencias que los alumnos bajo su responsabilidad están generando día a día en la institución.

CAPÍTULO 1

MARCO TEÓRICO

CAPÍTULO 1 MARCO TEÓRICO.

1.1 INGENIERÍA DE SOFTWARE

Disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Esta ingeniería trata con áreas muy diversas de la Computación, tales como construcción de compiladores, Sistemas Operativos o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de Sistema de Información y aplicables a infinidad de áreas (negocios, investigación científica, medicina, banca, meteorología, Internet, etc.).

Una definición precisa aún no ha sido contemplada en los diccionarios, sin embargo se pueden citar las enunciadas por algunos autores:

1. Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkowitz, 1978)
2. Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. (Bohem, 1976).
3. Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).
4. Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE, 1993).

1.1.1 Ciclo de Vida del Software

Un objetivo de décadas ha sido el encontrar procesos y metodologías, que sean sistemáticas, predecibles y repetibles, a fin de mejorar la productividad en el desarrollo y la calidad del producto software.

Etapas del proceso

La ingeniería de software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

- ❖ Análisis de requisitos
- ❖ Especificación
- ❖ Diseño y arquitectura
- ❖ Programación
- ❖ Pruebas
- ❖ Documentación
- ❖ Implementación

1.1.1.1 Análisis De Requisitos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, *Especificación de Requerimientos del Sistema*, cuya estructura puede venir definida por varios estándares, tales como CMM-I. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos

procesos de trabajo para estos fines. Aunque aun no está formalizada, ya se habla de la Ingeniería de Requisitos.

La IEEE Std. 830-1998 normaliza la creación de las Especificaciones de Requisitos Software (Software Requirements Specification).

1.1.1.2 Especificación

Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.

1.1.1.3 Diseño y arquitectura

Se refiere a determinar como funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc. Se definen los Casos de Uso para cubrir las funciones que realizará el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.

1.1.1.4 Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

1.1.1.5 Pruebas

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es revisar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que aparte de las pruebas que el desarrollador debe efectuar, las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en que condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

1.1.1.6 Documentación

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, diagramas, pruebas, manuales de usuario, manuales técnicos, etc. todo con el propósito de posibles correcciones, interoperabilidad, mantenimiento y futuras ampliaciones al sistema.

1.1.1.7 Mantenimiento

Mantener y mejorar el software para corregir errores descubiertos o implementar nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. La mayor parte de esta etapa consiste en extender el sistema para obtener un producto terminal más completo y con la menor cantidad de deficiencias.

1.1.2 Objetivos de la Ingeniería del Software

La ingeniería de software difiere de la programación tradicional ya que se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto

establecidos.

La ingeniería de software es la aplicación de técnicas de ingeniería y administración al proceso de crear productos de software utilizando metodologías, herramientas y procesos.

Las actividades que se desempeñan en la ingeniería de software están regidas por la aplicación de tres elementos importantes:

1. Metodología: Incluye la identificación de las necesidades, planificación y estimación de proyectos, sistemas de calidad, análisis de requerimientos, métodos de diseño, de codificación y de pruebas, así como el mantenimiento de productos.
2. Procesos: Un proceso de ingeniería de software es una definición del conjunto completo de actividades necesarias para transformar los requisitos del usuario en un producto de software. Definen la documentación de cada una de las fases de desarrollo y los controles que aseguren la calidad del producto a lo largo del proyecto.
3. Herramientas: Son sistemas automáticos que apoyan en la aplicación de las metodologías en el desarrollo del producto. Incluyen herramientas como las llamadas CASE (Computer Aided Software Engineering), de control automático de configuración, ensayo automático de software, bases de datos de registro, lenguajes de programación, sistemas operativos, computadoras, entornos de desarrollo, etc.

Dentro de los objetivos principales de la ingeniería de software se encuentran:

- ❖ Construir un producto de software o mejorar uno existente.
- ❖ Mejorar la calidad del producto que se entrega al cliente.
- ❖ Aumentar la productividad de los ingenieros de software.
- ❖ Asegurar la eficiencia del proceso de desarrollo y mantenimiento del producto.
- ❖ Fomentar la reusabilidad de los componentes del software.
- ❖ Mejorar la planificación de los proyectos.

En la industria del software, el control y la calidad de los procesos para aumentar la calidad del producto esta convirtiéndose en una norma, por lo que la necesidad de contar con una guía para el mejoramiento del proceso de desarrollo se convierte en algo imprescindible.

1.1.3 Tipos de Metodologías

1.1.3.1 Definición y objetivos de la metodología.

Una metodología de desarrollo es un conjunto de técnicas, fases, procedimientos, reglas, herramientas y documentación que indican como desarrollar un proyecto de software.

Este conjunto de fases se descompone a su vez en otras, las cuales guían a los desarrolladores en la elección de las técnicas que debe elegir para cada proyecto, facilitando así la planificación, gestión, control y evaluación de los proyectos.

Los objetivos de una metodología de desarrollo son:

- ❖ Registrar los requisitos de un sistema de información de forma acertada.
- ❖ Proporcionar un método sistemático de desarrollo de forma que se pueda controlar su progreso.
- ❖ Construir un sistema de información dentro de un tiempo apropiado y con costos aceptables.
- ❖ Elaborar la documentación del sistema detalladamente y que sea fácil de mantener.
- ❖ Proporcionar un sistema que satisfaga a todas las personas afectadas por el mismo, ya sean clientes, directivos, auditores o usuarios.
- ❖ Garantizar la calidad de la aplicación.
- ❖ Promover la creación de un proceso estándar.

Una metodología puede seguir uno o varios modelos de ciclo de vida. El ciclo de vida indica que es lo que hay que obtener a lo largo del desarrollo del proyecto, mientras que la metodología indica el como.

La ingeniería de software tiene varios modelos o paradigmas de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos enlistarlos a continuación:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo en espiral (modelo evolutivo)
- Modelo de prototipos
- Desarrollo por etapas
- Desarrollo iterativo y creciente o Iterativo Incremental
- RAD (Rapid Application Development)

1.1.3.2 Clasificación de las Metodologías

Las metodologías se clasifican en dos tipos de acuerdo al modo en que el sistema a desarrollar es analizado o modelado. Estos tipos de metodologías son las estructuradas y las orientadas a objetos.

1.1.3.2.1 Metodologías Estructuradas.

Examinan los sistemas desde el punto de vista de las funciones o tareas que deben realizar, tareas que se van descomponiendo sucesivamente en otras tareas de menor dimensión y que forman los bloques o módulos de las aplicaciones. Además, las metodologías estructuradas proponen la creación de modelos del sistema que representan los procesos, los flujos y la estructura de los datos de una manera descendente, dando lugar a metodologías orientadas a procesos y orientadas a datos (tanto a datos jerárquicos como no jerárquicos)

1.1.3.2.2 Metodologías Orientadas a Objetos

En las metodologías orientadas a objetos, el modelado del sistema, es un aspecto que cobra mucha más importancia, de tal manera, que el dominio del problema se examina como un conjunto de objetos que interactúan entre sí. A diferencia de las metodologías estructuradas, en las cuales existe una división entre las funciones que llevan a cabo los programas y los datos que se almacenan en las bases de datos, en las metodologías orientadas a objetos existe un enfoque unificador de ambos aspectos, los cuales se encapsulan en los objetos.

1.2. BASES DE DATOS

1.2.1 CONCEPTOS BÁSICOS

Base de datos. Término genérico que se utiliza para designar el sistema de administración de bases de datos relacionales (RDBMS, del inglés). Paquete de software que permite crear y manipular grandes cantidades de datos relacionados y organizados. Consulte también

1.2.2 MODELO RELACIONAL

1.2.2.1 DEFINICIÓN DE MODELO RELACIONAL

El **modelo relacional** para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Edgar Frank Codd, de los laboratorios IBM en San José (California), postuló las bases en 1970. Este modelo no tardó en consolidarse como un nuevo paradigma en los modelos de

base de datos.

Las bases de datos relacionales son el tipo de bases de datos actualmente más difundido. Los motivos de este éxito son fundamentalmente dos:

- 2 ofrecen sistemas simples y eficaces para representar y manipular los datos
- 3 se basan en un modelo, el relacional, con sólidas bases teóricas

El modelo relacional fue propuesto originariamente por E.F. Codd en un ya famoso artículo de 1970. Gracias a su coherencia y facilidad de uso, el modelo se ha convertido en los años 80 en el más usado para la producción de DBMS.

La estructura fundamental del modelo relacional es precisamente esa, "relación", es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. 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. Por ejemplo, si en la base de datos se tienen que representar personas, se podrá definir una relación llamada "Personas", cuyos atributos describen las características de las personas (tabla siguiente). Cada tupla de la relación "Personas" representará una persona concreta.

En realidad, siendo rigurosos, una relación es sólo la definición de la estructura de la tabla, es decir su nombre y la lista de los atributos que la componen. Cuando se puebla con las tuplas, se habla de "instancia de relación". Una representación de la definición de esa relación podría ser la siguiente:

Personas (nombre, apellido, fecha_nacimiento, sexo, estado_civil)

PERSONAS				
Nombre	Apellido	Nacimiento	Sexo	Estado Civil
Juan	Loza	15/06/1971	H	Soltero
Isabel	Galvez	23/12/1969	M	Casada
Maria	Ruiz	02/10/1985	M	Soltera

Las tuplas en una relación son un conjunto en el sentido matemático del término, es decir una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de "**llave primaria**", o sea a un conjunto de atributos que permiten identificar unívocamente una tupla en una relación. Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla ("**llaves candidatas**"), pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo (que significa un valor no determinado), en tanto que ya no permitirían identificar una tupla concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias está bajo el nombre de integridad de las entidades (*entity integrity*).

Es común que para obtener una llave primaria "económica", es decir compuesta de pocos atributos fácilmente manipulables, se introduzcan uno o más atributos ficticios, con códigos identificativos unívocos para cada tupla de la relación.

1.2.2.2 OPERADORES

Una de las grandes ventajas del modelo relacional es que define también un álgebra, llamada "**álgebra relacional**". Todas las manipulaciones posibles sobre las relaciones se obtienen gracias a la combinación de tan sólo cinco operadores: RESTRICT, PROJECT, TIMES, UNION y MINUS. Por comodidad, se han definido también tres operadores adicionales que de todos modos se pueden obtener aplicando los cinco fundamentales: JOIN, INTERSECT y DIVIDE. Los operadores relacionales reciben como argumento una relación o un conjunto de relaciones y restituyen una única relación como resultado.

Veamos brevemente estos ocho operadores:

RESTRICT

Restituye una relación que contiene un subconjunto de las tuplas de la relación a la que se aplica. Los atributos se quedan como estaban.

PROJECT

Restituye una relación con un subconjunto de los atributos de la relación a la que viene aplicado. Las tuplas de la relación resultado se componen de las tuplas de la relación original, de manera que siguen siendo un conjunto en sentido matemático.

TIME

Se aplica a dos relaciones y efectúa el producto cartesiano de las tuplas. Cada tupla de la primera relación está concatenada con cada tupla de la segunda.

JOIN

Se concatenan las tuplas de dos relaciones de acuerdo con el valor de un conjunto de sus atributos.

UNION

Aplicando este operador a dos relaciones compatibles, se obtiene una que contiene las tuplas de ambas relaciones. Dos relaciones son compatibles si tienen el mismo número de atributos y los atributos correspondientes en las dos relaciones tienen el mismo dominio.

MINUS

Aplicado a dos relaciones compatibles restituye una tercera que contiene las tuplas que se encuentran sólo en la primera relación.

INTERSECT

Aplicado a dos relaciones compatibles restituye una relación que contiene las tuplas que existen en ambas.

DIVIDE

Aplicado a dos relaciones que tengan atributos comunes, restituye una tercera que contiene todas las tuplas de la primera relación que se puede hacer que correspondan con todos los valores de la segunda relación.

En las siguientes tablas, se representan los resultados de la aplicación de algunos operadores relacionales a las relaciones Personas e Hijos. Como nombres para las relaciones resultado se han utilizado las expresiones que las producen.

Personas

numero_persona	nombre	apellido	fecha_nacimiento	sexo	estado_civil
2	Mario	Rojas	29/03/1965	M	Casado
1	Jorge	Vargas	15/11/1972	M	Soltero
3	Alessandra	Frías	13/06/1970	F	Soltera

Hijos

numero_persona	nombre_apellido	edad	sexo
2	Maria Rojas	3	F
2	Israel Rojas	5	M

RESTRICT (Personas)

sexo='M'

numero_persona	nombre	apellido	fecha_nacimiento	sexo	estado_civil
2	Mario	Rojas	29/03/1965	M	Casado
1	Jorge	Vargas	15/11/1972	M	Soltero

PROJECT sexo (Personas)

sexo
M
F

RESTRICT (Personas)

sexo='M'

n.	nombre	apellido	Nacimiento	sexo	estado_civil	nombre	edad	sexo
Mario	Rojas	apellido	29/03/1965	M	Casado	Maria Rojas	3	F
Mario	Rojas	Apellido	29/03/1965	M	Casado	Israel Rojas	5	M

Las bases de datos relacionales efectúan todas las operaciones en las tablas usando el álgebra relacional, aunque normalmente no le permiten al usuario hacer uso de ella. El usuario interactúa con la base de datos a través de una interfaz diferente: el lenguaje SQL, un lenguaje declarativo que permite escribir conjuntos de datos. Las instrucciones SQL vienen descompuestas por el motor de datos en una serie de operaciones relacionales.

1.2.2.3 PROCESO DE NORMALIZACIÓN

El proceso de normalización es un estándar que consiste básicamente en un proceso de conversión de las relaciones entre las entidades, con esto se evita:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

El proceso de normalización nos conduce hasta el modelo físico de datos y consta de varias fases denominadas formas normales, estas formas se detallan a continuación.

Definición de la clave (Llave Primaria)

Antes de proceder a la normalización de la tabla lo primero que debemos de retomar la definición de una clave, esta clave deberá contener un valor único para cada registro (no podrán existir dos valores iguales en toda la tabla) y podrá estar formado por un único campo o por un grupo de campos.

En la tabla de alumnos de una escuela no podemos definir como campo clave el nombre del alumno ya que pueden existir varios alumnos con el mismo nombre. Podríamos considerar la posibilidad de definir como clave los campos nombre y apellidos, pero estamos en la misma situación: podría darse el caso de alumnos que tuvieran los mismos apellidos y el mismo nombre.

La solución en este caso es asignar un código de alumno a cada uno, un número que identifique al alumno y que estemos seguros que es único.

Una vez definida la clave podremos pasar a estudiar la primera forma normal.

Primera forma normal (1NF)

Se dice que una tabla se encuentra en primera forma normal (1NF) si y solo si cada uno de los campos contiene un único valor para un registro determinado. Supongamos que deseamos realizar una tabla para guardar los cursos que están realizando los alumnos de un determinado centro de estudios, podríamos considerar el siguiente diseño:

Código	Nombre	Cursos
1	Marcos	Inglés
2	Lucas	Contabilidad, Informática
3	Marta	Inglés, Contabilidad

Podemos observar que el registro de código 1 si cumple la primera forma normal, cada campo del registro contiene un único dato, pero no ocurre así con los registros 2 y 3 ya que en el campo cursos contiene más de un dato cada uno. La solución en este caso es crear dos tablas del siguiente modo:

Tabla A

Código	Nombre
1	Marcos
2	Lucas
3	Marta

Tabla B

Código	Curso
1	Inglés
2	Contabilidad
2	Informática
3	Inglés
3	Informática

Como se puede comprobar ahora todos los registros de ambas tablas contienen valores únicos en sus campos, por lo tanto ambas tablas cumplen la primera forma normal.

Una vez normalizada la tabla en 1NF, podemos pasar a la segunda forma normal.

Segunda forma normal (2NF)

La segunda forma normal compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está es segunda forma normal (2NF).

Supongamos que construimos una tabla con los años que cada empleado ha estado trabajando en cada departamento de una empresa:

Código Empleado	Código Dpto.	Nombre	Departamento	Años
1	6	Juan	Contabilidad	6
2	3	Pedro	Sistemas	3
3	2	Sonia	RH	1
4	3	Verónica	Sistemas	10
2	6	Pedro	Contabilidad	5

Tomando como punto de partida que la clave de esta tabla está formada por los campos código de empleado y código de departamento, podemos decir que la tabla se encuentra en primera forma normal, por tanto vamos a estudiar la segunda:

1. El campo nombre no depende funcionalmente de toda la clave, sólo depende del código del empleado.
2. El campo departamento no depende funcionalmente de toda la clave, sólo del código del departamento.
3. El campo años si que depende funcionalmente de la clave ya que depende del código del empleado y del código del departamento (representa el número de años que cada empleado ha trabajado en cada departamento)

Por tanto, al no depender todos los campos de la totalidad de la clave la tabla no está en segunda forma normal, la solución es la siguiente:

Tabla A

Código Empleado	Nombre
1	Juan
2	Pedro
3	Sonia
4	Verónica

Tabla B

Código Departamento	Dpto.
2	I+D
3	Sistemas
6	Contabilidad

Tabla C

Código Empleado	Código Departamento	Años
1	6	6
2	3	3
3	2	1
4	3	10
2	6	5

Podemos observar que ahora si se encuentran las tres tablas en segunda forma normal, considerando que la tabla A tiene como índice el campo Código Empleado, la tabla B Código Departamento y la tabla C una clave compuesta por los campos Código Empleado y Código Departamento.

Tercera forma normal (3NF)

Se dice que una tabla está en tercera forma normal si y solo si los campos de la tabla dependen únicamente de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros. Tomando como referencia el ejemplo de los alumnos dentro de una escuela, supongamos que cada alumno sólo puede realizar un único curso a la vez y que deseamos guardar en que aula se imparte el curso. Se podría plantear la siguiente estructura:

Código	Nombre	Curso	Aula
1	Marcos	Informática	Aula A
2	Lucas	Inglés	Aula B
3	Marta	Contabilidad	Aula C

Estudieemos la dependencia de cada campo con respecto a la clave "Código":

- Nombre depende directamente del código del alumno.
- Curso depende de igual modo del código del alumno.
- El aula, aunque en parte también depende del alumno, está más ligado al curso que el alumno está realizando.

Por esta última razón se dice que la tabla no está en 3NF. La solución sería la siguiente:

Tabla A

Código	Nombre	Curso
1	Marcos	Informática
2	Lucas	Inglés
3	Marta	Contabilidad

Tabla B

Curso	Aula
Informática	Aula A
Inglés	Aula B
Contabilidad	Aula C

1.2.3 LENGUAJE DE CONSULTA ESTRUCTURADO

El Lenguaje de consulta estructurado SQL (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre la misma.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento", que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y las metodologías orientadas a objetos. De esta forma una sola sentencia puede equivaler a uno o más programas que se utilizan en un lenguaje de bajo nivel orientado a registro.

1.2.3.1 LENGUAJE DE DEFINICIÓN DE DATOS

El lenguaje de definición de datos (en inglés *Data Definition Language*, o *DDL*), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Existen cuatro operaciones básicas:

1. CREATE
2. ALTER
3. DROP
4. RUNCATE.

CREATE

Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Ejemplo:

```
CREATE TABLE [dbo].[app_SysSections] (
    [id_SysSection] [int] NOT NULL ,
    [sysSection] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [active] [bit] NULL
)
```

ALTER

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc. Ejemplo:

```
ALTER TABLE [dbo].[app_SysSections] WITH NOCHECK ADD
```

```
CONSTRAINT [PK_app_SysSections] PRIMARY KEY CLUSTERED
(
    [id_SysSection]
)
```

DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

```
ALTER TABLE "TABLA_NOMBRE"
(
    DROP COLUMN "CAMPO_NOMBRE1"
)
```

TRUNCATE

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DELETE, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande, la desventaja es que TRUNCATE solo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando truncate borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

```
TRUNCATE TABLE "TABLA_NOMBRE1"
```

1.2.3.2 LENGUAJE DE MANIPULACIÓN DE DATOS

Un lenguaje de manipulación de datos (*Data Manipulation Language*, o *DML* en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

INSERT

Una sentencia *INSERT* de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional. Ejemplo de su forma básica:

```
INSERT INTO tbl_UsuariosWeb (st_Nombre, dt_FechaNacimiento)
VALUES ('Fernando', CONVERT(DATETIME, '1984-09-20 00:00:00', 102))
```

UPDATE

Una sentencia *UPDATE* de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla. Ejemplo de su forma básica:

```
UPDATE tbl_UsuariosWeb
SET st_Nombre = 'José', dt_FechaNacimiento = CONVERT(DATETIME, '1984-09-20 00:00:00', 102)
```

DELETE

Una sentencia *DELETE* de SQL borra cero o más registros existentes en una tabla, Ejemplo de su forma básica:

```
DELETE FROM tbl_UsuariosWeb
```

1.3 SOFTWARE DE DISTRIBUCIÓN LIBRE

1.3.1 Definición

Software libre (en inglés *free software*) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la *Free Software Foundation*, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software:

1. La libertad de usar el programa, con cualquier propósito;
2. De estudiar el funcionamiento del programa, y adaptarlo a las necesidades;
3. De distribuir copias, con lo que puede ayudar a otros;
4. De mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo).

El software de distribución libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por ende no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en algunas ocasiones el código fuente; no obstante, este tipo de software *no es libre* en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Este último es aquél que no requiere de licencia, pues sus derechos de explotación son para todos los usuarios, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a los usuarios o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.

1.3.2 Diferencias con Open Source

Open Source apareció en 1998 con un grupo de personas que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento Open Source ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del movimiento Open Source como del término "Open Source".

Aunque el término "Open Source" elimina la ambigüedad de Libertad frente a Precio (en el caso del Inglés), introduce una nueva: entre los programas que se ajustan a la *Open Source Definition*, que dan a los usuarios la libertad de mejorarlos, y los programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es *open source*, puesto que lo pueden manipular. Sin embargo,

mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

1.3.3 PHP y JAVA como lenguajes de programación de distribución libre

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server - side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PHP significa *PHP Hypertext Pre-processor*. PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

1.4 Integración de Sistemas

1.4.1 Definición

La integración de sistemas reúne los componentes de subsistemas en un sistema. En tecnologías de la información, integración de sistemas es el proceso de vinculación de los distintos sistemas de computación y aplicaciones de software física o funcionalmente.

Un sistema es un conjunto de subsistemas que cooperaron para que el sistema sea capaz de entregar más de una única funcionalidad. La integración de sistemas podría involucrar la integración de subsistemas existentes. Integración implica incorporarse a los subsistemas juntos por "pegar" sus interfaces juntos. Sistema de integración es también de valor añadido al sistema, las capacidades que son posibles debido a la interacción entre los subsistemas.

En el actual mundo el papel de los ingenieros de integración de sistemas es cada vez más importante. Muchos sistemas están diseñados para conectarse entre sí, tanto en sistemas que están por implementarse como en los sistemas que ya están desplegados.

1.5 CONEXIONES A BASES DE DATOS

1.5.1 Definición

Vínculo de comunicación con una base de datos u otra fuente de datos. Los componentes pueden crear y manipular varias conexiones a bases de datos a la vez para acceder a los datos.

Una conexión a una base de datos es un método dentro del campo de la computación que permite dentro de una aplicación cliente-servidor al cliente comunicarse con el servidor de bases de datos, aun cuando este exista o no dentro de la misma computadora. Una conexión es requerida para enviar comandos y recibir respuestas, usualmente en forma de un conjunto de datos (conocido en inglés como result set)

Las conexiones han sido construidas para ser brindadas a través de una cadena de conexión (connection string), la cual requiere de la autenticación del usuario que intenta acceder a instancias de una base de datos o un servidor. Una vez que se ha logrado establecer una conexión, esta puede ser cerrada o abierta y las propiedades de la misma (tales como el tiempo en el que expira la conexión, las transacciones a realizar, etc.) pueden ser ajustadas.

Muchas bases de datos (como SQL SERVER 2000) únicamente permiten una operación a la vez para una conexión. En otras palabras, si una consulta (una sentencia SQL SELECT) es enviada a la base de datos, y un conjunto de resultados (result set) es regresado, la conexión se encontrará abierta, pero no disponible hasta que la transacción sea concluida. Sin embargo, actualmente existen bases de datos que permiten múltiples operaciones por conexión.

1.5.2 Conceptos Relacionados

A continuación se define una serie de conceptos relacionados con las conexiones a Bases de Datos:

Servicio de datos. Servicio web que admite la consulta y modificación de los datos relacionados con un usuario final. Un ejemplo de servicio de datos es un servicio web que aloja y expone la información del perfil del usuario como, por ejemplo, el nombre, la dirección y el número de teléfono.

Fuente de datos. Acceso a una fuente de datos; por ejemplo, una base de datos. Las fuentes de datos se registran con Application Server y después se recuperan mediante programación con el fin de establecer conexiones e interactuar con la fuente de datos. Una definición de fuente de datos especifica el modo de conexión a la fuente de datos.

Objeto de fuente de datos. Posee un conjunto de propiedades que identifica y describe la fuente de datos real que representa.

Conjunto de fuentes de datos. Fuentes de datos que contienen datos equivalentes.

Almacén de datos. Almacén que contiene información sobre el directorio, normalmente de un DIT completo. Base de datos en la que se almacena de forma permanente la información necesaria (suscripciones duraderas, datos sobre los destinatarios, mensajes persistentes o datos de auditoría).

Vista de datos. Enruta conexiones desde administradores de conexión de Directory Proxy Server hasta conjuntos de fuentes de datos.

1.6. INTERNET

1.6.1 Definición

Según la Real Academia de la Lengua Española, Internet se define como:
1. amb. Red informática mundial, descentralizada, formada por la conexión directa entre computadoras u ordenadores mediante un protocolo especial de comunicación.

Internet es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre algunas universidades estadounidenses.

1.6.2 WWW

Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Utiliza Internet como medio de transmisión.

1.6.3 USO Y FUNCIONALIDAD

Existen muchos otros servicios y protocolos en Internet, aparte de la World Wide Web, de los principales podríamos mencionar:

- ❖ El envío de correo electrónico (SMTP)
- ❖ Transmisión de archivos (FTP y P2P)
- ❖ Conversaciones en línea (IRC)
- ❖ Mensajería instantánea y presencia
- ❖ Transmisión de contenido
- ❖ Comunicación multimedia
- ❖ Telefonía (VoIP) y televisión (IPTV)
- ❖ Acceso remoto a otras máquinas (SSH y Telnet)

CAPÍTULO 2

ANÁLISIS

PRELIMINAR

CAPÍTULO 2 ANÁLISIS PRELIMINAR.

2.1. ANÁLISIS DEL PROBLEMA.

2.1.1 Planteamiento del problema

Personal administrativo de la institución escolar presentó la forma actual en la que se controla la asistencia de los alumnos. Se menciona que no existe un control general de asistencias, es decir, que únicamente los profesores de grupo tienen la información de los alumnos que no asistieron a clases, con sus listas de asistencia. De esta manera el director o subdirector de la institución no tienen conocimiento de la cantidad de alumnos que en un día no asistieron.

La información de las listas de asistencia solo está disponible con los profesores de cada grupo, únicamente bajo la solicitud del director, esta información es conocida por el personal administrativo de la institución. Y únicamente cuando se presentan problemas académicos con los alumnos, esta información es transmitida a los padres de familia de los alumnos.

2.1.2 Problemática Actual

Continuamente se presentan casos en los que los alumnos salen de sus casas hacia la escuela, sin embargo no asisten a la misma, llevan a cabo actividades fuera de las instalaciones escolares, poniendo en riesgo su persona, salud y seguridad. Se han presentado casos en que los alumnos no llegan a su domicilio y los padres de familia presentan quejas a los directivos, argumentando situaciones inventadas por los alumnos, lo que genera conflictos al no tener evidencia de que el alumno en ningún momento se presentó en la institución.

No existe información que permita a los directivos de la institución analizar el rendimiento de los alumnos conforme a las asistencias que estos han tenido, situación similar a los padres de familia que por motivos laborales no pueden llevar a sus hijos a la escuela y desean saber si estos asistieron y a que hora ingresaron.

La institución no cuenta actualmente con tecnología que permita implementar una solución automatizada a la problemática que presenta. Sin embargo, existe disposición por parte de la institución para adquirir los recursos que les permita de una manera económica y rápida implementar una solución tecnológica.

2.1.3 Levantamiento de Requerimientos

Los directivos de la institución escolar, han solicitado una solución tecnológica que les brinde información de las asistencias de los alumnos en tiempo real, de forma fidedigna. Así mismo, la solución propuesta deberá permitir a los padres de familia desde cualquier lugar con conexión a Internet poder consultar la misma información y para el caso de las familias que no cuenten con el servicio de Internet se les deberá notificar a través de la telefonía móvil mediante un mensaje de texto SMS (Short Message Service por sus siglas en inglés).

Para conocer completamente la situación actual, comprender la problemática actual y poder establecer una propuesta de solución al problema, se llevaron a cabo 4 reuniones de trabajo, documentadas a continuación.

MINUTA DE TRABAJO No. 1

Proyecto:	Sistema de Cómputo en Línea para Control de Asistencias	Fecha:	08/Feb/2008
Lugar:	Escuela Secundara Emiliano Zapata	Duración :	3 hrs
Objetivo:	Visita a instalaciones y Presentación de Servicios	Elaboró:	ECJ

Favor de leer la minuta y en caso de tener comentarios, indicarlos en un lapso no mayor a 5 días hábiles posteriores a la fecha de recepción de la minuta, una vez transcurrido este lapso, se dará por aceptada la minuta y pasará a formar parte del acervo documental del proyecto.

Asistentes			
Ing. Efrain Cruz Jiménez		Ing. de Software	ECJ
Ing. Fernando Ruiz Salazar		Ing. de Software	FRS
Lic. Baraquiel Martínez Ochoa	Emiliano Zapata	Subdirector	BMO

Agenda
<ul style="list-style-type: none"> • Presentación y descripción de los servicios que los Ingenieros de Software ofrecen a sus clientes. • Presentación de la situación actual en la secundaria Emiliano Zapata • Descripción del servicio/producto que el Director de la Secundaria Emiliano Zapata desea obtener.

Actividades de la reunión
<ol style="list-style-type: none"> 1. Se proyectó una presentación (formato Power Point) mostrando los servicios que como empresa, los ingenieros de software ofrecen a sus clientes y los beneficios que de sus productos se pueden obtener. 2. El Subdirector de la Secundaria Emiliano Zapata comentó en breve la historia la Secundaria y a continuación describió los procesos que actualmente se llevan a cabo dentro de la institución, haciendo énfasis en el proceso de control de acceso y asistencia de los alumnos. Se hizo mención sobre los conflictos que se han presentado respecto a las inasistencias de los alumnos a lo largo de la trayectoria de la institución, debidos a la falta de un proceso definido de administración de las mismas. 3. Conforme a la presentación de la situación actual, el Subdirector comentó sobre las expectativas que se tienen respecto de un producto que desean obtener, el cual les deberá ayudar a tener un mejor control de las asistencias e inasistencias de los alumnos. Este sistema deberá contar, según las especificaciones dadas por el Subdirector de la institución con las siguientes características: <ol style="list-style-type: none"> a. Fácil navegación. b. Accesible a todos los involucrados en los procesos por automatizar c. Interfaz amigable. d. Capaz de ser utilizado por personal no especializado en el área de la computación. e. Módulo de reportes específicamente definidos. f. Envío de notificaciones tanto a los administrativos de la institución, así como a los padres de familia de los alumnos. g. Contar con una base de datos robusta que permita tener de forma segura la información que en ella se estará almacenando. h. El sistema deberá contar con seguridad de accesos, a fin de contar con una herramienta confiable.

Compromisos				
No	Compromisos	Responsable	Fecha	Estatus (Pendiente Resuelto Cancelado)
1	Con base a las especificaciones dadas por el personal de la secundaria Emiliano Zapata, los ingenieros de software generarán una propuesta técnica de la solución	ECJ / FRS	29/Feb/08	Pendiente
2	La propuesta técnica será enviada vía correo electrónico al personal de la secundaria Emiliano Zapata.	ECJ / FRS	29/Feb/08	Pendiente
3	El personal de la secundaria Emiliano Zapata revisará la propuesta técnica enviada por los ingenieros de software y generará una lista con las observaciones, correcciones y/o cambios correspondientes.	BMO	05/Mar/08	Pendiente
4	Los ingenieros de software se encargarán de revisar y atender las correcciones, observaciones y/o cambios hechos a la propuesta generada, para generar una propuesta final definitiva y presentarla en la siguiente reunión	ECJ / FRS	14/Mar/08	Pendiente

Observaciones

Se mantendrá contacto telefónico entre el cliente y los desarrolladores, a fin de aclarar posibles dudas existentes sobre el desarrollo por llevar a cabo.

Para el caso en que existieran nuevos requerimientos, estos serán notificados a los ingenieros de software con un plazo no menor a 1 semana antes de la próxima reunión.

Próxima Reunión

Viernes 14 / Mar / 2008

MINUTA DE TRABAJO No. 2

Proyecto:	Sistema de Cómputo en Línea para Control de Asistencias	Fecha:	14/Mar/2008
Lugar:	Escuela Secundara Emiliano Zapata	Duración :	1.5 hrs
Objetivo:	Revisión y validación de avances en el desarrollo del sistema	Elaboró:	FRS

Favor de leer la minuta y en caso de tener comentarios, indicarlos en un lapso no mayor a 5 días hábiles posteriores a la fecha de recepción de la minuta, una vez transcurrido este lapso, se dará por aceptada la minuta y pasará a formar parte del acervo documental del proyecto.

Asistentes			
Ing. Efrain Cruz Jiménez		Ing. de Software	ECJ
Ing. Fernando Ruiz Salazar		Ing. de Software	FRS
Lic. Idalia Martínez Arreguín	Emiliano Zapata	Prefecta	IMA
Lic. Baraquiel Martínez Ochoa	Emiliano Zapata	Subdirector	BMO

Agenda
<ul style="list-style-type: none">• Se revisaron los puntos de la minuta generada el día 08 de Febrero de 2008• Se hizo un levantamiento de nuevos requerimientos.• Se revisaron los avances en el desarrollo del sistema y se establecieron acuerdos para dar cumplimiento a las metas planteadas al inicio del proyecto.

