UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA EN LÍNEA DE LA BOLSA DE TRABAJO DE LA FES CUAUTITLÁN.

TESIS

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

FABIOLA ROMERO CARRILLO

JOSÉ IVÁN HERNÁNDEZ ESPINOSA

ASESOR:

M. EN I. GERARDO VIGIL SANABRIA

CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO. 2013

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

ASUNTO: VOTO APROBATORIO

DRA. SUEMI RODRÍGUEZ ROMO
DIRECTORA DE LA FES CUAUTITLÁN
PRESENTE

ATN: L.A. ARACELI HERRERA HERNÁNDEZ
Jefa del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el Art. 28 del Reglamento de Exámenes Profesionales nos permitimos comunicar a usted que revisamos **LA TESIS**:

Desarrollo e implementación del sistema en línea de la bolsa de trabajo de la FES Cuautitlán

Que presenta la pasante: **Fabiola Romero Carrillo**
Con número de cuenta: **30226394-3** para obtener el Título de: **Licenciada en Informática**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 7 de Febrero de 2013.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	**FIRMA**
PRESIDENTE	MCC. Valentín Roldán Vázquez	
VOCAL	MI. Gerardo Vigil Sanabaria	
SECRETARIO	IME. Oscar Hernández Sánchez	
1er SUPLENTE	MGTI. Leonel G. López Salazar	
2do SUPLENTE	L.C.C. Liana López Pacheco	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 120).
HHA/pm

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

ASUNTO: VOTO APROBATORIO

DRA. SUEMI RODRÍGUEZ ROMO
DIRECTORA DE LA FES CUAUTITLÁN
PRESENTE

ATN: L.A. ARACELI HERRERA HERNÁNDEZ
Jefa del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el Art. 28 del Reglamento de Exámenes Profesionales nos permitimos comunicar a usted que revisamos **LA TESIS**:

Desarrollo e implementación del sistema en línea de la bolsa de trabajo de la FES Cuautitlán

Que presenta el pasante: **José Iván Hernández Espinosa**
Con número de cuenta: **30029754-2** para obtener el Título de: **Licenciado en Informática**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 7 de Febrero de 2013.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	MCC. Valentín Roldán Vázquez	
VOCAL	MI. Gerardo Vigil Sanabaria	
SECRETARIO	IME. Oscar Hernández Sánchez	
1er SUPLENTE	MGTI. Leonel G. López Salazar	
2do SUPLENTE	L.C.C. Liana López Pacheco	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 120).
HHA/pm

Agradecimientos

A su apoyo, constancia y cariño que no se desvaneció ni en los momentos más duros y que nos permitió seguir adelante sin importar los obstáculos que debíamos enfrentar para lograr culminar este trabajo.

Gracias por todas sus lecciones, esfuerzo, dedicación y sobre todo el no perder la esperanza ya que sin ello esto no sería posible.

Papá Gabriel Romero de Santiago.

Papá José Luis Hernández Martínez.

Mamá Juana Angélica Espinosa Martínez.

Mamá María del Rosario Carrillo Villegas.

A su paciencia infinita y apoyo incondicional que no dudo en ningún momento de nuestra capacidad para terminar con este trabajo.

Prof: MI. Gerardo Vigil Sanabria.

A ustedes que nos apoyaron con sus consejos y la oportunidad de faltar a nuestras labores en varias ocasiones pero sobre todo a su amistad y comprensión, que hoy nos permite dar fin a este proyecto.

Depto de Subdirección de Infraestructura Web DGTIC UNAM.

A ti que nos abriste las puertas dando la oportunidad de lograr nuestros sueños y que nos brindaste un lugar para formarnos como profesionales.

UNAM.

INTRODUCCIÓN

Hoy en día nos encontramos en la era de la información digital; cada vez más, los servicios en línea nos facilitan las tareas administrativas, por lo que el mundo ha cambiado con la tecnología de Internet.

Una manera en que vemos este cambio es el uso de Internet para llevar a cabo procesos como: pago de impuestos, registros escolares, consultas de estados bancarios, compras en línea, correo electrónico y muchos otros servicios que se ofrecen gracias a esta tecnología. Además se ofrece el servicio de bolsa de trabajo para el registro de candidatos y de empresas oferentes de vacantes, y así, optimizar la contratación de personal y ofrecer un servicio eficaz y eficiente a alumnos y egresados y cualquier entidad que lo requiere.

Una bolsa de trabajo es, de acuerdo al Diccionario de la Real Academia Española (2001), "Organismo encargado de recibir ofertas y peticiones de trabajo y de ponerlas en conocimiento de los interesados".

La bolsa de trabajo electrónica, da la posibilidad a las empresas de poder evaluar más perfiles de candidatos en menos tiempo y también permite a los candidatos poder aplicar a varias ofertas de empleo, sin la necesidad de presentarse físicamente en la empresa y sólo acudir a las entrevistas cuando se le solicite; por lo que se logra un eficiente proceso de reclutamiento de personal, al solo entrevistar a los candidatos que cubran el perfil deseado por las organizaciones.

Hoy en día la población mexicana cuenta con recursos tecnológicos para llevar a cabo esta campaña de empleo, la mayoría cuentan con servicio de Internet, de banda ancha, lo que permite a la población consultar datos y sitios Web más rápido.

Debido a que el sistema de bolsa de trabajo en la FES Cuautitlán implementado hoy en día es de tipo manual, presenta diferentes inconvenientes: la redundancia de datos e información no actualizada, además de que hay un retraso en tiempo ya que tanto las empresas como los candidatos tienen que trasladarse a la FESC para consultar vacantes y candidatos, provocando gastos innecesarios y acceso a información no actualizada.

Por lo anterior es necesario, implementar un sistema de información en línea para facilitar la difusión de las vacantes y perfiles de candidatos en tiempo real, con información actualizada y veraz, estando al alcance desde cualquier lugar donde se tenga acceso a una conexión de Internet.

El desarrollo del sistema se llevará a cabo con el paradigma de programación orientada a objetos o POO, ya que es más fácil relacionar el sistema al mundo real, agiliza la programación, permite la reutilización y extensión del código, facilita el mantenimiento, y es el paradigma más adecuado para trabajar un sistema en equipo.

El método a utilizar para el desarrollo del sistema será el Ciclo de vida clásico que está conformado por las etapas: análisis, diseño, desarrollo, pruebas e implementación del sistema.

Los lenguajes a utilizar para el desarrollo del sistema son:

HTML.- El cual permite describir la estructura y el contenido de las páginas Web en forma de texto para que por medio de los navegadores los usuarios puedan visualizarlas.

CSS.- Se usa para definir la presentación de un documento escrito en HTML o XML. Es muy útil ya que separa la estructura de un documento de su presentación, permitiendo así un mayor orden.

PHP.- Es un lenguaje que se pensó originalmente para su ejecución del lado del servidor, que es gratuito y se puede desplegar en la mayoría de los servidores, sistemas operativos y plataformas. Permite crear sitios Web con elementos dinámicos, realizar operaciones avanzadas para mostrar información, es capaz de conectarse con manejadores de bases de datos lo cual le permite almacenar información para su posterior uso.

JavaScript.- En la actualidad todos los navegadores interpretan este leguaje. Ayuda en gran medida a mejorar la interfaz del usuario, especialmente en páginas Web dinámicas.

MYSQL.- Sistema de Gestión de Base de Base de Datos que se caracteriza por su rapidez, consume pocos recursos y se integra muy bien con php. Además su licencia es GPL, lo que significa que es software libre. El lenguaje que se utiliza para acceder a las bases de datos es SQL.

OBJETIVO GENERAL

Desarrollar el sistema en línea de bolsa de trabajo para la Facultad de Estudios Superiores Cuautitlán, que permita dar un servicio en Internet de consulta de vacantes y candidatos.

OBJETIVOS PARTICULARES

Proveer a los alumnos y ex alumnos de la FES Cuautitlán un sitio web que les permita consultar ofertas de empleo desde cualquier computadora sin la necesidad de trasladarse a las instalaciones de la FES.

Facilitar el proceso de registro de empresas en la Bolsa de Trabajo de la FES Cuautitlán mediante un sitio web que les permita dar de alta sus vacantes desde una computadora sin la necesidad de trasladarse a las instalaciones de la FES.

HIPÓTESIS

El desarrollo del Sistema de Bolsa de Trabajo para la Facultad de Estudios Superiores Cuautitlán permitirá consultar vacantes publicadas por las empresas, así como los candidatos registrados en el sistema de Bolsa de Trabajo desde cualquier computadora con conexión a Internet facilitando el acceso a la información evitando que los interesados deban trasladarse a las instalaciones de la FES.

1. GENERALIDADES

1.1. Antecedentes de los lenguajes de programación

Desde la antigüedad el hombre busca facilitar la realización de diversas tareas diseñando herramientas que permiten desempeñar, una o varias tareas de una manera precisa, ágil y constante.

Las computadoras son herramientas que permiten desarrollar una enorme cantidad de tareas, por si solas son un conjunto de circuitos electrónicos que necesitan recibir instrucciones por parte de los humanos para ser capaces de resolver un problema, las cuales son interpretadas a través de un lenguaje de programación.

Los lenguajes de programación permiten transmitir los deseos del programador a la computadora, mediante palabras, reglas sintácticas y semánticas que definen una estructura y un significado de expresiones, de más fácil entendimiento para el hombre y que son traducidas al lenguaje binario que es hasta hoy en día el único lenguaje que pueden entender las computadoras.

Además de la sintaxis es necesario expresar el procedimiento que debe seguir la computadora para realizar tareas, la definición de la lógica de ese procedimiento se hace mediante un algoritmo que describe la secuencia ordenada de pasos sin ambigüedades que conducen a la solución de un problema (Joyanes Aguilar, 1999).

En la segunda mitad del siglo XX el desarrollo de los lenguajes de programación presentó un crecimiento sorprendente surgiendo una enorme cantidad de ellos los cuáles fueron creados para diferentes propósitos.

1.1.1. Lenguaje de Programación Orientado a Objetos (POO)

Los lenguajes de programación orientados a objetos surgieron en los años 70 y han tenido su mayor desarrollo desde mediados de los 90.

El paradigma de orientación a objetos es capaz de dividir programas largos en unidades semi-autónomas. El lema de la programación orientada a objetos es divide y vencerás (Holzner Steven, 2000).

De esta manera es posible dividir las partes de un programa, en varias unidades de manera sencilla y ordenada.

Características de la POO

- **Objeto.-** El mundo se encuentra conformado por objetos los cuáles tienen características y funciones Por ejemplo: El objeto persona, tiene las siguientes características: nombre, altura, peso y edad. Realizamos las siguientes funciones: comer, dormir, leer, escribir, hablar (Shmuller Joseph). Estas propiedades y acciones son comunes a todas las personas en general.

  El paradigma de orientación a objetos realiza una abstracción de los objetos que se encuentran en el mundo real al mundo virtual, simula al mundo (o un segmento de él) (Shmuller Joseph).

- **Clases.-** Las características y funciones que tienen los objetos son específicas para cada uno, por ejemplo: Una persona se puede llamar Pedro, Medir 1.70m pesar 70 kg, tener 25 años de edad y puede comer tacos, dormir a las 10:00 pm, leer libros de aventura y hablar con sus amigos. Estas actividades son únicas en el sentido que se pueden llamar igual o comer lo mismo, pero son realizadas por cada una las personas en específico. Lo que requiere una categorización de los objetos. En la POO los objetos se encuentran categorizados por clases, concentran las

características y las funciones de cada objeto. Además funcionan como una plantilla para fabricar objetos (Shmuller Joseph).

La instancia de una clase es la creación de un objeto a partir de su clase, el objeto persona una vez creado representa a esa persona que tiene sus atributos como nombre, edad y sus funciones como son comer, dormir, hablar y leer.

La clase nos permite generar infinidad de objetos y cada uno tendrá sus propios atributos y funciones.

Las características y las funciones de un objeto pueden ser modificadas por el objeto o por otro objeto, para mantener la integridad de cada objeto es posible mantenerlos ocultos a otras clases y objetos, mostrando solo lo necesario.

Entre más atributos y acciones tome en cuenta, mayor será la similitud de su modelo con la realidad (Shmuller Joseph).

- **Abstracción.-** Se refiere a quitar las propiedades y acciones de un objeto para dejar sólo aquellas que sean necesarias ya sean más o menos desde que la clase fue creada. Para algunos casos no es necesario contar con muchos atributos o métodos de un objeto todo el tiempo, esto depende de lo que al momento se requiera, si el objeto persona es un alumno que acaba de ingresar a la universidad serán necesarios datos como nombre, carrera, número de cuenta, grupo, etc. Si el objeto alumno ya no es de nuevo ingreso, necesita más métodos como pueden ser: calificaciones del semestre pasado, promedio, etc.

- **Herencia.-** Un objeto es una instancia de una clase lo que implica que tiene todas las características de la clase que proviene. A esto se le conoce como herencia (Shmuller Joseph). Así cada objeto hereda las características y funciones de su clase, de esta manera los objetos comparten propiedades y funciones entre sí pero son únicas para cada objeto que las usa.

  De igual manera una clase puede heredar de otra tomando sus características y funciones teniendo la posibilidad de agregar más.

- **Polimorfismo.-** En ocasiones una operación tiene el mismo nombre en diferentes clases (Shmuller Joseph). La función abrir la puerta puede ser la misma para una ventana, una persiana o un regalo. El polimorfismo es la capacidad de que varios objetos realicen la misma acción pero en diferentes situaciones.
- **Encapsulamiento.-** Es cuando un objeto oculta su funcionalidad (Shmuller Joseph). Así para usar las funciones de un objeto no es necesario como funciona si no el resultado que deseamos obtener. Al conducir un automóvil no necesitamos saber cómo funciona el mecanismo para bombear gasolina del motor. Nos basta con saber

que la gasolina hace funcionar el motor del vehículo.

- **Mensajes.-** Un objeto envía un mensaje a otro para realizar una operación (Shmuller Joseph). Los objetos se comunican entre sí para trabajar en conjunto.

Aspectos más importantes.

- **Uniformidad.** El diseño de un sistema con POO puede pasar a la etapa de desarrollo muy fácilmente. Ya que la representación de los objetos lleva implícita tanto el análisis como el diseño y la codificación de los mismos (Tejerina 2011).

- **Comprensión.** Las clases permiten una comprensión más clara y ordenada del modelo y de los programas que comprenden un sistema con POO. Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa trata (Tejerina 2011).

- **Flexibilidad.** La relación entre propiedades y funciones permite que cualquier cambio, se refleje al instante sin hacer algún procedimiento extra. Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos datos aparezcan (Tejerina 2011).

- **Estabilidad.** Los objetos instanciados tienen un lugar único en el transcurso de ejecución del sistema, permitiendo mantenerlos separados de otros objetos. Además con las propiedades públicas, privadas y protegidas se garantiza la integridad de sus propiedades y funciones. Permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo, sobre aquellos que cambian con frecuencia (Tejerina 2011).

