UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE RESERVACIÓN PARA VISITAS GUIADAS PARA LA DGDC UNIVERSUM

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :

ELIZABETH LUNA LUZ

DIRECTOR DE TESIS: ING. CARLOS ALBERTO ROMAN ZAMITIZ

CD. UNIVERSITARIA MÉXICO DF Noviembre 2005

Antes que nada quiero agradecerle a Dios por permitirme llegar hasta donde estoy y darme unos padres tan maravillosos . Enseguida quiero agradecerle a mis padres Maura y Francisco por todo el inmenso apoyo, esfuerzo, dedicación y amor incondicional que me han brindado para convertirme en una profesionista, gracias por todo lo que juntos hemos logrado papás, los quiero mucho!!! . También quiero agradecer a mis hermanos, Erica gracias por todo tu apoyo y consejos, y a mi hermanito Paco por apoyarme en todo lo que ha podido, los quiero mucho!!.

Gracias al museo de las ciencias Universum por darme la oportunidad de realizar mi tesis y brindarme su apoyo para el desarrollo de éste proyecto.

Gracias a Carlos por todo su apoyo, sus conocimientos y tiempo , gracias por ser mi director de tesis.

Por otra parte, quiero darle las gracias a todos mis profesores de la facultad de Ingeniería porque gracias a sus conocimientos he llegado a ser una ingeniera.

Por último quiero agradecerle a la máxima casa de estudios la UNAM , a la facultad de Ingeniería, que fue como mi segunda casa y a todas las personas que en ella laboran.

INDICE

PRÓLOGO

En la actualidad, diversas instituciones tienen la necesidad de llevar un control de su información , de tal forma que ésta pueda ser almacenada, procesada, y utilizada para diferentes fines, en algunas instituciones éste proceso aún se lleva acabo de manera manual involucrando el uso de formatos en papel, lo cual implica un gasto excesivo de este tipo de recurso, y aunque ya existen algunos sistemas que evitan el uso del papel, no cumplen con los requisitos indispensables para poder llevar a cabo todo un control de manera adecuada y robusta de su información.

En la presente tesis se llevó a cabo el desarrollo de un sistema computacional capaz de controlar la información aplicada a un sistema de reservaciones para visitas guiadas, el cual ofrece una solución práctica y eficiente del problema anteriormente mencionado.

La tesis consta básicamente de cuatro capítulos, los cuales están estructurados de la siguiente manera:

Introducción.
Este capítulo plantea de manera general como es que surge la idea de la creación del sistema que se desarrolló.

1. **Antecedentes teóricos.**

Éste capítulo inicia con una breve historia del museo de las ciencias Universum , esto para dar a conocer a nuestro usuario del sistema.
Se comienza con un marco conceptual, donde se explicarán temas que fueron necesarios a abordar para el desarrollo de ésta tesis.
Una vez definidos los conceptos a utilizar se prosiguió con el análisis del comportamiento general del sistema.

2. **Análisis de la situación actual, definición del problema y análisis de requerimientos.**

En éste capítulo como su nombre lo indica, se realiza un análisis de la situación actual del museo, se realiza una definición del problema y finalmente se hace un análisis de requerimientos para el desarrollo del sistema.

3. **Análisis del nuevo sistema a desarrollar.**

Éste capítulo se refiere a que una vez analizada la problemática que enfrenta el museo de las ciencias, planteado en el capítulo anterior, se busca la mejor manera para dar solución a dicho problema haciendo uso de los requerimientos analizados en el capítulo anterior.

4. **Diseño y planeación del sistema**

Dado el análisis de los dos últimos capítulos, como todo proceso relativo a la ingeniería de software, se realizó un diseño del sistema agrupando sus funciones en sus tres diferentes módulos desarrollados:

- Instituciones
- Reservaciones
- Usuario

Se prosiguió con los siguientes diagramas :

- Diagrama de Casos de Uso, en donde se destacan las funciones y procesos del sistema y sus actores.
- Diagrama de Actividades, en donde se detallan dichas funciones esenciales, identificando a su vez cada dato y su trayectoria a lo largo del mismo.
- Diagrama Entidad-Relación de la base de datos resultante para albergar toda la información del sistema.

Todo esto con el fin de que la información estuviera estructurada y que la consulta a la misma se llevara en el menor tiempo posible. Una vez hecho esto, se realizó el diccionario de datos, el cual contiene la información de todos los términos empleados.

5. Elección de las herramientas a utilizar , lenguaje de programación, sistema manejador de base de datos

En éste capítulo se hace una comparación entre las posibles herramientas a utilizar en el sistema, para la selección del lenguaje de programación se compara java con c++ y smalltalk, y para el caso del manejador de base de datos se hace una comparación entre Access, SQLServer, MySQL y postgreSQL. Al término de dichas comparaciones se elige como mejor herramienta de trabajo Java y PostgreSQL.

6. Programación

En éste capítulo se da una explicación paso a paso de lo que se desarrollo en dos de las clases de java una utilizada con su método correspondiente para la inserción de registros de instituciones en el sistema y la otra referente a una clase de tipo javabean para el paso valores de atributos necesarios para la primera clase , no se describen las demás clases debido a que todas las clases y métodos siguen la misma estructura. Al término de estas descripciones se muestra el código fuente de los JSP´s correspondientes al formulario de captura de datos y el último que fue utilizado para la llamada de métodos y clases que ejecutan la acción correspondiente.

7. Pruebas de instalación

Este capítulo se refiere a las pruebas de instalación y configuración de software y hardware necesarias para el desarrollo e implantación del sistema , tanto en plataforma Mac OSX como en Linux

Comentario introductorio

Este proyecto surge como necesidad del museo de las ciencias UNIVERSUM debido a las grandes cantidades de información que maneja para el registro de instituciones de las diferentes zonas de la república mexicana y las reservaciones de visitas guiadas que dichas instituciones requieren, dicha información requiere ser almacenada, procesada y manipulada para optimizar tiempos y espacios en el lugar de trabajo del museo, para este caso el área de Atención al Visitante.

Capítulo 1

1. Antecedentes teóricos

En éste capítulo se presentará una breve historia de cómo surge el museo de las ciencias UNIVERSUM, esto para tener una idea más clara de que es y que hace UNIVERSUM quien va a ser nuestro usuario final del sistema.

Es a fines de 1979 cuando en la UNAM se comienza a cristalizar el sueño de tener un museo de las ciencias; un museo donde jóvenes y niños puedan sentir lo que no se puede experimentar sólo con imágenes o con palabras: la posibilidad del contacto práctico y directo con la realidad, con el realismo de las maravillas que las ciencias ofrecen.

En aquel tiempo, la UNAM hacía muy poco para divulgar las ciencias , había que dar a futuros forjadores de México, un museo de ciencias moderno, participativo y de gran calidad.
No paso mucho cuando el museo ya contaba con una centena de colaboradores, entre técnicos, ingenieros, museógrafos y operarios; poseía un pequeño grupo para los servicios y el mantenimiento, y un taller general para la fabricación y el acabado de los más de 200 equipamientos que ya estaban funcionado en exposiciones temporales.

Los años 1990 y 1991 fueron de una febril actividad. Se visitaron museos extranjeros, pero se decidió hacer algo original y congruente con nuestra cultura y mexicanidad, con equipamientos diseñados de acuerdo con nuestras ideas; para ello, el grupo arriba mencionado generó los conceptos de equipamientos que se manejarían en cada sala y se construyeron los prototipos.

Sin embargo, la adaptación resultó mucho más complicada de realizar de lo que se había pensado, debido a que fue necesario cubrir los patios interiores, construir la entrada principal, ampliar los estacionamientos, etcétera; para lograrlo, el Departamento de Obras de la UNAM realizó un excelente trabajo y el Museo de las Ciencias Universum, que para entonces ya tenía nombre, con 11 salas dedicadas a diversas áreas, una más de exposiciones temporales y con espacios donde crecer, estuvo listo para su inauguración el 12 de diciembre de 1992.
En la antigua guardería del Conacyt, que se había adaptado previamente para albergar los gabinetes de ingeniería y diseño, se construyó su auditorio propio, un observatorio astronómico (Astrolab), un teatro-laboratorio (Fisilab) y varias salitas para cursos y talleres, así como una sala de hidroponia; de esta forma quedó constituida la actual “Casita de las Ciencias”.

Y ahora, recientemente se construyó en Universum una bóveda para un mini Planetario-Laserium y últimamente se ha abierto un bello y acogedor Espacio Infantil, así como una sala dedicada a exposiciones temáticas, la cual alberga una magnífica presentación sobre satélites de comunicaciones.
A 10 de años de su apertura, Universum ya recibió a su visitante número seis millones.
En la actualidad se proyecta la creación de un "Centro del Espacio", que contará con un gran planetario y un virtuarium electrónico, lo más moderno en la proyección de películas panorámicas: todo ello apoyado con restaurantes, tiendas y otros servicios que, en conjunto, podrán brindar a la comunidad universitaria, a la población de la ciudad, a nuestros visitantes de provincia las realidades del presente, la tradicional hospitalidad mexicana, así como las extraordinarias imágenes y sensaciones acerca de la ciencia, y de otra índole, que nos tiene reservado el porvenir.

José de la Herrán
septiembre de 2001

1.1 Panorama actual de las visitas guiadas al Universum

Panorama General de la Problemática

La Dirección General de Divulgación de la Ciencia (DGDC) tiene como objetivo divulgar la ciencia a todos niveles y públicos posibles; para ello utiliza diferentes medios para hacerlo, ya sea en revistas, cápsulas de radio, videos y actualmente basándose en la nuevas tecnologías y herramientas, pretende usar su infraestructura de red tanto alámbrica como inalámbrica y sus servidores de páginas web, para que por estos medios hacer streaming de videos y bajo demanda de contenidos científicos para toda la población que esté interesado en ello.
Universum recibe la visita de numerosas escuelas desde nivel preescolar hasta nivel medio superior, tanto escuelas públicas, privadas y las que tienen convenio con la SEP, tanto ubicadas en el Distrito Federal como de los estados de la República Mexicana, además de recibir al público en general. Por tal motivo es uno de los museos más concurridos entre la población estudiantil.
Para brindar un mejor servicio al visitante Universum cuenta con una Área de Atención al Visitante la cual se encarga, de entre otras actividades, de brindar visitas guiadas en las diferentes salas del museo(son 12 Salas y unas Sala de Exposiciones Temporales), proporcionando así al visitante una mejor experiencia dentro del museo. Para ello se llena un formato de manera manual en papeletas, en el cual se obtiene información de la Institución que va a realizar su reservación incluyendo nombre de la Institución, nombre de la persona responsable del grupo de personas a las que se les hará la visita guiada, nombre de la persona que está realizando la reservación; ya sea vía telefónica o personalmente, domicilio de la Institución, colonia a la que pertenece, delegación o municipio, teléfono(s), número de lada en caso de estar fuera del DF. , número de fax, correo electrónico, número de personas a las que se les va a brindar el recorrido, número de profesores que van a asistir a la visita (en caso de no serlo se toma como monitor), número de padres de familia que asistirán a la visita, nivel de estudios de las personas a las que se les hará el recorrido (maternal, preescolar, primaria, secundaria, bachillerato, etc.), así como el grado al que pertenecen(año / semestre), tiempo de permanencia en el museo, tipo de escuela(oficial/particular/con convenio de la SEP), nombre de la(s) sala(s) a las que se les va a dar mayor preferencia para ser visitadas, actividad a realizar(conferencia, taller, etc.), nombre de la persona que atendió, observaciones, nombre de la persona que confirma la visita y fecha de confirmación, clave que se le

asigna a la Institución para conservar su fecha de visita guiada, día en que se elabora la reservación, fecha planeada para la visita y hora a la que se va a realizar el recorrido. Toda esta información, como lo había mencionado anteriormente, se hace de manera manual, es decir se hace en papel, se consulta en un archivero y se inventa una clave de acuerdo al ingenio de la persona que atiende; sé ha dado el caso en que la clave ha sido adivinada por los visitantes, además esto implica mayor tiempo de espera para quien esta realizando la reservación.
El formato que se utiliza en hojas es el siguiente:

DEPARTAMENTO DE ATENCIÓN AL VISITANTE

DIA:_____
FECHA:_____
HORA:_____

REGISTRO DE VISITAS PROGRAMADAS

INSTITUCION:_____

RESPONSABLE:_____

QUIEN RESERVA:_____

DOMICILIO DE LA ESCUELA:_____

COLONIA:_____ DELEGACIÓN O MUNICIPIO:_____

TELS: _____ FAX:_____

CORREO ELECTRONICO:_____

ESTUDIANTES:_____ PROFESORES: _____ PADRES DE FAM:_____

GRUPOS: _____ NIVEL: _____GRADO(S):_____

TIEMPO DE PERMANENCIA:_____

OFICIAL ( ) PARTICULAR ( ) SEP ( )

AVENTURA	**(__|__)**	**ENERGIA**	**(__|__)**	**MOVIMIENTO**	**(__|__)**
BALSA	**(__|__)**	**E. INFANTIL**	**(__|__)**	**QUÍMICA**	**(__|__)**
BIODIVERSIDAD	**(__|__)**	**ESTRUCTURA.M**	**(__|__)**	**REPRODUCCION**	**(__|__)**
CONCIENCIA	**(__|__)**	**INFRAESCTRUC**	**(__|__)**	**SENDA**	**(__|__)**
COSECHANDO	**(__|__)**	**MATEMATICAS**	**(__|__)**	**TEC.SATELITAL**	**(__|__)**
				UNIVERSO	**(__|__)**

ACTIVIDAD:_____

ATENDIÓ:_____FECHA:_____

OBSERVACIONES:_____

CONFIRMADA POR: _____FECHA:_____

CLAVE: [ ]

Universum obsequia cortesías a los profesores que reservan sus visitas guiadas, para ello el Área de Atención al Visitante debe de llevar un control de cuantos profesores van a llegar a las visitas guiadas, lo cual implica consultar todas las papeletas para localizar el número de profesores de la escuela que hace la visita.
La taquilla necesita conocer esta información para saber cuantos boletos van a ser de cortesía y cuantos deben ser vendidos, de ésta manera se lleva un control de las entradas monetarias al museo.

En donde la "clave" es generada de la siguiente manera:

CLAVE	XddmmN

Donde :
X =Letra aleatoria que se le asigno al mes en que se esta haciendo la reservación.
dd = día planeado para la visita
mm = mes planeado para la visita
N = Número de llamada del día.

1.2 Antecedentes Académicos

1.2.1 Ingeniería de Software

Según la definición del IEEE, citada por [Lewis 1994] "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (SE del inglés Software Engineering) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" [Cota 1994].

La práctica de la ingeniería de software tiene por objeto la construcción de grandes y complejos sistemas de una forma rentable.

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" [Jacobson 1998].El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [Jacobson 1998].

1.2.2 Programación Orientada a Objetos

La Programación Orientada a Objetos desde el punto de vista computacional "es un método de implementación en el cuál los programas son organizados como grupos cooperativos de objetos, cada uno de los cuales representa una instancia de alguna clase, y estas clases, todas son miembros de una jerarquía de clases unidas vía relaciones de herencia" [Greiff 1994].

La programación orientada a objetos (POO) es un modelo de programación que utiliza objetos ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización de código.

Los programas orientados a objetos se crean a partir de módulos conocidos como objetos. Estos son los bloques fundamentales que permiten la creación de programas complejos. Incluso hasta los programas pequeños se benefician de haber sido creados a partir de objetos.
Los objetos constituyen uno de los aspectos claves de la POO (Programación Orientada a Objetos). Un objeto consta de ciertos datos agrupados con instrucciones que actúan sobre los mismos datos. Dicha combinación constituye un paquete autónomo ordenado. La selección de datos y acciones no es arbitraria; éstos deben estar relacionados estrechamente para que el objeto sea significativo. Los objetos están constituidos por datos y sus métodos asociados.
La POO también trata sobre el proceso de simular o modelar los problemas del mundo real. Un modelo se crea a partir de objetos que interactúan entre sí [Douglas, 2003].

“Los mecanismos básicos de la programación orientada a objetos son: objetos , mensajes, métodos y clases”[Ceballos, 2000].

1.2.2.1 Definición de objeto

Un objeto es aquello que tiene estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos). La estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancia y objeto son intercambiables [Internet 3].

Un programa orientado a objetos se compone solamente de objetos, entendiendo por objeto una encapsulación genérica de datos y de los métodos para manipularlos. Dicho de otra forma, un objeto es una entidad que tiene unos atributos particulares, las propiedades, y unas formas de operar sobre ellos, los métodos [Ceballos, 2000]

1.2.2.2 Definición de clase

Las clases son declaraciones de objetos, también se podrían definir como abstracciones de objetos. Esto quiere decir que la definición de un objeto es la clase. Cuando programamos un objeto y definimos sus características y funcionalidades en realidad lo que estamos haciendo es programar una clase [Internet 2].
Una clase es un conjunto de objetos que comparten una estructura y comportamiento común.

La diferencia entre un objeto y una clase es que un objeto es una entidad concreta que existe en tiempo y espacio, mientras que una clase representa una abstracción, la "esencia" de un objeto, tal como son. De aquí que un objeto no es una clase, sin embargo, una clase puede ser un objeto [Internet 1].

Una clase es un tipo de objetos definido por el usuario. Una clase equivale a la generalización de un tipo específico de objetos.
Un objeto de determinada clase se crea en el momento en que se define una variable de dicha clase.

Cuando se escribe un programa utilizando un lenguaje orientado a objetos, no se definen objetos verdaderos, se definen clases de objetos, donde una clase se ve como una plantilla para múltiples objetos con características similares [Ceballos, 2000].

Propiedades en clases

Las propiedades o atributos son las características de los objetos. Cuando definimos una propiedad normalmente especificamos su nombre y su tipo. Nos podemos hacer a la idea de que las propiedades son algo así como variables donde almacenamos datos relacionados con los objetos.

Métodos en las clases

Son las funcionalidades asociadas a los objetos. Cuando estamos programando las clases las llamamos métodos. Los métodos son como funciones que están asociadas a un objeto.

1.2.2.3 Características de la Programación Orientada a Objetos: Abstracción, Encapsulamiento , Herencia, Polimorfismo.

Abstracción

Por medio de la abstracción conseguimos no detenernos en los detalles concretos de las cosas que no interesen en cada momento, sino generalizar y centrarse en los aspectos que permitan tener una visión global del problema.

Encapsulamiento

Esta característica permite ver un objeto como una caja negra en la que se ha introducido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna.

La abstracción y la encapsulación están representadas por la clase. La clase es una abstracción, porque en ella se definen las propiedades o atributos de un determinado conjunto de objetos con características comunes, y es una encapsulación porque constituye una caja negra que encierra tanto los datos que almacena cada objeto como los métodos que permiten manipularlos.

Herencia

La herencia permite el acceso automático a la información contenida en otras clases. De esta forma, la reutilización del código está garantizada. Con la herencia todas las clases están clasificadas en una jerarquía estricta. Cada clase tiene su superclase (la clase superior en la jerarquía), y cada clase puede tener una o más subclases (las clases inferiores en la jerarquía).

Las clases que están en la parte inferior en la jerarquía se heredan de las clases que están en la parte superior de la jerarquía.

El término heredar significa que las subclases disponen de todos los métodos y propiedades de su superclase. Este mecanismo proporciona una forma rápida y cómoda de extender la funcionalidad de una clase.

En Java cada clase sólo puede tener una superclase, lo que se denomina herencia simple. En otros lenguajes orientados a objetos, como C++, las clases pueden tener más de una superclase, lo que se conoce como herencia múltiple. En este caso, una clase comparte los métodos y propiedades de varias clases. Esta característica, proporciona un poder enorme a la hora de crear clases, pero complica excesivamente la programación, por lo que es de escasa o nula utilización. Java, intentando facilitar las cosas, soluciona este problema de comportamiento compartido utilizando interfaces.

Una interfaz es una colección de nombres de métodos, sin incluir sus definiciones, que puede ser añadida a cualquier clase para proporcionarla compartimientos adicionales no incluidos en los métodos propios o heredados.

Polimorfismo

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso [Ceballos, 2000 , pág.32].

1.2.3 Bases de Datos

La explosión de nuevas tecnologías que empezó con la introducción del PC y la llegada del Internet en los noventas, le ha brindado a muchas organizaciones nuevas opciones y herramientas que son explotadas con gran intensidad en la actualidad. Una de ellas es la utilización de instrumentos de información en la generación de bases de datos, lo cual ha sido de gran ayuda para las empresas, ya que gracias a las bases de datos se tiene una mejor manipulación y almacenamiento de la información , logrando así un mejor control de los clientes que solicitan los servicios de dichas empresas.

1.2.3.1 Definición

Definición de Base de Datos
Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.

Componentes principales

Datos. Es la unidad mínima de información.

Hardware. El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.

Software. Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.

Usuarios. Existen tres clases de usuarios relacionados con una Base de Datos:

1. El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
2. El usuario final, quien accesa la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
3. El administrador de la Base de Datos (DBA: Data Base Administrator), quien se encarga del control general del Sistema de Base de Datos.

[Internet 4]

Ventajas en el uso de Bases de Datos.

Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

Eliminación de información redundante. Duplicada

Eliminación de información inconsistente. Si el sistema esta desarrollado a través de archivos convencionales, dicha cancelación deberá operarse tanto en el archivo de facturas del Sistema de Control de Cobranza como en el archivo de facturas del Sistema de Comisiones.

Permite compartir información. Varios sistemas o usuarios pueden utilizar una misma entidad.

Permite mantener la integridad en la información. Solo se almacena la información correcta.

Independencia de datos. La independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos

o tener acceso a la base de datos de diferente manera, sin hacer cambios en las aplicaciones o en los programas.

El sistema organizador de Base de Datos (DBMS)

El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un lenguaje de manipulación de datos (DML: Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language).

El lenguaje de definición de datos (DDL) es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos.

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos.

El lenguaje de consulta (SQL) es empleado por el usuario para extraer información de la base de datos. El lenguaje de consulta permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el SELECT, el PROJECT y el JOIN.

La secuencia conceptual de operaciones que ocurren para acceder cierta información que contiene una base de datos es la siguiente:

1 El usuario solicita cierta información contenida en la base de datos.

El DBMS intercepta este requerimiento y lo interpreta.

El DBMS realiza las operaciones necesarias para acceder y/o actualizar la información solicitada.

El administrador de la Base de Datos (DBA)
El DBA es la persona encargada de definir y controlar las bases de datos corporativas, además proporciona asesoría a los usuarios y ejecutivos que la requieran.
[Internet 4]

1.2.3.2 Tipos de bases de datos

Hay cinco tipos principales de bases de datos:

- Simple o plana.
- Red.
- Jerárquica.
- Relacional.
- Orientada a objetos.

Las **bases de datos simples** son similares a las hojas de cálculo, donde los registros se incluyen en listados simples. La flexibilidad de las bases de datos plana es limitada y solo son apropiadas para tareas relativamente sencillas, como el rotulado de correspondencia.

Las **bases de datos jerárquicas** tienen una estructura de árbol, donde cada nivel de registro se desagrega en un conjunto de categorías más pequeña. Este tipo de base de datos también es relativamente limitada, ya que solo contiene ligas simples entre conjuntos de registros en distintos niveles, con restricciones de flexibilidad.

Las **bases de datos en red** contienen ligas múltiples entre conjuntos de información, lo que permite una mayor flexibilidad.

Las **bases de datos relacionales** van más lejos que las de red al permitir que se utilicen las relaciones entre distintos conjuntos de información para generar consultas complejas. Por ejemplo, la tabla "personal" puede ser ligada a la de "puestos" para ofrecer una descripción completa de la posición ocupada por cada miembro de la institución, y la tabla "puestos" puede ser ligada a la de nómina" para mostrar el salario que se le paga a cada persona.

Las **bases de datos orientadas a objetos** tienen atributos similares a las relacionales. Sin embargo, se utilizan estructuras de información más complejas llamadas "objetos". Estas bases de datos son las más flexibles y maleables.

La mayoría de las bases de datos modernas son relacionables, orientadas a objetos o una mezcla de ambos [Internet 5].

Bases de Datos distribuidas

Son las Bases de Datos que no están almacenadas totalmente en un solo lugar físico, (esta segmentada) y se comunican por medio de enlaces de comunicaciones a través de una red de computadoras distribuidas geográficamente[Internet 4].

Los programas de bases de datos que se ofrecen con los programas más modernos para la automatización de las oficinas son muy poderosas y permiten que personal relativamente poco especializado del organismo electoral pueda generar fácilmente bases de datos sencillas. Los usuarios experimentados pueden desarrollar productos muy sofisticados. Por ejemplo, una autoridad electoral puede utilizar programas de bases de datos para:

- Capturar, almacenar y utilizar información sobre el registro de electores.
- Automatizar distintas partes del proceso electoral, como la emisión y recepción de votos postales.
- Capturar y analizar resultados electorales.
- Almacenar y manejar registros del personal.
- Automatizar la correspondencia personalizada a gran escala.
- Control de inventarios.
- Registrar información sobre los candidatos, facilitar la impresión de papeletas y reportes sobre donaciones y gastos electorales.

[Internet 5]

Tendencias futuras

La explotación efectiva de la información dará ventaja competitiva a las organizaciones.
Las bases de datos orientadas a objetos empleadas para diseño y manufactura asistida por computadora CAD/CAM serán utilizados a un mismo nivel que las Bases se Datos relacionales de la actualidad.

Los lenguajes de consulta (SQL) permitirán el uso del lenguaje natural para solicitar información de la Base de Datos, haciendo más rápido y fácil su manejo.

Algunas Bases de Datos
PostgreSQL , MySQL, ORACLE, DBASE, IV, FOXPRO, FOXBASE, PARADOS, ACCES, APPROACH [Cohen]

1.2.4 Redes de Computadoras

Debido al tremendo impacto de las computadoras y las redes de computadoras en la sociedad durante la década pasada, este periodo de la historia se ha denominado “era de la información”. La productividad y el rendimiento de las organizaciones y las personas ha aumentado significativamente con el uso de estas herramientas revolucionarias. Muchas personas utilizan las redes de computadoras prácticamente a diario en sus relaciones personales y profesionales. Esta tendencia se está acelerando a medida que la gente va descubriendo la potencia de las computadoras y de las redes de comunicación para uso profesional y doméstico. Las transacciones diarias de grandes almacenes, bancos, agencias de viaje y otros muchos negocios se basan en las redes de computadoras. La era de la información se basa por igual en las computadoras y en las redes de computadoras.

1.2.4.1 Definición de Red

Conjunto de máquinas conectadas entre sí mediante uno o más vías de transmisión a fin de llevar a cabo la transferencia e intercambio eficiente y confiable de la información entre las computadoras y terminales. La vía de transmisión es a menudo la línea telefónica, debido a su comodidad y a su presencia universal.

Transmisión de datos. Intercambio de datos (unos y ceros) entre dos dispositivos a través de alguna forma de medio de comunicación.

Comunicación. Intercambio de información

Topologías de red

Topologías y objetivos de diseño

Una configuración de red se denomina topología de red. Por tanto, la topología establece la forma (en cuanto a conectividad física) de la red. El término topología se utiliza en geometría para describir la forma de un objeto. El diseñador de una red tiene tres objetivos al establecer la topología de la misma:

- Proporcionar la máxima fiabilidad a la hora de establecer el tráfico (por ejemplo, mediante encaminamientos alternativos).
- Encaminar el tráfico utilizando la vía de costos mínimos entre los ETD trasmisor y receptor(no obstante, a veces no se escoge la vía de costos mínimos porque otros factores, como la fiabilidad, pueden ser más importantes).
- Proporcionar al usuario el rendimiento óptimo y el tiempo de respuesta mínimo.

ETD es un término genérico para designar a la máquina de usuario final, habitualmente una computadora o una terminal .

En la industria, un ETD puede tomar muy diversas formas, como por ejemplo una estación de trabajo para control de tráfico aéreo, un cajero automático de un banco, un dispositivo sensor para medir la pureza del aire, una terminal o computadora con correo electrónico, una computadora personal en casa o en la oficina.

Al hablar de redes, el concepto de fiabilidad hace referencia a la capacidad de enviar los datos correctamente (es decir, sin errores) entre los ETD. Involucra la posibilidad de recuperación de errores o de datos perdidos en la red por motivos de fallos en el canal. La fiabilidad también tiene que ver con el mantenimiento del sistema: pruebas diarias, relevo de componentes defectuosos o en fallo manifiesto, y aislamiento de fallos en caso de aparecer problemas. Si algún componente es causa de problemas, el sistema de diagnóstico de la red debería buscar rápidamente el fallo, encontrarlo y, si es posible, aislar el componente de la red.
[Black, 1995]

Modelos de topología Las principales modelos de topología son:

Topología de bus
La topología de bus tiene todos sus nodos conectados directamente a un enlace y no tiene ninguna otra conexión entre nodos. Físicamente cada host está conectado a un cable común, por lo que se pueden comunicar directamente, aunque la ruptura del cable hace que los hosts queden desconectados.

La topología de bus permite que todos los dispositivos de la red puedan ver todas las señales de todos los demás dispositivos, lo que puede ser ventajoso si desea que todos los dispositivos obtengan esta información. Sin embargo, puede representar una desventaja, ya que es común que se produzcan problemas de tráfico y colisiones, que se pueden paliar segmentando la red en varias partes. Es la topología más común en pequeñas LAN, con hub o switch final en uno de los extremos.

Topología de anillo

Una topología de anillo se compone de un solo anillo cerrado formado por nodos y enlaces, en el que cada nodo está conectado solamente con los dos nodos adyacentes.

Los dispositivos se conectan directamente entre sí por medio de cables en lo que se denomina una cadena margarita. Para que la información pueda circular, cada estación debe transferir la información a la estación adyacente.

Topología de anillo doble Una topología en anillo doble consta de dos anillos concéntricos, donde cada host de la red está conectado a ambos anillos, aunque los dos anillos no están conectados directamente entre sí. Es análoga a la topología de anillo, con la diferencia de que, para incrementar la confiabilidad y flexibilidad de la red, hay un segundo anillo redundante que conecta los mismos dispositivos. La topología de anillo doble actúa como si fueran dos anillos independientes, de los cuales se usa solamente uno por vez.

Topología en estrella La topología en estrella tiene un nodo central desde el que se irradian todos los enlaces hacia los demás nodos. Por el nodo central, generalmente ocupado por un hub, pasa toda la información que circula por la red. .

La ventaja principal es que permite que todos los nodos se comuniquen entre sí de manera conveniente. La desventaja principal es que si el nodo central falla, toda la red se desconecta.

Topología en estrella extendida La topología en estrella extendida es igual a la topología en estrella, con la diferencia de que cada nodo que se conecta con el nodo central también es el centro de otra estrella. Generalmente el nodo central está ocupado por un hub o un switch, y los nodos secundarios por hubs. La ventaja de esto es que el cableado es más corto y limita la cantidad de dispositivos que se deben interconectar con cualquier nodo central. La topología en estrella extendida es sumamente jerárquica, y busca que la información se mantenga local. Esta es la forma de conexión utilizada actualmente por el sistema telefónico.

Topología en árbol La topología en árbol es similar a la topología en estrella extendida, salvo en que no tiene un nodo central. En cambio, un nodo de enlace troncal, generalmente ocupado por un hub o switch, desde el que se ramifican los demás nodos.

El enlace troncal es un cable con varias capas de ramificaciones, y el flujo de información es jerárquico. Conectado en el otro extremo al enlace troncal generalmente se encuentra un host servidor.

Topología en malla completa En una topología de malla completa, cada nodo se enlaza directamente con los demás nodos. Las ventajas son que, como cada todo se conecta físicamente a los demás, creando una conexión redundante, si algún enlace deja de funcionar la información puede circular a través de cualquier cantidad de enlaces hasta llegar a destino. Además, esta topología permite que la información circule por varias rutas a través de la red.

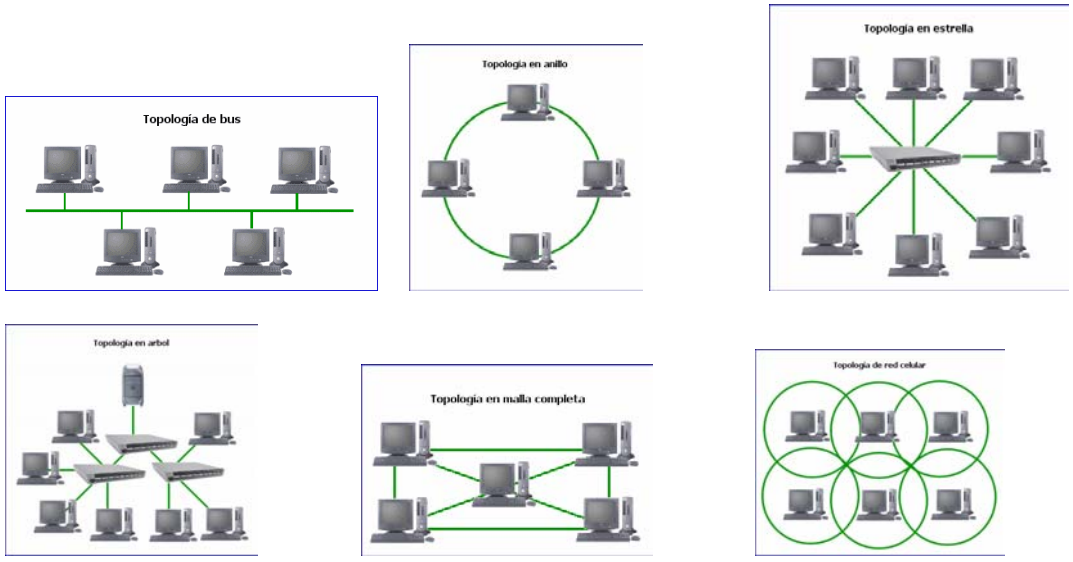
La desventaja física principal es que sólo funciona con una pequeña cantidad de nodos, ya que de lo contrario la cantidad de medios necesarios para los enlaces, y la cantidad de conexiones con los enlaces se torna abrumadora.

Topología de red celular La topología celular está compuesta por áreas circulares o hexagonales, cada una de las cuales tiene un nodo individual en el centro.

La topología celular es un área geográfica dividida en regiones (celdas) para los fines de la tecnología inalámbrica. En esta tecnología no existen enlaces físicos; sólo hay ondas electromagnéticas. La ventaja obvia de una topología celular (inalámbrica) es que no existe ningún medio tangible aparte de la atmósfera terrestre o el del vacío del espacio exterior (y los satélites). Las desventajas son que las señales se encuentran presentes en cualquier lugar de la celda y, de ese modo, pueden sufrir disturbios y violaciones de seguridad. Como norma, las topologías basadas en celdas se integran con otras topologías, ya sea que usen la atmósfera o los satélites.

Topología irregular En este tipo de topología no existe un patrón obvio de enlaces y nodos. El cableado no sigue un modelo determinado; de los nodos salen cantidades variables de cables. Las redes que se encuentran en las primeras etapas de construcción, o se encuentran mal planificadas, a menudo se conectan de esta manera. Las topologías LAN más comunes son:

Ethernet: topología de bus lógica y en estrella física o en estrella extendida.
Token Ring: topología de anillo lógica y una topología física en estrella.
FDDI: topología de anillo lógica y topología física de anillo doble.



[Internet 20]
Topologías de la red

1.2.4.2 Tipos de Redes

Red de comunicación.

Es todo aquel conjunto de elementos basados en hardware y software que permite establecer un enlace entre los clientes y los servidores, se clasifican por su tamaño LAN, MAN y WAN.

Redes de área local

Las redes de área local, generalmente llamadas LAN (Local Area Network), son redes de propiedad privada dentro de un solo edificio o campus de hasta unos cuantos kilómetros de extensión. Se usan ampliamente para conectar computadoras personales y estaciones de trabajo en oficinas de compañías y fábricas con objeto de compartir recursos (por ejemplo, impresoras) e intercambiar información. Las LAN se distinguen de otro tipo de redes por tres características: (1) su tamaño, (2) su tecnología de trasmisión , y (3) su topología.

Las LAN a menudo usan una tecnología de transmisión que consiste en un cable sencillo al cual están conectadas todas las máquinas. Las LAN tradicionales operan a velocidades de 10 a 100 Mbps, tienen bajo retardo (décimas de microsegundos) y experimentan muy pocos errores. Las LAN más nuevas pueden operar a velocidades muy altas, de hasta cientos de megabits/segs. Un megabits es 1,048,576 (2ˆ20) bits.

Las LAN de transmisión pueden tener diversas topologías. En una red de bus en cualquier instante una computadora es la máquina maestra y puede transmitir; se pide a las otras máquinas que se abstengan de enviar mensajes. Es necesario un mecanismo de arbitraje para resolver conflictos cuando dos o más máquinas quieren transmitir simultáneamente. Las computadoras de una Ethernet pueden transmitir cuando quieran; si dos o más paquetes chocan, cada computadora sólo espera un tiempo al azar y lo vuelve a intentar.

Las redes de difusión se pueden dividir también en estáticas y dinámicas, dependiendo de cómo se asigna el canal. Una asignación estática típica divide el tiempo en intervalos discretos y ejecuta un algoritmo de asignación cíclica, permitiendo a cada máquina transmitir únicamente cuando le llegue su turno. La asignación estática desperdicia la capacidad del canal cuando una máquina no tiene nada que decir durante su segmento asignado, por lo que muchos sistemas intentan asignar el canal dinámicamente (es decir, por demanda).

Los métodos de asignación dinámica para un canal común son centralizados o descentralizados. En el método de asignación del canal centralizado hay una sola entidad, por ejemplo una unidad de arbitraje del bus, la cual determina quién es el siguiente. Podría hacer esto aceptando peticiones y tomando una decisión de acuerdo con un algoritmo interno. En el método de asignación de canal descentralizado no hay una entidad central; cada máquina debe decidir por sí misma si transmite o no.

El otro tipo de LAN se construye con líneas de punto a punto. Las líneas individuales conectan una máquina específica a otra. Una LAN así es realmente una red de área amplia en miniatura.

Redes de área metropolitana

Una red de área metropolitana, o MAN (Metropolitan Area Network) es básicamente una vesión más grande de una LAN y normalmente se basa en una tocnología similar. Podría abarcar un grupo de oficinas corporativas cercanas o una ciudad y podría ser privada o pública. Una MAN puede manejar datos y voz, e incluso podría estar relacionada con la red de televisión por cable local. Una MAN sólo tiene uno o dos cables y no contiene elementos de conmutación, los cuales desvían los paquetes por una de varias líneas de salida potenciales. Al no tener que conmutar, se simplifica el diseño.

La principal razón para distinguir las MAN como una categoría especial es que se ha adoptado un estándar para ellas, y este estándar ya se está implementando: se llama DQDB (distributed queue dual bus, o bus dual de cola distribuida) o 802.6 (el número de la norma IEEE que lo define). El DQDB consiste en dos buses (cables) unidireccionales, a los cuales están conectadas todas las computadoras, como se muestra en la figura 1.

Cada bus tiene una cabeza Terminal (head-end), un dispositivo que inicia la actividad de transmisión. El tráfico destinado a una computadora situada a la derecha del emisor usa el bus superior. El tráfico hacia la izquierda usa el de abajo.

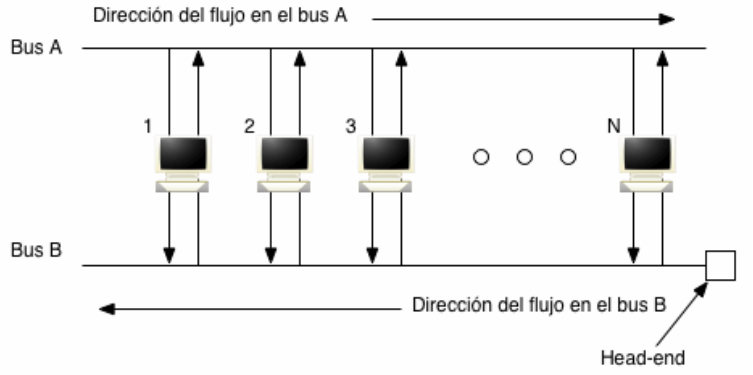


Figura 1. Arquitectura de la red de área metropolitana DQDB.

Un aspecto clave de las MAN es que hay un medio de difusión al cual se conectan todas las computadoras. Esto simplifica mucho el diseño comparado con otros tipos de redes.

Redes de área amplia

Una red de área amplia, o WAN (Wide Area Network), se extiende sobre un área geográfica extensa, a veces un país o un continente; contiene una colección de máquinas dedicadas a ejecutar programas de usuario (es decir, de aplicación). Llamaremos a estas máquinas **hosts**. Los hosts están conectados por una **subred de comunicación,** o simplemente **subred.** El trabajo de la subred es conducir mensajes de un host a otro, así como el sistema telefónico conduce sonidos del que habla al que escucha. La separación entre los aspectos exclusivamente de comunicación de la red(la subred) y los aspectos de aplicación (los hosts), simplifica enormemente el diseño total de la red.
En muchas redes de área amplia, la subred tiene dos componentes distinto: las líneas de transmisión y los elementos de conmutación. Las líneas de transmisión (también llamadas **circuitos, canales o troncales**) mueven bits de una máquina a otra.
Los elementos de conmutación son computadoras especializadas que conectan dos o más líneas de transmisión. Cuando los datos llegan por una línea de entrada, el elemento de conmutación debe escoger una línea de salida para reenviarlos. Desafortunadamente, no hay una terminología estándar para designar estas computadoras; se les denomina **nodos conmutadores de paquetes, sistemas intermedios y centrales de conmutación de datos**, entre otras cosas.
En este caso, usaremos la palabra **enrutador**. En este modelo, mostrado en la figura1 , cada *host* generalmente está conectada a una LAN en la cual está presente un enrutador, aunque en algunos casos una host puede estar conectada directamente a un enrutador. La colección de líneas de comunicación y enrutadores (pero no los hosts) forman la subred.

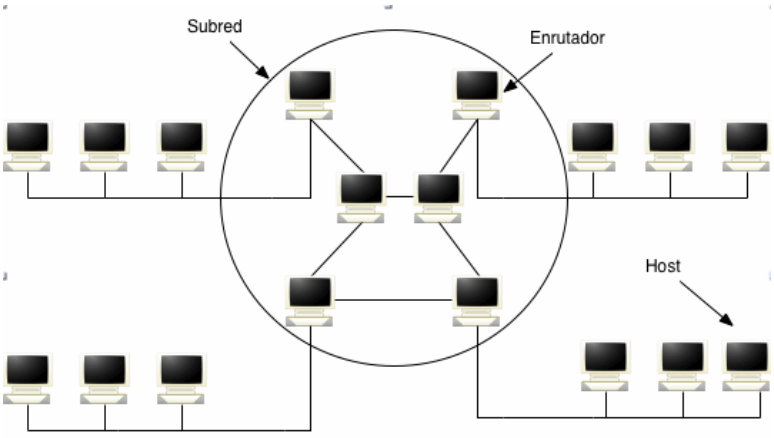


Figura 1. Relación entre las hosts y la subred

En casi todas las WAN, la red contiene numerosos cables o lìneas telefónicas, cada una conectada por un par de enrutadores. Si dos enrutadores que no comparten un cable desean comunicarse, deberán hacerlo indirectamente, por medio de otros enrutadores. Cuando se envía un paquete de un enrutador a otro a través de uno o más enrutadores intermedios, el paquete se recibe completo en cada enrutador intermedio, se almacena hasta que la línea de salida requerida está libre, y a continuación se reenvía. Una subred basada en este principio se llama, de **punto a punto**, de **almacenar y reenviar**, o de **paquete conmutado.** Casi todas las redes de área amplia (excepto aquellas que usan satélites) tienen subredes de almacenar y reenviar. Cuando los paquetes son pequeños y el tamaño de todos es el mismo, suelen llamarse **celdas.**
Cuando se usa una subred punto a punto, una consideración de diseño importante es la topología de interconexión del enrutador. Algunas posibles topologías son: Estrella, Anillo, Árbol , Intersección de anillos, Completa e Irregular.
Las redes locales que fueron diseñadas como tales usualmente tienen una topología simétrica. En contraste, las redes de área amplia típicamente tienen topologías irregulares.

Una segunda posibilidad para una WAN es un sistema de satélite o de radio en tierra. Cada enrutador tiene una antena por medio de la cual puede enviar y recibir. Todos los enrutadores pueden oír las salidas enviadas desde el satélite y en algunos casos pueden también oír la transmisión ascendente de los otros enrutadores hacia el satélite. Algunas veces los enrutadores están conectados a una subred punto a punto de gran tamaño, y únicamente algunos de ellos tienen una antena de satélite. Por su naturaleza, las redes de satélite son de difusión y son más útiles cuando la propiedad de difusión es importante [Tanenbaum, 1997].

Redes inalámbricas

Una de las tecnologías más prometedoras y discutidas en esta década es la de poder comunicar computadoras mediante tecnología inalámbrica. La conexión de computadoras mediante Ondas de Radio o Luz Infrarroja, actualmente está siendo ampliamente investigado. Las Redes Inalámbricas facilitan la operación en lugares donde la

computadora no puede permanecer en un solo lugar, como en almacenes o en oficinas que se encuentren en varios pisos.