Actividades de la reunión
<ol style="list-style-type: none">1. Se comentó sobre la propuesta que se hizo llegar al personal de la escuela secundaria Emiliano Zapata, siendo un punto importante la implementación de un sistema de credencialización, ya que al implicar un gasto extra para los padres de familia se puede ver en riesgo el éxito del proyecto.2. Respecto de las inquietudes de la institución acerca de la viabilidad del proyecto, se aclararon puntos como el costo promedio de manufactura y creación por cada credencial, así como el precio que se podría manejar para su venta.3. Para el proceso de credencialización, se solicitó que exista un proceso manual de carga de asistencia de los alumnos para el caso en que la credencial no sea detectada por el sensor, así como un plan de contingencia para el caso en que el sistema llegase a fallar.4. Se solicitó que la credencial cuente con la fotografía del estudiante y que cuente con código de barras para poder ser leído por el dispositivo correspondiente (lector de código de barras).5. Se solicitó que previo a la implementación se lleve a cabo una prueba piloto del sistema con un grupo no mayor a 15 alumnos, entre los cuales, para fines de pruebas podría estar incluido equipo de trabajo administrativo de la institución.6. Se dio fecha de implementación del sistema en internet para que el personal de la institución pueda dar un seguimiento más tangible y continuo al proyecto.

Compromisos				
No	Compromisos	Responsable	Fecha	Estatus (Pendiente Resuelto Cancelado)
1	Se implementará en el módulo de credencialización el código de barras	ECJ / FRS	22/Mar/08	Pendiente
2	El sistema será implementado en un servidor web para que pueda ser consultado desde cualquier computadora con acceso a internet	ECJ / FRS	28/Mar/08	Pendiente
3	Una vez que el Subdirector de la secundaria Emiliano Zapata valide el buen funcionamiento del sistema se llevará a cabo la capacitación al personal acerca del uso y beneficios del sistema	ECJ/FRS	Por Definir	Pendiente
4	Al término de la capacitación del sistema, en un plazo no mayor a 2 semanas se llevará a cabo la prueba piloto del sistema.	ECJ / FRS	Por Definir	Pendiente

Observaciones

Una vez que sea implementado el sistema en internet, se enviarán los accesos al mismo vía correo electrónico al Lic. Baraquiel Martínez Ochoa.

Por cada cambio sustantivo que se lleve a cabo al sistema se enviará una notificación por correo electrónico al Lic. Baraquiel Martínez Ochoa para su respectiva aprobación.

Próxima Reunión

Viernes 04 / Abril / 2008

MINUTA DE TRABAJO No. 3

Proyecto:	Sistema de Cómputo en Línea para Control de Asistencias	Fecha:	04/Abril/2008
Lugar:	Escuela Secundara Emiliano Zapata	Duración :	2 hrs
Objetivo:	Revisión y observaciones finales para la puesta a punto del sistema	Elaboró:	FRS

Favor de leer la minuta y en caso de tener comentarios, indicarlos en un lapso no mayor a 5 días hábiles posteriores a la fecha de recepción de la minuta, una vez transcurrido este lapso, se dará por aceptada la minuta y pasará a formar parte del acervo documental del proyecto.

Asistentes			
Ing. Efrain Cruz Jiménez		Ing. de Software	ECJ
Ing. Fernando Ruiz Salazar		Ing. de Software	FRS
Lic. Baraquiel Martínez Ochoa	Emiliano Zapata	Subdirector	BMO

Agenda
<ul style="list-style-type: none"> • Se asignaron fechas para la capacitación y prueba piloto del sistema. • Se establecieron los lineamientos de la capacitación y la prueba piloto. • Se hicieron las últimas observaciones al sistema.

Actividades de la reunión	
1.	Con base a las revisiones que se estuvieron haciendo al sistema, se establece que la fecha para la capacitación debe corresponder a un día sábado para no interrumpir las actividades laborales del personal administrativo de la institución.
2.	La prueba piloto tendrá una duración de tres semanas, al término de las cuales se tendrá una reunión para reunir todas las observaciones y posibles problemas que el sistema deberá corregir previo a su implementación.
3.	Se hicieron observaciones sobre la información que debe tomarse en cuenta en los reportes generados por el sistema, ya que la información que actualmente se muestra no es del todo útil.

Compromisos				
No	Compromisos	Responsable	Fecha	Estatus (Pendiente Resuelto Cancelado)
1	Se capacitará a un grupo no mayor a 7 personas de la secundaria Emiliano Zapata en cuanto al correcto uso y administración del sistema. La capacitación durará 3 horas.	ECJ / FRS	03/May/08	Pendiente
2	Se llevará a cabo la prueba piloto del sistema únicamente con personal administrativo de la institución durante 10 días efectivos	ECJ / FRS / BMO	05/May/08 - 23/May/08	Pendiente
3	De las observaciones que llegasen a existir después de la prueba piloto, se corregirán los posibles errores detectados en un periodo no mayor a 2 semanas.	ECJ/FRS	26/May/08 - 06/Jun/08	Pendiente

Compromisos				
4	Una vez corregidas y atendidas todas las observaciones generadas por la prueba piloto, se llevará a cabo durante un día una sesión de pruebas exhaustivas a fin de poder asegurar un nivel de madurez del sistema. (en sábado)	ECJ / FRS / BMO	14/Jun/08	Pendiente
5	El personal de la secundaria previamente capacitado llevará a cabo la carga por completo en la Base de Datos de toda la información mínima necesaria para el funcionamiento del sistema (datos demográficos y académicos del alumno y padres de familia)	BMO	09/Jun/08 – 27/Jun/08	Pendiente
6	Se llevará a cabo la credencialización de alumnos (fotografía, generación de claves y sensores)	BMO	16/Jun/08 – 27/Jun/08	Pendiente
7	Durante el periodo vacacional será implementada toda la infraestructura necesaria para el funcionamiento del sistema (red, sensores, impresora de credenciales, servidor y terminales)	ECJ / FRS	07/Jul/08 – 09/Ago/08	Pendiente
8	Se dará inicio con la actividad diaria del sistema con todos los alumnos registrados durante la credencialización.	BMO / ECJ /FRS	18/Ago/08	Pendiente

Observaciones
<p>Una vez que sea implementado el sistema en internet, se enviarán los accesos al mismo vía correo electrónico al Lic. Baraquiel Martínez Ochoa.</p> <p>Por cada cambio sustantivo que se lleve a cabo al sistema se enviará una notificación por correo electrónico al Lic. Baraquiel Martínez Ochoa para su respectiva aprobación.</p>

Próxima Reunión
Viernes 12 / Septiembre / 2008

2.1.4 Solución Propuesta

Basada en el análisis realizado a la problemática presentada y los requerimientos solicitados, se presenta a continuación la descripción general de la solución propuesta.

- ❖ Se implementará una red local dentro de la institución. Contará con un servidor dedicado y una serie de terminales.
- ❖ Se desarrollará utilizando el lenguaje PHP y JavaScript un módulo de administración de alumnos, en el que serán capturados y almacenados todos los datos personales necesarios del alumno.
- ❖ Se implementará una base de datos relacional, utilizando como manejador de base de datos SQL Server 2000.
- ❖ Serán adquiridos dispositivos para la expedición de credenciales, tales como impresora de credenciales y sensores de proximidad
- ❖ Se llevará a cabo un proceso de credencialización para todos los alumnos inscritos en la institución. Las credenciales que se generen serán de proximidad. Dichas credenciales contarán con fotografía del alumno y deberán presentarse en un sensor a una distancia no mayor a 15 centímetros, para que sea registrada en la base de datos la entrada del alumno.
- ❖ Se llevará a cabo un corte a una hora determinada para dejar de realizar registros de acceso de alumnos. Inmediatamente después de dicho corte, se llevará a cabo automáticamente el proceso de notificación vía SMS a los celulares de los padres de familia que hayan contratado el servicio.
- ❖ Se desarrollará un módulo de reportes, en el cual el personal administrativo de la institución podrá consultar información específica que se haya almacenado durante los accesos de los alumnos.
- ❖ Se desarrollará un portal Web de consulta para los padres de familia. Este reporte será en línea y en tiempo real, el tutor podrá consultar el historial de asistencias de su(s) hijos(s) eligiendo un periodo de fechas determinado.

2.1.4.1 Implementación De La Tecnología Necesaria

Serán adquiridos los siguientes equipos:

- ❖ 1 servidor
Suficiente para soportar todas las aplicaciones que será necesario instalar, así como las exigencias de rendimiento que las solicitudes generadas desde la aplicación desarrollada se requieran cubrir.
- ❖ 3 terminales (PC)
Para uso exclusivo del personal administrativo encargado de administrar la solución a desarrollar. Estos equipos pertenecerán a una red local que será necesario instalar, a fin de poder establecer más de un centro de captura y/o administración del sistema por desarrollar, incluyendo actividades como credencialización.
- ❖ 3 sensores de proximidad
Cantidad mínima necesaria para poder dar acceso a aproximadamente 480 alumnos en un lapso de 20 minutos.
- ❖ 1 impresora de credenciales de proximidad
Necesaria para la implementación de un módulo de credencialización que brinde las credenciales para minteractuar con los sensores elejidos para la solución de registro de asistencia.
- ❖ 1 switch de 16 puertos
Necesario para la implementación de una red local dentro de la institución, comprendida por las 3 terminales y el servidor, siendo a través de esta más rápidos los accesos a la aplicación por desarrollar.
- ❖ 1 celular con conexión USB y que funcione como módem.

Necesario para el envío de las notificaciones vía SMS. La conexión USB es la más común hoy en día en los equipos de telefonía móvil y las computadoras de escritorio, así como las portátiles.

2.1.4.2 Módulo De Administración De Alumnos Y Tutores

Se desarrolla con el lenguaje de programación PHP un módulo de administración de alumnos y tutores. En este módulo se captura la información necesaria tanto de los alumnos como de los padres de familia. Toda la información solicitada en los formularios de captura será validada y obligatoria, a fin de evitar que se capture información incompleta y evitar en la medida de lo posible que sea errónea.

Previa a la captura de esta información, el alumno deberá tener asignado un grupo, turno y se deberá saber en que grado se encuentra inscrito. La información que sea generada en el día a día de los accesos de los alumnos será asociada a la que en este módulo se capture.

Existirá una pantalla en la que pueda ser consultada la información que se ha capturado, mostrando todos los alumnos registrados, el total de inasistencias que se han presentado en el actual ciclo escolar, el total de envíos de SMS realizados, así como una estadística del total de accesos que han realizado los padres de familia al portal Web de consulta.

2.1.4.3 Credencialización

Se tomará una fotografía de cara completa de cada uno de los alumnos de la institución. Se asociará esta fotografía al registro correspondiente generado durante la captura de datos de cada alumno.

Utilizando la impresora de credenciales, se imprimirán todas las credenciales generadas y se les asignará un sensor. Posteriormente en una fecha determinada estas serán entregadas al alumno correspondiente.

2.1.4.4 Módulo de reportes

Una vez que se comience con el registro de acceso de los alumnos será necesario poder ver toda la información almacenada en la base de datos en una pantalla amigable y de fácil navegación. Ya que mediante dicha pantalla los administrativos de la institución podrán analizar el comportamiento de las asistencias de los alumnos en los diferentes turnos, grados y grupos, tanto para un día, así como para un periodo de tiempo determinado.

Los reportes a desarrollar serán los siguientes:

- ❖ Reporte de alumnos inscritos
- ❖ Reporte de alumnos que aún no tienen asignado un sensor
- ❖ Reporte de alumnos que pagaron el servicio de SMS y no tienen celular
- ❖ Reporte de inasistencias por cantidad
- ❖ Reporte de inasistencias del día actual por turno (matutino y vespertino)

Los datos que se desplegarán en los reportes son:

- ❖ Nombre completo del alumno
- ❖ Grado y grupo
- ❖ Turno
- ❖ Matrícula
- ❖ Sensor asignado

2.1.4.5 Portal Web De Consulta Para Los Tutores

Hasta el momento, todos los reportes desarrollados tienen como usuario final al personal administrativo de la institución, sin embargo se ha pensado que los padres de familia querrán consultar vía web la información que día a día el registro de sus hijos genera.

Para cubrir con esta necesidad, será desarrollado utilizando el lenguaje de programación PHP un módulo de consulta para los padres de familia. Para poder ingresar a la información almacenada en la base de datos, se asignará a los padres de familia un usuario y una contraseña. Los padres de familia sólo podrán consultar la información de su(s) hijo(s).

La pantalla inicial del reporte, mostrará el nombre del padre de familia al que le fueron asignados el usuario y contraseña, así como el registro para el día actual. Se podrá hacer una consulta dentro de un periodo de tiempo determinado por el padre de familia, pudiendo elegir la fecha de inicio y fecha final del reporte. Dentro del reporte se mostrará únicamente el nombre del alumno, la fecha del registro y la hora de entrada. Para el caso en el que el alumno no se haya presentado a clases en alguno de los días que se hizo la consulta se presentará la leyenda "Sin registro".

2.1.4.6 Seguridad de la Solución

Toda la información personal, tanto de alumnos como de tutores, almacenada en la base de datos de la solución propuesta se encontrará físicamente en el servidor que está bajo resguardo de la dirección de la institución, siguiendo con el esquema actual del almacenamiento de expedientes impresos de información de los alumnos.

Así mismo, el acceso al portal será únicamente con un usuario y contraseña generados durante el registro de los tutores, esta información de acceso será encriptada en la base de datos utilizando MD5 (abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) que es un algoritmo de reducción criptográfico de 128 bits ampliamente utilizado y que brinda elevados niveles de seguridad ya que toda cadena que es encriptada en MD5 no se puede volver a su forma original.

CAPÍTULO 3

FLUJO FÍSICO Y

COMPONENTES

CAPÍTULO 3 FLUJO FÍSICO Y COMPONENTES.

3.1. Generalidades

La información de las asistencias/inasistencias viaja de los sensores al servidor en él se procesa la información y se envían las notificaciones vía SMS o en el portal web para padres o tutores, los clientes pueden acceder al sistema a través de la red local y utilizar recursos compartidos como la impresora de credenciales y compartir archivos.

El diseño del “Sistema De Cómputo En Línea Para Control De Asistencias” se describe bajo la estructura mostrada en la figura 3.1:

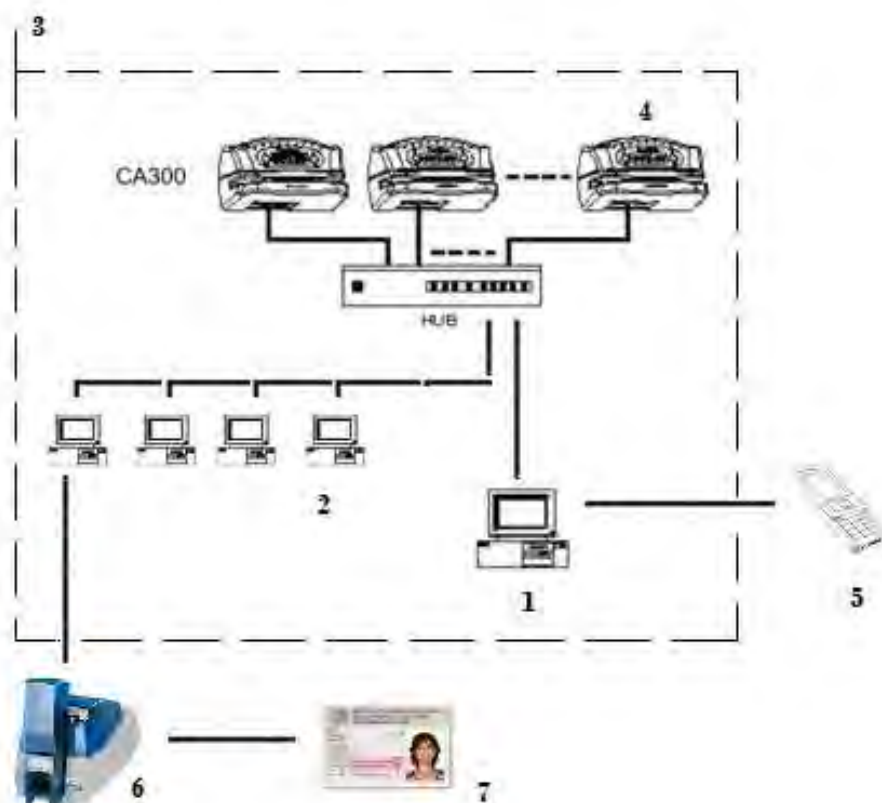


FIG. 3.1 Diagrama del “Sistema De Cómputo En Línea Para Control De Asistencias”.

- Servidor
- Clientes (Terminales)
- Red
- Sensores CA300
- SMS
- Impresora Datacard SP55
- Credencial

A continuación se describen las características y/o requisitos mínimos solicitados de cada uno de los componentes del sistema para su implementación:

3.2 SERVIDOR

3.2.1 Requerimientos Mínimos

Requerimientos para un óptimo desempeño

- Procesador de 1.7 GHz 32-bit o uno de 64-bit
- 4 GB de RAM.
- 40 GB de capacidad en disco con 15 GB libres.
- Acceso a Internet
- Tarjeta de red Ethernet
- Sistema Operativo Windows Server 2000

3.2.2 Instalación de servidor PHP

WAMP

Existe una manera de comenzar a utilizar PHP5 en Windows sin tener que sufrir las complicaciones típicas de la instalación de los servidores necesarios para programar en PHP. Se trata de instalar un paquete llamado WAMP, que permite instalar y configurar en un solo proceso el servidor Apache, la base de datos MySQL y el módulo de programación en PHP.

WAMP es un sistema indicado para los usuarios que no tienen instalado en el sistema ninguno de los programas necesarios para programar en PHP (Apache, PHP y MySQL), ya que realiza una instalación de PHP5 completa y desde cero. Pero también pueden utilizar este programa los usuarios que disponen de Apache, PHP y/o MySQL en su sistema. En cuyo caso, simplemente se realizará otra copia de las aplicaciones en un directorio distinto, que en principio, no tiene por qué interferir con las otras instalaciones alojadas en nuestro equipo.

El software que se instala con WAMP5 contiene los siguientes servidores y programas:

- Apache 1.3.31.
- PHP5. El motor renovado del lenguaje.
- MySQL. La base de datos más extendida para utilizar con PHP.
- PHPmyadmin. Un software que permite administrar una base de datos a través de una interfaz web.
- SQLitemanager. Un sistema para administrar una base de datos a partir de sentencias SQL.

Instalación de WAMP

La instalación se realiza a través de un ejecutable Windows donde se pueden introducir muy pocas configuraciones, apenas el directorio donde deseamos que se instalen los programas. Después del proceso de instalación se habrán creado un par de servicios con el servidor web (Wamp Apache) y el de bases de datos (Wamp Mysql).

Además, dentro del directorio donde hayamos instalado WAMP se habrá creado una carpeta llamada "www", que corresponde con el directorio de publicación, es decir, el lugar donde se deben colocar las aplicaciones web.

Durante la instalación también debemos decidir si deseamos que WAMP5 se inicie automáticamente al arrancar el equipo o si deseamos que su puesta en marcha se realice manualmente.

Puesta en marcha de los servicios

Cuando se instala WAMP se crea un grupo de programas llamado WampServer, donde podremos encontrar una opción que pone "Start Wampserver", que será necesario ejecutar si no hemos seleccionado que

el servidor se inicie automáticamente.

Una vez arrancado WampServer dispondremos de un icono en la barra de tareas con una forma similar a la de un marcador de velocidad. Si pulsamos sobre ese icono se abrirá un menú con opciones variadas para gestionar los servicios relacionados con el paquete. La figura 3.2 muestra ese menú.

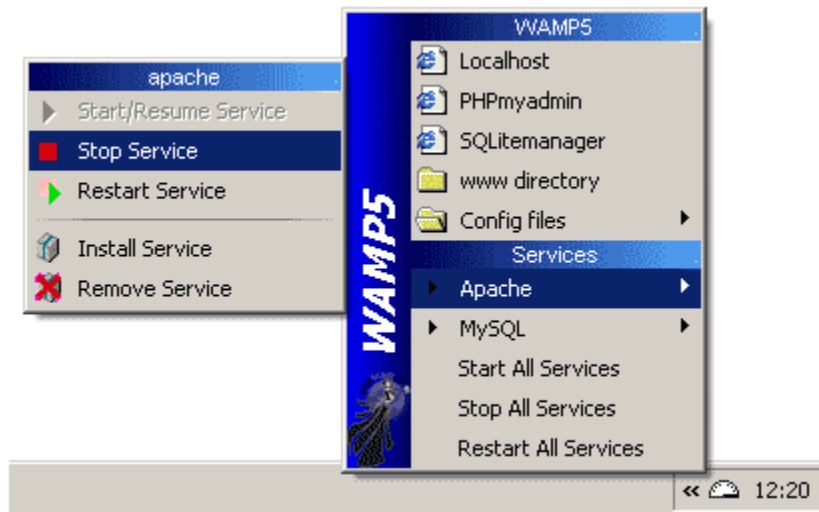


FIG. 3.2 Menú de administración de servicios de PHP.

Así mismo, se genera una serie de directorios en el disco duro local, dentro de los cuales se deberá de almacenar todo el código que se desea procesar con el servidor Apache de PHP. La figura 3.3 muestra el directorio creado.

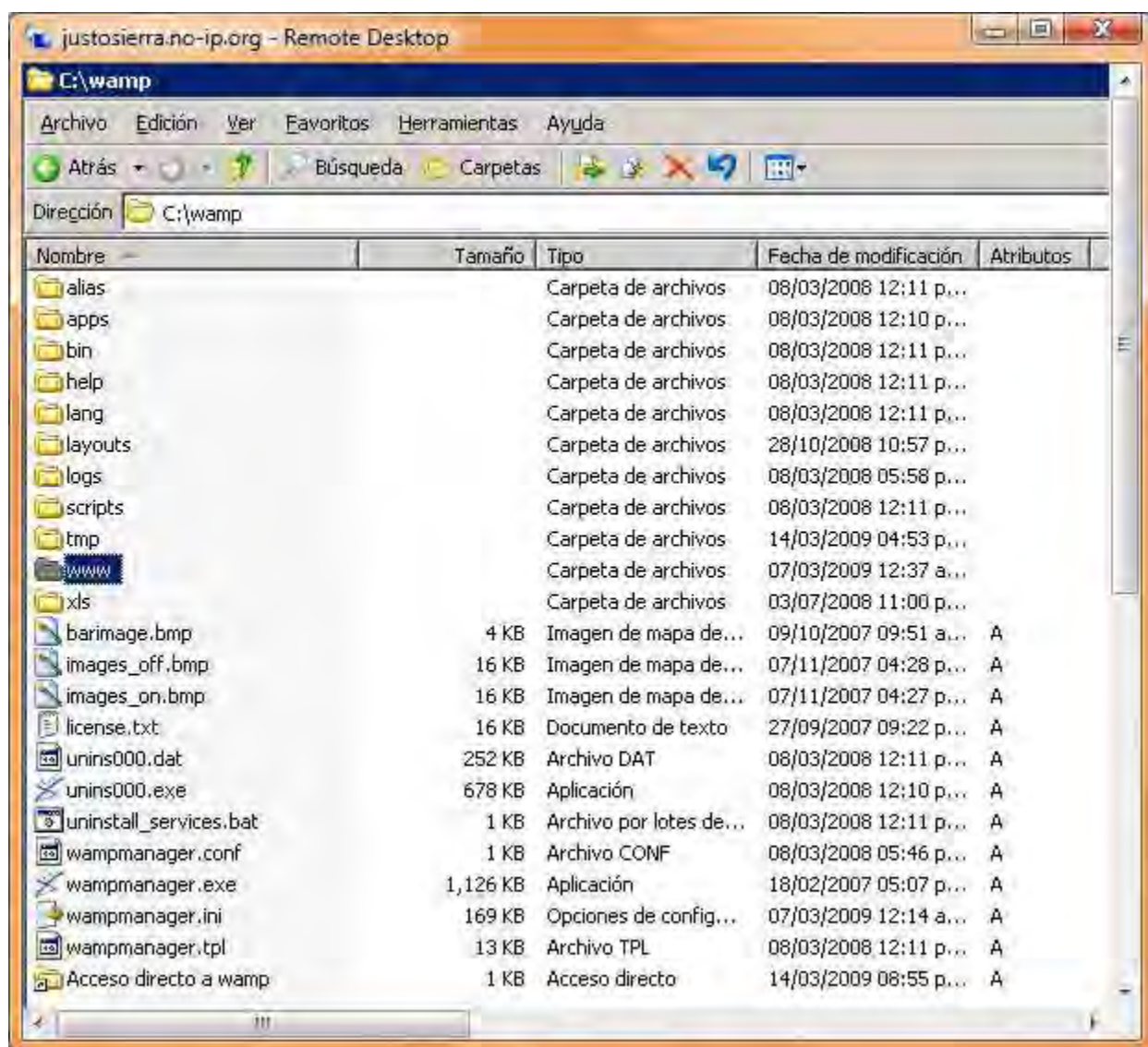


FIG. 3.3 Ruta del directorio que almacena los archivos del sistema en el servidor.

Podemos probar si los servicios están corriendo perfectamente accediendo a la página de inicio del servidor, escribiendo en la barra de direcciones del navegador de internet algo como <http://localhost/>.

Entonces nos debería de aparecer una página con varios enlaces a las distintas herramientas instaladas con WAMP, además de algunas páginas de prueba de PHP.

3.2.2.1 INSTALACIÓN DEL SISTEMA DE CÓMPUTO EN LÍNEA PARA CONTROL DE ASISTENCIAS

Una vez que tenemos instalado en nuestro servidor el aplicativo para PHP procederemos a depositar la carpeta de desarrollo "tesis" dentro del fichero "C:\wamp\www"

3.2.3 INSTALACIÓN DE SQL SERVER 2000 Y SERVIDOR SQL

Instalación del SQL Server Enterprise

Instalar SQL Server es muy sencillo, dispone de un asistente que seguiremos para realizar la instalación proporcionando unos datos que se nos pedirá a lo largo del proceso. El proceso es sencillo y hay que tener en cuenta durante la instalación las opciones que se necesitan instalar de SQL Server,

¿Qué es el Nivel de Intercalación? O Collation Settings en inglés, es la forma que SQL Server tratará los caracteres de un idioma, el orden para utilizar los tipos de datos y la página de códigos.

Para instalar el Microsoft SQL Server 2000 Enterprise hay que seguir las instrucciones como lo indican los pasos de instalación del 1 al 9, mencionados más adelante.

Después de ejecutarse el fichero autorun.exe seleccionaremos de dos ventanas previas a la instalación: Componentes de SQL Server 2000 -> Instalar Servidor de bases de datos.

SQL Server 2000 puede instalarse de dos formas diferentes o mejor dicho pueden instalarse varias instancias en una misma máquina, esto es muy útil si por ejemplo tenemos un SQL Server 7.0 y queremos instalar el 2000, podemos hacerlo y además podemos tenerlos ejecutándose en la misma máquina los dos al mismo tiempo.

PASOS DE INSTALACIÓN:

Paso 1: Pide el nombre del equipo y la forma de instalación, si es local o remota, si es remota deberemos introducir desde que equipo lo vamos a instalar.

Paso 2: Seleccionar una opción de instalación, instalar una nueva instancia del SQL Server, instalar las partes clientes u opciones avanzadas donde podremos crear instalaciones desatendidas o reconstruir el registro del SQL Server si estuviera dañado. La opción central solo estará activa si ya tenemos un SQL Server instalado y queremos modificar su instalación.

Paso 3: Herramientas que instalaremos, es decir, si instalamos solo las herramientas de cliente para tener acceso a un servidor remoto, herramientas de cliente y servidor como gestor de bases de datos o solo conectividad que instala únicamente el MDAC.

Paso 4: Nombre de la instancia que vamos a crear, si es nueva instancia (no hay ningún SQL Server instalado) por defecto toma el nombre de la máquina aunque podemos cambiarlo, si es una segunda instalación debemos darle un nuevo nombre.

Paso 5: Tipo de instalación, personalmente recomiendo una instalación personalizada por lo comentado anteriormente es importante escoger el juego de caracteres para no tener problemas en futuras instalaciones y no tener que ir tocando el nivel de intercalación en cada base de datos que creamos, y elegir la ruta de los datos a una unidad con suficiente espacio en disco.

Paso 6: Selección de componentes a instalar, entre ellos ejemplos y ayuda (manual de SQL Server).

Paso 7: Usuario que ejecutará los servicios del SQL Server y de SQL Agent, por defecto lo ejecuta el usuario administrador.

Paso 8: Modo de autenticación a SQL Server, se puede elegir entre autenticación windows (la autenticación se realiza por medio de usuarios pertenecientes al dominio) o modo mixto que la autenticación se realiza por medio de usuarios dados de alta en el SQL Server.

Paso 9: Configuración regional, en esta opción es donde vamos a elegir la forma que nuestro SQL Server trabajará con los datos, es decir, elegiremos el nivel de intercalación. Es recomendable no dejarlo por defecto y seleccionar un nivel de intercalación apropiado a nuestro lenguaje como por ejemplo el que muestra la imagen, en la imagen selecciona un SQL Server que no distinguirá entre mayúsculas ni entre acentos, ¿qué significa esto? Que cuando se realicen búsquedas, consultas o

transacciones nos dará lo mismo poner "Tabla" que "tabla". Si en todos los servidores SQL Server de nuestra organización tenemos la misma opción de nivel de intercalación, cuando tengamos necesidad de restaurar bases de datos o moverlas de un SQL Server a otro será tan fácil como hacer una restauración.

Por último nos pedirá el puerto y las bibliotecas de red que utilizará. Este paso lo podemos dejar por defecto.

A partir de este punto SQL Server instalará las opciones seleccionadas y cuando finalice la instalación tendremos un SQL Server listo para trabajar.

Una vez instalado por completo, el SQL Server 2000 nos brinda por defecto unos servicios, bases de datos por defecto y utilidades que vamos a describir a continuación.

SERVICIOS DEL SQL SERVER

SQL Server 2000 instala varios servicios en nuestra computadora, el nombre del servicio se puede ver en el administrador de servicios de Windows desde Inicio -> Programas -> Herramientas de administración -> Servicios, desde donde se pueden parar, arrancar y cambiar el usuario que los ejecuta. También podemos utilizar una utilidad que instala SQL Server "El Administrador de Servicios de SQL Server".

- SQL Server: Es el servicio principal y arranca el SQL Server, se llama MSSQLSERVER y si hay varias instancias en una misma máquina, el servicio se llama MSSQL\$Nombre_de_instancia.

- SQL Server Agent: Agente de SQL Server, se utiliza para la programación de tareas, alertas, mail y se llama SQLSERVERAGENT. Igual que el anterior si hay múltiples instancias se llama SQLAGENT\$Nombre_de_instancia.

- Coordinador de transacciones distribuidas: DTC, utilizado para tareas entre varios servidores distribuidos. Se llama "Distributed Transaction Coordinator".

- Microsoft Search: Utilizado para la búsqueda de texto completo en las bases de datos e índices de texto, este servicio solo estará si se ha instalado esta utilidad. Se llama "Microsoft Search".

- Asistente del Active Directory.

BASES DE DATOS

SQL Server instala por defecto seis bases de datos, cuatro de sistema que utiliza el SQL Server: master, model, tempdb y msdb y dos de ejemplos para poder realizar pruebas. Las bases de datos se guardan en el directorio de instalación en MSSQL\Data, las nuevas bases de datos que se creen si no se especifica lo contrario también se crearan en ese directorio.

Master: Guarda información de todas las bases de datos que tiene SQL Server, gestiona usuarios, permisos y parámetros de configuración.

Model: Es la plantilla para crear nuevas bases de datos.

Tempdb: Es un espacio de trabajo temporal para procesar consultas y procesar tareas de SQL Server.

Msdb: Utilizada por el Agente de SQL (SQL Server Agent), mantiene datos de alertas, trabajos, notificaciones y tareas programadas.

HERRAMIENTAS DE SQL SERVER

Después de la instalación tenemos varias herramientas de administración y utilidades, entre ellas la más importante es el "Administrador corporativo" desde el cual podremos realizar todas las operaciones que deseemos o necesitemos sobre el SQL Server. A parte del administrador hay instaladas las siguientes herramientas:

- Libros en pantalla (la ayuda del SQL Server 2000)
- Administrador corporativo (Enterprise Manager).
- Administrador de servicios (Service Manager).
- Analizador de consultas (Query Analyzer).
- Analizador (Profiler).

- Configurar la compatibilidad con SQL XML en IIS.
- Herramientas de red de cliente (Client Network Utility).
- Herramientas de red de servidor (Server Network Utility).
- Importar y exportar datos.

3.2.4 Instalación de software para plataforma de SMS

Es necesario contar con el software J2EE 1.4.2 para Microsoft Windows, el cual se puede descargar desde: <http://java.sun.com/j2ee/1.4/download.html>. Una vez descargado es necesario iniciar el programa de instalación y seguir las instrucciones, tal como se ilustra a partir de la figura 3.4.

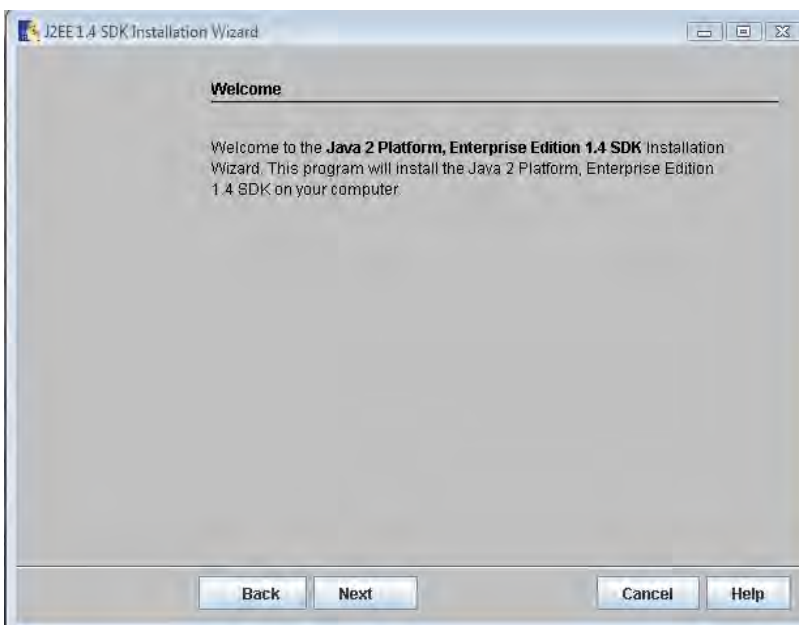


FIG. 3.4 Inicio de instalación de Java Enterprise Edition

Es necesario aceptar los términos de la Licencia del programa para poder continuar con la instalación. Para ello se debe seleccionar “Yes” y dar clic en “Next”. Ver figura 3.5.