- **Reusabilidad.** Las clases generadas para un sistema pueden ser muy parecidas en otro, esto permite la reutilización de las clases y ayuda a la fácil expansión del sistema. La noción de objeto permite que programas que traten las mismas estructuras de información reutilicen las definiciones de objetos empleadas en otros programas e incluso los procedimientos que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos donde estos están relacionados de una manera particular (Tejerina 2011).

La POO en las aplicaciones.

En el desarrollo de un sistema con POO las clases son las entidades de cada elemento que conforma el sistema, así el sistema de bolsa de trabajo tiene varias clases como el Candidato que tiene propiedades como: Nombre, dirección, teléfono, etc. Y funciones como: Consultar la bolsa de trabajo, crear su cuenta de usuario, etc.

También está la clase Empresa que tiene como características: RFC, nombre, teléfono, etc. Y funciones como: Crear vacantes, consultar candidatos, etc.

Las clases Candidato y Empresa se comunican a través de mensajes. Por ejemplo: si el Candidato consulta una vacante publicada por una empresa le envía un mensaje solicitando la información de esa vacante y la Empresa responde mostrando los requisitos para cubrirla.

El polimorfismo se ve reflejado en las clases Candidato y Empresa, ambas tienen la función de insertar, modificar, eliminar, etc. Pero actúan dependiendo de cada caso.

El encapsulamiento se usa en todo el sistema para mantener su integridad y así un objeto Candidato no pueda modificar una vacante o el nombre de una empresa. De esta manera el sistema se encuentra más ordenado al mantener separado el funcionamiento de cada objeto, esto le da mayor estabilidad.

El sistema se mantiene flexible por la comunicación existente entre todas las clases. Por ejemplo: si ocurre alguna modificación en la clase paginador se refleja de inmediato en su forma de presentar los datos de las clases Vacante, Candidato y Empresa lo que implica que los cambios realizados al sistema sean más fáciles de realizar.

Con el uso de la POO el sistema de bolsa de trabajo puede ser ampliado con muy poco esfuerzo, porque las clases pueden ser heredadas o extendidas sin afectar el sistema, esto implica la reutilización del código.

En algunos sistemas dependiendo su comportamiento se utilizan algunas o todas las propiedades de la POO.

1.2. Antecedentes de los sistemas Web

Internet se inició como un proyecto de defensa de los Estados Unidos. A finales de los años 60, la ARPA (Agencia de Proyectos de Investigación Avanzados) del Departamento de Defensa definió el Protocolo de Control de Transmisión/Protocolo de Internet TCP/IP (FRANCO 2007).

La red ARPA logró establecer la comunicación de varias computadoras en lugares muy alejados, mediante el protocolo de comunicación TCP/IP.

En 1975, ARPA NET comenzó a funcionar como red, sirviendo como base para unir centros de investigación militares y universidades (FRANCO 2007).

ARPANET dejó de ser proyecto para convertirse en una red estable y confiable para las investigaciones militares. Con el tiempo las universidades comenzaron a utilizar la red para compartir información entre sus diferentes campus.

En 1983 se interconectaron las tres redes ARPANET, CSNET y MILNET naciendo la red de redes: INTERNET. La esencia de la operación fueron los protocolos TCP/IP que fueron la clave que permitiría comunicarse con ordenadores de diferentes entornos con UNIX, MS-DOS o MacOS (ROBLES 2005).

Con el éxito de la comunicación entre las 3 principales redes, se logró establecer la manera de comunicar los 3 sistemas más importantes, permitiendo la interoperabilidad de los mismos.

En 1983 se adoptó el TCP/IP como estándar principal para todas las comunicaciones (FRANCO 2007).

Gracias a la facilidad y estabilidad que presentó el protocolo TCP/IP, fue estandarizado para que todas las comunicaciones se basaran en él, así se podría establecer un medio de comunicación entre diferentes sistemas computacionales.

En 1986 nació la red NSFnet (National Science Foundation) para facilitar el acceso de toda la comunidad científica americana a cinco grandes centros de supercomputarización. Esta red privada se convirtió en la espina dorsal de Internet. Ante el carácter abierto de esta red, surgieron muchas conexiones sobre todo por parte de las universidades (ROBLES, 2005).

Las universidades comenzaron a usar el correo electrónico y a realizar investigaciones en esta tecnología.

Tim Berners Lee es reconocido como el creador del hipertexto global que hoy conocemos como WWW (BUSTAMANTE, 2009).

El World Wide Web fue presentado en 1991 por Tim Berners Lee, como un sistema para conectar páginas Web, a través de Internet.

Gracias a esto se logró expandir el potencial de la Web, como medio de libre expresión y colaboración.

Poco a poco, todos los fabricantes de ordenadores personales y redes han incorporado el TCP/IP a sus sistemas operativos, de modo que en la actualidad cualquier equipo está listo para conectarse a Internet (FRANCO, 2007).

Los fabricantes de computadoras consientes de la importancia que estaba tomando la Internet, prepararon sus sistemas para que fueran capaces de conectarse a ella fácilmente, adoptando el protocolo TCP/IP.

En la primera mitad de la década de los 90, los usuarios de Internet eran científicos, las primeras páginas Web eran de una sola columna, basadas en texto y hechas en HTML (AGUILAR, 2011).

La primera generación de páginas Web contaban con mucho texto y algunas imágenes de manera muy lineal y poco interactiva, los navegadores Web estaban muy limitados, sus renderizadores solo eran capaces de mostrar algunas imágenes planas y texto.

Las principales limitantes de aquellas páginas eran las conexiones basadas en MODEM que impedían la descarga de grandes volúmenes de datos.

Los Websites parecían más una serie de documentos de texto unidos por enlaces (AGUILAR, 2011). La estructuración de estas páginas era muy desordenada.

La segunda generación de páginas Web llegó a mediados de los años 90. El uso de tablas hizo posible crear sitios Web con múltiples columnas. Esto amplió el diseño basado en texto (AGUILAR, 2011). El uso de las tablas permitió que las páginas Web lograran distribuir y ordenar mejor la información que se presentaba en la pantalla.

Las imágenes de fondo eran a menudo insertadas dentro de las tablas (Aguilar 2011). Se popularizó el uso de los marcos, aparecieron los contadores de visitas, Textos animados, textos con movimiento, imágenes con extensión .gif empezaban a danzar por la Web (Aguilar, 2011).

MACROMEDIA presenta FLASH en 1996. Aparece el software que hace una revolución en el aspecto y la interacción de las páginas Web. A finales de los 90. FLASH empezó a capturar la atención de los diseñadores. Muchos sitios Web empezaron a combinar el diseño basado en tablas con elementos Flash (AGUILAR, 2011).

Gracias al enorme dinamismo de flash, aparecieron las páginas de bienvenida o “intros” con efectos sorprendentes.

El lenguaje de programación PHP comenzó a ganar mucha popularidad a finales de los 90, debido a que dio la posibilidad de ejecutar scripts del lado del servidor, esto lo convierte en un lenguaje multiplataforma, ejecutando los scripts en distintos sistemas operativos. PHP empieza a ganar popularidad con el lanzamiento de PHP 3 en 1998 (AGUILAR, 2011).

El año 2000 se presentan formalmente las hojas de estilo. Las hojas de estilo en cascada (CSS) permitieron a los diseñadores gestionar de manera separada el diseño y la estructura del sitio (AGUILAR, 2011).

Las hojas de estilo dan la posibilidad de mantener ordenado el aspecto de las páginas Web, dan la posibilidad de realizar cambios a la presentación del sitio de una manera ordenada y ágil. Esta manera de desarrollo hizo más fácil controlar de manera uniforme el aspecto visual de las páginas y de los contenidos vertidos en ellas (AGUILAR, 2011).

Los navegadores Web ahora debían soportar el uso de hojas de estilo. Internet Explorer 5 se convirtió en el primer navegador Web en soportar CSS1 en su totalidad (AGUILAR, 2011).

En la primera mitad de la década del año 2000 se popularizó el uso del lenguaje Java Script, dicho lenguaje fue uno de los primeros intentos exitosos de dotar a la Web de inteligencia. Con JavaScript los diseñadores podían animar menús de navegación sin usar flash, realizar cálculos de cómputo y más (AGUILAR, 2011).

JavaScript es un lenguaje de programación que se ejecuta del lado del cliente, tiene como objetivo darle mayor dinamismo e interacción al usuario con los sitios Web. Los menús desplegables se convirtieron en una opción popular, los formularios comenzaron aparecer (AGUILAR, 2011).

Debido a las enormes ventajas que presenta JavaScript los navegadores Web fueron adoptándolo para que se ejecutara sin problemas. Cerca del 2002 ya todos los navegadores los soportaban (AGUILAR, 2011).

Entre los años 2005 y 2010 llega la Web 2.0 con contenidos interactivos muy similares a las aplicaciones de escritorio, llegan las redes sociales y el diseño Web se centra más en la publicación de contenidos gráficos e informativos, no solo ventas. Nuevas y poderosas aplicaciones inundaban la Web, capaces de procesar contenido en imágenes, vídeo y mucho más, las redes sociales se orientaron a involucrar al usuario en la creación de contenido y compartirlo con otros (AGUILAR, 2011).

En el año 2005, Apareció AJAX, una tecnología capaz de evitar recargar totalmente las páginas para actualizar el contenido de manera asíncrona. AJAX es la combinación de JavaScript y XML.

La web 2.0 empieza a ser considerada, comienza a usarse intensivamente JavaScript asíncrono y XML (AJAX) para el desarrollo de aplicaciones interactivas y la transición fluida de contenido (AGUILAR, 2011).

1.2.1 Los sistemas Web en México

México tuvo su primer contacto con la Internet hace más de 20 años en el año de 1986 con la conexión del Instituto Tecnológico de Estudios Superiores de Monterrey, ITESM hacia una Universidad de Texas en San Antonio (ROBLES, 2007).

Posteriormente la UNAM se conectó a Internet, fue el segundo nodo conectado en México a través de BITNET, justamente en el Instituto de Astronomía en la Ciudad de México. Esto mediante una conexión vía satélite de 56 Kbps, con el Centro Nacional de Investigación Atmosférica (NCAR) de Boulder, Colorado, en los Estados Unidos de Norteamérica. Por lo tanto, se trataba de una línea digital en Octubre de 1987 (ROBLES, 2007).

Poco a poco más Universidades en México se conectaron a la Internet, en el mundo había cada vez más países interesados en comunicarse con la red de redes. Estados Unidos experimentó un crecimiento acelerado de conexiones a la Internet. Debido este crecimiento, la National Science Fundation, en los Estados Unidos, requería de una respaldada red de telecomunicaciones para todos aquellos países que se integraban a Internet, por lo tanto, se tomaron algunas decisiones en México (ROBLES 2007).

Debido a la necesidad de una red unificada surge en México MEXnet. El 1 de Junio de 1992, MEXnet establece una salida digital de 56kbps al Backbone de Internet (ROBLES, 2007).

Con la creación de MEXnet el crecimiento de las conexiones a Internet en México aumentó rápidamente. Para 1993 ya existía una serie de redes bien establecidas.

Para finales de 1993 existían las redes: MEXnet, Red UNAM, Red ITESM, BAJAnet y Red Total CONACYT (ROBLES, 2007).

Entre los años 1994 y 1995, se abrió Internet con fines comerciales en México. Se consolidaron redes como Red Tecnológica Nacional RTN, creando un Backbone nacional y agrupando a un gran número de instituciones educativas y comerciales en toda la República (ROBLES, 2007).

Para 1996 ya hay host locales con extensión ".mx".

Desde 1996 Internet comenzó su crecimiento en México, cada vez más empresas y universidades se conectaron y crearon sus sistemas Web. En este año 2011 se alcanzaron los 34.9 millones de Internautas (Chagoya, 2011).

1.3 Bases de Datos

Desde el inicio de la informática los datos se han considerado como el elemento principal con el cual se trabaja, es por eso que se han pensado diferentes herramientas para la gestión de los datos e información.

Inicialmente los datos que se necesitaba almacenar y gestionar eran pocos, pero poco a poco han ido creciendo (Sánchez Asenjo, 2010). Es por eso que surgió la necesidad de tener Bases de Datos para una administración óptima de un sistema.

1.3.1 ¿Qué es una Base de Datos?

Para Damián Pérez (2007), una Base de Datos es un "almacén" que nos permite guardar grandes cantidades de información de forma organizada para que luego la podamos encontrar y utilizar fácilmente.

En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. En un principio recibieron el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases o Base de Datos (BD) (Camps, Casillas, Costal, Gibert, Martín, Pérez, (2005)).

Características de las bases de datos

Las características principales de una base de datos es que los datos tienen Independencia lógica y física, la redundancia es mínima, muchos usuarios puede acceder a la información, los datos son íntegros, se pueden realizar consultas complejas y optimizadas, se facilita la recuperación y respaldo, se puede acceder a la información por medio de lenguajes de programación estándar.

Modelo de Bases de Datos

Se puede clasificar a las bases de datos de acuerdo a su modelo de administración de datos, que es una "descripción" de un contenedor de datos y los métodos para almacenar y recuperar información de esos contenedores, no es algo físico, sino abstracto (algoritmos, conceptos matemáticos).

Los modelos de bases de datos más utilizados son:

Bases de datos jerárquicas.

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

Sin embargo, los árboles, como instrumentos para la representación de estructuras de datos, presentan problemas por su poca flexibilidad, lo que da origen a una falta de adaptación a muchas organizaciones reales (Ruiz, 2001).

Modelo de red.

Los aspectos estáticos como la estructura de los datos (tipos de entidades, tipos de interrelaciones, etc.), se representa en forma de grafo.

En cuanto a los aspectos dinámicos, los modelos en red se caracterizan por ser navegacionales, es decir, la recuperación y la actualización de la base de datos se lleva a cabo registro a registro, y procedimentales, es decir, asumen el conocimiento de la estructura interna de la base de datos por parte del programador (Rivera, 2007).

Bases de datos transaccionales.

El fin de este tipo de base de datos es el envío y recepción de datos a gran velocidad, pero también deben ser de gran calidad, por lo que se debe conocer su estado y ser consistente, asegurando que todas las operaciones sean realizadas o que todas se han cancelado, es decir, no se pueden dejar acciones a la mitad o inconclusas. Por lo general se conectan a bases de datos relacionales para su mayor aprovechamiento.

Bases de datos relacionales.

Fundamentalmente es el uso de "relaciones", es el modelo más utilizado en la actualidad debido a que es el que se asemeja más a la realidad. En este modelo el lugar y la forma en

que se almacenan los datos no tiene relevancia, lo que lo hace más fácil de entender, además de que brinda una gran flexibilidad y poder para administrar la información.

Bases de datos multidimensionales.

Son muy semejantes a las bases de datos relacionales, pero los campos se representan en más dimensiones, es decir, almacena sus datos con más de un valor. Este modelo aporta mucha rapidez de respuesta, pero da muchos problemas para dar mantenimiento.

Bases de datos orientadas a objetos.

Este modelo es muy reciente y se pretende almacenar en la base de datos objetos completos, es decir, almacenar su estado y comportamiento. También es capaz de incorporar todos los conceptos importantes del paradigma de objetos como son encapsulación, herencia, polimorfismo.