También es útil para hacer posibles sistemas basados en plumas. Pero la realidad es que esta tecnología está todavía en pañales y se deben de resolver varios obstáculos técnicos y de regulación antes de que las redes inalámbricas sean utilizadas de una manera general en los sistemas de cómputo de la actualidad.

No se espera que las redes inalámbricas lleguen a remplazar a las redes cableadas. Estas ofrecen velocidades de transmisión mayores que las logradas con la tecnología inalámbrica. Mientras que las redes inalámbricas actuales ofrecen velocidades de 54 Mbps, en lo que se refiere a redes cableadas, existen tres tipos de conexión con tres velocidades de transmisión de datos relacionadas: 10 Mbps, 100 Mbps y 1.000 Mbps.

Sin embargo se pueden mezclar las redes cableadas y las inalámbricas, y de esta manera generar una "Red Híbrida".
Existen dos amplias categorías de Redes Inalámbricas:

1. *De Larga Distancia*.- Estas son utilizadas para transmitir la información en espacios que pueden variar desde una misma ciudad o hasta varios países circunvecinos (mejor conocido como Redes de Area Metropolitana MAN).
2. *De Corta Distancia*.- Estas son utilizadas principalmente en redes corporativas cuyas oficinas se encuentran en uno o varios edificios que no se encuentran muy retirados entre si.

[Internet 6] .

Por otro lados dentro de las redes locales inalámbricas tenemos:

1. Wi-Fi

Wireless Fidelity
Estándares IEEE 802.11 norman su funcionamiento
Tiene limitaciones notables en cuanto a seguridad, calidad de servicio, movilidad e interferencias con otras redes
Tecnología: Nueva
Velocidades: 64 Kbps – 4 Mbps
Difusión: Media
Costo: Medio
Complejidad: Media/Alta

2. WiMax

Worldwide Interoperability for Microwave WiMAX
Se fundamenta en el estándar IEEE 802.16
Se utiliza para proveer accesos en redes metropolitanas (MAN)
Ventajas:
Rapidez de instalación
Velocidad
Seguridad
Calidad de Servicio

La tecnología WiMAX se utiliza en:
Banda Ancha por Demanda
Áreas urbanas sin planta externa
Banda Ancha en zonas rurales
Velocidades :40 Mbps
Difusión: Baja
Costo: Alto
Complejidad: Alta

(Consultar Apéndice B)

1.2.4.3 Protocolos: TCP/IP

Un protocolo es un conjunto de normas de obligado cumplimiento.
Toda la información en Internet ha de ser transmitida mediante el protocolo TCP/IP. Por lo tanto, ya sabemos que todos los datos que van por la red siguen un conjunto de normas.
Antes de ser transmitida, la información se parte en paquetes de unos 4 Kb. A cada paquete se le pone la dirección donde va a ir (el destino), la dirección desde donde parte (el remitente) y el número de orden (entre otros datos) y se envía cada trozo por separado. Cada paquete puede viajar por distinto camino hasta llegar a su destino. El ordenador que recibe la información (los paquetes), vuelve a juntarla.

La ARPANET era una red de investigación patrocinada por el DoD (Departanmento de Defensa de Estados Unidos). Al final conectó a cientos de universidades e instalaciones del gobierno usando líneas telefónicas rentadas. Cuando más tarde se añadieron redes de satélite y radio, los protocolos existentes tuvieron problemas para interactuar con ellas, de modo que se necesitó una arquitectura de referencia nueva. Así, la capacidad de conectar entre sí múltiples redes de manera inconsútil fue uno de los principales objetivos de diseño desde el principio. Esta arquitectura se popularizó después como el modelo de referencia TCP/IP, por las iniciales de sus dos protocolos primarios(*Transport Control Protocol / Internet Protocol*, protocolo de control de transporte / protocolo de Internet). Este modelo se definió por primera vez en (Cerf y Kahn, 1974).
Debido a la preocupación del DoD por que alguno de sus costosos nodos, enrutadores o pasarelas de interredes pudiera ser objeto de un atentado en cualquier momento, otro de los objetivos principales fue que la red fuera capaz de sobrevivir a la pérdida del *hardware* de subred sin que las conversaciones existentes se interrumpieran. En otras palabras, el DoD quería que las conexiones permanecieran intactas mientras las máquinas de origen y destino estuvieran funcionando, aun si alguna de las máquinas o de las líneas de transmisión en el trayecto dejara de funcionar en forma repentina. Es más , se necesitaba una arquitectura flexible, pues se tenía la visión de aplicaciones con requerimientos divergentes, abarcando desde la transferencia de archivos hasta la transmisión de discursos en tiempo real.
Lo sorprendente de TCP/IP es que no fue pensado para resistir el espionaje: los protocolos originales transmiten las contraseñas y datos sin codificación alguna [[Black, 1995]].

“TCP/IP es el protocolo de Internet . En la actualidad es la elección recomendada para casi todas las redes, especialmente si la red tiene salida a Internet “ [Internet 8].

El protocolo TCP/IP tiene que estar a un nivel superior del tipo de red empleado y funcionar de forma transparente en cualquier tipo de red. Y a un nivel inferior de los programas de aplicación (páginas WEB, correo electrónico…) particulares de cada sistema operativo. Todo esto nos sugiere el siguiente modelo de referencia:

Capa de aplicación (HTTP, SMTP, FTP, TELNET...)
Capa de transporte (UDP, TCP)
Capa de red (IP)
Capa de acceso a la red (Ethernet, Token Ring...)
Capa física (cable coaxial, par trenzado...)

TCP/IP y comunicación entre redes

Con el objeto de comprender la forma de funcionar de TCP/IP, hay que presentar previamente algunos términos y conceptos. Los términos *pasarela (gateway) y sistema de encaminamiento (router)* se utilizan para denominar a las máquinas que realizan la retransmisión entre redes.
Las redes A , B y C se denominan comúnmente *subredes*. Esta denominación no quiere decir que realicen menos funciones que las redes convencionales, sino que las tres redes forman una única red lógica y las subredes contribuyen en las operaciones globales de interconexión.
Las pasarelas entre redes se diseñan de forma que sean transparentes a las aplicaciones de los usuarios finales. De hecho, las aplicaciones de usuarios residen en computadoras conectadas a las redes. Es raro que las pasarelas tengan aplicaciones de usuario. Este esquema resulta atractivo desde diversos puntos de vista. En primer lugar, la pasarela no necesita cargase con los protocolos del nivel de aplicación. Como no son invocados por la pasarela, ésta se puede dedicar a otras tareas, como la gestión del tráfico entre las redes. No se ocupa de funciones del nivel de aplicación como acceso a bases de datos, correo electrónico y gestión de archivos.

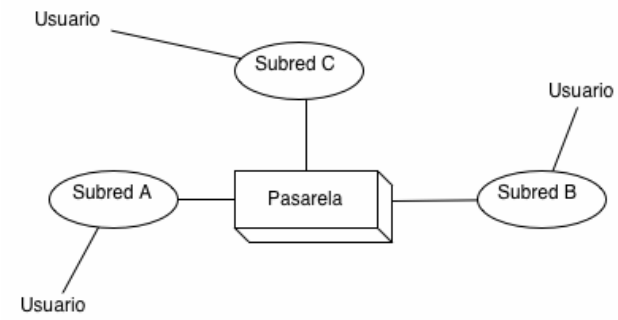


Figura 2. Subredes y pasarelas(gateway).

En segundo lugar, este esquema permite que la pasarela dé soporte a cualquier tipo de aplicación , ya que considera que el mensaje de la aplicación no es más que una unidad de datos de protocolo transparente (PDU), para ver más detalles sobre PDU consultar la página: http://www.saulo.net/pub/tcpip/b.htm#3-2.

Además de la transparencia para el nivel de aplicación, la mayoría de los diseñadores intenta que las pasarelas sean transparentes a las subredes, y viceversa. El propósito principal de la pasarela es recibir una PDU que contenga información de direccionamiento

suficiente para que se pueda encaminar hacia su destino final o hasta la pasarela siguiente. Esta característica es muy atractiva, ya que convierte a la pasarela en algo modular, que se puede utilizar con diferentes tipos de redes.

Protocolo IP

IP es el principal protocolo de la capa de red. Este protocolo define la unidad básica de transferencia de datos entre el origen y el destino. Además, IP es el encargado de elegir la ruta más adecuada por la que los datos serán enviados. Se trata de un sistema de entrega de paquetes (llamados *datagramas IP*) que tiene las siguientes características:

Es no orientado a conexión debido a que cada uno de los paquetes puede seguir rutas distintas entre el origen y el destino. Entonces pueden llegar duplicados o desordenados.

Es no fiable porque los paquetes pueden perderse, dañarse o llegar retrasados.

Nota: *El protocolo IP está definido en la RFC 791 (en español consultar la pagina http://www.arrakis.es/~pjleon/rfc-es/rfc/rfc0791-es.txt).*

[Internet 9]

Estructura de direcciones IP

Las redes TCP/IP identifican las computadoras y las redes a las que están conectadas utilizando direcciones de 4 bytes (32 bits).
El formato de una dirección IP es DIRECCIÓN IP = DIRECCIÓN DE RED + DIRECCIÓN DE LA COMPUTADORA [Black, 1995].

La dirección IP no identifica por sí misma a una computadora, sino más bien la conexión de una computadora con su red. En consecuencia, si una máquina se traslada a otra red, su espacio de direcciones deberá ser modificado [Tanenbaum, 1997].

Cada host conectado a una red tiene una dirección IP asignada, la cual debe ser distinta a todas las demás direcciones que estén vigentes en ese momento en el conjunto de redes visibles por el host. En el caso de Internet, no puede haber dos ordenadores con 2 direcciones IP (públicas) iguales. Pero sí podríamos tener dos ordenadores con la misma dirección IP siempre y cuando pertenezcan a redes independientes entre sí (sin ningún camino posible que las comunique) [Internet 11].

Las direcciones IP se clasifican en:

Direcciones IP públicas. Son visibles en todo Internet. Un ordenador con una IP pública es accesible (visible) desde cualquier otro ordenador conectado a Internet. Para conectarse a Internet es necesario tener una dirección IP pública.

Direcciones IP privadas (reservadas). Son visibles únicamente por otros hosts de su propia red o de otras redes privadas interconectadas por routers. Se utilizan en las empresas para los puestos de trabajo. Los ordenadores con direcciones IP privadas pueden salir a Internet por medio de un router (o *proxy*) que tenga una IP pública. Sin embargo, desde Internet no se puede acceder a ordenadores con direcciones IP privadas.

A su vez, las direcciones IP pueden ser:

Direcciones IP estáticas (fijas). Un host que se conecte a la red con dirección IP estática siempre lo hará con una misma IP. Las direcciones IP públicas estáticas son las que utilizan los servidores de Internet con objeto de que estén siempre localizables por los usuarios de Internet.

Direcciones IP dinámicas. Un host que se conecte a la red mediante dirección IP dinámica, cada vez lo hará con una dirección IP distinta. Las direcciones IP públicas dinámicas son las que se utilizan en las conexiones a Internet mediante un módem. Los proveedores de Internet utilizan direcciones IP dinámicas debido a que tienen más clientes que direcciones IP (es muy improbable que todos se conecten a la vez) .

Las direcciones IP se suelen representar por una secuencia de cuatro enteros separados por puntos de la forma a.b.c.d donde cada una de estas letras es un número comprendido entre el 0 y el 255 [Internet 11].

De esos cuatro números, algunos se utilizan como *dirección de red* y los restantes como *dirección de host*. Todos los hosts que pertenezcan a la misma red deberán tener en común la dirección de red y diferir en la dirección de host.

Las direcciones IP no se encuentran aisladas en Internet, sino que pertenecen siempre a alguna red. Todas las máquinas conectadas a una misma red se caracterizan en que los primeros bits de sus direcciones son iguales. De esta forma, las direcciones se dividen conceptualmente en dos partes: el *identificador de red* y el *identificador de host*.
Dependiendo del número de hosts que se necesiten para cada red, las direcciones de Internet se han dividido en las **clases A**, **B** y **C**. [Internet 9]

Clase	Máscara de red	Dirección de Red
Clase A	255.0.0.0	1.0.0.0 - 126.255.255.255
Clase B	255.255.0.0	128.0.0.0 - 191.255.255.255
Clase C	255.255.255.0	224.0.0.0 - 239.255.255.255

[Internet 10]

Principales características de IP

IP es un ejemplo de servicio no orientado a conexión. Permite, sin establecimiento de llamada previa, el intercambio de datos entre dos computadoras (sin embargo, las dos computadoras generalmente comparten un protocolo común de transporte orientado a conexión). Como IP no es orientado a conexión, se pueden perder datagramas entre las dos estaciones de usuario. Por ejemplo, las pasarelas IP utilizan un tamaño máximo de cola, y si sobrepasa, los buffers se desbordarán. En esta situación se descartarán datagramas en la red. Por esta razón es fundamental un protocolo de transporte de nivel superior (como TCP) que solucione esos problemas.

IP oculta la subred que hay debajo a los usuarios finales. Crea para ellos una red virtual. Este aspecto de IP permite que diferentes redes se conecten a una pasarela IP. Como

resultado, IP es razonablemente simple de instalar y, debido a su diseño no orientado a conexión, es muy versátil [Black, 1995].

Dado que IP es un protocolo de tipo datagrama, no dispone de mecanismos para proporcionar fiabilidad. No proporciona procedimientos de recuperación de errores en las redes subyacentes ni mecanismos de control de flujo. Los datos de usuario (datagramas) se pueden perder, duplicar o incluso llegar desordenados. No es trabajo de IP ocuparse de esos problemas. La mayoría de esos problemas se pasan al nivel superior TCP.

Valor del nivel del transporte

El Protocolo Internet no está diseñado para solucionar problemas ni garantiza el envío de tráfico. IP está diseñado para solucionar problemas obsoletos o que han sobrepasado el número de saltos entre redes permitidos.

Ciertas aplicaciones de usuario requieren asegurarse de que todos los datagramas han llegado correctamente a su destino. Es más, el usuario transmisor puede necesitar asegurarse de que el tráfico se ha enviado a la computadora receptora. Los mecanismos para realizar esos importantes servicios residen en TCP (UDP es no orientado a conexión y no proporciona esos servicios).

TCP debe ser capaz de satisfacer un amplio rango de requerimientos de las aplicaciones e, igualmente importante, deber ser capaz de adaptarse a un entorno dinámico en la interred . Debe ser capaz de establecer y gestionar sesiones entre los usuarios locales y remotos. Esto significa que TCP debe tener conocimiento de las actividades de los usuarios para dar soporte a la transferencia de sus datos por la interred [Black, 1995].

Protocolo TCP

El TCP (Transmission Control Protocol, protocolo de control de transmisión) se diseñó específicamente para proporcionar una corriente de bytes confiable a través de una interred no confiable. Una interred es diferente a una sola red porque las distintas partes pueden tener topologías, anchos de banda, retardos, tamaños de paquete y otros parámetros con grandes diferencias.

Cada máquina que reconoce el TCP tiene una entidad de transporte TCP , ya sea un proceso de usuario o una parte del núcleo que maneja las corrientes TCP y tiene una interfaz con la capa IP. Una entidad TCP acepta corrientes de datos de usuario de los procesos locales, los divide en partes que no excedan 64K bytes, y envía cada parte como datagrama IP independiente. Cuando llegan a una máquina datagramas IP que contienen datos TCP, son entregados a la entidad TCP, que reconstruye las corrientes originales de bytes.

La capa IP no ofrece ninguna garantía de que los datagramas se entregarán adecuadamente por lo que es responsabilidad del TCP terminar de temporizar y retransmitirlos según se necesite. Los datagramas que sí llegan pueden hacerlo desordenadamente; también es responsabilidad del TCP reensamblarlos en mensajes con la secuencia adecuada. El TCP debe proveer la confiabilidad que la mayoría de los usuarios quiere y que el IP no proporciona.

Esta diseñado para residir en las computadoras o en las máquinas que se ocupan de conservar la integridad de la transferencia de datos entre extremos. Lo más común es que TCP resida en las computadoras de los usuarios.

Como IP es una red no orientada a conexión, es TCP quien se debe encargar de las tareas de fiabilidad, control de flujo, secuenciamiento , aperturas y cierres [Black, 1995].

Modelo de servicio TCP

El servicio de TCP se obtiene haciendo que el transmisor como el receptor creen puntos terminales, llamados sockets. Cada socket tiene un número (dirección) de socket que consiste en la dirección IP del host y en un número de 16 bits local a ese host, llamado puerto. Para obtener el servicio TCP, debe establecerse explícitamente una conexión entre un socket de la máquina transmisora y un socket de la máquina receptora.

Todas las conexiones TCP son dúplex integral y punto a punto. Dúplex integral significa que el tráfico puede ir en ambos sentidos al mismo tiempo. Punto a punto significa que cada conexión tiene exactamente dos puntos terminales.

Una conexión TCP es una corriente de bytes, no una corriente de mensajes.
Cuando una aplicación pasa datos al TCP, el TCP puede enviarlos de inmediato o guardarlos en un buffer (a fin de reunir una cantidad mayor de información para enviarla junta).
[Black , 1995].

El protocolo TCP (*Transmission Control Protocol*, protocolo de control de transmisión) está basado en IP que es no fiable y no orientado a conexión, y sin embargo es:

Orientado a conexión. Es necesario establecer una conexión previa entre las dos máquinas antes de poder transmitir ningún dato. A través de esta conexión los datos llegarán siempre a la aplicación destino de forma ordenada y sin duplicados. Finalmente, es necesario cerrar la conexión.

Fiable. La información que envía el emisor llega de forma correcta al destino.

El protocolo TCP permite una comunicación fiable entre dos aplicaciones. De esta forma, las aplicaciones que lo utilicen no tienen que preocuparse de la integridad de la información: dan por hecho que todo lo que reciben es correcto.

El flujo de datos entre una aplicación y otra viajan por un *circuito virtual*. Sabemos que los datagramas IP pueden seguir rutas distintas, dependiendo del estado de los encaminadores intermedios, para llegar a un mismo sitio. Esto significa que los datagramas IP que transportan los mensajes siguen rutas diferentes aunque el protocolo TCP logré la ilusión de que existe un único circuito por el que viajan todos los bytes uno detrás de otro (algo así como una tubería entre el origen y el destino). Para que esta comunicación pueda ser posible es necesario abrir previamente una conexión. Esta conexión garantiza que los todos los datos lleguen correctamente de forma ordenada y sin duplicados. La unidad de datos del protocolo es el *byte*, de tal forma que la aplicación origen envía bytes y la aplicación destino recibe estos bytes.

Sin embargo, cada byte no se envía inmediatamente después de ser generado por la aplicación, sino que se espera a que haya una cierta cantidad de bytes, se agrupan en un *segmento* y se envía el segmento completo. Para ello son necesarias unas *memorias intermedias* o *buffers*. Cada uno de estos segmentos viaja en el campo de datos de un datagrama IP. Si el segmento es muy grande será necesario fragmentar el datagrama, con la consiguiente pérdida de rendimiento; y si es muy pequeño, se estarán enviando más cabeceras que datos.
El protocolo TCP envía un *flujo de información no estructurado*. Esto significa que los datos no tienen ningún formato, son únicamente los bytes que una aplicación envía a otra. Ambas aplicaciones deberán ponerse de acuerdo para comprender la información que se están enviando.

Cada vez que se abre una conexión, se crea un canal de comunicación bidireccional en el que ambas aplicaciones pueden enviar y recibir información, es decir, una conexión es *full-dúplex*.

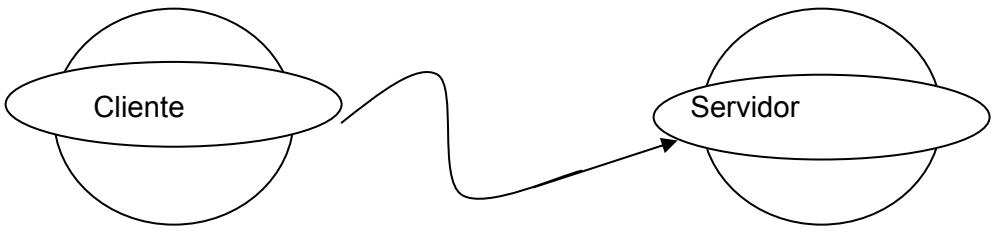
[Internet 12]

1.2.4.4 Arquitectura Cliente Servidor

Normalmente en Internet se usa la arquitectura cliente-servidor.
Este tipo de organización se basa en que entre todas las computadoras que están en la red, unas ofrecen servicios (los llamados servidores) y otras usan esos servicios (los denominados clientes). Como ejemplo , cuando están viendo páginas en Internet, están accediendo a un servicio (pidiendo una página WEB concreta) que les ofrece un servidor de páginas WEB (sirviéndoles la página solicitada). Por lo tanto, su computadora es un cliente y el que hospeda estas páginas es un servidor [Internet 13].

Una arquitectura es un conjunto de reglas, definiciones, términos y modelos que se emplean para producir un producto.

La arquitectura Cliente/Servidor agrupa conjuntos de elementos que efectúan procesos distribuidos [Internet 14].



Generador de peticiones Elemento de enlace Ofrece el servidor
[Internet 14]
El Modelo Cliente/ Servidor es una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En este modelo, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo *mainframe*(ver glosario). Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario. En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar el mantenimiento.

Beneficios:

1. Mejor aprovechamiento de la potencia de cómputo (Reparte el trabajo).
2. Reduce el tráfico en la Red. (Viajan requerimientos).
3. Opera bajo sistemas abiertos.
4. Permite el uso de interfaces gráficas variadas y versátiles.

Definición de Cliente

Conjunto de Software y Hardware que invoca los servicios de uno o varios servidores.
Los clientes en una red cliente-servidor son las máquinas o procesos que piden información, recursos y servicios a un servidor unido. Estas peticiones pueden ser cosas como proporcionar datos de una base de datos, aplicaciones, partes de archivos o archivos completos a la máquina cliente. Los datos, aplicaciones o archivos pueden residir en un servidor y ser simplemente accedidos por el cliente o pueden ser copiados o movidos físicamente a la máquina cliente. Esta disposición permite a la máquina cliente ser relativamente pequeña. Para cada tipo de entorno de cliente, hay habitualmente software específico (y a veces hardware) en el cliente, con algún software y hardware análogo en el servidor [Internet 15].

Los servidores pueden ser sistemas operativos diferentes como Windows NT, Windows 95, OS/2, Unix. Unix es popular porque como sistema operativo de servidores puede ser utilizado en muchos tipos de configuraciones sobre máquinas servidor además de como servidores de archivos y servidores de impresión [Internet 15].

Funciones Comunes del Cliente:

1. Mantener y procesar todo el dialogo con el usuario.
2. Manejo de pantallas.

3. Menús e interpretación de comandos.
4. Entrada de datos y validación.
5. Procesamiento de ayudas.
6. Recuperación de errores.
7. Administrar la interfaz de usuario.
8. Interactuar con el usuario.

[Internet 14]

Definición de Servidor

Conjunto de Hardware y Software que responde a los requerimientos de un cliente [Internet 14].
Los servidores en una red cliente-servidor son los procesos que proporcionan información recursos y servicios a los clientes de la red. Cuando un cliente pide un recurso como, por ejemplo, un archivo, datos de una base de datos, acceso a aplicaciones remotas o impresión centralizada, el servidor proporciona estos recursos al cliente. Como se mencionó antes, los procesos del servidor pueden residir en una máquina que también actúa como cliente de otro servidor. Además de proporcionar este tipo de recursos, un servidor puede dar acceso a otras redes, actuando como un servidor de comunicaciones que conecta a otros servidores o minicomputadoras que actuan como hosts de la red [Internet 16].

También puede permitir enviar faxes o correo electrónico desde un cliente en una red a un cliente en otra red. Puede actuar como servidor de seguridad, como servidor de gestión de la red, como servidor de directorios o de acceso. [Rosen, 1997]

Tipos Comunes de Servidores:

1. Servidor de Archivos (FTP, Novell).
2. Servidor de Bases de Datos (SQL, CBASE, ORACLE, INFORMIX).
3. Servidor de Comunicaciones
4. Servidor de Impresión.
5. Servidor de Terminal.
6. Servidor de Aplicaciones (Windows NT, Novell).

Funciones Comunes del Servidor:

1. Acceso, almacenamiento y organización de datos.
2. Actualización de datos almacenados.
3. Administración de recursos compartidos.
4. Ejecución de toda la lógica para procesar una transacción.
5. Procesamiento común de elementos del servidor (Datos, capacidad de CPU, almacenamiento en disco, capacidad de impresión, manejo de memoria y comunicación).

[Internet 14]

1.2.5 El Proceso Unificado

El **Proceso Unificado** "es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Se basa en muchos

años de experiencia en el uso de la tecnología orientada a objetos en el desarrollo de software de misión crítica en una variedad de industrias por la compañía Rational", donde confluyen Grady Booch, James Rumbaugh e Ivar Jacobson

El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.
El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

- Interacción con el usuario continua desde un inicio
- Mitigación de riesgos antes de que ocurran
- Aseguramiento de la calidad
- Involucramiento del equipo en todas las decisiones del proyecto
- Anticiparse al cambio de requerimientos

El Proceso Unificado es un proceso porque "define quién está haciendo qué, cuándo lo hace y cómo alcanzar cierto objetivo, en este caso el desarrollo de software" [Jacobson 1998]. Según [Booch 1998], los **conceptos clave del Proceso Unificado** son:

Fases e iteraciones	¿Cuándo se hace?
Flujos de trabajo de procesos (actividades y pasos)	¿Qué se está haciendo?
Artefactos (modelos, reportes, documentos)	¿Qué se produjo?
Trabajador: un arquitecto	¿Quién lo hace?

[Internet 18]

Resumiendo, el Proceso Unificado de Desarrollo (Unified Process) es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Como una de sus características principales tenemos que el Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial de Proceso Unificado.

Básicamente, los aspectos del Proceso Unificado se pueden resumir en frases clave: *Dirigido por casos de uso*, *Centrado en la arquitectura*, *Iterativo e incremental*.

El Proceso unificado consta de 4 fases:

- La Fase de inicio.
- La Fase de Elaboración.
- La Fase de Construcción.
- La Fase de Transición.

Estas fases aplican de manera iterativa cada uno de los 9 flujos de trabajo fundamentales de los que consta el Proceso Unificado:

1. Modelado del Negocio: Describe la estructura y la dinámica de la organización.
2. Requisitos: Describe el método basado en casos de uso para extraer los requisitos.
3. Análisis y Diseño: Describe las diferentes vistas arquitectónicas.
4. Implementación: Tiene en cuenta el desarrollo de software, la prueba de unidades y la integración.
5. Pruebas: Describe los casos de pruebas, los procedimientos y las métricas para evaluación de defectos.
6. Despliegue: Cubre la configuración del sistema entregable.
7. Gestión de configuraciones: Controla los cambios y mantiene la integridad de los artefactos de un proyecto.
8. Gestión del Proyecto: Describe varias estrategias de trabajo en un proceso iterativo.
9. Entorno: Cubre la infraestructura necesaria para desarrollar un sistema.

A continuación se dará una breve descripción de que es y como funciona UML, ya que será el lenguaje de modelado a utilizar durante el desarrollo del sistema.

UML

El lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje para especificar, construír, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO).

Lenguaje

Un lenguaje de modelado es un lenguaje cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema. UML es un lenguaje estándar para describir planos de software.
UML es una técnica de modelado de objetos, por lo que supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado es la construcción de un modelo a partir de una especificación. Un modelado es una simplificación de la realidad, se elabora para comprender mejor el sistema a desarrollar. Se consigue un modelo completo de la realidad cuando el modelo captura los aspectos importantes del problema y omite el resto.

Visualizar

Es un lenguaje gráfico. Detrás de cada símbolo hay una notación bien definida; por lo que un desarrollador puede escribir un modelo en UML, y otro desarrollador puede interpretar ese modelo sin ambigüedad.

Construir

Sus modelos pueden conectarse directamente a una gran cantidad de lenguajes de programación. Permite la generación de código a partir de un modelo UML, en un lenguaje de programación , y también reconstruir un modelo UML a partir de una implementación.

Documentar

Cubre la documentación de la arquitectura de un sistema y todos sus detalles, también proporciona un lenguaje para expresar requisitos y pruebas.

Con la creación de UML se busca obtener un lenguaje capaz de abstraer cualquier tipo de sistema, sea informático o no, mediante diagramas, es decir, mediante representaciones gráficas que contienen toda la información relevante del sistema.

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.
Los artefactos de UML se especifican en forma de diagramas, éstos, y la documentación del sistema constituyen los artefactos principales que el modelador puede observar.

Diagrama de Clases

Sirve para visualizar las relaciones entre las clases que involucran al sistema.

Elementos

- Clase

Es la unidad básica que encapsula toda la información de un Objeto. Se representa por un rectángulo que posee tres divisiones:

- Nombre de la clase
- Característica de la Clase
- Son la forma como interactúa el objeto con su entorno

- Relaciones

Es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con diferentes características y objetivos diferentes).

Ejemplo de relaciones

(Comunicación) Indica la invocación desde un actor o caso de uso a otra operación(caso de caso).
Si la fecha no tiene dirección indica comunicación entre elementos.

Asociación

Es una relación entre clasess, en la que una clase depende de la otra, ó se a que se instancia. El estereotipo de la relación indica como es su comunicación(<<incluye>> indica que el caso de uso se debe ejecutar necesariamente, <<extend>> su ejecución condicionada).

Dependencia o Instanciación

Indica herencia, cuando un elemento obtiene las características y comportamientos de otro, además de los propios. También se puede marcar con un estereotipo para saber como se relaciona con su padre.

Generalización

La cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación éstas pueden ser:

1. uno o muchos: 1..* (1..n)
2. 0 o muchos: 0..* (0..n)
3. número fijo : m (m denota el número)

Permite asociar objetos que colaboran entre sí; el término de vida de un objeto no depende del otro.

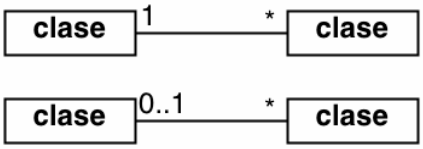


Diagrama de Casos de Uso

Un caso de uso es una interacción típica entre un usuario y un sistema de cómputo. Las características del caso de uso son:

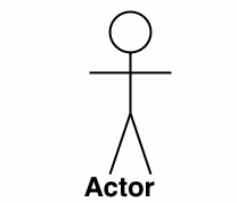
- Capta alguna función visible al usuario
- Puede ser pequeño o grande
- Logra un objetivo discreto para el usuario

El caso de uso se obtiene hablando con los usuarios habituales y analizando con ellos las distintas cosas que deseen hacer con el sistema.

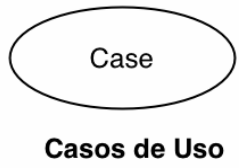
El diagrama de Casos de Uso representa la forma en como un Cliente(Actor) opera con el sistema en desarrollo, además de la forma , tipo y orden en como los elementos interactúan (operaciones o casos de uso).

En la elaboración del caso de uso respondemos a la pregunta **¿Qué?** (es lo que tiene que hacer el sistema o modelo).

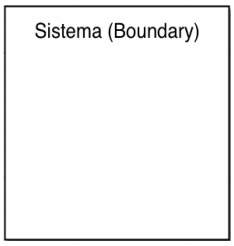
***Elementos:**



Es un rol que un usuario juega con respecto al sistema. Rol indica que un Actor no necesariamente representa a una persona en particular, sino la labor que realiza frente al sistema.



Es una operación / tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.



Indica la agrupación en un sistema de los elementos contenidos dentro de su área.

Boundary

<u>Diagrama de Actividad</u>

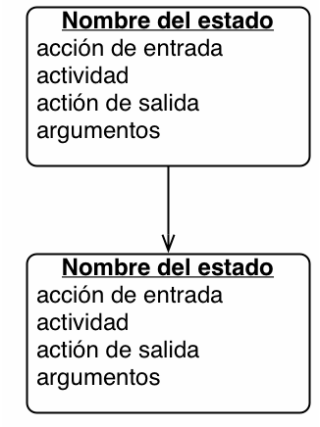
Se utiliza para mostrar el flujo de operaciones que se desencadenan en un procedimiento interno del sistema.
Son una generalización de los diagramas de casos de uso, en varios casos se combinan para formar una actividad o subsistema.

En éste diagrama, casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la aciión ejecutada en el estado anterior. Generalmente modelan los estado de un algoritmo y pueden dar detalle de un caso de uso, u objeto o un mensaje en un objeto. Sirven para representar transiciones internar, sin hacer énfasis en transiciones o eventos externos.

***Elementos**

Representa una estado con acción interna.

Estado de Acción



Relación entre dos estados y se encuentran unidos por flechas; indicando que un objeto que está en el primer estado realizará una acción específica y entrará en el segundo estado cuando un evento implícito ocurra y unas condiciones específicas sean satisfechas.

***Otros Elementos**

La ventaja de utilizar UML como lenguaje de modelado es que es el lenguaje de modelado más utilizado y además reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos.

Una vez que se han descrito los conceptos de una de las herramientas a utilizar durante el desarrollo del proyecto veremos al análisis de la situación actual y definiremos el problema que existe actualmente en el museo de las ciencias UNIVERSUM y lo requerimientos necesarios para erradicarlo.

Capítulo 2

2. Análisis de la situación actual, definición del problema y análisis de requerimientos

Actualmente Universum recibe la visita de numerosas instituciones desde nivel preescolar hasta nivel medio superior, tanto públicas, privadas y las que tienen convenios con la SEP . Para poder realizar una visita guiada a las diferentes salas del museo, el área de Atención al Visitante realiza el registro de las instituciones que desean tener la visita guiada, ingresando como datos necesarios:

- Nombre de la Institución
- Nombre de la persona responsable del grupo de personas a las que se les hará la visita guiada
- Nombre de la persona que está realizando la reservación; ya sea vía telefónica o personalmente
- Domicilio de la Institución
- Colonia a la que pertenece la Institución
- Delegación o municipio
- Teléfono(s)
- Número de lada en caso de estar fuera del D.F.
- Número de fax
- Correo electrónico
- Número de estudiantes a las que se les va a brindar el recorrido
- Número de profesores que van a asistir a la visita (en caso de no serlo se toma como monitor)
- Número de padres de familia que asistirán a la visita
- Nivel de estudios de los estudiantes a los que se les hará el recorrido (maternal, preescolar, primaria, secundaria, bachillerato, etc.)
- Grado al que pertenecen(año / semestre)
- Tiempo de permanencia en el museo
- Tipo de escuela(oficial/particular/con convenio de la SEP)
- Nombre de la(s) sala(s) a las que se les va a dar mayor preferencia para ser visitadas
- Actividad a realizar(conferencia, taller, etc.)
- Nombre de la persona que atendió en la llamada
- Observaciones
- Nombre de la persona que confirma la visita (en caso de haber confirmación)
- Fecha de confirmación
- Clave que se le asigna a la Institución para conservar su fecha de visita guiada
- Día en que se elabora la reservación
- Fecha planeada para la visita
- Hora a la que se va a realizar el recorrido

Los datos anteriores son registrados manualmente haciendo uso de papeletas recicladas, donde sólo una de las caras de la hoja puede ser utilizada, la cual tiene impresos los datos que requiere el Área de Atención al Visitante para hacer la reservación de la visita guiada.
El formato que se utiliza actualmente es el descrito en el capítulo uno.

Definición del problema

La reservación de una visita guiada dentro del museo es tardada, ya que este proceso se está llevando a cabo manualmente, es decir, dependiendo de la persona que tome la llamada (recepcionista) será menos tardado o más tardado dependiendo de la rapidez que tenga para escribir los datos en las papeletas. Una vez hecho el registro de la Institución dichos registros son archivados, por lo que en caso de haber una cancelación en una visita guiada se tiene que llevar a cabo la búsqueda de la Institución que está haciendo la cancelación, lo cual lleva bastante tiempo.
La generación de la clave se lleva a cabo manualmente de la siguiente manera:

CLAVE	XddmmN

Donde :
X =Letra aleatoria que se le asigno al mes en que se esta haciendo la reservación.
dd = día planeado para la visita
mm = mes planeado para la visita
N = Número de llamada del día.

X depende de la letra que la recepcionista le asignó al mes en el año en curso, esta letra puede ser cualquiera siempre y cuando no existan letras iguales para diferentes meses, por lo que se tiene un solo calendario a lo largo del año que no puede ser cambiado ni extraviado, ya que tiene escrito las letras asignadas para cada mes en el transcurso del año. Esto implica pérdida de tiempo para la recepcionista, tiempo en el que podría estar atendiendo una llamada que requiera de los servicios del museo.
En el momento en el que se utiliza papel se están desperdiciando recursos de la institución que no es necesario gastar.

Análisis de requerimientos

Una vez que se realizó el análisis de la problemática, se comprendió una parte de lo que se esperaba del producto e hice un análisis de sus posibles requerimientos. Lo importante aquí, es llegar a un consenso con los futuros usuarios y los involucrados para conocer los alcances del proyecto. De ésta forma se concluyó que el sistema a realizar será dividido en módulos para llevar un mejor control del mismo y su correcta funcionalidad . Por lo que los módulos a realizar serán:

- Validación de usuario
- Instituciones
- Usuarios
- Confirmar/Cancelar Visita
- Hacer reservación
- Consulta

Requerimientos: Validación de usuario

Inicialmente, al usuario se le muestra una pantalla que consta de un campo de texto para el login del usuario y un campo para la contraseña. Después de que el usuario llenó los campos de texto, selecciona el botón “Submit” o “Reset” en caso de que el usuario haya llenado incorrectamente los campos. Después de seleccionar el botón

"Aceptar", la aplicación toma los datos y los compara contra los datos registrados, y si coinciden, se le permite el acceso a la aplicación.
En caso contrario se le muestra una pantalla de datos incorrectos para que lo intente nuevamente.

Requerimientos: Menú

Una vez que el login y el password han sido tecleados correctamente, el usuario puede acceder al menú de opciones que ofrece el sistema, en el cual solo podrá elegir una de las siguientes opciones: Registrar, Confirmar/Cancelar Visita, Hacer Reservación, Consulta o Salir del sistema). Una vez elegida la operación a realizar oprimirá el botón de "Submit" que lo llevará a la aplicación elegida.

Requerimientos: Instituciones

Si el usuario escogió la opción de "Instituciones" dentro del menú entrará a ésta aplicación donde deberá escoger entre tres opciones: Borrar , Agregar o Modificar Institución.
Si la opción seleccionada es "Borrar Institución", entrará a otra aplicación que le pedirá teclear la clave de la institución, ésta aplicación comparará la clave tecleada con las claves ya registradas, si la clave coincide con alguna de las que ya están registradas le permitirá acceder a otra aplicación que le mostrará los datos de la institución de la cual se tecleó la clave y se le preguntará si realmente desea borrar dicho registro, si escoge la opción "Si", automáticamente se borrará dicho registro, pero si escoge la opción "No", entonces lo regresará y le preguntará que operación desea realizar de las tres opciones descritas anteriormente.
Si la opción seleccionada es "Agregar Institución", el usuario entrará una aplicación donde deberá llenar un formato para registrar los datos de la institución, realizado esto , entrará a otra aplicación donde se le mostrarán los datos que fueron llenados en le formulario y se le preguntará si los datos mostrados son correctos, si selecciona la opción "Si" entonces se generará una clave de la institución que servirá como identificador de cada institución, pero si selecciona la opción "No", entonces se regresará a la aplicación donde se encuentra nuevamente el formulario.
Si la opción seleccionada es "Modificar Institución", entrará a una aplicación donde se le pedirá la clave de la institución, si la clave existe, entonces se le mostrarán los datos de la institución de la cual tecleó su clave, sino se le volverá a pedir que tecleé nuevamente la clave. Una vez que la clave es correcta y los datos que se le han mostrado coinciden con los datos de la institución de la cual se requiere modificar su registro, se le mostrará una pantalla con todos los datos de la escuela y con los campos abiertos para realizar cualquier modificación.

Requerimientos: Usuarios

En ésta aplicación van a existir tres opciones a escoger: Borrar, Agregar o Modificar Usuario.
Si la opción elegida es "Borrar Usuario",lo primero que se le pedirá al usuario teclear será el "login del usuario" que se quiere borrar , y por seguridad se pedirá también el "password del administrador" ya que solo él será la persona autorizada para realizar éste tipo de operaciones, ambas claves tecleadas serán comparadas con otra aplicación para verificar si son correctas, de serlo, se mostrarán los datos del usuario del cual se desea borrar su registro, y se le preguntará si realmente desea borrar el

registro de dicho usuario, si elige la opción “Si”, entonces el registro se borrará automáticamente, pero si elige la opción “No”, entonces se le preguntará nuevamente que tipo de operación desea realizar.
Si la opción seleccionada es “Agregar Usuario” entrará automáticamente a un formulario, donde se deberán llenar todos los campos correspondientes para llevar un registro de los usuarios que utilizan el sistema. Una vez que los datos han sido registrados, al usuario se le mostrarán en otra aplicación para verificar que han sido tecleados correctamente, si los datos son correctos, el usuario quedará registrado automáticamente y terminará la aplicación, pero si existió algún error se le regresará nuevamente al formulario para corregir el error.
Por otra parte, si la opción seleccionada fue “Modificar Usuario”, lo primero que se le pedirá al usuario será el “login del usuario” del cual se desea modificar su registro y el “password del administrador” por razones que ya se mencionaron anteriormente, una vez hecho ésto, otra aplicación se encargará de hacer la verificación de ambas claves, si son correctas al usuario se le mostrarán los datos del usuario a modificar y se le preguntará si realmente es el usuario del cual se desea hacer la modificación en su registro, si elige la opción “Si” entonces entrará a un formulario que tendrá los campos abiertos para realizar cualquier modificación, pero si elige la opción “No” entonces se le pedirá que vuelva a teclear el “login del usuario” y el “password del administrador”.

Requerimientos: Confirmar/ Cancelar Visita

Esta aplicación permite al usuario realizar una confirmación o cancelación de una visita guiada. Primero se le mostrará una pantalla que pregunté - ¿Desea Confirmar Visita?- si el usuario selecciona el botón “Sí” automáticamente se registrará la confirmación de la visita guiada de dicha escuela, pero si el usuario selecciona el botón “No”, entonces se le preguntará si está realmente seguro de querer cancelar la visita guiada de dicha institución mostrándole los datos de la escuela a la cual se le desea cancelar la visita.