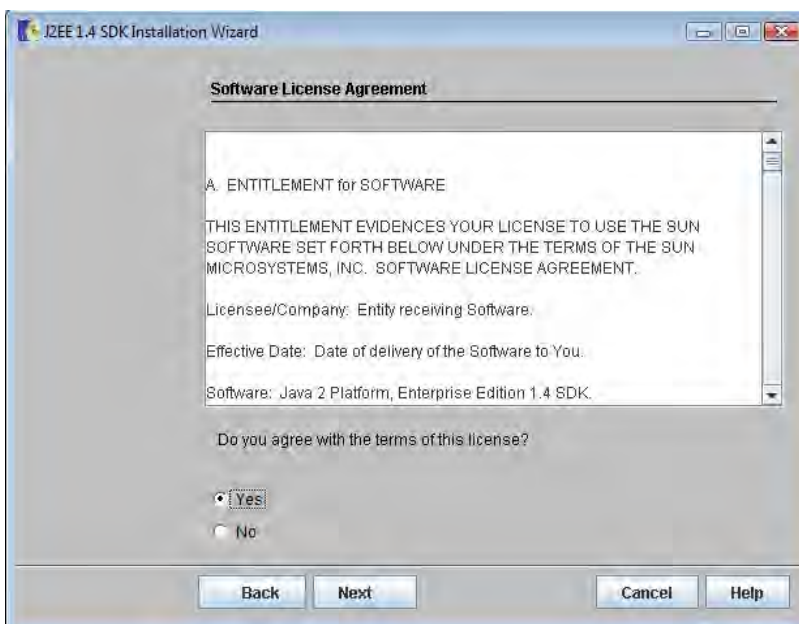


FIG. 3.5 Aceptación de términos y condiciones de la licencia del software.

Una vez aceptados los términos de la licencia, es necesario seleccionar el directorio en el que se desea se instale el software, tal como lo muestra la figura 3.6.

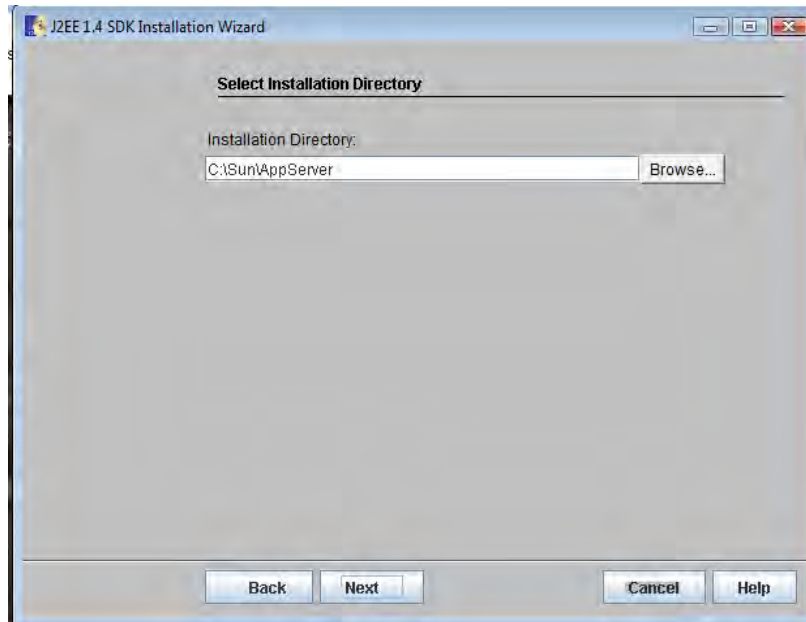


FIG. 3.6 Selección de directorio de instalación

En la figura 3.7 se muestra el ejemplo para cuando el directorio seleccionado no exista.

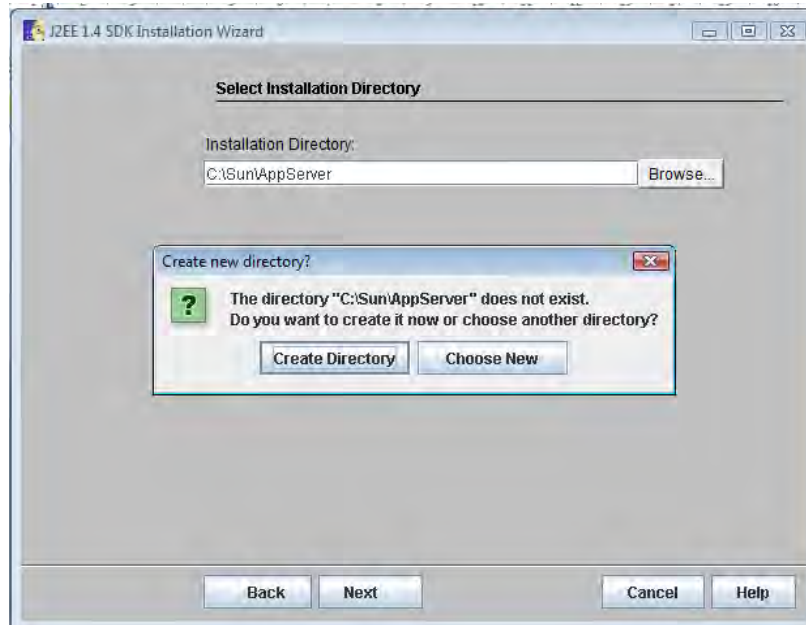


FIG. 3.7 En caso de no existir el directorio, el instalador lo creará.

Una vez creado el directorio, es necesario introducir la información de administración del programa, tanto para usuario como para los puertos de administración, llenando los campos mostrados en la figura 3.8.

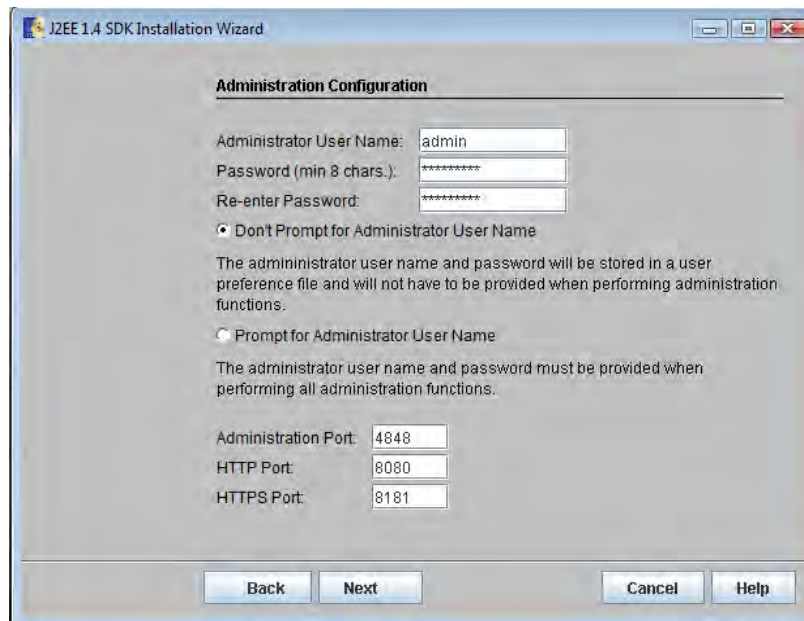


FIG. 3.8 Información de usuario y puertos de administración.

Tal como se muestra en la figura 3.9 se deberán seleccionar las opciones que se desea sean instaladas junto con el software, tales como ejemplos, actualizaciones, accesos directos, etc.

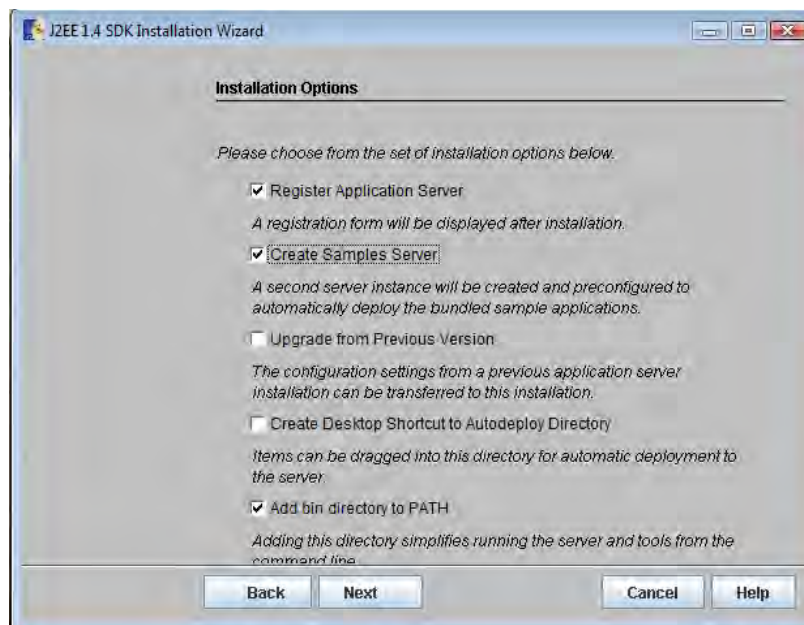


FIG. 3.9 Selección de opciones de instalación.

Cuando el instalador determina que existe suficiente espacio en el Disco Duro, simplemente hay que dar clic en "Install Now" para que la instalación de inicio con los parámetros que previamente hemos capturado. Figura 3.10.

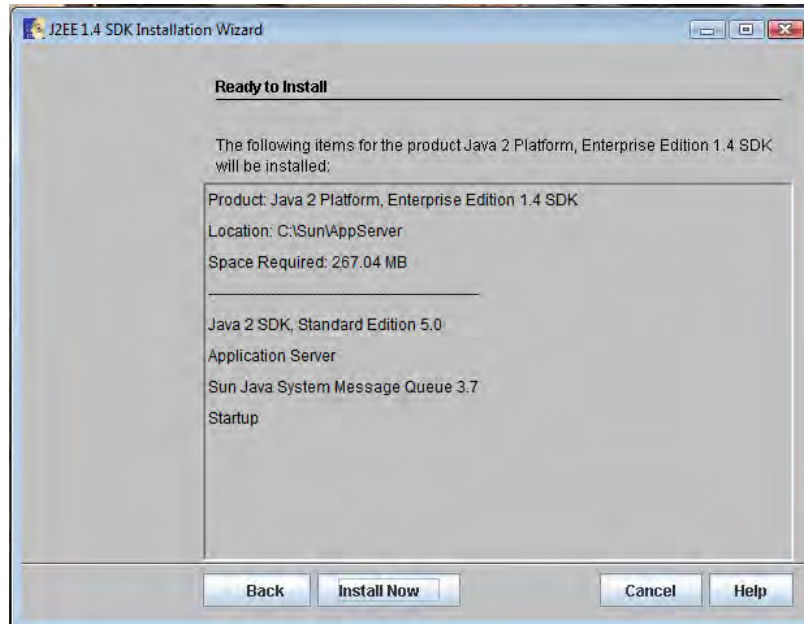


FIG. 3.10 Pantalla de programa listo para ser instalado.

Al dar clic en "Install Now" se mostrará la pantalla mostrada en la figura 3.11 que nos indica el progreso de la instalación.



FIG. 3.11 Pantalla de progreso de instalación

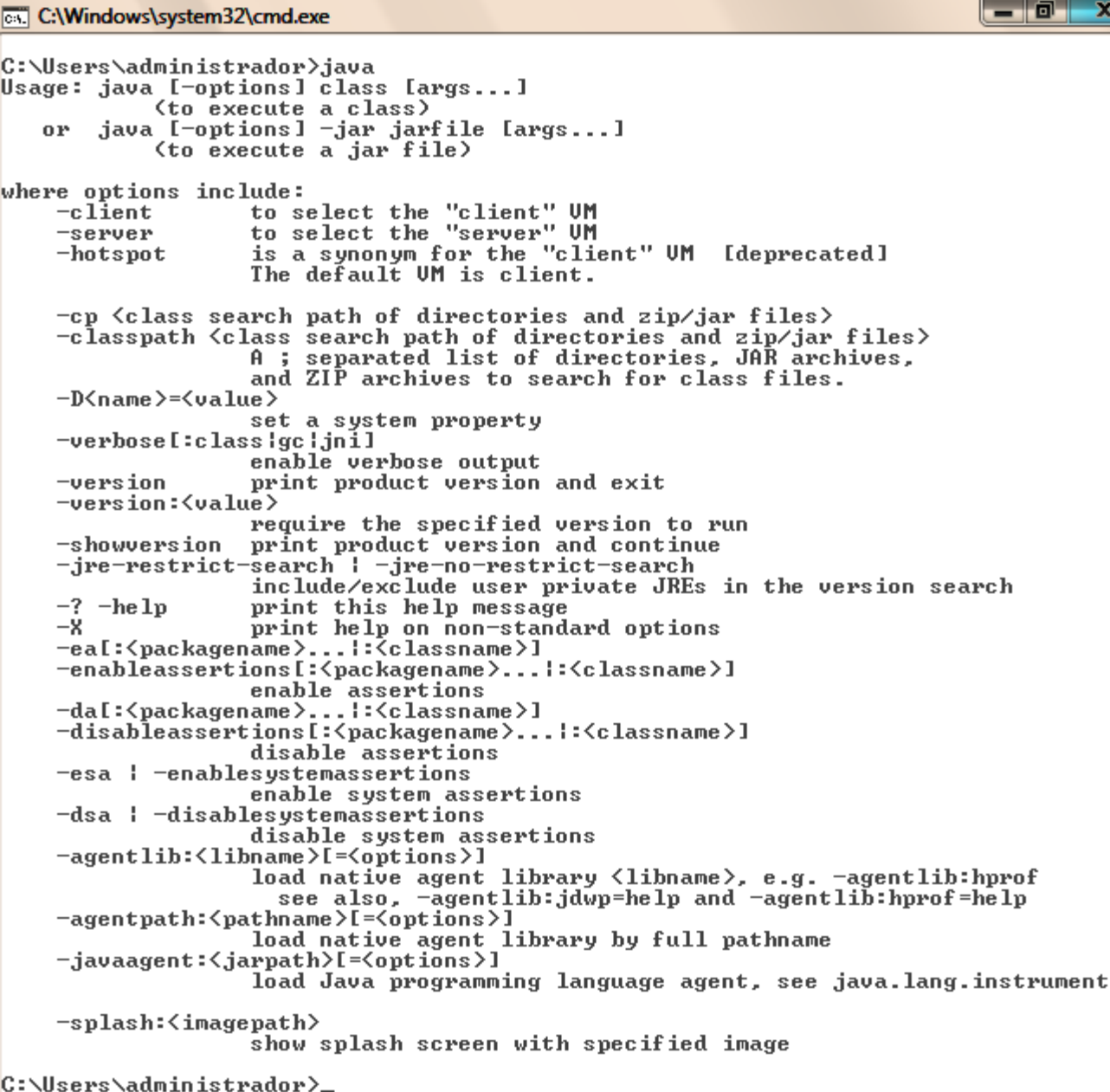
Cuando la instalación se ha completado se muestra la pantalla mostrada en la figura 3.12.



FIG. 3.12 Instalación completada

3.2.4.1 Pruebas de Instalación

Una vez instalado el programa por completo, bastará con abrir la ventana de comandos de Windows y teclear la palabra "java". Si la instalación ha sido exitosa, la pantalla mostrada será como la figura 3.13:



```
C:\Windows\system32\cmd.exe
C:\Users\administrador>java
Usage: java [-options] class [args...]
           (to execute a class)
   or java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
  -client          to select the "client" VM
  -server          to select the "server" VM
  -hotspot         is a synonym for the "client" VM [deprecated]
                  The default VM is client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
              A ; separated list of directories, JAR archives,
              and ZIP archives to search for class files.
  -D<name>=<value>
              set a system property
  -verbose[:class!gc!jni]
              enable verbose output
  -version         print product version and exit
  -version:<value>
              require the specified version to run
  -showversion    print product version and continue
  -jre-restrict-search | -jre-no-restrict-search
              include/exclude user private JREs in the version search
  -? -help        print this help message
  -X              print help on non-standard options
  -ea[:<packagename>...!:<classname>]
  -enableassertions[:<packagename>...!:<classname>]
              enable assertions
  -da[:<packagename>...!:<classname>]
  -disableassertions[:<packagename>...!:<classname>]
              disable assertions
  -esa | -enablesystemassertions
              enable system assertions
  -dsa | -disablesystemassertions
              disable system assertions
  -agentlib:<libname>[=<options>]
              load native agent library <libname>, e.g. -agentlib:hprof
              see also, -agentlib:jdwp=help and -agentlib:hprof=help
  -agentpath:<pathname>[=<options>]
              load native agent library by full pathname
  -javaagent:<jarpath>[=<options>]
              load Java programming language agent, see java.lang.instrument

  -splash:<imagepath>
              show splash screen with specified image

C:\Users\administrador>_
```

FIG. 3.13 Pantalla que muestra la instalación exitosa del software.

3.2.5 Instalación De Software De Relojes Checadores

Para iniciar con la instalación del módulo de relojes checadores en el servidor inserte el CD y siga las instrucciones.

1. Si el instalador no se inicio automáticamente, ejecútelo desde setup.exe
2. Siga los pasos del asistente, tal como se muestra a partir de la figura 3.14.

Instale los componentes previos.

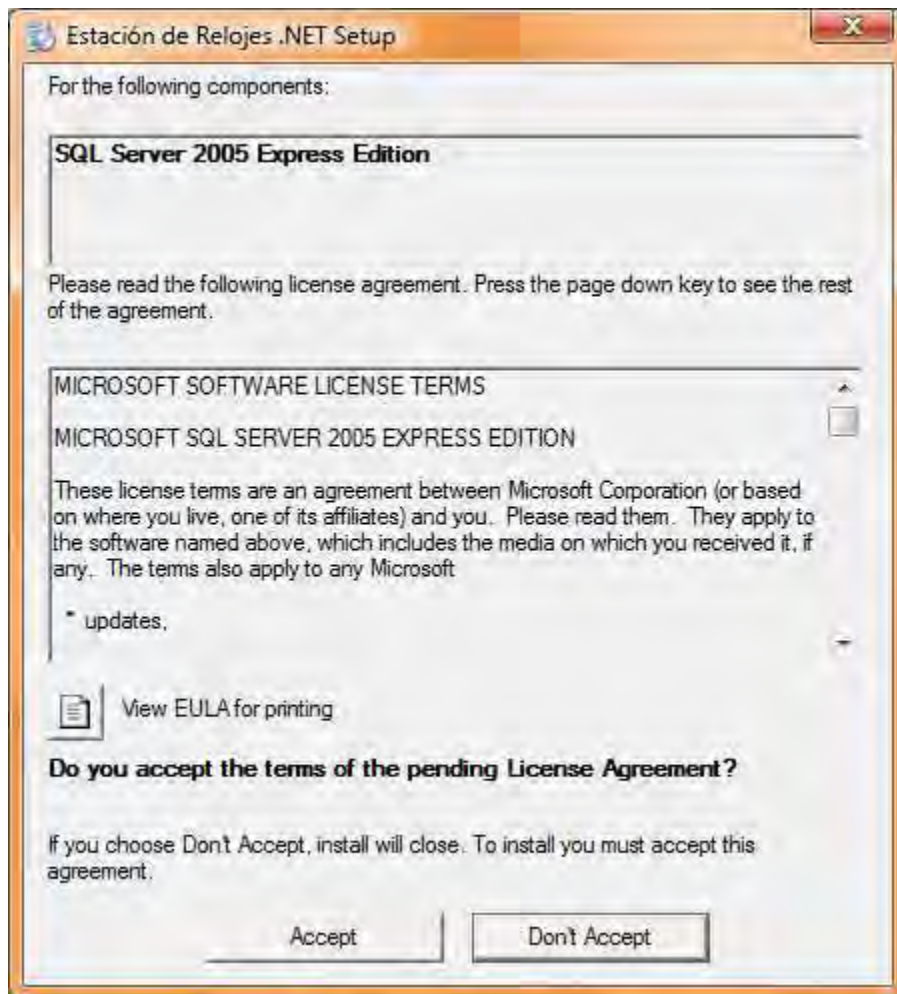


FIG 3.14 Primer pantalla de instalación de los sensores en el servidor

3. Una vez instalados los componentes previos. Continúe con los pasos del asistente de instalación del Modulo de relojes checadores.

Una vez instalado el software se debe proceder a reiniciar el servidor para que las instalaciones se vean reflejadas , una vez reiniciado aparecerá un icono en la parte inferior derecha este significa que la instalación ha sido exitosa. Ver figura 3.15.

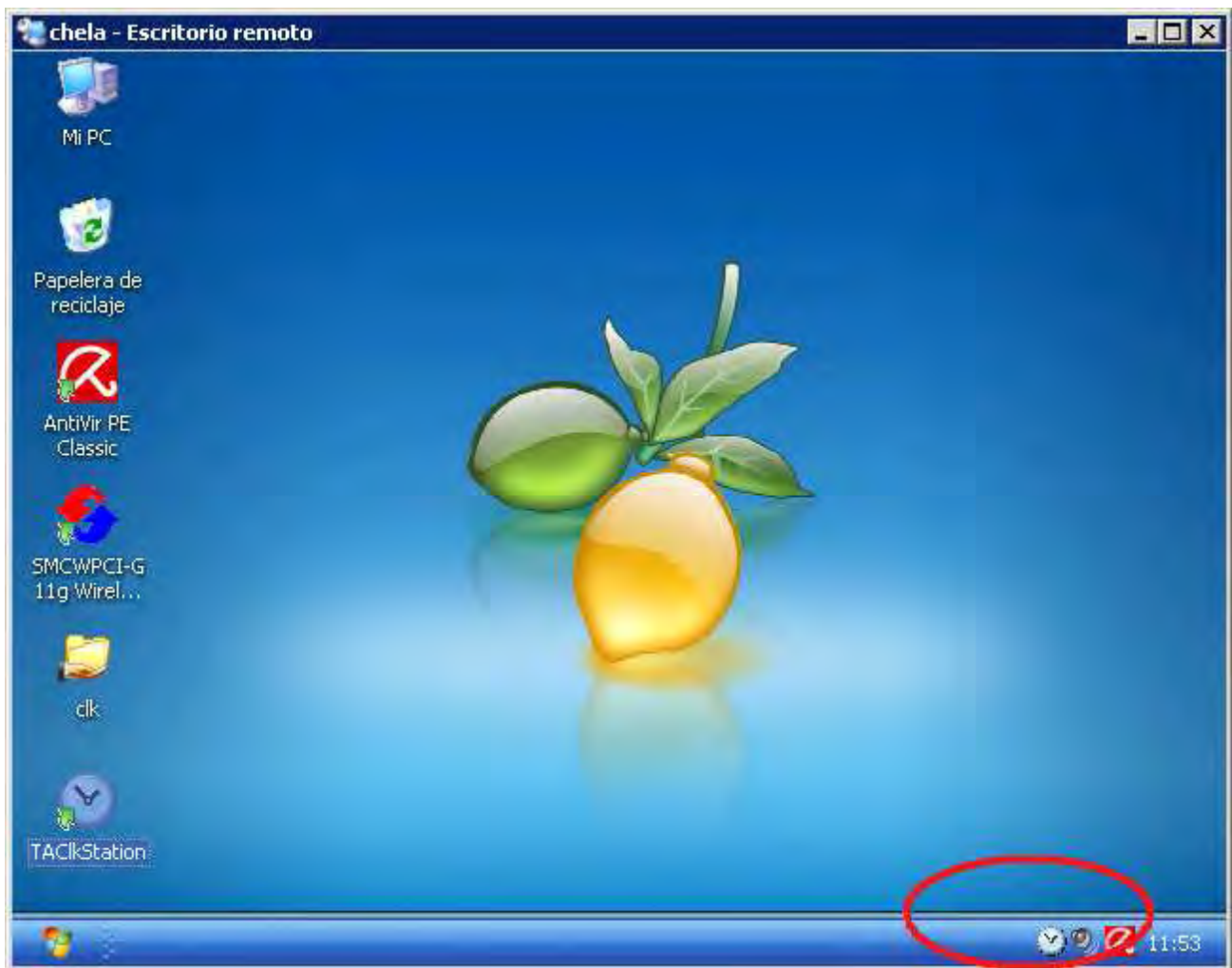


FIG 3.15 Iconos indicadores de instalación y operación de sensores.

3.2.6 Instalación De Servidor FTP

FileZilla Server es un servidor de FTP desarrollado para la plataforma Windows, válido para todas las versiones, desde Windows 95 hasta XP. Es un software libre bajo la licencia GNU General Public License (GPL) por lo que podemos tener la garantía de su eficacia y existencia futura.

Esta aplicación no está disponible en español, pero está considerada como uno de los servidores más fáciles de configurar. El sistema está compuesto por dos utilidades que permiten iniciar y parar el servidor en nuestra máquina. Además se complementa con una aplicación para la administración del servidor.

Las principales funcionalidades de este servidor FTP son:

- Permite bloquear y reanudar el servicio.
- Entre sus opciones principales permite cambiar el puerto, número máximo de conexiones y número de hilos de ejecución.
- Admite la personalización del mensaje de bienvenida.
- Opciones específicas de seguridad y auditoría de conexiones.
- Configuración de usuarios y grupos, así como límite de velocidad para cada uno de ellos (tanto para descargas como subir archivos).

- Configuración de límite máximo de conexiones por IP.

Una vez descargado el programa, deberemos ejecutar el archivo de instalación tan solo haciendo doble clic sobre él. Lo primero que debemos aceptar es la licencia GNU/GPL pulsando el botón "I Agree".

A continuación veremos una pantalla en la que se nos solicita el tipo de instalación. Podemos definir algunas opciones de la instalación aunque no será necesario para que concluya correctamente, pulsamos por tanto "Next".

Seleccionamos ahora el directorio donde se guardará el programa pulsando el botón "Browse" o bien "Next" si no queremos cambiar la selección hecha por defecto.

Por último se requiere que las opciones iniciales de arranque en las que podemos indicar el modo de arranque automático como servicio o bien manual, el puerto de escucha del servidor para la conexión de la utilidad de administración y el modo de arranque de dicha utilidad.

Finalmente comienza la instalación visualizándose en la pantalla los ficheros instalados.

Una vez instalado el software se arranca automáticamente el servidor y se ejecuta la utilidad de administración (solicitando previamente la contraseña de administración que será nula por defecto).

La configuración del servidor se realiza mediante la utilidad "FileZilla Server Interface" del grupo de programas de FileZilla Server. Para poder usarla deberemos tener previamente el servidor en ejecución. Si en la instalación hemos indicado que se arranque como servicio al inicio no tendremos que hacer nada. En otro caso deberemos utilizar la opción de arranque existente en el grupo de programas de FileZilla Server, en la que tenemos dos menús para arrancar y para parar el servidor que son "Start FileZilla Server" y "Stop FileZilla Server".

Al utilizar la utilidad de configuración "FileZilla Server Interface" nos aparecerá una pantalla inicial que nos solicita la dirección IP, puerto y contraseña del administrador del servidor que queremos administrar. Si estamos en la máquina donde se está ejecutando el servidor, simplemente debemos indicar la dirección 127.0.0.1 o nuestra dirección IP, el puerto por defecto es el 14147 y la contraseña está en blanco si es la primera vez que lo configuramos (se recomienda modificar la contraseña de administrador una vez dentro).

Una vez conectados con el servidor la ventana de administración nos permite ver el log de las acciones que va realizando el servidor FTP y los usuarios conectados a la derecha.

En FileZilla Server se pueden configurar tres elementos:

- Los usuarios de conexión.
- Los grupos de usuarios.
- Las opciones generales del servidor

Configuración de opciones generales del servidor

La configuración de las opciones generales de FileZilla Server se realiza desde el menú Edit -> Settings (o Edición -> Configuración) a partir del cual aparece un diálogo con un árbol de opciones de configuración a la izquierda y que están agrupadas en varios bloques:

Configuraciones Generales (General Settings): En esta rama se puede configurar el puerto del servidor FTP, el número máximo de conexiones admitidas, número de hilos (threads) que se ejecutan y la configuración de tiempo de caducidad de las conexiones (timeouts) del servidor (dejar los indicados por defecto). También permite la personalización del mensaje de bienvenida, todo esto se ilustra en la figura 3.16, donde se muestra cada una de las opciones.



FIG 3.16 Menú de opciones de configuración del servidor FTP para la transferencia de archivos

- *Passive mode settings*: Configuración del modo pasivo del servidor.
- *Security settings*: Opciones de seguridad para evitar ataques al mismo.
- *Miscellaneous*: Opciones generales como ocultar las contraseñas en la pantalla de log.
- *Admin Interface settings*: Permite cambiar el puerto de escucha para la utilidad de administración, así como la contraseña del administrador.
- *Logging*: Permite almacenar el log en disco.
- *GSS Settings*: Activación de la autenticación con Kerberos.
- *Speed Limits*: Especificación de reglas para limitar la velocidad de las conexiones, evitando que unas sesiones monopolicen el ancho de banda disponible en detrimento de otras sesiones.

3.3 TERMINALES (CLIENTES)

3.3.1 Especificaciones (Requerimientos Mínimos)

La utilidad de tener clientes dentro de la red respecto al sistema es poder consultar vía intranet el sistema de control de asistencia así como compartir recursos de impresión (de credenciales)

Requerimientos para un óptimo desempeño

- Procesador de 1 GHz 32-bit
- 1 GB de RAM.
- 40 GB de capacidad en disco duro con al menos 15 GB de espacio libre.
- Tarjeta de Red Ethernet 10/100.
- Puertos USB.

3.3.2. Instalación De Software Auxiliares

3.3.2.1 Clientes FTP

Para poder acceder vía ftp al servidor y poder modificar los scripts del sistema necesitaremos tener instalado el navegador Internet Explorer y utilizaremos el gestor FTP del mismo.

En la barra del navegador (tal como se muestra en la figura 3.17) digitaremos ftp://192.168.1.150/, que corresponde a la IP del servidor. Se abrirá una ventana emergente en la cual ingresaremos el nombre de usuario y contraseña proporcionados por el administrador.

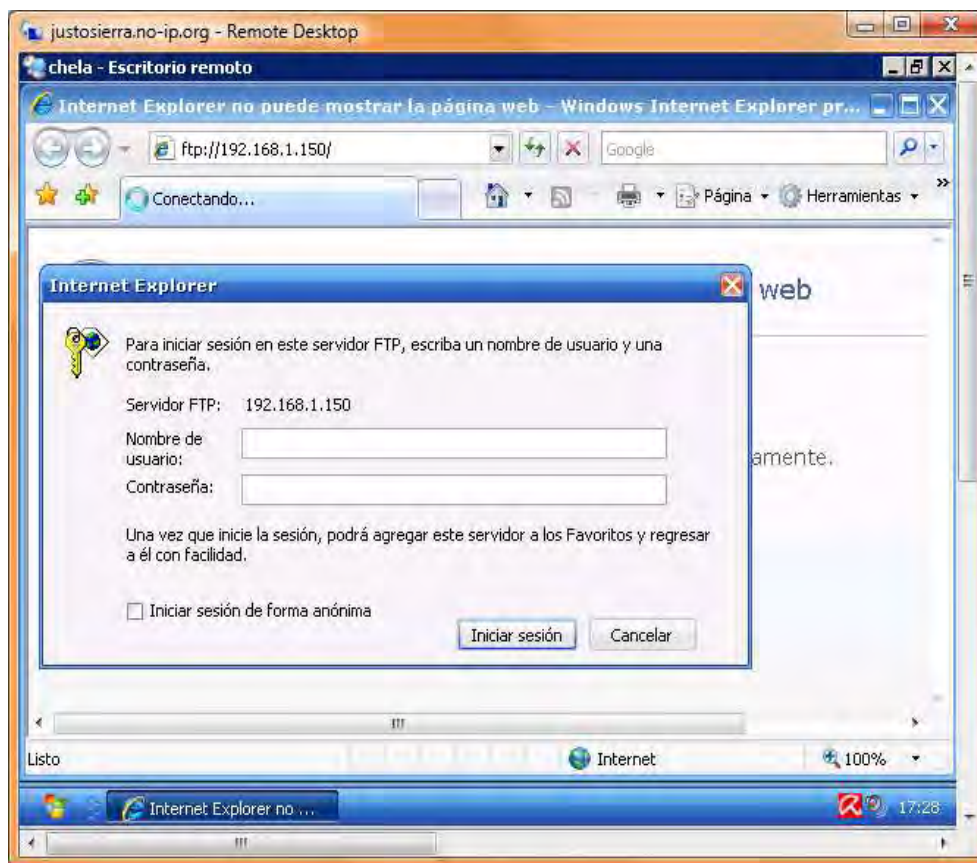


FIG 3.17 Administración de archivos vía conexión FTP.

Una vez autenticados abriremos el menú página y daremos clic en “Abrir el sitio FTP en el explorador de Windows”. Ver figura 3.18.

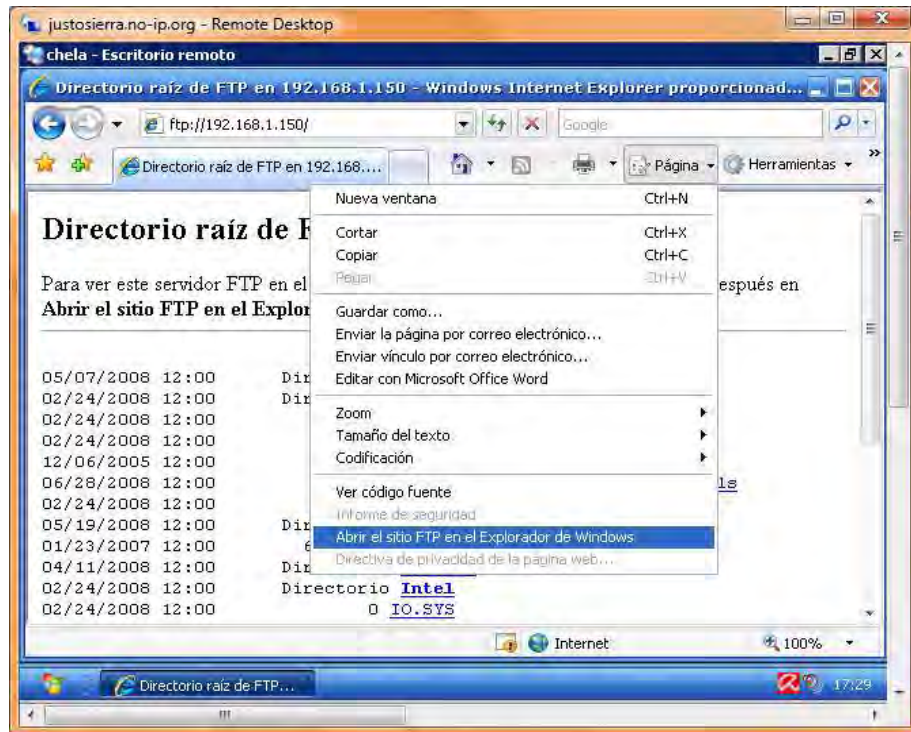


FIG. 3.18 Administración y manipulación de archivos vía FTP desde el explorador de Windows.