Bases de datos documentales.

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.

Bases de datos deductivas.

Debido a que se base en lógica matemática también son llamadas base de datos lógicas. Es capaz de definir y hechos que se almacenan en la base de datos, permitiendo hacer deducciones a través de inferencias.

1.3.2. Sistemas de Gestión de Base de Datos

Los Sistemas de Gestión de Base de Datos, son aplicaciones que permiten a los usuarios, definir, crear, darle mantenimiento y proporcionar un acceso controlado a las bases de datos (Gil, Albrigo, Do Rosario, 2005). Es un software que permite un entorno adecuado para la manipulación de las bases de datos.

Para Gil, Albrigo y Do Rosario (2005), los principales objetivos de un sistema de Gestión de Base de Datos son:

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos.

- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el

Lenguaje de Manejo de Datos. Hay dos tipos de Lenguajes de Manejo de Datos, y se diferencian por la forma en que acceden a los datos.

Lenguajes procedurales: Manipulan la base de datos registro a registro y se deben especificar las operaciones a realizar para obtener los datos.

Lenguajes no procedurales: Manipulan la base de datos en conjuntos de registros y se especifican qué datos deben obtenerse como resultado sin plantear las forma de hacerlo. El lenguaje no procedural más utilizado es SQL (Structure Query Languaje) que se ha convertido en un estándar y el lenguaje por defecto de los SGBD relacionales.

1.3.3 Lenguaje SQL

El lenguaje más común para construir las consultas a las bases de datos relacionales es SQL, es un estándar para sistemas de base de datos relacionales. Los responsables de publicar este lenguaje como estándar son ANSI (Instituto Americano de Normalización) y la ISO (Organismo Internacional de Normalización) (García, 2010).

El SQL agrupa tres tipos de sentencias de acuerdo a sus objetivos.

Lenguaje de Definición de Datos (DDL, Data Definiton Language)

Es el grupo de Sentencias que soportan la definición de los objetos de la base de datos así como la misma base de datos, las tablas, las vistas, los índices, los procedimientos almacenados, reglas, dominios y valores por defecto.

Lenguaje de Manipulación de Datos (DML, Data Management Language)

Sentencias SQL para manipular los datos que están almacenados en la base de datos, a nivel de filas y/o columnas, ya sea para modificar, eliminar, consultar o agregar nuevas filas a las tablas.

Lenguaje de Control de Datos (DCL, Data Control Language)

Grupo de Sentencias para controlar las funciones de administración que realiza el DBMS, como atomicidad y seguridad.

1.3.4 Sistema de Gestión de Base de Datos MySQL

MySQL es un sistema de gestión de bases de datos relacional que utiliza el lenguaje SQL, es de Open Source, es un servidor muy rápido, fiable y fácil de usar.

MySQL Server se desarrolló originalmente para trabajar con grandes bases de datos mucho más rápido, y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años (*Oracle and/or its affiliates*, 2011).

Su conectividad, velocidad y seguridad hacen que MySQL sea tan apropiado para acceder a bases de datos en Internet.

Historia de MySQL

Originalmente se empezó a utilizar mSQL para realizar la conexión a las tablas utilizando rutinas rápidas de bajo nivel (ISAM). Sin embargo se llegó a la conclusión que no era lo suficientemente rápido o flexible. Esto provocó la creación de una nueva API pero casi con la misma interfaz API que mSQL. Esta API fue diseñada para permitir código de terceras partes que fue escrito para poder usarse con mSQL para ser fácilmente portado para el uso con MySQL.

La derivación del nombre MySQL no está clara. Ya que el directorio base y un gran número de bibliotecas y herramientas han tenido el prefijo "my" por más de 10 años. Sin embargo, la hija del co-fundador Monty Widenius también se llama My. Por lo que no se sabe cuál de los dos dio su nombre a MySQL.

El logo de MySQL es un delfín que se llama "Sakila", que fué elegido por los fundadores de MySQL AB de una gran lista de nombres sugerida por los usuarios en el concurso "Name the Dolphin" (ponle nombre al delfín). El nombre ganador fué enviado por Ambrose Twebaze, un desarrollador de software Open Source de Swaziland, África.

Características

Según Oracle and/or its affiliates (2011), algunas de las características más importantes de MySQL son las siguientes:

- Portabilidad.

- Está escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible.
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial).
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS.
- Registros de longitud fija y longitud variable.
- Sentencias y funciones.
- Los nombres de funciones no colisionan con los nombres de tabla o columna.
- Puede mezclar tablas de distintas bases de datos en la misma.
- Seguridad, ya que es un sistema de privilegios y contraseñas. El tráfico de contraseñas está cifrado cuando se conecta a un servidor.
- Soporte a grandes bases de datos.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2).
- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
- Soporte completo para distintos conjuntos de caracteres.
- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.
- Todos los programas MySQL pueden invocarse con las opciones --help o -? para obtener asistencia en línea.

Portabilidad

Es posible portar MySQL a todos los sistemas modernos que tengan un compilador de C++ y una implementación funcional de sistemas subprocesos (threads) POSIX.

MySQL se desarrolló y se utilizó sobre todo en Linux, FreeBSD, y Sun Solaris (Versiones 8 y 9). Sin embargo también ha sido compilado en diferentes combinaciones de sistemas operativos y paquetes de subprocesos como Mac OS X, Windows.

Sin embargo hay algunos factores que pueden determinar si una plataforma está preparada para un servidor MySQL para "misiones críticas", ya que no todas las plataformas son igualmente aptas, depende de la capacidad de núcleo o kernel de Sistema Operativo, biblioteca de subprocesos, rendimiento y estabilidad general del sistema de archivos, entre otros.

Tipos de Datos que soporta MySQL

Al crear nuestras tablas en es necesario indicar el tipo y longitud de los datos que se podrán almacenar, y es de suma importancia hacerlo de una forma correcta ya que de esto depende el que no se quede corta la capacidad de almacenamiento, así como su rápida ejecución.

Datos Numéricos.

Este tipo de datos solo acepta como su nombre lo dice, datos numéricos ya sean positivos, negativos, enteros, decimales, en notación hexadecimal, científica o decimal. Los tipos numéricos pueden ser los siguientes:

INTEGER: Acepta los atributos SIGNED, indicando que puede tener valor negativo y UNSIGNED indicando que sólo pueden tener valor positivo. Además aceptan el atributo ZEROFILL en donde los números se completarán hasta la máxima anchura disponible con ceros. Una columna con el atributo zerofill es al mismo tiempo unsigned aunque no se especifique.

BIT o BOOL: En este tipo de datos se almacena los enteros 0 ó 1 (falso o verdadero).*TINYINT:* Este tipo de dato es un número entero con rango de valores válidos desde -128 a 127. Si se configura como unsigned (sin signo), el rango de valores es de 0 a 255. *SMALLINT:* Son para números enteros, con rango desde -32768 a 32767. Si se configura como unsigned, 0 a 65535.

MEDIUMINT: Tipo de dato para números enteros; el rango de valores va desde -8.388608 a 8388607. Si se configura como unsigned, 0 a 16777215.

INT: Para almacenar números enteros, en un rango de -2147463846 a 2147483647. Si configuramos este dato como unsigned, el rango es 0 a 4294967295.

BIGINT: Este tipo de dato es para un número entero con rango de valores desde -9223372036854775808 a 9223372036854775807. Unsigned, desde 0 a 18446744073709551615.

FLOAT (m,d): Representa números decimales. Podemos especificar cuantos dígitos (m) pueden utilizarse, y cuantos en la parte decimal (d). Mysql redondeará el decimal para ajustarse a la capacidad, por lo que es de precisión simple.

DOUBLE: Este tipo de dato es también de coma flotante pero el rango numérico que abarca tiene doble precisión.

DECIMAL: Almacena los números como si fueran cadenas.

Caracteres o cadenas.

CHAR: Almacena cadenas de longitud fija y su longitud es de 1 a 255 caracteres. Es más rápido el acceso a los datos que el tipo VARCHAR.

VARCHAR: Este tipo de dato es muy parecido a CHAR y su longitud es variable pero también va de 1 a 255, lo que hace la diferencia es que VARCHAR solo ocupa el tamaño del dato almacenado, a diferencia del CHAR que siempre ocupara el tamaño establecido aunque el dato sea menor, lo que hace que la base de datos sea menor. Se cambia automáticamente a CHAR si usamos VARCHAR con valor de 4 o menos.

TINYTEXT, TINYBLOB: Este tipo de datos es de capacidad de 255 caracteres como máximo. La diferencia entre el tipo text y blob es que el primero es para cadenas de texto plano (sin distinguir formatos) y el blob es para objetos binarios que es para cualquier tipo de información, desde un archivo de texto (con formato) hasta imágenes, archivos de sonido y video.

TEXT y BLOB: Este tipo de dato es muy parecido al anterior excepto que es con un rango de 255 - 65535 caracteres.

MEDIUMTEXT, MEDIUMBLOB: Este tipo de dato permite hasta 16777215 caracteres.

LONGTEXT, LONGBLOB: Permite hasta máximo de 4.294.967.295 caracteres.

Varios.

DATE: Este tipo de dato es para almacenar la fecha. El formato por defecto es YYYY MM DD desde 0000 00 00 a 9999 12 31.*DATETIME:* Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas:minutos:segundos.

TIMESTAMP: Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo.

TIME: Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'.

YEAR: almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.

SET: Un campo que puede contener cero, uno ó varios valores de una lista, la cual puede tener un máximo de 64 valores.

ENUM: Es igual que el SET pero solo puede tener un único valor de una lista que se especifica y admite hasta 65535 valores distintos.

Sentencias básicas de MySQL

MySQL soporta una gran cantidad de sentencias e instrucciones para el manejo de las bases de datos, a continuación se muestran las más comunes y básicas.

CREATE DATABASE [nombre]: Crea una base de datos con el nombre dado:

mysql> create database miprueba;

CREATE TABLE [nombre]: Este comando además de crear una tabla dentro de la base de datos permite especificar su estructura, donde se especifica los tipos de variable para cada ítem.

mysql> CREATE TABLE mascota (nombre VARCHAR(20), especie VARCHAR(20), sexo CHAR(1), fecha_nacimiento DATE);

SHOW: Este comando muestra las tablas o las bases de datos que hay en MySQL.

mysql> show databases;

mysql> show tables;

USE db_nombre: indica a MySQL que use la base de datos *db_nombre.*

mysql> use miprueba;

DESCRIBE: Permite saber que campos tiene un tabla y de que tipo.

mysql> describe prueba;

ALTER TABLE: Este comando permite cambiar la estructura de una tabla existente. Por ejemplo, puede añadir o borrar columnas, cambiar el tipo de columnas existentes, o renombrar columnas o la misma tabla, etc.

mysql> *ALTER TABLE personas RENAME usuarios*

SELECT: Es un comando utilizado para traer información desde una tabla

SELECT seleccionar_Esto

FROM desde_tabla

WHERE condiciones;

*mysql> selec * from tablaPrueba where id = 1;*

La instrucción anterior "le dice" a MySQL que muestre todos los campos de la tabla *tablaPrueba* del registro que tenga el id = 1

INSERT INTO: Esta sentencia permite ingresar un nuevo registro en una tabla dentro de la base de datos.

mysql> insert into usuarios (Nombre, Clave, Fecha) VALUES ('Juan', 'PaSw0rD', '2011-05-25');

DELETE FROM : Este comando es para eliminar algún registro seleccionado.

mysql> delete from usuarios Where nombre='Juan';

UPDATE: Comando que modifica el campo indicado en el registro en cuestión y no requiere que se vuelva a llenar la tabla.

mysql> UPDATE mascota SET fecha_nacimiento = '2007-08-31' WHERE nombre = 'Pelusa';

COUNT (*): Esta función junto con la cláusula SELECT es para dar a conocer el total de registros de acuerdo a ciertas condiciones.

mysql>SELECT COUNT(*) AS numero_dep FROM departamentos WHERE ciudad_dep = 'Guadalajara';

LIKE: Es un elemento muy importante que hace más valiosa las consultas en MySQL, ya que hace reconocimiento de patrones. Se utiliza con diferentes caracteres pero el más común es %. Este carácter se usa para encontrar datos o valores de cualquier cantidad de caracteres:

mysql> LIKE 'J%' ← todos los que empiezan con J.

mysql> LIKE '%ANA' ← todos los que terminen con ANA.

mysql> LIKE '%JUAN%' ←- todos los que lleven en medio del dato JUAN.

DROP DATABASE: Elimina la base de datos, incluyendo sus tablas.

QUIT: Permite salir de la línea de comandos de MySQL.

mysql> quit

Estos son algunos comandos en cuanto al manejo de Usuarios:

CREATE USER: crea nuevas cuentas MySQL

DROP USER: elimina una o más cuentas MySQL

CURRENT_USER: Devuelve el nombre de usuario y el del host para el que está autentificada la conexión actual.

MySQL y la programación orientada a objetos.

El diseño de un sistema basado en programación orientado a objetos con UML da la posibilidad de crear clases entity que representan las tablas de una base de datos de la siguiente manera:

- El nombre de la tabla es el mismo nombre que la clase.
- Los nombres de los campos de la tabla y sus tipos, son los mismos que las propiedades de la clase.
- En los campos de la tabla se almacenan los valores de las propiedades de la clase.

Con las clases entity se puede tener un mejor diseño de sistemas unificando clases y diagramas entidad relación.

2. LENGUAJES DE PROGRAMACIÓN UTILIZADOS

2.1 HTML

2.1.1 ¿Qué es HTML?

HTML significa Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto) y es el lenguaje utilizado con el fin de crear páginas de Internet, que son documentos para la World Wide Web. Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada navegador.

La World Wide Web (WWW) es un sistema para la distribución de información que se basa en hipermedios (texto o multimedia) enlazados y accesibles a través de Internet. Cabe aclarar que la WorldWideWeb no es Internet, es un servicio que ofrece Internet, algunos otros servicios son el e-mail, algunos sistemas de juegos en línea, sistemas de chat, el FTP (Protocolo de Transferencia de Archivos) y Telnet.

Cuando se creó este lenguaje HTML se pensó en que fuera portable al 100%, es decir, que se pueda visualizar independientemente del sistema operativo que tuviera la computadora. Esta característica se logra gracias a todo lo que hay en la página es texto, caracteres ASCII, los cuales son interpretados por todos los tipos de sistemas operativos (Soria, 2002).

Funcionamiento de la Web

Lo primero que hace la Web es traducir la parte del nombre del servidor de la URL en una dirección IP usando la base de datos distribuida de Internet DNS. La dirección IP se necesita para contactar al servidor Web y poder enviarle paquetes de datos.

La Web está basada en una serie de páginas que utilizan el protocolo de transferencia de hipertexto (HTTP), que es la manera en que se transfieren hacia las computadoras las páginas Web. Por lo que se envía una petición HTTP al servidor Web solicitando el recurso.

La mayoría de las páginas Web ofrecen enlaces a otras páginas Web (hiperenlaces), descargas, definiciones, entre otros.