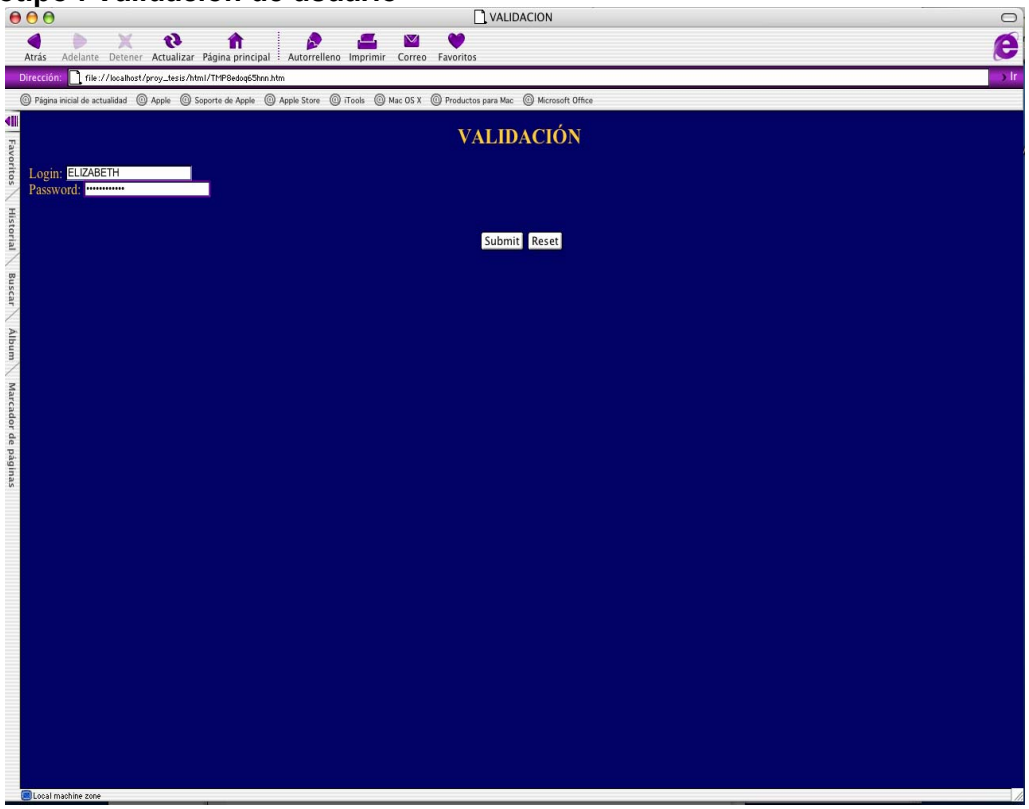
Requerimientos: Hacer Reservación

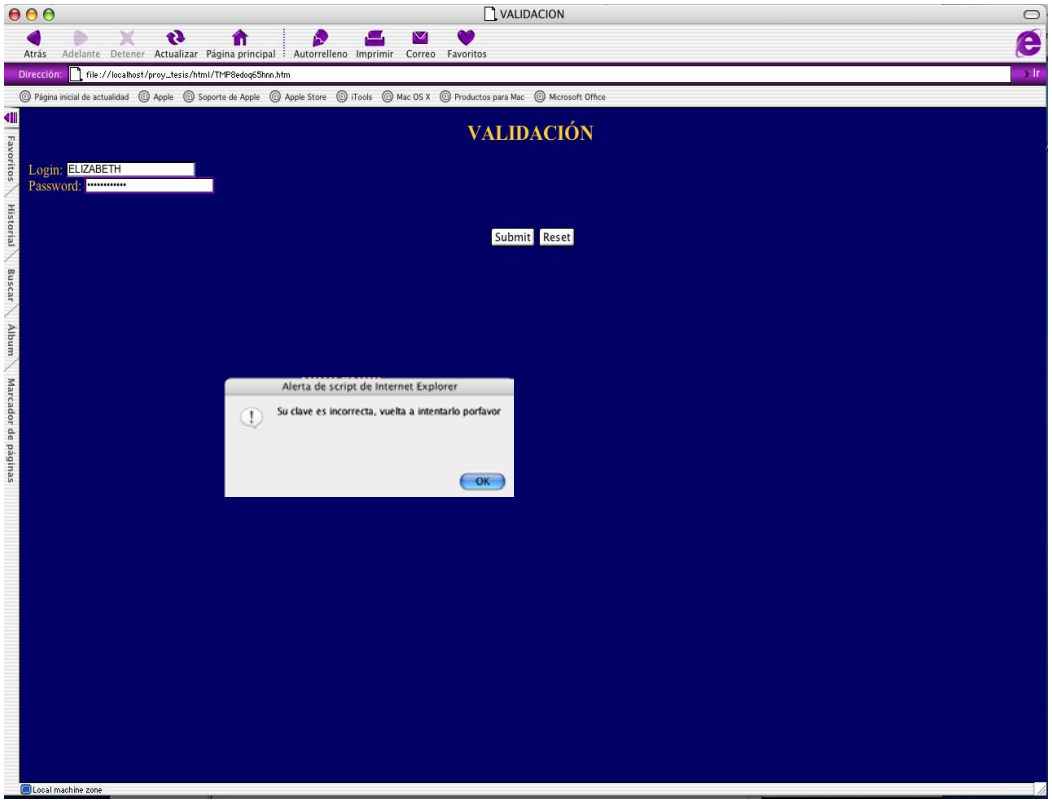
El usuario podrá ingresar a ésta aplicación una vez que haya seleccionado en el menú la opción “Hacer Reservación” y haya oprimido el botón “Submit”,después de esto entrará a una aplicación que le preguntará la clave de la escuela a la cual se le desea reservar la visita guiada, la clave tecleada será comparada con las claves de las escuelas ya registradas por lo que si la clave tecleada no se encuentra registrada, la aplicación no permitirá realizar la reservación y volverá a pedir la clave . Si la clave es correcta entrará a otra aplicación que le mostrará las fechas disponibles para realizar una visita guiada aquí la persona que representa a la institución tendrá la opción de escoger de entre las fechas mostradas y la hora. Una vez seleccionada la fecha de la visita automáticamente se generará la clave de la reservación y posteriormente se le preguntará si está seguro de querer esa fecha, si no lo está se le volverán a mostrar las fechas disponibles, en caso de seleccionar otra fecha se generará automáticamente otra clave de reservación, y en caso de estar seguro de querer esa fecha se dará de baja dicha fecha y cambiará de estado a no disponible. Una vez realizado esto se le mostrará una pantalla con la fecha, hora y clave de reservación de la visita guiada, y terminará la aplicación.

Requerimientos: Consulta

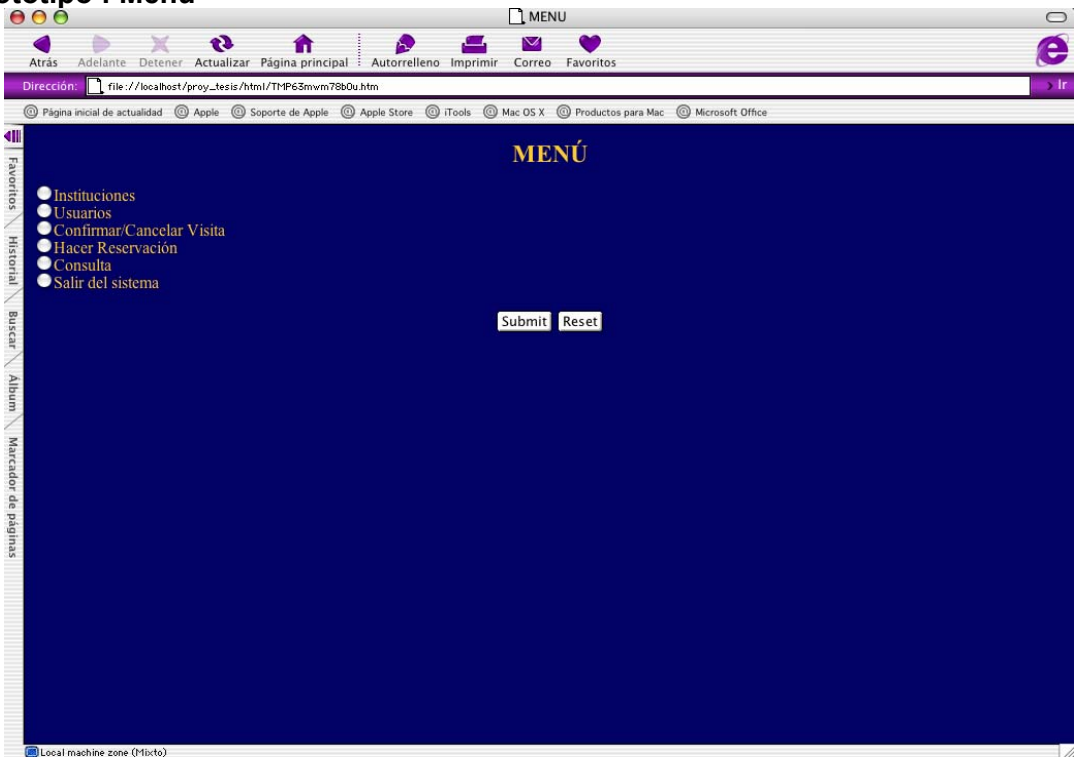
Una vez que el usuario ingresó al menú y seleccionó la opción “Consulta”, entrará a ésta aplicación donde en caso de tener password de administrador, podrá realizar consultas por : clave de institución, nombre de institución, fecha de reservación, mes de registro, mes de reservación, año de registro, todas las instituciones, todos los usuarios. Pero en el caso de no tener password de administrador sólo podrá realizar consultas por : clave de institución, nombre de institución y fecha de reservación.
Una vez tecleada el tipo de consulta con su respectivo dato, dicho dato será comparado con los datos ya registrados, en caso de no existir el dato tecleado al usuario se le pedirá que vuelva a realizar la consulta, pero en caso de que el dato si exista se le mostrará(n) el (los) registro(s) de la(s) institución(es) deseada Después de que se le ha mostrado el registro de la institución se le preguntará si desea imprimir el registro, si responde “Si” y oprime el botón “Submit” se imprimirá automáticamente el registro y regresará al menú original. En caso contrario, regresará automáticamente al menú principal.

Prototipo : Validación de usuario

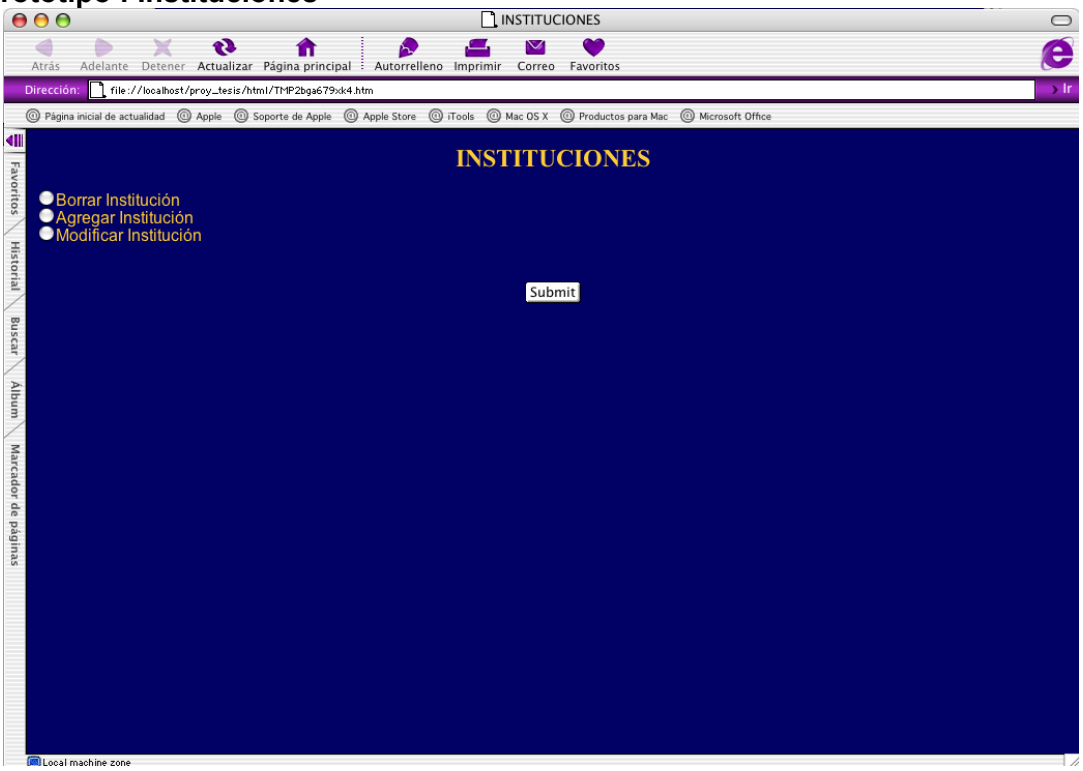




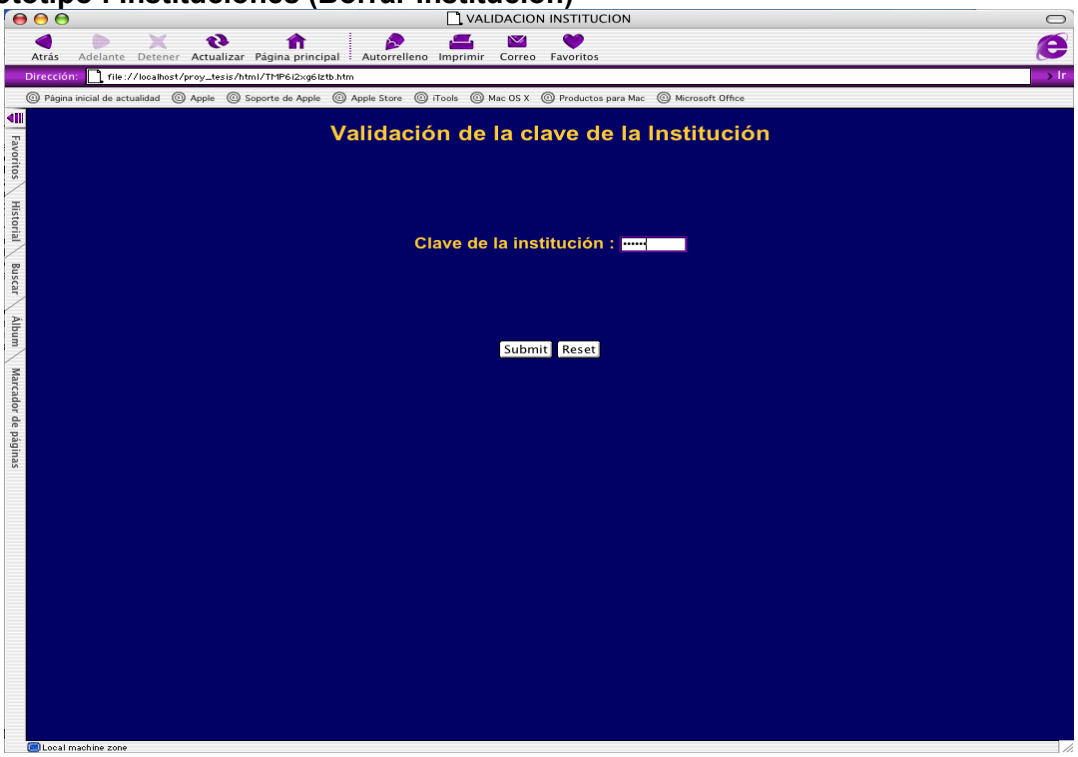
Prototipo : Menú

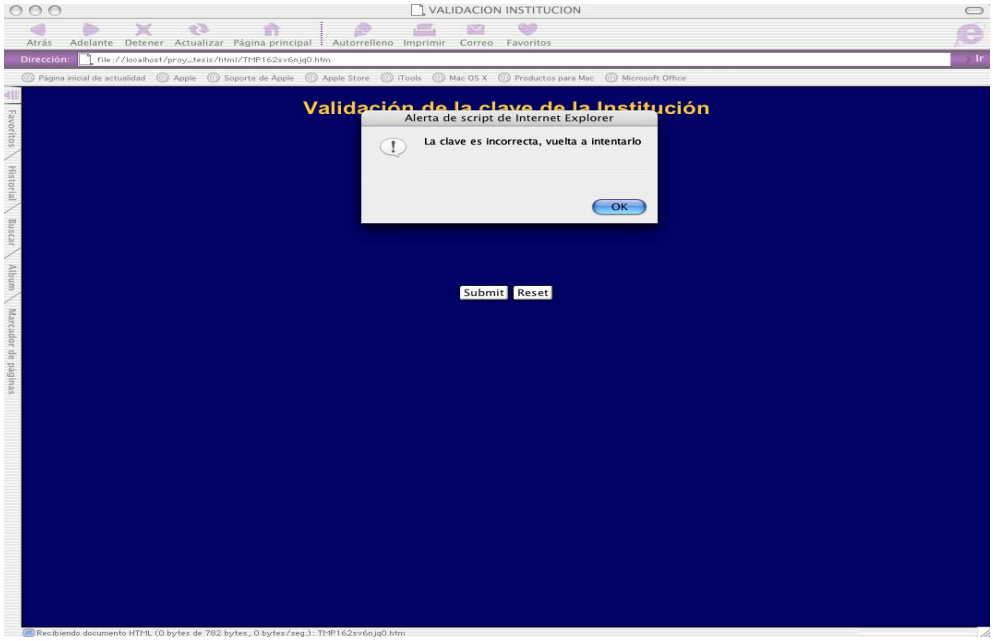


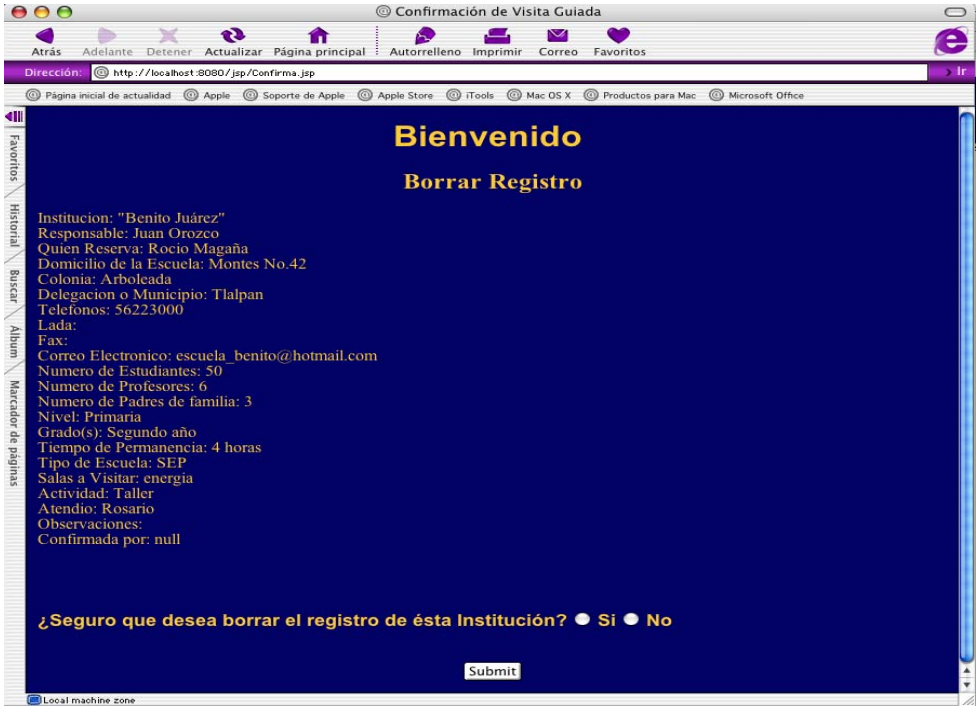
Prototipo : Instituciones



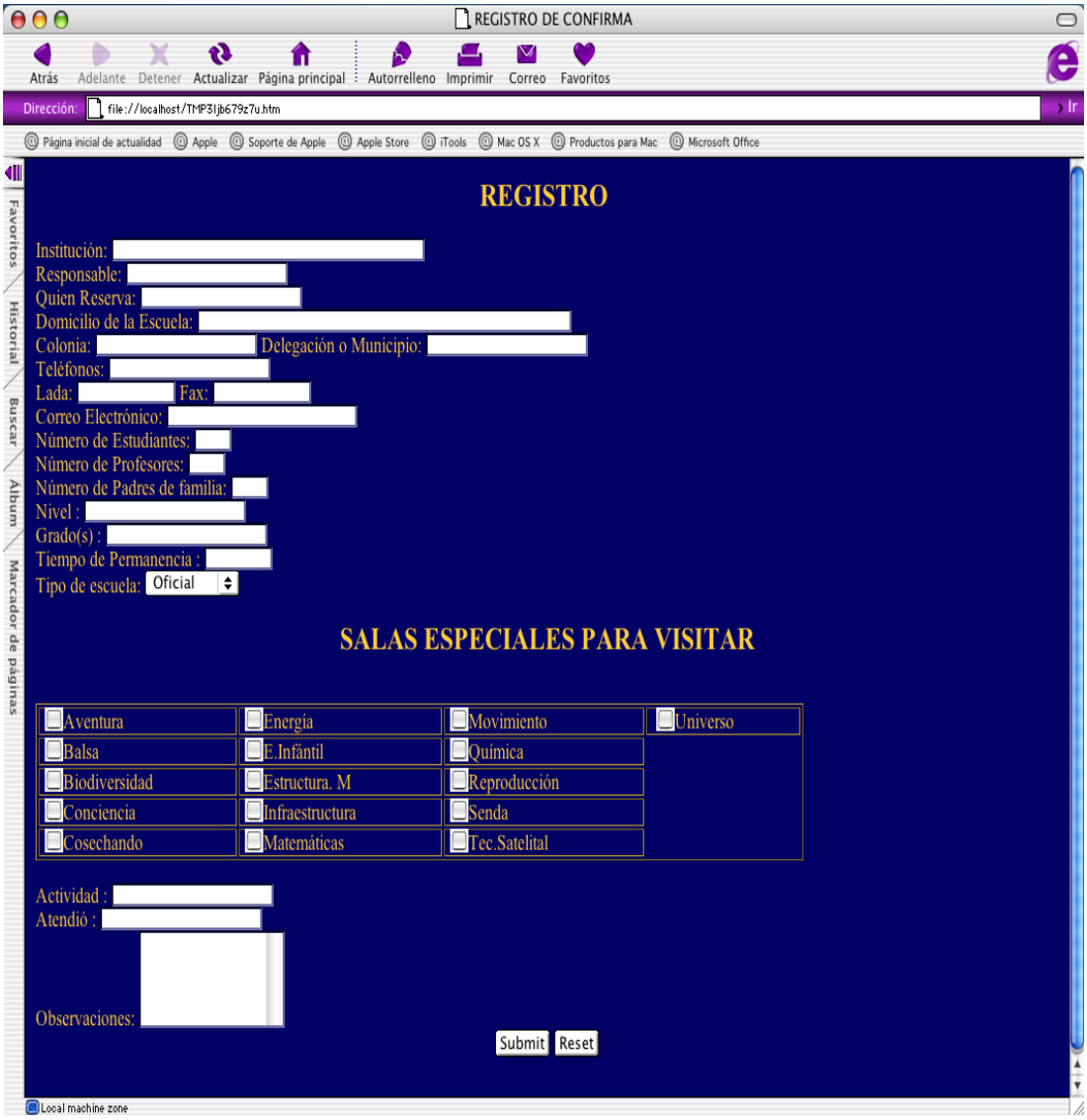
Prototipo : Instituciones (Borrar Institución)

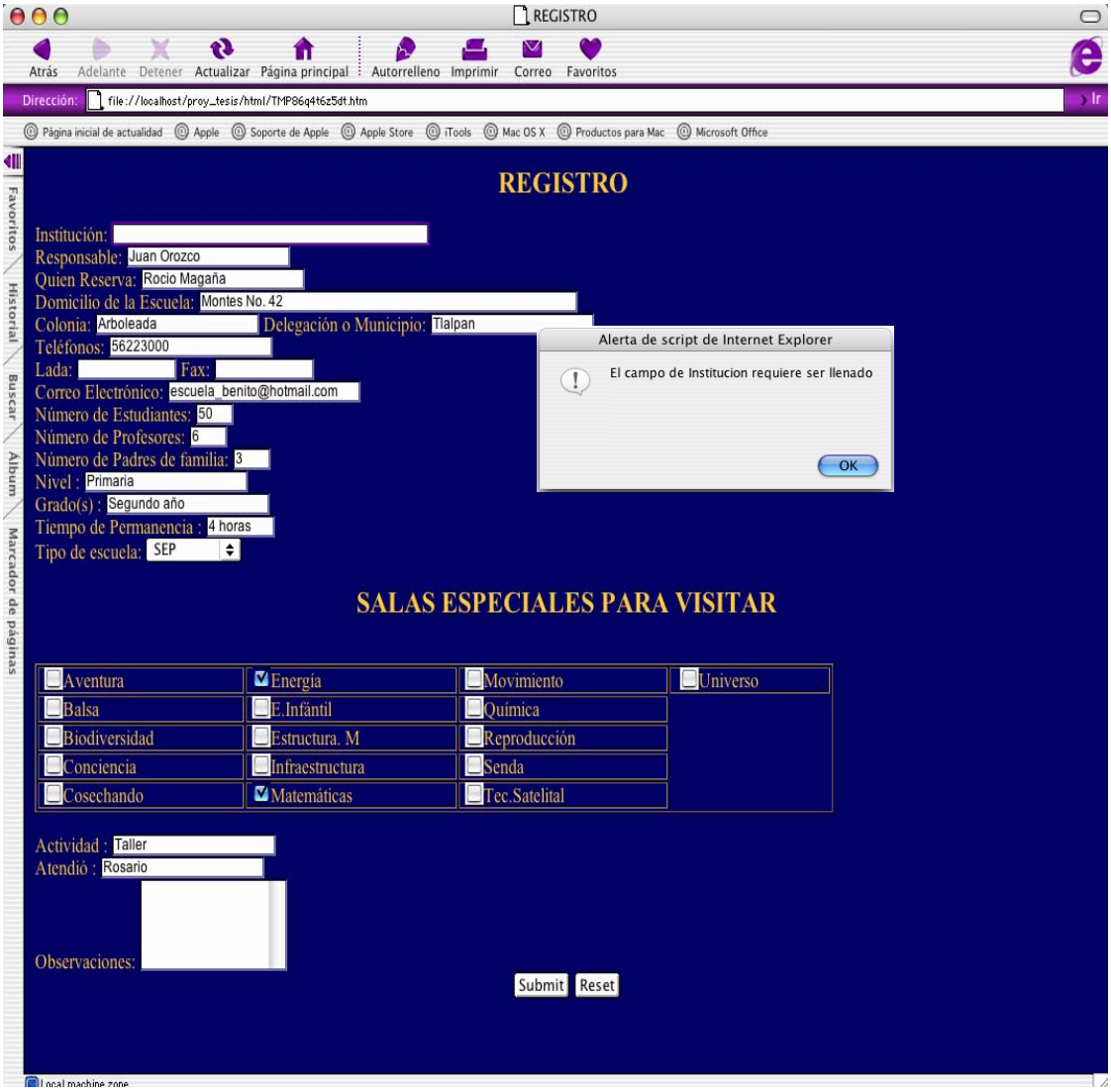


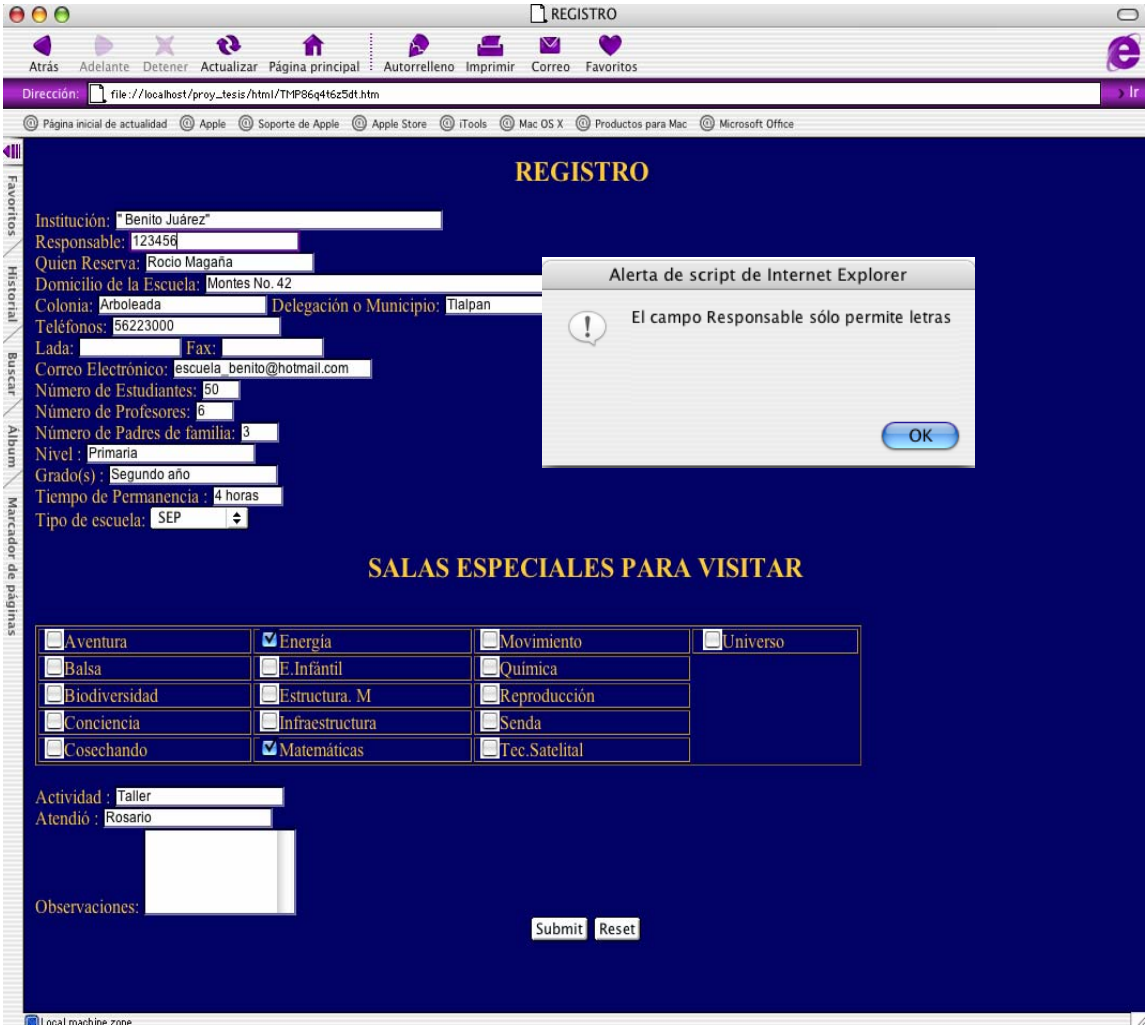

VALIDACION INSTITUCION
Atrás Adelante Detener Actualizar Página principal Autorrelleno Imprimir Correo Favoritos
Dirección: file://localhost/proy_tesis/html/TMP162sv6njq0.htm
Ir
Página inicial de actualidad Apple Soporte de Apple Apple Store iTools Mac OS X Productos para Mac Microsoft Office
Favoritos Historial Buscar Album Marcador de páginas
Validación de la clave de la Institución
Alerta de script de Internet Explorer
La clave es incorrecta, vuelta a intentarlo
OK
Submit Reset
Recibiendo documento HTML (0 bytes de 782 bytes, 0 bytes/seg.): TMP162sv6njq0.htm

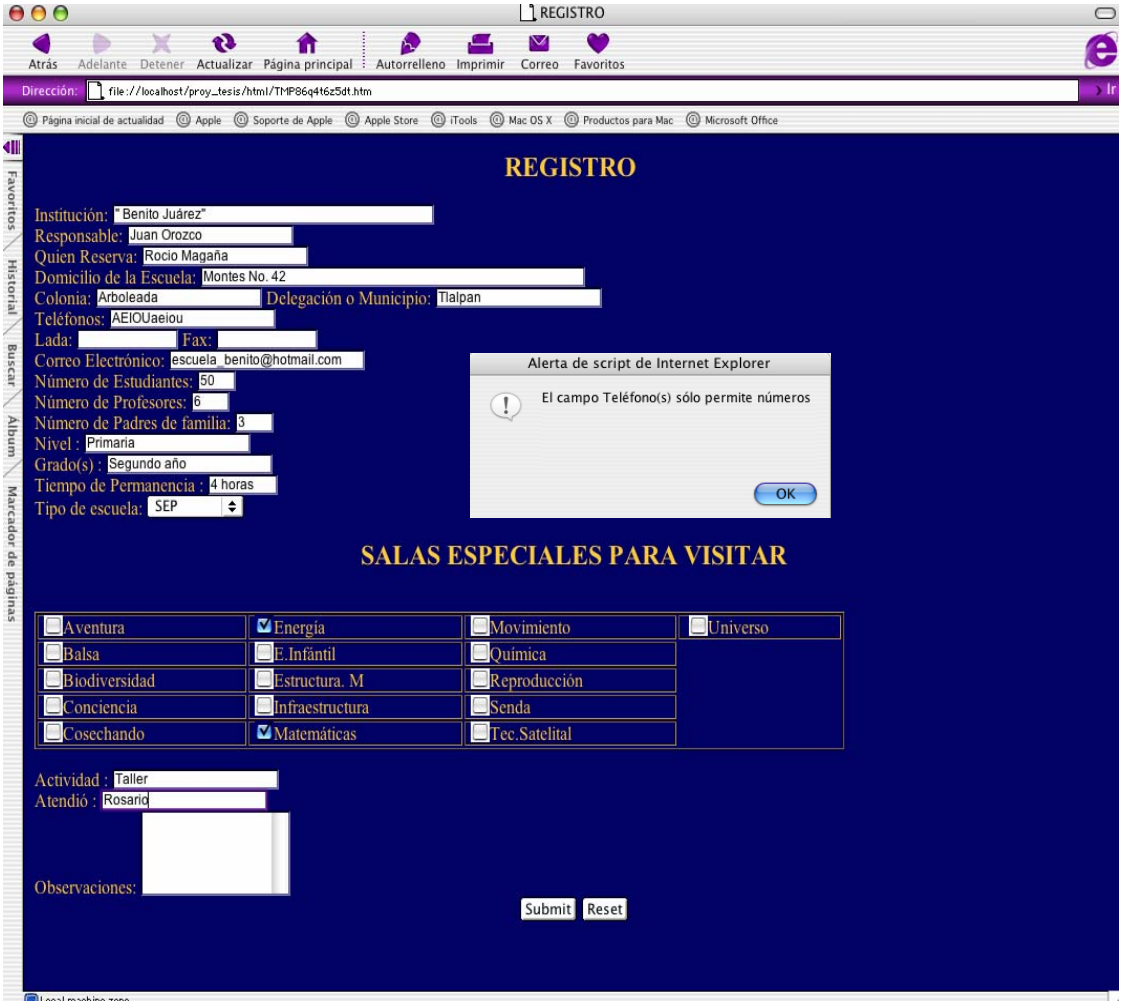

Confirmación de Visita Guiada
Atrás Adelante Detener Actualizar Página principal Autorrelleno Imprimir Correo Favoritos
Dirección: http://localhost:8080/jsp/Confirma.jsp
Ir
Página inicial de actualidad Apple Soporte de Apple Apple Store iTools Mac OS X Productos para Mac Microsoft Office
Favoritos Historial Buscar Album Marcador de páginas
Bienvenido
Borrar Registro
Institucion: "Benito Juárez"
Responsable: Juan Orozco
Quien Reserva: Rocio Magaña
Domicilio de la Escuela: Montes No.42
Colonia: Arboleada
Delegacion o Municipio: Tlalpan
Telefonos: 56223000
Lada:
Fax:
Correo Electronico: escuela_benito@hotmail.com
Numero de Estudiantes: 50
Numero de Profesores: 6
Numero de Padres de familia: 3
Nivel: Primaria
Grado(s): Segundo año
Tiempo de Permanencia: 4 horas
Tipo de Escuela: SEP
Salas a Visitar: energia
Actividad: Taller
Atendio: Rosario
Observaciones:
Confirmada por: null
¿Seguro que desea borrar el registro de ésta Institución? Si No
Submit
Local machine zone

Prototipo : Instituciones (Agregar Institución)

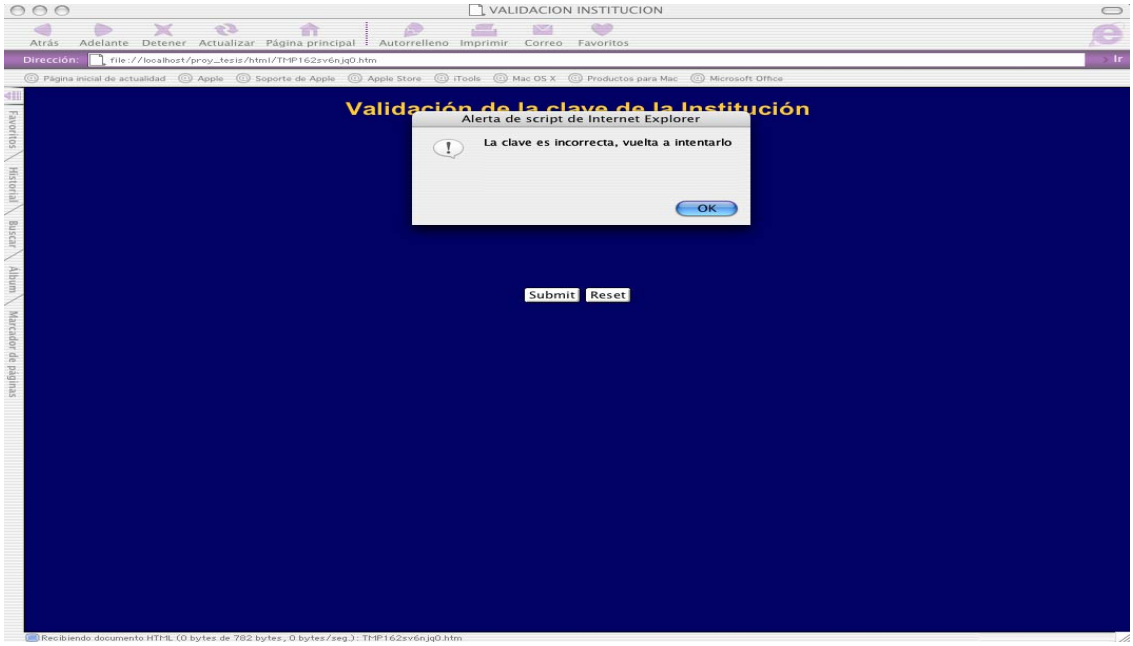


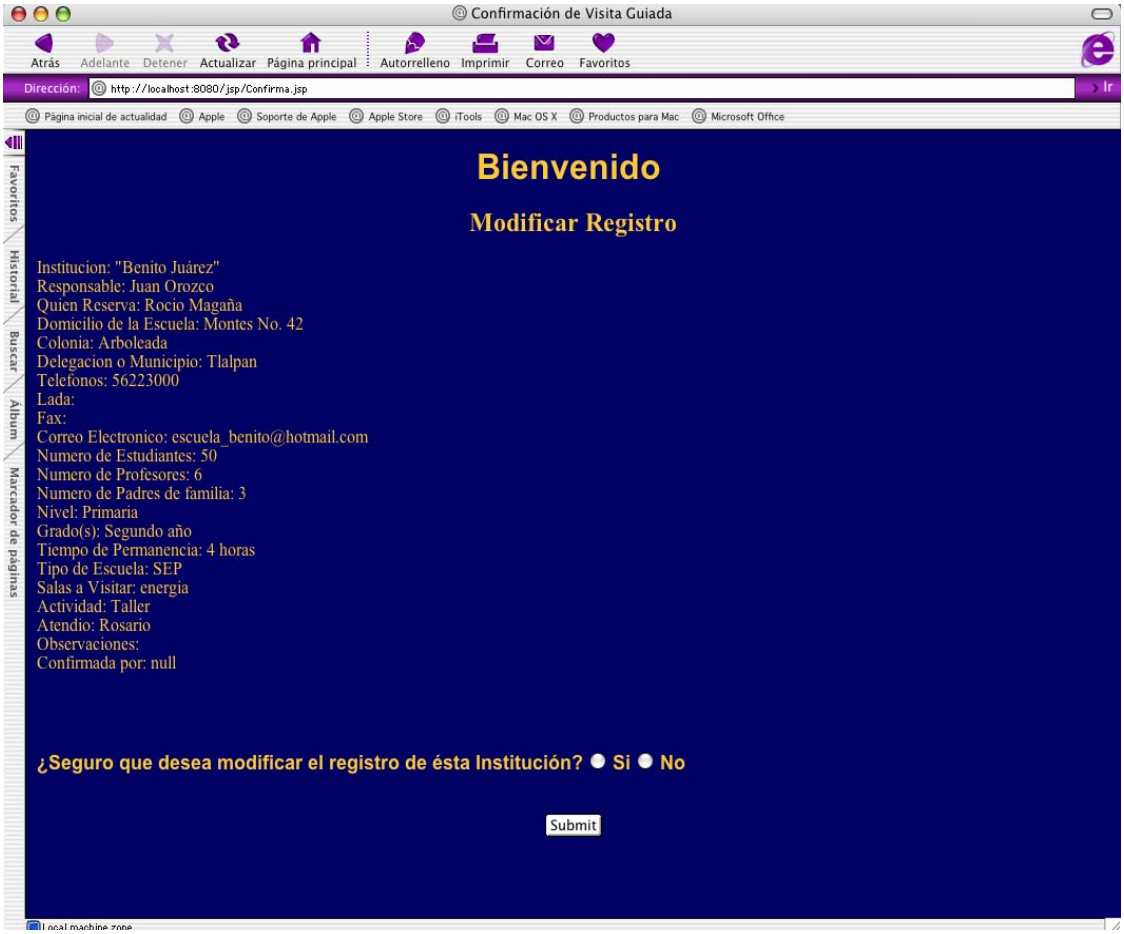

REGISTRO
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/proy_tesis/html/TMP86q4t6z5dt.htm
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
REGISTRO
Institución:
Responsable: Juan Orozco
Quien Reserva: Rocio Magaña
Domicilio de la Escuela: Montes No. 42
Colonia: Arboleada
Delegación o Municipio: Tlalpan
Teléfonos: 56223000
Lada:
Fax:
Correo Electrónico: escuela_benito@hotmail.com
Número de Estudiantes: 50
Número de Profesores: 6
Número de Padres de familia: 3
Nivel : Primaria
Grado(s) : Segundo año
Tiempo de Permanencia : 4 horas
Tipo de escuela: SEP
Alerta de script de Internet Explorer
El campo de Institucion requiere ser llenado
OK
SALAS ESPECIALES PARA VISITAR
Aventura
Energía
Movimiento
Universo
Balsa
E.Infántil
Química
Biodiversidad
Estructura. M
Reproducción
Conciencia
Infraestructura
Senda
Cosechando
Matemáticas
Tec.Satelital
Actividad : Taller
Atendió : Rosario
Observaciones:
Submit
Reset
Local machine zone

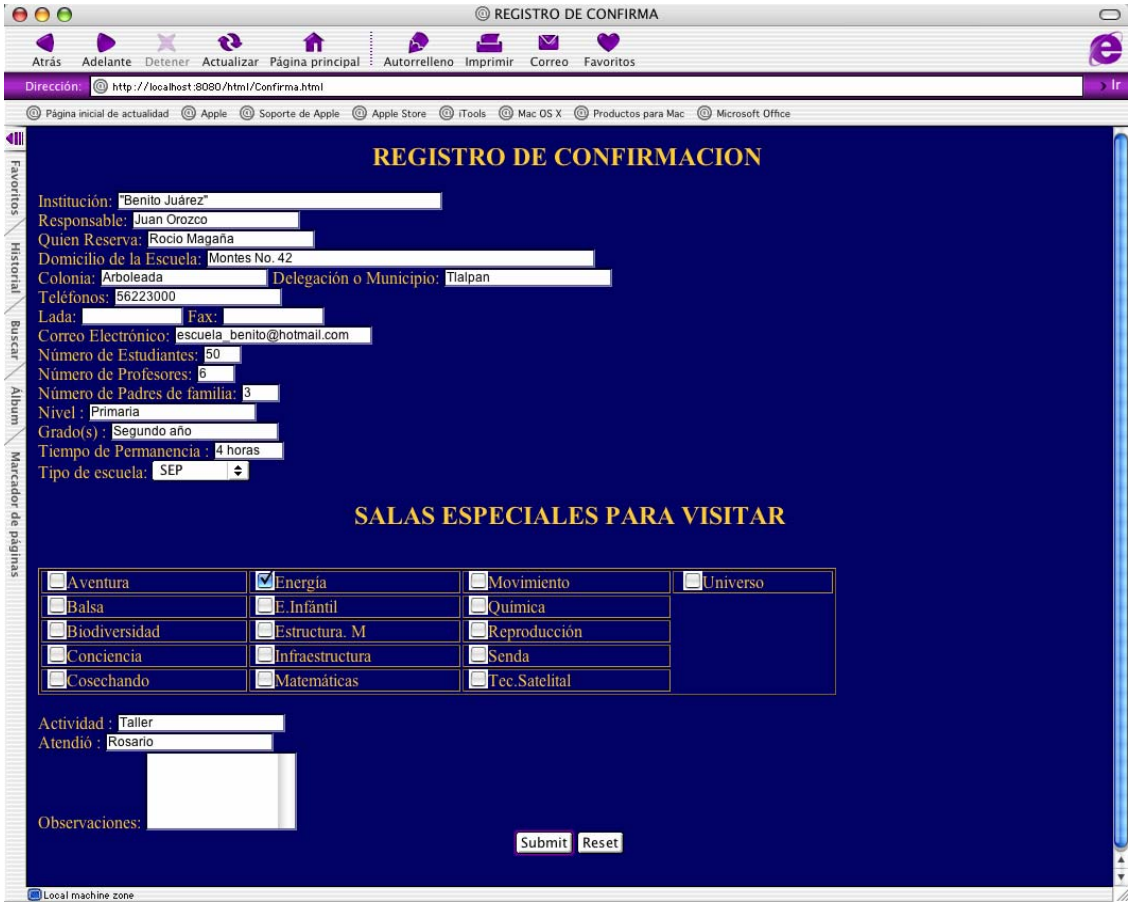

REGISTRO
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/proy_tesis/html/TMP86q4t6z5dt.htm
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
REGISTRO
Institución: " Benito Juárez"
Responsable: 123456
Quien Reserva: Rocio Magaña
Domicilio de la Escuela: Montes No. 42
Colonia: Arboleada
Delegación o Municipio: Tlalpan
Teléfonos: 56223000
Lada:
Fax:
Correo Electrónico: escuela_benito@hotmail.com
Número de Estudiantes: 50
Número de Profesores: 6
Número de Padres de familia: 3
Nivel : Primaria
Grado(s) : Segundo año
Tiempo de Permanencia : 4 horas
Tipo de escuela: SEP
Alerta de script de Internet Explorer
El campo Responsable sólo permite letras
OK
SALAS ESPECIALES PARA VISITAR
Aventura
Energía
Movimiento
Universo
Balsa
E.Infántil
Química
Biodiversidad
Estructura. M
Reproducción
Conciencia
Infraestructura
Senda
Cosechando
Matemáticas
Tec.Satelital
Actividad : Taller
Atendió : Rosario
Observaciones:
Submit
Reset
Local machine zone