Ingresaremos vía FTP y tendremos a la vista dos carpetas ilustradas en la figura 3.19:

1. "admin". En esta carpeta se encuentra los scripts que refieren al sistema a nivel administrador.
2. "portal". Esta carpeta contiene los scripts del portal web para padres de familia.

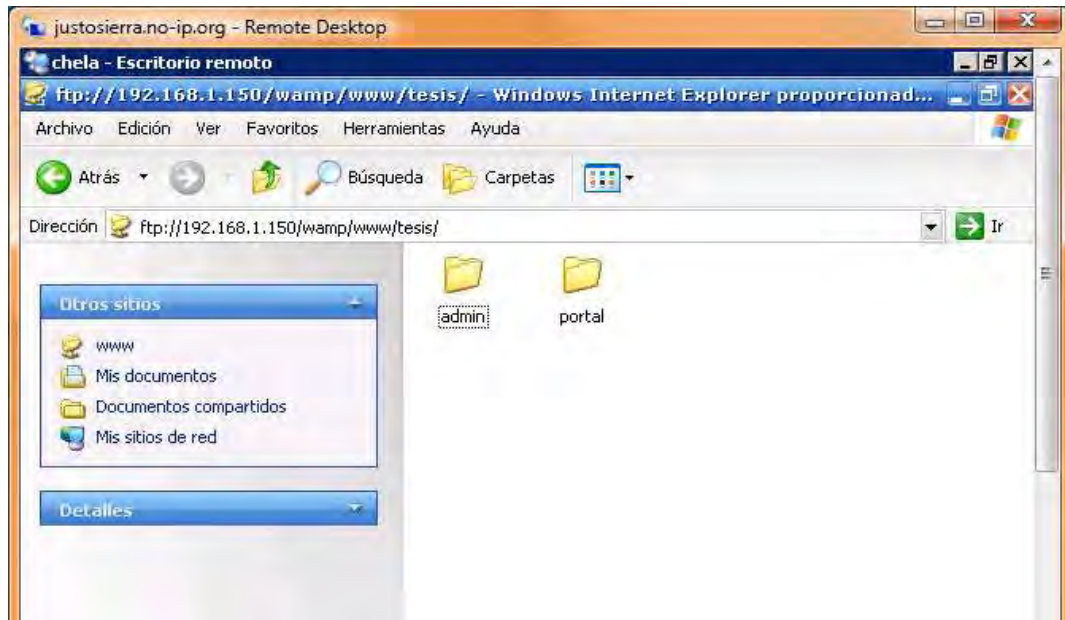


FIG 3.19 Administración de archivos vía FTP desde el explorador de Windows.

3.3.2.2 Software De Impresión De Credenciales

Para iniciar con la instalación del software de diseño e impresión de credenciales “Preface Datacard” inserte el CD y siga las instrucciones a continuación descritas:

1. Dar clic en “Instalar productos”, el cual es el primero del menú de opciones mostrado en la figura 3.20.



Fig. 3.20 Primer pantalla de instalación de software “Data Card” para la impresión de credenciales.

4. Seleccionar el idioma, para nuestro proyecto se elijió Español tal como lo muestra la figura 3.21.



Fig. 3.21 Selección del idioma en que se instalará el programa.

5. Siga los pasos del asistente así como se muestra a partir de la figura 3.22.



Fig. 3.22 Inicio del asistente de instalación.

Una vez terminada la instalación se mostrará una pantalla como la que se ilustra a continuación, esto nos indicará que el software fue instalado exitosamente y esta listo para su uso, la pantalla que se mostrará será



como la que figura 3.23

Fig. 3.23 Pantalla final. Indica una instalación exitosa y muestra las herramientas

3.3.3 Acceso Al Sistema Vía Red Local

Para poder acceder al sistema dentro de la red escolar (en este caso no es necesario una conexión a internet ya que estamos en la red local) tendremos que hacerlo a través de un Navegador Web, en este caso lo haremos con Internet Explorer.

En la barra del navegador pondremos la siguiente URL <http://192.168.10.150/tesis/admin> donde 192.168.10.150 denota la configuración IP del servidor. Con esta URL no se podrá acceder fuera de la red local, ya que la IP es exclusiva de la misma y no identifica al servidor en internet. Para poder ver la ejemplificación citada ver figura 3.24.



Fig. 3.24 Pantalla de acceso local al sistema.

3.3.4 Acceso Al Sistema Vía Internet

También se podrá acceder vía web (ver figura 3.24) desde cualquier lugar en el que exista una conexión a internet a través de la URL <http://justosierra.no-ip.org/tesis/admin/> ya que el dominio justosierra.no-ip.org identifica al servidor de asistencias online dentro de Internet



Fig. 3.24 Pantalla de acceso al sistema vía web.

3.4. RED

3.4.1 Componentes

3.4.1.1 RJ45

RJ-45 es una interfaz física comúnmente usada para conectar redes de cableado estructurado. *RJ* es un acrónimo inglés de *Registered Jack* que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho "pines" o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado

Para el caso de nuestros aplicativos utilizaremos la configuración 568 B, descrita a continuación.

- PIN 1 → Blanco – Naranja
- PIN 2 → Naranja
- PIN 3 → Blanco - Verde
- PIN 4 → Azul
- PIN 5 → Blanco – Azul

- PIN 6 → Verde
- PIN 7 → Blanco – Café
- PIN 8 → Café

3.4.1.2 Cable UTP

El cable de par trenzado es una forma de conexión en la que dos conductores son entrelazados para cancelar las interferencias electromagnéticas (IEM) de fuentes externas y la diafonía de los cables adyacentes.

El entrelazado de los cables disminuye la interferencia debido a que el área de bucle entre los cables, la cual determina el acoplamiento magnético en la señal, es reducida. En la operación de balanceado de pares, los dos cables suelen llevar señales iguales y opuestas (modo diferencial), las cuales son combinadas mediante sustracción en el destino. El ruido de los dos cables se cancela mutuamente en esta sustracción debido a que ambos cables están expuestos a IEM similares.

Este tipo de cable, está formado por el conductor interno el cual está aislado por una capa de polietileno coloreado. Debajo de este aislante existe otra capa de aislante de polietileno la cual evita la corrosión del cable debido a que tiene una sustancia antioxidante.

Normalmente este cable se utiliza por pares o grupos de pares, no por unidades, conocido como cable multipar. Para mejorar la resistencia del grupo se trenzan los cables del multipar.

Los colores del aislante están estandarizados, y son los siguientes: Naranja/ Blanco-Naranja, Verde/ Blanco-Verde, Azul/ Blanco-Azul, Café/Blanco-Café.

Cuando ya están fabricados los cables unitariamente y aislados, se trenzan según el color que tenga cada uno. Los pares que se van formando se unen y forman subgrupos, estos se unen en grupos, los grupos dan lugar a superunidades, y la unión de superunidades forma el cable.

3.4.1.3 Tarjetas de red

El dispositivo más utilizado en estos momentos para conectar un dispositivo a red son las tarjetas de red o más conocido como NIC (Network Interface Card), este dispositivo es del tamaño de una tarjeta estándar que puede venir de forma integrada en la tarjeta madre (motherboard) o individualmente, se coloca en ranuras de ampliación de las PC o en las computadoras portátiles mediante puertos USB, PCMCIA, etc.

En la actualidad existen una gran cantidad de variedad de tarjetas de red desde las que se colocan dentro de las computadoras o de manera externa, tanto las de conexión física o inalámbricas, desde las que se utilizan en las PC normales o en otros dispositivos como Hubs, Routers y Switches, e incluso impresoras, escáner y demás, todos estos dispositivos necesitan de la tarjeta de red para conectarse con otros dispositivos o para compartir recursos.

3.4.1.4 Switch

Switch (conmutador) es un dispositivo analógico de lógica de interconexión de redes de computadoras que opera en la capa 2 del modelo OSI (*Open Systems Interconnection*). Un conmutador interconecta dos o más segmentos de red, funcionando de manera similar a los puentes (bridges), pasando datos de un segmento a otro, de acuerdo con la dirección MAC de destino de los datagramas en la red.

3.4.2 Configuración De Equipos

En este caso la configuración de los equipos para la red se realizara por medio de IPs estáticas esto para evitar colisiones con los sensores, ya que estos últimos manejan IP estática.

3.4.2.1 SERVIDOR

El servidor está definido por la ip 192.168.1.150

La máscara de subred está definida por 255.255.255

Con una puerta de enlace autorizada para tener acceso a internet y poder ser consultado vía internet. Estas configuraciones se pueden realizar en las pantallas mostradas en la figura 3.25.

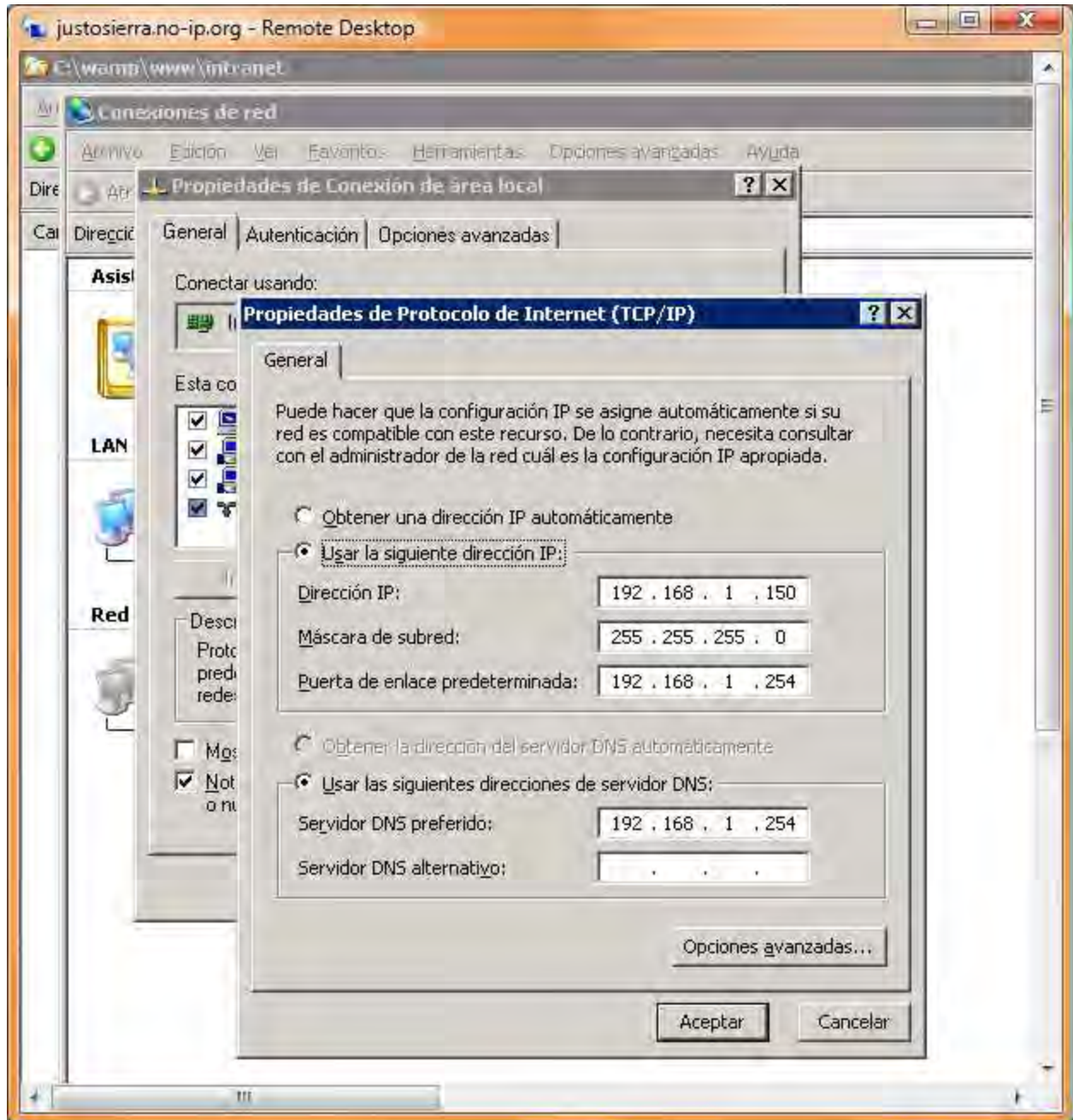


Fig. 3.25 Configuración de red TCP/IP.

3.4.2.2 CLIENTES

Para los equipos de cómputo (Clientes) utilizaremos las ips del tipo 192.168.1.x, donde x puede ir

desde 1 hasta 159. La máscara de subred está definida por 255.255.255. Estas configuraciones se pueden realizar en las pantallas mostradas en la figura 3.25.

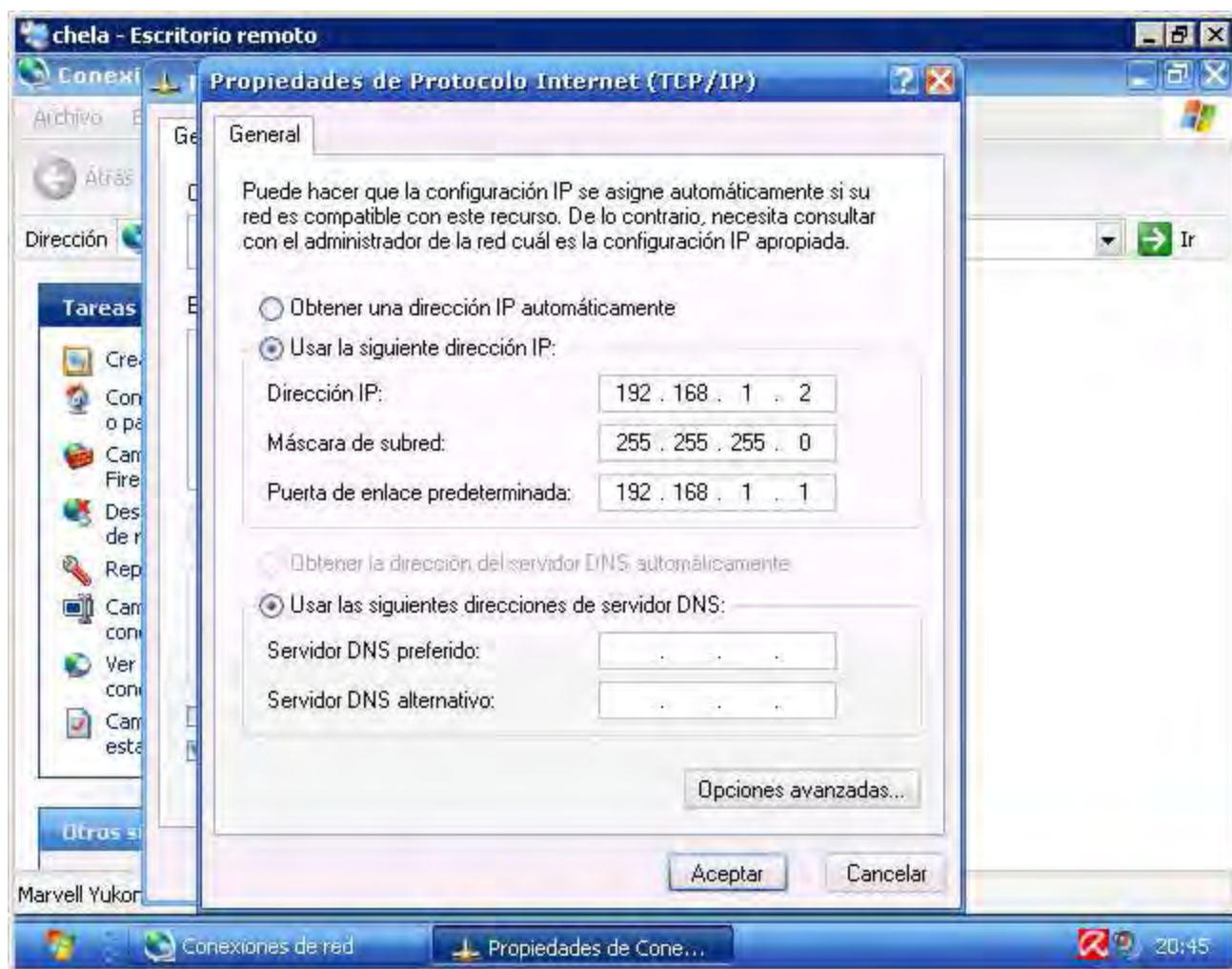


Fig. 3.26 Asignación de IP estática.

3.4.2.3 SENSORES

Para los sensores utilizaremos las IP del tipo 192.168.1.x, donde x puede ir desde 160 y en este caso hasta 163. La máscara de subred está definida por 255.255.255 y la configuración gráficamente se observaría como la figura 3.27.

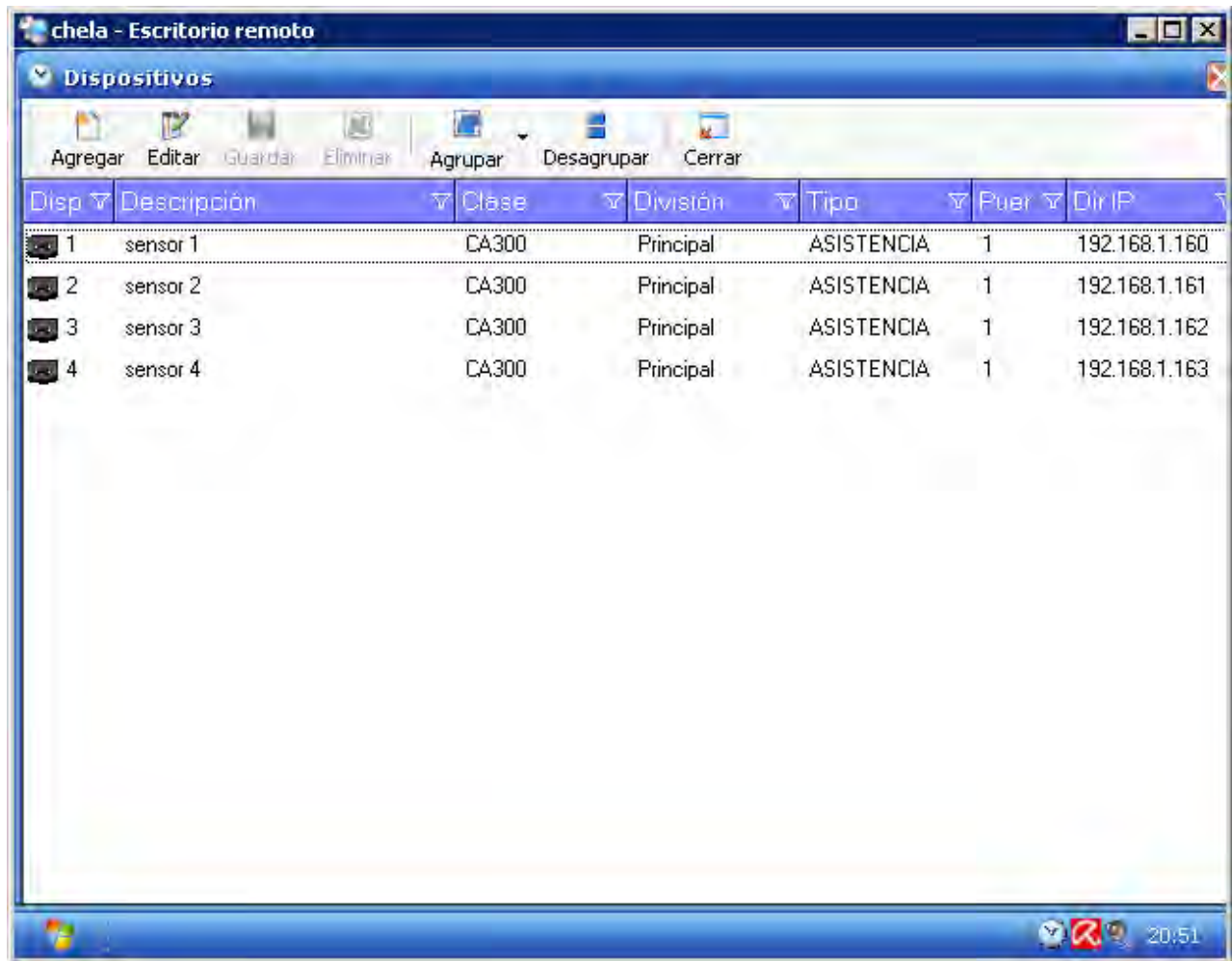


Fig. 3.27 Sensores instalados e IP asignada a cada uno.

3.5 Sensores

Existen diferentes tipos de sensores en este caso utilizaremos la terminal ca300 de proximidad. La terminal CA300 se puede utilizar ya sea instalada en una pared o sobre una mesa o escritorio, pues la orientación de la pantalla puede girarse a 180° y así tener un excelente ángulo de visualización en cualquier modo de instalación.

3.5.1 Especificaciones

Especificaciones técnicas de los sensores:

- Pantalla tipo LED en color azul
- Lector de banda magnética : ISO Track 2 , 75 BPI
- Lector de código de barras : Infrarojo, de tipo Código 39
- Capacidad de Almacenaje de la base de datos : Aprox. 8200 registros
- Interfase : Ethernet (TCP/IP) ó Serial (Rs232)
- Relevador de campanas : 1A , 12 V max.
- Fuente de poder : DC 9 V , 500 mA
- Dimensiones : 185mm Ancho x 110mm Alto x 51mm Profundidad
- Instalación : montaje en la pared o en el escritorio

- Temperatura de operación: 0 ~ 55 C
- Humedad : 5 ~ 95 % relativa CA300

Características:

- Opción de comunicaciones a la PC : Interfase RS232 ó TCP/IP (Ethernet)
- Pantalla con iluminación LED brillante en color azul, de alto contraste, que muestra la hora y los últimos 4 dígitos del código del empleado o bien mensajes de error.
- Capacidad de almacenaje de aproximadamente 8,200 registros
- Fácil y práctica configuración de número de Terminal permite distinguir entre múltiples terminales instaladas con el cliente
- Botones para entrada (ENT) y salida (SAL)
- Relevador integrado para control de campanas que permitan señalar el inicio/fin de turnos de trabajo
- Permite programar hasta 32 señales para campanas
 - Opciones de lectura de tarjetas: Banda Magnética, Código de Barras, Proximidad Mifare

3.5.2 Pruebas de funcionamiento

Una vez implementada la configuración de las IP para los sensores procedemos a verificar si hay comunicación con ellos mediante el comando PING.

Realizaremos Ping a la dirección de los 4 sensores en caso de fallo la pantalla de comandos se mostrara así, si esto ocurre tendremos que verificar la continuidad de los cables, verificar la configuración de los sensores y reiniciar el proceso de pruebas. Un ejemplo de un intento de comunicación fallido con los sensores es como el que se muestra en la figura 3.28.

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrador>ping 192.168.1.161

Haciendo ping a 192.168.1.161 con 32 bytes de datos:

Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.161:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos).

C:\Documents and Settings\Administrador>_
  
```

Fig. 3.28 Ping a los sensores. Ejemplo para cuando la comunicación con el sensor no ha sido exitosa.

Si la comunicación es exitosa la pantalla será como se muestra en la figura 3.30:

```

C:\ Símbolo del sistema
Microsoft Windows [Versión 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrador>ping 192.168.1.161

Haciendo ping a 192.168.1.161 con 32 bytes de datos:

Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.161:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),

C:\Documents and Settings\Administrador>_

```

Fig. 3.29 Ping a los sensores. Ejemplo para cuando la comunicación con el sensor ha sido exitosa.

Una vez que tenemos comunicación con los sensores estos están listos para operar.

3.6. SMS

3.6.1 Instalación de interfaz de conectividad

Instalación de Java

La versión utilizada de java en el presente proyecto es la J2EE 1.4.2 (<http://java.sun.com/>), aplicación que contiene librerías y lenguaje, no incluye un ambiente de desarrollo

En el servidor utilizado (que para el presente proyecto puede ser con el sistema operativo Windows Server 2003 o Windows XP) se deben agregar las siguientes variables de entorno para la instalación:

Dentro del PATH

; C:/Archivos de programa/Java/j2sdk1.4.2_06/bin

Dentro del CLASSPATH

;

Librerías adicionales para la comunicación con los puertos

Se deben agregar las siguientes librerías, las cuales sirven para establecer la comunicación entre la computadora y el celular, es decir, permiten establecer conexión por medio del puerto USB para reconocer el celular como un modem.

Nombre	Ubicación
win32com.dll	C:/Java/jdk1.1.6/bin
comm.jar	C:/Java/jdk1.1.6/lib
javax.comm.properties	C:/Java/jdk.16/lib

Agregar adicionalmente las siguientes variables de entorno, dentro del CLASSPATH:

`C:/jdk1.1.6/lib/comm.jar;%classpath%`

3.6.2 Componentes

3.6.2.1 Especificaciones de equipos (requerimientos mínimos)

Para poder dar lugar al envío de los mensajes de texto, es necesario contar con el siguiente hardware:

COMPUTADORA

- Procesador de 1 GHz 32-bit
- 1 GB de RAM.
- 40 GB de capacidad en disco duro con al menos 15 GB de espacio libre.
- Tarjeta de Red Ethernet 10/100.
- Puertos USB 2.0.

CELULAR

- Conectividad hacia la computadora vía USB 2.0.
- Servicio de mensajería de texto (SMS).
- Aceptar conectividad hacia la computadora como módem.
- Capacidad de recargar la pila a través del puerto USB.
- Contar con crédito para los envíos de mensajes.
- Compatibilidad con JAVA

3.6.2.2 Instalación de controladores de dispositivos

Los controladores de los puertos USB empleados para la comunicación entre la computadora y el celular son instalados por default durante la instalación del sistema operativo en el servidor.

Si el celular cumple con la compatibilidad de conexión USB como módem, puede ser el caso que al momento de adquirirlo el controlador para dicha conectividad venga incluido, de no ser así, dependiendo de la marca del celular, será necesario descargar de la página web del proveedor y/o fabricante los controladores e instalarlos en el servidor.

3.6.2.3 Pruebas de conectividad

Los archivos desarrollados en PHP que llaman a las funciones de java corren estas aplicaciones y la salida que se desplegaría en la consola de comandos es capturada vía base de datos, esto para el caso de la recepción de los mensajes.

Para correr los programas ya compilados desde la consola se utiliza el comando java:

```
java enviarsms 5534253459 "Mensaje de prueba" "COM7"
```

Es importante hacer notar que los puertos deben ir siempre con mayúsculas, de lo contrario no funciona. Lo que hace en si para identificar que el puerto exista es comparar el parámetro que se le está pasando con la primera coincidencia que encuentre:

```
idPuerto = (CommPortIdentifier)listaPuertos.nextElement();
```


Imprimir todos los puertos encontrados:

```
System.out.println(idPuerto.getName());
```

Si no encuentra coincidencia indica el error del puerto.

Comandos AT:

Envío de mensajes

Iniciar operación de envío, apertura de canal SMS

```
AT+CMGF=1
```

Modo en el que se despliegan los códigos y los errores

```
AT+CMEE=2
```

Indicar el número donde se enviará el mensaje

```
AT+CMGS=(Aquí se concatena el número telefónico usando "+")
```

Al final se envía el mensaje a través del buffer. Ctrlz indica final de comando
mensajeCel+(char)Ctrlz

Lector de mensajes de SMS

Iniciar operación de envío, apertura de canal SMS

```
AT+CMGF=1
```

Modo en el que se despliegan los códigos y los errores

```
AT+CMEE=2
```

Lectura de posibles datos en la memoria SIM del celular

```
AT+CNMI=0,1,0,0,0
```

Respuesta de teclas presionadas en el celular

```
AT+CMER=3,2,0,1,0
```

Obtener el número de serie del modelo

```
AT+CGSN
```

Leer todos los mensajes del celular

```
AT+CMGL="ALL"
```

Borrado de mensaje de SMS

El programa que borra los mensajes de celular es muy similar al que da lectura de los mismos, con la excepción que recibe un índice del mensaje a borrar en cuestión, esto es así porque se corre el riesgo de que no se leyeran todos los mensajes antes de proceder a borrar la memoria.

Lectura del mensaje

```
AT+CMGL=+indice
```


Eliminar mensaje indicado en el índice

```
AT+CMGD=+indice
```

Una vez que finalizan los comandos, se cierran las comunicaciones con los puertos correspondientes.

3.6.3 Pruebas de envíos manuales

Para hacer envíos manuales y corroborar que el servidor de SMS este activo, tenemos que ejecutar la siguiente instrucción: "java enviarsms 5514521986 "Hola" COM17", tal como se ilustra en la figura 3.30:



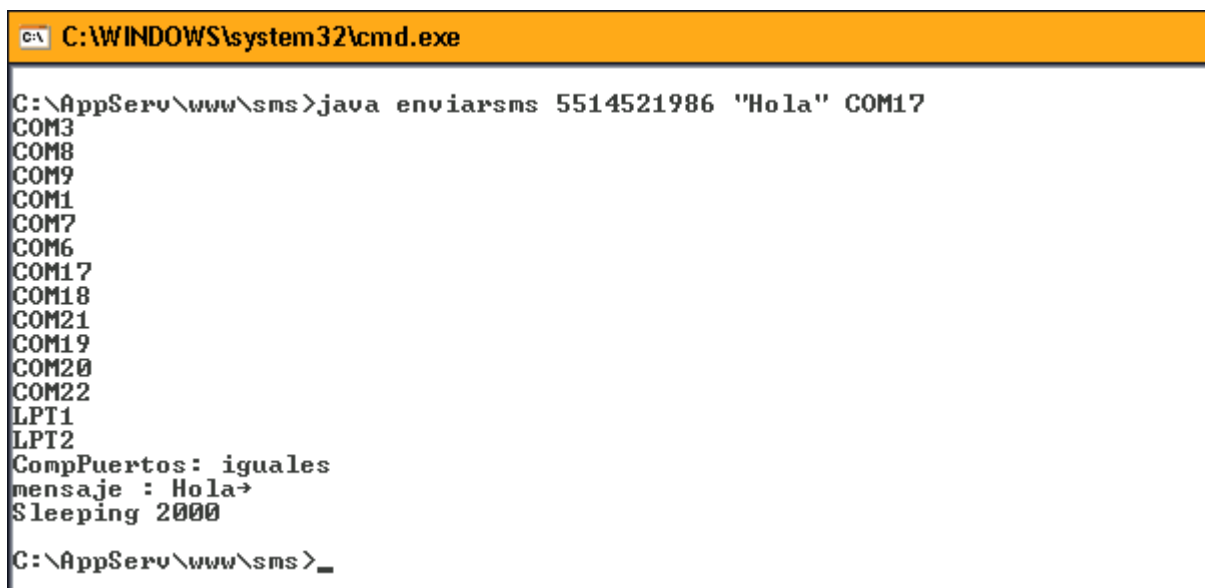
```
C:\WINDOWS\system32\cmd.exe
C:\AppServ\www\sms>java enviarsms 5514521986 "Hola" COM17_
```

FIG. 3.30 Prueba de envío manual de SMS

En donde:

enviarsms	Es el nombre del programa
5514521986	Número del celular al cual se le envía el mensaje
"Hola"	Cuerpo del mensaje
COM17	Puerto de comunicación con el dispositivo móvil

Si la instrucción es correcta el programa listará los puertos disponibles e imprimirá las en pantalla lo que en la figura 3.31 se muestra:



```
C:\WINDOWS\system32\cmd.exe
C:\AppServ\www\sms>java enviarsms 5514521986 "Hola" COM17
COM3
COM8
COM9
COM1
COM7
COM6
COM17
COM18
COM21
COM19
COM20
COM22
LPT1
LPT2
CompPuertos: iguales
mensaje : Hola→
Sleeping 2000
C:\AppServ\www\sms>
```

FIG. 3.31 Resultado de un envío manual de SMS exitoso.

Si la instrucción es escrita incorrectamente, la pantalla mostrada será la que a continuación la figura 3.32 ilustra:

```
C:\WINDOWS\system32\cmd.exe
COM22
LPT1
LPT2
CompPuertos: iguales
mensaje : Hola
Sleeping 2000
C:\AppServ\www\sms>java enviarsms 5514521986 "Hola" COM199
COM3
COM8
COM9
COM1
COM7
COM6
COM17
COM18
COM21
COM19
COM20
COM22
LPT1
LPT2
CompPuertos: distintos
C:\AppServ\www\sms>
```

FIG. 3.32 Resultado de un envío manual de SMS fallido.

3.7 Impresora de Credenciales

La impresora SP55 ofrece una impresión de 180 tarjetas por hora en color o hasta 700 tarjetas por hora con cintas monocromáticas. La Datacard SP55 cuenta con opciones como tarjetas inteligentes, bandas magnéticas o códigos de barra, que facilitan crear un gran rango de aplicaciones para tarjetas. Es ideal para identificaciones con foto de multi-tecnología que nos permite accesos seguros para facilitar accesos, así como otros privilegios.

3.7.1 Especificaciones y características

Características:

- Una o dos caras, de borde a borde de la foto de impresión
- Full color o monocromo de imágenes
- Conectividad USB
- Datacard © Certified Suministros Suministros con Intelligent Technology™ con funciones avanzadas
- Audio y visuales indicaciones mensaje
- Continúa la limpieza de rodillos
- Operador de impresión reemplazable
- Rápido cambio ERRORES cartucho de cinta de carga
- 24 meses de depósito de garantía estándar para impresora
- 24 meses de garantía de impresión - sin prorrateo, sin contar la tarjeta

Especificaciones

- Full impresión en color: Hasta 190 tarjetas por hora (YMCKT cinta) - la velocidad de impresión, hasta

- 155 tarjetas por hora (YMCKT - K)
- Impresión Monocromática: Hasta un máximo de 1000 tarjetas por hora (K cinta) - la velocidad de impresión
- Dimensiones físicas:
 - 16,5 pulgadas (Largo) X 7,78 pulgadas (Ancho) X 9 pulgadas (Alto)
 - Peso: menos de 11 libras (4 kilogramos), en función de las opciones de configuración de la impresora
- Controladores de impresora Datacard compatibles con Microsoft Windows 2000 y XP.

3.8 Credenciales

3.8.1 Especificaciones y características

La tarjeta ISOProx II ofrece tecnología de proximidad con capacidad de identificación fotográfica en una sola tarjeta. Cumple la normativa ISO 7810 con un grosor nominal de 0,030". La tarjeta cuenta con una superficie de calidad gráfica optimizada para la impresión de identificaciones fotográficas y está disponible en PVC o en compuesto de poliéster y PVC para entornos más exigentes y de gran desgaste.

Al igual que todas las tarjetas ISO de HID, la tarjeta ISOProx se puede producir con funciones de seguridad visual y antifalsificación como hologramas, tintas fluorescentes ultravioletas, microimpresión o un logo de cliente para identificar rápida y fácilmente las tarjetas auténticas.

Características principales

- Ofrece tecnología de proximidad con capacidad de identificación fotográfica en una única tarjeta de proximidad
- Superficie de calidad gráfica para su uso con impresoras de imagen directa
- Tiene el mismo tamaño y grosor que una tarjeta de crédito estándar
 - Capacidad de perforado de agujeros vertical u horizontal

Construcción de la tarjeta Fino y flexible laminado de cloruro de polivinilo (PVC) o compuesto de poliéster/PVC.

CAPÍTULO 4

ARQUITECTURA DEL

SISTEMA

CAPÍTULO 4 ARQUITECTURA DEL SISTEMA

4.1. PROCESO GENERAL

4.1.1 Descripción

La decisión de qué software y qué hardware se utilizará es fundamental, han sido seleccionados de acuerdo a las expectativas de crecimiento y a los servicios que requerirán los usuarios, apegándose a los estándares internacionales y a las tendencias en los sistemas y servicios Web de información.

El sistema se sirve de un servidor de bases de datos, SQL SERVER, ubicado en un servidor de bases de datos. Así mismo, todos los archivos generados durante el desarrollo y programación, están almacenados en un servidor Web y estos interactúan directamente con la Base de Datos, en consecuencia, toda la información del sistema, como datos adquiridos en tiempo real, el histórico de información almacenada, así como información de accesos, administración de cuentas, etc. es accesible desde cualquier computadora que cuente con conexión a internet.

4.2. BASES DE DATOS

4.2.1 Diccionario De Datos

El diccionario de datos del presente proyecto es un conjunto de metadatos (datos que describen otros datos) que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización. Este diccionario se desarrolló durante el análisis del problema y ayudó en la determinación y definición de los requerimientos del sistema.

4.2.1.1 PREFIJOS

Para mantener consistencia en la asignación de los nombres y tipos de datos dentro de la Base de Datos generada, se han asignado una serie de prefijos que ayudan a identificar el tipo de dato que se está trabajando, esto es de gran utilidad cuando se desean realizar operaciones dentro de la programación con las variables que contienen valores obtenidos de la base de datos. Conociendo los prefijos establecidos, se puede identificar si la variable o el campo corresponde a un valor tipo numérico o cadena de caracteres.

Los prefijos asignados para el diseño de la Base de Datos utilizada en el presente proyecto son los siguientes:

Tablas

La sintaxis adoptada para la base de datos consta de prefijos que ayudarán a encontrar información de una manera visual sencilla e intuitiva.

- **tbl_** Este prefijo se utiliza cuando la tabla correspondiente contiene información independiente y eventualmente datos de catálogos a través de llaves foráneas.
- **cat_** Este prefijo denota la existencia de una tabla tipo catalogo.
- **App_** Tablas de la aplicación, es decir con información correspondiente a la administración del sistema

Campos

La sintaxis adoptada para los campos denotará el tipo de información y tipo de datos que se almacena en ellos a través de un prefijo.

- id_ Identificador
- st_ String - Varchar
- dt_ Fecha
- i_ Boleano

4.2.1.2 DICCIONARIO DE DATOS

1. tbl_UsuariosWeb. En esta entidad se guarda la información correspondiente a los alumnos (datos socio-demográficos.)

Nombre del campo	Propiedades	Descripción
id_UsuarioWeb	bigint IDENTITY(1, 1)	Identificador único del alumno (PK)
st_Email	varchar(255) NULL,	Correo Electronico
st_Tarjeta	varchar(255) NOT NULL,	Numero de sensor en la credencial
st_Matricula	varchar(255) NOT NULL,	Numero de cuenta
st_Nombre	varchar(255) NULL,	Nombre del alumno
st_ApellidoPaterno	varchar(255) NULL,	Apellido Paterno
st_ApellidoMaterno	varchar(255) NULL,	Apellido Materno
dt_FechaNacimiento	datetime NULL,	Fecha de Nacimiento
id_Genero	int NULL,	Identificador correspondiente al genero(FK)
st_Direccion	varchar(255) NULL,	Lugar de residencia (Calle)
st_NumeroExterior	varchar(255) NULL,	Lugar de residencia (Numero Exterior)
st_NumeroInterior	varchar(255) NULL,	Lugar de residencia (Numero Interior)
st_ReferenciaDomicilio	varchar(255) NULL,	Referencias al lugar de residencia
st_Colonia	varchar(255) NULL,	Lugar de residencia (Colonia)
st_CodigoPostal	varchar(255) NULL,	Codigo Postal
st_Ciudad	varchar(255) NULL,	Ciudad
dt_FechaRegistro	datetime NULL,	Fecha de alta del registro
id_Municipio	bigint NULL,	Identificador correspondiente al municipio del lugar de residencia (FK)
id_Estado	bigint NULL	Identificador correspondiente al Estado del lugar de residencia (FK)

2. cat_Estados. Catalogo de Entidades Federativas de la República Mexicana

Nombre del campo	Propiedades	Descripción
id_Estado	bigint NOT NULL,	Identificador Único del Estado (PK)
st_Estado	varchar NOT NULL,	Nombre del estado
dt_FechaRegistro	datetime NOT NULL	Fecha de Alta en el sistema

3. cat_Municipios. Catálogo de municipios y delegaciones de la República Mexicana

Nombre del campo	Propiedades	Descripción
id_Municipio	bigint NOT NULL,	Identificador del Municipio (PK)
id_Estado	bigint NOT NULL	Identificador del Estado al cual corresponde dicho municipio (FK)
st_Municipio	varchar NOT NULL,	Nombre del Municipio
dt_FechaRegistro	date NOT NULL,	Fecha de alta en el sistemas

4. cat_Generos. Catalogo descriptivo de géneros

Nombre del campo	Propiedades	Descripción
id_Genero	int NOT NULL,	Identificador único del genero (PK)
st_Genero	varchar NOT NULL,	Nombre descriptivo del genero
dt_FechaRegistro	datetime NOT NULL	Fecha de alta en el sistema

5. tbl_UsuariosWebSituacionEscolar. Entidad intersección en la cual esta almacenada la información correspondiente a la situación escolar actual (ciclo escolar en curso), es decir grado grupo y turno.

Nombre del campo	Propiedades	Descripción
id_Turno	int NOT NULL,	Identificador correspondiente al turno (FK)
id_Grado	int NOT NULL,	Identificador correspondiente al grado (FK)
id_Grupo	int NOT NULL,	Identificador correspondiente al grupo (FK)
id_UsuarioWeb	bigint NOT NULL	Identificador del alumno (FK)
dt_FechaRegistro	datetime NOT NULL,	Fecha de alta en el sistema de la situacion escolar actual del alumno

6. cat_Grupos. Catálogo utilizado por la institución escolar en la cual se le atribuye un nombre a cada grupo.

Nombre del campo	Propiedades	Descripción
id_Grupo	int NOT NULL,	Identificador único del grupo (PK)
st_Grupo	varchar NOT NULL,	Nombre designado al grupo
dt_Fecharegistro	datetime NOT NULL	Fecha de alta en el sistema

7. cat_Turnos. Catálogo en el cual se especifican los turnos existentes dentro de la institución escolar.

Nombre del campo	Propiedades	Descripción
id_Turno	int NOT NULL	Identificador único del grupo (PK)
st_Turno	varchar NOT NULL	Nombre descriptivo del turno
dt_FechaRegistro	datetime NOT NULL	Fecha de alta en el sistema

8. cat_Grados. Catálogo en el cual se especifican los grados existentes dentro de la institución escolar.

Nombre del campo	Propiedades	Descripción
id_Grado	int NOT NULL,	Identificador único del grupo (PK)
st_Grado	varchar(25) NOT NULL,	Nombre descriptivo del grado
dt_FechaRegistro	datetime NULL	Fecha de alta en el sistema

9. tbl_InasistenciasUsuariosWeb. Entidad en la cual se almacena la información de asistencias/inasistencias por alumno.

Nombre del campo	Propiedades	Descripción
id_Asistencia	bigint IDENTITY(1, 1),	Identificador único del Evento (PK)
id_UsuarioWeb	bigint NOT NULL,	Identificador del alumno (FK)
i_asistio	bigint NULL	Valor entero booleano positivo que identifica si el registro corresponde a una asistencia o inasistencia asistencia = 1 inasistencia = 0
id_Clock	int NULL	Identificador del reloj en donde se llevo acabo el evento de asistencia (FK)
dt_FechaRegistro	datetime NULL,	Fecha del evento asistencia/inasistencia

10. cat_Clocks. Catálogo descriptivo de relojes checadores para control de inasistencias.

Nombre del campo	Propiedades	Descripción
id_Clock	int NOT NULL,	Identificador único del reloj (PK)
st_NumbreClock	varchar NOT NULL,	Nombre del reloj
st_Comentarios	varchar NOT NULL	Notas importantes del reloj
dt_FechaRegistro	date NOT NULL,	Fecha de alta en el sistema

11. tbl_PadresTutores. Entidad en la que esta almacenada la información correspondiente a los padres o tutores.

Nombre del campo	Propiedades	Descripción
id_PadreTutor	int IDENTITY(1, 1),	Identificador único del Padre o Tutor (PK)
st_nomPadreTutor	varchar(100) NULL,	Nombre
st_apPaterno	varchar(100) NULL,	Apellido Paterno
st_apMaterno	varchar(100) NULL,	Apellido Materno
st_user	varchar(50) NULL,	Nombre de usuario para acceder al portal
st_pwd	varchar(50) NULL,	Contraseña encriptada en MD5 para acceder al portal web
st_mail	varchar(50) NULL,	Correo electrónico
st_telefono	varchar(50) NULL,	Teléfono Celular al cual será enviado el SMS notificando la inasistencia del hijo
dt_fechaRegistro	datetime NULL	Fecha de alta en el sistema

12. cat_UsuariosWeb_PadreTutores. Entidad intersección en la cual esta almacenada la información correspondiente a la relación padre e hijos , en donde un padre puede tener “n” hijos.

Nombre del campo	Propiedades	Descripción
id_UsuarioWeb	bigint NOT NULL,	Identificador correspondiente al alumno (FK)
id_PadreTutor	int NOT NULL,	Identificador correspondiente al padre/tutor (FK)
dt_FechaRegistro	datetime NULL	Fecha de alta de la relacion padre/hijo

13. tbl_LoginLogs. Tabla en la cual se registran los accesos de los padres al portal web para realizar consultas de las asistencias/inasistencias de sus hijos.

Nombre del campo	Propiedades	Descripción
id_LoginLog	bigint IDENTITY(1, 1),	Identificador único del evento login del padre/tutor (PK)
id_PadreTutor	int NOT NULL,	Identificador correspondiente al padre/tutor (FK)
st_RemoteServer	varchar(50) NULL	IP desde la cual se realiza petición de logueo al portal
dt_FechaLogin	datetime NULL,	Fecha de logueo en el portal Web

14. app_SysUsers. Tabla del sistema en el cual se almacena la información correspondiente a los usuarios y administradores del sistema a nivel Institucional.

Nombre del campo	Propiedades	Descripción
id_SysUser	int IDENTITY(1, 1),	Identificador único del usuario (PK)
username	varchar(100) NULL,	Nombre de usuario para acceso al sistema a nivel consulta/administrador dependiendo el nivel de permisos
password	varchar(255) NULL,	Contraseña. encriptado en MD5 para acceder al sistema a nivel consulta/administrador dependiendo el nivel de permisos
cedulaCURP	varchar(50) NULL,	Clave única de registro de población
nombre	varchar(100) NULL,	Nombre del Usuario
paterno	varchar(50) NULL,	Apellido Paterno
materno	varchar(50) NULL,	Apellido Materno
email	varchar(255) NULL,	Correo electrónico
puesto	varchar(100) NULL,	Puesto dentro de la institución escolar
fechaAlta	datetime NULL	Fecha de alta en el sistema

15. app_SysSectionsUsers. En esta entidad esta almacenada la información correspondiente a los permisos por secciones que tienen los usuarios dentro del sistema escolar.

Nombre del campo	Propiedades	Descripción
id_SysUser	int NOT NULL,	Identificador correspondiente al usuario (FK)
id_SysSection	int NOT NULL,	Identificador correspondiente a la seccion a la cual tendra acceso el usuario (FK)
fechaAsignacion	datetime NULL	Fecha de asignacion del permiso a nivel sistema

16. cat_Sections. Catálogo de módulos con el que el sistema web de control de asistencias cuenta.

Nombre del campo	Propiedades	Descripción
id_SysSection	int NOT NULL,	Identificador único de la seccion (PK)
st_SysSection	varchar NOT NULL,	Nombre descriptivo de la sección
dt_FechaRegistro	datetime NOT NULL	Fecha de alta en el sistema

17. tbl_EventosUsuariosWeb. Tabla en la que esta almacenada el LOG de todas las acciones realizadas por el alumno dentro del sistema escolar (asistencias, accesos al portal web, altas alumnos, etc.)

Nombre del campo	Propiedades	Descripción
id_Evento	int NOT NULL,	Identificador único del evento (PK)
id_TipoEvento	int NOT NULL,	Identificador correspondiente al tipo de evento este nos informara si el evento es de tipo alta alumno , inasistencia etc. (FK)
id_UsuarioWeb	bigint NOT NULL,	Identificador correspondiente al alumno (FK)
st_LogEvento	varchar NULL,	Cadena para describir detalles importantes del evento
id_SysUser	int NULL	Identificador correspondiente al usuario de sistema que genero el evento en caso de existir. (FK)
dt_FechaRegistro	datetime NULL,	Fecha de registro en el sistema

18. cat_TipoEventos. Catálogo en el que se describen los eventos existentes dentro de la institución escolar Ejemplo (Alta de alumnos, inasistencias, asistencias, etc.)


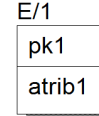

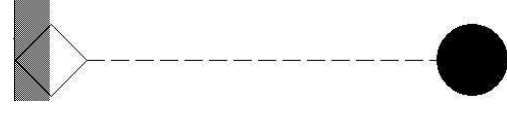
Nombre del campo	Propiedades	Descripción
id_TipoEvento	int NOT NULL,	Identificador único del evento (PK)
st_TipoEvento	varchar NOT NULL,	Nombre descriptivo del evento
dt_FechaRegistro	datetime NOT NULL	Fecha de alta en el sistema

4.2.2 DIAGRAMA ENTIDAD RELACIÓN

4.2.2.1 NOMENCLATURA

Para el diseño del diagrama entidad relación se generó en primer instancia manualmente un boceto a mano. Una vez que se concluyó con el diseño de la base de datos, las tablas y las relaciones resultantes se diagramaron utilizando el software Erwin Data Modeler, en su versión 4.1. Esta herramienta es muy utilizada en organizaciones de todos los sectores (público y privado) para la documentación digital de los diagramas entidad-relación de las bases de datos que se generen.

La nomenclatura que Erwin Data Modeler utiliza para la diagramación es la siguiente

	Entidad de nombre "E/1". Es la representación gráfica de una tabla.
	"E/1" es el nombre de la tabla. "pk1" es el nombre de la llave primaria de la tabla "E/1" "atrib1" es el nombre de un atributo de la tabla "E/1" y que depende de la llave primaria "pk1".
	Es la representación de IDENTIFICACIÓN de una relación entre tablas. La línea continua significa que dentro de la relación la tabla que está conectada en el extremo del "círculo" dependerá y tendrá como parte de su propia llave primaria la llave primaria de la tabla conectada en el otro extremo, lo que implica que dicho campo nunca será nulo (NOT NULL).
	Es la representación de NO IDENTIFICACIÓN de la relación entre tablas. La línea punteada significa que la llave primaria de la tabla en el extremo del rombo será transmitida a la tabla ubicada en el extremo del círculo pero no aparecerá como llave primaria, simplemente como un atributo más que inclusive podría contener valores nulos (NULL).

4.2.2.2 DIAGRAMA ENTIDAD RELACIÓN

Con base a las primeras tres formas de normalización se diseñó el diagrama que en la figura 4.1 se presenta:

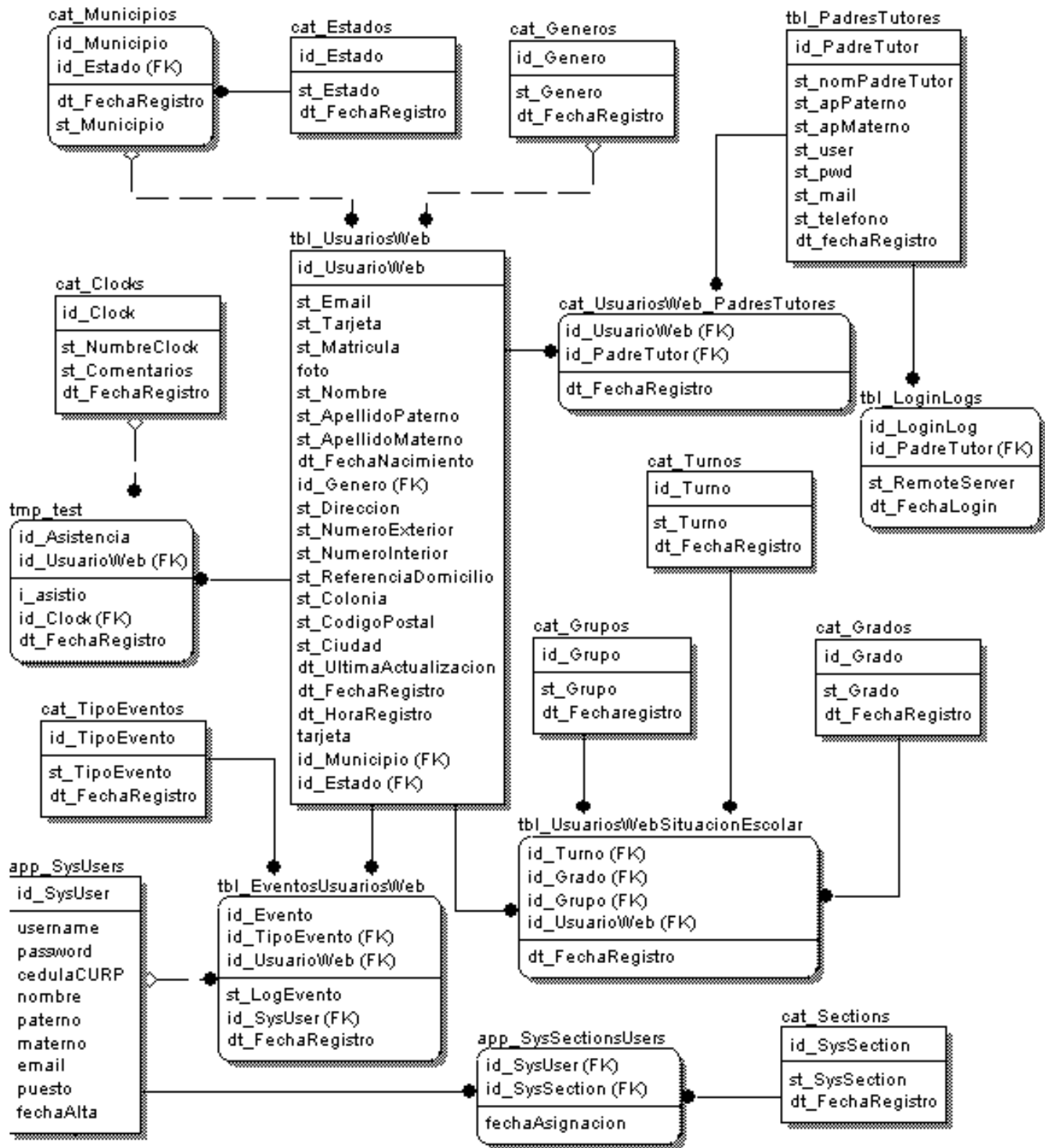


FIG. 4.1 Diagrama Entidad Relación – Modelo Lógico

4.2.3 DIAGRAMA FÍSICO

En el capítulo 3 se presentó el diagrama físico de componentes, en donde se ilustró la forma en que estos componentes interactúan entre sí a través de una red local (LAN) para el módulo de credencialización, para el registro de asistencias y para la consulta de todos los registros generados a través del portal web.

En la figura 4.2 se presenta el diagrama físico de la base de datos. A diferencia del diagrama lógico Entidad-Relación, se muestran los tipos de dato que se asignó.

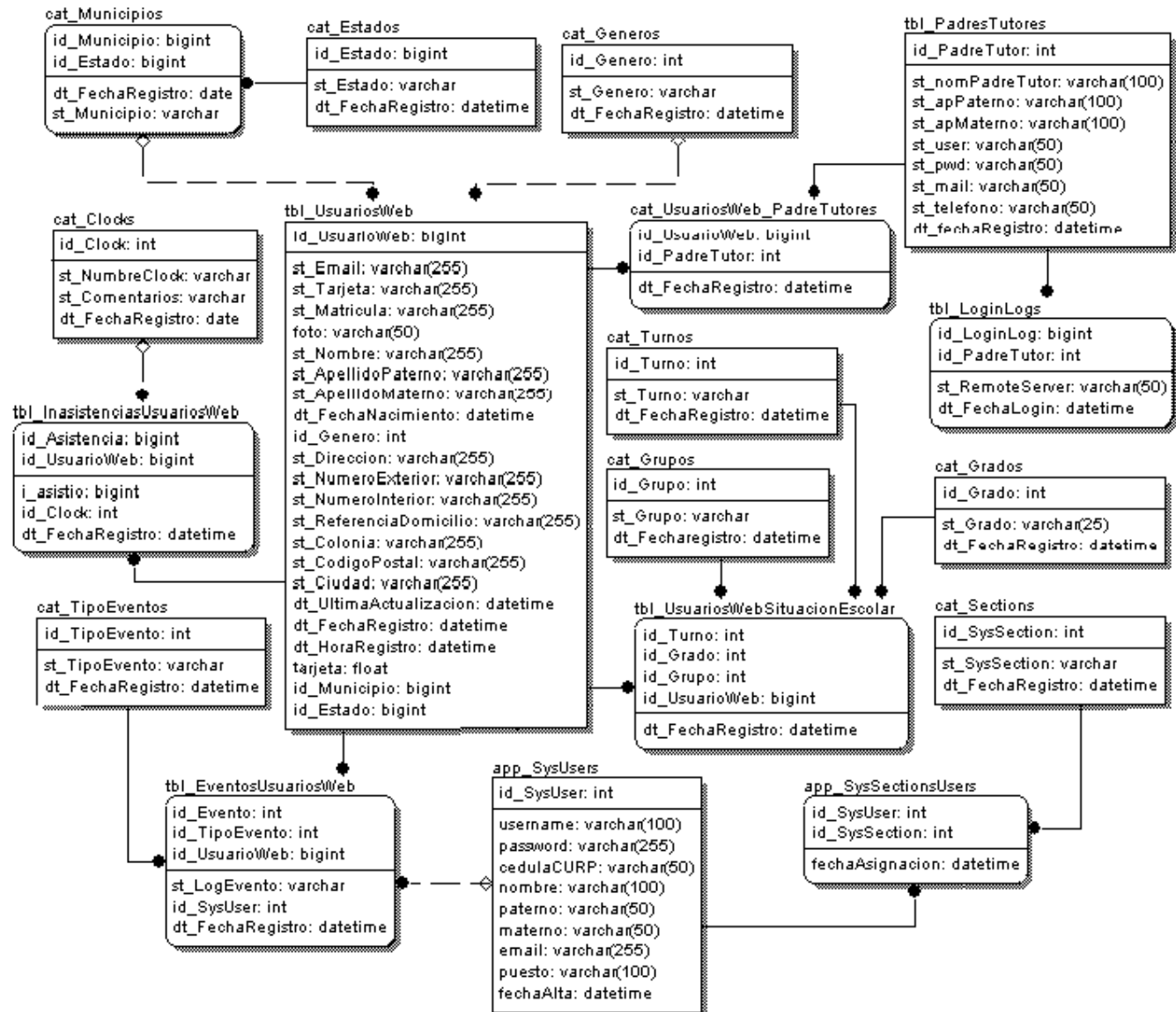


FIG. 4.2 Diagrama Entidad Relación – Modelo Físico.

4.2.4 SCRIPT SQL DE BASE DE DATOS MODELADA

A continuación se documenta el código SQL generado a partir del diagrama físico de la base de datos, el cuál se puede ejecutar dentro de un analizador de sentencias SQL dentro de un manejador de bases de datos. Este script ha sido utilizado para la Base De Datos que se utiliza para el funcionamiento actual del sistema.

```
CREATE TABLE app_SysUsers (
    id_SysUser int IDENTITY(1, 1),
```

```

username      varchar(100) NULL,
password      varchar(255) NULL,
cedulaCURP    varchar(50) NULL,
nombre        varchar(100) NULL,
paterno       varchar(50) NULL,
materno       varchar(50) NULL,
email         varchar(255) NULL,
puesto        varchar(100) NULL,
fechaAlta     datetime NULL,
PRIMARY KEY (id_SysUser)
)
go

```

```

CREATE TABLE cat_Estados (
  id_Estado    bigint NOT NULL,
  st_Estado    varchar NOT NULL,
  dt_FechaRegistro  datetime NOT NULL,
  PRIMARY KEY (id_Estado)
)
go

```

```

CREATE TABLE cat_Municipios (
  id_Municipio  bigint NOT NULL,
  id_Estado     bigint NOT NULL,
  st_Municipio  varchar NOT NULL,
  dt_FechaRegistro  date NOT NULL,
  PRIMARY KEY (id_Municipio, id_Estado),
  FOREIGN KEY (id_Estado)
    REFERENCES cat_Estados
)
go

```

```

CREATE TABLE cat_Generos (
  id_Genero     int NOT NULL,
  st_Genero     varchar NOT NULL,
  dt_FechaRegistro  datetime NOT NULL,
  PRIMARY KEY (id_Genero)
)
go

```

```

CREATE TABLE tbl_UsuariosWeb (
  id_UsuarioWeb  bigint IDENTITY(1, 1),
  st_Email       varchar(255) NULL,
  st_Tarjeta     varchar(255) NULL,
  st_Matricula   varchar(255) NULL,
  foto           varchar(50) NULL,
  st_Nombre      varchar(255) NULL,
  st_ApellidoPaterno  varchar(255) NULL,
  st_ApellidoMaterno  varchar(255) NULL,
  dt_FechaNacimiento  datetime NULL,
  id_Genero      int NULL,
  st_Direccion   varchar(255) NULL,
  st_NumeroExterior  varchar(255) NULL,
  st_NumeroInterior  varchar(255) NULL,

```

```

    st_ReferenciaDomicilio varchar(255) NULL,
    st_Colonia            varchar(255) NULL,
    st_CodigoPostal      varchar(255) NULL,
    st_Ciudad            varchar(255) NULL,
    dt_UltimaActualizacion datetime NULL,
    dt_FechaRegistro    datetime NULL,
    dt_HoraRegistro     datetime NULL,
    tarjeta              float NULL,
    id_Municipio         bigint NULL,
    id_Estado            bigint NULL,
    PRIMARY KEY (id_UsuarioWeb),
    FOREIGN KEY (id_Municipio, id_Estado)
        REFERENCES cat_Municipios,
    FOREIGN KEY (id_Genero)
        REFERENCES cat_Generos
)
go

```

```

CREATE TABLE cat_TipoEventos (
    id_TipoEvento    int NOT NULL,
    st_TipoEvento    varchar NOT NULL,
    dt_FechaRegistro datetime NOT NULL,
    PRIMARY KEY (id_TipoEvento)
)
go

```

```

CREATE TABLE tbl_EventosUsuariosWeb (
    id_Evento        int NOT NULL,
    id_TipoEvento    int NOT NULL,
    id_UsuarioWeb    bigint NOT NULL,
    st_LogEvento     varchar NULL,
    id_SysUser       int NULL,
    dt_FechaRegistro datetime NULL,
    PRIMARY KEY (id_Evento, id_TipoEvento, id_UsuarioWeb),
    FOREIGN KEY (id_SysUser)
        REFERENCES app_SysUsers,
    FOREIGN KEY (id_UsuarioWeb)
        REFERENCES tbl_UsuariosWeb,
    FOREIGN KEY (id_TipoEvento)
        REFERENCES cat_TipoEventos
)
go

```

```

CREATE TABLE cat_Sections (
    id_SysSection    int NOT NULL,
    st_SysSection    varchar NOT NULL,
    dt_FechaRegistro datetime NOT NULL,
    PRIMARY KEY (id_SysSection)
)
go

```

```

CREATE TABLE cat_Clocks (
    id_Clock         int NOT NULL,
    st_NumbreClock   varchar NOT NULL,
    st_Comentarios   varchar NOT NULL,
    dt_FechaRegistro date NOT NULL,
    PRIMARY KEY (id_Clock)
)

```



```
)  
go
```

```
CREATE TABLE cat_Grupos (  
    id_Grupo      int NOT NULL,  
    st_Grupo      varchar NOT NULL,  
    dt_Fecharegistro  datetime NOT NULL,  
    PRIMARY KEY (id_Grupo)  
)
```

```
)  
go
```

```
CREATE TABLE cat_Grados (  
    id_Grado      int NOT NULL,  
    st_Grado      varchar(25) NOT NULL,  
    dt_FechaRegistro  datetime NULL,  
    PRIMARY KEY (id_Grado)  
)
```

```
)  
go
```

```
CREATE TABLE cat_Turnos (  
    id_Turno      int NOT NULL,  
    st_Turno      varchar NOT NULL,  
    dt_FechaRegistro  datetime NOT NULL,  
    PRIMARY KEY (id_Turno)  
)
```

```
)  
go
```

```
CREATE TABLE tbl_UsuariosWebSituacionEscolar (  
    id_Turno      int NOT NULL,  
    id_Grado      int NOT NULL,  
    id_Grupo      int NOT NULL,  
    id_UsuarioWeb  bigint NOT NULL,  
    dt_FechaRegistro  datetime NOT NULL,  
    PRIMARY KEY (id_Turno, id_Grado, id_Grupo, id_UsuarioWeb),  
    FOREIGN KEY (id_UsuarioWeb)  
        REFERENCES tbl_UsuariosWeb,  
    FOREIGN KEY (id_Grupo)  
        REFERENCES cat_Grupos,  
    FOREIGN KEY (id_Grado)  
        REFERENCES cat_Grados,  
    FOREIGN KEY (id_Turno)  
        REFERENCES cat_Turnos  
)
```

```
)  
go
```

```
CREATE TABLE tbl_InasistenciasUsuariosWeb (  
    id_Asistencia  bigint IDENTITY(1, 1),  
    id_UsuarioWeb  bigint NOT NULL,  
    i_asistio      bigint NULL,  
    id_Clock       int NULL,  
    dt_FechaRegistro  datetime NULL,  
    PRIMARY KEY (id_Asistencia, id_UsuarioWeb),  
    FOREIGN KEY (id_Clock)  
        REFERENCES cat_Clocks,  
    FOREIGN KEY (id_UsuarioWeb)  
        REFERENCES tbl_UsuariosWeb  
)
```

```
)  
go
```

```

CREATE TABLE tbl_PadresTutores (
  id_PadreTutor    int IDENTITY(1, 1),
  st_nomPadreTutor varchar(100) NULL,
  st_apPaterno     varchar(100) NULL,
  st_apMaterno     varchar(100) NULL,
  st_user          varchar(50) NULL,
  st_pwd           varchar(50) NULL,
  st_mail          varchar(50) NULL,
  st_telefono      varchar(50) NULL,
  dt_fechaRegistro datetime NULL,
  PRIMARY KEY (id_PadreTutor)
)
go

```

```

CREATE TABLE tbl_LoginLogs (
  id_LoginLog      bigint IDENTITY(1, 1),
  id_PadreTutor    int NOT NULL,
  st_RemoteServer  varchar(50) NULL,
  dt_FechaLogin    datetime NULL,
  PRIMARY KEY (id_LoginLog, id_PadreTutor),
  FOREIGN KEY (id_PadreTutor)
    REFERENCES tbl_PadresTutores
)
go

```

```

CREATE TABLE cat_UsuariosWeb_PadreTutores (
  id_UsuarioWeb    bigint NOT NULL,
  id_PadreTutor    int NOT NULL,
  dt_FechaRegistro datetime NULL,
  PRIMARY KEY (id_UsuarioWeb, id_PadreTutor),
  FOREIGN KEY (id_UsuarioWeb)
    REFERENCES tbl_UsuariosWeb,
  FOREIGN KEY (id_PadreTutor)
    REFERENCES tbl_PadresTutores
)
go

```