2.1.2 Antecedentes del lenguaje HTML

El origen de HTML comienza en 1980 cuando el físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos (librosweb.es, 2011).

En el CERN se han creado gran parte de los planes de lo que es la tecnología de la Red, empezando en los 80. Cisco, que sostiene casi toda la infraestructura de la Internet, fue la empresa que implementó el primer ruteador IP (protocolo de Internet) en Europa.

Los sistemas de "hipertexto" habían sido desarrollados años antes, pero al finalizar su desarrollo, Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de “hipertexto” para Internet. Junto al ingeniero de sistemas Robert Cailliau presentaron la propuesta ganadora llamada *WorldWideWeb (W3).*

En 1991 se presentó el primer documento formal con la descripción de HTML con el nombre “HTML Tags”.

En 1993 fué cuando se realizó la primer propuesta oficial para convertir HTML en un estándar, y aunque hubo un gran avance fué hasta 1995 que el organismo IETF consigue publicar el estándar HTML 2.0, que es el primer estándar oficial de HTML.

A partir de 1996 la W3C (World Wide Web Consortium) es la encargada de publicar los estándares de HTML.

El 24 de Abril de 1998 se publicó HTML 4.0, entre las novedades se encuentran las hojas de estilo CSS, la posibilidad de incluir scripts, mejoras en los formularios, tablas complejas etc.

En el 2000 nace el XHTML 1.0 que está diseñado para adaptar el HTML 4.01 al lenguaje XML, mantiene casi todas sus etiquetas y características pero añade las restricciones y elementos propios de XML.

En el 2004 Apple, Mozilla y Opera proponen evolucionar el estándar HTML 4.0. Aunque son rechazados, forman el WHATWG (Web Hypertext Application Technology Working Group) y en el 2007 El W3C adopta el trabajo de WHATEWG publicándolo como lo que sería el borrador HTML5 y en el 2010 es cuando ya se tiene el borrador de W3C para HTML5 (Martín, 2011).

HTML5 es la evolución de HTML4 ya que se podrá disponer de nuevas API's que ayudarán a crear aplicaciones Web mucho más dinámicas y ricas. Sin embargo debido a que se encuentra en etapa de borrador no se encuentra lo suficientemente maduro para implementarlo en grandes proyectos o aplicaciones.

2.1.3 Navegadores Web

Un navegador Web es una aplicación que interpreta el código, HTML generalmente, para que podamos leer la información de sitios Web, permitiendo al usuario interactuar y navegar en su contenido a través de la red por medio de enlaces.

En caso de una página Web típica, se solicita el texto HTML y se analiza inmediatamente por el navegador, el cual, hace peticiones adicionales para los gráficos y otros recursos que formen parte de la página. El navegador renderiza la página tal como se describe en el código HTML, el CSS y otros lenguajes Web.

Los navegadores son capaces de llevar a cabo el seguimiento de enlaces de una página a otra, a esto se le llama navegación y es de ahí de donde se origina el nombre "navegador".

Cache de Agente de Usuario

Los navegadores almacenan en cache del disco duro del cliente todos los recursos web de las páginas a las que va accediendo, ya que tiene "a la mano" los recursos utilizados para que cuando vuelva a entrar a la misma página no tarde tanto en volver a mostrar todos los recursos. Este tipo de cache se denomina "cachés de agente de usuario" (User-Agent), que son cachés privados.

El navegador enviará una petición HTTP sólo si la página ha sido actualizada desde la última carga, en otro caso, la versión almacenada se reutilizará en el paso de renderizado para agilizar la visualización de la página.

Los recursos almacenados caducan de forma independiente, es decir no es el mismo tiempo para imágenes, que para hojas de estilo, archivos JavaScript, etc.

Esto ayuda mucho para reducir la gran cantidad de tráfico Web en Internet, la carga del servidor, los tiempos de carga de una página entre otros. Sin embargo para páginas sensibles a seguridad, los diseñadores de páginas Web pueden controlar las cabeceras HTTP enviadas a los usuarios para que no sean guardadas en caché, un ejemplo de esto son los bancos on line.

Para Gorostiza (2009) existen tres tipos de caches que pueden actuar durante el proceso de solicitud de un documento Web:

Cache de Agente de Usuario (User-Agent): Está presenten en los navegadores web y lógicamente sólo funcionan para un único usuario.

Cache Compartida o proxy-cachés directos: Este tipo de cache son utilizados por los proveedores de servicios de Internet y empresas para ahorrar ancho de banda. La comparten todos los usuarios que accedan.

Cache pasarela o proxy-cachés inversos: Funcionan como respaldo de un servidor web de forma transparente para los usuarios.

Renderizado

Para que el navegador pueda "pintar" el contenido de las páginas Web es necesario que tenga un motor de renderizado (layout engine o rendering engine en inglés), que es un software que permite mostrar en la pantalla del usuario, el contenido con estilo, estructura y forma de acuerdo a la información de formato, es decir que interpreta marcas HTML, XML, archivos de imágenes, información del formato como CSS, entre otros.

Debido a que cada navegador tiene un motor de renderizado diferente, una misma página se visualiza de forma diferente, lo cual es un problema para los programadores y diseñadores ya que tienen que desarrollar la página de forma que se vea de la mejor manera en todos los navegadores y a veces realizar diferentes versiones, una para cada navegador.

Para López (2010) algunos de los motores de renderizado más importantes son:

Gecko: Es un motor multiplataforma y libre, originalmente desarrollado por Netscape, actualmente su desarrollado es gestionado por la Fundación Mozilla. Era muy criticado por su complejidad y su gran uso de memoria, pero se logró resolver ese problema en Firefox 3.

KHTML/WebCore: Es de código libre (con la licencia LGPL) que originalmente fué desarrollado para el navegador de KDE:Konqueror, pero en el 2003 fué adoptado por Apple para su navegador Safari.

WebKit: Es de código abierto desarrollado por Apple sobre la base del código KHTML. Es sumamente liviano, su código es compacto, simple, claro y riguroso respeto a los estándares de HTML, y usa muy poca memoria para su funcionamiento.

*Trident o MSHTML:*Es de Microsoft y es considerado por la mayoría de desarrolladores como el cáncer de Internet. Es el motor menos compatible por los estándares.

Presto: Es el motor de código cerrado desarrollado por Opera Software para el navegador de Opera. Probablemente con este motor de renderizado Opera 10 sea el navegador con mejor soporte para los estándares Web.

2.1.4 Estructura de un documento HTML

Todos los navegadores usan reglas básicas para poder mostrar una página Web con un buen formato:

El espacio en blanco es ignorado: Ya que un documento HTML puede estar en cualquier tipo de fuente y además la ventana del navegador puede ser de cualquier tamaño.

Las etiquetas pueden ser escritas en mayúsculas o en minúsculas: En todo caso se aconseja su escritura en mayúsculas para poder distinguirlas del texto normal.

Existe normalmente una etiqueta de inicio y otra de fin: La etiqueta de fin contendrá el mismo texto que la de inicio añadiéndole al principio una barra inclinada /. La etiqueta afectará por tanto a todo lo que esté incluido entre las etiquetas de inicio y fin. No obstante, existen algunas que no necesitan cierre, ya que en estas etiquetas se presupone su final, como por ejemplo; <P> párrafo,
 salto de línea ó <IMG> inclusión de una imagen (Grupo Eidos, 2000).

Sin embargo no todas las etiquetas no solo indican órdenes tan sencillas, sino que en ocasiones, cuando se requiere de algo más personalizado se requieren parámetros.

<ETIQUETA ATRIBUTO="valor">

Dichas etiquetas pueden ponerse en el documento de acuerdo a las necesidades de la página Web. Sin embargo un documento HTML debe tener como base una estructura.

En primer lugar debemos poner el tipo de documento, aquí deberemos especificar a que estándar del HTML responde nuestra página entre una de las siguientes opciones:

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

Cumple el estándar HTML 2.0

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

Cumple el estándar HTML 3.2

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"

Cumple el estándar HTML 4.0

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"

Cumple el estándar HTML 4.0 y no contiene además elementos desaconsejables

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"

Es una definición de marcos que cumple el estándar HTML 4.0

El HTML 4.0 considera desaconsejables aquellos elementos que, aún siendo soportados, han sido sustituidos por otros más potentes y, por ello, es posible que sean eliminados del estándar HTML en el futuro para HTML.

Después sigue la etiqueta <HTML> que es para indicar que el documento que estamos creando es un documento HTML, y a partir de este momento escribiremos dentro de estas etiquetas y será parte de la página Web.

<HTML>

Código de la página

</HTML>

El contenido de estas etiquetas se divide en dos, el encabezado y el cuerpo de la página.

La cabecera: Este es el lugar más indicado para colocar aquellos elementos que no alteran el contenido de la página pero si la forma de presentarlo y su comportamiento. Aquí es recomendable colocar los scripts y hojas de estilo. También un elemento muy importante es el título de la página, este aparecerá en la cabecera de la ventana del navegador.

<TITLE>Título</TITLE>

También se pueden incluir los META, que sirven para indicar propiedades de la página como puede ser el nombre del autor, la herramienta con la que se creó la página, las palabras que permiten clasificar la página dentro de un buscador, descripción del contenido de la página.

<META NAME="GENERATOR" CONTENT="Mozilla/4.03 [es] (Win95; I) [Netscape]">

Un elemento muy parecido al anterior y que lo complementa, es META HTTP-EQUIV que también son usadas para darle información al navegador. Algunos de sus atributos son: la fecha en que la página debe ser recargada por los navegadores, si el contenido debe ser no-cache, el idioma en el que está escrita, entre otros.

<META HTTP-EQUIV="expires" CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">

El cuerpo: Aquí esta todo lo que la página muestra, tablas, imágenes, texto, etc. Algunos de los atributos que acepta la etiqueta <BODY> son: El color de fondo de la página, imagen de fondo, color de texto.

La estructura de un documento HTML se puede resumir así:

tipo de documento

<HTML>

<HEAD>

<TITLE>titulo de la página</TITLE>

cosas que afectan a la página pero no a su contenido

</HEAD>

<BODY parámetros>

contenido de la página

</BODY>

</HTML>

2.2. CSS

2.2.1 ¿Qué es?

CSS (Cascading Style Sheets u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación (Barcia, 2003). El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

Al crear una página Web se utiliza un lenguaje para crear el contenido, darle funcionalidad a cada elemento, párrafos, títulos, botones, textos, etc. Una vez creado el contenido se debe utilizar el lenguaje CSS para definir el aspecto de cada elemento (o grupo de elementos), como puede ser la posición de cada elemento, el color, tamaño, si tiene imagen de fondo, márgenes, etc. Es por eso que gracias a las CSS somos capaces de dar mejores resultados finales a la página final, teniendo más control sobre ella.

Hay 3 formas diferentes de aplicar las reglas de estilo a una página Web:

Hoja de Estilo Externa: Es una hoja que se encuentra almacenada en otro archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera más adecuada y potente ya que separa completamente la estructura del documento y la presentación del mismo.

“CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web complejas” (Eguíluz, 2009).

Hoja de Estilo Interna: Es una hoja de estilo que esta incrustada dentro de un documento HTML, dentro del elemento <HEAD>. Aquí se separa la información del estilo y el código HTML.

Estilo en línea: Es la forma de incrustar el lenguaje de estilo dentro de las etiquetas de HTML, aunque es otra opción no es recomendable, ya que a la larga el código del documento se hace tedioso, complejo y difícil de dar mantenimiento, esta forma de insertar el código sólo se podría usar en el caso de querer modificar una página Web “al vuelo”, con prisa.

Un caso especial son los clientes de correo electrónico ya que estos no soportan las hojas de estilos externas, ya que todavía no existen estándares que los fabricantes de clientes de correo respeten, por lo que hay que usar CSS dentro de las etiquetas (hojas de estilo en línea).

2.2.2 Antecedentes de las hojas de estilo (CSS)

HTML originalmente se creó para fines científicos u otros usos diferentes a los que le damos en la actualidad, es por eso que está limitado para mostrar una presentación más compleja y personalizada. Por esta razón los diseñadores han implementado tablas, imágenes transparentes, entre otros “trucos” para ayudar a visualizar lo que se tiene planeado. Sin embargo esto ha causado problemas a la hora de mostrarse en diferentes navegadores; la frustración de los diseñadores por no poder llevar a la pantalla lo que habían creado en papel, donde el control sobre la forma del documento es absoluto; el leer código HTML mezclado con etiquetas y presentación se hace muy pesado y difícil a la hora de querer hacer cambios, dar mantenimiento o depurar páginas, es por todo esto que surgió la necesidad de separar el formato de la estructura.

Después que el lenguaje de etiquetas SGML (Standard Generalized Markup Language) apareciera, alrededor del año 1970 aparecieron las hojas de estilo. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos (Eguíluz, 2009).

El W3C propuso la creación de un lenguaje de hojas de estilos específicamente para el lenguaje HTML, se presentaron nueve propuestas.

Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal). La propuesta CHSS fué realizada

por Hakon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets) (Eguíluz, 2009).

En 1995 El W3C añade la estandarización de CSS al grupo de trabajo de HTML y en 1996 publica la primer recomendación oficial, conocido como “CSS nivel 1”.

En 1997 el W3C separa los trabajos del grupo de HTML en tres:

El grupo de trabajo HTML

El grupo de trabajo de DOM

El grupo de trabajo de CSS

En 1998 el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como “CSS nivel 2”.

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. El mismo año que se publicó CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000.

Desde 1998 se ha estado desarrollando CSS3 y en la actualidad prácticamente ya esta desarrollado y lo que promete es un CSS más fácil de usar e implementar y al mismo tiempo más potente. CSS3 incluye todo lo del nivel 2 e incorpora nuevos selectores, hipertexto enriquecido, bordes y fondos, texto vertical, interacción del usuario, reproducción en dispositivos multimedia y mucho más (Orós, 2008).

2.2.3. Navegadores que lo soportan

Los desarrolladores y/o diseñadores de páginas Web tienen la difícil tarea de mostrar su página de la misma forma en distintos navegadores, y tomando en cuenta que cada navegador tiene su propio motor de renderizado, es un tanto complicado, por lo que es de suma importancia conocer el comportamiento de CSS en cada uno de ellos.

Los navegadores Safari y Opera son los más avanzados en el soporte de CSS, ya que incluyen muchos elementos de la versión CSS 3.

Para mayor referencia sobre los navegadores ver Tabla1 y Tabla2.

Navegador	Motor	CSS 1	CSS 2.1
Internet Explorer	Trident	Completo desde la versión 6.0	Completo desde la versión 8.0
Firefox	Gecko	Completo	Casi completo
Safari	WebKit	Completo	Casi completo
Opera	Presto	Completo	Casi completo
Google Chrome	WebKit	Completo	Casi completo

Tabla 1. Compatibilidad de CSS1 y CSS2.1 con diferentes navegadores.

Nota: Fuente: Eguíluz Pérez, Javier (8 de mayo de 2009). Introducción a CSS. 223 Disponible en: http://www.librosweb.es/css/

Tabla 2. Compatibilidad de CSS3 con diferentes navegadores.