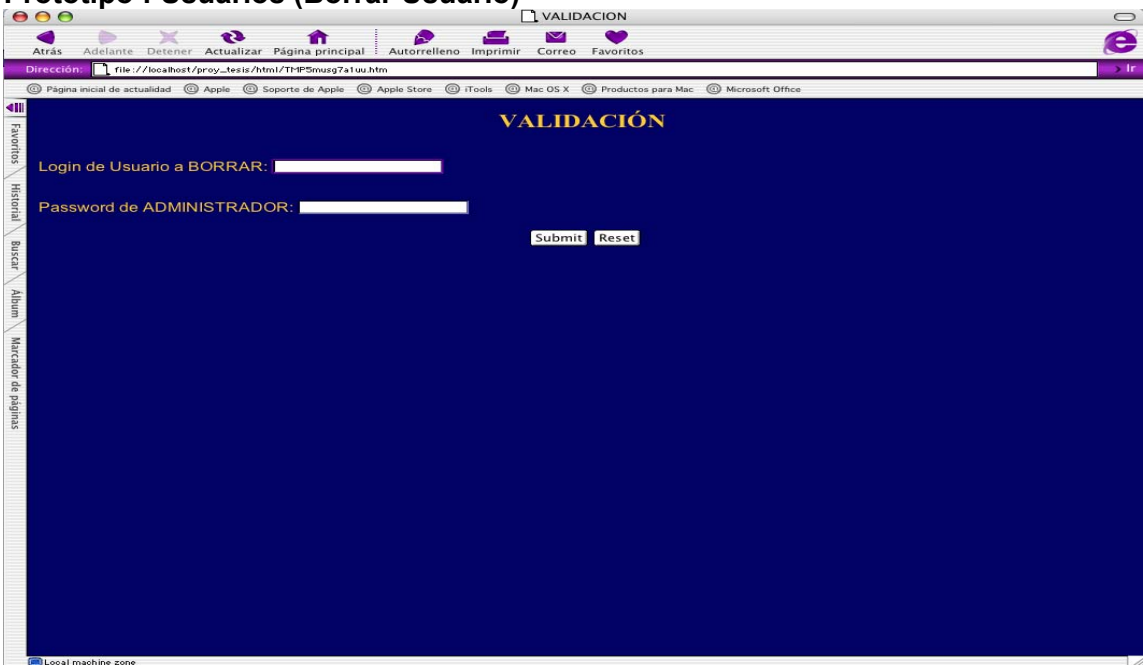
Prototipo : Instituciones (Modificar Institución)

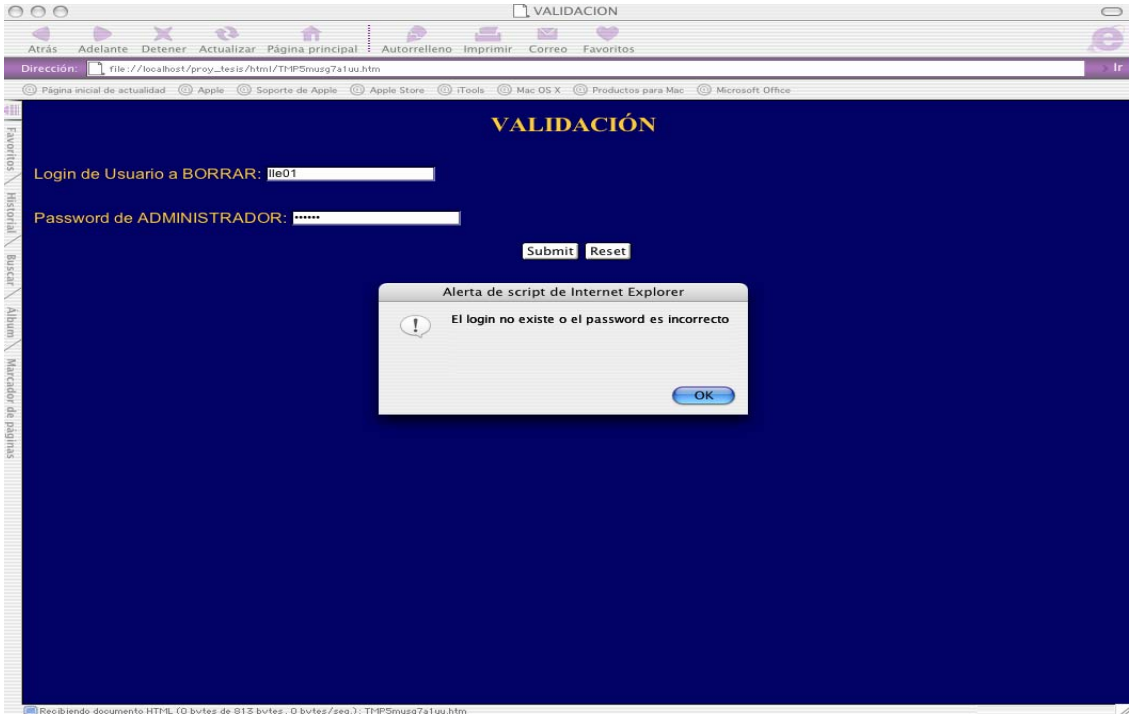

VALIDACION INSTITUCION
Validación de la clave de la Institución
Alerta de script de Internet Explorer
La clave es incorrecta, vuelta a intentarlo
OK
Submit
Reset

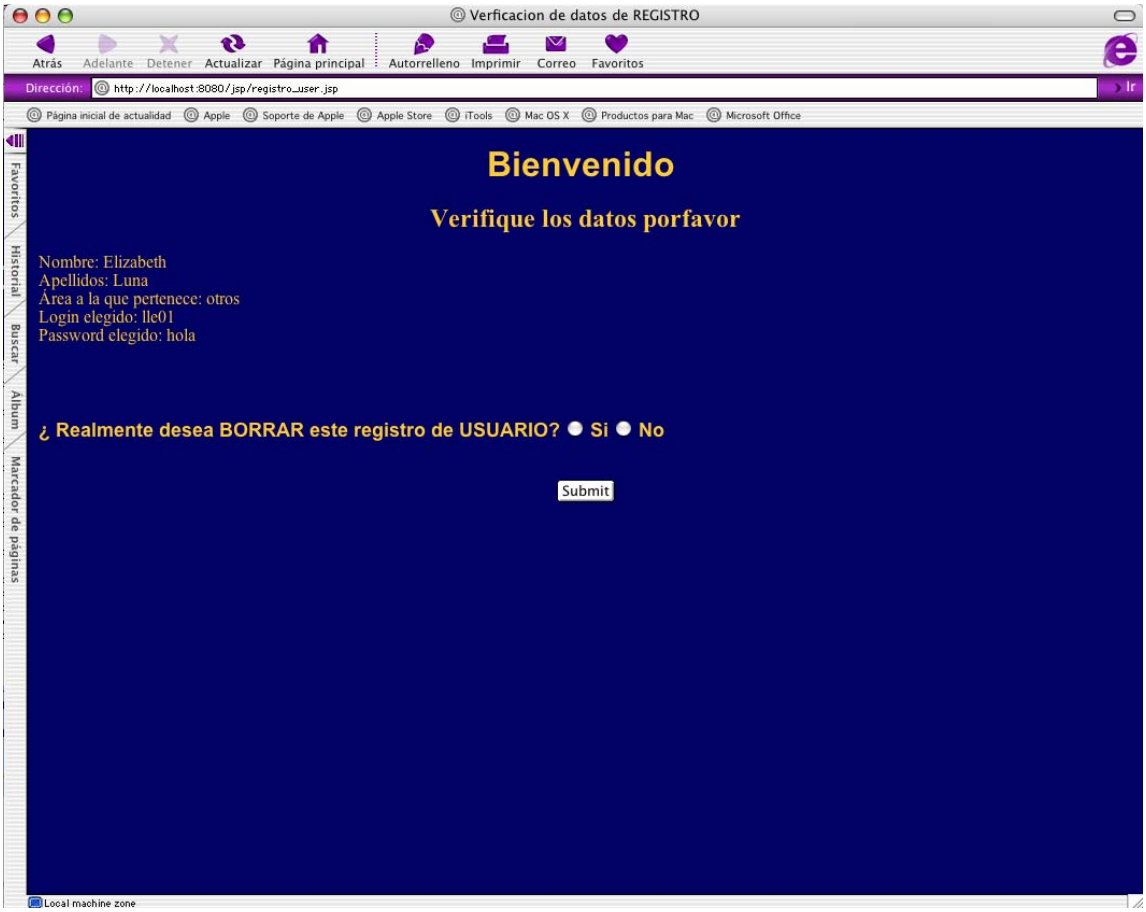

Confirmación de Visita Guiada
Bienvenido
Modificar Registro
Institucion: "Benito Juárez"
Responsable: Juan Orozco
Quien Reserva: Rocio Magaña
Domicilio de la Escuela: Montes No. 42
Colonia: Arboleada
Delegacion o Municipio: Tlalpan
Telefonos: 56223000
Lada:
Fax:
Correo Electronico: escuela_benito@hotmail.com
Numero de Estudiantes: 50
Numero de Profesores: 6
Numero de Padres de familia: 3
Nivel: Primaria
Grado(s): Segundo año
Tiempo de Permanencia: 4 horas
Tipo de Escuela: SEP
Salas a Visitar: energia
Actividad: Taller
Atendio: Rosario
Observaciones:
Confirmada por: null
¿Seguro que desea modificar el registro de ésta Institución? Si No
Submit



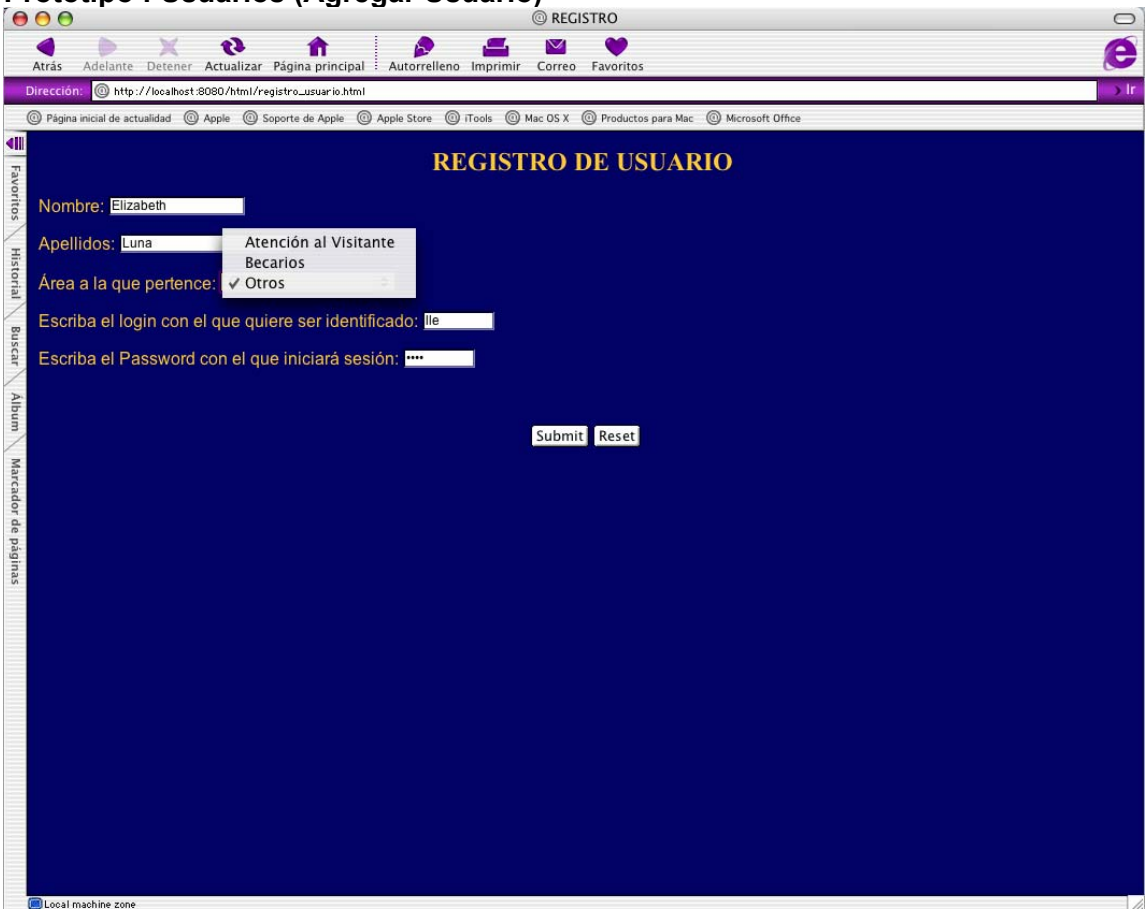
Prototipo : Usuarios (Borrar Usuario)

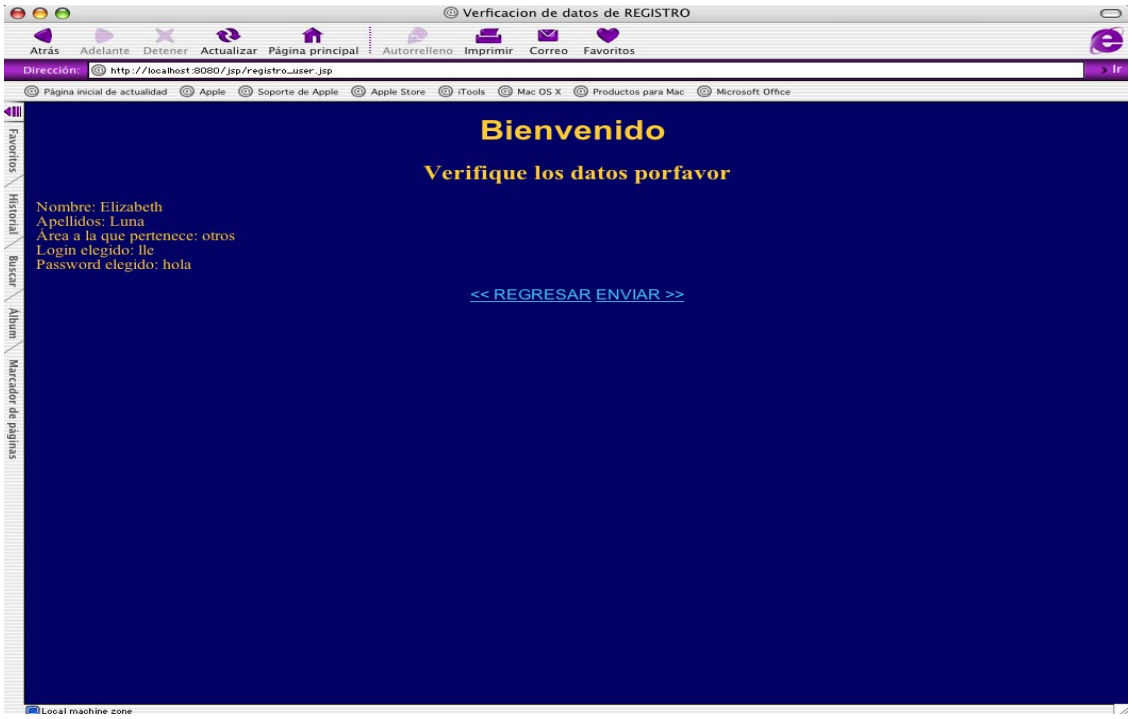



VALIDACION
VALIDACIÓN
Login de Usuario a BORRAR: lle01
Password de ADMINISTRADOR:
Submit
Reset
Alerta de script de Internet Explorer
El login no existe o el password es incorrecto
OK

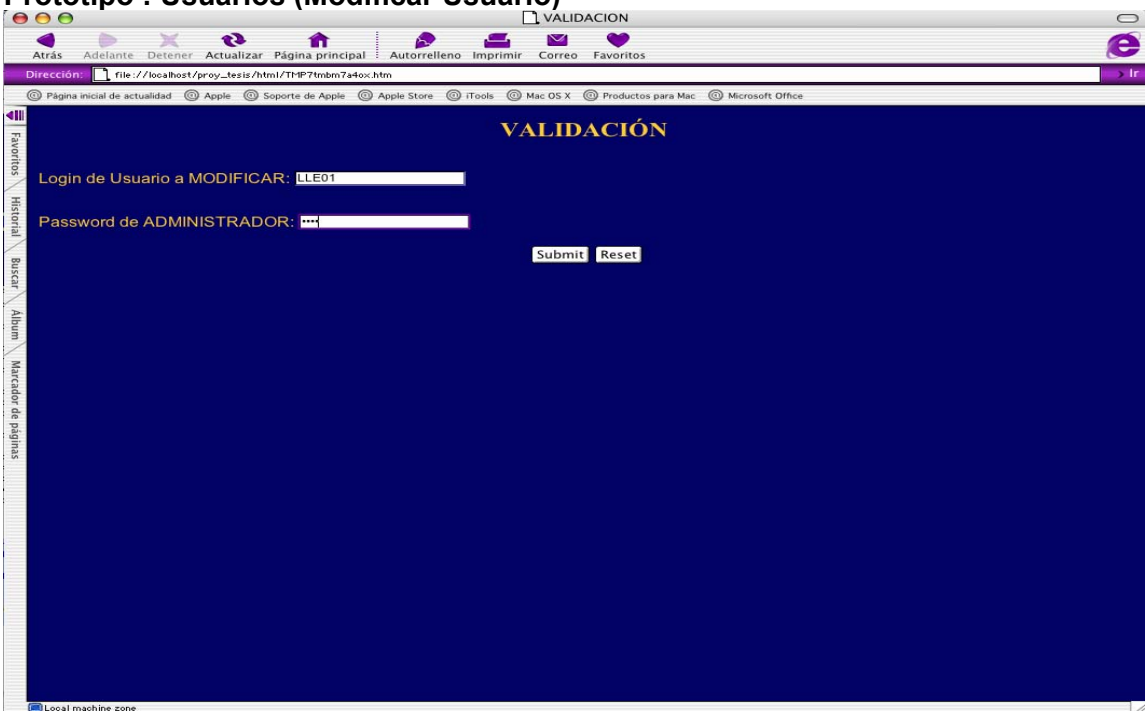

Verficacion de datos de REGISTRO
Bienvenido
Verifique los datos porfavor
Nombre: Elizabeth
Apellidos: Luna
Área a la que pertenece: otros
Login elegido: lle01
Password elegido: hola
¿ Realmente desea BORRAR este registro de USUARIO? Si No
Submit

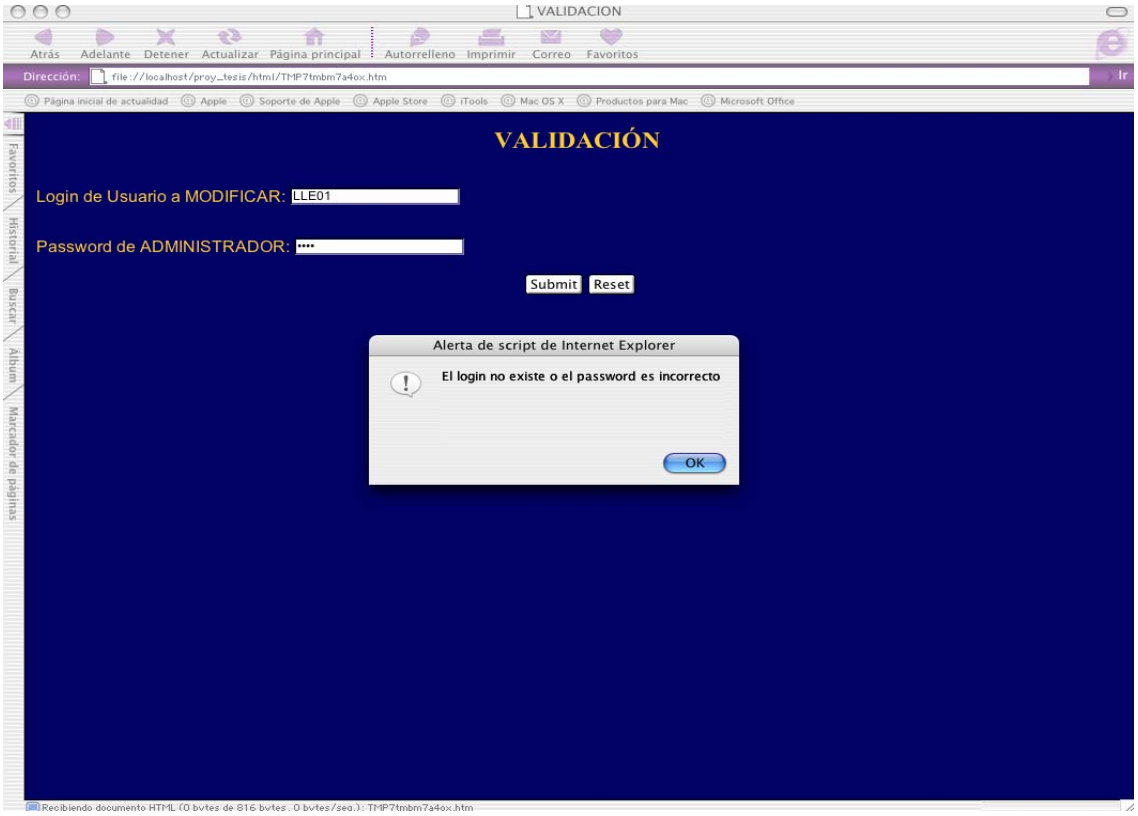
Prototipo : Usuarios (Agregar Usuario)

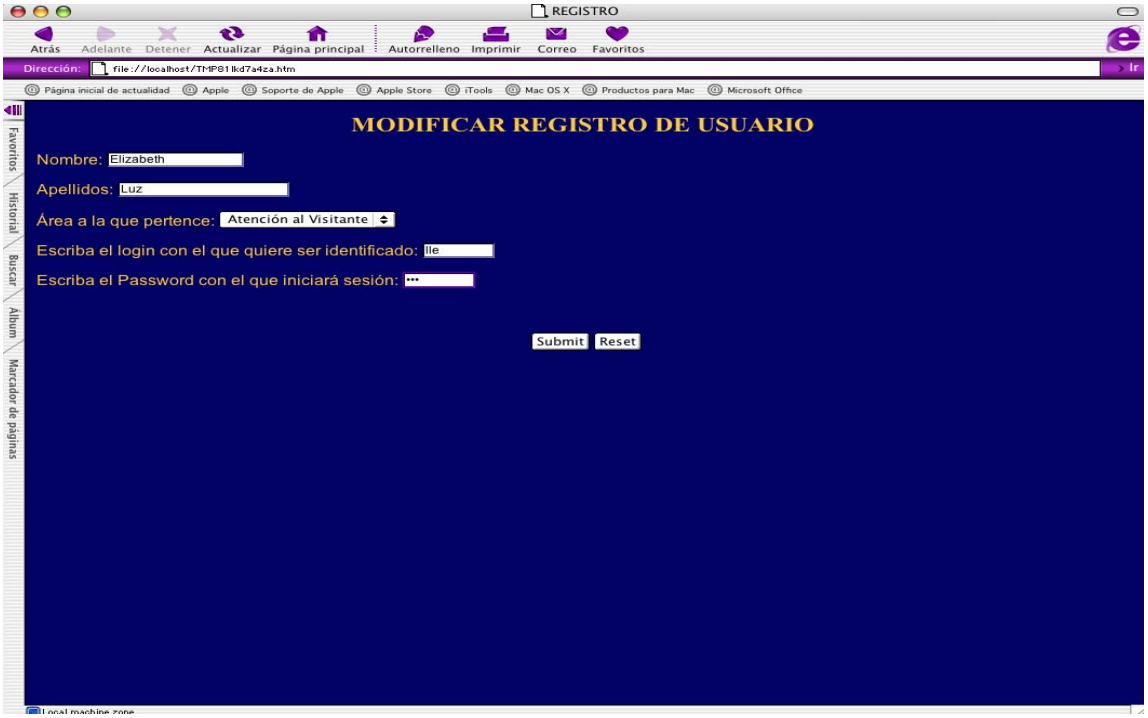


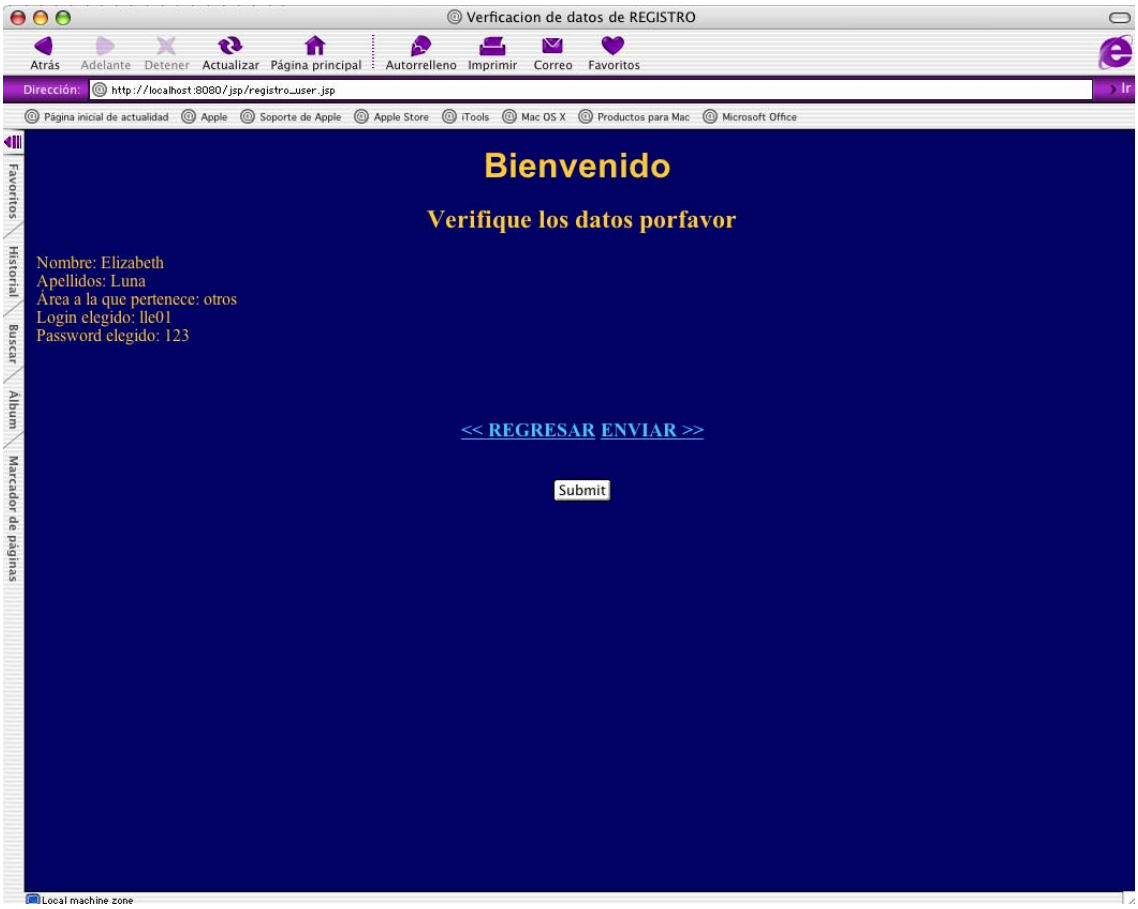


Prototipo : Usuarios (Modificar Usuario)

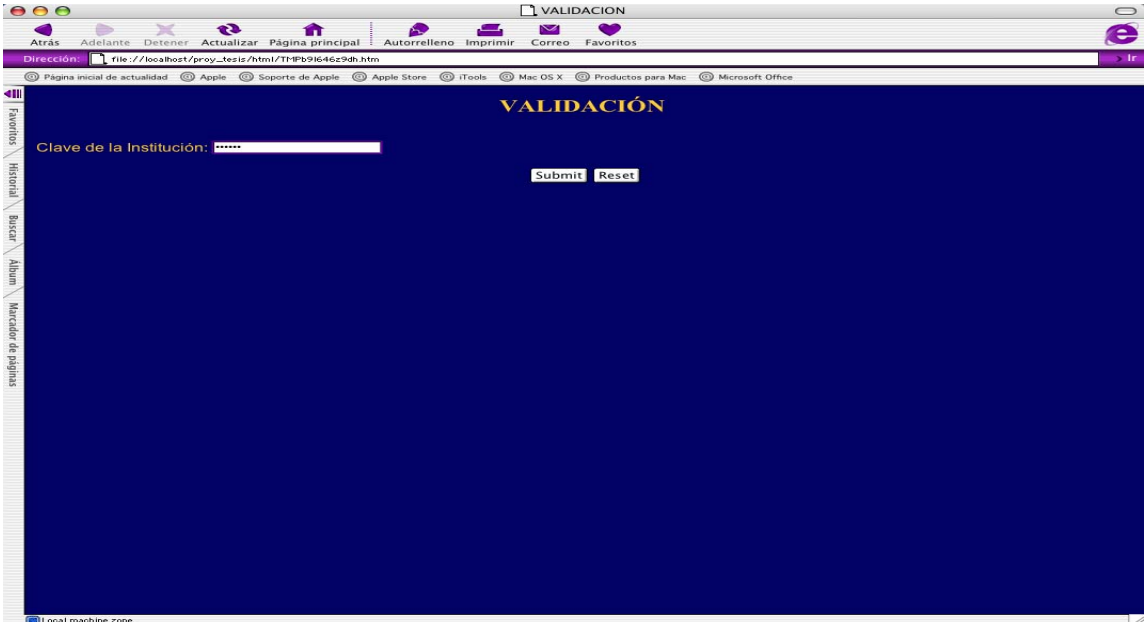


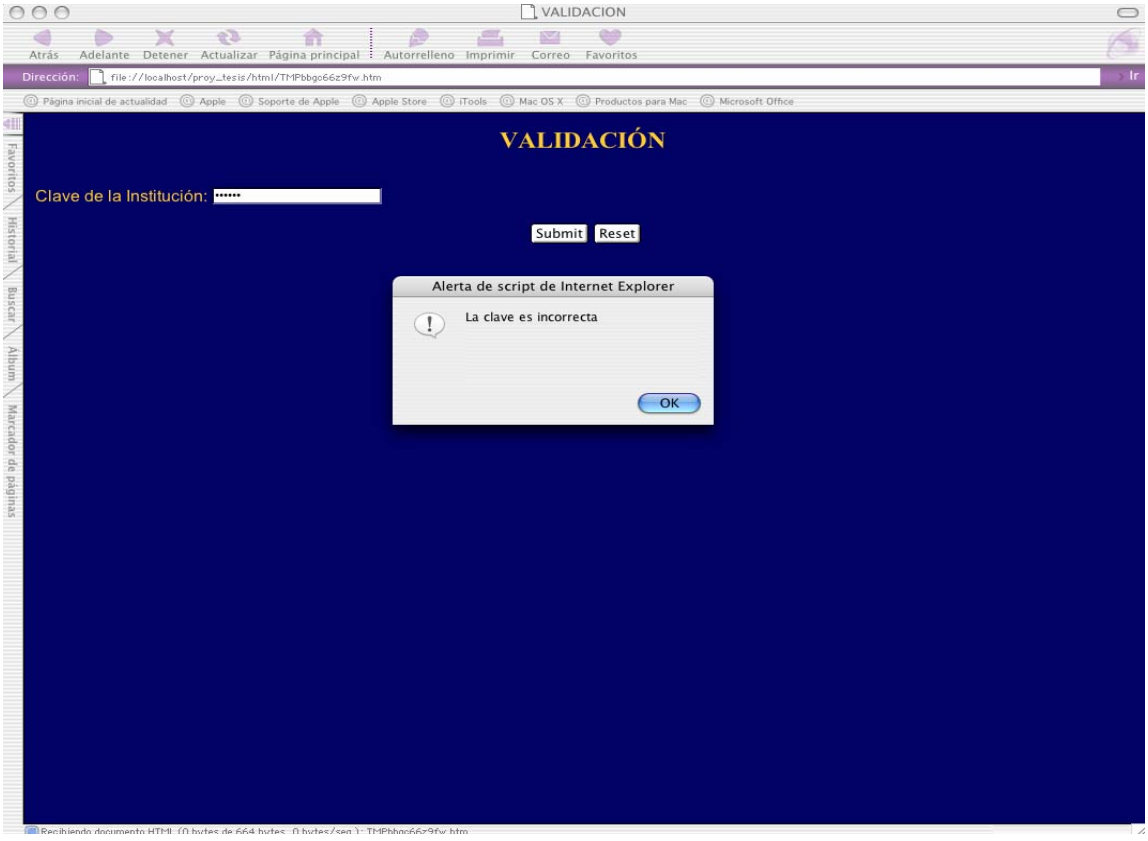


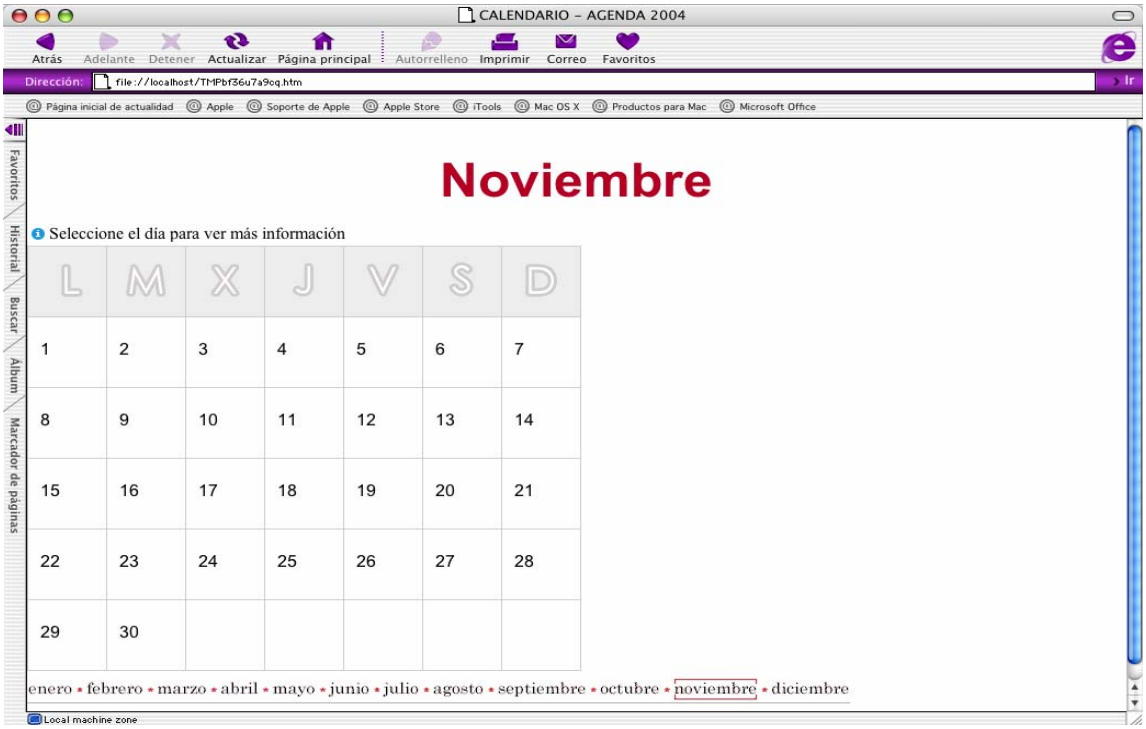

REGISTRO
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/TMP81lkd7a4za.htm
Ir
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
MODIFICAR REGISTRO DE USUARIO
Nombre: Elizabeth
Apellidos: Luz
Área a la que pertence: Atención al Visitante
Escriba el login con el que quiere ser identificado: lle
Escriba el Password con el que iniciará sesión:
Submit
Reset
Local machine zone

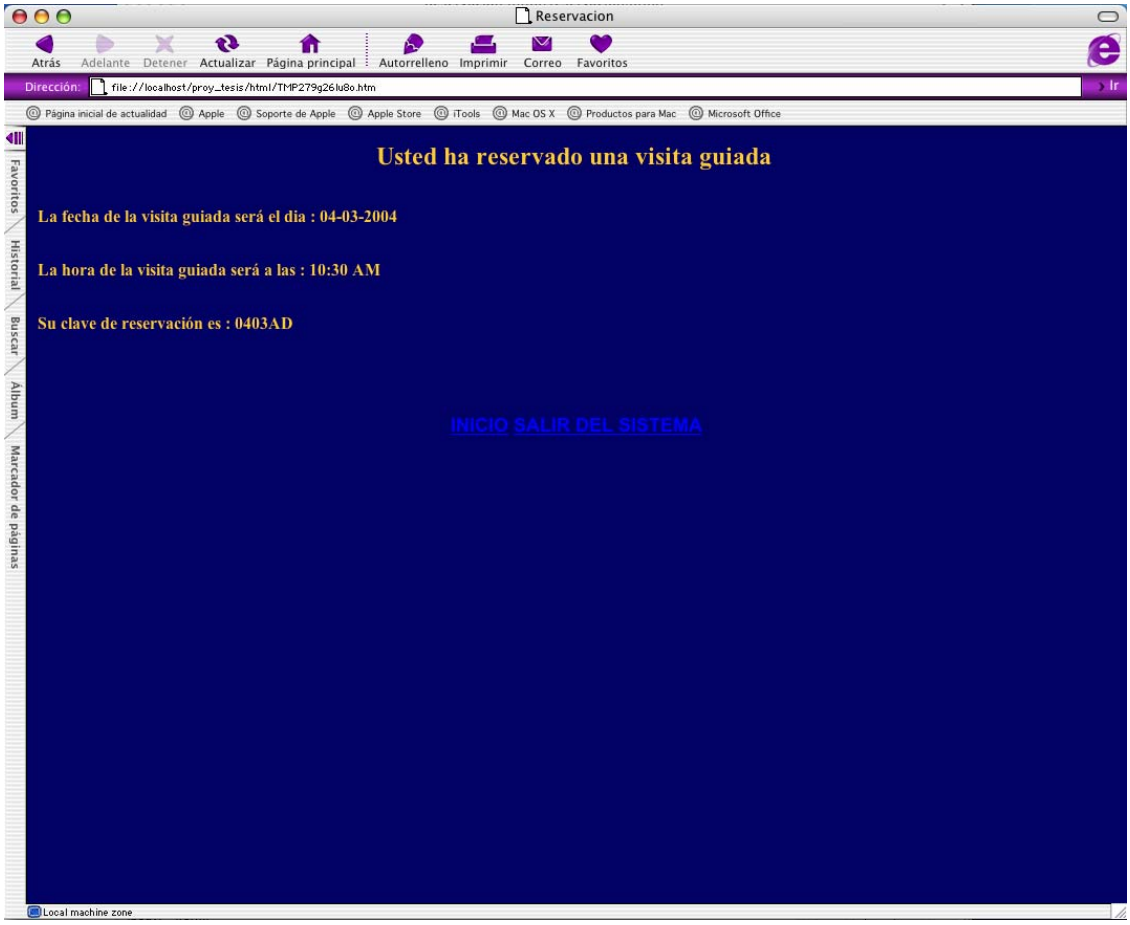

Verficacion de datos de REGISTRO
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
http://localhost:8080/jsp/registro_user.jsp
Ir
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
Bienvenido
Verifique los datos porfavor
Nombre: Elizabeth
Apellidos: Luna
Área a la que pertenece: otros
Login elegido: lle01
Password elegido: 123
<< REGRESAR ENVIAR >>
Submit
Local machine zone

Prototipo : Hacer Reservación

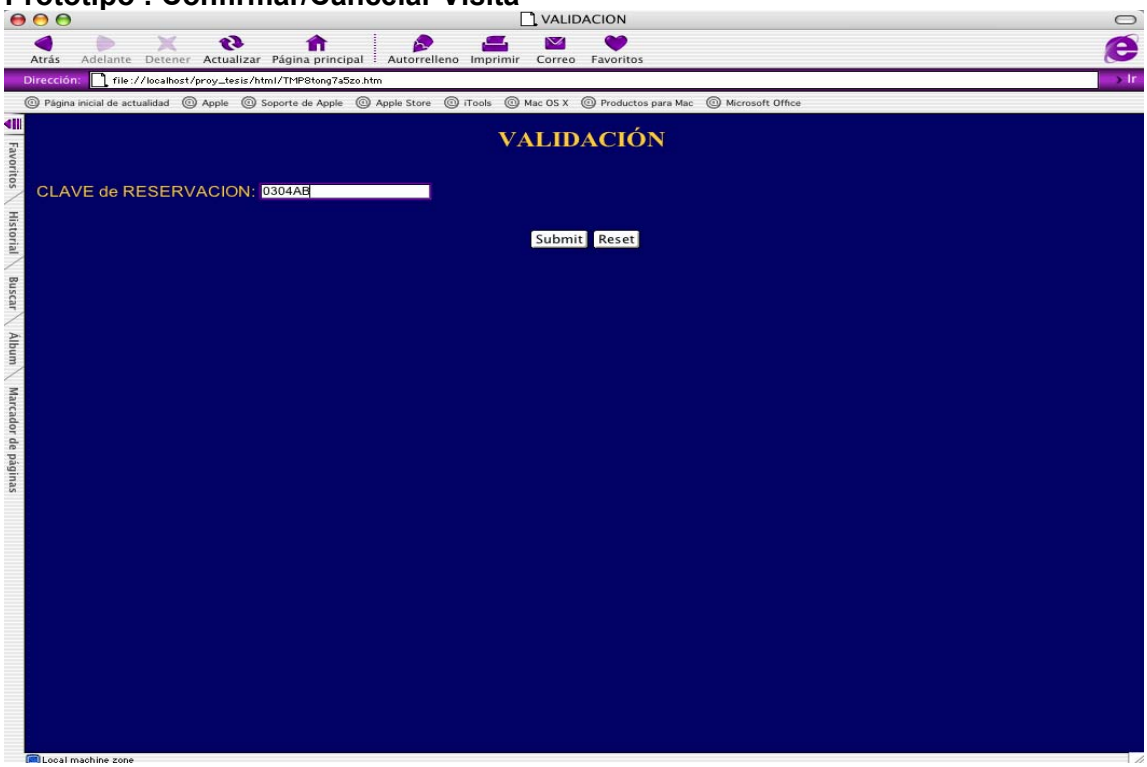


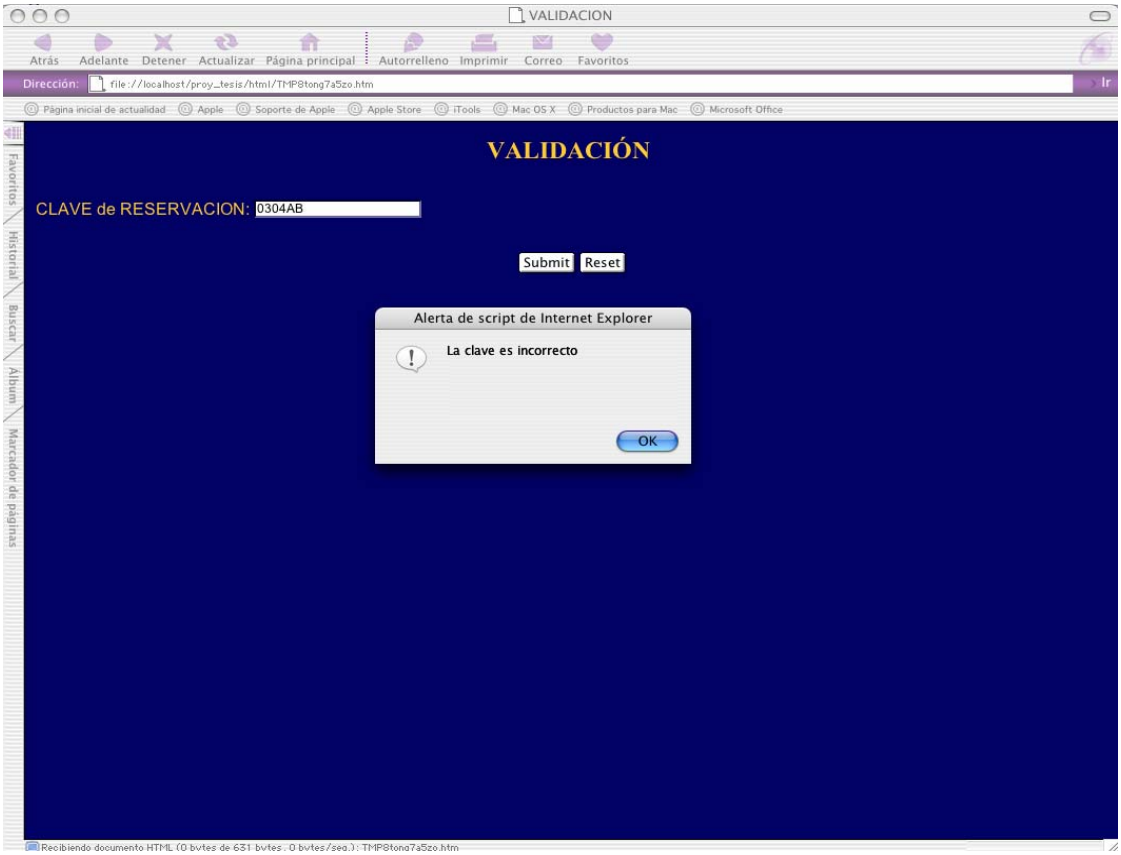


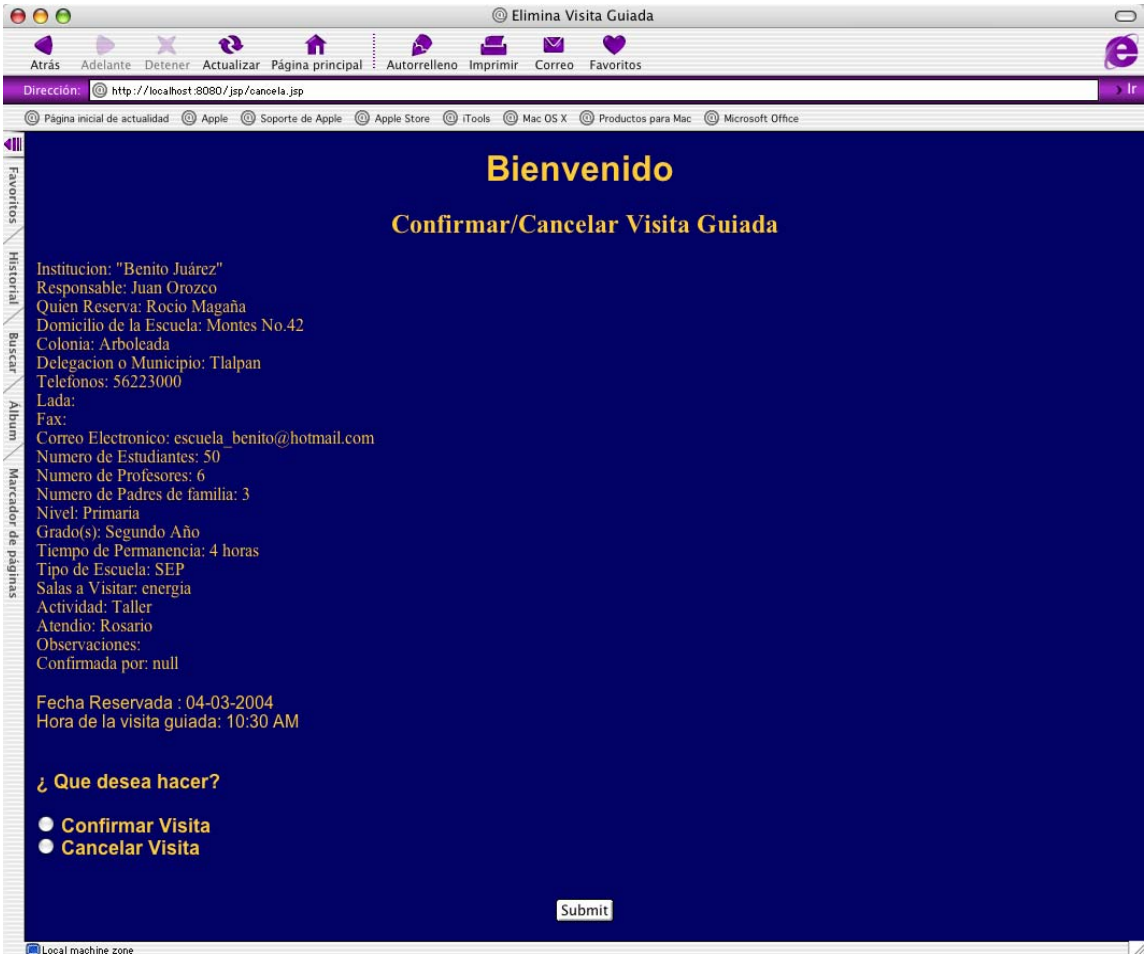

CALENDARIO – AGENDA 2004
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/TMPbf36u7a9oq.htm
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Album
Marcador de páginas
Noviembre
Seleccione el día para ver más información
L M X J V S D
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
enero • febrero • marzo • abril • mayo • junio • julio • agosto • septiembre • octubre • noviembre • diciembre
Local machine zone