```

CREATE TABLE app_SysSectionsUsers (
  id_SysUser       int NOT NULL,
  id_SysSection    int NOT NULL,
  fechaAsignacion  datetime NULL,
  PRIMARY KEY (id_SysUser, id_SysSection),
  FOREIGN KEY (id_SysSection)
    REFERENCES cat_Sections,
  FOREIGN KEY (id_SysUser)
    REFERENCES app_SysUsers
)
go

```

CAPÍTULO 5

MANUAL TÉCNICO

CAPÍTULO 5 MANUAL TÉCNICO.

5.1. Sección Administrador

El “Sistema de cómputo en línea para control de asistencias” consta de una sección de administrador, en la cual se dan de alta los alumnos con su respectiva credencial, se generan usuarios de acceso al sistema, tanto para la administración como para la consulta del mismo y se pueden generar reportes personalizados.

Para ingresar a la sección de administración es necesario contar con una cuenta de usuario provista por el administrador del sistema. Una vez ingresado a dicha sección desde el explorador de internet, se mostrará la pantalla inicial que muestra la figura 5.1.



FIG. 5.1 Pantalla inicial de acceso al módulo de administración del *Sistema De Cómputo En Línea Para Control De Asistencias*

Una vez que son ingresados los datos solicitados, la sección de administrador muestra una interfaz como la mostrada en la figura 5.2. El título de la página, el nombre del proyecto, el nombre del cliente y el pie de página son variables globales de todo el proyecto, es decir, tienen valores fijos que nunca cambian y que aplican a lo largo y ancho del proyecto.

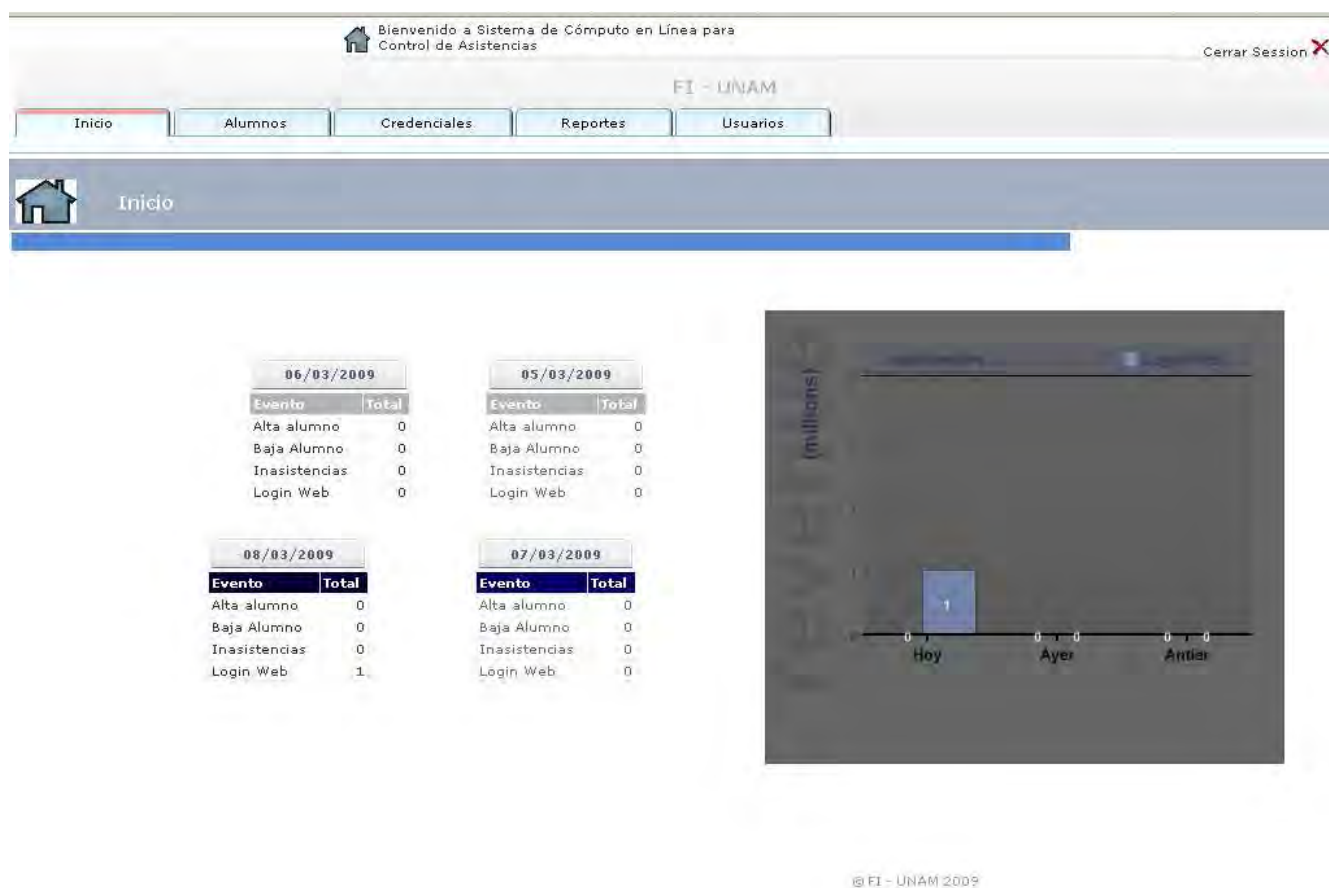


FIG. 5.2 Pestaña de inicio. Gráficas y tablas de actividad reciente en el sistema

Esta pantalla inicial consta de 2 frames, distribuidos de manera vertical. El primero de ellos, en la parte superior, muestra el menú de navegación que aplicará para toda la sección de administrador. En él se pueden observar todas y cada una de las actividades que como usuario administrador se pueden llevar a cabo. En el segundo se muestra todo el contenido que se desplegará al ingresar a cada una de las secciones enlistadas en el menú de navegación.

La pantalla que por default se muestra al ingresar al sistema, corresponde a la opción "Inicio" del menú de navegación. Se muestra un resumen de la actividad que recientemente se ha dado en el sistema. Por un lado se nos muestran cuatro tablas con registros capturados en los últimos cuatro días, incluyendo el día actual de la consulta. Por otro lado, se muestra la actividad que en específico se ha tenido en el portal web de consulta de asistencias en los últimos 3 días, incluyendo el día actual de la consulta.

5.1.1 EXPLICACIÓN DE PESTAÑAS

Como ya se mencionó en el anterior subtema, la pantalla inicial mostrada al ingresar al sistema, corresponde a la opción "Inicio" en el menú de navegación. A continuación se documentarán cada una del resto de las opciones del menú correspondientes a los diferentes módulos de los que se compone la sección de administrador.

Alumnos

Esta sección contiene pantallas que muestran al usuario administrador información referente a los alumnos registrados. Información como nombres y matriculas de los alumnos, inasistencias registradas, accesos de los padres de familia al portal web de consulta y los mensajes de texto que se han enviado a los padres de familia.

En cada una de las opciones mostradas con resumen se puede ver un mayor detalle dando clic a cada una de ellas. Todas estas opciones se muestran en la figura 5.3, en la cual se encuentra seleccionada la pestaña “Alumnos”.



FIG. 5.3 Pestaña de alumnos. Tabla de estadística del total de actividades hechas con los alumnos.

En la figura 5.4 se muestra el detalle de dar clic en el enlace “Alumnos”. Se muestran los nombres y las claves.



FIG. 5.4 Pestaña de alumnos. Tabla de lista general de alumnos registrados en el sistema.

Sección de Alumnos. Muestra un listado de todos los alumnos registrados y su correspondiente clave dentro del sistema. Se muestran ordenados alfabéticamente. Se puede hacer uso de las opciones de paginación para poder ver más alumnos por página (10, 25, 50 y 100). Si se desea dar de alta un nuevo alumno, bastará con dar clic al link “Registro Nuevo”, mostrado en el extremo derecho superior de la ventana de alumnos desplegada. La interfaz mostrada es como la de la figura 5.5:

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Sesión X

FI - LINAM

[Inicio](#) | [Alumnos](#) | [Credenciales](#) | [Reportes](#) | [Usuarios](#)

Alumnos

ALTA DE USUARIOS

DATOS PERSONALES

Nombre(s) de Alumno:

Apellido Paterno:

Apellido Materno:

Teléfono Casa:

Teléfono Celular:

Fecha de Nacimiento:

Sexo: Masculino Femenino

DIRECCIÓN

Calle y Número:

Colonia:

Código Postal:

Delegación/Municipio:

Estado:

DATOS ESCOLARES

Matrícula:

Tarjeta:

Información de Alumnos

Alumnos: 1,705

Inasistencias: 17,080

Envíos SMS: 489

Login Web: 36

Seleccione la información de Alumnos que desee consultar

Registro Nuevo

Registros 1 al 10 de 1,705

Mostrar

Anterior | 1 | 2 | 3 | Siguiente

Alumno	Clave
123123 123123 123123	123123
Abdiel Santin Torres	00329
Abigail Aguilar Gutierrez	00531
Abigail Cortes Hernandez	00758
Abigail Gallegos Valdivieso	00187
Abigail Gonzalez Morales	08697
Abigail Jalmez Merida	00285
Abisai Campechano Silverio	08900
Abraham Adolfo Ramirez Galindo	08632
Abraham Rogelio Laguna Garcia	00768

FIG. 5.5 Pestaña de alumnos. Pantalla de alta de nuevos alumnos y de padres / tutores.

Se deberán ingresar todos los datos solicitados, de lo contrario el sistema a través de una validación mostrará una serie de mensajes alerta que indicarán la información obligatoria por capturar. Durante el alta de un alumno nuevo, se registra el padre o tutor que tendrá acceso a consultar en el portal web la información almacenada de dicho alumno. Para el caso en que el padre de familia ya exista en el sistema, el sistema automáticamente mostrará las opciones correspondientes, basándose en coincidencias con el nombre y/o información que el administrador del sistema vaya ingresando en los campos correspondientes, estos campos se muestran en la figura 5.6:

Bienvenido a Sistema de Cómputo en Línea para
Control de Asistencias

Cerrar Sesión X

EL - LINAM

Inicio Alumnos Credenciales Reportes Usuarios

Alumnos

Delegación/Municipio:

Estado: --> Seleccione un Estado v

DATOS ESCOLARES

Matrícula:

Tarjeta:

Grado: Seleccione --> v

Grupo: Seleccione --> v

Turno: Seleccione --> v

Datos del Padre

Buscar Padre:

Coincidencias con Felipe

No se encontraron coincidencias con el criterio de búsqueda Felipe

ALTA DE INFORMACIÓN DEL PADRE/TUTOR

Nombre:

Apellido Paterno:

Apellido Materno:

Email:

Teléfono:

Cancelar Agregar

Información de Alumnos

Alumnos 1,705

Inasistencias 17,080

Envíos SMS 480

Login Web 300

Seleccione la información de Alumnos que desee consultar

Registro Nuevo

Registros 1 al 10 de 1,705

Mostrar v

Anterior | 1 | 2 | 3 | Siguiente

Alumno	Clave
123123 123123 123123	123123
Abdiel Santin Torres	00329
Abigail Aguilar Gutierrez	00531
Abigail Cortes Hernandez	00758
Abigail Gallegos Valdivieso	00187
Abigail Gonzalez Morales	08697
Abigail Jaimes Merida	00285
Abisai Campechano Silverio	08900
Abraham Adolfo Ramirez Galindo	08632
Abraham Rogelio Laguna Garcia	00768

FIG. 5.6 Pestaña de alumnos. Alta de padres de familia o tutores.

Para el ejemplo mostrado en la figura 5.6, el sistema no detectó coincidencias con el nombre "Felipe" por lo que será necesario introducir la información correspondiente al padre/tutor del alumno, sin embargo para el ejemplo de la figura 5.7 se muestra, el sistema detecta más de una coincidencia, por lo que despliega las opciones.

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Sesión X

FI - UNAM

Inicio **Alumnos** Credenciales Reportes Usuarios

DATOS ESCOLARES

Matrícula:

Tarjeta:

Grado:

Grupo:

Turno:

Datos del Padre

Buscar Padre:

Coincidencias con f

- **Francisco Corona Olmedo** (Telefono : 5556581111)
Matricula: 108001 - DERIANNT MANUEL ALPIZAR VILLAR
Vespertino , Grado: 1 , Grupo: A
- **Cesar Vinicio Amador Cardenas**
Matricula: 108002 - CESAR VINICIO AMADOR CARDENAS
Vespertino , Grado: 1 , Grupo: A
- **Rodrigo Ivan Amuchategui Salinas**
Matricula: 108003 - RODRIGO IVAN AMUCHATEGUI SALINAS
Vespertino , Grado: 1 , Grupo: A
- **Fernando Ruiz Salazar** (Telefono : 55694160)
Matricula: 108004 - DIANA KAREN ANSELMO AMARO
Vespertino , Grado: 1 , Grupo: A

Cancelar Agregar

Envíos SMS: 489
Login Web: 36
Seleccione la información de Alumnos que desee consultar

Registro Nuevo

Registros 1 al 10 de 1,705 Anterior | 1 | 2 | 3 | Siguiente

Mostrar

Alumno	Clave
123123 123123 123123	123123
Abdiel Santin Torres	00329
Abigail Aguilar Gutierrez	00531
Abigail Cortes Hernandez	00758
Abigail Gallagos Valdivieso	00167
Abigail Gonzalez Morales	08697
Abigail Jaimes Merida	00285
Abisai Campechano Silverio	08900
Abraham Adolfo Ramirez Galindo	08632
Abraham Rogelio Laguna Garcia	00768

FIG. 5.7 Pestaña de alumnos. Selección de padres de familia previamente registrados.

Se muestra el nombre completo del padre, así como el(los) alumnos que ya tenga asignados en el sistema, esto con la finalidad de dar un mayor nivel de seguridad en la selección del mismo. Bastará con seleccionar alguna de las opciones mostradas para que ya no sea necesario volver a capturar todos los datos del padre/tutor.

Sección de Inasistencias. Tal como en el ejemplo de la figura 5.8 dentro del sistema se enlistan todas las inasistencias generadas a partir del inicio del presente ciclo escolar, ordenadas descendientemente por fecha, es decir, la más reciente es la que se muestra al inicio.

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Session X

FI - UNAM

Inicio **Alumnos** Credenciales Reportes Usuarios

Alumnos

Información de Alumnos

Alumnos: 1,705
Inasistencias: 17,078
Envíos SMS: 489
Login Web: 35

Seleccione la información de Alumnos que desee consultar

Inasistencias

Registros 1 al 8 de 17,078 Anterior | 1 | 2 | 3 | Siguiente

Mostrar

Alumno	Fecha Inasistencia
KAREN DANIELA CHIÑAS GARCIA	6 Mar 2009 5:00PM
BRENDA ITZEL CASTRO LOPEZ	6 Mar 2009 5:00PM
MARIA FERNANDA CARRILLO RUIZ	6 Mar 2009 5:00PM
CYNTHIA SAMANTHA CASTILLO RODRIGUEZ DE SAN MIGUEL	6 Mar 2009 5:00PM
FERNANDO ABRAHAM CARMONA MENDOZA	6 Mar 2009 5:00PM
ITZEL XIMENA CAMACHO OVIEDO	6 Mar 2009 5:00PM
FERNANDA CAMPOS ROBLES	6 Mar 2009 5:00PM
ITZAMAREY VIRDIANA AMAYA ROA	6 Mar 2009 5:00PM

FIG. 5.8 Pestaña de alumnos. Tabla de inasistencias de alumnos.

Sección de últimos envíos de SMS realizados. En esta sección se enlistan todos los alumnos que han recibido las últimas notificaciones vía mensaje de texto. En la figura 5.9 se ilustra un ejemplo.

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Session X

FI - UNAM

Inicio Alumnos Credenciales Reportes Usuarios

Alumnos

Información de Alumnos

Alumnos	1,705
Inasistencias	17,078
Envíos SMS	489
Login Web	35

Seleccione la información de Alumnos que desee consultar

Envíos SMS

Registros 1 al 10 de 489 Anterior | 1 | 2 | 3 | Siguiente

Mostrar

Alumno	Contenido	Fecha envio
MARIA ITZANAYA MALAGON VILLEGAS	MENSAJE SMS - PRUEBA 1	27/05/2008
BRIGET SAMANTHA LOPEZ LEDESMA	MENSAJE SMS - PRUEBA 1	27/05/2008
KARLA NAYELI ISLAS REYES	MENSAJE SMS - PRUEBA 1	27/05/2008
OMAR HERRERA MENESES	MENSAJE SMS - PRUEBA 1	27/05/2008
MARA DANIELA HERNANDEZ OLGUIN	MENSAJE SMS - PRUEBA 1	27/05/2008
DIANA GONZALEZ GRANILLO	MENSAJE SMS - PRUEBA 1	27/05/2008
OMAR GARCIA HERNANDEZ	MENSAJE SMS - PRUEBA 1	27/05/2008
EDUARDO ULISES GALICIA GALICIA	MENSAJE SMS - PRUEBA 1	27/05/2008
KENIA ALONDRA ECHEVERRIA GARCIA	MENSAJE SMS - PRUEBA 1	27/05/2008

FIG. 5.9 Pestaña de alumnos. Últimos envíos de SMS realizados.

Sección de Envíos SMS. Se enlista el total de mensajes de texto enviados durante el ciclo escolar. Se muestra dentro del sistema (tal como la figura 5.10) el nombre del alumno cuyo padre de familia (tutor y/o responsable registrado en el sistema) recibió el mensaje de texto y la fecha de envío.

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Session X

FI - UNAM

Inicio Alumnos Credenciales Reportes Usuarios

Alumnos

Información de Alumnos

Alumnos	1,705
Inasistencias	17,078
Envíos SMS	489
Login Web	35

Seleccione la información de Alumnos que desee consultar

Login Web

Registros 1 al de 0 Anterior | 1 | Siguiente

Mostrar

Padre/tutor	Fecha Login
Usuario De Prueba	08/03/2009
Usuario De Prueba	15/02/2009
Usuario De Prueba	15/02/2009
Usuario De Prueba	03/02/2009
Usuario De Prueba	02/02/2009
Usuario De Prueba	30/01/2009
Usuario De Prueba	30/01/2009
Usuario De Prueba	24/01/2009
Usuario De Prueba	24/01/2009
Usuario De Prueba	23/01/2009

FIG. 5.10 Pestaña de alumnos. Tabla de últimos accesos al portal Web.

Sección de Login Web. Se enlista el nombre de todos los padres de familia que han ingresado al portal web de consulta, así como la fecha en que efectuaron dicha acción.

CREDENCIALES

En esta sección se administra todo lo referente a las credenciales, asignación de nuevas credenciales, baja de credenciales y un apartado muy útil llamado Asistencia Manual, para cuando un alumno extravíe u olvide su credencial y se presente sin ella en los sensores de registro. El administrador podrá registrar sin necesidad de una credencial la asistencia del alumno.

La pantalla inicial que el sistema muestra al ingresar a esta sección es como la figura 5.11.



FIG. 5.11 Pestaña de credenciales. Pantalla inicial.

Asignar nuevas credenciales.

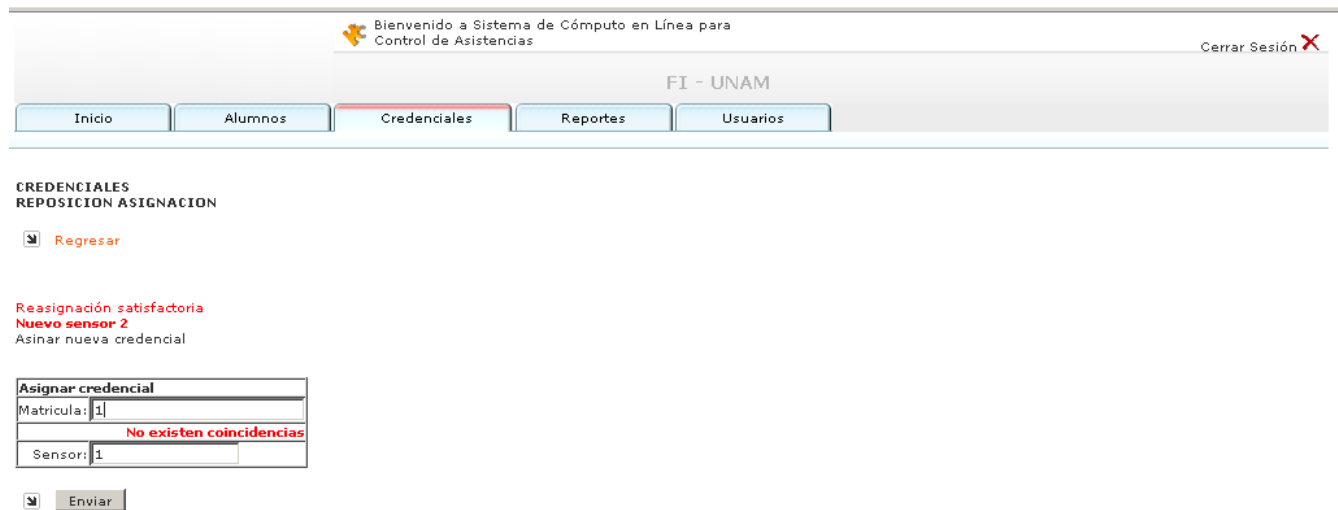


FIG. 5.12 Pestaña de credenciales. Reposición o reasignación de credenciales.

Se debe ingresar la matrícula del alumno cuya credencial se dará de alta, así como el número del sensor que tendrá asignado, tal como se muestra en la figura 5.12. Al presionar enviar, se llevará a cabo la asignación y el sistema mostrará el mensaje de éxito de la operación ejecutada. En la figura 5.13 se muestra un ejemplo de una asignación satisfactoria.

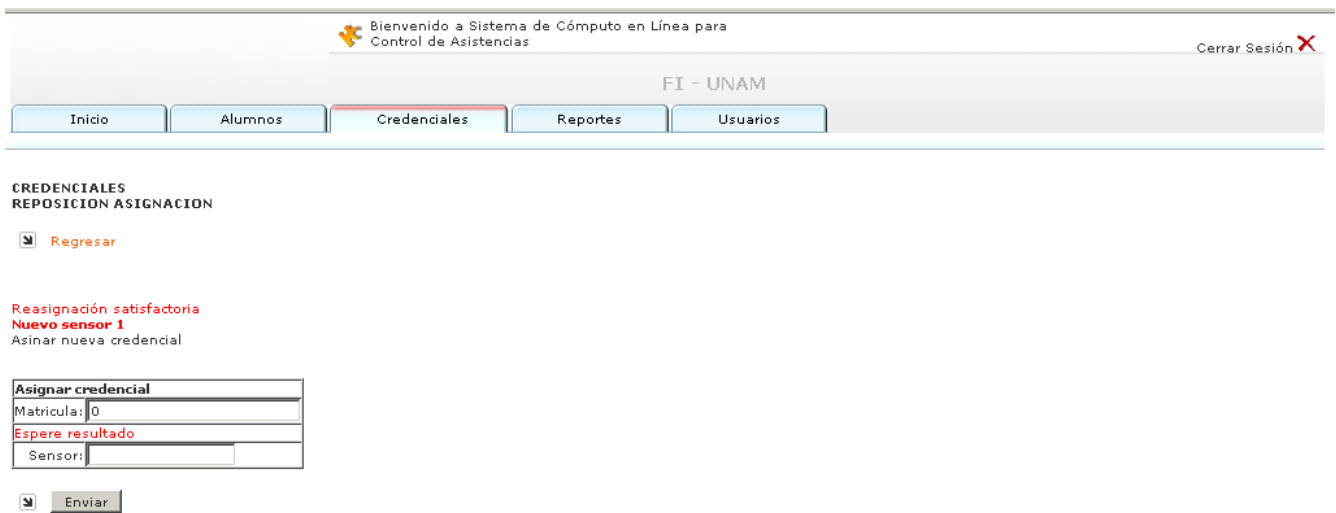


FIG. 5.13 Pestaña de credenciales. Resultado de reasignación de sensor.

Baja de credenciales.

Se debe ingresar la matrícula del alumno cuya credencial se dará de baja. En la figura 5.14 se muestran los campos solicitados.

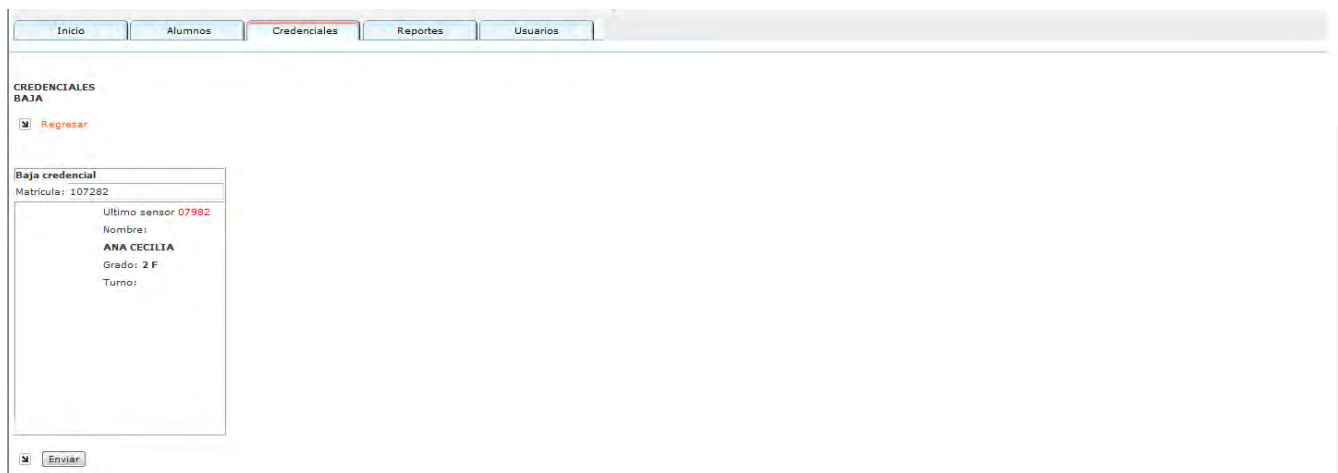


FIG. 5.14 Pestaña de credenciales. Baja de credenciales.

Al dar clic en enviar, el sistema automáticamente mostrará un mensaje que indique el éxito en la ejecución de la operación solicitada.

Asistencia manual.

Se debe ingresar la matrícula y el nombre del alumno que se desea registrar en la base de datos de asistencias (tal como se muestra en la figura 5.15). Al ingresar la información solicitada, el sistema mostrará las coincidencias halladas en los registros de la base de datos. Dar clic en “asistió” para generar la asistencia manual.

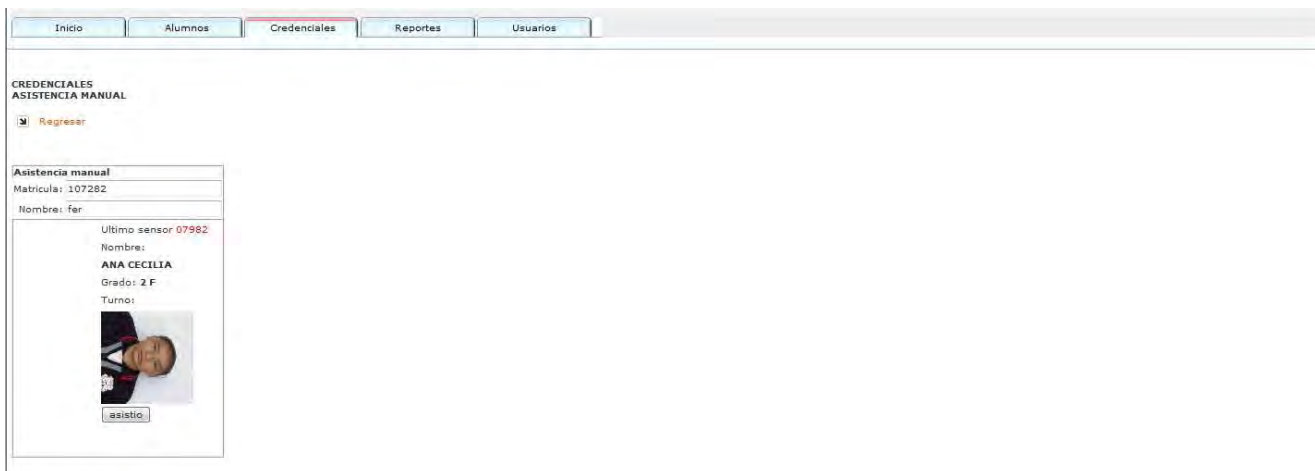


FIG. 5.15 Pestaña de credenciales. Registro de asistencia de forma manual.

REPORTES

En esta sección se enlistan todos los reportes que el administrador puede obtener del sistema y que puede personalizar conforme a sus necesidades, ya sea ajustando las fechas de consulta, seleccionando el turno, etc. La pantalla inicial mostrada en esta sección es la ejemplificada con la figura 5.16:

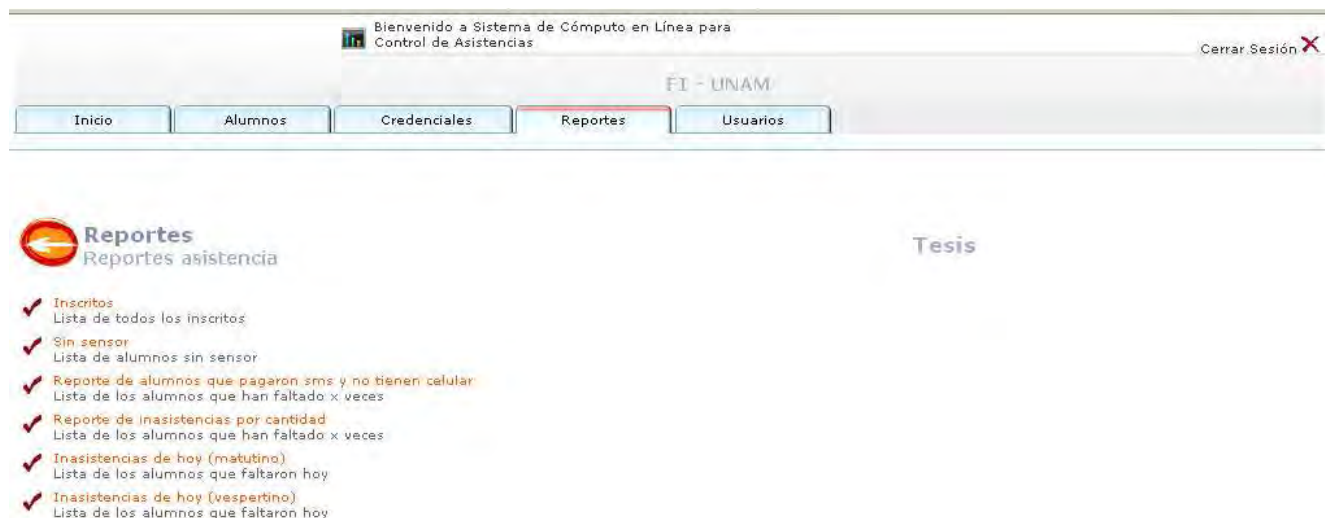


FIG. 5.16 Pestaña de reportes. Pantalla inicial.

Para ingresar a cualquiera de los reportes enlistados, bastará con dar un clic y para el caso en el que el reporte lo amerite, se solicitará que el usuario administrador seleccione algunos parámetros de consulta, a fin de obtener los resultados deseados.

REPORTE DE ALUMNOS INSCRITOS

En la figura 5.17 se ilustra un ejemplo de la pantalla inicial del reporte de alumnos inscritos por turno



FIG. 5.17 Pestaña de reportes. Lista de alumnos inscritos por turno.

Al seleccionar el turno, dar clic en generar reporte, los resultados se mostrarán tal como la figura 5.18.

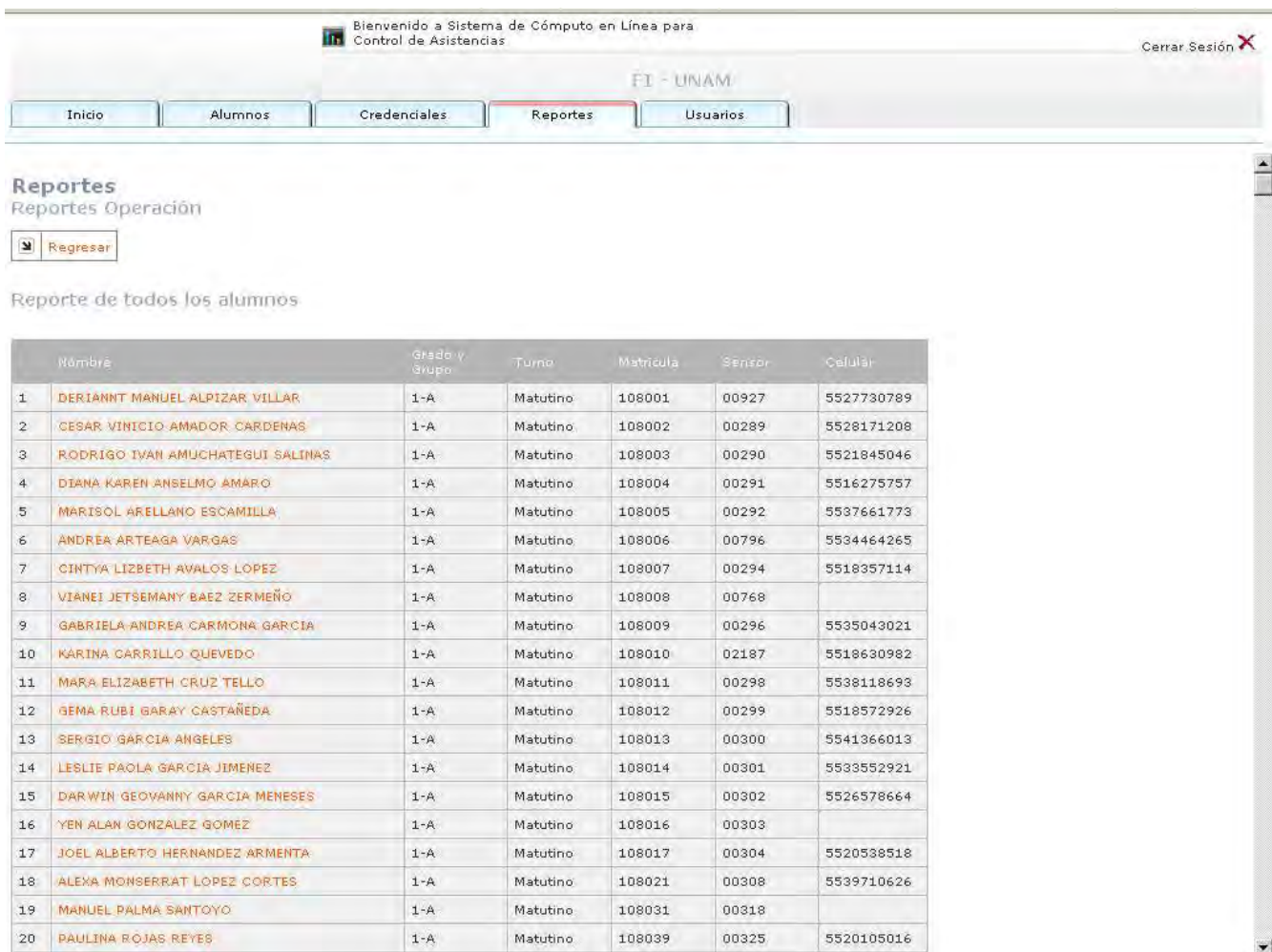


FIG. 5.18 Pestaña de reportes. Resultado de reporte de alumnos inscritos.

Se muestra el listado de todos los alumnos inscritos en el turno seleccionado, ordenados por grado, grupo y alfabéticamente.

REPORTE DE ALUMNOS SIN SENSOR

Reporte de alumnos que ya tienen asignada una matrícula y que están registrados en el sistema, pero aun no cuentan con una credencial y por ende dicha credencial aun no se ha asignado a algún sensor. Ejemplo, figura 5.19.

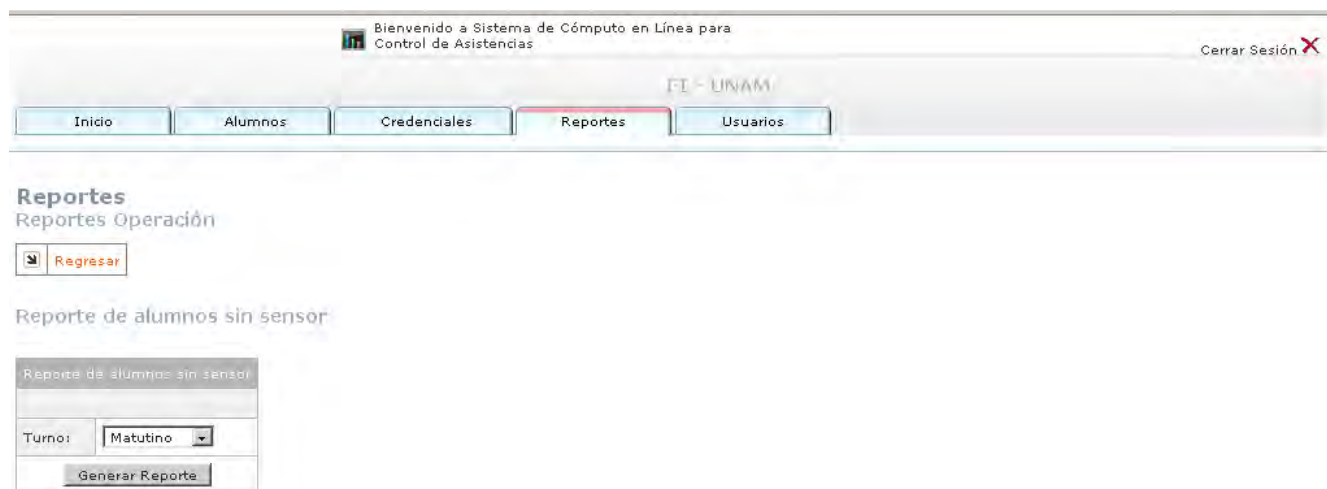


FIG. 5.19 Reporte de alumnos registrados en el sistema que aun no tienen un sensor asignado.

Seleccionar el turno que se desea consultar y a continuación dar clic en "Generar Reporte". Los resultados se mostrarán tal como la figura 5.20

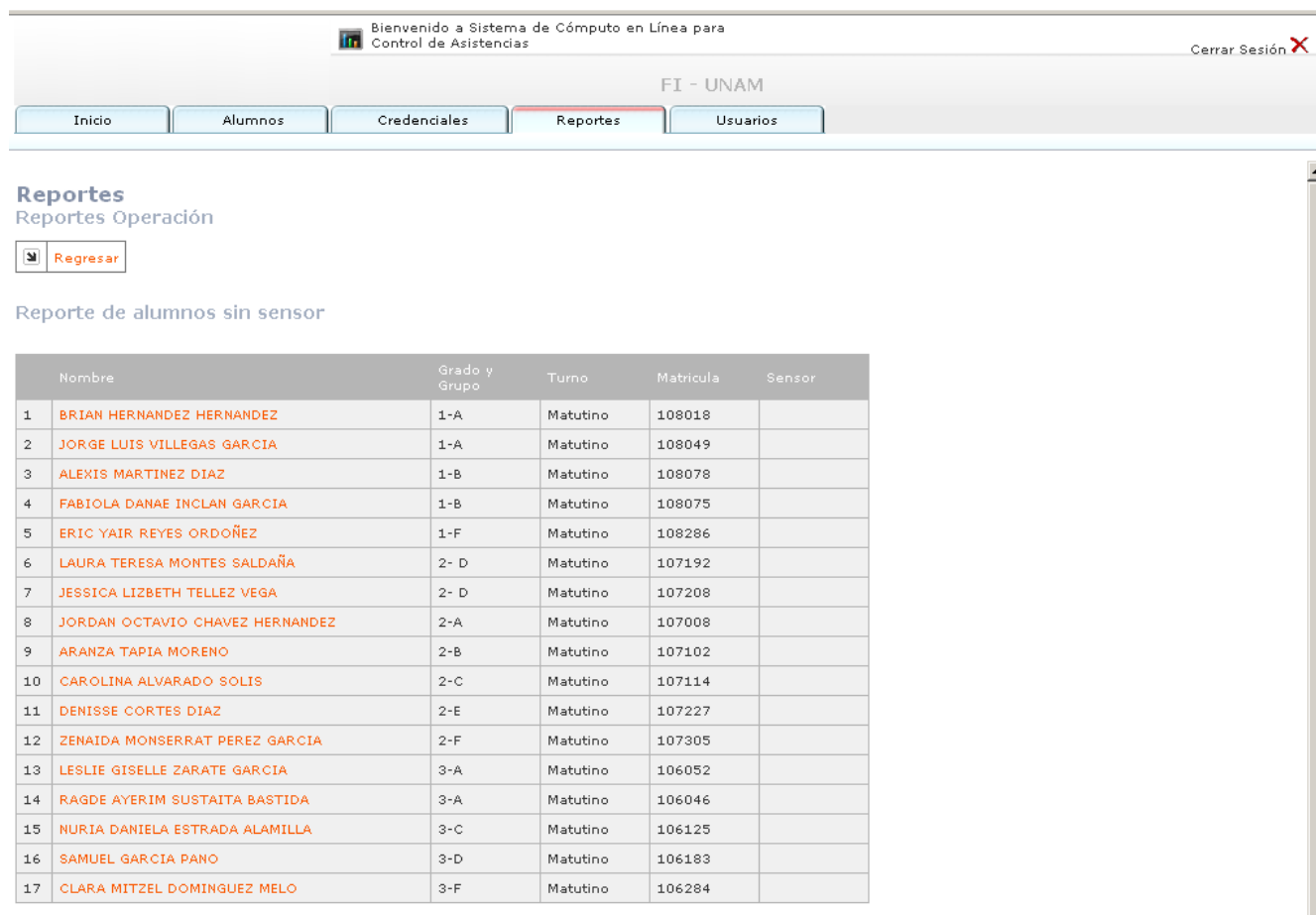


FIG. 5.20 Resultado de reporte de alumnos registrados en el sistema que aun no tienen un sensor asignado.

REPORTE DE ALUMNOS QUE PAGARON SMS Y NO TIENEN CELULAR

Reporte que muestra a todos los alumnos inscritos y registrados en el sistema, que pagaron el servicio de notificaciones por celular vía SMS y que actualmente no cuentan con un número de celular registrado en la base de datos del sistema. El resultado del reporte citado es como el ilustrado a continuación en la figura 5.21.

Bienvenido a Sistema de Cómputo en Línea para Control de Asistencias Cerrar Sesión X

FI - UNAM

Inicio Alumnos Credenciales **Reportes** Usuarios

Reportes
Reportes Operación

[Regresar](#)

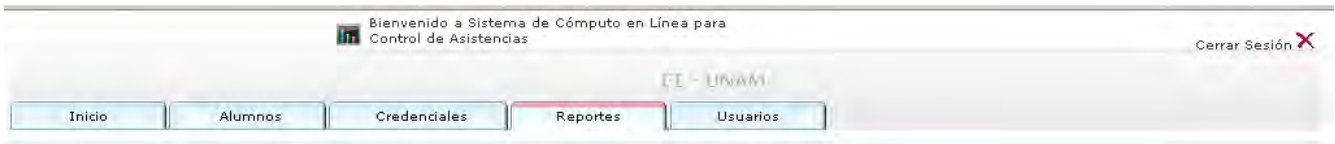
Reporte de todos los alumnos que pagaron y no tienen celular

	Nombre	Grado y grupo	Turno	Matrícula	Sensor
1	SAMANTHA MONSERRAT HERNANDEZ SANCHEZ	1-E	Vespertino	108214	00400
2	OSCAR DIAZ FRAGOSO	1-C	Vespertino	108112	00348
3	MAURICIO ESCAMILLA AGUINIGA	1-C	Vespertino	108114	00350
4	GABRIELA SALVADOR ARIAS	1-E	Vespertino	108239	00425
5	BRENDA JAIME HERNANDEZ	1-D	Vespertino	108178	00601
6	KENIA RUBI AYALA MARTINEZ	3-F	Vespertino	108309	00677
7	JAIR DE JESUS GARCIA LOPEZ	2-A	Vespertino	107013	08212
8	VANESSA Yael MONTES CHAPARRO	1-F	Vespertino	108282	00467
9	JOSE DE JESUS ROMERO MENDIOLA	1-F	Vespertino	108292	00478
10	EDUARDO NOE RODRIGUEZ DE LA CRUZ	3-C	Vespertino	206140	08969
11	SOFIA AIDA CORONA SANDOVAL	3-B	Vespertino	206056	08924
12	CLAUDIA ALEJANDRA BALTAZAR GARCIA	3-C	Vespertino	206101	08988
13	BRENDA DE LA CRUZ MELCHOR	2-D	Vespertino	207146	08794
14	GUILLERMO MARIN INIESTA	2-D	Vespertino	207156	08749
15	ADOLFO SANTIAGO CABALLERO	2-B	Vespertino	207085	00752
16	JESSICA LIZBETH GALLEGOS VALDIVIESO	2-C	Vespertino	207104	08751
17	ANA LIZBETH ROMERO ROMERO	2-A	Vespertino	207041	09506
18	DAVID AXEL JIMENEZ ZAVALA	2-B	Vespertino	207067	00779
19	JUAN ANTONIO MAZA GONZALEZ	1-E	Vespertino	208200	00899
20	JORDI YAIR HERRERA HERNANDEZ	1-C	Vespertino	208108	00108
21	JOSCELYN MEZTLI LOPEZ ZARAGOZA	3-F	Vespertino	106295	09144

FIG. 5.21 Pestaña de reportes. Reporte de alumnos registrados en el sistema que no tienen un número de celular registrado en el sistema y que han cubierto el monto del servicio.

REPORTE DE INASISTENCIAS POR CANTIDAD

Este reporte permite al usuario administrador ver todos los alumnos que están registrados en el sistema y que tienen cierta cantidad de faltas (dicha cantidad es seleccionada por el usuario administrador) y que pertenecen al turno seleccionado. Se deberán seleccionar los valores deseados de los combos que se muestran en la interfaz ejemplificada con la figura 5.22.



Reportes
Reportes Operación

[Regresar](#)

Reporte de Inasistencias

Reporte de inasistencias.

Seleccione rango de fechas de inscripción al programa.

Faltas:

Turno:

FIG. 5.22 Pestaña de reportes. Reporte de alumnos que tienen a partir de una inasistencia registrada en el sistema.

Se muestra el listado de alumnos ordenado por grado, grupo y alfabéticamente. Ver figura 5.23.



Reportes
Reportes Operación

[Regresar](#)

Reporte de alumnos con 3 faltas

Nombre	Grado y Grupo	Turno
SERGIO	1-A	Matutino
LESLIE PAOLA	1-A	Matutino
BRAYAN ALEJANDRO	1-A	Matutino
ABDIEL	1-A	Matutino
JESUS CARLOS ALEJANDRO	1-C	Matutino
MAURICIO	1-C	Matutino
KENYA JOCELYN	1-C	Matutino
ALLYSON ENID	1-C	Matutino
KEVIN	1-E	Matutino
JESSICA	1-E	Matutino
MARIA FERNANDA	1-E	Matutino
ALEXIS DAVID	1-E	Matutino
DAYHANA BERENICE	1-F	Matutino
ALEXIS	1-F	Matutino
JORGE GONZALO	1-F	Matutino
IRENE DONAJI	1-F	Matutino
FRANCISCO MANUEL	1-F	Matutino
KATIA LIZET	1-F	Matutino
CHRISTIAN	2- D	Matutino
ALEXIS STEFAN	1-B	Matutino

FIG. 5.23 Pestaña de reportes. Resultado de reporte de alumnos que tienen a partir de una inasistencia registrada en el sistema. Se muestra el ejemplo para tres inasistencias.

INASISTENCIAS DE HOY (MATUTINO)

Reporte que muestra las inasistencias generadas el día en que se efectúa la consulta, mostrando únicamente los alumnos del turno matutino. Ejemplo, figura 5.24.



Reportes

Reportes Operación

[Regresar](#)

Reporte de inasistencias hoy (09-03-2009) matutino

Credencial	Nombre	Grado y Grupo	Turno	Matricula
08372	ALEXIS JESUS EDUARDO OCHOA SANCHEZ	2- D	Matutino	107194

FIG. 5.24 Pestaña de reportes. Reporte de alumnos por turno que registraron inasistencia en el sistema en la fecha actual de la consulta.

INASISTENCIAS DE HOY (VESPERTINO)

Reporte similar al anterior que únicamente muestra las inasistencias generadas el día en que se lleva a cabo la consulta, mostrando únicamente los alumnos del turno vespertino. Ejemplo, figura 5.25



Reportes

Reportes Operación

[Regresar](#)

Reporte de inasistencias hoy (09-03-2009) vespertino

Credencial	Nombre	Grado y Grupo	Turno	Matricula
00011	FERNANDA JAZMIN COYOC CARDOSO	1-A	Vespertino	208011

FIG. 5.25 Pestaña de reportes. Reporte de alumnos por turno que registraron inasistencia en el sistema en la fecha actual de la consulta.

PESTAÑA DE USUARIOS

En esta sección se podrá observar y editar toda la información de los usuarios administradores del sistema que hasta el momento se hayan dado de alta, así mismo, se podrán dar de baja a los usuarios que se seleccione y también es posible dar de alta nuevos usuarios. En la figura 5.26 se ilustra la pantalla inicial de esta sección

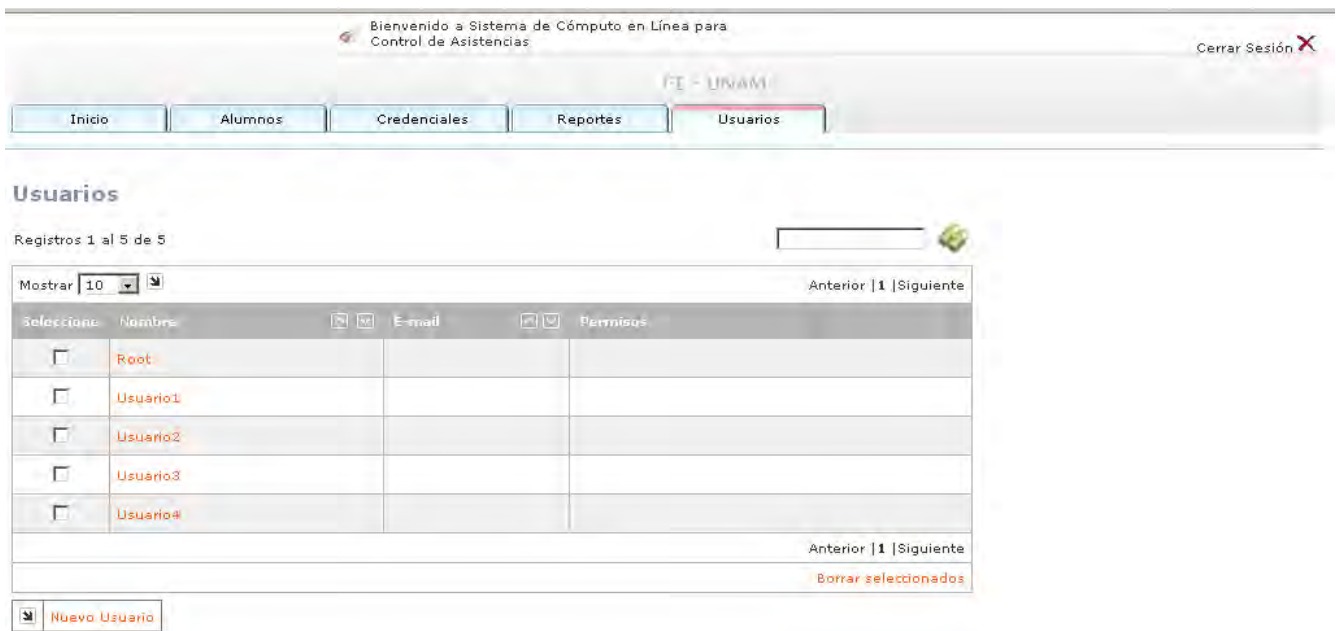


FIG. 5.26 Pestaña de usuarios. Pantalla inicial.

Al dar clic en el nombre de cualquier usuario se podrá editar su información personal almacenada en la base de datos. Para el caso en que existan muchos usuarios almacenados en la base de datos, puede utilizarse el buscador proporcionado para poder encontrar el usuario de interés. Ver figura 5.27.

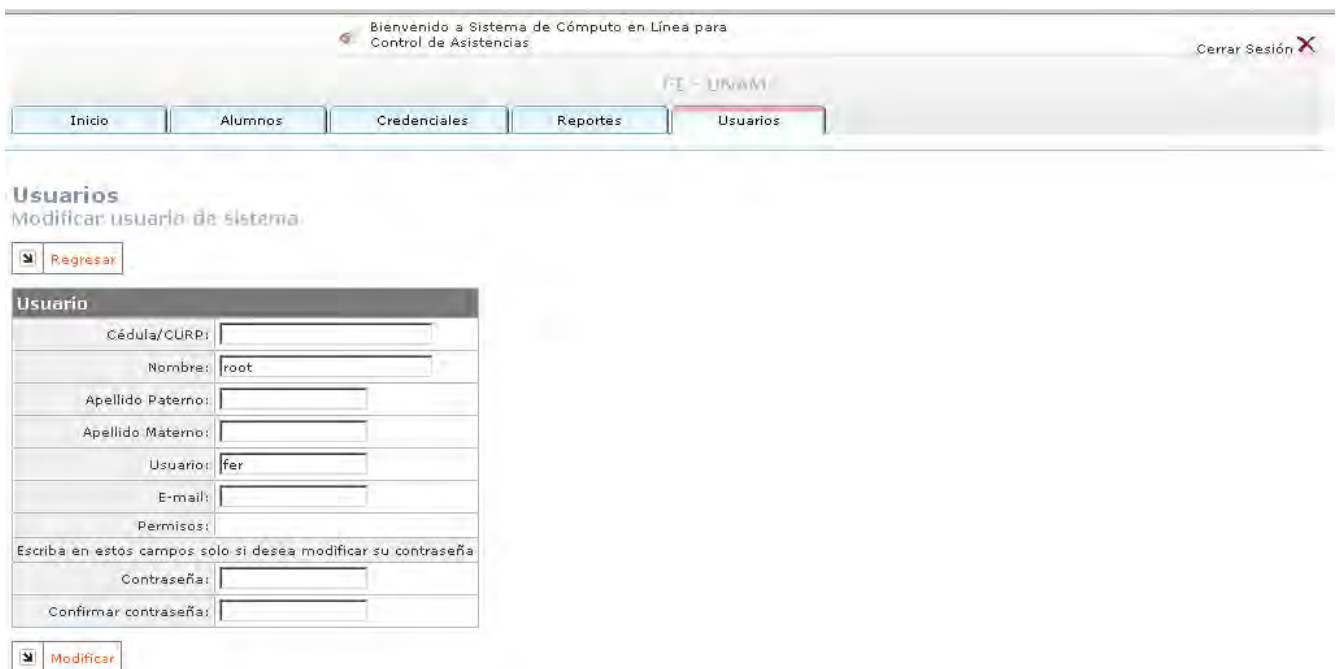


FIG. 5.27 Pestaña de usuarios. Edición de información capturada de los usuarios registrados.

Si se desea editar o agregar algún dato, bastará con introducirlo. Para guardarlo en la base de datos dar clic en modificar.

Al dar clic en el link "Nuevo Usuario" se mostrará una pantalla de captura como la de la figura 5.28.



FIG. 5.28 Pestaña de usuarios. Pantalla de registro de nuevos usuarios.

Ingresar la información mínima solicitada y para almacenarla en la base de datos dar clic en el “Agregar”.

Si se desea eliminar un usuario, es necesario seleccionar la casilla ubicada junto a su nombre y dar clic en “Borrar Seleccionados”. El sistema preguntará si se está seguro de la acción que esta por ejecutarse, el usuario administrador podrá decidir entre continuar o cancelar dicha operación. Ver ejemplo en figura 5.29.

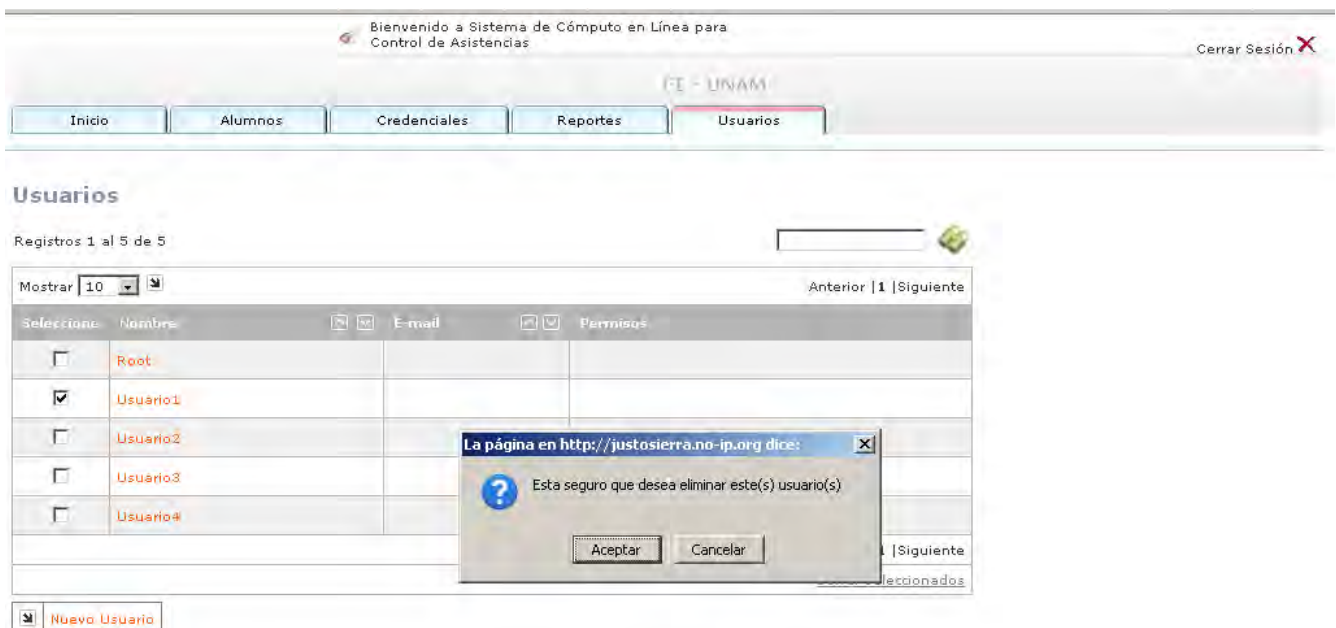


FIG. 5.29 Pestaña de usuarios. Baja de usuarios registrados.

5.1.2 PANTALLAS CON CÓDIGO Y EXPLICACIÓN DE LAS MISMAS

PESTAÑA DE INICIO

El siguiente código se encarga de generar las primeras tablas que muestran la reciente actividad que se ha registrado en el sistema. La primera consulta trae de la base de datos los tipos de eventos que se han dado de alta en el sistema: Login Web, Asistencias, Altas de Alumnos y Bajas de Alumnos. La segunda consulta se llevará a cabo tantas veces como lo indiquen la cantidad de eventos almacenados, esta segunda consulta trae de la base de datos la cantidad de registros que se han generado entre el día actual de la consulta y dos días previos.