Nota: Fuente: findmebyIp (2010). HTML5 & CSS3 Support. FindmebyIp.com. Disponible en: http://www.findmebyip.com/litmus/#target-selector

2.2.4. Ventajas y desventajas de utilizar CSS

Ventajas:

- Al separar la presentación y el contenido nos obliga a crear documentos bien definidos.
- Podemos modificar la presentación de todos los elementos sin tener que modificar el código HTML.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.);
- Se dispone de una mayor variedad de comandos más potentes y precisos que nos

ayudarán a maquetar el documento exactamente como lo queremos.

- Es un lenguaje muy fácil de utilizar ya que se basa en propiedades muy intuitivas.
- Al separar la estructura de la presentación podemos utilizar un solo archivo css y así modificar varios documentos HTML.
- Es uno de los pilares de DHTML y puede combinarse con varios lenguajes como JavaScript, VBScript, etc.
- Su uso adecuado permite ahorrar muchas líneas de código.

Desventajas:

- Su compatibilidad con los navegadores no es muy buena.

2.3. PHP

2.3.1. ¿Qué es?

Es un lenguaje de secuencia de comandos que corre del lado del servidor, diseñado especialmente para la Web. Dentro de una página Web se puede incrustar código php que se ejecutará cada que se visite dicha página.

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor (Bakken, 2002).

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI (tecnología que permite a un cliente solicitar datos de un programa ejecutado en un servidor Web), como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

Sin embargo para poder crear páginas Web no basta con PHP, este es un lenguaje muy poderoso pero tiene que trabajar con otras tecnologías básicas para poder realizar páginas interactivas como son:

Un servidor Web, un lenguaje de programación en éste, y una base de datos.

2.3.2. Antecedentes del lenguaje PHP

PHP fué creado en 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Originalmente las personas le pedían a

Rasmus que les permitiera utilizar sus programas en sus propias páginas. Con esto se vió el éxito que tuvo PHP, así que Rasmus Lerdorf hizo algunas mejoras.

Las siglas PHP equivalían inicialmente a Personal Home Page (Página de inicio personal) pero se modificaron de acuerdo con la convención de designación de GNU (del inglés, Gnu's Not Unix, Gnu no es Unix) y ahora equivale a PHP Hipertext Preprocesor (Prepocesador de hipertexto PHP) (Welling, 2005).

Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI (Alvarez, 2001).

En 1997 se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras dieron la base para la versión 3 de PHP.

En el 2000 se lanzó la versión 4 donde se utiliza el motor Zend, además hubo mejoras en la rapidez ya que primero se compila y luego se ejecuta, mientras que antes se ejecutaba mientras se interpretaba el código, además tuvo una mayor independencia del servidor Web y un API más elaborado y con más funciones.

El 13 de julio de 2004, lanzaron PHP 5, este ya utilizando el motor Zend Engine 2.0 (o Zend Engine 2). Algunas de sus ventajas sobre las versiones anteriores son:

- Mejor soporte para la programación orientada a objetos.
- Mejor soporte para MySQL.
- Mejor soporte para XML.
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.

2.3.3. Ejecución en servidor

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

PHP es un lenguaje de programación diseñado para generar páginas Web de forma interactiva en el ordenador que las proporciona, denominado servidor Web. El código PHP se ejecuta entre la página solicitada y el servidor Web, añadiéndose al código HTML y modificando su resultado (Davis, 2008).

Apache es un servidor Web que convierte solicitudes del navegador en páginas Web, y conoce como procesar código PHP. PHP es sólo un lenguaje de programación, y sin la potencia de un servidor Web como Apache tras él, sería imposible para los usuarios visualizar sus páginas Web elaboradas con este código (Davis, 2008). Apache no es la única opción pero tiene ventajas de ser libre, publicar su código fuente, y estar bajo una licencia sin restricciones.

Apache, PHP y MySQL soportan un gran número de sistemas operativos, por lo que no importa en donde se empleará, tanto en servidor como cliente.

2.3.4. Ventajas del lenguaje PHP

Para Welling (2005) las principiantes ventajas de PHP sobre su competencia son:

- *Alto rendimiento:* Mediante el uso de un único servidor, puede servir millones de acceso al día.

- *Interfaces para diferentes sistemas de base de datos:* Dispone de una conexión propia a todos los sistemas de base de datos. Permite establecer una conexión a cualquier base de datos que suministre un controlador ODBC. Entre ellas, se incluyen los productos de Microsoft, y muchos más.

- *Bibliotecas incorporadas para muchas tareas Web habituales:* Incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web en pocas líneas de código.

- *Bajo coste:* Es gratuito, por lo que se puede descargar sin ningún costo.

- *Facilidad de aprendizaje y uso:* Es muy fácil de aprender debido a que está basado en otros lenguajes de programación, principalmente en C y Perl.

- *Portabilidad:* Su código está disponible para una gran cantidad de sistemas operativos diferentes, además funcionará sin la necesidad de aplicar ninguna

modificación a los diferentes sistemas que ejecute PHP.

- *Disponibilidad de código abierto:* Se puede modificar algún elemento al programa con toda libertad.

- *Disponibilidad de asistencia técnica:* Zend Technologies es la empresa responsable del motor de PHP y basa su desarrollo en la oferta de asistencia técnica y software de forma regular.

La verdadera belleza de PHP radica en la sencillez unida a la potencia.

2.3.5. Programación orientada a objetos en PHP

Desde la versión 3 y 4 de PHP tenía compatibilidad con el enfoque orientado a objetos, pero es hasta la versión 5 que esta compatibilidad es mucho más completa.

La versión 5 de PHP cuenta con completas funciones orientadas a objetos, como por ejemplo herencia, atributos y métodos privados y protegidos, clases y métodos abstractos, interfaces, constructores y destructores. (Welling, 2005).

El software orientado a objetos se diseña y construye como un conjunto de objetos independientes con atributos y operaciones que interactúan para cubrir nuestras necesidades.

La ventaja principal del software orientado a objetos es su capacidad para fomentar la encapsulación u ocultación de datos. Además la programación orientada a objetos ayuda a administrar un sitio complejo, ya que se pueden agregar o modificar funciones sin alterar otras, se reutiliza el código y se reduce el costo de mantenimiento.

Un lenguaje orientado a objetos debe admitir el polimorfismo, que significa que las clases pueden tener diferentes comportamientos con respecto a la misma operación. En PHP, solo las funciones miembro de una clase pueden ser polimórficas.

En PHP también podemos contar con la herencia, que es la que nos permite crear una relación jerárquica entre clases utilizando subclases.

Existen constructores para las clases, que son las que se llaman al crear un objeto y suele realizar tareas útiles de inicialización. Un constructor se declara de la misma forma que

otras operaciones, pero tiene el nombre especial _construct() (en versiones anteriores a la 5, el constructor tenía el mismo nombre que la clase).

PHP5 permite destructores, estos permiten ejecutar una funcionalidad concreta antes de que se destruya una clase y es necesario asignar _destruct() al nombre del destructor.

También se incorpora nuevos modificadores de acceso que controlan la visibilidad de atributos y métodos que son public, private y protected.

Una de las novedades de PHP5 es la palabra clave *final*. Al utilizarla para una función se evita que se remplace en alguna clase y si se aplica a una clase se evita que se creen subclases.

No es válida la herencia múltiple (varias clases padre). Pero existen lo que se llama interfaz y es una solución a la herencia múltiple, ya que una interfaz especifica una serie de funciones que deben implementarse en las clases que implementen dicha interfaz.

Aparece el concepto de constante de clase. Se puede acceder a la constante si se utiliza el operador :: para especificar la clase a la que pertenece la constante.

En PHP esta la palabra clave *static* que se aplica a los métodos para que puedan invocarse sin crear una instancia de la clase.

A partir de PHP5 aparece por primera vez la palabra clave *instanceof* que nos permite comprobar el tipo de un objeto. Por ejemplo ($b instanceof B).

La palabra clave *clone* es para poder copiar un objeto ya existente, con los mismos valores de atributo.

PHP permite las clases y métodos abstractos y estos se utilizan en jerarquías de clases complejas en las que todas las subclases deben incluir y reemplazar un método concreto, lo que también se puede hacer con una interfaz.

También podemos utilizar un bucle *foreach()* para iterar por los atributos de un objeto como si se tratara de una matriz.

Lo anterior es solo algunas de las acciones que se pueden realizar en PHP, sin embargo todavía hay más.

2.3.6. Conexión con MYSQL

PHP y MySQL son una poderosa combinación a la hora de crear páginas Web, ya que en conjunto facilitan mucho la creación de sitios Web completos y con almacenamiento de datos.

PHP es un potente lenguaje de programación pero éste no puede guardar información, es por eso que se necesita una base de datos, que solo con un manejador de base de datos como lo es MySQL se puede obtener.

MySQL automatiza las tareas más comunes relacionadas con el almacenamiento y la recuperación de información específica del usuario, en base a los criterios proporcionados (Davis, 2008).

Para Davis (2008) trabajar conjuntamente con PHP y MySQL tiene las siguientes ventajas:

PHP y MySQL funcionan muy bien de forma conjunta: Su uso conjunto es muy sencillo, ya que sus interfaces se han emparejado entre sí.

PHP y MySQL son código abierto: Al ser proyectos de código abierto, tanto PHP como MySQL pueden ser empleados de forma libre. Los usuarios avanzados y que deseen hacer alguna mejora o cambio en la funcionalidad, en los programas y en el lenguaje, pueden hacerlo. Tampoco se debe adquirir licencias de uso para cada ordenador en el que se utilice.

PHP y MySQL tienen comunidades de apoyo: Ambas herramientas tienen comunidades muy activas en las que sus participantes pueden brindar apoyo; también se puede encontrar ayuda profesional para MySQL.

PHP y MySQL son rápidos: Su diseño es muy sencillo y eficiente que permite el proceso más rápido.

PHP y MySQL no importunan con detalles innecesarios: No se necesita conocer a fondo como PHP y MySQL interactúan, ya que tiene un interfaz estándar para invocar procesos MySQL desde PHP.

Los pasos básicos para llevar a cabo una consulta (sin importar si está en línea de comandos de MySQL o desde php):

- Conexión a la base de datos.
- Selección de la base de datos que vamos a usar.
- Creación de una instrucción SELECT.
- Ejecución de la consulta.

- Resultados

Cuanto se lleve a cabo la conexión desde php se recomienda crear un archivo que llevará la información necesaria para conectarse a la base de datos, este archivo servirá para que cada que se requiera conectarse sea llamado y en caso de alguna modificación, sólo se tendrá que modificar ese archivo.

La función de conexión de MySQL es:

mysql_connect ($db_servidor, $db_nombre_usuario, $db_contraseña);

2.4 Javascript

2.4.1 ¿Qué es?

JavaScript es un lenguaje de alto nivel basado en objetos, diseñado para permitir a los programadores la generación de documentos Web interactivos de un modo sencillo (Bobadilla, Alcocer, Alonso 2001).

JavaScript es un lenguaje de programación creado con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas (Orós, 2008).

Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Eguíluz, 2009).

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página Web, o en programas más grandes orientados a objetos mucho más complejos. Con JavaScript podemos crear diferentes efectos e interactuar con nuestros usuarios (Pérez, 2009).

De acuerdo a las definiciones JavaScript es un lenguaje de programación que tiene como propósito crear páginas Web dinámicas, es decir que genera o modifica su contenido a partir de los datos que introduce un usuario en un formulario o las acciones que el usuario realiza sobre los elementos de la página. Esto tiene como consecuencia una mayor interacción del sitio Web con el usuario.

El lenguaje JavaScript se integra con HTML de esta manera es posible ejecutarlo en un navegador Web, se encuentra basado en objetos lo que permite separar los elementos de la página Web. Es de uso libre no requiere licencia ni instalación ya que la mayoría de los navegadores actuales lo incluyen.

2.4.2. Antecedentes del lenguaje Javascript

En la primera mitad de los años 90 los usuarios se conectaban a Internet con módems a una velocidad máxima de 28.8 Kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones Web y por tanto, las páginas Web comenzaban a incluir formularios complejos (Eguíluz, 2011).

Con la inclusión de los formularios en las páginas Web fue necesario validar sus campos, lo que representaba un proceso muy lento por la baja velocidad de las conexiones y los usuarios debían de esperar demasiado para saber si los datos ingresados fueron correctos o no.

Un programador pensó que podría resolver el problema de las respuestas lentas del sitio Web, desarrollando un lenguaje de programación que se ejecutara del lado del cliente que visitaba los sitios. El lenguaje JavaScript fue creado por el programador Brendan Eich, de la empresa Netscape para su navegador. Inicialmente se llamaba LiveScript (López, 2007).

Eich desarrolló el lenguaje LiveScript, posteriormente Netscape firmó una alianza con la empresa Sun Microsystems para continuar el desarrollo del nuevo lenguaje, justo antes de su presentación cambió el nombre por JavaScript.

La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época (Eguíluz, 2011).

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1 (Eguíluz, 2011).

El lenguaje JavaScript comenzó a utilizarse en el navegador Netscape. Microsoft inmediatamente lanzó el lenguaje Jscript.

Internet Explorer en su versión 3 incluyó Jscript, era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales (Eguíluz, 2011).

Con 2 versiones del mismo lenguaje presentadas por 2 empresas distintas era muy probable que comenzara una guerra de tecnologías, por ello Netscape decidió que debían estandarizar el lenguaje JavaScript.

En 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association) (Eguíluz, 2011).

Se creó un comité encargado de estandarizar el lenguaje JavaScript, el cuál debía ser multiplataforma e independiente de cualquier empresa (Eguíluz, 2011). Se creó el primer estándar ECMA-262.

Posteriormente, la organización internacional para la estandarización (ISO) adoptó el estándar ECMA-262 a través de su comisión IEC, dando lugar al estándar ISO/IEC-16262 (Eguíluz, 2011).

La tercera edición del estándar ECMA-262 (publicada en Diciembre de 1999) es la versión que utilizan los navegadores actuales y se puede consultar gratuitamente en:

http://www.ecma-international.org/publications/standards/Ecma-262.htm (Eguíluz 2011).

Actualmente se encuentra en desarrollo la cuarta versión de ECMA-262, que podría incluir novedades como paquetes, namespaces, definición explícita de clases, etc.

ECMA también ha definido varios estándares relacionados con ECMAScript, como el estándar ECMA-357, que define una extensión conocida como E4X y que permite la integración de JavaScript y XML (Eguíluz, 2011).

2.4.3 Ejecución en cliente

La arquitectura Cliente/Servidor es aquella en la que confluyen una serie de aplicaciones basadas en dos categorías, que cumplen funciones diferentes (una requiere servicios y la otra los brinda). Pero que a la vez pueden realizar actividades en forma conjunta o independiente (Lanzillotta, 2005).

El cliente son todas aquellas computadoras que se conectan a la red para solicitar servicios como pueden ser: enviar y recibir correos electrónicos, consultar los precios de algún producto, recibir su historial académico, etc. Dichos servicios son proporcionados por un servidor.

El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor (Lanzillotta, 2005).