Reservacion
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/proy_tesis/html/TMP279g26lu8o.htm
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Album
Marcador de páginas
Usted ha reservado una visita guiada
La fecha de la visita guiada será el dia : 04-03-2004
La hora de la visita guiada será a las : 10:30 AM
Su clave de reservación es : 0403AD
INICIO SALIR DEL SISTEMA
Local machine zone

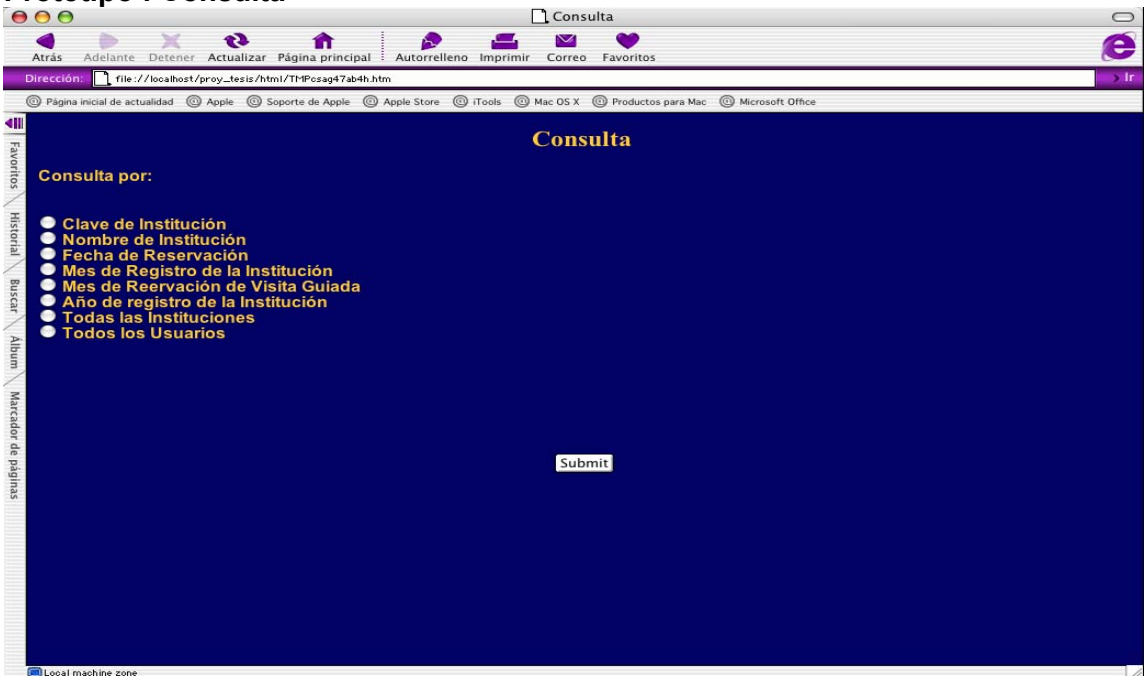
Prototipo : Confirmar/Cancelar Visita

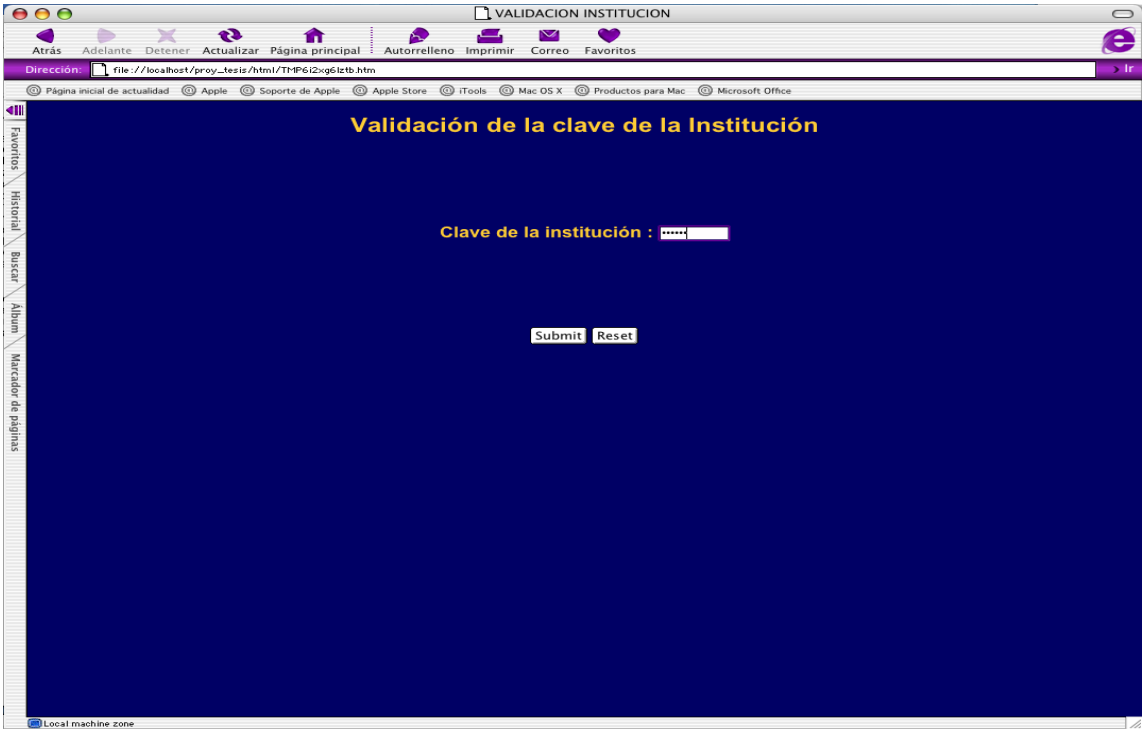


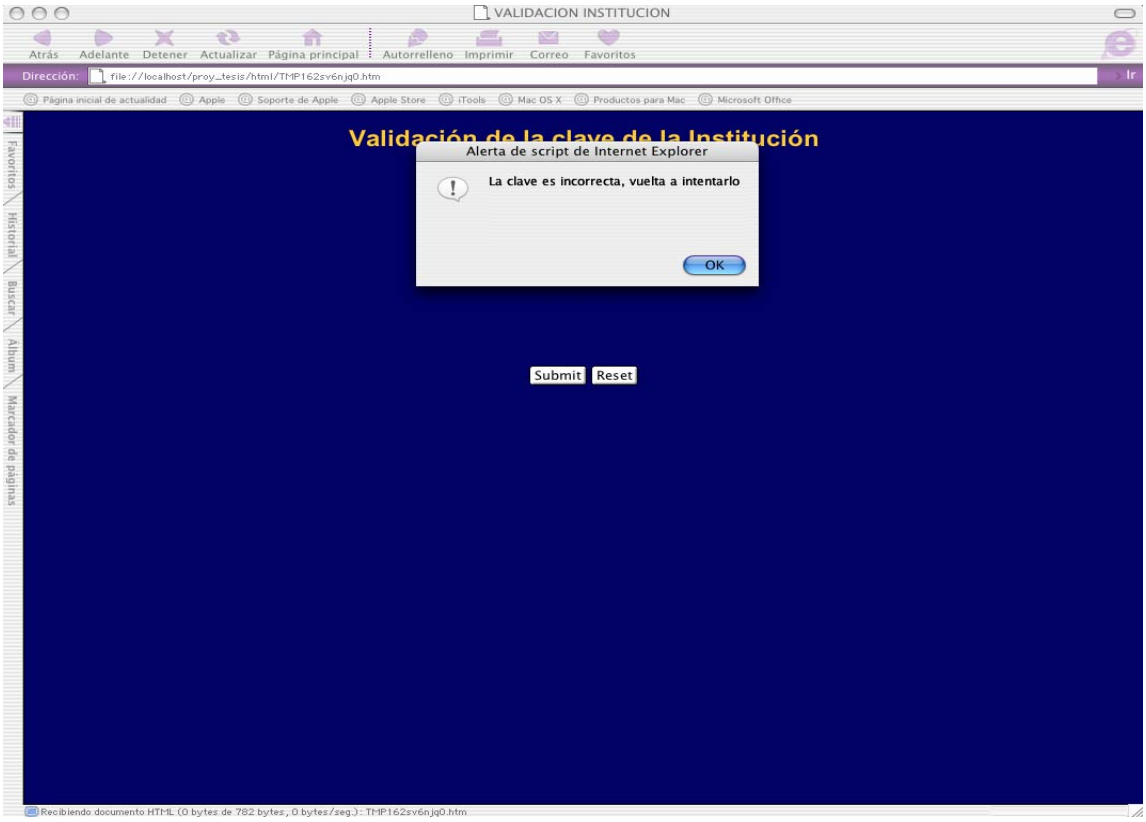


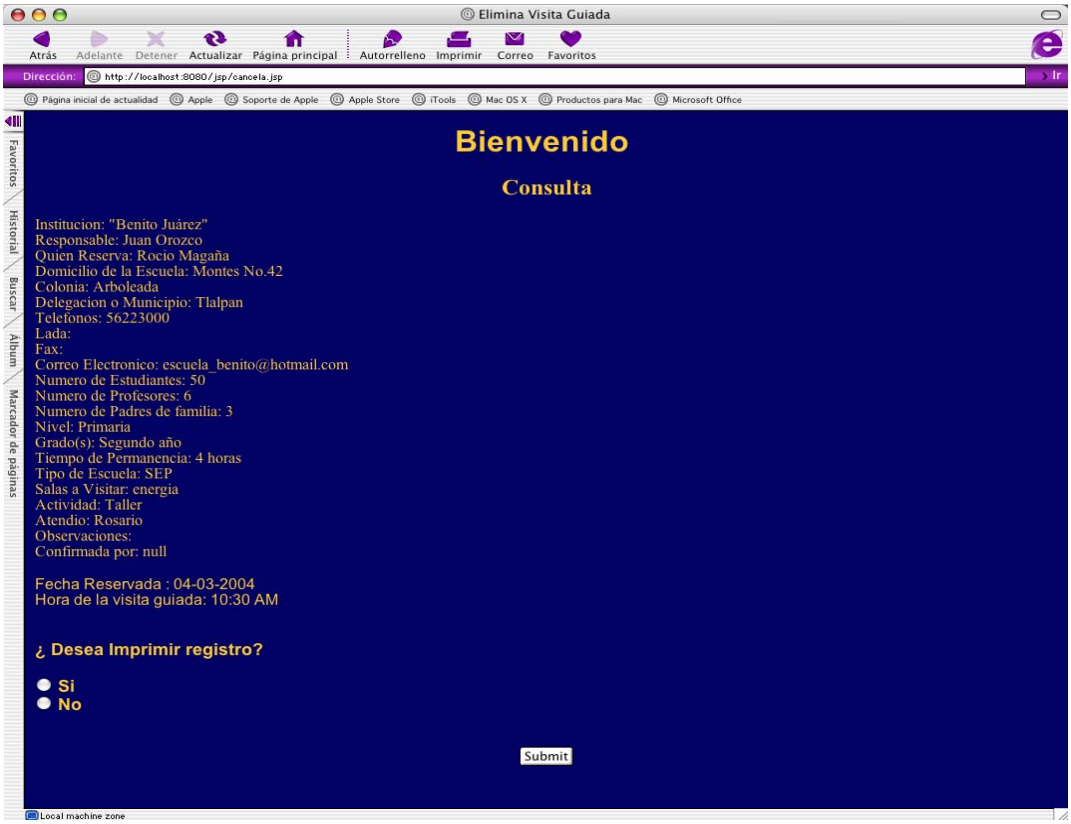


Prototipo : Consulta




VALIDACION INSTITUCION
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/proy_tesis/html/TMP6i2xg6lztb.htm
Ir
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
Validación de la clave de la Institución
Clave de la institución :
Submit
Reset
Local machine zone


VALIDACION INSTITUCION
Atrás
Adelante
Detener
Actualizar
Página principal
Autorrelleno
Imprimir
Correo
Favoritos
Dirección:
file://localhost/proy_tesis/html/TMP162sv6njq0.htm
Ir
Página inicial de actualidad
Apple
Soporte de Apple
Apple Store
iTools
Mac OS X
Productos para Mac
Microsoft Office
Favoritos
Historial
Buscar
Álbum
Marcador de páginas
Validación de la clave de la Institución
Alerta de script de Internet Explorer
La clave es incorrecta, vuelta a intentarlo
OK
Submit
Reset
Recibiendo documento HTML (0 bytes de 782 bytes, 0 bytes/seg.): TMP162sv6njq0.htm



Hasta aquí hemos visto los requerimientos del producto y su prototipo de acuerdo a las funciones especificadas por el usuario final.

Capítulo 3

3. Análisis del nuevo sistema a desarrollar

La problemática que enfrenta el Área de Atención al Visitante es la falta de control de las escuelas que ingresan al museo mediante una visita guiada y la automatización de las reservaciones con sus respectivas claves para cada reservación que realiza una Institución, así como la falta de rapidez y poca eficiencia al momento de verificarla.
Para resolver estos problemas se tienen contempladas las tecnologías de software en el manejo de información para lograr así un Sistema de Información y Control de Visitantes por Visitas guiadas.

El objetivo en ésta tesis es colaborar con la DGDC Universum en el desarrollo de un sistema de Información y Control Automatizado que cumpla con las necesidades de la Institución contribuyendo así al mejor desempeño de la misma.

Particularmente:

1. Reservar visitas guiadas a diferentes Instituciones.

Este punto incluye todas las instituciones que radican en toda la república mexicana , siempre y cuando estén registradas en el sistema.

1.1 Encontrar la forma rápida y eficiente de hacer las reservaciones.

Al mencionar la palabra automatizar y precisar la información me refiero a la realización de un sistema capaz de asignar claves de manera automática , en éste caso no sólo las claves serán asignadas automáticamente sino que también se registrarán fechas de registro de manera automática, por lo que dichas acciones serán ejecutadas de manera exacta en el momento preciso .

1.2 Controlar, Manipular, Almacenar y Transportar dicha información.

Para poder controlar, manipular ,almacenar y transportar la información es necesario un manejador de base de datos, dicho manejador elegido ha sido PostgreSQL ya que ha sido diseñado para almacenar grandes cantidades de información, lo cual es uno de los requerimientos para la creación de éste sistema, además de que la seguridad con la que ha sido diseñado éste manejador ha sido de tal forma que nuestra información puede estar almacenada de manera confiable , disminuyendo así el riesgo de que cualquier otra persona ajena a la institución pueda acceder a la información recabada.
Los datos van a ser manipulados mediante el sistema, de tal forma que se puedan insertar datos en la base, borrar datos en la base y actualizar los datos, cabe mencionar que la inserción, borrado y actualización de datos de los usuarios del sistema va a ser única y exclusivamente tarea de los usuarios registrados como administradores, en cuanto a la manipulación de datos de las instituciones podrá ser realizada por un administrador o un usuario normal ya registrado en el sistema.

1.3 Consulta de la información en los sistemas operativos existentes en la dependencia.

Dado que Universum no posee una homogeneidad en equipo de cómputo y sistemas operativos , con esto me refiero a que utiliza PC´s con sistema operativo Windows en sus diferentes versiones y Mac´s en sistemas 9.X y X, es necesario que el sistema sea desarrollado en Java ya que es un lenguaje multiplataforma, es decir, no importa el equipo de cómputo ni el sistema operativo que se utilice , el sistema funcionará sobre cualquier equipo, siempre y cuando se tenga instalado el lenguaje en dicho equipo.

2. Logística
2.1 Llevar un seguimiento de las instituciones que han hecho reservaciones

Esto es almacenando la información en la base de datos, para después poder ser consultada, para ello es necesario que la base contenga las siguiente información:

Tipo de institución.
Fecha de registro de la institución.
Clave de registro de la institución.
Nombre de la institución.
Domicilio de la institución.
Colonia de la institución.
Delegación o municipio a la que pertenece la institución.
Entidad Federativa
Clave ante la SEP.
Teléfono de la institución.
Lada de la institución.
Fax de la institución.
Correo de la institución.
Número de estudiantes que van ingresar a la institución.
Número de profesores que van ingresar a la institución.
Número de padres de familia que van ingresar a la institución.
Nivel de estudios de los estudiantes.
Grado de estudios de los estudiantes.
Tipo de visita.
Fecha de la reservación.
Clave de la reservación.
Responsable de la institución.
Persona que llama por parte de la institución.
Fecha planeada para la visita.
Hora planeada para la visita.
Tiempo de permanencia.
Sala(s) de preferencia para visitar.
Tipo de actividad a realizar dentro del museo.
Nombre de la persona que atendió a la institución para su reservación.
*Observaciones de la visita realizada.
*Persona que confirma la visita
* Fecha de confirmación

En caso de que existan cancelaciones de visitas:

Nombre de la persona por parte de la institución que cancela la visita.
Fecha de cancelación.
Nombre de la persona por parte de la institución que realizó la cancelación.

Información extra para llevar un mejor control del sistema y de sus funciones:

Tipo de usuario (Administrador/Normal)
Fecha de registro del usuario.
Nombre del usuario.
Apellidos del usuario (aprobado en un solo campo).
Login del usuario.
Password del usuario.

2.1.1 Manipulación de base de datos.

Esto es haciendo uso de subprogramas que realicen la inserción, borrado y actualización de datos dentro del sistema.

2.2 Proveer información de utilidad a diferentes áreas del museo

Además de proveer información al Área de Atención al Visitante, también proveerá de información al Área de Becarios quién requiere saber cuantas visitas hay al día y así obtener el número de becarios que requiere, esto debido a que los becarios son quienes ofrecen las explicaciones en las diferentes salas del museo.
No obstante gracias al sistema , taquilla podrá saber el número de boletos que deben ser reservados para su venta los días de visitas.
Toda está información puede ser automatizada y es una de las extensiones que el sistema podrá tener a futuro, no incluido en el desarrollo actual del sistema , pero si contemplado para su expansión.

La razón por la cual no se continúa con la expansión del sistema es debido a que se están desarrollando sistemas en el museo que aún no han sido implementados, de otra manera el sistema tiene la capacidad de absorber dichos sistemas, además de que los alcances del sistema fueron definidos desde el inicio del sistema .

2.3 Manipular la información en forma dinámica.

Esto es , se podrán hacer consultas , inserciones y actualizaciones a la base de datos de manera dinámica haciendo uso del sistema, logrando así una disminución de tiempo en llamadas en espera por parte de las instituciones.

2.4 Consulta y manipulación de la información en múltiples plataformas.

Gracias al lenguaje con el que se desarrolla el sistema , la información recabada y almacenada en la base de datos va a poder ser consultada desde cualquier equipo de cómputo en cualquier sistema operativo.

3. Datos Generales de las Instituciones

Los datos generales de las instituciones van a poder ser consultados haciendo uso de la base de datos, cuyos datos ya han sido descritos en el punto 2.1

3.1 Proporcionar una base de datos.

La institución contará con un nuevo manejador de base de datos no antes implantado en la institución, el cual como ya había mencionada es PostgreSQL , el cual va a poder almacenar grandes cantidades de información, con mayor control en cuanto a su administración y a un bajo costo, ya que es software libre y no requiere de licencias para su uso.

Al tener como fundamento lo anterior, el Sistema de Reservaciones de Visitas guiadas para la DGDC Universum plantea como ámbitos de acción lo siguiente:

1. Registro de Instituciones que desean ingresar al Universum , ya sea como visita libre o visita guiada. Para ello los datos a ingresar serán:
    a) Nombre de la Institución.
    b) Responsable de la Institución.
    c) Nombre de la persona que llama para hacer la reservación.
    d) Domicilio de la escuela.
    e) Colonia
    f) Delegación o municipio.
    g) Teléfono(s).
    h) Lada (Adicional al formato que se maneja).
    i) Fax.
    j) Correo electrónico.
    k) Número de Estudiantes.
    l) Número de profesores.
    m) Número de padres de familia.
    n) Nivel de estudios.
    o) Grado de estudios.
    p) Tiempo de permanencia.
    q) Tipo de Institución(Oficial/ Particular/ SEP).
    r) Sala(s) de preferencia para visitar.
    s) Actividad.
    t) Nombre de la persona que atendió la llamada (esto lo realiza el sistema de manera automática).
    u) Observaciones.
    v) Fecha en la que se hace la reservación.
    w) Día planeado para la visita.
    x) Fecha planeada para la visita.
    y) Hora planeada para la visita.

2. Registro de usuarios que van a hacer uso del sistema. Para ello los datos con los que los usuarios se darán de alta en el sistema serán los siguientes:

    a) Tipo de usuario (Administrador/Normal)

b) Fecha de registro del usuario, esto lo hace el propio sistema de manera automática.
c) Nombre del usuario.
d) Apellidos del usuario (aprobado en un solo campo).
e) Login del usuario.
f) Password del usuario.

3. Asignación de claves de registro para cada institución con la oportunidad de que el Universum tenga control de los accesos.
4. Asignación de claves de reservación para una visita.

5. Cancelación de visitas guiadas. Para ello se registrará :
    a) Nombre de la persona por parte de la institución que realiza la cancelación.
    b) Fecha en la que se realiza la cancelación, esto lo hace el propio sistema de manera automática.
    c) Nombre del usuario del sistema que realiza la cancelación, el cual es registrado automáticamente por el propio sistema.

6. Confirmación de visitas guiadas.
    a) Nombre de la persona por parte de la institución que hace la confirmación de la vista.
    b) Fecha de confirmación de la reservación, ésta tarea la realiza el sistema de manera automática.

7. Consulta por parte de la DGDC Universum, de los datos registrados de las instituciones. Esto abarca los datos registrados en el punto 1.

8. Consulta de los datos de quienes hacen uso del sistema. Esto es :
    a) Tipo de usuario (Administrador/Normal)
    b) Fecha de registro del usuario.
    c) Nombre del usuario.
    d) Apellidos del usuario (aprobado en un solo campo).
    e) Login del usuario.
    f) Password del usuario.

Los datos anteriores sólo podrán ser consultados por un usuario tipo administrador.

9. Actualización y borrado de los datos registrados , en donde el propio sistema tendrá la capacidad de realizar tales acciones.

10. Registro del número de estudiantes, profesores y familiares que van a ingresar al museo. Dicho registro se jerarquizará por las siguientes variables:
    10.1 Nivel académico.
    10.2 Grado que cursan.

Capítulo 4

4. Diseño y Planeación

Para está tesis la metodología utilizada para el diseño del sistema se basó en el Proceso Unificado de Desarrollo (Unified Process) el cual es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Como una de sus características principales tenemos que el Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial de Proceso Unificado.

Básicamente, los aspectos del Proceso Unificado se pueden resumir en frases clave: *Dirigido por casos de uso*, *Centrado en la arquitectura* e *Iterativo e incremental*.

El Proceso unificado consta de 4 fases:

- La Fase de inicio.
- La Fase de Elaboración.
- La Fase de Construcción.
- La Fase de Transición.

Estas fases aplican de manera iterativa cada uno de los 9 flujos de trabajo fundamentales de los que consta el Proceso Unificado:

1. Modelado del Negocio: Describe la estructura y la dinámica de la organización.
2. Requisitos: Describe el método basado en casos de uso para extraer los requisitos.
3. Análisis y Diseño: Describe las diferentes vistas arquitectónicas.
4. Implementación: Tiene en cuenta el desarrollo de software, la prueba de unidades y la integración.
5. Pruebas: Describe los casos de pruebas, los procedimientos y las métricas para evaluación de defectos.
6. Despliegue: Cubre la configuración del sistema entregable.
7. Gestión de configuraciones: Controla los cambios y mantiene la integridad de los artefactos de un proyecto.
8. Gestión del Proyecto: Describe varias estrategias de trabajo en un proceso iterativo.
9. Entorno: Cubre la infraestructura necesaria para desarrollar un sistema.

Dada la descripción en el capítulo 2 sección 2.5 de cada uno de los flujos de trabajo fundamentales de los que consta el Proceso Unificado, se procederá con el Modelado del Negocio.

Modelado del Negocio

En éste punto se hará una descripción de la estructura y organización del sistema a desarrollar. En donde cada módulo descrito cubre con las necesidades del usuario especificadas anteriormente.

Módulo	Funciones
Instituciones	Registrar Instituciones Modificar cualquier registro de Institución Consultar un registro de alguna Institución. Consultar todos los registros de las Instituciones.
Reservaciones	Registrar reservaciones Consultar un registro de alguna reservación Confirmar una reservación Cancelar una reservación Consultar todos los registros de reservaciones
Usuario	Registrar un usuario Consultar un registro de algún usuario Modificar registros de usuarios Borrar registros de usuarios Consultar todos los registros de usuarios dados de alta en el sistema. Asignación de privilegios para usuarios

Requisitos

En este punto veremos los casos de uso, sus actores, y diagramas de actividades involucrados en el sistema desarrollado para después proseguir con la fase de programación.

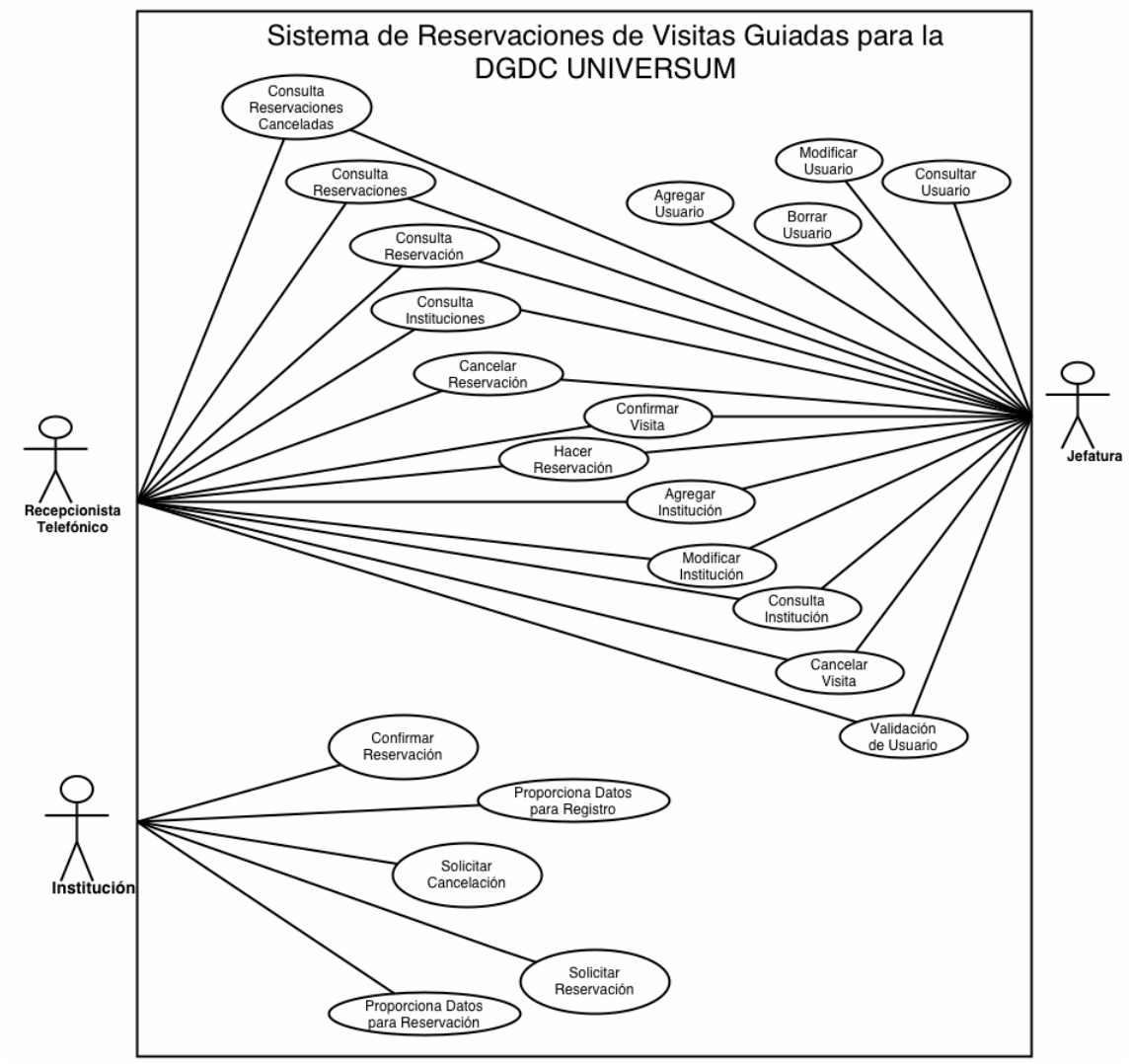


Diagrama1. Diagrama de Casos de Uso y Actores del Sistema desarrollado.

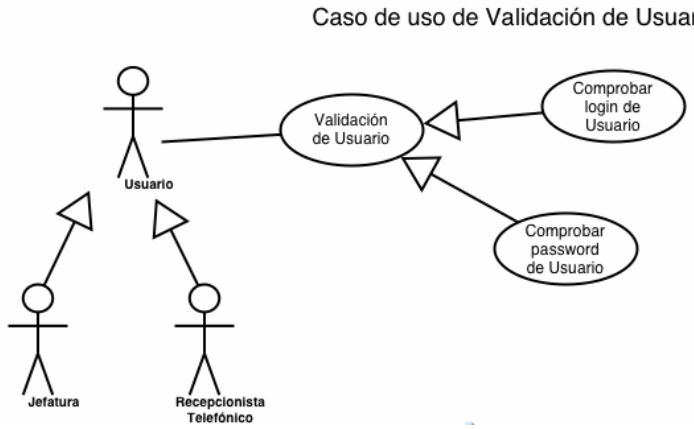


Diagrama 2.

Diagrama 3.

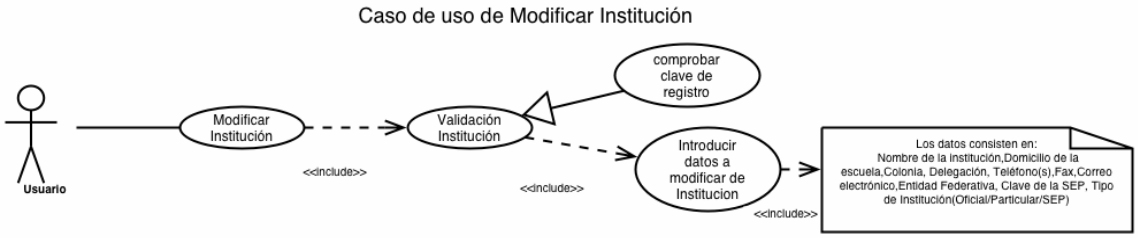


Diagrama 4.

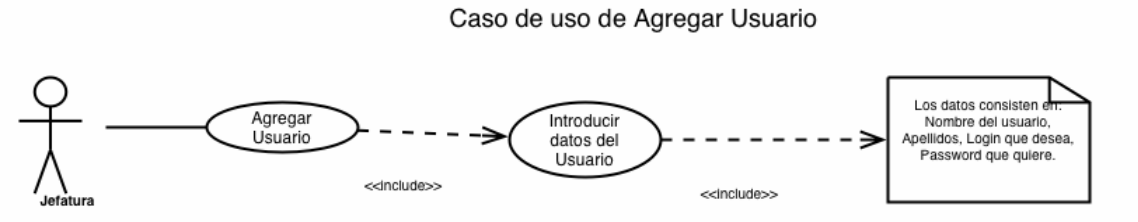


Diagrama 5.

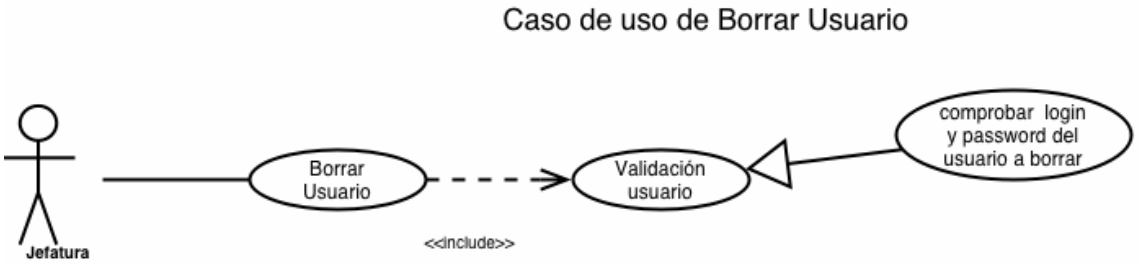


Diagrama 6.

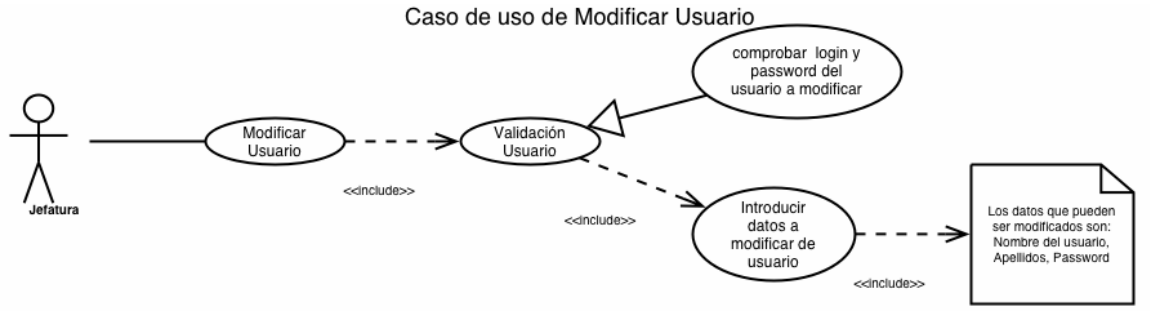


Diagrama 7.

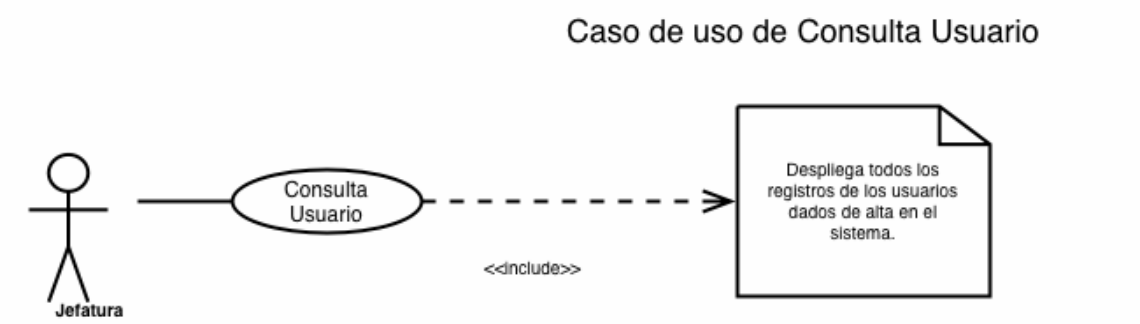


Diagrama 8.

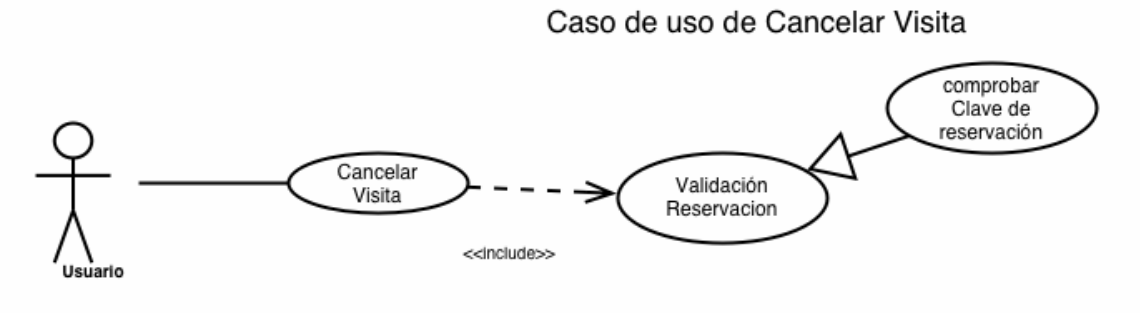


Diagrama 9.

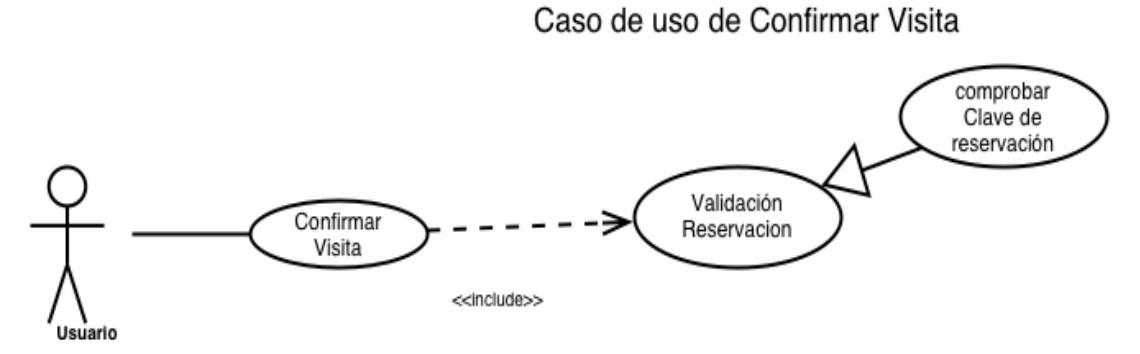


Diagrama 10.

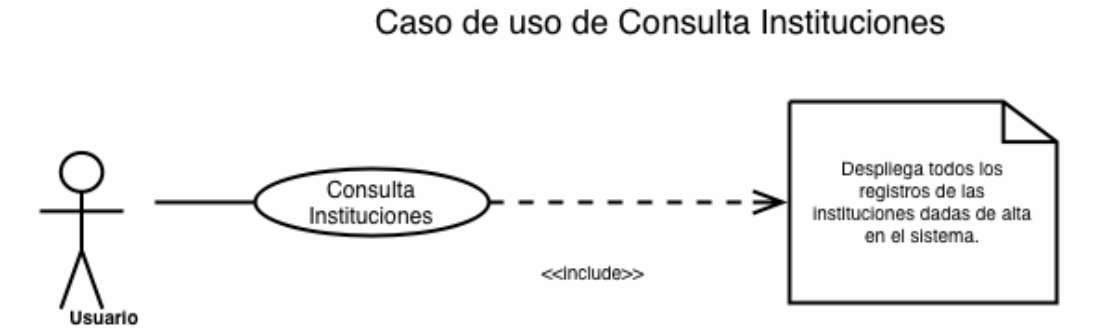


Diagrama 11.

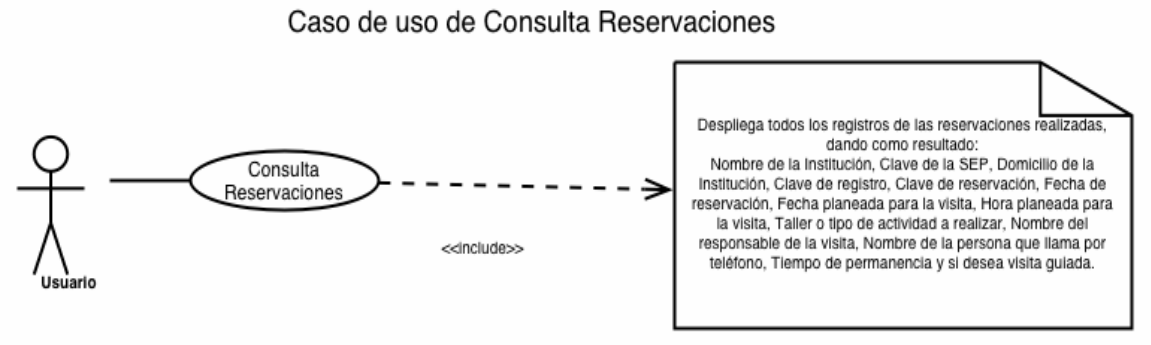


Diagrama 12.

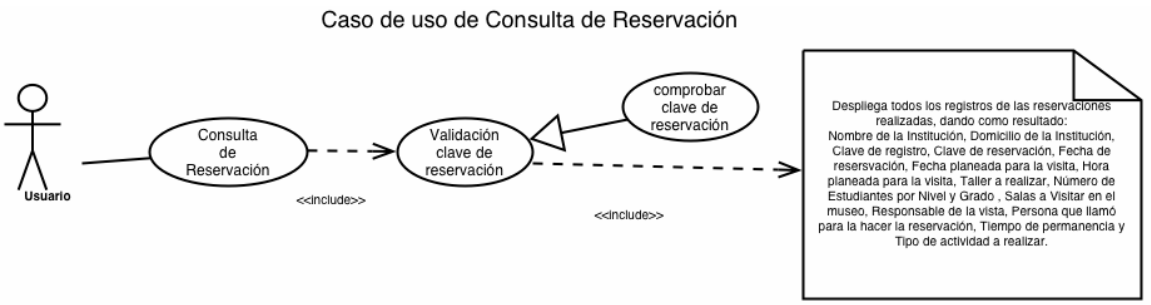


Diagrama 13.