```
<?php
$st_Query= "SELECT * FROM cat_TipoEventos WHERE id_TipoEvento in (3,8,5,1,4,9,2,15)
ORDER BY i_Orden;";

$rs_Resultado= mssql_query($st_Query);

while ($rc_Registro=mssql_fetch_array($rs_Resultado)){

echo "<tr class='answerCellBG'>";
echo "<td>".$rc_Registro['st_TipoEvento']."</td>";
$total_anio = 0;
$total_meses = 0;

$st_Query2="SELECT COUNT(*) AS eventos FROM dbo.tbl_EventosUsuariosWeb INNER JOIN
dbo.tbl_UsuariosWeb ON dbo.tbl_EventosUsuariosWeb.id_UsuarioWeb = dbo.tbl_UsuariosWeb.id_UsuarioWeb
WHERE (dbo.tbl_EventosUsuariosWeb.dt_FechaEvento = convert(datetime, ".$hoy.", 103) OR
(dbo.tbl_EventosUsuariosWeb.dt_FechaEvento BETWEEN CONVERT(datetime, ".$hoy." 00:00:00', 103) AND
CONVERT(datetime, ".$hoy." 23:59:59', 103))) AND (dbo.tbl_EventosUsuariosWeb.id_TipoEvento =
".$rc_Registro['id_TipoEvento'].");";

$rs_Resultado2= mssql_query($st_Query2);
while ($rc_Registro2=mssql_fetch_array($rs_Resultado2)){

echo "<td align='right'>".$rc_Registro2 ['eventos']."</td>\n";
$total_anio = $total_anio + $rc_Registro2 ['eventos'];
}

echo "</tr>";
}

?>
```

PESTAÑA DE ALUMNOS.

El siguiente código se encarga de dar de alta un nuevo alumno, así como el padre / tutor asignado al mismo.

```
<?
$search = str_replace(" ", "", $search0);
$searchshow = $search0;
//DECLARAMOS VARIABLES
$st_Tarjeta = $_POST['tarjeta'];
$st_Matricula = $_POST['matricula'];
$i_Turno = $_POST['turno'];
$i_grado = $_POST['grado'];
```

```

$st_grupo = $_POST['grupo'];
$st_Nombre = $_POST['nombre'];
$st_ApellidoPaterno = $_POST['aPat'];
$st_ApellidoMaterno = $_POST['aMat'];
$dt_FechaNacimiento = $_POST['tarjeta'];
$st_Direccion = $_POST['tarjeta'];
$st_NumeroExterior = $_POST['tarjeta'];
$st_NumeroInterior = $_POST['tarjeta'];
$st_ReferenciaDomicilio = $_POST['tarjeta'];
$st_Colonia = $_POST['tarjeta'];
$st_CodigoPostal = $_POST['tarjeta'];
$st_Ciudad = $_POST['tarjeta'];
$id_Genero = $_POST['sexo'];
$st_nomPadreTutor = $_POST['tarjeta'];
$st_apPaterno = $_POST['tarjeta'];
$st_apMaterno = $_POST['tarjeta'];
$st_user = $st_Matricula;
$st_mail = $_POST['tarjeta'];
$st_telefono = $_POST['tarjeta'];
$valorpapa = $_POST['tarjeta'];
$valorp = $_POST['tarjeta'];

```

//INSERTAMOS EN LA TABLA DE ALUMNOS

```

$queryalumnos = "INSERT INTO tbl_UsuariosWeb
(st_Tarjeta, st_Matricula, i_Turno, i_grado, st_grupo, st_Nombre, st_ApellidoPaterno,
st_ApellidoMaterno, dt_FechaNacimiento, st_Direccion, st_NumeroExterior, st_NumeroInterior,
st_ReferenciaDomicilio, st_Colonia, st_CodigoPostal, st_Ciudad, id_Genero)
VALUES
('.$st_Tarjeta.', ".$st_Matricula.", ".$i_Turno.", ".$i_grado.", ".$st_grupo.", ".$st_Nombre.",
" ".$st_ApellidoPaterno.", ".$st_ApellidoMaterno.", ".$dt_FechaNacimiento.", ".$st_Direccion.", ".
$st_NumeroExterior.", ".$st_NumeroInterior.", ".$st_ReferenciaDomicilio.", ".$st_Colonia.",
" ".$st_CodigoPostal.", ".$st_Ciudad.", ".$id_Genero.");
$queryalumnos = mssql_query($queryalumnos);

```

// Obtenemos el id del alumno

```

$querysel = "SELECT TOP 1 id_UsuarioWeb FROM tbl_UsuariosWeb
WHERE (st_Matricula = ".$st_Matricula.") ORDER BY id_UsuarioWeb DESC";
$querysel = mssql_query($querysel);
$rowid = mssql_fetch_array($querysel);
$id_UsuarioWeb = $rowid['id_UsuarioWeb'];

```

// INSERTAMOS EN LA TABLA DE PADRES

```

$st_pwd0 = md5($id_UsuarioWeb);
$st_pwd = substr($st_pwd0, 0, 7);

```

```

$id_PadreTutor = $valorpapa;

```

// PARA CUANDO EL PADRE AUN NO SE ENCUENTRA REGISTRADO EN LA BD

```

if($valorp == 9999) {
$querypapas = "INSERT INTO tbl_PadresTutores
(st_nomPadreTutor, st_apPaterno, st_apMaterno, st_user, st_pwd, st_mail, st_telefono)
VALUES
('.$st_nomPadreTutor.', ".$st_apPaterno.", ".$st_apMaterno.", ".$st_user.", ".$st_pwd.", ".$st_mail.", ".$st_telef
ono.");

```

```

$querypapas = mssql_query($querypapas);
// obtenermos el id del padre
$rowpapa = "SELECT TOP 1 id_PadreTutor FROM          tbl_PadresTutores WHERE          (st_pwd =
".$st_pwd.") ORDER BY id_PadreTutor DESC";
$rowpapa = mssql_query($rowpapa);
$rowidpap = mssql_fetch_array($rowpapa);
$id_PadreTutor = $rowidpap['id_PadreTutor'];
}

```

```

//INSERTAMOS EN LA TABLA DE RELACIONES
$queryrelacion = "INSERT INTO cat_UsuariosWeb_PadresTutores (id_UsuarioWeb, id_PadreTutor)
VALUES (".$id_UsuarioWeb.", ".$id_PadreTutor.")";
$queryrelacion = mssql_query($queryrelacion);

```

?>

PESTAÑA DE CREDENCIALES

Reasignación de credenciales.

```

<?
if($_POST['control']==1)
{
require("../db.php");

$queryupdate= "UPDATE          tbl_UsuariosWeb
                SET            st_Tarjeta = '".$_POST['sensor']."'
                WHERE          (st_Matricula = '".$_POST['st_Matricula'].")";

$queryupdate=mssql_query($queryupdate);

$queryupdates= "exec BDAjustes";
$queryupdates=mssql_query($queryupdates);

        if($queryupdate) {
?>
        <font color="#FF0000">Reasignación satisfactoria<br /><b>Nuevo sensor
        <?=$_POST['sensor']?>
                </b></font><br>Asignar nueva credencial<br><br><br>
<?
        }
}
?>

```

Baja de credenciales.