Las computadoras personales de casa u oficina, tablets y celulares que se conectan a Internet para solicitar algún servicio son clientes.

El servidor es la máquina desde la que se suministran servicios y que está a la espera del requerimiento del cliente (Lanzillotta, 2005).

Así se observa que el servidor puede ser cualquier computadora que de algún servicio, no necesariamente debe ser un equipo con grandes capacidades de hardware, dependiendo del tipo de servicio que dará, son los requerimientos de almacenamiento y procesamiento del equipo que brinda el servicio.

Los servidores pueden enviar varios servicios a la vez (Lanzillotta, 2005).

La arquitectura Cliente/Servidor permite la interoperabilidad de los sistemas, ya que el servidor puede estar implementado en un sistema operativo Linux y el cliente en un sistema Windows o Mac OS.

Entre las características fundamentales de esta arquitectura encontramos que tanto el cliente como el servidor, pueden realizar tareas en forma conjunta o separada, ya que el cliente también tiene sus propias aplicaciones, archivos, bases de datos y además, pueden estar en la misma plataforma o en plataformas diferentes (Lanzillotta, 2005).

El lenguaje JavaScript se ejecuta del lado del cliente, de esta manera es posible realizar múltiples procesos con la ayuda de la computadora que se conecta a la red, solicitando algún servicio, eliminando mucha carga de procesamiento para el servidor, mejorando el tiempo de respuesta para el usuario final.

La ejecución del lado del cliente implica que debe ejecutarse en un navegador Web. La mayoría de los navegadores lo soportan.

El lenguaje JavaScript se puede ejecutar desde una página Web de dos maneras:

Incluyéndolo dentro de un documento HTML. El código JavaScript se encierra entre etiquetas <script> y se incluye en cualquier parte del documento HTML (Eguíluz, 2011).

Definiendo un archivo con extensión .js. Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos HTML enlazan mediante la etiqueta <script>. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento HTML puede enlazar tantos archivos JavaScript como necesite (Eguíluz, 2011).

2.4.4 Seguridad que ofrece JavaScript

El lenguaje JavaScript se ejecuta del lado del cliente, lo que implica que la computadora del usuario tiene una conexión abierta. JavaScript puede ejecutar muchos procesos sin que el usuario se entere de lo que sucede, de esta manera JavaScript debe cumplir con normas de seguridad.

JavaScript fue diseñado para cumplir ciertas normas básicas de seguridad, dirigidas fundamentalmente a proteger la integridad del sistema del usuario (Nieto, 2008).

De estas normas, las más importantes son:

Acceso bloqueado a recursos externos: La primera norma impuesta por los diseñadores de JavaScript es la imposibilidad de acceder a elementos externos a la página Web (Nieto, 2008). Esto impide a JavaScript leer y escribir archivos de la computadora del usuario o ejecutar programas externos a la página Web y realizar conexiones externas a servidores ajenos a la página, por ejemplo los virus downloaders rompen con esta regla ya que al ejecutarse se conectan con programas externos a la página donde se encuentran alojados.

La única excepción son las cookies (Nieto, 2008). Las cookies se guardan como archivos especiales en la computadora del usuario, pero se verifica que pertenezcan al mismo dominio que las genera, se encuentran regulados por un estándar de la World Wide Web Consortium (W3C). Para evitar su uso malicioso.

Acceso limitado a recursos potencialmente peligrosos: Si el mundo exterior queda fuera del alcance de un script JavaScript, algunos elementos de la propia página HTML también han sido protegidos por los diseñadores de JavaScript (Nieto, 2008).

No se permite leer algún elemento del objeto history del navegador del usuario, para evitar que algún script acceda a datos privados del usuarios, durante la navegación por la red.

Tampoco es posible establecer el valor de un campo de formulario de tipo file (Nieto, 2008). Los campos tipo file se usan para transferir archivos desde el cliente al servidor son conocidos como (uploads), resultaría demasiado peligroso que se pudiera establecer el nombre y tipo de algún archivo de la máquina del usuario desde JavaScript.

Confirmación del usuario para realizar ciertas acciones: La definición de JavaScript especifica la necesidad de que el navegador pida la confirmación del usuario antes de realizar ciertas acciones (Nieto, 2008).

Las peticiones más comunes son:

- Cerrar cualquier ventana que no haya sido abierta desde un script JavaScript (Nieto, 2008). Si una ventana no se abrió desde el comando window open de JavaScript.

- Imposibilidad de abrir ventanas con JavaScript de tamaño inferior a 100 pixeles (Nieto, 2008). Con el fin de mantener visible su contenido.

- Abrir ventanas fuera de las coordenadas de la pantalla (Nieto, 2008). De esta

manera se evita la creación de ventanas fuera del rango de visión del usuario.

- Acceso bloqueado a elementos descargados de otros servidores.- Ningún script puede acceder a las propiedades de documentos que procedan de un servidor distinto (Nieto, 2011). Un script cargado desde un determinado origen no puede ser modificado desde un script ajeno a ese origen.

Scripts relacionados: Se refiere a la protección del contenido de las páginas Web:

Proteger las imágenes de un documento (Nieto, 2008). Evita que las imágenes se puedan descargar pulsando el botón derecho del ratón, tan solo permite visualizarlas en una ventana nueva.

Protección de una imagen mediante CSS (Nieto, 2008). Evita que las imágenes se descarguen con la opción guardar como, colocando una imagen en blanco sobre la imagen que se intenta descargar.

Validación de campos de formulario: Una de las grandes aportaciones de JavaScript a la creación de interfaces Web es la posibilidad de acceder al contenido de los campos de los formularios para realizar acciones sobre los valores introducidos por el usuario (Nieto, 2008).

Las validaciones más básicas son:

- Comprobar que se han suministrado todos los campos obligatorios (Nieto, 2008).- Una de las acciones más básicas en la validación es verificar que todos los campos obligatorios de un formulario sean ingresados.

- Comprobar que el formato de un campo es el esperado (Nieto, 2008).- Este filtrado es muy usado para verificar que el dato "teléfono" sea solo numérico, que el dato "fecha" este en un formato válido, que los campos no sean capaces de recibir caracteres raros como símbolos de pregunta, de número, de porcentaje, etc.

- Comprobar la validez (sintáctica) de las direcciones de correo y URLs (Nieto, 2008).- Los correos electrónicos tienen una arroba y otras características muy especiales, por medio de JavaScript es posible definir una regla de sintaxis para verificar que el dato ingresado cumpla con ella.

- Comprobar que no se sobrepasa la longitud, número de líneas o tamaño de la entrada (Nieto, 2008).- Es muy común limitar el tamaño de caracteres, para el dato

ingresado, salvo algunas excepciones donde se requiere un tamaño indefinido, esto dependerá del tipo de dato que se desea obtener.

La validación de los datos se lleva a cabo cuando el usuario pulsa el botón enviar del formulario en ese momento se activa la validación desde JavaScript.

Es importante mencionar que la validación desde JavaScript no sustituye la validación que debe realizarse desde el servidor. Por motivos de seguridad, en la aplicación del servidor que recibe la información (Nieto, 2008). La validación desde JavaScript es solo para que la experiencia de interacción del usuario sea más cómoda.

2.4.5 JQUERY

¿Qué es el DOM?

DOM es una abreviatura de Document Objet Model por sus siglas en inglés. En español podríamos traducirlo por Modelo de Objeto de Documento (Álvarez, 2008).

El Modelo de Objetos del Documento (DOM) permite ver un documento HTML de otra manera, describiendo el contenido del documento como un conjunto de objetos en los cuales un programa JavaScript puede actuar sobre ellos (González, 2011).

Acorde al W3C el Modelo de Objetos del Documento es una interfaz de programación de aplicaciones (API) para documentos válidos HTML y bien construidos XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula (González, 2011).

DOM es un conjunto de utilidades específicamente diseñadas para manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente (Eguíluz, 2011).

Los navegadores Web transforman la página Web original en un conjunto de elementos llamados nodos, por medio del DOM, los nodos se encuentran interconectados entre sí. A la unión de todos los nodos se le llama árbol de nodos (Eguíluz, 2011).

Los nodos más utilizados son:

- Document, nodo raíz del que derivan todos los demás nodos del árbol (Eguíluz, 2011).
- Element, representa cada una de las etiquetas XHTML. Se trata del único nodo que

puede contener atributos y el único del que pueden derivar otros nodos (Eguíluz, 2011).

- Attr, se define un nodo de este tipo para representar cada uno de los atributos de las etiquetas XHTML, es decir, uno por cada par atributo=valor (Eguíluz, 2011).
- Text, nodo que contiene el texto encerrado por una etiqueta XHTML (Eguíluz, 2011).
- Comment, representa los comentarios incluidos en la página XHTML (Eguíluz, 2011).

Acceso directo a los nodos.

Una vez construido automáticamente el árbol completo de nodos DOM, ya es posible utilizar las funciones DOM para acceder de forma directa a cualquier nodo del árbol (Eguíluz, 2011).

La manipulación se realiza obteniendo la etiqueta de cada elemento mediante su identificador.

Los programas se dirigen mediante eventos; responden a la entrada asíncrona del usuario en forma de clicks del ratón y de pulsaciones de teclas de teclado. Esto se realiza mediante una Interfaz gráfica de usuario (GUI) incrustada, por lo que el código JavaScript en la parte del cliente utiliza el modelo de programación dirigida por eventos (Flanagan, David).

Los elementos tienen características que pueden ser modificadas mediante eventos como: pasar el ratón sobre nuestro elemento, presionar algún botón del ratón sobre el elemento. En el caso de los formularios presionar la tecla enter sobre el botón que envía los datos.

Cuando se produce un evento, el explorador Web intenta llamar a una función de controlador de evento apropiada para responder en la parte del cliente. Tenemos que definir los controladores de eventos apropiados y registrarlos con el sistema para que el explorador pueda llamarlos cuando los necesite (Flanagan, David).

Existen eventos que pueden detectar si una página Web se cargó o se descargó del navegador.

Antecedentes del framework JQUERY

JQuery es un Framework de JavaScript. Un Framework es un conjunto integrado de componentes que colaboran para proporcionar una arquitectura reutilizable para una familia de aplicaciones (Arias, 2002).

También se define como un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales (Álvarez, 2009).

De acuerdo a las definiciones anteriores podemos decir que un Framework es un conjunto de librerías y códigos, que facilitan el desarrollo de muchas tareas comunes a distintos proyectos de software.

El programador John Resig comenzó el desarrollo de un Framework, allá por el año 2005 (Chaffer, 2009).

Resig reunió una serie de funciones para que por medio de programación fuera sencillo encontrar elementos en una página Web y asignarles comportamientos (Chaffer 2009).

El nombre JQuery hace énfasis en el objetivo de la tecnología, que es encontrar y consultar partes de una página Web (Chaffer, 2009).

La biblioteca JQuery puede mejorar los sitios Web con independencia de su formación previa. Proporciona una amplia variedad de características, una sintaxis fácil de aprender, y compatibilidad multiplataforma robusta en un solo archivo compacto. Además, se han desarrollado cientos de plug-ins para ampliar la funcionalidad de JQuery, convirtiéndolo en una herramienta esencial para casi cualquier ocasión de programación del lado del cliente (Chaffer, 2009).

Eventos

JQuery mejora y amplía los mecanismos básicos de gestión de eventos para asignarles una sintaxis más elegante, mientras que al mismo tiempo los hacemos más potentes (Chaffer, 2009).

Evento **$(document).ready()**: El evento onload de JavaScript permite realizar acciones sobre un documento cuando se ha cargado completamente. JQuery invoca el evento $(document).ready (), cuando el DOM está complemente listo para su uso (Chaffer, 2009). Gracias a la carga realizada desde JQuery, es posible presentar la página con los estilos y características de los elementos que la componen, sin necesidad de cargarla completamente, debido a que los elementos son accesibles por nuestros scripts.

Esto tiene una gran ventaja por ejemplo, una página que presenta una galería de imágenes; dicha página puede tener muchas imágenes grandes, que podemos ocultar, mostrar, mover y manipular con jQuery. Si configuramos nuestra interfaz utilizando el evento onload, los

usuarios tendrán que esperar hasta que cada una de las imágenes se ha descargado por completo antes de que puedan utilizar la página.

Inclusive, si los comportamientos todavía no se han anexado a elementos que tienen comportamientos predeterminados (como los vínculos), las interacciones de usuario podrían producir resultados no deseados.

Sin embargo, cuando utilizamos **$ (document). ready ()** para la configuración, la interfaz se prepara mucho antes con el comportamiento correcto (Chaffer, 2009).

Otra ventaja del evento es que permite la creación de múltiples scripts de una manera más sencilla. El mecanismo **$ (document) . ready ()** gestiona toda llamada al método o métodos que se desean cargar en una página Web, añade cada uno a una cola interna de comportamientos; cuando se carga la página, se ejecutarán todas las funciones (Chaffer, 2009). De esta manera no tenemos que realizar ningún tipo de gestión de llamadas a funcioes, JQuery lo hace de manera automática y transparente.

- Evento .click().- El evento permite detectar si se seleccionó algún elemento con el puntero del ratón.
- Evento .mouseover().- El evento permite detectar si pasó el puntero del ratón por algún elemento de la página Web.
- Evento .mouseout().- El evento permite detectar si el puntero del ratón, se alejó de algún elemento.
- Evento .submit().- El evento permite enviar datos de un formulario.
- Evento .show().- El evento permite mostrar algún elemento.
- Evento .hide().- El evento permite ocultar algún elemento.
- Evento .keypress().- El evento permite detectar que una tecla del teclado se presionó.

Efectos y animaciones

Con JQuery, podemos añadir fácilmente impacto a nuestras acciones por medio de un conjunto de sencillos efectos visuales (Chaffer, 2009).

Existen momentos en los que necesitamos aplicar estilos que no se han podido fácilmente definir en una hoja de estilo (Chaffer, 2009). Mediante el uso del método .css() es posible aplicar estilos dinámicamente, este método actúa como un get y un set, para obtener el valor de una propiedad de estilo, simplemente pasamos el nombre de la propiedad como una cadena.

Por ejemplo .css('background-color'), lo que modifica el color del fondo de algún elemento o una página (Chaffer 2009).

Es posible modificar los elementos del DOM mediante eventos que ocurren sobre ellos, dichos eventos pueden ser: cuando se pasa el ratón sobre algún elemento, cuando se presiona un botón del ratón o una tecla del teclado.

JQuery permite conocer el estado de algún elemento para realizar algún cambio.

Los eventos .hide() y .show(), permiten ocultar o mostrar algún elemento del DOM (Chaffer, 2009). Si los eventos no reciben algún parámetro solo ocultan o muestran el elemento sin ninguna animación.

El parámetro que reciben los eventos hide y show, permiten realizar una animación con una duración de tiempo como cambiar la altura de un elemento, cambiar su opacidad, color, etc (Chaffer, 2009). También es posible mostrar u ocultar algún elemento dependiendo de una o varias condiciones.