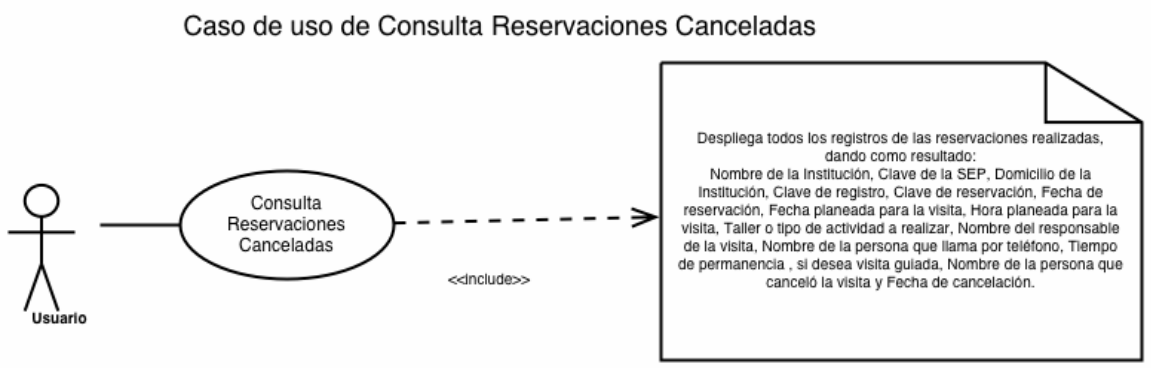


Diagrama 14.

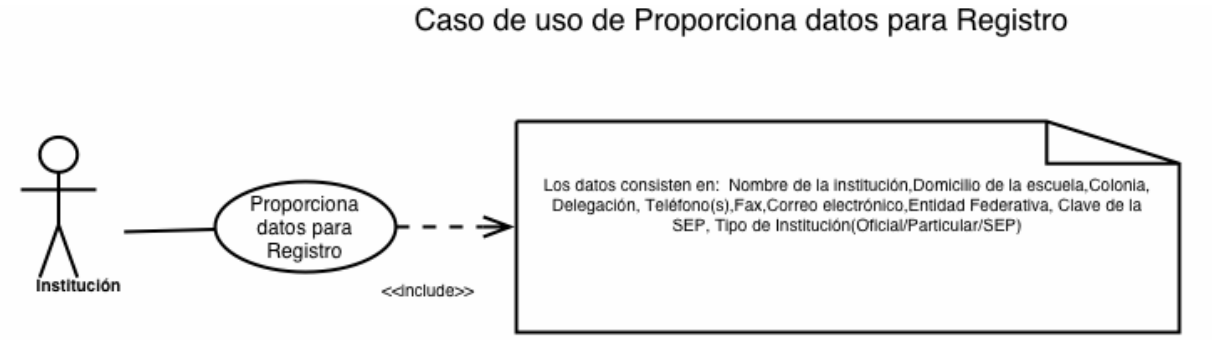


Diagrama 15.

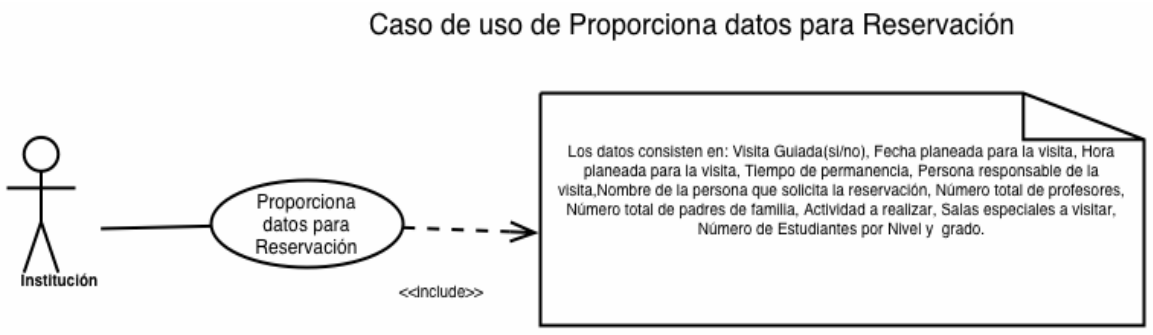


Diagrama 16.

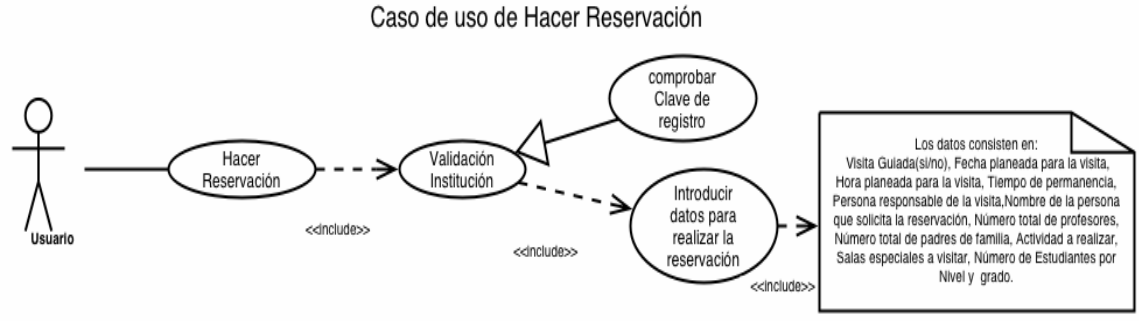


Diagrama 17.

A continuación veremos los diagramas de Actividades obtenidos a partir de los casos de uso realizados .

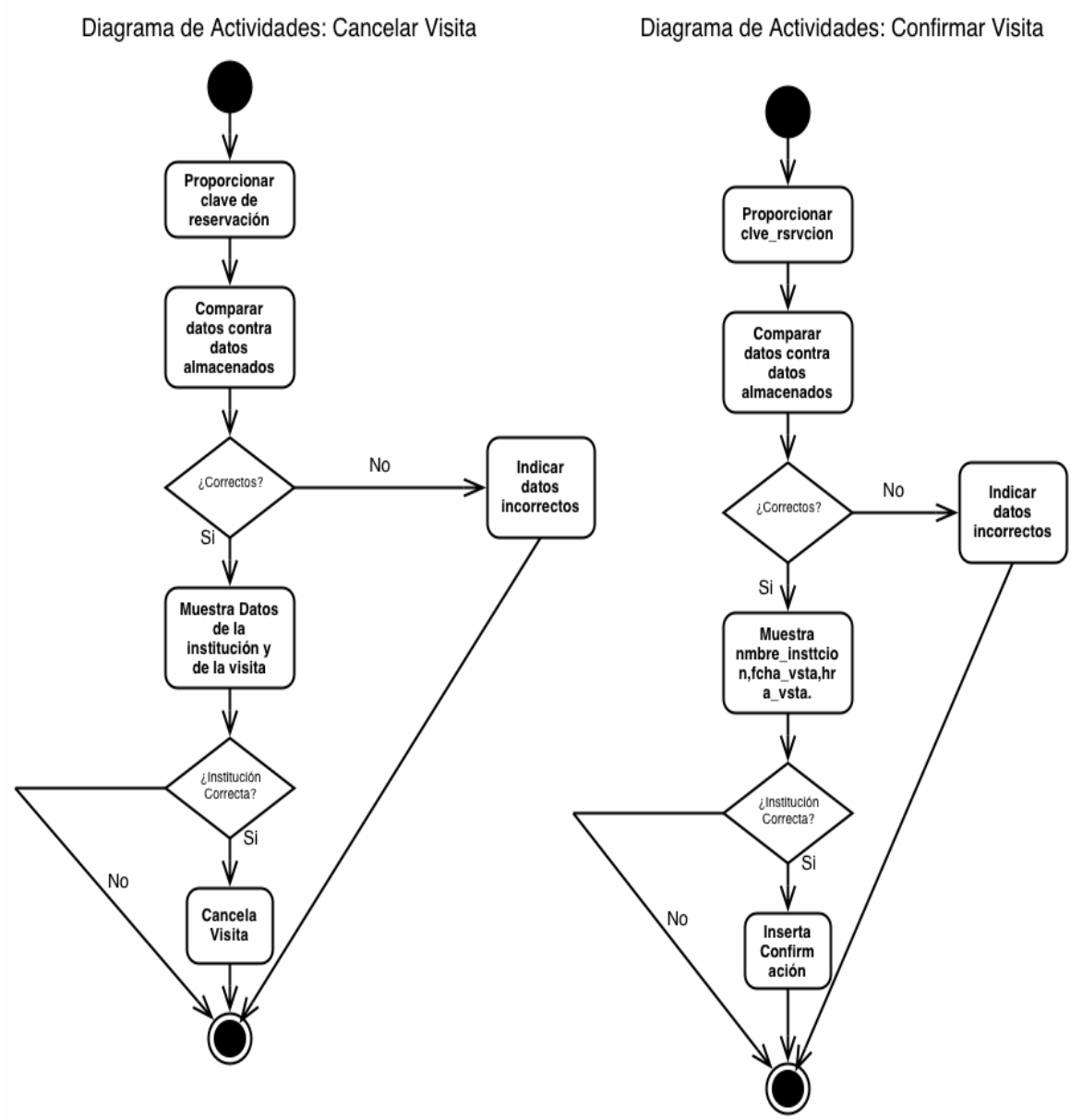


Diagrama de Actividades: Modificar Institución

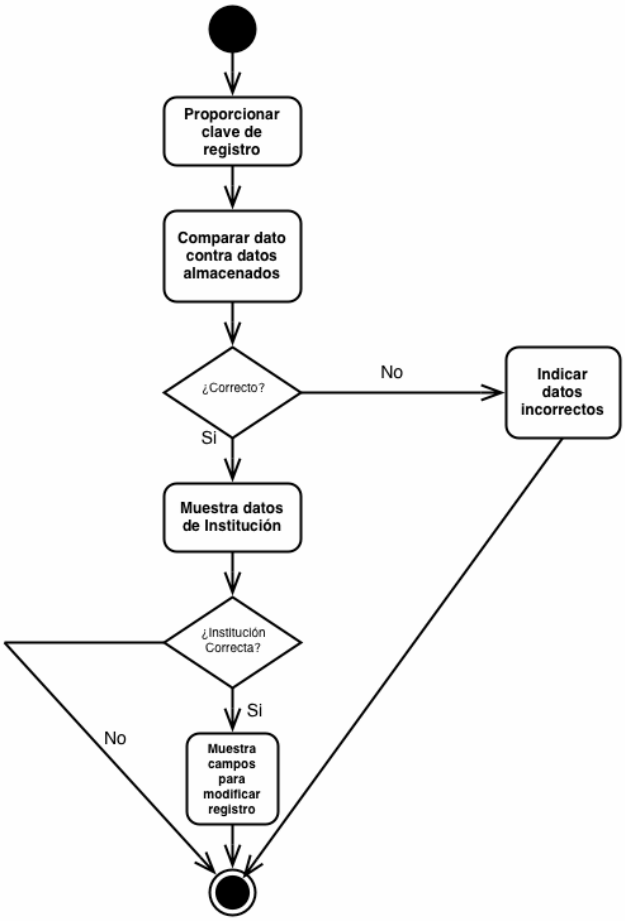


Diagrama de Actividades: Agregar Usuario

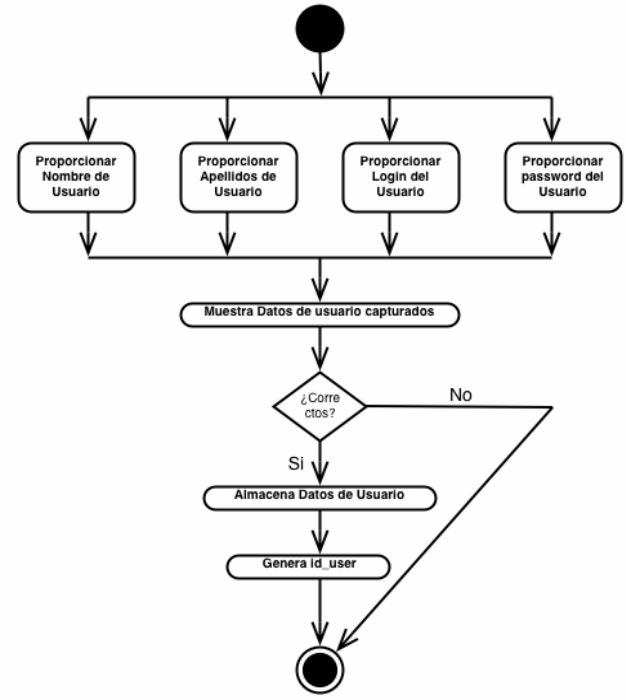


Diagrama de Actividades: Modificar Usuario

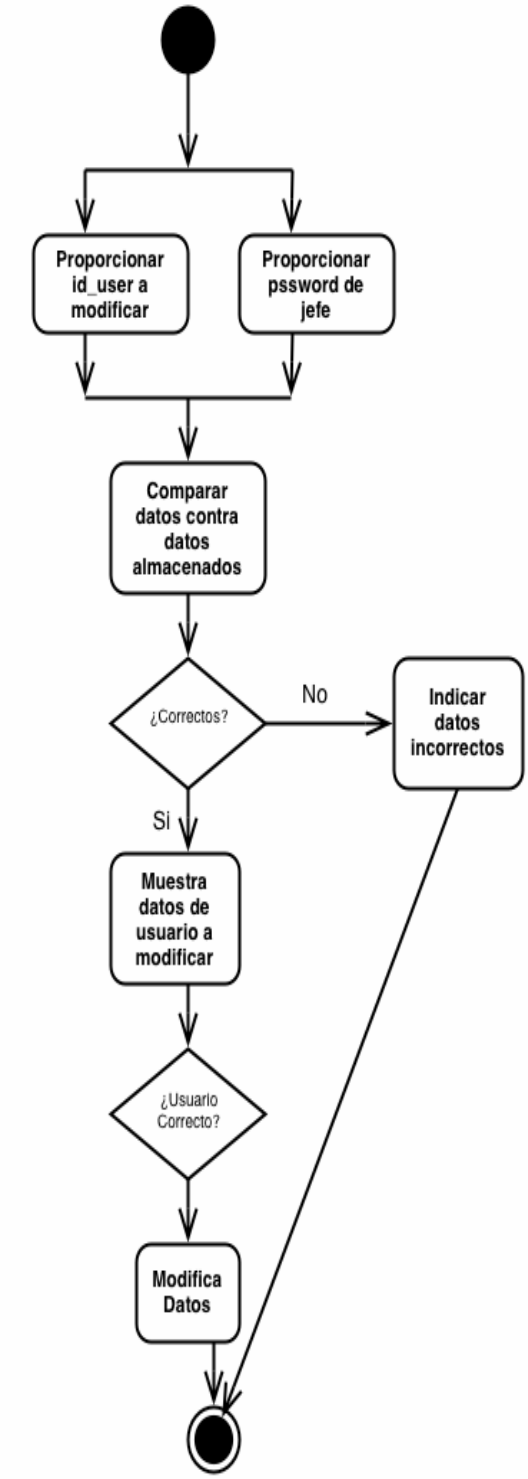
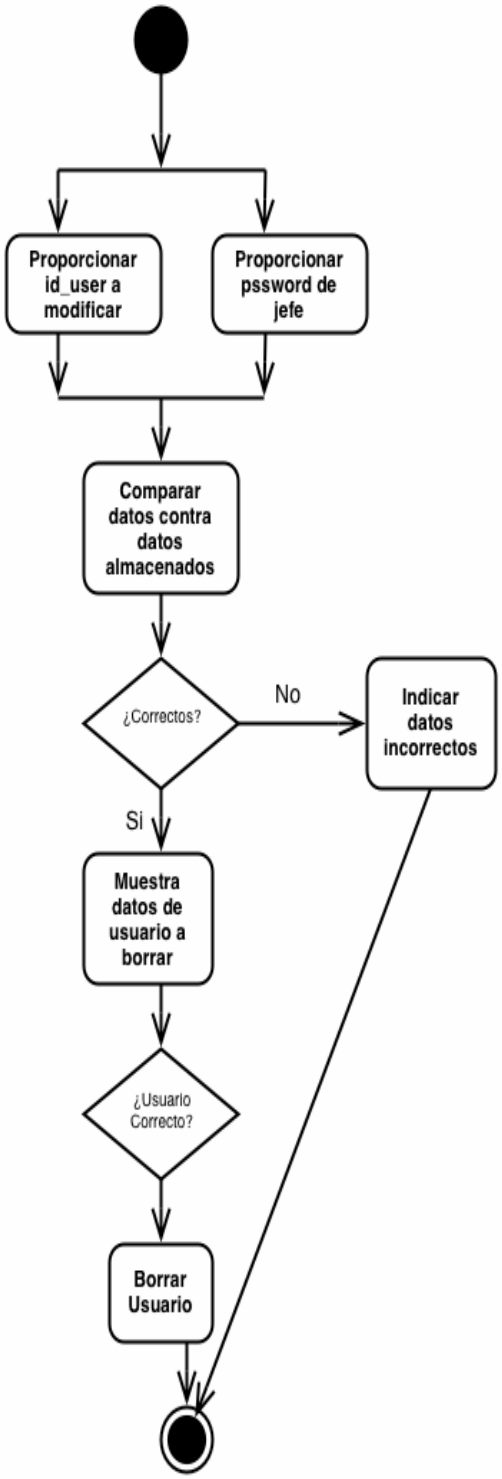


Diagrama de Actividades: Borrar Usuario



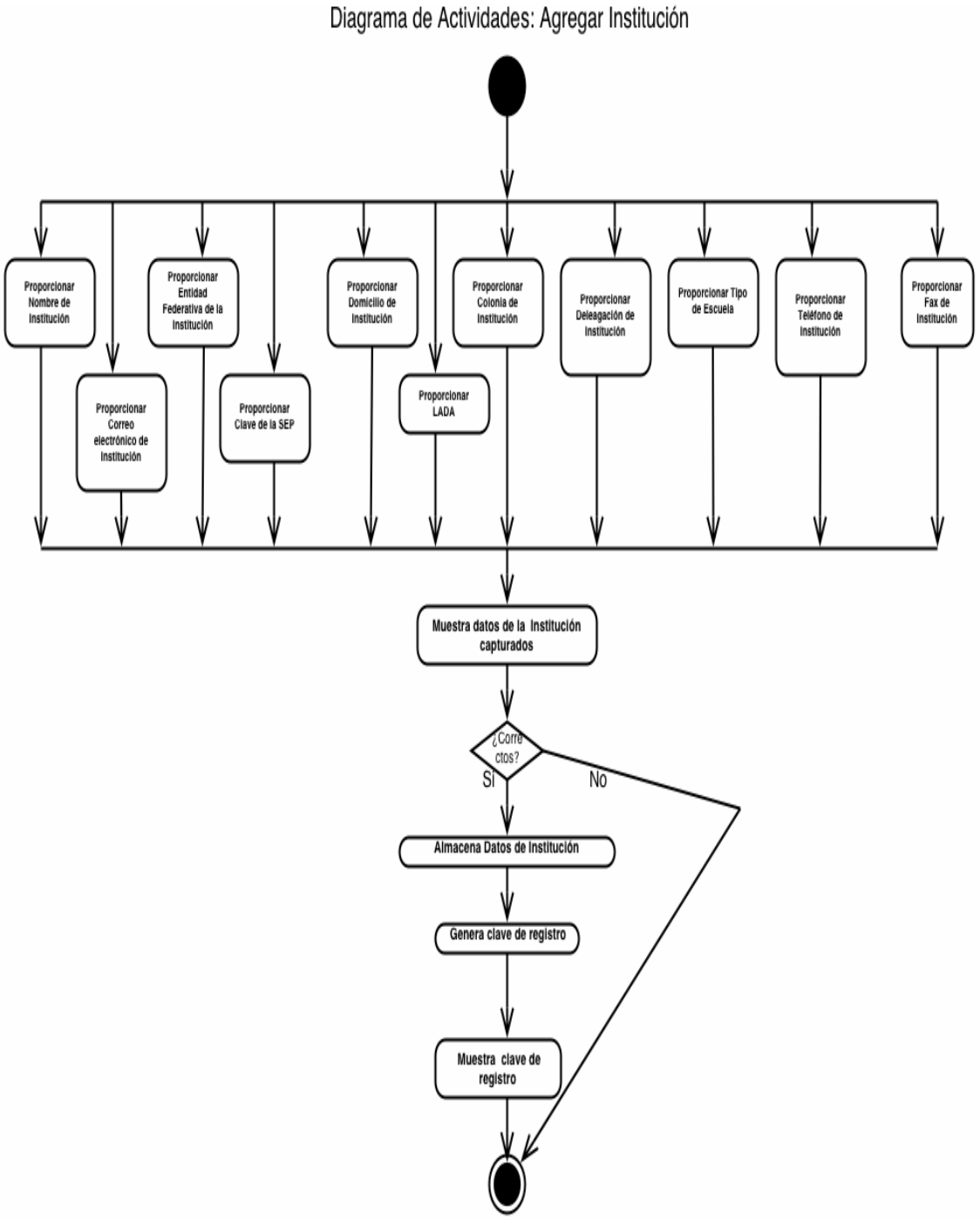


Diagrama de Actividades: Agregar Institución
Proporcionar Nombre de Institución
Proporcionar Correo electrónico de Institución
Proporcionar Entidad Federativa de la Institución
Proporcionar Clave de la SEP
Proporcionar Domicilio de Institución
Proporcionar LADA
Proporcionar Colonia de Institución
Proporcionar Deleagación de Institución
Proporcionar Tipo de Escuela
Proporcionar Teléfono de Institución
Proporcionar Fax de Institución
Muestra datos de la Institución capturados
¿Correctos?
Sí
No
Almacena Datos de Institución
Genera clave de registro
Muestra clave de registro

Diagrama de Actividades: Hacer Reservación

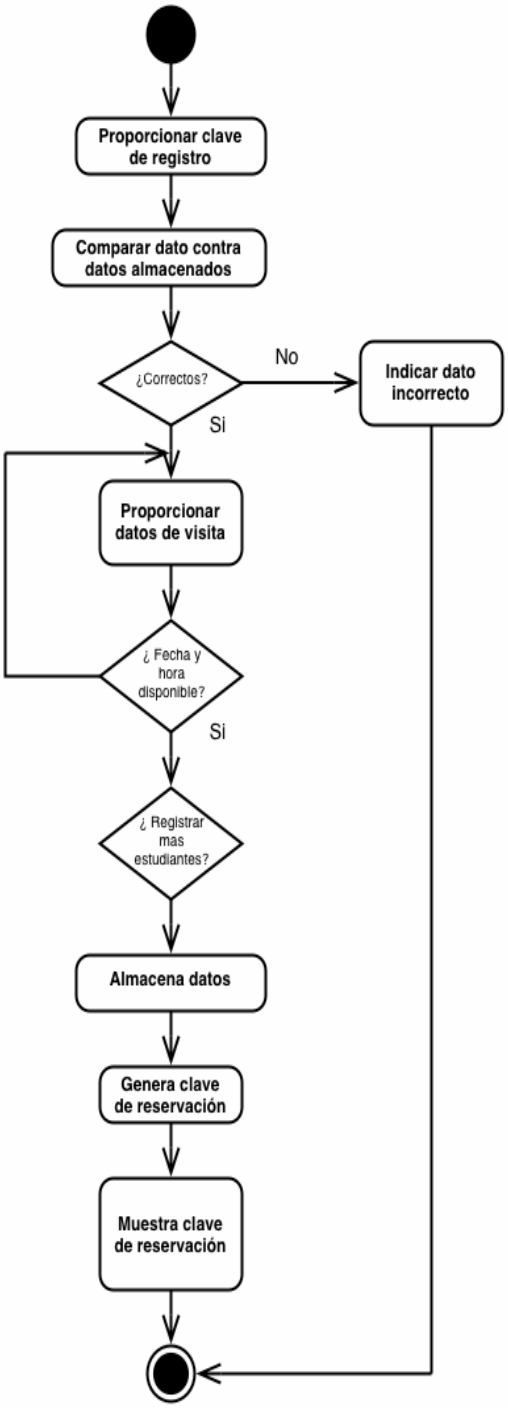
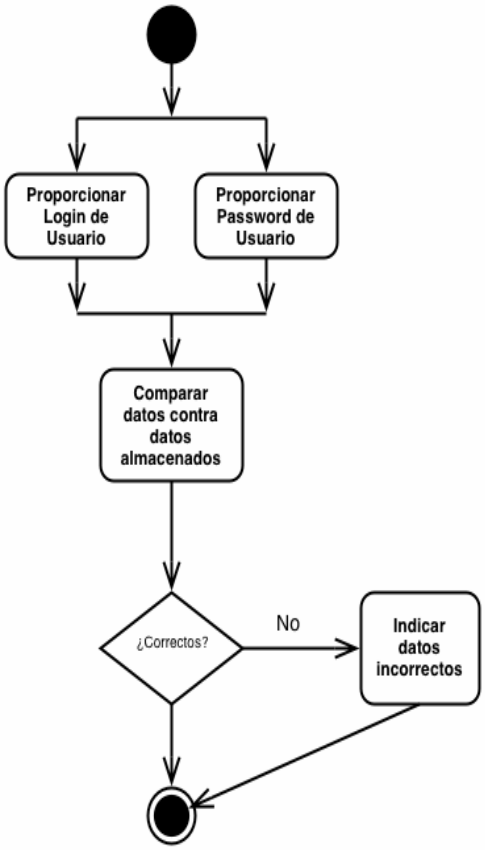


Diagrama de Actividades: Validación Usuario



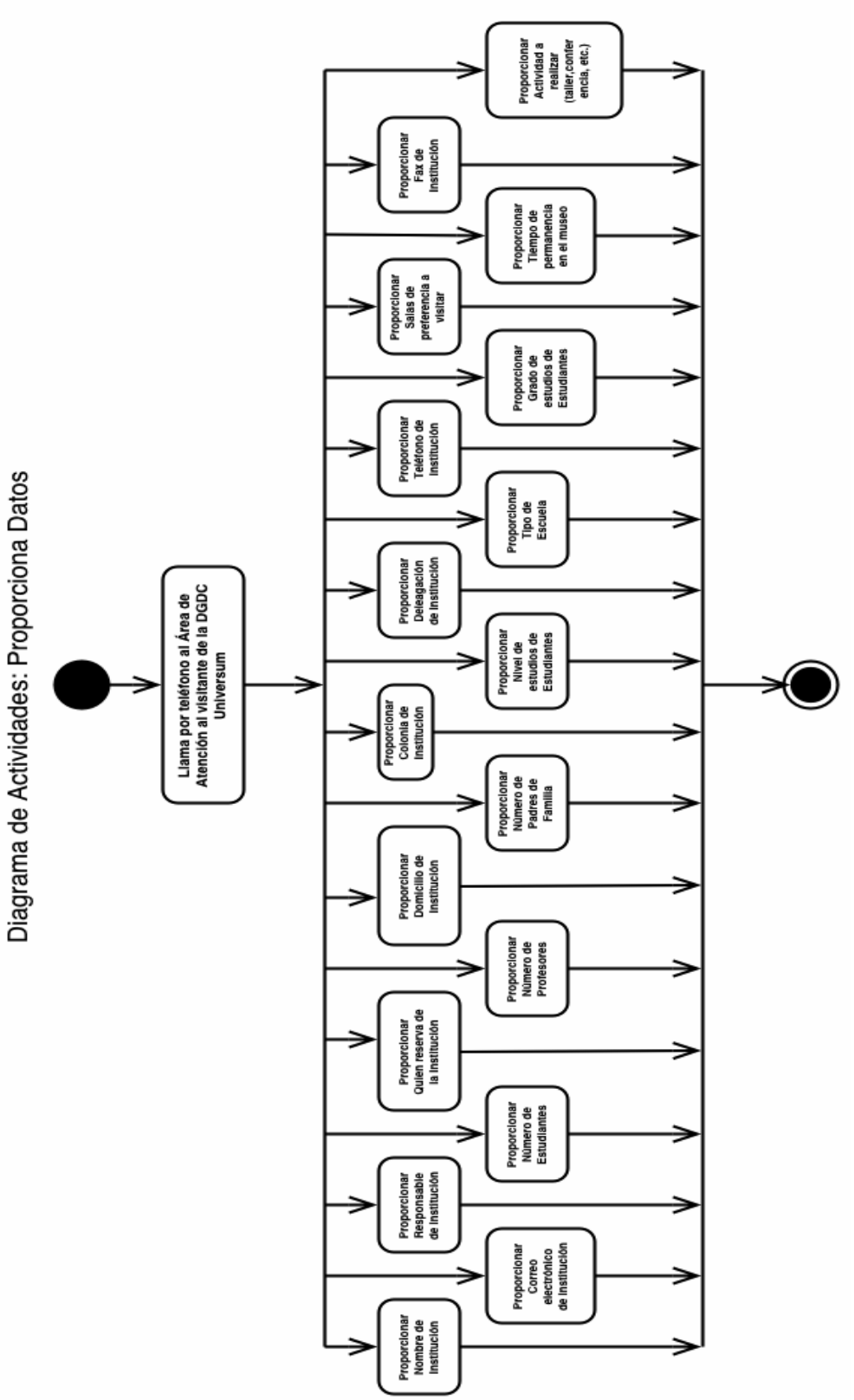


Diagrama de Actividades: Proporciona Datos
Llama por teléfono al Área de Atención al visitante de la DGDC Universum
Proporcionar Nombre de Institución
Proporcionar Correo electrónico de Institución
Proporcionar Responsable de Institución
Proporcionar Número de Estudiantes
Proporcionar Quien reserva de la Institución
Proporcionar Número de Profesores
Proporcionar Domicilio de Institución
Proporcionar Número de Padres de Familia
Proporcionar Colonia de Institución
Proporcionar Nivel de estudios de Estudiantes
Proporcionar Delegación de Institución
Proporcionar Tipo de Escuela
Proporcionar Teléfono de Institución
Proporcionar Grado de estudios de Estudiantes
Proporcionar Salas de preferencia a visitar
Proporcionar Tiempo de permanencia en el museo
Proporcionar Fax de Institución
Proporcionar Actividad a realizar (taller,conferencia, etc.)

Diagrama de Actividades: Solicitar Cancelación

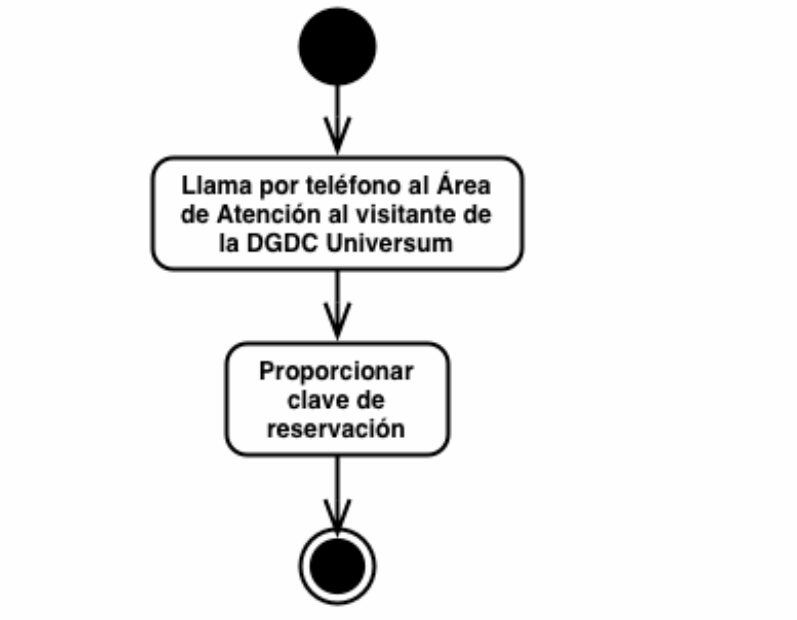


Diagrama de Actividades: Confirmar Reservación

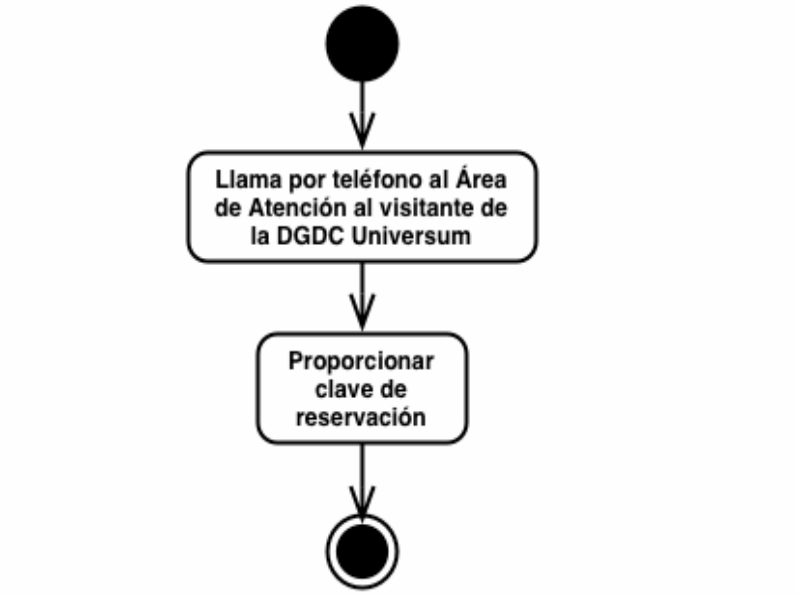


Diagrama de Actividades: Solicitar Reservación

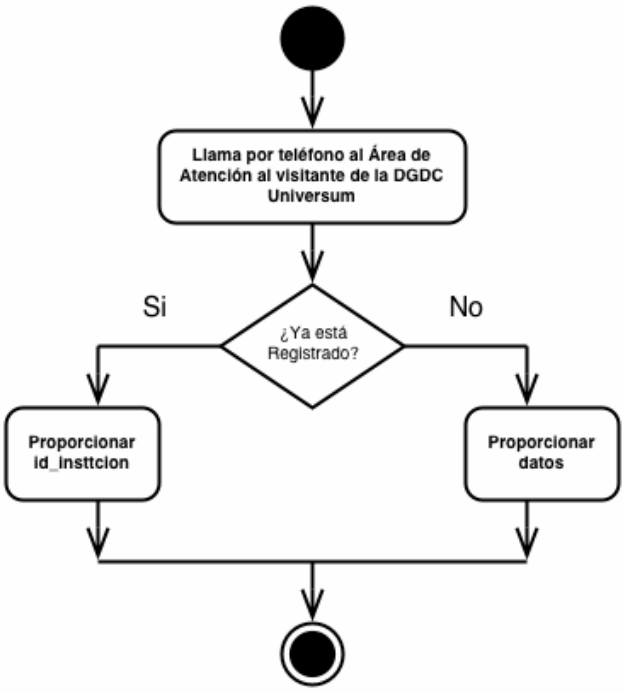
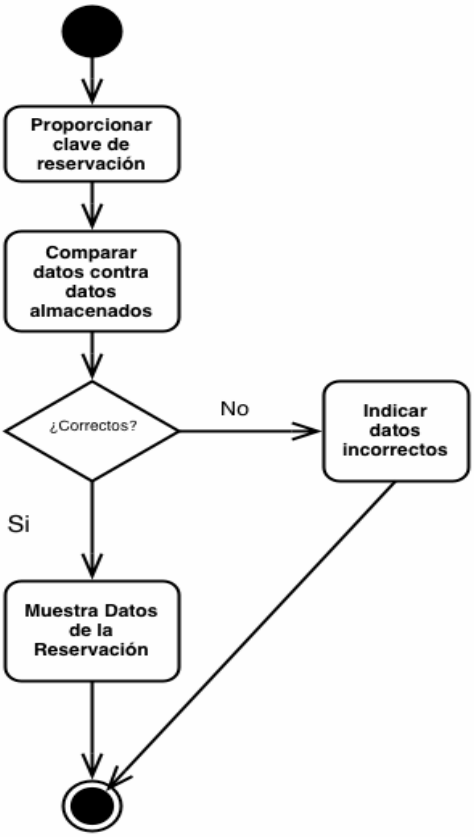


Diagrama de Actividades: Consulta Reservación



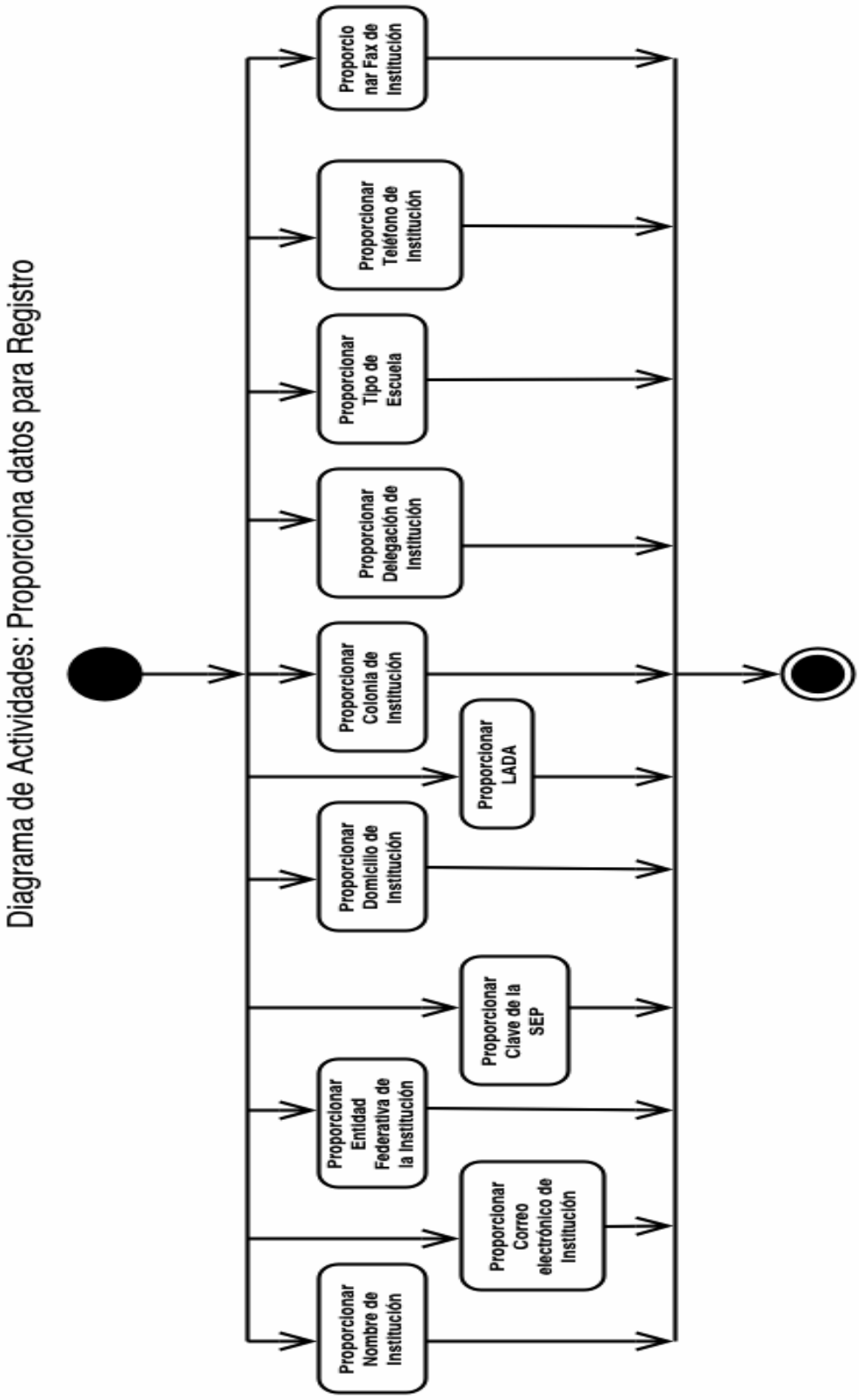


Diagrama de Actividades: Proporciona datos para Registro
Proporcionar Nombre de Institución
Proporcionar Correo electrónico de Institución
Proporcionar Entidad Federativa de la Institución
Proporcionar Clave de la SEP
Proporcionar Domicilio de Institución
Proporcionar LADA
Proporcionar Colonia de Institución
Proporcionar Delegación de Institución
Proporcionar Tipo de Escuela
Proporcionar Teléfono de Institución
Proporcionar Fax de Institución

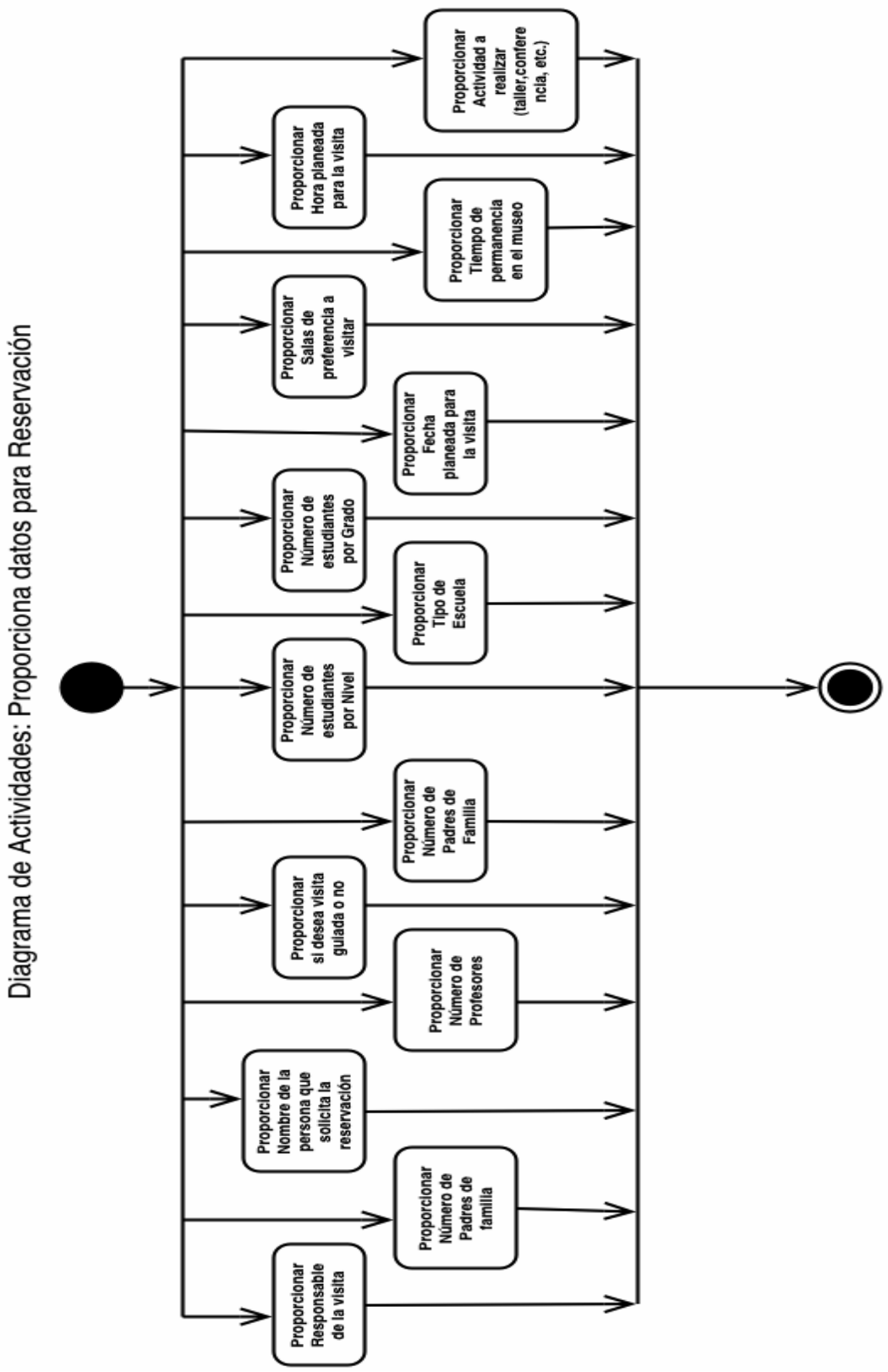
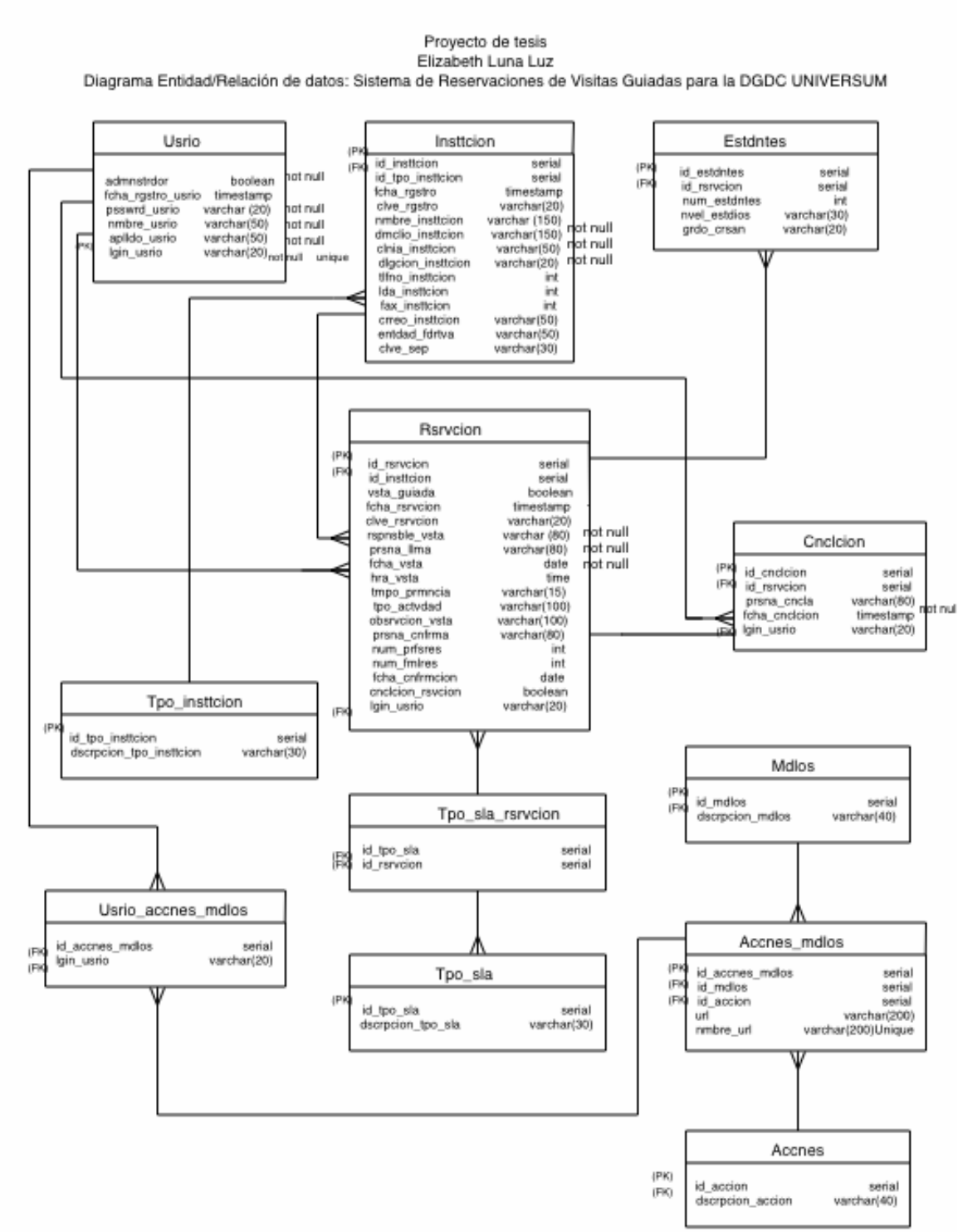


Diagrama de Actividades: Proporciona datos para Reservación
Proporcionar Responsable de la visita
Proporcionar Nombre de la persona que solicita la reservación
Proporcionar Número de Padres de familia
Proporcionar si desea visita guiada o no
Proporcionar Número de Profesores
Proporcionar Número de Padres de Familia
Proporcionar Número de estudiantes por Nivel
Proporcionar Tipo de Escuela
Proporcionar Número de estudiantes por Grado
Proporcionar Fecha planeada para la visita
Proporcionar Salas de preferencia a visitar
Proporcionar Tiempo de permanencia en el museo
Proporcionar Hora planeada para la visita
Proporcionar Actividad a realizar (taller,conferencia, etc.)