```

<?
if($_POST['control']==1){
require("../db.php");

$queryupdate= "UPDATE          tbl_UsuariosWeb
                SET            st_Tarjeta = NULL
                WHERE          (st_Matricula = '".$_POST['st_Matricula'].")";
$queryupdate=mssql_query($queryupdate);

        if($queryupdate)

```

```

?>
{
<font color="#FF0000">Reasignación satisfactoria<br /><b>Baja sensor
<?=$_POST['st_Matricula']?></b></font><br>Baja nueva credencial<br><br><br>
<?
}
}
?>

```

Asistencias Manuales

```

<?
if($_POST['control']==1)
{
require("../db.php");

$skkey=md5(date("Y-m-da"));

$queryborrar = "DELETE FROM tmp_test
WHERE (dt_time = CONVERT(DATETIME, ".date("Y-m-d")." 00:00:00', 102)) AND (st_tar =
"$_POST['tarjeta'].") AND (i_asistio = 0)";
$queryborrar=mssql_query($queryborrar);

$queryfalla= "INSERT INTO dbo.tmp_test (st_tar, dt_hour,dt_time, i_asistio,keyk,dt_charfecha)
VALUES ("$_POST['tarjeta'].",".date("m/d/Y")."."1,'x".$skkey."",".date("d-m-Y").")";
$queryfalla= mssql_query($queryfalla);

if($queryfalla)
{
?>
<font color="#FF0000">Asistencia satisfactoria<br /></font><br>
Asignar nueva asistencia manual <br><br><br>
<?
}
}
?>

```

PESTAÑA DE REPORTES

ALUMNOS INSCRITOS:

```

<table border='0' cellpadding='5' cellspacing='1' class='secTableBG'>
<tr>
<td width="17" class="secTableTitle">&nbsp;</td>
<td width="283" class="secTableTitle">Nombre </td>
<td width="74" class="secTableTitle">Grado y Grupo </td>
<td width="74" class="secTableTitle">Turno</td>
<td width="74" class="secTableTitle">Matricula</td>
<td width="74" class="secTableTitle">Sensor</td>
<td width="74" class="secTableTitle">Celular</td>
</tr>
<?
require("../db.php");

```



```

$query= "SELECT          st_Tarjeta AS sensor, st_Matricula, i_grado, st_grupo, st_Nombre,
                    st_ApellidoPaterno, st_ApellidoMaterno, st_CodigoPostal
FROM      tbl_UsuariosWeb
WHERE     (i_Turno = ".$_POST['turno'].") ORDER BY i_Turno, i_grado, st_grupo ";
$i=1;

$query=mssql_query($query);
while($row=mssql_fetch_array($query))
    {
        if($_POST['turno']==1) $turno = "Matutino";
        else $turno = "Vespertino";

    }

    ?>

    <tr class='conceptCellBG'>
    <td width="17">
    <?=$i++?>
    </td>

    <td><a href="detalleinasistencias.php" alt="detalle">
    <?=$row["st_Nombre"]?>
    <?=$row["st_ApellidoPaterno"]?>
    <?=$row["st_ApellidoMaterno"]?></a></td>

    <td><?=$row["i_grado"]. "-" . $row["st_grupo"]?></td>
    <td><?=$turno?></td>
    <td><?=$row["st_Matricula"]?></td>
    <td><?=$row["sensor"]?></td>
    <td><?=$row["st_CodigoPostal"]?></td>

    </tr>
    <?
    }
    ?>

</table>

```

REPORTE DE INASISTENCIAS POR CANTIDAD

```

<table border='0' cellpadding='5' cellspacing='1' class='secTableBG'>
<tr>
<td width="150" class="secTableTitle">Nombre </td>
<td width="74" class="secTableTitle">Grado y Grupo </td>
<td width="74" class="secTableTitle">Turno</td>
</tr>

<?
require("../db.php");

$query= "SELECT      *
FROM      nofaltas
WHERE     (faltas = ".$_POST['faltas'].") AND (i_Turno = ".$_POST['turno'].") ";

//(tbl_UsuariosWeb.i_Turno = ".$filter.")
$query=mssql_query($query);
while($row=mssql_fetch_array($query)){
if($row["i_Turno"]==1) $turno = "Matutino";
else $turno = "Vespertino";

```

```

?>

<tr class='conceptCellBG'>
<td><a href="detalleinasistencias.php" alt="detalle">
<?=$row["st_Nombre"]?></a>
</td>
<td>
<?=$row["i_grado"]. "-" . $row["st_grupo"]?>
</td>
<td><?=$turno?></td>
</tr>
<?
    }
?>
</table>

```

PESTAÑA DE USUARIOS

```

<?
$rs_result = mssql_query ($st_Query);
$st_class="conceptCellBG";
$i_TotalCheck = 0;

while ($rc_row = mssql_fetch_object($rs_result))
// Iniciamos la lectura de registros que traemos de la funcion fn_PaginarUsuarios
{

//Inicializamos la variable que contendrá los permisos asignados al usuario en curso.
$st_PermisosUsuario = "";
//Formamos el Query que se ejecutará para traer los permisos del usuario.
$st_Query1 = " SELECT sysSection ";
$st_Query1 .= " FROM app_SysSectionsUsers ";
$st_Query1 .= " INNER JOIN app_SysSections ON app_SysSectionsUsers.id_SysSection =
app_SysSections.id_SysSection ";

$st_Query1 .= " WHERE id_SysUser= " . $rc_row->id_SysUser;
//Ejecutamos el query.
$rs_resultado = mssql_query ($st_Query1);
while ($rc_registro = mssql_fetch_object($rs_resultado))
{
//Vamos concatenando los permisos en la variable $st_PermisosUsuario
$st_PermisosUsuario .= $rc_registro->sysSection . "|";
}

// PERMISOS
<?
if (Permisos($sectionUserAdmin))
{
//Si tiene permisos para el módulo de usuarios entonces mostramos la liga para que eliminar los
usuarios seleccionados.
?>
<tr>
<td colspan="4" class="answerCellBG" align="right" ><a href="javascript:confirma(this.form,
<?=$i_TotalCheck ?>)">Borrar seleccionados</a></td>
</tr>
<?
}
?>

```

5.1.3 ESPECIFICACIÓN DE CADA LENGUAJE DE PROGRAMACIÓN

DEFINICIONES.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting).

JAVA es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, como la manipulación directa de punteros o memoria.

JAVASCRIPT es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. JavaScript es un lenguaje orientado a objetos. Todos los navegadores web modernos interpretan el código JavaScript integrado dentro de las páginas web.

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o **RIA** (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

UTILIZACIÓN DE PHP

Este lenguaje de programación fue utilizado en todas las pantallas y aplicaciones desarrolladas para el presente proyecto. Dentro de las funcionalidades principales y más comunes:

1. Conexión a Base de Datos
2. Consultas a la Base de Datos.
3. Generación de contenido dinámico en las páginas web.

Utilización De Java

Java es un lenguaje de programación que permite interactuar directamente con el hardware del equipo de cómputo en el que se encuentra instalado, es por ello que se ha utilizado para poder hacer la conectividad vía USB con un celular y ser a través de esta interacción como sean enviados los mensajes de texto a los celulares de los padres de familia que contraten el servicio en la institución que será implementado el Sistema de Cómputo en Línea para el Control de Asistencia.

En el presente proyecto, la interacción entre JAVA y PHP se lleva a cabo a través de la función en PHP llamada "**exec**" la cual se encarga de ejecutar un programa externo. A continuación se documenta la descripción de esta función:

```
string exec ( string $comando [, array &$salida [, int &$var_retorno ] ] )
```

Lista De Parámetros

comando

El comando que será ejecutado.

salida

Si el argumento *salida* está presente, entonces la matriz especificada será llenada con cada línea de la salida del comando. El espacio en blanco extra, como `\n`, no es incluido en esta matriz. Si la matriz ya contiene algunos elementos, **exec()** anexará sus resultados al final de la matriz.

var_retorno

Si el argumento *var_retorno* está presente junto con el argumento *salida*, entonces el estatus de retorno del comando ejecutado será escrito en esta variable.

Utilización De Javascript

Para el desarrollo del sistema de cómputo en línea para el control de asistencias se ha utilizado Javascript para poder validar los datos obligatorios a capturar o seleccionar en los diferentes formularios (FORM) que el sistema tiene. A continuación se documenta el ejemplo de una de las validaciones desarrolladas para las validaciones:

```
<script type="text/JavaScript">
```

```
function Valida()
{
    if(document.Guion.nombre.value=="")
    {
        alert("Escriba el nombre del alumno");
        document.Guion.nombre.focus();
    }
    return false;
}
if(document.Guion.aPat.value==0)
{
    alert("Escriba al apellido paterno del alumno");
    document.Guion.aPat.focus();
}
return false;
}
if(document.Guion.aMat.value==0)
{
    alert("Escriba el apellido materno del alumno");
    document.Guion.aMat.focus();
}
return false;
}
if(document.Guion.date2562.value==0)
{
    alert("Introduzca la fecha de nacimiento del alumno");
    document.Guion.date2562.focus();
}
return false;
}
if(document.Guion.matricula.value==0)
{
    alert("Escriba la matricula del alumno");
    document.Guion.matricula.focus();
}
return false;
}
if(document.Guion.turno.value==0)
{
    alert("Elija el turno del alumno");
    document.Guion.turno.focus();
}
return false;
}
```

```

if(document.Guion.valorP.value==9999)
{
    if (document.Guion.nombrepapa.value=="")
    {
        alert("Ingrese el nombre del Padre/Tutor");
        document.Guion.nombrepapa.focus();
    }
    return false;
}

if (document.Guion.apaternopa.value==0)
{
    alert("Ingrese el apellido paterno del Padre/Tutor");
    document.Guion.apaternopa.focus();
}
return false;
}

else
{
    if (confirm("¿Está seguro que desea continuar?"))
    {
        document.Guion.submit();
    }
    else
    {
        return false;
    }
}
}

</script>

```

Utilización De Ajax

Gracias a las características de esta técnica de programación la interfaz desarrollada para el Sistema de Cómputo en Línea para el Control de Asistencias brinda una navegación más amigable para los usuarios, además de que parte de los beneficios que desde inicio esta técnica proporciona es la agilización en el acceso a la información sin interrumpir la navegación del usuario en el sistema, dejando todas las tareas del lado del servidor.

A continuación se presenta una línea de código que manda llamar la función ChangeAjax() para llevar a cabo una petición.

```

<input type="text" name="datosPadre" id="datosPadre" size="50"
onKeyUp="changeAjax('ajax/usuarioWebAltaBuscadorStep1.php','datosPadre','div_padremadre');" >

```

Los parámetros que la función ChangeAjax recibe son:

1. La URL del archivo que se ejecutará
2. El nombre del componente que al cambiar de estado mandará llamar la función
3. El nombre del componente que recibirá los datos devueltos por el archivo ajax invocado.

```

<script type="text/javascript">

```

```

var peticion = false;
var testPasado = false;
try {

```

```

    petición = new XMLHttpRequest();
  } catch (trymicrosoft) {
  try {
    petición = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (othermicrosoft) {
  try {
    petición = new ActiveXObject("Microsoft.XMLHTTP");
  } catch (failed) {
    petición = false;
  }
  }
  }
  if (!petición)
  alert("ERROR AL INICIALIZAR!");

function changeAjax (url, comboAnterior, element_id) {
  var element = document.getElementById(element_id);
  var valordepende = document.getElementById(comboAnterior)
  var x = valordepende.value
  if(url.indexOf('?') != -1) {
    var fragment_url = url+'&Id='+x;
  }else{
    var fragment_url = url+'?Id='+x;
  }
  element.innerHTML = 'Cargando...<!---->';
  petición.open("GET", fragment_url);
  petición.onreadystatechange = function() {
  if (petición.readyState == 4) {
    element.innerHTML = petición.responseText;
  }
  }
  petición.send(null);
}
</script>

```

5.1.4 Sesiones y Conectividad con la Base De Datos

A continuación se documenta el archivo que se encarga de validar los datos ingresados por los usuarios. A través de este archivo se puede saber si el usuario existe o no en la base de datos y una vez que se ha confirmado su existencia, se verifica a que secciones tiene acceso.

```
<?php
```

```

require("db.php");
$username=$_POST["username"];
$password=$HTTP_POST_VARS["password"];

if (($username <> "") and ($password <> ")){

// Se genera la consulta para saber si el usuario se encuentra almacenado en la BD
$query = "SELECT id_SysUser,id_Perfil FROM app_SysUsers WHERE username=" . $username . "
AND password=" . $password . " ";

$result = mssql_query ($query);
$row = mssql_fetch_object($result);
$id_SysUser=$row->id_SysUser;
$id_Perfil=$row->id_Perfil;

```

```

$cuantos = mssql_num_rows ($result);

if (mssql_num_rows ($result) > 0){
    $query="SELECT id_SysSection FROM app_SysSectionsUsers WHERE id_SysUser='$id_SysUser'
ORDER BY id_SysSection;";
    $result = mssql_query ($query);
    $strPermisos="";

    while ($row = mssql_fetch_object($result)){
        $strPermisos.=$row->id_SysSection . ",";
    }
    $strPermisos = substr($strPermisos, 0, strlen($strPermisos)-1);

session_start(); // SE INICIA LA SESIÓN

$_SESSION["id_SysUser"] = $id_SysUser; // Se asigna a la variable de sesión el id almacenado en la BD
$_SESSION["strPermisos"] = $strPermisos;
$_SESSION["id_Perfil"] = $id_Perfil; //Perfil correspondiente al usuario que ingresará al sistema

$st_Query= "INSERT INTO app_SysAccessLogs (id_SysUser, sysSection, IPaddress, referrer,
browser_type) VALUES ($id_SysUser, 'INIT', "._SERVER['REMOTE_ADDR'].",
"._SERVER['HTTP_REFERER'].", "._SERVER['HTTP_USER_AGENT'].");";

$rs_Resultado= mssql_query($st_Query);

        echo "<meta http-equiv='REFRESH' content='0;url=inicio.php'>";
    }
    else {
        echo "<meta http-equiv='REFRESH' content='0;url=index2.php'>";
    }
}
else {
    echo "<meta http-equiv='REFRESH' content='0;url=index2.php'>";
}
?>

```

Conexión A La Base De Datos A Través De PHP

El siguiente código mostrado corresponde al archivo a través del cual el sistema se conecta a la Base de Datos. La función de conexión devuelve una variable de enlace MS SQL (que corresponde al manejador de base de datos "Microsoft SQL Server" utilizado para el desarrollo de este proyecto) de tener éxito, o **FALSE** si ocurre un error.

```

<?php
error_reporting (0);

$_dbHost = "localhost";
$_dbUser = "UsuarioBD";
$_dbPasswd = "PasswordBD";
$_dbName = "BDnombre";

$link = mssql_connect($_dbHost, $_dbUser, $_dbPasswd) or die ("No se pudo conectar a la BD");
mssql_select_db ($_dbName) or die ("No se pudo seleccionar la BD");

?>

```

CAPÍTULO 6

MANUAL DE

USUARIO

CAPÍTULO 6 MANUAL DE USUARIO

En el capítulo anterior se documentó todo lo referente a la administración del sistema de cómputo en línea para el control de asistencias, en el presente capítulo se documentará la sección que corresponde a las consultas que los padres de familia pueden hacer respecto de toda la información referente a las asistencias de sus hijos alumnos inscritos.

6.1. SERVICIO A USUARIOS

Se desarrolló un portal web con acceso desde cualquier equipo de cómputo con conexión a internet, en el cual los padres de familia podrán consultar toda la información almacenada en la base de datos del sistema, referente a las asistencias de sus hijos. Pudiendo generar un reporte por cada hijo que tenga registrado, así como elegir un rango de fechas específico que le permita observar el histórico de asistencias/inasistencias de sus hijos.

6.1.1 PORTAL PARA PADRES DE FAMILIA

Para poder tener acceso al portal web, el padre de familia deberá contar con un usuario y una contraseña proporcionados por el administrador del sistema. La contraseña será almacenada por seguridad encriptada. EN la figura 6.1 se muestra la pantalla inicial del portal.



FIG. 6.1 Pantalla inicial de acceso al portal web.

Una vez que se ingresan los datos correctos solicitados, se mostrará una pantalla de inicio de sesión como la mostrada en la figura 6.2.1:



FIG. 6.2.1 Pantalla inicial del portal web. Presentación para cuando se tiene un único alumno inscrito registrado.

En el ejemplo de la figura 6.2.1, se muestra el caso en que sólo sea un alumno el que se tiene registrado para el padre/tutor que ingresó el sistema. En el caso en que exista más de un alumno registrado para el padre/tutor se muestra una tabla que contiene un listado con todos los alumnos registrados en el sistema bajo su responsabilidad, la interfaz mostrada corresponde a la figura 6.2.2.



FIG. 6.2.2 Pantalla inicial del portal web. Presentación para cuando se tiene más de un alumno inscrito registrado.

El padre/tutor deberá seleccionar el alumno que desea ver en el portal web y dar clic en el botón consultar. A continuación el sistema mostrará los datos del alumno previamente seleccionado y el padre/tutor podrá seleccionar el rango de fechas en que desea consultar las asistencias del alumno (ver figura 6.3).



FIG. 6.3 Portal web. Pantalla de selección de fechas para generación de reporte de inasistencias.

Una vez seleccionadas las fechas entre las cuales se desea realizar la consulta, se deberá dar clic en el botón “Consultar” y el sistema a continuación generará automáticamente el reporte correspondiente. Ejemplo, figura 6.3.

DERIANNT MANUEL ALPIZAR VILLAR :: Reporte de asistencia	
Fecha Inicio: 1/2/2009 - Fecha Fin: 09/03/2009	
Fecha	Hora de Entrada
20 Feb 2009	1:05PM
22 Feb 2009	8:44AM
23 Feb 2009	8:44AM
24 Feb 2009	1:05PM
25 Feb 2009	8:44AM
26 Feb 2009	1:05PM
27 Feb 2009	8:44AM
28 Feb 2009	1:05PM
1 Mar 2009	6:57AM
2 Mar 2009	6:53AM
3 Mar 2009	6:53AM
4 Mar 2009	1:07PM
5 Mar 2009	6:57AM
6 Mar 2009	6:51AM
9 Mar 2009	6:57AM

[Regresar](#)

Facultad de Ingeniería
UNAM
México 2009

FIG. 6.3 Portal web. Resultado de reporte de inasistencias para un intervalo específico de fechas.

Para cuando el usuario trate de hacer una consulta sin haber ingresado previamente la fecha de inicio del reporte y/o la fecha final del reporte, el sistema enviará un mensaje de alerta notificando la situación, tal como se muestra en la figura 6.4.



FIG. 6.4 Portal web. Validación de selección de ambas fechas.

6.1.2 Pantallas Con Código Y Explicación De Las Mismas

Pantalla De Inicio Y Validación De Acceso

El siguiente código mostrado se encarga por un lado desde un archivo enviar al servidor las variables de acceso para que un segundo archivo realice la validación necesaria.

INICIO

```

<form action="login.php" method="post" name="loginform" onSubmit="return CheckRequiredFields()">

  <table border="0" cellspacing="1" cellpadding="3" align="center" class="tableStyle">
    <tr class="tableTitle">
      <td colspan="2" bgcolor='#0099CC'>Acceso</td>
    </tr>

    <tr>
      <td class="tableHeader">Usuario:</td>
      <td class="rowOdd"><input name="username" type="text" size="25" /></td>
    </tr>
    <tr>
      <td class="tableHeader">Contrase&ntilde;a:</td>
      <td class="rowOdd"><input name="password" type="password" size="25" /></td>
    </tr>

    <tr>
      <td class="rowOdd" colspan="2" align="right">
        <input name="Ingresar" type="submit" value="Ingresar" />
      </td>
    </tr>
  </table>
</form>

```

VALIDACIÓN

```
<?
include ("../admin/db.php");

$user= $_POST['username'];
$password = $_POST['password'];

$sql="SELECT * FROM tbl_PadresTutores WHERE st_user='$user' AND st_pwd='".md5($password)."'";

$res = mssql_query($sql);

if (mssql_num_rows($res)==0)
{
    // No existe el usuario en la BD, se redirecciona al inicio enviando un mensaje
    header("location:index.php?msg=1");
}
else
{
    session_start();
    while ($row=mssql_fetch_object($res))
    {
        $_SESSION['ID_TUTOR']=$row->id_PadreTutor;
    }

    $sql ="INSERT INTO tbl_LoginLogs (id_PadreTutor,st_RemoteServer) VALUES (";
    $sql.= $_SESSION['ID_TUTOR'];
    $sql.=" , ";
    $sql.= $_SERVER['REMOTE_ADDR'];
    $sql.=")";
    $res=mssql_query($sql);

    $sql="SELECT TOP 1 * FROM tbl_LoginLogs ORDER BY id_LoginLog DESC";
    $res=mssql_query($sql);

    while ($row=mssql_fetch_object($res))
    {
        // Se inserta la acción ejecutada en la table de Logs
        $sevent ="INSERT INTO tbl_EventosUsuariosWeb (id_TipoEvento, id_UsuarioWeb, st_NombreEvento,
id_Evento) VALUES (4,";
        $sevent.= $_SESSION['ID_TUTOR'];
        $sevent.= " , 'Login web', ";
        $sevent.= $row->id_LoginLog;
        $sevent.= ")";

        $resEvent=mssql_query($sevent);
    }
    //Acceso exitoso, se redirecciona al portal
    header("location: begin.php");
}
?>
```

SESIÓN INICIADA EN EL PORTAL WEB

```
<?
session_start();

include ("../admin/db.php");

//Se valida que la session del usuario aun no haya expirado, de lo contrario redirecciona
if (!(ISSET($_SESSION['ID_TUTOR'])))
{
    header("location: index.php?msg=2");
}

else
{
    //Se traen los datos almacenados en la BD del padre que acaba de ingresar al sistema
    $sql="SELECT * FROM tbl_PadresTutores WHERE id_PadreTutor = ".$_SESSION['ID_TUTOR'];
    $res=mssql_query($sql);
    while ($row=mssql_fetch_object($res))
    {
        $nomTutor = $row->st_nomPadreTutor." ".$row->st_apPaterno." ".$row->st_apMaterno;

    }

    $id_UsuarioWeb = "";

    //Se buscan alumnos que estén registrados en el sistema bajo la responsabilidad del padre que ingresó
    $sql = "SELECT * FROM cat_UsuariosWeb_PadresTutores WHERE id_PadreTutor= ";
    $sql.= $_SESSION['ID_TUTOR'];

    $res=mssql_query($sql);

    if (mssql_num_rows($res)==0)
    {
        //No hay alumnos inscritos en el sistema asignados al padre
        $flag=0;
    }

    elseif (mssql_num_rows($res)==1)
    {
        //Existe solo un alumno almacenando en la BD
        $flag=1;
        while ($row=mssql_fetch_object($res))
        {
            $id_UsuarioWeb = $row->id_UsuarioWeb;
        }

        // Se traen de la BD los datos personales del alumno
        $query = "SELECT * FROM tbl_UsuariosWeb WHERE id_UsuarioWeb = ".$id_UsuarioWeb;
        $rquery = mssql_query($query);
        while($rowUser=mssql_fetch_object($rquery))
        {
            $nomAlumno= $rowUser->st_Nombre." ".$rowUser->st_ApellidoPaterno." ".$rowUser->st_ApellidoMaterno;
            $tarjeta= $rowUser->st_Tarjeta;
        }
    }
}
```

```

//Se consulta la asistencia / inasistencia del alumno para el día actual
$query="SELECT * FROM tmp_test WHERE st_tar='$tarjeta' AND dt_time =".date('d/m/Y')."";
$rquery = mssql_query($query);
if (mssql_num_rows($rquery) >0)
{
    while ($rowUser=mssql_fetch_object($rquery))
    {
        $hora= substr($rowUser->dt_hour, 11,13);
    }
}
else
{
    $hora ="Sin Registro";
}
}

elseif (mssql_num_rows($res)>=1)
{
    $flag=2;
    // Existe más de un alumno en la BD registrado bajo la responsabilidad del padre que ingresó
    while ($row=mssql_fetch_object($res))
    {
        $id_UsuarioWeb .= $row->id_UsuarioWeb.", ";
    }
}
}
?>

```

Se muestra un formulario que solicita las fechas entre las cuales se desea obtener un reporte de asistencias / inasistencias del alumno

```

<form action="reporte.php" method="post" name="Guion" onSubmit="return CheckRequiredFields()">
  <input type="hidden" name="tarjeta" value="<?=$tarjeta?>">
  <input type="hidden" name="nomAlumno" value="<?=$nomAlumno?>">
  <table border="0" cellspacing="1" cellpadding="3" align="center" class="tableStyle">
    <tr class="tableTitle">
      <td colspan="2" bgcolor='#0099CC'>Reporte por Fechas</td>
    </tr>
    <tr>
      <td class="tableHeader">Seleccione fecha de inicio</td>
      <td class="rowOdd"><input readonly="readonly" name="date2562" value="" size="12"
type="text">&nbsp;  <a href="javascript:;"
onClick="cal2562.select(Guion.date2562,'anchor2562','dd/MM/yyyy'); return false;" name="anchor2562"
id="anchor2562"></a>
      <div id="testdiv2562" style="visibility: hidden; background-color: white; position: absolute;"></div></td>
    </tr>
    <tr>
      <td class="tableHeader">Seleccione fecha de fin</td>
      <td class="rowOdd"><input readonly="readonly" name="date2563" value="" size="12"

```

```

type="text">&nbsp;<a href="javascript:
onClick="cal2563.select(Guion.date2563,'anchor2563','dd/MM/yyyy'); return false;" name="anchor2563"
id="anchor2563"></a>
<div id="testdiv2563" style="visibility: hidden; background-color:
white; position: absolute;"></div></td>
</tr>
<tr>
<td class="rowOdd" colspan="2" align="right">
<input name="ingresar" type="submit" value="Consultar" /> </td>
</tr>
</table>
<input name="id_Tutor" type="hidden" value="<?=$_SESSION['ID_TUTOR']?>" />
<input name="id_UsuarioWeb" type="hidden" value="<?=substr($id_UsuarioWeb,1)?>" />
</form>

```

Si existe más de un alumno registrado para el padre / tutor que ingresó al sistema, se despliega una lista para seleccionar al alumno que desea ser consultado

```

<form action="reporteUsuario.php" method="post" name="loginform">
<table border="0" cellspacing="1" cellpadding="3" align="center" class="tableStyle">
<tr class="tableTitle">
<td colspan="2" bgcolor='#0099CC'>Existe m&acute;s de un alumno registrado</td>
</tr>
<tr>
<td class="tableHeader">Seleccione el alumno y de clic en consultar</td>
<td class="rowOdd">
<select name="UsuariosWeb">
<?
$sql = "SELECT * FROM tbl_UsuariosWeb WHERE id_UsuarioWeb IN (";
$sql.= substr($id_UsuarioWeb,0,strlen($id_UsuarioWeb)-1);
$sql.= ") ORDER BY st_Nombre";

$res = mssql_query($sql);
while ($row = mssql_fetch_object($res))
{
echo "<option value="";
echo $row->id_UsuarioWeb;
echo ">";
echo $row->st_Nombre;
echo " ";
echo $row->st_ApellidoPaterno;
echo " ";
echo $row->st_ApellidoMaterno;
echo "</option>";
}
?>
</select></td>
</tr>
<td class="rowOdd" colspan="2" align="right">
<input name="ingresar" type="submit" value="Consultar" /> </td></tr>

```



```

</table>
<input name="id_Tutor" type="hidden" value="<?=$_SESSION['ID_TUTOR']?>" />
</form>

```

GENERACIÓN DEL REPORTE DE ASISTENCIAS / INASISTENCIAS

```

<?
// Se inician las variables de sesión y se incluye el archivo de conexión a la BD
include ("../admin/db.php");
session_start();

// Se obtienen las variables enviadas por el formulario en el archivo anterior
$fe_inicio = $_POST['date2562'];
$fe_fin = $_POST['date2563'];
$tarj= $_POST['tarjeta'];
$nomAlumno = $_POST['nomAlumno'];

// Se genera una variable que almacena la condición para la consulta en la BD
// que devolverá los datos solicitados en el reporte

$rangofechas = "AND dt_time between (convert(datetime, '$fe_inicio.' 00:00:00',103)) and
(convert(datetime, '$fe_fin.' 23:59:59',103))";

// Se genera y ejecuta la consulta a la BD
$sql = "SELECT * FROM tmp_test WHERE st_tar='$tarj' ". $rangofechas;
$res = mssql_query($sql);
$fondo= "fondoGris";
?>

```

Para el caso en que no haya ni un solo registro en la BD del alumno consultado se despliega el siguiente código

```

<?
if (mssql_num_rows($res)==0){
echo "<tr><td colspan='2' class='<?=$fondo?>'>Sin registros para las fechas seleccionadas</td></tr>";
}
?>

```

Si fueron encontrados registros de asistencias, se genera y despliega el reporte con el siguiente código

```

<?
while($row=mssql_fetch_object($res))
{
echo "<tr><td class='$fondo'>".substr($row->dt_time,0,11)."</td>";
echo "<td class='$fondo'>".substr($row->dt_hour, 11,13)."</td></tr>";

if ($fondo=="fondoGris"){
    $fondo="fondoBlanco";
}
else
{
    $fondo="fondoGris";
}
}
?>

```

CONCLUSIONES

CONCLUSIONES

Al término del desarrollo del Sistema de Cómputo en Línea para Control de Asistencias, se concluye que los objetivos planteados han sido cubiertos. Los tiempos de acceso a la institución no se han visto alterados, el control de acceso es más organizado, el porcentaje de inasistencias ha disminuido en un 20% en comparación con el año anterior. Inclusive se han generado nuevas expectativas sobre los beneficios que una herramienta web automatizada puede brindar dentro de una institución académica, tal es el caso de la inquietud de las autoridades de la institución de implementar nuevos módulos al sistema donde no sólo se cubra la necesidad del control de asistencias, sino abarcar otros rubros como lo son la cooperativa escolar, la administración de grupos, turnos y horarios. Esto nos deja como muestra lo importante y funcional que resulta ser un sistema que no necesita elevados niveles de presupuesto económico para cumplir con las necesidades del cliente.

Como todos los sistemas desarrollados el presente proyecto es perfectible, así que parte de las mejoras que una vez terminada la implementación hemos detectado se encuentran las notificaciones vía e-mail que podrían estarse enviando a los padres de familia, de manera paralela a los SMS, ya que parte de las novedades y nuevos usos que la telefonía móvil ofrece, se encuentra el manejo de cuentas de correo y en general la navegación por Internet, lo cual ofrecería una opción más para que la información generada por el sistema sea notificada. De igual manera, como parte de las posibles mejoras a implementar al actual sistema, es un lector de código de barras que pueda hacer las mismas funciones de los sensores brindando una segunda opción automatizada de registrar la asistencia de los alumnos.

Se ha dado soporte y capacitación continuamente al sistema desarrollado ya que hay nuevos usuarios administradores que lo requieren. Es importante mencionar que pese a tener ya varios meses de estar funcionando en producción, el presente sistema no ha requerido de mantenimiento correctivo, sin embargo, no se descarta que surjan fallas en un momento dado de la operación, es por eso que continuamente se mantiene el contacto vía correo electrónico con los directivos de la institución donde fue implementado a fin de dar seguimiento al proyecto.

La interacción con usuarios que desconocen muchas de las implicaciones técnicas que el desarrollo de un sistema implica no afectó en el desarrollo del proyecto, ya que parte de la ingeniería por aplicar significa tener la capacidad de reflejar todos esos requerimientos y necesidades en un sistema que los satisfaga al cien por ciento. Muy por el contrario, esta situación brinda una visión más amplia de aplicación y desarrollo, ya que la parte usuaria posee el panorama más amplio del área o rubro en que la ingeniería estará aplicando, no dejando toda la parte creativa a los desarrolladores del sistema.

Hoy por hoy, este sistema se encuentra operando satisfactoriamente en la secundaria "Emiliano Zapata", brindando a sus usuarios tanto administradores como padres de familia información confiable en línea y en tiempo real referente a las asistencias de sus hijos.

GLOSARIO

GLOSARIO

A continuación se describen conceptos técnicos de uso común en la redacción de la presente tesis. Algunos de ellos más utilizados en el campo específico de la programación y el desarrollo de sistemas, y algunos otros de uso cotidiano en la computación en general.

Software

De la Real Academia de la Lengua Española tenemos que software se define como: 1. m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. La palabra software se refiere al equipamiento lógico o soporte lógico de una computadora, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica. Tales componentes lógicos incluyen, entre otros, aplicaciones informáticas tales como algún procesador de textos o un sistema operativo.

Hardware

De la real academia de la lengua española tenemos que hardware se define como: 1. m. Inform. Conjunto de los componentes que integran la parte material de una computadora.

FTP

Siglas en inglés de File Transfer Protocol - Protocolo de Transferencia de Archivos. Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

TCP/IP

Protocolo de Control de Transmisión (TCP – Transmission Control Protocol) y Protocolo de Internet (IP – Internet Protocol). Protocolos que Permiten la transmisión de datos entre redes de equipos computacionales. Existen tantos protocolos en este conjunto, entre ellos se encuentra el popular HTTP (HyperText Transfer Protocol), que es el que se utiliza para acceder a las páginas web, además de otros como el ARP (Address Resolution Protocol) para la resolución de direcciones, el FTP (File Transfer Protocol) para transferencia de archivos, y el SMTP (Simple Mail Transfer Protocol) y el POP (Post Office Protocol) para correo electrónico, TELNET para acceder a equipos remotos, entre otros.

SMTP

Simple Mail Transfer Protocol (SMTP) Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.). Está definido como un estándar oficial de Internet.

SSH

Secure SHell, en español: intérprete de órdenes seguro. Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos. Además de la conexión a otras máquinas, SSH nos permite copiar datos de forma segura (tanto ficheros sueltos, como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro.

TELNET

Telnet (TELEcommunication NETwork) es el nombre de un protocolo de red, que sirve para acceder mediante una red a otra máquina en modo terminal, para manejarla remotamente como si estuviéramos sentados delante de ella. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

FRAME

Técnica de diseño web mediante la cual la pantalla del ordenador queda dividida en ventanas

independientes, se utiliza prioritariamente para presentar barras de navegación que permanecen fijas a lo largo del web. Principalmente son elementos HTML que permiten al navegador separar su ventana en unidades independientes para recibir código HTML de distintos archivos en cada una.

KERBEROS

Es un protocolo de autenticación de redes de computadoras que permite a dos computadoras en una red insegura demostrar su identidad mutuamente de manera segura. Sus diseñadores se concentraron primeramente en un modelo de cliente-servidor, y brinda autenticación mutua: tanto cliente como servidor verifican la identidad uno del otro. Los mensajes de autenticación están protegidos para evitar eavesdropping y ataques de Replay.

LOG

Un log es un registro oficial de eventos durante un periodo de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why, W5) un evento ocurre para un dispositivo en particular o aplicación. La mayoría de los logs son almacenados o desplegados en el formato estándar, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones.

URL

Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

HTTP

El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

TIA 568 B

TIA/EIA-568-B tres estándares que tratan el cableado comercial para productos y servicios de telecomunicaciones. Los tres estándares oficiales: ANSI/TIA/EIA-568-B.1-2001, -B.2-2001 y -B.3-2001. TIA/EIA-568-B intenta definir estándares que permitirán el diseño e implementación de sistemas de cableado estructurado para edificios comerciales y entre edificios en campus. El sustrato de los estándares define los tipos de cables, distancias, conectores, arquitecturas, terminaciones de cables y características de rendimiento, requisitos de instalación de cable y métodos de pruebas de los cables instalados. El estándar principal, el TIA/EIA-568-B.1 define los requisitos generales, mientras que -568-B.2 se centra en componentes de sistemas de cable de pares balanceados y el -568-B.3 aborda componentes de sistemas de cable de fibra óptica.

WEB

World Wide Web (Red Global Mundial), es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

HIPERVÍNCULO

Un hipervínculo (también llamado enlace, vínculo, hiperenlace o link) es un elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento. Combinado con una red de datos y un protocolo de acceso, un hiperenlace permite acceder al recurso referenciado en diferentes formas, como visitarlo con un agente de navegación, mostrarlo o guardarlo localmente.

SCRIPT

El **guion** o **archivo de procesamiento por lotes** (en inglés *script*) es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

BIBLIOGRAFÍA Y MESOGRAFÍA

BIBLIOGRAFÍA

INGENIERÍA DE SOFTWARE, UNA PERSPECTIVA ORIENTADA A OBJETOS

Eric Braude
Alfaomega
México
2003

SOFTWARE ENGINEERING 6th EDITION

Ian Sommerville
Pearson Education Limited
Estados Unidos
2001

OBJECT - ORIENTED SOFTWARE ENGINEERING

Ivar Jacobson, Magnus Christerson, Patrick Jonsson y Gunnar Övergaard
Adisson Wesley
Suecia y Estados Unidos
1995

SOFTWARE QUALITY ENGINEERING – TESTING, QUALITY ASSURANCE, AND QUANTIFIABLE IMPROVEMENT

Jeff Tian
Wiley – Interscience
2005

BEGINNING SQL SERVER 2005 EXPRESS FOR DEVELOPERS: FROM NOVICE TO PROFESSIONAL

Robin Dewson
Apress
Estados Unidos
2007

FUNDAMENTOS DE BASES DE DATOS, 5ta EDICIÓN

Abraham Silberschatz, Henry F. Korth y S. Sudarshas
Mc Graw Hill
Madrid, España
2006

USING SQL

James R. Groff y Paul N. Weinberg
Osborne McGraw – Hill
California, Estados Unidos
1990

DHTML, CREACIÓN DE PÁGINAS WEB DINÁMICAS

Oscar González
Paraninfo, Thomson Learning
Madrid, España
2000

PUBLICAR CON HTML EN INTERNET

Brent Heslop y Larry Budnick
Paraninfo
Madrid, España
1996

HTML HELP WITH DHTML & CSS

Jeannine M. E. Klein
Hewlett Packard, Professional Books
New Jersey, Estados Unidos
2001

JAVASCRIPT PRIMER PLUS

Enhancing Web Pages With The Javascript Programming Language
Gabriel Torok, Jeff Payne y Matt Weisfeld
Waite Group Pr
Estados Unidos
1996

JAVASCRIPT 1.1 ADVANCED JAVASCRIPT TECHNIQUES

Arman Danesh, Wes Danesh
Sams.Net Publishing
Estados Unidos
1996

MICROSOFT WINDOWS 2000 SERVER – Instalación, Configuración Y Administración

José Luis Raya y Elena Raya
Alfaomega Ra-Ma Editorial
Madrid, España
2001

BEGINNING PHP AND ORACLE – FROM NOVICE TO PROFESSIONAL

W. Jason Gilmore y Bob Bryla
Apress
Estados Unidos
2007

MESOGRAFÍA

<http://www.php.net/>

<http://www.php.org/>

<http://www.sun.com>

<http://www.java.com/es/>

<http://www.wikipedia.org/>