Efecto .fadeIn() y fadOut(). Estos efectos son similares a hide y show solo que permiten aparecer y desaparecer algún elemento gradualmente mediante el uso de la opacidad (Chaffer, 2009). Existen tres tipos de parámetro que recibe el evento para realizar la animación:

- slow.- Muestra una animación lenta al ocultar o mostrar algún elemento.
- normal.- Muestra una animación de velocidad media.
- fast.- Muestra una animación rápida.

AJAX

¿Qué es AJAX?

AJAX es una mezcla de tecnologías en las que se encuentran JavaScript, DOM (Document Object Model) y las hojas de estilo en cascada CSS, que conviven en el mundo de la Web, donde las nuevas tecnologías y la información llegan a millones de personas en cuestión de milisegundos (Perry, 2007).

El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML" (Eguíluz 2008).

AJAX es una nueva tecnología que permite realizar la presentación de información sin necesidad de recargar completamente una página Web.

Antecedentes de la tecnología AJAX.

La historia de las tecnologías que conforman AJAX se remonta al año 1995 con la iniciativa de Microsoft en el desarrollo de Scripting Remoto, con la introducción del elemento iframe en Internet Explorer 3 y el elemento layer de Netscape 4.

Los elementos de Microsoft y Netscape compartían el atributo src(Pacheco 2011). Que permitía tomar cualquier URL solicitada por una página y cargarla mediante una página hija con JavaScript que manipula la página paterna logrando efectos muy parecidos al AJAX.

Para el año 1998 el Microsoft Remote Scripting fue una opción más elegante que la versión de Netscape, con envío de datos a través de un applet Java, el cual se puede comunicar con el cliente usando JavaScript (Pacheco, 2011).

Alex Hopmann y su equipo se encontraba desarrollando el Outlook Web Access se evaluaron 2 opciones, una de ellas fue generar páginas Web estáticas que se recargaban cada vez. La segunda fue realizar un cliente con HTML dinámico (Eguíluz, 2008).

Se optó por la segunda pero se determinó que faltaba un componente esencial: “algo” que evitara tener que enviar continuamente los formularios con datos al servidor.

Motivado por las posibilidades futuras de OWA, Alex creó en un solo fin de semana la primera versión de lo que denominó XMLHTTP. La primera demostración de las posibilidades de la nueva tecnología fue un éxito, pero faltaba lo más difícil: incluir esa tecnología en el navegador Internet Explorer (Eguíluz, 2008).

Si el navegador no incluía XMLHTTP de forma directa, el éxito del OWA se habría reducido enormemente. El mayor problema es que faltaban pocas semanas antes de que se lanzara la última beta de Internet Explorer 5 antes de su lanzamiento final. Gracias a sus contactos en la empresa, Alex consiguió que su tecnología se incluyera en la librería MSXML que incluye Internet Explorer (Eguíluz, 2008).

Microsoft presentó el Outlook Web Access provisto con la versión 2000 de Microsoft Exchange Server.

La comunidad de desarrolladores Web, primero colaborando por medio del grupo de noticias microsoft.public.scripting.remote y después usando blogs, desarrollaron una gama de técnicas de scripting remoto para conseguir los mismos resultados en diferentes navegadores. Los primeros ejemplos incluyen la librería JSRS, la introducción a la técnica imagen/cookie en el mismo año y la técnica JavaScript bajo demanda (JavaScript on Demand) en 2002.

En ese año se realizó una modificación muy importante al Microsoft Remote Scripting por parte de su comunidad de usuarios, remplazaron el applet de Java que utilizaban antes por el XMLHttpRequest.

Desde que XMLHttpRequest está implementado en la mayoría de los navegadores, raramente se usan técnicas alternativas para realizar la comunicación asíncrona con el servidor. El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005 (Eguíluz, 2008).

Ventajas de la tecnología AJAX.

Las ventajas de la tecnología AJAX son:

- XHTML y CSS: Permiten crear una presentación basada en estándares (Eguíluz, 2008).

- Garantizando el soporte y funcionamiento en la mayoría de los navegadores Web existentes.

- DOM: Para la interacción y manipulación dinámica de la presentación (Eguíluz, 2008). Permitiendo crear páginas Web dinámicas ágiles y ordenadas con poco esfuerzo.

- XML, XSLT y JSON: Para el intercambio y la manipulación de información (Eguíluz, 2008). Permiten estructurar la información que devuelve el servidor.

- XMLHttpRequest: Para el intercambio asíncrono de información (Eguíluz, 2008). La ventaja más grande del uso de AJAX es que permite enviar peticiones al servidor de manera asíncrona evitando refrescar las páginas Web.

- JavaScript: Para unir todas las demás tecnologías (Eguíluz, 2008). Utilizando un lenguaje de programación estable y potente, que permite unir varias tecnologías Web de una manera ordenada y de fácil desarrollo.

- Mejorar la experiencia del usuario (Perry, 2007): Con conexiones cliente/servidor que no interrumpen la experiencia del usuario cambiando elementos dinámicamente en regiones de la página, logrando hacer aplicaciones de una sola página muy parecidas a las aplicaciones de escritorio. Además es posible expandirlas a más

páginas según la complejidad de la aplicación.

- Uso en aplicaciones robustas: Permite desarrollar aplicaciones robustas de manera estable. Por ejemplo: Gmail y Yahoo Mail están desarrolladas con esta tecnología, lo que implica mayor soporte y actualizaciones para AJAX.

- Documentación: Amplia documentación y libros de referencia sobre la tecnología AJAX. Permitiendo consultar dudas en distintos medios.

- Plugins y Frameworks: Existen distintos frameworks y plugins que facilitan el desarrollo mediante funciones simplificadas y automatizadas.

3. ELABORACIÓN DEL SISTEMA DE BOLSA DE TRABAJO PARA LA FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN.

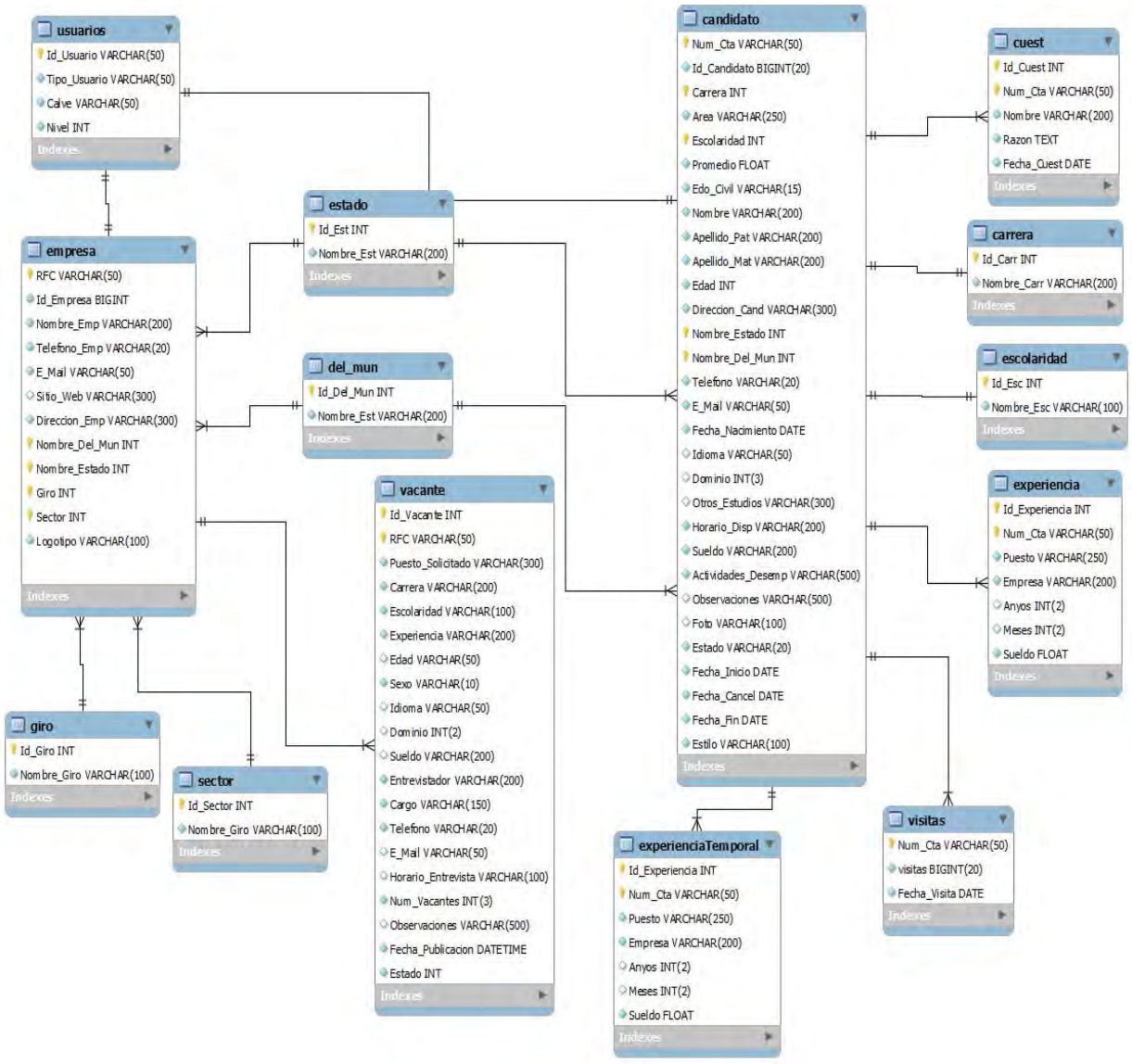
3.1. Script para la creación de la base de datos

Sin duda, algo que no debemos olvidar a la hora de crear un sitio Web es crear una base de datos, pues es aquí donde se guardará toda la información de los usuarios o de los catálogos necesarios para que nuestro sistema trabaje adecuadamente.

La base de datos debe crearse con un previo análisis de lo que el sistema requiere, se deben tomar en cuenta las tablas que son catálogos, es decir que la información se ingresa antes de poner el sitio en producción, ya que es información que se requiere para que éste funcione adecuadamente. Se debe tener en cuenta las tablas que contendrán información que el usuario llenará. Otra cosa que debemos considerar son las tablas que se relacionarán con otras, las tablas que tienen algún tipo de dependencia, entre muchas otras cosas.

El buen diseño de la base de datos es de suma importancia, pues depende de éste para el óptimo funcionamiento del sistema Web. Para lograr un buen diseño y análisis de la base de datos se requiere realizar un diagrama de entidad-relación y diagrama de tablas.

Para el caso del Sistema de la Bolsa de Trabajo para la FESC el diagrama de tablas es el siguiente.



Una vez teniendo nuestro diagrama entidad-relación, lo primero que debemos hacer es crear una base de datos, ya sea desde phpMyAdmin o desde la consola de mysql.

El comando para crear la base de datos para este sistema es:

create database db_bolsa;

Después debemos usar dicha base de datos con el siguiente comando:

use db_bolsa

Una vez seleccionada la base de datos debemos crear las tablas que formarán las bases de datos, el comando que debemos usar para crear las tablas es:

create table

Seguido de este comando debemos indicar el campo o campos de los cuales se compondrá la tabla, junto con el tipo de dato del campo (integer, varchar, text, etc.), seguido por la longitud, y en caso de que el campo sea obligatorio, es decir no deba ser nulo, se debe especificar con las palabras: *not null*. También debemos tomar en cuenta que las tablas, en la mayoría de las veces necesitan tener un campo que sea la llave primaria, es decir, un campo que no permite que se repita la información, una clave de cada registro. Por ejemplo:

CREATE TABLE `carrera` (

`Id_Carr` int(11) NOT NULL AUTO_INCREMENT,

`Nombre_Carr` varchar(200) NOT NULL,

PRIMARY KEY (`Id_Carr`)

);

Para ver el script de la creación de la base de datos del sistema de la bolsa de trabajo por favor revisar anexo B.

3.2. Funciones principales

El Sistema de la Bolsa de Trabajo para la FES Cuautitlán tiene como principal objetivo publicar vacantes que las empresas tengan disponibles, para que los candidatos registrados puedan consultarlas y que las empresas tengan la posibilidad de buscar candidatos para esas vacantes. Facilitando así que las vacantes sean ocupadas y colocar a los candidatos (alumnos y exalumnos) en el ámbito laboral.

Debemos tomar en cuenta que existen 3 actores principales:

- La empresa.
- El candidato.
- El administrador de sistema.

Lo primero que presenta el sistema es una pantalla en la cual se podrá:

- Iniciar Sesión si es usuario registrado.
- Registrarse si es Candidato.
- Registrarse si es Empresa.
- Recuperar contraseña.

Dependiendo de la opción elegida y del actor que sea, será lo que muestre la siguiente pantalla.

A continuación se mostrará lo que se permite hacer en cada caso:

3.2.1 En caso de ser Candidato

Significa que es un alumno o exalumno de la FES Cuautitlán, éste deberá registrarse como usuario del sistema, de tipo "Candidato", al elegir esta opción se mostrará un segundo formulario donde podrá registrar sus datos personales, académicos y laborales. También esta un apartado donde se puede registrar la experiencia del candidato en otras empresas, creando así su perfil. Dicho perfil podrá ser modificado y actualizado en el momento que el candidato lo desee sin la necesidad de acudir físicamente a la FES Cuautitlán.

Dicho perfil ayuda a las empresas a encontrar un candidato de acuerdo a sus necesidades.

El candidato puede a su vez, buscar vacantes ya sea por Puesto Solicitado o por Escolaridad y si alguna le interesa, puede ponerse en contacto con la empresa que la publico.

Es así como las empresas y los candidatos se ponen en contacto, logrando el objetivo de una Bolsa de Trabajo.

La cuenta del candidato puede caducar si éste no visita la página con regularidad, ya sea para actualizar su información o para buscar vacantes más recientes, con esto se busca evitar que las empresas vean un candidato que ya esté trabajando en otro lugar, o simplemente no le interese las vacantes publicadas.

En caso de que caduque la cuenta, cuando entre de nuevo el candidato deberá decir la razón por la cual abandono el sistema, esto con la finalidad de conocer de manera general el motivo por el cuál dejan de visitarlo y si es por falta de motivación por parte de la Bolsa de Trabajo (tanto del sistema como del procedimiento) poder mejorar el servicio. O si es por haber encontrado un trabajo, y saber también si lo encontró con ayuda del sistema de la bolsa de trabajo o si lo encontró por un motivo externo.

Todo esto es para tener una idea general del Sistema de la Bolsa de Trabajo, conocer su utilidad, sus debilidades y poder mejorar el sistema o el procedimiento.

3.2.2. En caso de ser Empresa

Deberá registrarse como usuario del sistema, de tipo "Empresa", después de esto aparecerá un segundo formulario que pedirá los datos de la empresa, después de haber concluido este paso, se puede crear una vacante o simplemente ver su perfil. Aquí se puede modificar el perfil, administrar vacantes, cambiar contraseña y cerrar la sesión.

En administrar vacantes se puede crear una nueva vacante, ver las existentes y éstas pueden eliminarse o modificarse, según las necesidades de la empresa.