Una vez creados los diagramas de casos de uso , actores y diagramas de actividades del sistema el siguiente paso a seguir es la creación de los diagramas Entidad-Relación los cuales nos van a ayudar a obtener un buen diseño de la base de datos.
A continuación veremos los diagramas Entidad-Relación que se desarrollaron para la creación de la base de datos y su diccionario de datos.



Diccionario de Datos

Usrio		
Atributos	**Tipo de dato**	**Característica**
admnstrdor	boolean	Campo que indica si el usuario registrado en el sistema es un administrador, en dado caso su valor será “1” de no serlo su valor será “0”.
fcha_rgstro_usrio	timestamp	Fecha en la que se registra cada usuario del sistema.
lgin_usrio	varchar(20)	Nombre con el que ingresa el usuario al sistema.
psswrd_usrio	varchar(20)	Contraseña que tiene asignado cada usuario.
nmbre_usrio	varchar(50)	Nombre del usuario del sistema.
aplldo_user	varchar(50)	Apellido del usuario del sistema.

Insttcion		
Atributos	**Tipo de dato**	**Característica**
id_insttcion	serial	Llave primaria que sirve de identificador único de cada Institución registrada en el sistema.
id_tpo_insttcion	serial	Llave foránea perteneciente a la entidad Tpo_insttcion.
fcha_rgstro	timestamp	Fecha en la que se registra la Institución al sistema.
clve_rgstro	varchar(20)	Clave que se le asigna a cada institución registrada en el sistema para tener derecho a realizar reservaciones en el museo.
nmbre_insttcion	varchar(150)	Nombre con el que se registra la Institución.
dmclio_insttcion	varchar(150)	Domicilio de la Institución registrada.
clnia_insttcion	varchar(50)	Colonia a la que pertenece la Institución.
dlgcion_insttcion	varchar(20)	Delegación a la que pertenece la Institución.
tlfno_insttcion	int	Teléfono perteneciente a la Institución.
lda_insttcion	int	Lada de la Institución
fax_insttcion	int	Fax de la Institución
crreo_insttcion	varchar(50)	Correo electrónico de la Institución.
entdad_fdrtva	varchar(50)	Entidad Federativa a la que pertenece la institución.
clve_sep	varchar(20)	Clave que la SEP le asigna a algunas instituciones.

Estdntes		
Atributos	**Tipo de dato**	**Característica**
id_estdntes	serial	Llave primaria que sirve de identificador de los datos relacionados con los estudiantes que van a ingresar al museo.
id_rsrvcion	serial	Llave foránea que sirve de identificador único de cada reservación realizada por cada institución registrada en el sistema.
num_estdntes	int	Número de estudiantes que van a ingresar al museo.
nvel_estdios	varchar(30)	Nivel de estudios de los estudiantes que van ingresar al museo como son : Maternal, Kinder, Preescolar, Primaria, Secundaria, Bachillerato, Licenciatura, Tercera Edad, Discapacidad, Por Empresa, Vulnerables.
grdo_crsan	varchar(20)	Grado de estudios de los alumnos que van a ingresar al museo tales como: (Primer /Segundo/Tercer/Cuarto/Quinto/Sexto)año o Ninguno.

Tpo_sla		
Atributos	**Tipo de dato**	**Característica**
id_tpo_sla	serial	Llave primaria que sirve de identificador de las diferentes salas pertenecientes al museo.
dscrpcion_tpo_sla	varchar(30)	Descripción de las diferentes salas pertenecientes al museo, tales como: Aventura,Balsa,Biodiversidad, Conciencia, Cosechando,Energía, E.Infantil, Estrctura. M. , Infraestructura, Matemáticas, Movimiento, Química, Reproducción, Senda, Tec.Satelital, Universo.

Rsrvcion		
Atributos	**Tipo de dato**	**Característica**
id_rsrvcion	serial	Llave primaria que sirve como identificador de todos los datos relacionados con las reservaciones de las instituciones.
id_insttcion	serial	Llave foránea que sirve de identificador único de cada Institución registrada en el sistema.
lgin_usrio	varcha(20)	Nombre con el que ingresa el usuario al sistema.
vsta_guiada	boolean	Campo que indica si la reservación realizada es con visita guiada o no, en caso de serlo su valor es de “1”, y en caso de ser visita libre (sin guia) su valor es de “0”.
fcha_rsrvcion	timestamp	Fecha en la que se hace la reservación de una visita guiada.
clve_rsrvcion	varchar(20)	Clave asignada a cada institución una vez que ha realizado la reservación de una visita guiada.
rspnsble_insttcion	varchar(80)	Nombre de la persona responsable de los estudiantes que van a ingresar al museo.
prsna_llma	varchar(80)	Nombre de la persona que llama de parte de la institución para realizar la reservación de una visita guiada.
cnclcion_rsrvcion	varchar(5)	Dice si se canceló o no una visita guiada.
fcha_vsta	date	La fecha planeada para la visita guiada por día, mes y año.
hra_vsta	time	Hora en la que se les va a dar la visita guiada a los estudiantes.
tmpo_prmncia	varchar(15)	Tiempo en que va a permancer una institución dentro del museo.
tpo_actvdad	varchar(100)	Actividad extra que se va a realizar dentro del museo, tales como: Tomar un taller, una conferencia, etc.
obsrvcion_vsta	varchar(30)	Observaciones registradas por el museo de acuerdo al comportamiento de los estudiantes dentro de las instalaciones.
prsna_cnfrma	varchar(80)	Nombre de la persona que llama de parte de la institución para confirmar una visita guiada.
fcha_cnfrmcion	date	Fecha en la que se realiza la confirmación de una visita guiada.

Cnclcion

Atributos	Tipo de dato	Característica
id_cnclcion	serial	Llave primaria que sirve de identificador para realizar la consulta de datos relacionados a alguna cancelación hecha.
id_rsrvcion	serial	Llave foránea que sirve como identificador de todos los datos relacionados con las reservaciones de las instituciones.
lgin_usrio	varchar(20)	Nombre con el que ingresa el usuario al sistema.
prsna_cncla	varchar(80)	Nombre de la persona que llama de parte de la institución para cancelar una visita .
fcha_cnlcion	timestamp	Fecha en la que se realiza la cancelación de una visita.

Tpo_insttcion		
Atributos	**Tipo de dato**	**Característica**
id_tpo_insttcion	serial	Llave primaria que sirve de identificador del tipo de institución que se registra en el sistema.
dscrpcion_tpo_insttcion	varchar(30)	Descripción del tipo de institución registrada en el museo, ya sea como institución perteneciente a la SEP , como institución PARTICULAR o como institución que tiene algún CONVENIO con la SEP.

Tpo_sla_rsrvcion		
Atributos	**Tipo de dato**	**Característica**
id_tpo_sla	serial	Llave foránea , sirve como identificador de las diferentes salas pertenecientes al museo.
id_rsrvcion	serial	Llave foránea que sirve como identificador de todos los datos relacionados con las reservaciones de las instituciones.

Mdlos		
Atributos	**Tipo de dato**	**Característica**
id_mdlos	serial	Llave primaria, sirve como identificador de los diferentes módulos pertenecientes al sistema.
dscrpcion_mdlos	varchar(40)	Nombre de los módulos que posee el sistema.

Accnes_mdlos		
Atributos	**Tipo de dato**	**Característica**
id_accnes_mdlos	serial	Llave primaria, sirve como identificador de las acciones referidas a sus módulos.
id_mdlos	serial	Llave foránea, sirve como identificador de los diferentes módulos pertenecientes al sistema.
Id_accion	serial	Llave foránea, sirve como identificador de las acciones existentes en el sistema.
url		Ruta donde se encuentran contenidas las aplicaciones a utilizar en el sistema.
nmbre_url		Nombre de las ligas desplegadas en el menú

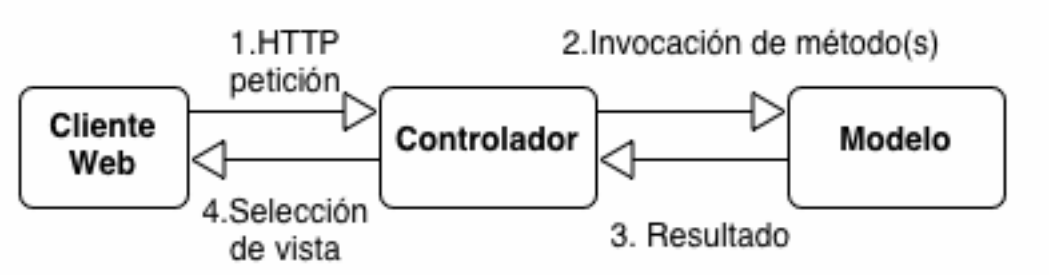
Accnes		
Atributos	**Tipo de dato**	**Característica**
id_accion	serial	Llave primaria, sirve como identificador de las acciones existentes en el sistema.
dscrpcion_accion	varchar(40)	Nombre de las acciones a realizar en el sistema

Usrio_accnes_mdlos		
Atributos	**Tipo de dato**	**Característica**
id_accnes_mdlos	serial	Llave primaria, sirve como identificador de las acciones referidas a sus módulos.
lgin_usrio	varchar(20)	Llave foránea que sirve como identificador de todos los datos relacionados con las reservaciones de las instituciones.

Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos.

Existe una aquitectura de aplicaciones de tres-capas, basadas en componentes, la cual fue presentada por Windows (Universal Access Data), llamada “Model View Controller Tecnology”. Donde la meta es unificar las aplicaciones cliente y servidor y las aplicaciones basadas en la Web, lo cual es posible para aplicaciones de cualquier tamaño.
La arquitectura “Model View Controller Tecnology” de tres capas se muestra en la siguiente gráfica.

Las capas de arquitectura son las siguientes:



Capa de presentación

Los servicios de presentación proporciona la interfaz necesaria para presentar información, reunir datos e integrar al usuario con la aplicación para ejecutar un proceso.

El cliente proporciona el contexto de presentación, generalmente un browser como Microsoft Internet Explorer o Netscape Navigator o Safari para Mac, que permite ver los datos remotos a través de una capa de presentación HTML.

La capa de servicios de presentación es responsable de :

1. Obtener información del usuario.
2. Enviar la información del usuario.
3. Recibir los resultados del procesamiento de los servicios.
4. Presentar estos resultados al usuario.

Capa de Negocio
Los servicios de negocios son el “puente” entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea.

El nivel de servicios de negocio es responsable de:

1. Recibir la entrada del nivel de presentación.
2. Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a autorizar.
3. Enviar el resultado procesado al nivel de presentación.

Capa de Datos
El nivel de servicios de datos es responsable de:

1. Almacenar los datos.
2. Recuperar los datos.
3. Mantener los datos.
4. La integridad de los datos.

[Internet 30]

El modelo “Model View Controller” es utilizado dentro de este proyecto de Tesis para modularizar tanto la presentación del sistema, el control de Datos y la comunicación entre este control de datos y la presentación o interfaz del sistema.

Dado que ya se realizaron los diagramas de diseño del sistema y se eligió el modelo para el desarrollo del sistema, el siguiente paso a seguir será la elección de las herramientas a utilizar.

Capítulo 5

5. Elección de las herramientas a utilizar: lenguaje de programación y sistema manejador de base de datos.

Hoy en día existen numerosos lenguajes de programación: C, C++, Microsoft® Visual Basic®, COBOL, Microsoft C#, Java, etc. Ante la presencia de tantos lenguajes, ¿en qué se basa un ingeniero de software para adoptar en un proyecto el uso de uno en concreto? En ocasiones, el lenguaje se elige por preferencia personal de los desarrolladores de una empresa, o bien, porque se posee un conocimiento del mismo, lo cual es razonable. Otras veces se opta por un lenguaje debido a que es el más reciente y el más potente, por lo que dicho lenguaje se convierte en una herramienta de mercadotecnia para generar más interés de relaciones públicas en un producto, lo cual no parece que sea algo razonable en sí mismo. Lo ideal sería seleccionar el lenguaje basándose en su capacidad para realizar un tarea determinada; el problema que hay que resolver debe determinar el uso de un lenguaje específico. [Internet 21]

Lenguaje de programación

Es una técnica estándar de comunicación para entregarle instrucciones a la computadora. Un lenguaje le da la capacidad al programador de especificarle a la computadora, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

Un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, es decir, ser traducido al lenguaje de máquina para que pueda ser ejecutado por el ordenador.

Los lenguajes de programación pueden ser clasificados de acuerdo a su forma de ejecución en:

- Lenguajes interpretados
- Lenguajes compilados

Un lenguaje compilado es un lenguaje de programación en el que una vez escrito el programa, éste se traduce a partir de su código fuente por medio de un compilador en un archivo ejecutable para una determinada plataforma (Unix, Windows, etc.).

Los lenguajes compilados son lenguajes de bajo nivel en los que las instrucciones se traducen del lenguaje utilizado a código máquina para una ejecución rápida. Por el contrario un lenguaje interpretado es aquel en el que las instrucciones se traducen o interpretan una a una siendo típicamente unas 10 veces más lentos que los programas compilados.[Internet 22]

A continuación se verá una comparación de lenguajes de programación, esto es para ver que lenguaje de programación es el que se apega más para cubrir con los requerimientos del usuario final del sistema a desarrollar.

Comparación de lenguajes

En este apartado se va a comparar Java con los lenguajes C++ y Smalltalk (primer lenguaje que presentaba un modelo de objeto).

Característica	**Java**	**Smalltalk**	**C++**
Sencillez	Sí	Sí	No
Robustez	SÍ	Sí	No
Seguridad	SÍ	Algo	No
Interpretado	SÍ	Sí	No
Dinamicidad	SÍ	Sí	No
Portabilidad	SÍ	Algo	No
Neutralidad	SÍ	Algo	No
Threads	SÍ	No	No
Garbage Colection	SÍ	Sí	No
Exceptions	SÍ	Sí	Algunas
Representación	Alta	Media	Alta

[Internet 23]

Sencillez

Java tiene una sencillez que no posee C++ aunque sí Smalltalk. Esto es debido a que una de las razones de la creación de Java es la de obtener un lenguaje parecido a C++ pero reduciendo los errores más comunes de la programación, algo que se logra con mucho éxito puesto que Java reduce un 50% los errores que se comenten en C++ entre los que destacan:

- Eliminación de la aritmética de punteros y de las *referencias*.
- Desaparecen los registros (*struct*), heredados del paradigma estructurado.
- No se permite ni la definición de tipos (*typedef*) ni la de macros (*#define*).
- Ya no es necesario liberar memoria (*free o delete*).

De todas formas, lo que Java hace, en realidad, es la eliminación de palabras reservadas, y la utilización de un intérprete bastante pequeño.[Internet 23]

Familiar. Como la mayoría de los programadores están acostumbrados a programar en C o en C++, el sintaxis de Java es muy similar al de estos.
[Internet 24]

Robustez

El sistema de Java maneja la memoria de la computadora por el usuario, por lo que el usuario no se tiene que preocupar por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique. [Internet 24]
Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución, lo que hace que se detecten errores lo antes posible, normalmente en el ciclo de desarrollo. Algunas de estas verificaciones que hacen que Java sea un lenguaje robusto son:

- Verificación del *código de byte*.
- Gestión de excepciones y errores.
- Comprobación de punteros y de límites de vectores.

Se aprecia una clara diferencia con C++ quién no realiza ninguna de estas verificaciones.[Internet 23]

Seguridad

El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
[Internet 24]

En Java no se permiten los accesos ilegales a memoria, algo que sí se permitía en C++.
El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de *código de byte* que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal, código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto.

Algunos de los conocimientos que podemos obtener de los códigos de byte si pasan la verificación sin generar ningún mensaje de error son:

- El código no produce desbordamiento de operandos en la pila.
- El tipo de los parámetros de todos los códigos de operación es conocido y correcto.
- No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros.
- El acceso a los campos de un objeto se sabe si es legal mediante las palabras reservadas *public*, *private* y *protected*.
- No hay ningún intento de violar las reglas de acceso y seguridad establecidas.

Por todo esto, y por no permitirlo mediante Java la utilización de métodos de un programa sin los privilegios del núcleo (*kernel*) del sistema operativo, la obligación de autentificación por clave pública para la realización de modificaciones, se considera Java un lenguaje seguro. Todo esto no lo incorporan ni C++ ni Smalltalk, por lo que Java es el único de los tres considerable como seguro. [Internet 23]

Lenguaje interpretado

Java es un lenguaje que puede ejecutar el código directamente, es decir es un "lenguaje interpretado". Esto es una característica que sí que posee Smalltalk, aunque no C++. No obstante, y aunque en teoría se consumen menos recursos siendo los lenguajes interpretados, el actual compilador que existe es bastante lento, unas 20 veces menos rápido que C++. Esto normalmente no es vital para la aplicación ni demasiado apreciable por el usuario, y además esta diferencia se está reduciendo con los nuevos compiladores JIT (*Just In Time).* [Internet 23]

Dinamicidad

Dinámico. Java no requiere que compiles todas las clases de un programa para que este funcione. Si realizas una modificación a una clase Java se encarga de realizar un Dynamic Bynding o un Dynamic Loading para encontrar las clases. [Internet 24]

Para la obtención de un mayor provecho de la tecnología orientada a objetos, Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Esta característica ya es contemplada por Smalltalk, aunque no C++, que enlaza todos los módulos cuando se compila. [Internet 23]

Portabilidad

Un programa Java puede ser ejecutado en diferentes entornos, algo imposible para C++.
Portable. Como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el interprete de Java.[Internet 24]

Neutralidad

Se dice que Java tiene una arquitectura neutra puesto que compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.

Independiente a la arquitectura. Al compilar un programa en Java, el código resultante un tipo de código binario conocido como byte code. Este códido es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida. .[Internet 24]

Cualquier máquina que tenga el sistema de ejecución (*JRE* o *Java Runtime Enviroment*) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado.

Actualmente existen sistemas de ejecución (*JRE*) para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Mac y probablemente haya grupos de desarrollo trabajando el portado a otras plataformas.
No es así para C++ y para Smalltalk, donde el código generado podrá ejecutarse únicamente en la plataforma en la que se generó.

Threads

Un lenguaje que soporta múltiples hilos es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.
[Internet 24]

Java permite múltiples hilos (*multithreading*) antes de su ejecución y en tiempo de ejecución. La posibilidad de construir pequeños procesos o piezas independientes de un gran proceso permite programar de una forma más sencilla y es una herramienta muy potente que no se ofrece en C++.[Internet 23]

Recolección automática de basura (Garbage colection)

Java modifica completamente la gestión de la memoria que se hace en C/C++. En C/C++ se utilizan punteros, reservas de memoria (con las ordenes *malloc*, *new*, *free*, *delete*...) y otra serie de elementos que dan lugar a graves errores en tiempo de ejecución difícilmente depurables.

La recolección de basura (objetos ya inservibles) es una parte integral de Java durante la ejecución de sus programas. Una vez que se ha almacenado un objeto en el tiempo de ejecución, el sistema hace un seguimiento del estado del objeto, y en el momento en que se detecta que no se va a volver a utilizar ese objeto, el sistema vacía ese espacio de memoria para un uso futuro.

Esta gestión de la memoria dinámica hace que la programación en Java sea más fácil. [Internet 23]

Representación

Uno de los objetivos perseguidos en el desarrollo de Java era la obtención de programas con interfaces cómodas e intuitivas. Esto también se permite en C++, aunque con unos métodos más costosos, y en ningún caso con interfaces portables como los que Java crea.

Tanto en Java como en C++ se logran unas interfaces con una representación mejor que la que se puede alcanzar con Smalltalk. [Internet 23]

Lenguaje Seleccionado

Ya que el software desarrollado será utilizado en el Web y dados los requerimientos del usuario el lenguaje utilizado será java, ya que es un lenguaje de alto nivle orientado a objetos que cuenta con diferentes tecnologías entre las cuales se encuentran los Servlets y Java Server Pages(JSP).

Java

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc.),y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma.
[Internet 24]
Sun desarrolló Java, basado en C++ e iniciado como un lenguaje independiente de la plataforma con el que se podían crear aplicaciones denominadas Applets que operaban dentro del navegador Web. A diferencia de CGI, Java corre en la máquina cliente lo que significa que la conexión con el servidor no es permanente reduciendo considerablemente las necesidades de proceso. Java apareció asegurando portabilidad, orientación a objetos y reusabilidad de código. Bien en principio, pero como siempre las crecientes aplicaciones cliente – servidor con potentes interfaces de usuario obligaban a generar código demasiado especializado como para ser realmente portable.
[Internet 29]

Java funciona de la siguiente manera: El compilador de Java deja el programa en un □stil□m-código (no es código maquinal) y luego el intérprete de Java ejecuta el programa (lo que se conoce como el "Java Virtual □stil□m"). Por eso Java es multiplataforma, existe un intérprete para cada máquina diferente. *Nota: El código maquinal es el código binario que la computadora entiende y puede ejecutar.*
[Internet 24]

Selección de manejador de bases de datos

A la hora de decidir el manejador de base de datos sobre el que se desea trabajar para el proyecto, debemos saber que cada sistema de bases de datos tiene sus peculiaridades que lo convierten en el más adecuado según el tipo de proyecto a desarrollar o en función de los intereses del programador . [Internet 25]

A continuación se presentará una comparativa de los diferentes manejadores de base de datos, la cual fue utilizada para tomar la decisión de que manejador de base de datos utilizar en el proyecto:

Criterios	Bases de datos			
	Access	SQL Server	MySQL	PostgreSQL
Plataforma			/	/
Velocidad	↓	↑	↑	↓
Volumen Datos	↓	↑	↑	↑
Integridad	↓	↑	↓	↑
Potencia	↓	↑	↑	↑
Coste/MB	↑	↓	↑	↑

↑ Positivo ↓ Negativo

Bases de datos Access

Realizar una aplicación ASP**(Active Server Pages)** sobre bases de datos Access es recomendable en los casos en que sea especialmente cómoda la actualización de la información por el procedimiento de enviar el archivo .mdb al servidor mediante FTP.

Para ver más información sobre ASP consulte el glosario o la página: http://www.desarrolloweb.com/articulos/393.php?manual=15

Ahora bien, para que la aplicación sobre base de datos Access no tenga problemas, es recomendable que cumpla estas condiciones:

El volumen de datos a manejar es pequeño. (Además así será más rápida su actualización por FTP).

El número de visitantes simultáneos no es muy alto.

La aplicación ASP no cambia la base de datos, simplemente muestra datos. Esto es consistente con el hecho de enviar periódicamente el archivo .mdb al servidor, pues si la aplicación ASP cambiase la base de datos, esos cambios se perderían al sobrescribirse con la nueva base de datos.

Cuando por alguno de los anteriores motivos, su aplicación no es consistente, o no va a poder cumplir alguno de los mismos, es preferible el uso de un sistema de base de datos más robusto, como SQL Server.

Bases de datos MS SQL Server

SQL Server es el sistema de bases de datos más completo y potente y resulta ideal para los programadores especializados en productos Microsoft: ASP, Visual Basic, modelos de objetos componentes, etc. Además, es un sistema de base de datos perfectamente adecuado para aplicaciones críticas y con cualquier grado de complejidad.

SQL Server utiliza una parte del espacio de la base de datos para guardar el log de transacciones con los comandos pendientes, lo que asegura que, independientemente de si el programador usa o no transacciones en su código, en ningún caso la base de datos quedaría en un estado inconsistente debido a una ejecución parcial de comandos.

También ofrece otras muchas características avanzadas orientadas a mantener la integridad de la base de datos, como son los triggers, y ofrece soporte completo ACID (Atomicity Consistency Isolation Durability).

Bases de datos MySQL

MySQL tiene como principales características su velocidad. Es el servidor de bases de datos más rápido de todos .
MySQL es muy utilizado en aplicaciones PHP o Perl en servidores Linux. En general, si no necesita características como transacciones, procedimientos almacenados, triggers o sentencias SQL complejas, MySQL cumplirá la misma función que otras bases de datos más potentes, pero de forma más rápida .
Para aplicaciones Windows, MySQL es una alternativa económica a SQL Server, y además se puede aprovechar todo el espacio para datos, mientras que SQL Server necesita una parte del espacio para el log de transacciones.

Las limitaciones de MySQL vienen dadas por sus carencias respecto de los otros sistemas de bases de datos y por el grado de criticidad de su aplicación. MySQL no es adecuada para aplicaciones críticas. Al no utilizar transacciones, un problema de cualquier tipo que interrumpiese una serie de comandos podría dejar la base de datos en un estado inconsistente, lo cual nunca ocurriría con SQL Server o PostgreSQL. Tampoco tiene triggers por lo que no se pueden establecer reglas de integridad y consistencia a nivel de servidor. [Internet 25]

- Su principal objetivo de diseño fue la VELOCIDAD. Se sacrificaron algunas características esenciales en sistemas más “serios” con este fin.
- Otra característica importante es que consume MUY POCOS RECURSOS, tanto de CPU como de memoria.
- Licencia GPL a partir de la versión 3.23.19.

- **Ventajas:**
- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla “en caliente”.

- **Inconvenientes:**

- No soporta transacciones, “roll-backs” ni subselects.
- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.
- [Internet 27]

Bases de datos PostgreSQL

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

PostgreSQL es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix.

Para aplicaciones Windows, PostgreSQL es una alternativa económica a SQL Server. Esta diferencia económica es especialmente sustancial si se necesita un Servidor Dedicado de bases de datos.

La mayor limitación de PostgreSQL viene dada por su velocidad: es el sistema de bases de datos más lento.
[Internet 25]

- Postgres es una manejador de bases de datos de mayor nivel que MySQL, a la altura de Oracle o Sybase.
- Licencia BSD.

- **Ventajas:**

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.
- Tiene mayor capacidad de almacenamiento que cualquier otro manejador de base de datos de software libre.

- **Inconvenientes:**
- Consume BASTANTES más recursos y carga más el sistema.
- Es de 2 a 3 veces más lenta que MySQL.

En general, sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante (pueda bastar con hacer backups periódicos que se restaurarán tras una caida del servidor). En cambio, para sistemas más serios en las que la consistencia de la BD sea **fundamental** (BD con información realmente importante, bancos, etc.) PostgreSQL es una mejor opción pese a su mayor lentitud. [Internet 27]

Por lo tanto el Manejador de Base de datos a utilizar en este proyecto será PostgreSQL ya que lo que nos interesa es el buen almacenamiento de la información y la seguridad de la información que se va a manejar.

Manejador de Base de Datos Seleccionado

PostgreSQL es un sistema manejador de base de datos orientado a objetos, basado en POSTGRES, Versión 4.2, desarrollado en la Universidad de California en el departamento de Ciencias de la Computación en Berkeley. POSTGRES inició muchos conceptos que

sólo llegaron a estar disponibles en algunos sistemas de bases de datos comerciales mucho después.
PostgreSQL es un software libre descendiente de su código original de Berkeley. Ofrece muchas herramientas modernas como:

- queries complejos
- Llaves foráneas
- Triggers
- Vistas
- transactional integrity
- multiversion de control de concurrencia

Además, PostgreSQL puede ser extendido por el usuario de varias formas, por ejemplo agregando:

- Tipos de datos
- Funciones
- Operadores
- Funciones agregadas
- Etc.

Capítulo 6

6.Programación

Una vez seleccionadas las herramientas a utilizar para la realización del sistema , se procedió a las instalaciones y configuraciones del software seleccionado en el capítulo anterior. Éste capítulo se enfocará a explicar la parte de programación realizada en el sistema, más adelante se hará una descripción de las instalaciones y configuraciones del software necesario.
Con una de las Clases desarrolladas en el sistema se explicará la estructura que lleva cada una de las clases y los métodos desarrollados en la misma. Para ello primero se dará una explicación de la estructura de un JavaBean puesto que se ocupan clases de este tipo. Ocuparé como referencia todo lo que se desarrolló para hacer la parte de Registro de Instituciones.

Clase JavaBean

`package sistema;`	En ésta parte empaqueto la clase en un directorio llamado sistema, el cual tiene como ruta: `/usr/local/jakarta-tomcat-4.1.2/webapps/AtencionVisitante/WEB-INF/classes/sistema`

`import java.io.* ;`
`import java.sql.* ;`
`import java.util.* ;`
`import java.util.Date ;`
`import java.sql.Timestamp;`
	Aquí mando a llavar a otra clase llamada GeneraClaveRegistro, la cual se encuentra en el mismo directorio ¨sistema¨.

`Public class AgregaInstitucionBean implements Serializable` `{`	Aquí se define la clase JavaBean con el nombre “AgregaInstitucionBean” seguido debe de tener “implements Serializable” para poder cubrir con los requisitos para ser una clase JavaBean

`private int id_insttcion ;`
`private int id_tpo_insttcion;`
`private Timestamp fcha_rgstro;`
 	
`private String clve_rgstro="";`
`private String nmbre_insttcion="";`
`private String dmclio_insttcion="";`
`private String clnia_insttcion="";`
`private String dlgcion_insttcion="";`
`private int tlfno_insttcion;`
`private int lda_insttcion;`
`private int fax_insttcion;`
`private String crreo_insttcion="";`
`private String dscrpcion_tpo_insttcion="";`
`private String entdad_fdrtva = "" ;`
`private String clve_sep = "";`
	En ésta parte hago la declaración de la variables cuyo valor va a ser requerido por las Clases y métodos que más adelante se explicarán.

`Public void setId_insttcion(int valor)`
`{`
`this.id_insttcion=valor;`
`}`
`public void setId_tpo_insttcion(int valor)`
`{`
	En ésta parte del JavaBean se crean los métodos SET y GET, para poder colocar y obtener los valores de la variables o atributos de un objeto creado a partir de la clase JavaBean.

```
this.id_tpo_insttcion=valor;
 }
public void setFcha_rgstro(Timestamp valor)
{
this.fcha_rgstro=valor;
}
public void setClve_rgstro(String valor)
{
this.clve_rgstro=valor;
}
public void setNmbre_insttcion(String valor)
{
this.nmbre_insttcion=valor;
 }
public void setDmclio_insttcion(String valor)
{
this.dmclio_insttcion=valor;
}
public void setClnia_insttcion(String valor)
{
this.clnia_insttcion=valor;
}
public void setDlgcion_insttcion(String
valor)
{
this.dlgcion_insttcion=valor;
}
public void setTlfno_insttcion(int valor)
{
this.tlfno_insttcion=valor;
}
public void setLda_insttcion(int valor)
{
this.lda_insttcion=valor;
}
public void setFax_insttcion(int valor)
{
this.fax_insttcion=valor;
}
public void setCrreo_insttcion(String valor)
{
this.crreo_insttcion=valor;
}
public void setDscrpcion_tpo_insttcion(String
valor)
{
this.dscrpcion_tpo_insttcion=valor;
}
public void setEntdad_fdrtva(String valor)
{
this.entdad_fdrtva= valor;
}
public void setClve_sep(String valor)
{
this.clve_sep = valor;
}
public int getId_insttcion()
{
return this.id_insttcion;
}
public int getId_tpo_insttcion()
{
return this.id_tpo_insttcion;
```

```
}
public Timestamp getFcha_rgstro()
{
return this.fcha_rgstro;
}
public String getNmbre_insttcion()
{
return this.nmbre_insttcion;
}
public String getClve_rgstro()
{
return this.clve_rgstro;
}

public String getDmclio_insttcion()
{
return this.dmclio_insttcion;
}
public String getClnia_insttcion()
{
return this.clnia_insttcion;
}
public String getDlgcion_insttcion()
{
return this.dlgcion_insttcion;
}
public int getTlfno_insttcion()
{
 return this.tlfno_insttcion;
 }
public int getLda_insttcion()
{
return this.lda_insttcion;
}
public int getFax_insttcion()
{
 return this.fax_insttcion;
}
public String getCrreo_insttcion()
{
return this.crreo_insttcion;
}
public String getDscrpcion_tpo_insttcion()
{
return this.dscrpcion_tpo_insttcion;
}
public String getEntdad_fdrtva()
{
return this.entdad_fdrtva;
}
public String getClve_sep()
{
return this.clve_sep;
}
}
```

A continuación se dará una explicación detallada de la estructura de una clase java , en este caso de la clase de AgregaInstitucionBusiness.

`Package sistema;`	En ésta parte empaqueto la clase en un directorio llamado sistema, el cual tiene como ruta: `/usr/local/jakarta-tomcat-4.1.2/webapps/AtencionVisitante/WEB-INF/classes/sistema`
`import sistema.GeneraClavesRegistro;`	Aquí mando a llavar a otra clase llamada GeneraClaveRegistro, la cual se encuentra en el mismo directorio ¨sistema¨.

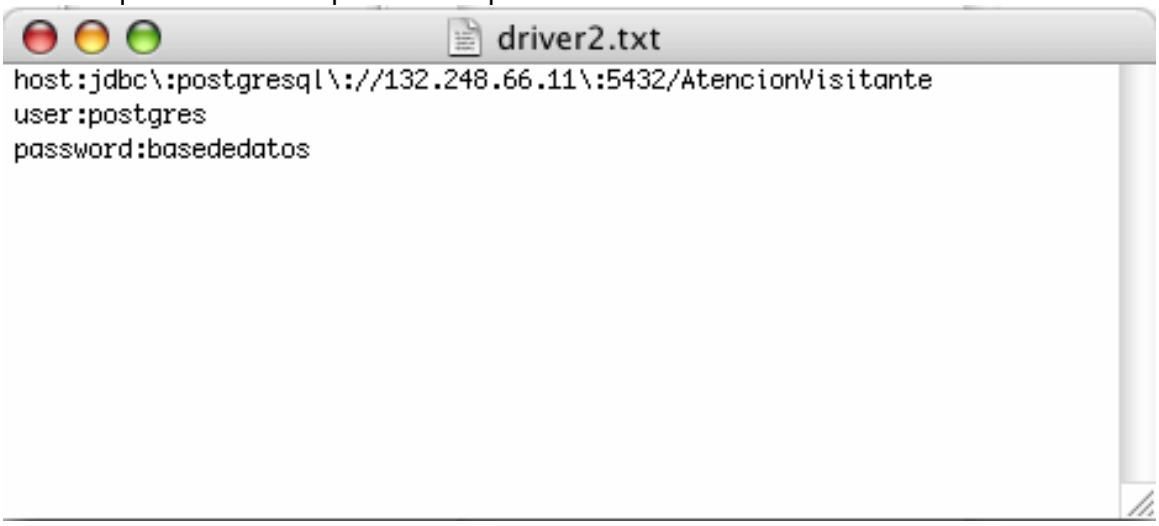
`Import java.io.* ;`
`import java.util.* ;`
`import java.sql.* ;`
`import java.sql.Timestamp ;`
	Aquí importo las librerías necesarias para la utilización de algunos de los métodos que se utilizarán en la parte de programación.

`Public class AgregaInstitucionBusiness` `{`	Aquí creo la clase “AgregaInstitucionBusiness”, la cual es creada como pública debido a que tiene que ser vista por otras clases.

`Leearchivo lee = new Leearchivo();`
`Properties property = lee.claves();`
 	
`String host = property.getProperty(“host”);`
`String user = property.getProperty(“user”);`
`String password = property.getProperty(“password”);`
	En esta parte se manadan a llamar los parámetros necesarios desde un archivo para que se lleve a cabo la conexión con la base de datos , más adelante veremos el contenido del archivo que contiene los parámetros.

`Public int insertaInstitucion(AgregaInstitucionBean agreinst, int respuesta)` `{`	Aquí creo el método “insertaInstitución”, el cual como su nombre lo indica, se encarga de hacer la inserción de los registros de las instituciones, éste metodo tiene como parámetros un objeto de tipo javabean perteneciente a la clase AgregaInstitucionBean, el cual será explicado a mas detalla más adelante, el otro parámetro que recibe en éste caso es un dato de tipo entero, el cual proviene del resultado de la ejecución de otra clase.
`Connection con = null;`	Aquí creo un objeto llamado “con” del tipo Connection.
`Int resultado2 = 0 ;`	Inicializo la variable de tipo entero llamado “resultado2”
`int claveregistro = genera.random();`	Creo una varible de tipo entero llamado “claveregistro”, la cual va a contener un “□sti” o un “false” dependiendo si se ejecutó exitósamente el método “random()” perteneciente a la clase “genera”
`try` `{`	Empleo los métodos “try” y “□stil” para cachar los errores que pudieran surgir a lo largo de la ejecución del método, esto para ubicar los errores.
`Class.forName(“org.postgresql.Driver”);`	Cargo el controlador del manejador de base de datos, en éste caso el jdbc para postgresql.
`Con = DriverManager.getConnection(host,user,password) ;`	Al objeto “con” le asigno los valores del controlador de conección, pasándole como parámetros la “ip” del servidor donde se encuentra instalado el manejador de base de datos, el directorio donde se están guardando todos los archivos *.java y *.class, el nombre de usuario asignado en el servidor(postgres), y la contraseña para acceder a la base de datos.
`PreparedStatement pstmt = con.prepareStatement(“INSERT INTO Insttcion (id_tpo_insttcion,nmbre_insttcion,dmclio_insttcion, clnia_insttcion,dlgcion_insttcion,tlfno_insttcion, lda_insttcion,fax_insttcion,crreo_insttcion, clve_rgstro,entdad_fdrtva,clve_sep)`	Aquí se le pasan los parámetros necesarios al método prepareStatement para poder hacer la inserción de los datos particulares de la institución que desea realizar una visita al museo.

Archivo que contiene los parámetros para la conexión a la base de datos:



A continuación veremos el JSP ocupado para la captura de los datos requeridos para el registro de una institución.

```
<%@ include file="ValidaSessionHeader.jsp" %>

<html>
<script language="javascript">
function emailCheck (emailStr) {

var emailPat=/^(.+)@(.+)$/
var specialChars="\\(\\)<>@,;:\\\\\\\"\\.\\[\\]"
var validChars="\[^\\s" + specialChars + "\]"
var quotedUser="(\"[^\"]*\")"
var ipDomainPat=/^\[(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})\]$/


var atom=validChars + '+'
var word="(" + atom + "|" + quotedUser + ")"
var userPat=new RegExp("^" + word + "(\\." + word + ")*$")
var domainPat=new RegExp("^" + atom + "(\\." + atom +")*$")

var matchArray=emailStr.match(emailPat)
if (matchArray==null) {
alert("La direccion de correo es incorrecta (verrifica la @ y los .'s)")
return false
}

var user=matchArray[1]
var domain=matchArray[2]

if (user.match(userPat)==null) {
alert("El nombre de usuario no es válido.")
return false
}

var IPArray=domain.match(ipDomainPat)
if (IPArray!=null) {
```

```
for (var i=1;i<=4;i++) {
if (IPArray[i]>255) {
alert("IP de destino inválida")
return false
}
}
return false
}

var domainArray=domain.match(domainPat)
if (domainArray==null) {
alert("El dominio parece no ser válido.")
return false
}

var atomPat=new RegExp(atom,"g")
var domArr=domain.match(atomPat)
var len=domArr.length
if (domArr[domArr.length-1].length<2 || domArr[domArr.length-1].length>3) {

alert("La dicrección debe tener 3 letras si es .'com' o 2 si en de algún pais.")
return false

}

if (len<2) {
            var errStr="La dirección es erronea"
            alert(errStr)
            return false
            }
return '1';
            }
             function valida(theForm){
        if (theForm.nmbre_insttcion.value=="")
                        {
                        alert ('Escriba un nombre de institución');
                        theForm.nmbre_insttcion.focus();
                        return false;
                        }

              if(theForm.nmbre_insttcion.value.length < 5)
                        {
                        alert ('El nombre de institución debe ser por lo menos de 5 caracteres');
                        theForm.nmbre_insttcion.focus();
                        return false;

                        }
        if (theForm.dmclio_insttcion.value=="")
                        {
                        alert ('Escriba una dirección');
                        theForm.dmclio_insttcion.focus();
                        return false;
                        }

              if(theForm.dmclio_insttcion.value.length < 5)
                        {
                        alert ('La direccion debe ser por lo menos de 5 caracteres');
                        theForm.dmclio_insttcion.focus();
                        return false;

                        }
```

```
 if (theForm.clnia_insttcion.value==””)
            {
            alert (‘Debe escribir el nombre de la colonia’);
            theForm.clnia_insttcion.focus();
            return false;
            }

  if(theForm.clnia_insttcion.value.length < 5)
            {
            alert (‘La colonia debe ser por lo menos de 5 caracteres’);
            theForm.clnia_insttcion.focus();
            return false;

            }

 if (theForm.dlgcion_insttcion.value==””)
            {
            alert (‘Escriba la delegación o municipio’);
            theForm.dlgcion_insttcion.focus();
            return false;
            }

  if(theForm.dlgcion_insttcion.value.length < 5)
            {
            alert (‘La delegación o municipio debe ser por lo menos de 5 caracteres’);
            theForm.dlgcion_insttcion.focus();
            return false;

            }

if(theForm.tlfno_insttcion.value!=””){
var checkOK = “0123456789”;
  var checkStr = theForm.tlfno_insttcion.value;
  var allValid = true;
  var validGroups = true;
  for (i = 0;  i < checkStr.length;  i++)
  {
    ch = checkStr.charAt(i);
    for (j = 0;  j < checkOK.length;  j++)
      if (ch == checkOK.charAt(j))
        break;
    if (j == checkOK.length)
    {
      allValid = false;
      break;
    }
  }
  if (!allValid)
  {
    alert(“Introduce solo numeros en el teléfono”);
    theForm.tlfno_insttcion.focus();
    return (false);
  }
}

if(theForm.lda_insttcion.value!=””){
var checkOK = “0123456789”;
  var checkStr = theForm.lda_insttcion.value;
  var allValid = true;
  var validGroups = true;
  for (i = 0;  i < checkStr.length;  i++)
```

```
  {
    ch = checkStr.charAt(i);
    for (j = 0;  j < checkOK.length;  j++)
      if (ch == checkOK.charAt(j))
        break;
    if (j == checkOK.length)
    {
      allValid = false;
      break;
    }
  }
  if (!allValid)
  {
    alert("Introduce solo numeros en lada");
    theForm.lda_insttcion.focus();
    return (false);
  }
}

if(theForm.fax_insttcion.value!=""){
var checkOK = "0123456789";
  var checkStr = theForm.fax_insttcion.value;
  var allValid = true;
  var validGroups = true;
  for (i = 0;  i < checkStr.length;  i++)
  {
    ch = checkStr.charAt(i);
    for (j = 0;  j < checkOK.length;  j++)
      if (ch == checkOK.charAt(j))
        break;
    if (j == checkOK.length)
    {
      allValid = false;
      break;
    }
  }
  if (!allValid)
  {
    alert("Introduce solo numeros en lada");
    theForm.fax_insttcion.focus();
    return (false);
  }
}


if (theForm.entdad_fdrtva.value=="")
            {
            alert ('Debe elegir una entidad de la lista');
            theForm.entdad_fdrtva.focus();
            return false;
            }

if (theForm.dscrpcionTpoInst.value=="")
            {
            alert ('Debe elegir un tipo de Institución');
            theForm.dscrpcionTpoInst.focus();
            return false;
            }


  if (theForm.crreo_insttcion.value!="")
```

```
                    {

                    var checa=emailCheck(theForm.crreo_insttcion.value);
                    if (checa==false) {

                    return checa;
                    }
                    }

             else
                    return true;
            }
           </script>

  <body bgcolor="#FFFFFF">
   <H2 ><center >
    <font color="#1EB1FA"> REGISTRO DE INSTITUCIONES </font>
  </center >
</H2>

   <form action=/AtencionVisitante/AgregaInstitucion2.jsp method=post onSubmit="return valida(this)"
name=form1>

   <div align="center">
   <table bgcolor="#85C9F6">
<tr><td align=center colspan=2>Proporcione los siguientes datos</td> <tr>
<td align=center>Institución </td>
<td><input type=text name=nmbre_insttcion size=50 maxlength=35></td>
</tr><tr> <td align=center>Domicilio de la Escuela</td>
 <td><input type=text name=dmclio_insttcion size=50 maxlength=35></td>
 </tr><tr><td align=center>Colonia</td>
<td><input type=text name=clnia_insttcion size=20 maxlength=20></td>
</tr><tr><td align=center>Delegacion o Municipio</td>
 <td><input type=text name=dlgcion_insttcion size=20 maxlength=20></td>
 </tr><tr><td align=center>Entidad Federativa</td>
<td>
<select name=entdad_fdrtva>
<option value="" selected>Elija una entidad</option>
<option value=AGUASCALIENTES>Aguascalientes</option>
 <option value=B.CALIFORNIA_NORTE>B.California Norte</option>
 <option value=B.CALIFORNIA_SUR>B.California Sur</option>
<option value=CAMPECHE>Campeche</option>
<option value=CHIAPAS>Chiapas</option>
<option value=CHIHUAHUA>Chihuahua</option>
<option value=COAHUILA>Coahuila</option>
<option value=COLIMA>Colima</option>
<option value=DISTRITO_FEDERAL>Distrito Federal</option>
<option value=DURANGO>Durango</option>
<option value=ESTADO_DE_MÉXICO>Estado de México</option>
<option value=GUANAJUATO>Guanajuato</option>
<option value=GUERRERO>Guerrero</option>
<option value=HIDALGO>Hidalgo</option>
<option value=JALISCO>Jalisco</option>
<option value=MICHOACÁN>Michoacán</option>
<option value=MORELOS>Morelos</option>
<option value=NAYARIT>Nayarit</option>
<option value=NUEVO_LEÓN>Nuevo León</option>

<option value=OAXACA>Oaxaca</option>
<option value=PUEBLA>Puebla</option>
<option value=QUERÉTARO>Querétaro</option>
<option value=QUINTANA_ROO>Quintana Roo</option>
```

```
<option value=SAN_LUIS_POTOSÍ>San Luis Potosí</option>
<option value=SINALOA>Sinaloa</option>
<option value=SONORA>Sonora</option>
<option value=TABASCO>Tabasco</option>
<option value=TAMAULIPAS>Tamaulipas</option>
<option value=TLAXCALA>Tlaxcala</option>
<option value=VERACRUZ>Veracruz</option>
<option value=YUCATÁN>Yucatán</option>
<option value=ZACATECAS>Zacatecas</option>
</select>
 </td></tr>
       <tr>
       <td align=center>Clave de registro ante la SEP(opcional)</td>
       <td><input type=text name=clve_sep size=20 maxlength=20></td> </tr>
       <tr>
       <tr> <td align=center>Teléfono</td>
      <td><input type=text name=tlfno_insttcion size=20 maxlength=20></td>
      </tr>
      <tr>
       <td align=center>Lada</td>
      <td><input type=text name=lda_insttcion size=20 maxlength=20></td>
     </tr>
     <tr><tr>
     <td align=center>Fax</td>
   <td><input type=text name=fax_insttcion size=20 maxlength=20></td>
  </tr>
  <tr><tr>
   <td align=center>Correo Electrónico</td>
 <td><input type=text name=crreo_insttcion size=20 maxlength=50></td>
 </tr><tr>
<tr>
<td align=center>Tipo de Institución :</td><td>
<select name=dscrpcionTpoInst>
    <option value="" selected>Elija un tipo</option>
   <option value=Oficial>Oficial</option>
   <option value=Particular>Particular</option>
  <option value=SEP>SEP</option>
   </select>
 </td>
   </table><br>
 <br>
   <CENTER><input type=submit value=Enviar>
  <input type=reset value=Limpiar>
  </CENTER>

  </div>

   <h5 align="center"> I have been invoked by
   <% out.print (request.getAttribute("servletName").toString());
  %>Desde AtencionVisitante </h5>
</form>
 </body>
</html>
<%@ □stil□m file="ValidaSessionFooter.jsp" %>
```

A continuación veremos el JSP ocupado para la ejecución de los métodos y clases explicadas anteriormente, para éste caso verémos el JSP "AgregaInstitucion2.jsp"

<%@ □stil□m file="ValidaSessionHeader.jsp" %> Aquí se hace la validación de las sesiones de Usuario, de no haber

iniciado sesión para ejecutar este jsp, el sistema no dejará que continúe , ésta validación termina con la última etiqueta <%@ □stil□m file="ValidaSessionFooter.jsp" %>

```
<%@ page import="sistema.* " %>
<%@ page import="java.util.*" %>
<%@ page import="java.sql.*" %>
```

Aquí se mandan a llamar clases necesarias para la ejecución de métodos que se utilizarán

```
<jsp:useBean id="AgregaInstitucionBean" class="sistema.AgregaInstitucionBean" scope="request">
<jsp:setProperty name="AgregaInstitucionBean" property="*"/>
</jsp:useBean>
```

Se asignan los valores a los métodos SET y GET ubicados en la clase "AgregaInstitucionBean"

```
<html>
<head>
<title>Inserta una □stil□mació en la base</title>
</head>
<body>
<center>
  <jsp:useBean id="ad3" scope="page" class="sistema.AgregaInstitucionBusiness3"/>
```

Se instancia la clase "AgregaInstitucionBusiness3" como ad3, el cual nos va a ayudar a ejecutar los métods de dicha clase

```
   <table>
  <%
          String tpoInst = request.getParameter("dscrpcionTpoInst");
          int respuesta = ad3.idInstitucion(tpoInst);
          if(respuesta != 0)
          {
%>
 <jsp:useBean id="ad2" scope="page" class="sistema.AgregaInstitucionBusiness"/>
 <%
          int prueba2=ad2.insertaInstitucion(AgregaInstitucionBean, respuesta);
           if (prueba2==1)
            {
              String clave = ad2.claveInstitucion();
              if(clave ¡= null)
              {
               out.println("<td><tr>El registro se ha realizado con éxito los datos registrados son: </b></td><tr>");
               out.println("<td><tr>Institución:<b>"+request.getParameter("nmbre_insttcion")+"</b></td><tr>");
               out.println("<td><tr>Domicilio: <b>"+request.getParameter("dmclio_insttcion")+"  </b></td></tr>");
               out.println("<td><tr>Colonia:<b> "+request.getParameter("clnia_insttcion")+"  </b><tr></td>");

out.println("<td><tr>DelegaciónoMunicipio:<b>"+request.getParameter("dlgcion_insttcion")+"<tr></td>");
               out.println("<td><tr>Teléfono(s) :<b> "+ request.getParameter("tlfno_insttcion") +" </b></td><tr>");
               out.println("<td><tr>Lada: <b>"+request.getParameter("lda_insttcion")+"  </b></td></tr>");
               out.println("<td><tr>Fax:<b> "+request.getParameter("fax_insttcion")+"  </b><tr></td>");
               out.println("<td><tr>Entidad Federativa:<b> "+request.getParameter("entdad_fdrtva")+" </b><tr></td>");
               out.println("<td><tr>Correo Electrónico:<b> "+request.getParameter("crreo_insttcion")+" </b><tr></td>");
               out.println("<td><tr>Tipo de Institución:<b> "+request.getParameter("dscrpcionTpoInst")+" </b><tr></td>");
                out.println("<td><tr>Entidad Federativa:<b> "+request.getParameter("entdad_fdrtva")+" </b><tr></td>");
```

Se ejecuta el método "idInstitucion"

```
                out.println("<td><tr>Clavederegistroante la SEP:<b>"+request.getParameter("clve_sep")+"
</b><tr></td>");
                out.println("<br><br><td><tr>Su CLAVE de REGISTRO es :<b> "+clave+"  </b></td><tr>");
               }
        }
 else
    out.println("No se pudo registrar la □stil□mació por business<a
□sti=javascript:history.back(2)>Regresar</a>");
       }
      else
       out.println("No se pudo registrar la □stil□mació por la selección del TIPO DE INSTITUCION<a
□sti=javascript:history.back(2)>Regresar</a>");
%>
</table></body>
</html>
<%@ □stil□m file="ValidaSessionFooter.jsp" %>
```

Hasta el momento he explicado la estructura de las clases y métodos que se utilizaron para la parte de Registro de Instituciones, así como de las etiquetas que se utilizaron en los JSP´s , por lo tanto considero que no será necesario explicar las subsecuentes, debido a que todas siguen la misma estructura, la única diferencia son las funciones que realiza cada uno.

Capítulo 7

7. Pruebas de Instalación

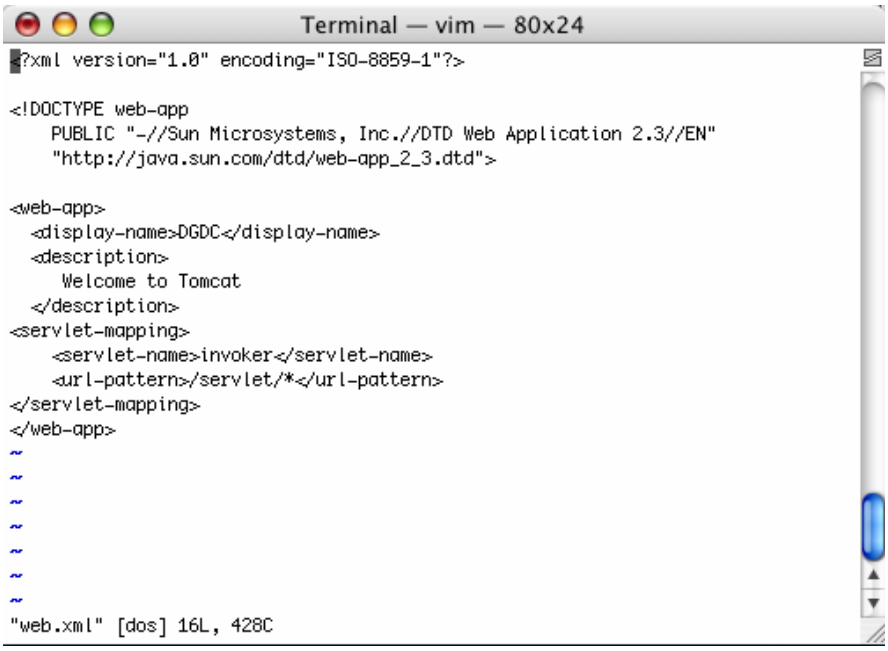
Para el desarrollo del sistema se hicieron instalaciones similares tanto en el servidor Linux como en la terminal de trabajo Mac OSX donde se desarrollo todo el sistema, dado que el software de instalación fue el mismo lo único que se hizo fue copiar los archivos de configuración de Tomcat que estaban en la Mac y pasarlos al servidor Linux en sus respectivos directorios de Tomcat en Linux , por lo que la migración del sistema de la Terminal al servidor Linux no tuvo problema alguno, las únicas modificaciones que tuvieron mayor relevancia fueron las configuraciones del archivo bash_profile en Linux ya que los directorios donde fue instalado el software fue diferente para ambas plataformas.

7.1 Requerimientos de Hardware y Software

Requerimientos de Software

Instalaciones y configuraciones en OSX.

1. Se descarga de http://jakarta.apache.org el paquete jakarta-tomcat-4.1.2.tar.gz, se descomprime el archivo, se mueve el directorio creado a /usr/local
2. Una vez hecho esto se crean un directorio llamado "AtencionVisitante" dentro del directorio "webapps".
3. En el directorio AtencionVisitante se guardan todos los JSP´S del sistema.
4. Para que los servlets sean reconocidos por tomcat se tuvo que configurar el archivo web.xml, cuyo contenido se muestra en la siguiente figura:

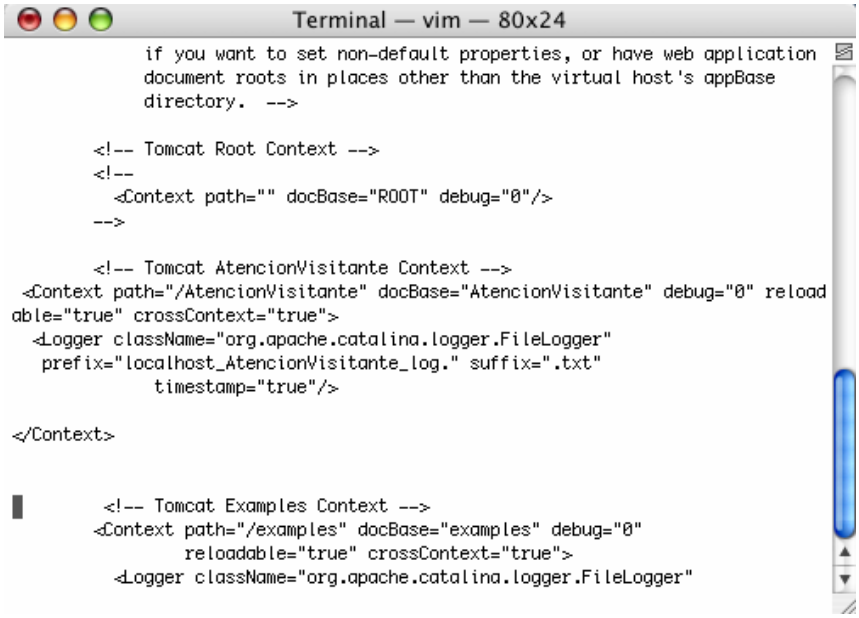


5. Después se creó un directorio llamado WEB-INF y classes dentro del último , y dentro de classes se creó otro llamado sistema, cuyo nombre corresponde al directorio donde se estuvieron empaquetando todas las clases.

6. En el directorio “sistema” se almacenaron todas las clases java.
7. Para que Tomcat pudiera reconocer el directorio donde tenemos todo nuestro sistema se configuró el archivo “□stil□.xml” contenido en:

   /usr/local/jakarta-tomcat-4.1.2/conf

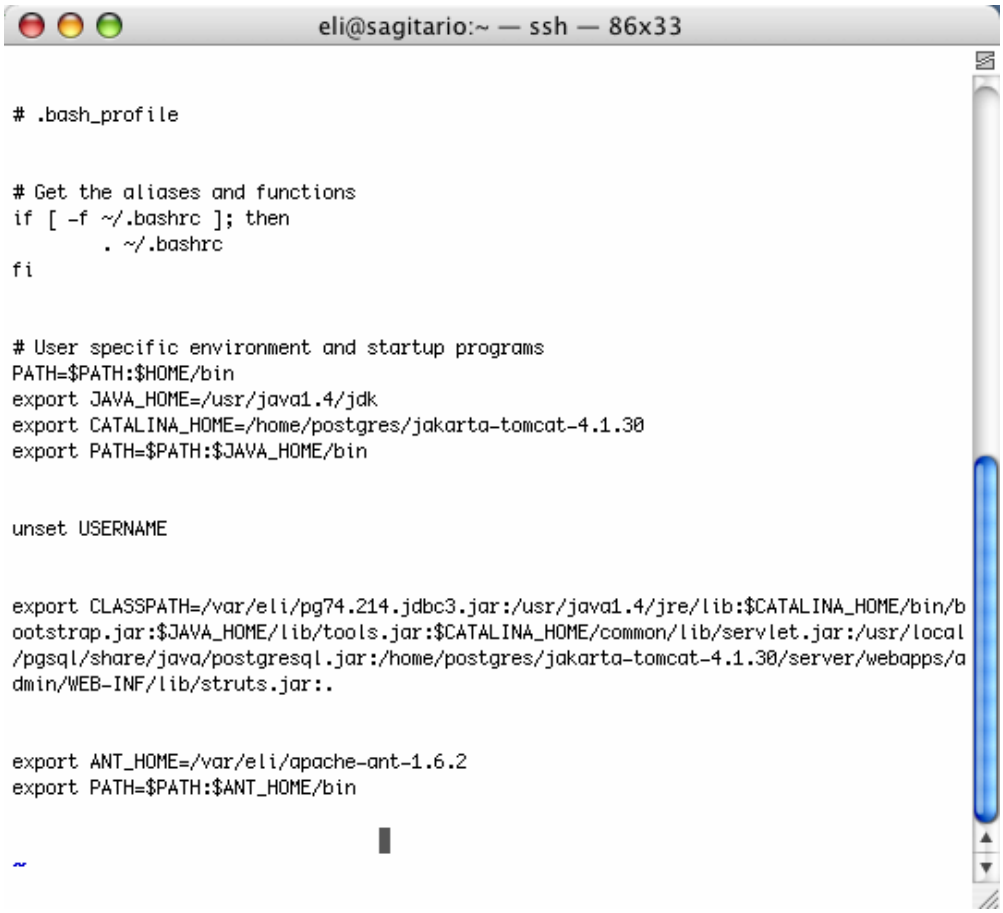
   agregándole las siguientes líneas según muestra la siguiente figura:



8. En cuanto a las configuraciones de java en Mac OSX no se tuvo ninguna complicación ya que como Mac fue diseñado para desarrolladores ya se tenía instalado por default java

Instalaciones y configuraciones en el servidor Linux

1. De la página http://postgresql.org se descarga el paquete postgresql-7.4.tar
2. Después se descargó el paquete apache-ant-1.6.2-bin.tar de la página http://jakarta.apache.org
3. Después se descargo el paquete j2eesdk-1_4_01-linux.bin para instalar java
4. Después se descargó el paquete jwsdp-1_4-unix.sh , estos dos últimos de la página http://sun.com
5. La última descarga que se hizo fue la del paquete pg74.214.jdbc3.jar obtenida de la página de la página http://postgresql.org
6. Ya con todas estas instalaciones se hizo una modificación al bash_profile del servidor , cuyo contenido fue el siguiente:



7. Una vez realizados los pasos anteriores el manejador de base de datos se instaló siguiendo los siguientes pasos :

tar –cxvz postresql-7.4.tar
./configure –with-java
gmake
su
gmake install
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su – postgres
/usr/local/pgsql/bin/initdb –D /usr/local/pgsql/data
/usr/local/pgsql/bin/postmaster –D /usr/local/pgsql/data >logfile 2>&1 &
/usr/local/pgsql/bin/createdb AtencionVisitante
/usr/local/pgsql/bin/psql AtencionVisitante

Requerimientos de Hardware

Para que el sistema funcione en óptimas condiciones los requisitos mínimos que deben de cumplir los equipos de cómputo son los siguientes:

Servidor Linux

PC procesador Pentium III, mínimo a 500Mhz, mínimo 256 RAM

Clientes

Tener instalado cualquier browser como: Internet Explorer, Mozilla, Safari, NetScape
Memoria en Ram 128 Mb.
Procesador Pentium o Superior.
Tarjeta de red PCI 10/100.

CONCLUSIONES

Se consideró la problemática que enfrenta Universum para la realización de reservaciones de visitas guiadas, esto se llevó a cabo con la continua interacción que se tuvo con el usuario final del sistema, el cual corresponde al área de Atención al Visitante. Gracias a dicha interacción se pudo hacer un análisis de un sistema automatizado con la capacidad de satisfacer con sus requerimientos.

Se realizó un diseño del sistema basado en los requerimientos del usuario, dicho diseño tiene como ventaja el poder ser comprendido por cualquier desarrollador debido a que se utilizó el Lenguaje de Modelado Unificado (UML).

Se desarrolló el sistema en un lenguaje de programación que cumpliera como característica primordial el poder ser interpretado para las plataformas más comerciales, como lo es Java, esto debido a que Universum utiliza varias plataformas como lo es Mac, Windows y Linux en su equipo de cómputo, además de que el uso de ésta herramienta de software (Java) ayuda al museo a reducir costos de mantenimiento para el sistema debido a que Java funciona también bajo licencia pública GNU.

Con todo lo realizado se implementó un sistema computacional para la reservación de visitas guiadas en el museo de las ciencias Universum, que permite efectuar reservaciones a las diferentes instituciones de la República Mexicana, agilizando la manera en la que se efectúan las reservaciones, automatizando, controlando y almacenando toda la información recabada, con esto se logró llevar un mejor seguimiento de las instituciones que han realizado reservaciones para visitas guiadas sustituyendo las papeletas con las que se contaba . Así mismo se implantó un nuevo manejador de base de datos bajo licencia pública GNU con grandes capacidades de almacenamiento para llevar un mejor control de su información.

Una de las principales ventajas obtenidas en este sistema, indiscutiblemente, es el tiempo en el que se llevan a cabo las reservaciones, dicha optimización se debe principalmente a que toda la información se encuentra alojada en un servidor, lo cual no sólo facilita su búsqueda y disponibilidad.

Otra de las ventajas que conlleva la utilización de este sistema, la cual permitirá aun más su crecimiento, es el diseño y la creación de la base de datos que se utilizó, pues gracias a ello el sistema tiene la capacidad de absorber otros sistemas que puedan ser creados a futuro .

Una última ventaja, pero no menos importante, que ofrece éste sistema es que gracias a la información que almacena, no solamente contribuye con el área de Atención al Visitante para la realización de reservaciones sino que también ayuda al área de Becarios para saber cuántas visitas se realizarán por día y así saber cuántos becarios deben tener disponibles para las explicaciones en las diferentes salas con las que cuenta el museo. Adicionalmente ayuda a taquilla para saber cuántos boletos debe reservar para una determinada fecha

GLOSARIO

ASP

Es el lenguaje de scripting del lado del servidor creado por Microsoft.

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).
[Internet 26]

Datagrama
Son los datos del usuario
[Internet 11]

RFC

RFC - Request For Comment, *Petición de comentarios*, son una serie de documentos o informes técnicos que edita la IAB (*Internet Architecture Board* - Consejo de Arquitectura de Internet), con el propósito de regular los estándares y procedimientos de estandarización en el Internet. En esta sección se incluyen las ligas a los documentos que se encuentran disponibles en el servidor IIO-2, al final se incluye una lista de sitios en donde pueden buscarse mas documentación. Algunos documentos no se encuentran disponibles, pero incluimos el nombre de dicho documento, para que pueda buscarlo en algun sitio automatizado.
[Internet 31]

Gateway(pasarela)

Si bien es cierto que el término router (enrutador) ha desplazado a la definición original de gateway diremos que se trata de un dispositivo de comunicaciones que sirve como enlace entre redes que funcionan de manera similar, pero que sin embargo cuentan con implantaciones diferentes.

Mainframe

Macrocomputador. En la actualidad se utiliza esta palabra para referirse a los grandes ordenadores. Un ejemplo típico sería la arquitectura 390 de IBM. Es decir, máquinas capaces de gestionar muchos terminales y unidades periféricas (Ver: Periférico) de memoria con capacidad para varios gigabytes. Con el aumento de potencia de los llamados miniordenadores, la frontera entre éstos y los mainframes está cada vez menos clara. Originalmente, mainframe no era sino el armario metálico que contenía la unidad central de los grandes ordenadores.

PDU

El protocolo UDP (*User Datagram Protocol*, protocolo de datagrama de usuario) proporciona una comunicación muy sencilla entre las aplicaciones de dos ordenadores. Al igual que el protocolo IP, UDP es:

No orientado a conexión. No se establece una conexión previa con el otro extremo para transmitir un mensaje UDP. Los mensajes se envían sin más y éstos pueden duplicarse o llegar desordenados al destino.

No fiable. Los mensajes UDP se pueden perder o llegar dañados.

UDP utiliza el protocolo IP para transportar sus mensajes. Como vemos, no añade ninguna mejora en la calidad de la transferencia; aunque sí incorpora los puertos origen y destino en su formato de mensaje. Las aplicaciones (y no el protocolo UDP) deberán programarse teniendo en cuenta que la información puede no llegar de forma correcta.

- **Puerto UDP de origen** (16 bits, opcional). Número de puerto de la máquina origen.

- **Puerto UDP de destino** (16 bits). Número de puerto de la máquina destino.

- **Longitud del mensaje UDP** (16 bits). Especifica la longitud medida en bytes del mensaje UDP incluyendo la cabecera. La longitud mínima es de 8 bytes.

- **Suma de verificación UDP** (16 bits, opcional). Suma de comprobación de errores del mensaje. Para su cálculo se utiliza una *pseudo-cabecera* que también incluye las direcciones IP origen y destino. Para conocer estos datos, el protocolo UDP debe interactuar con el protocolo IP.

- **Datos**. Aquí viajan los datos que se envían las aplicaciones. Los mismos datos que envía la aplicación origen son recibidos por la aplicación destino después de atravesar toda la Red de redes.

Servlets

Esta tecnología se ejecuta dentro de la máquina virtual de Java junto con el servidor Web por lo cual no requiere de la descarga de u programa externo. Debido a que Java cuenta con la capacidad de descargar clases en el tiempo de ejecución. Los Servlets pueden ser utilizados en el momento que se soliciten.

Los servlets permiten tener un control de la información que se va a desplegar en el HTML, demás utiliza como medio de presentación documentos HTML. Otra ventaja es que los servlets se encuentran en el servidor y no saturan la máquina cliente. Una posible desventaja esel tiempo que tardan en desplegarse las páginas HTML por medio de los servlets.

TRAMA

Formato de la trama

La comunicación entre una estación y otra a través de una red Ethernet se realiza enviando tramas Ethernet. El mensaje que se quiere transmitir se descompone en una o más tramas con el siguiente formato:

8 bytes	6 bytes	6 bytes	2 bytes	64-1500 bytes	4 bytes
Preámbulo	Dirección física destino	Dirección física origen	Tipo de trama	Datos de la trama	CRC

Las *direcciones origen* y *destino* son las direcciones físicas de los adaptadores de red de cada ordenador. El campo *Tipo de trama* indica el formato de los datos que se transfieren en el campo *Datos de la trama*. Por ejemplo, para un datagrama IP se utiliza el valor hexadecimal de 0800 y para un mensaje ARP el valor 0806. Todos los mensajes (*datagramas*) que se envíen en la capa siguiente irán encapsulados en una o más tramas Ethernet utilizando el campo *Datos de la trama*. Y esto mismo es aplicable para cualquier otro tipo de red distinta a Ethernet. Como norma general, cada mensaje que transmite una capa se coloca en el campo datos de la capa anterior. Aunque es muy frecuente que el mensaje no quepa en una sola trama y se utilicen varias.

TRIGGER

TRIGGER o disparador. Se define así a una subrutina que es ejecutada de manera automática cuando se produce algún tipo de transacción (inserción, borrado o actualización) en la tabla de una base de datos.

APENDICE A

Fig.1. Pantalla de inicio del sistema.

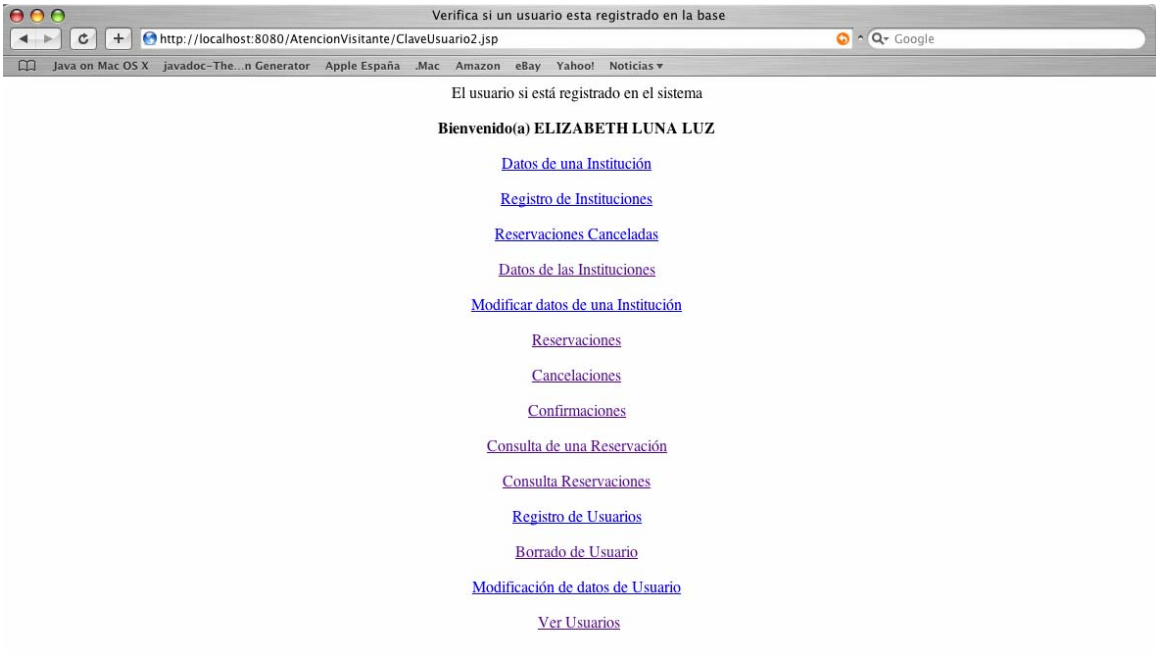


Fig2. Menú principal del sistema a nivel de administrador.

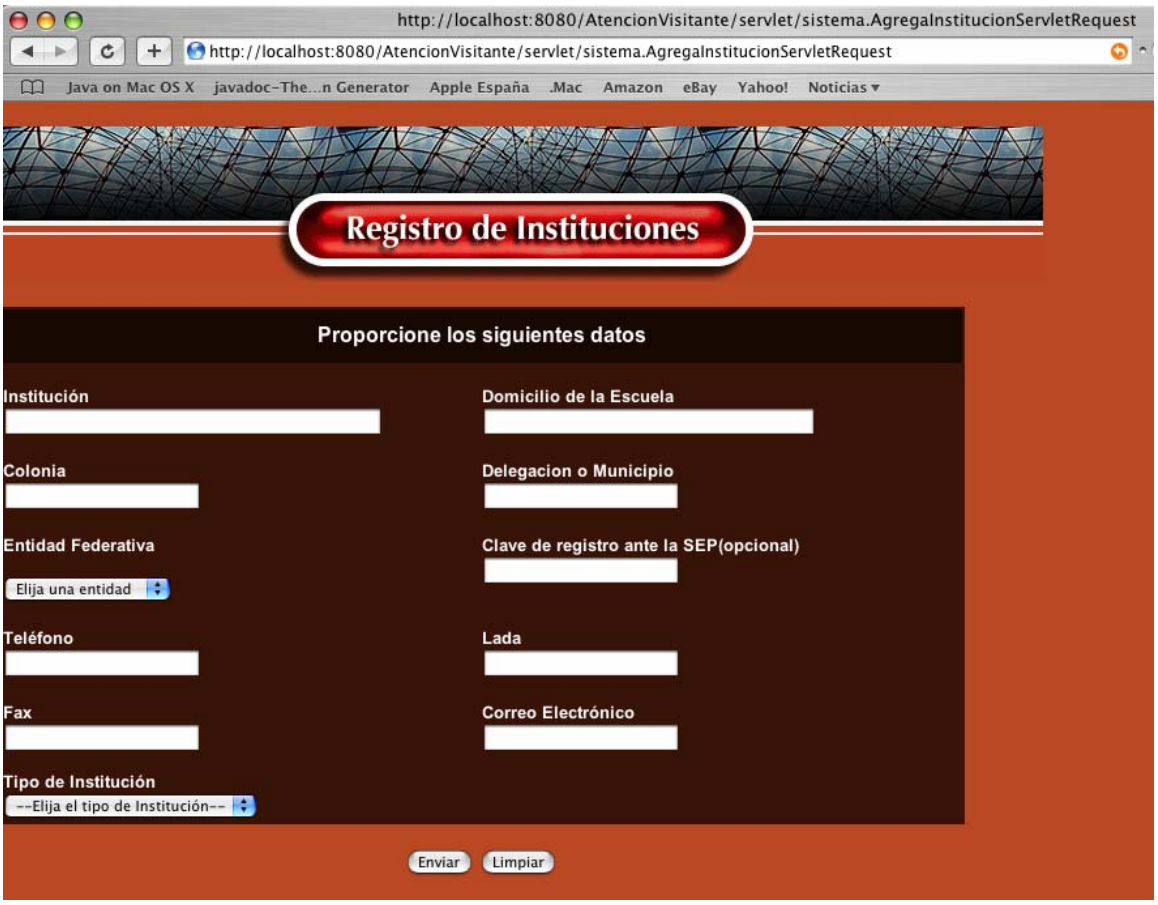


Fig3. Pantalla de registro de instituciones.

Fig.4. Pantalla inicial para la modificación del registro de instituciones.

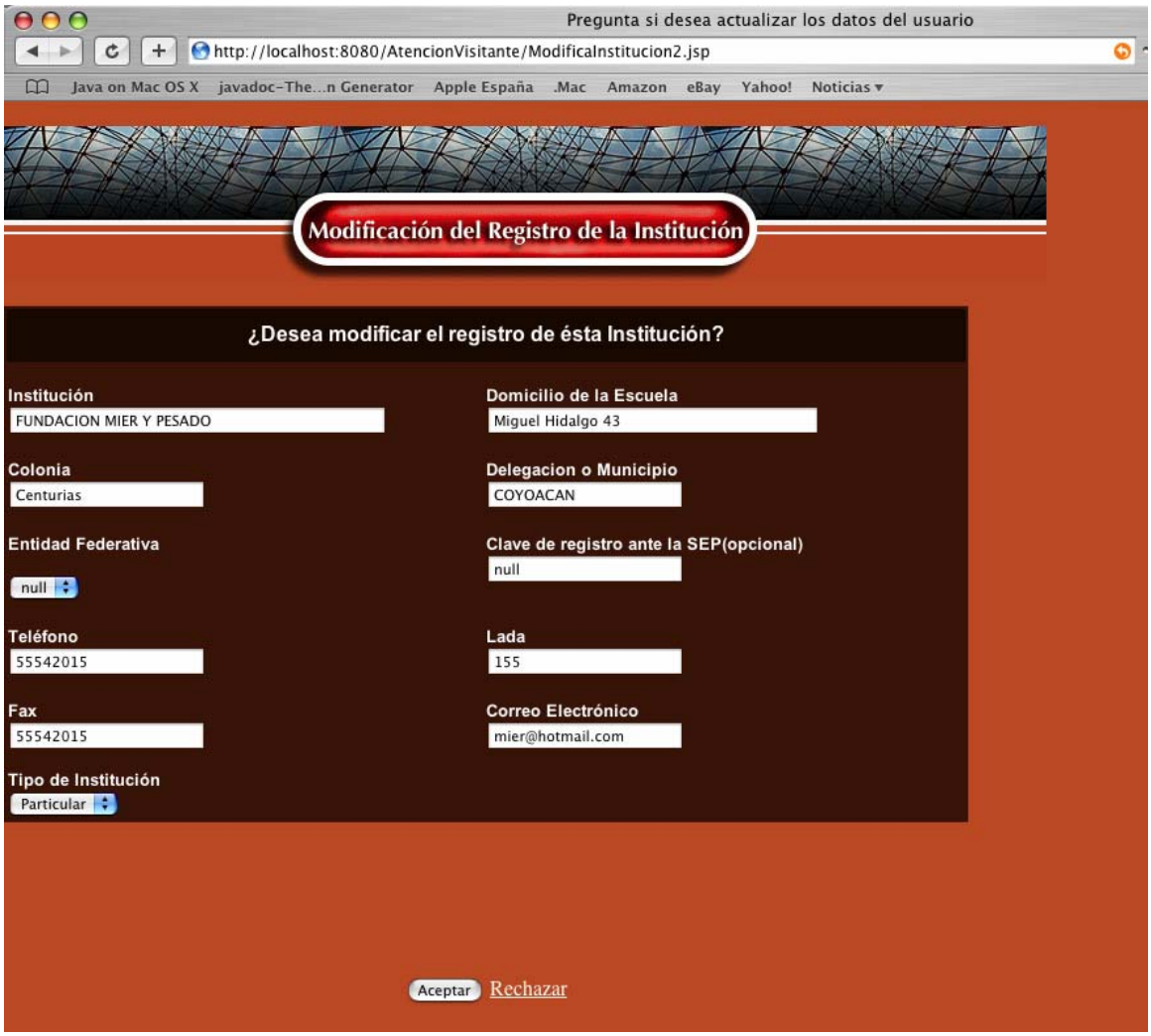


Fig.5. Pantalla que sirve para la verificación de registros de instituciones para el caso de realizar alguna modificación

APENDICE B

1. IEEE 802.11. Concebido en 1990 y aprobado siete años después, es la norma principal para la conexión de las distintas WLAN utilizando la banda de frecuencia 2.4 GHz. El sistema se divide en celdas, cada una de ellas controlada por una estación base o punto de acceso.

2. IEEE 802.11b. Trabaja en la frecuencia 2,4 GHz, ofrece velocidad de 11 Mbps. La interoperabilidad entre los distintos dispositivos ha quedado resuelta gracias a la marca Wi-Fi.

3. IEEE 802.11a. Es una mejora de los anteriores estándares en cuanto a seguridad. Operará en la banda de 5 GHz y brindará velocidades de datos que oscilarán entre 6 y 54 Mbps.

4. IEEE 802.11g. Pendiente de ratificación por parte del IEEE, sin embrago permitirá conseguir transmisiones inalámbricas de alta velocidad a 20 Mbps. En cuanto a la banda de frecuencia, utiliza los 2.5 GHz

BIBLIOGRAFIA

REFERENCIA	BIBLIOGRAFIA
[Booch 1998]	Booch G. 1998. Software Architecture and the UML. Presentación disponible en: http://www.rational.com/uml como arch.zip.
[Booch 1986]	Booch, G. 1988. Object Oriented Development. Trans. Of Soft. Eng. Vol. SE-12. Num. 2. Feb. 1986. pp. 211-221.
[Black, 1995]	Redes de Computadores, protocolos , redes e interfaces. BLACK Uyless. Edit. Computec ra-ma. Ed. Segunda México, 1995.
[Ceballos, 2000]	Java2,Curso de programación. CEBALLOS Sierra Fco.Javier. Edit. Alfaomega. Ed. Primera. Mexico D.F, 2000 pag. 23,24
[Cohen]	Daniel Cohen Mc Graw Hill Sistemas de información para la toma de decisiones
[Conallen 1999ª]	Conallen, J. "Modeling Web Applications with UML" Conallen, Inc. 9-Mar-1999 Disponible en: http://www.conallen.com/whitepapers/webapps/ModelingWebApplications.htm
[Conallen 1999B]	Conallen, J. "UML □stil□mac for Web Applications 0.91" Drafted Conallen, Inc. 22-Mar-1999 Disponible en: http://www.conallen.com/technologyCorner/webextension/WebExtension091.htm
[Cota 1994]	Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13.
[Douglas, 1996]	Comer E., Douglas: *Redes globales de información con Internet y TCP/IP*, tercera edición. Prentice Hall, 1996. [Protocolos TCP/IP]
[Douglas, 2003]	Java para estudiantes. DOUGLAS Bell ,Mike Parr. Edit. Pearson. Edi. Tercera. Mexico , 2003
[Grady , 2000]	El lenguaje Unificado de Modelado. Grady Booch, James Rumbaugh, Ivar Jacobson. Edit. Addison-Wesley ed. Primera España , 2000.
[Greiff 1994]	Greiff W. R. Paradigma vs Metodología; El Caso de la POO (Parte II). *Soluciones Avanzadas*. Ene-Feb 1994. pp. 31-39.
[Jacobson 1998]	Jacobson, I. 1998. "Applying UML in The Unified Process" . Presentación disponible en http://www.rational.com/uml como UMLconf.zip
[Jacobson 1992]	Jacobson, I. *Et. Al*. 1992. Object-Oriented Software Engineering; A Use Case Driven Aproach. ACM Press. Adison-Wesley Publishing. Co. U.S.A. 524 p. Ilus. Pp. 465-493.
[Lewis 1994]	Lewis G. 1994. "What is Software Engineering?" DataPro (4015). Feb 1994. pp. 1-10.
[Microsoft 1997]	Microsoft 1997. Microsoft Solutions Framework 1.0. Microsoft Corporation. USA.
[M&R 1998]	Microsoft y Rational 1998. *A White Paper on the Benefits of Integrating Microsoft Solutions Framework and The Rational Process.* Rational Software

	Corporation y Microsoft Corporation. Documento msfratprocs.doc Disponible en http://www.rational.com/uml/papers.
[OMG 1999]	Object Management Group. 1999. OMG Unified Modeling Language Specification (Draft). Versión 1.3. alfa R5, marzo de 1999. Disponible en: http://www.rational.com/uml
[Rosen, 1997]	Rosen, K. H. (1997). *UNIX Sistema V*. Mc Graw Hill, Madrid, España.
[Semmerville, 1985]	Ingenieria de Software. I.SEMMERVILLE. Edit. Addison-Wesley Iberoamericana ed. Segunda. E.U.A ,1985
[Tanenbaum, 1997]	Redes de Computadoras. TANENBAUM Andrew S. Edit. Prentice-Hall ed. Tercera México, 1997.
[Zavala 2000]	Zavala R. 2000. *Diseño de un Sistema de Información Geográfica sobre □stil□ma*. Tesis de Maestría en Ciencias de la Computación. Universidad Autónoma Metropolitana-Azcapotzalco. México, D.F. En prensa.
[Internet 1]	http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html
[Internet 2]	http://www.desarrolloweb.com/articulos/499.php?manual=15
[Internet 3]	http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#POO
[Internet 4]	http://www.monografias.com/trabajos12/basdat/basdat.shtml
[Internet 5]	http://www.aceproject.org/main/□stil□m/et/etg03.htm
[Internet 6]	http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml
[Internet 7]	http://www.saulo.net/pub/tcpip/
[Internet 8]	http://www.saulo.net/pub/redes/a.htm#3-2-4
[Internet [9]	http://www.investigacion.frc.utn.edu.ar/labsis/Publicaciones/AdminTcp/direcciones-ip.html
[Internet 10]	http://www.pegaso.ls.fi.upm.es/~lmengual/anexo/sld007.htm
[Internet 11]	http://www.saulo.net/pub/tcpip/a.htm#2-1
[Internet 12]	http://www.saulo.net/pub/tcpip/b.htm#3-3
[Internet 13]	http://frodo.escet.urjc.es/adamadrid/ofimatica_e_internet/Temas/Tema%201/clieserv.htm
[Internet 14]	http://www.itlp.edu.mx/publica/tutoriales/sistsdist1/u1parte6.htm
[Internet 15]	http://www.fismat.umich.mx/~elizalde/tesis/node21.html

[Internet 16]	http://www.fismat.umich.mx/~elizalde/tesis/node22.html
[Internet 17]	http://www.pue.udlap.mx/~tesis/lis/marquez_a_bm/capitulo5.pdf
[Internet 18]	http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#ProcUnificado
[Internet 19]	http://www.razonypalabra.org.mx/anteriores/n35/cscolari.html
[Internet 20]	http://mx.geocities.com/alfonsoaraujocardenas/topologias.html
[Internet 21]	http://www.microsoft.com/spanish/□sti/articulos/archivo/140303/voices/C_n_Java.asp
[Internet 22]	http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
[Internet 23]	http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_4.htm
[Internet 24]	http://www.monografias.com/trabajos/lengprog/lengprog.shtml
[Internet 25]	http://www.arsys.es/soporte/□stil□mación/comparativa.htm
[Internet 26]	http://www.desarrolloweb.com/articulos/393.php?manual=15
[Internet 27]	http://www.mmlabx.ua.es/mysql-postgres.html
[Internet 28]	http://www.postgresql.org/about/
[Internet 29]	Evolución y perspectivas para el desarrollo Web. Juan Carvajal. Coordinador Desarrollo. CISICRET – Sec. Estado Comercio http://www.astic.es/□stil/Boleweb/Articulos/Internet/desa.htm
[Internet 30]	http://www.pue.udlap.mx/~tesis/lis/martinez_v_lm/capitulo2.pdf
[Internet 31]	http://iio.ens.uabc.mx/~jmilanez/escolar/redes/rfc/rfc-index.html