La empresa puede buscar candidatos ya sea por el Área en que se desenvuelve el posible candidato o por su carrera. Se mostrará una lista con un breve resumen del perfil de cada candidato que haya cumplido con la búsqueda de la empresa. Si la empresa se interesa por alguno, con sólo un click puede elegirlo y así se mostrará el perfil completo del candidato.

3.2.3. En caso de ser administrador

En caso de ser administrador ya no se debe registrar, pues su registro se hace directamente en la base de datos, sólo deberá iniciar la sesión.

En la pantalla del administrador tiene un menú el cuál tiene varias opciones, las cuales son:

- Candidato
- Empresa
- Vacante
- Reportes
- Cerrar Sesión

Además del menú, se despliega una lista de los candidatos registrados, la lista muestra entre otros campos el de estado del candidato, es decir si está activo o inactivo. Este estado solo el administrador puede cambiarlo y es cuando el candidato visita la FES Cuautitlán, para verificar que sea candidato autorizado.

Al ser candidato inactivo quiere decir que se hizo el registro dentro del sistema, pero que aún no está habilitado para buscar vacantes ni para que las empresas lo puedan encontrar dentro de sus búsquedas.

Al ser candidato activo quiere decir que el candidato se registró en el sistema, fue a la FES Cuautitlán para que el administrador del sistema comprobara que es un alumno o exalumno de la FES y diera el "permiso" para que ahora si el candidato pueda buscar vacantes, actualizar su información y ser visible para las empresas.

El administrador puede editar el perfil del candidato, eliminarlo o buscar uno en específico, ya sea por su número de cuenta, nombre, correo electrónico o carrera.

Con la opción de "Empresa" se puede ver la lista de las empresas registradas, el administrador puede editar su información o eliminarla. Además podrá buscar alguna empresa en específico ya sea por su RFC, el nombre o por correo electrónico.

En la opción de "Vacante" se muestra la lista de las vacantes registradas, éstas se pueden buscar ya sea por el título de la vacante, o el RFC de la empresa que la publicó. El administrador puede editarlas o eliminarlas.

En la opción de "Reporte" se puede ver la lista de los candidatos que han entrado más recientemente, mostrando el Número de Cuenta, el nombre, las visitas que ha hecho al sitio, el estado del candidato y la fecha de la última visita de cada candidato. Por medio de un calendario también se puede indicar la fecha en la que se desea ver quién visitó el sitio.

Por último en la opción de "Cerrar Sesión" que está presente en cualquier caso, ya sea empresa, candidato o administrador es para salirse o cerrar la sesión en la que se encuentren.

4. DOCUMENTACIÓN DEL SISTEMA DE BOLSA DE TRABAJO PARA LA FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

4.1 Manual De Usuario

4.1.1 Introducción

Este manual tiene como objetivo ser una guía para conocer las secciones que conforman el sistema de bolsa de trabajo, con el fin de hacer un correcto uso del mismo y así lograr obtener el mayor provecho a las funcionalidades que ofrece dicho sistema.

Cuenta con una interfaz de usuario muy amigable que lleva de la mano al usuario para realizar una navegación fácil y ágil de manera muy intuitiva.

El sistema cuenta con una jerarquía de usuarios basada en roles que pueden ser: candidato, empresa o administrador del sistema.

4.1.2 Inicio de Sesión

Para empezar tenemos la pantalla de inicio de sesión, aquí es donde el usuario ya registrado puede iniciar sesión con sus datos: usuario y contraseña Ver Fig. No. 1.

Fig. No. 1: Inicio de Sesión.

Dentro de esta página también se encuentran las opciones de Recuperar Clave, Crear Cuenta de Candidato y Crear Cuenta de Empresa.

Crear cuentas de usuario

El sistema permite crear 2 tipos de usuario los cuales son: Candidato y Empresa.

Al dar click en la opción Candidato esta mostrará la siguiente pantalla Ver Fig. No. 2

Crear cuenta de candidato.

Fig. No. 2: Crear cuenta de candidato.

Una vez creada la cuenta de usuario con el rol candidato éste deberá ingresar sus datos personales en el sistema Ver Fig. No. 3

Formulario para registrar candidato:

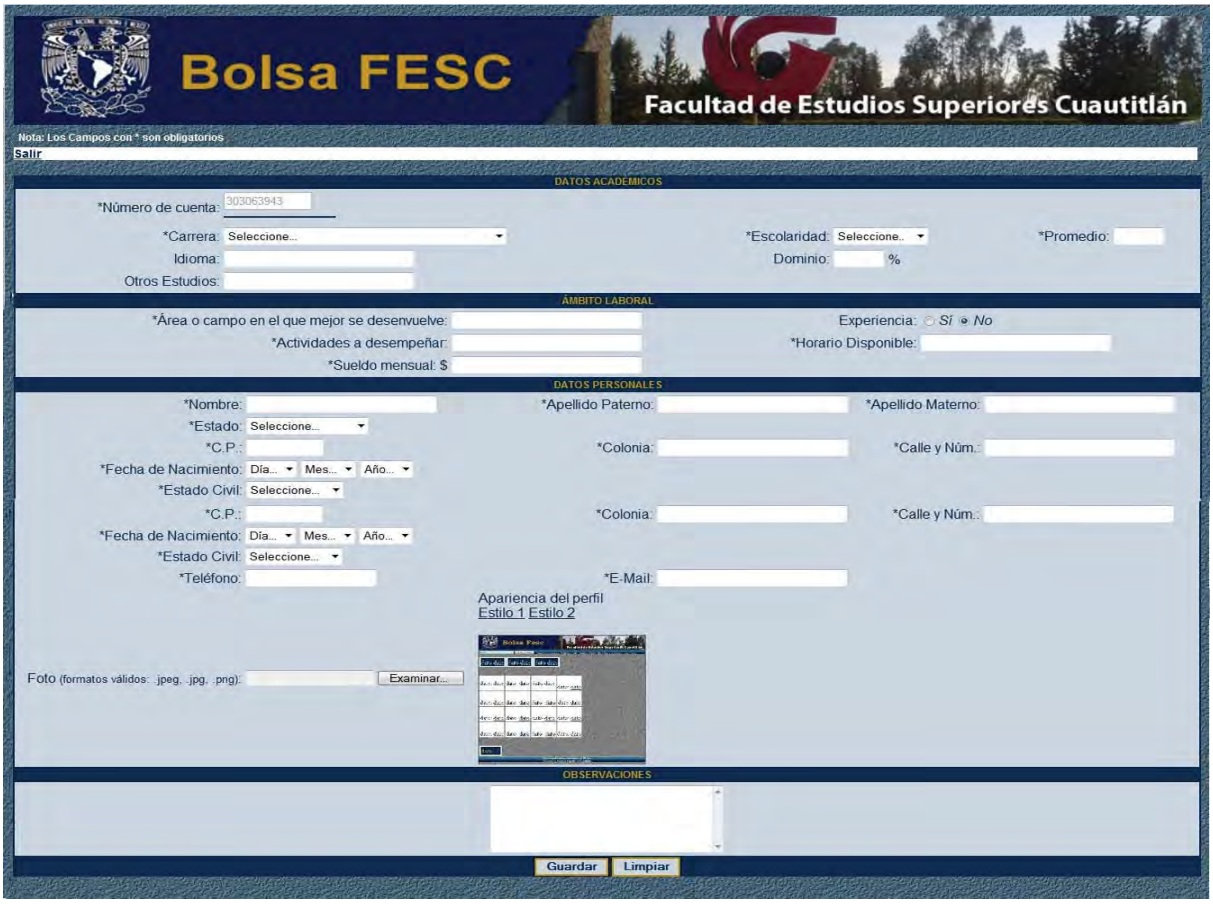


Fig. No. 3: Registro de candidato

Si el candidato cuenta con experiencia laboral previa, tiene la posibilidad de ingresar esa información, dentro de un formulario especial Ver Fig. No. 4.

Formulario para registrar experiencia del candidato:

Fig. No. 4: Formulario registro de experiencia del candidato.

Dicho formulario aparece en el momento que el candidato selecciona “Sí” cuando se le pregunta si tiene experiencia.

Si el candidato no tiene experiencia puede continuar su registro sin ingresar datos de la experiencia.

Una vez llenado el formulario correctamente el sistema mostrará un mensaje de confirmación Ver Fig. No. 5:

Fig. No. 5: Mensaje de aviso de registro de la cuenta candidato.

Con lo anterior finaliza la creación de la cuenta de usuario candidato. Sin embargo deberá acudir a la dirección de bolsa de trabajo ubicada en la FESC Cuautitlán Campo 4. Para el administrador verifique si es candidato autorizado para cambiar el estado a activo.

Al dar click en la opción Crear cuenta de Empresa el sistema un formulario Ver Fig. No. 6.

Fig. No. 6: Crear cuenta de empresa.

Crear cuenta de empresa.

Una vez creada la cuenta de usuario, se debe llenar un formulario donde se registraran los datos de la empresa Ver Fig. No. 7.

.

Formulario para registrar empresa:

Fig. No. 7: Formulario para registrar empresa.

Una vez ingresados los datos al formulario se muestra el perfil de la empresa Ver Fig. No.8.

Fig. No. 8: Perfil Empresa.

Con lo anterior finaliza el registro usuario con el rol de empresa.

Para el caso de la empresa su cuenta queda activada en el momento de terminar su registro.

Recuperar Contraseña

Si el usuario desea recuperar su contraseña deberá seleccionar la opción recuperar contraseña, de esta manera el sistema mostrará un formulario Ver Fig. No. 9 donde solicita su RFC para las empresas o Número de cuenta para los candidatos y su correo electrónico que previamente proporcionó al sistema.

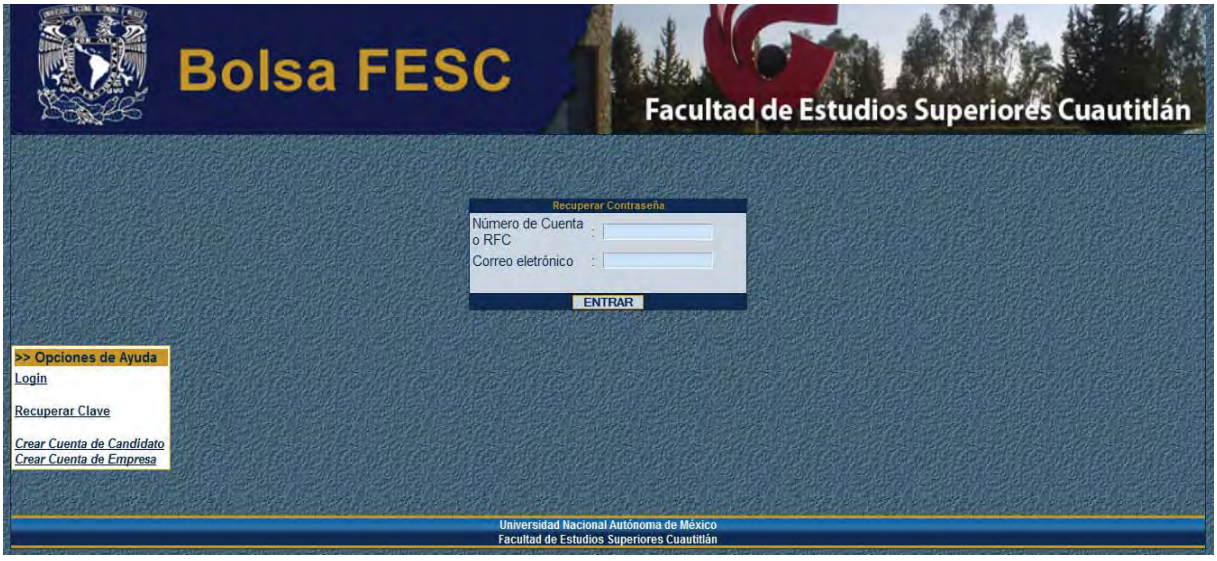


Fig. No. 9: Formulario para recuperar contraseña.

Una vez realizado lo anterior el sistema generará una nueva contraseña y enviará un correo electrónico con la nueva clave de acceso para el usuario.

4.1.3. Perfiles

Con la información de los usuarios se crea un perfil de cada uno de ellos según sea su rol, esta forma de mostrar la información ayuda a los usuarios a consultar los datos tanto de las empresas como de los candidatos de una manera ordenada y fácil.

Perfil Candidato

Este perfil muestra los datos personales y la experiencia profesional del candidato Ver Fig. No. 10 cuando es consultado por las empresas.

Fig. No. 10: Perfil Candidato.

Barra de navegación

La barra de navegación Ver Fig. No. 11 permite al candidato realizar las acciones:

Barra de navegación del candidato:

Fig. No. 11: Barra de navegación.

- Buscar vacantes: Esta opción permite realizar búsquedas de vacantes, que se mostrarán ordenas en una lista de 10 elementos Ver Fig. No. 12, que pueden ser consultadas dando click sobre algún elemento Ver Fig. No. 13, una vez desplegado el elemento es posible consultar el perfil de la empresa que publicó la vacante.

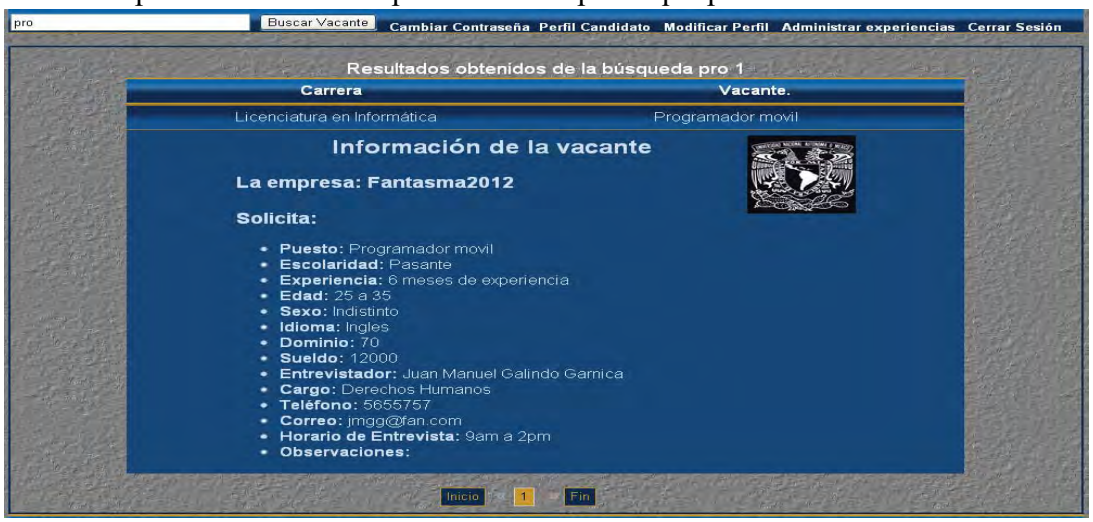


Fig. No. 12: Información de la vacante.

Perfil de la empresa consultado por el candidato:

Fig. No. 13: Vista perfil candidato.

- Cambiar Contraseña: Permite modificar la contraseña al Usuario Ver Fig. No. 14.

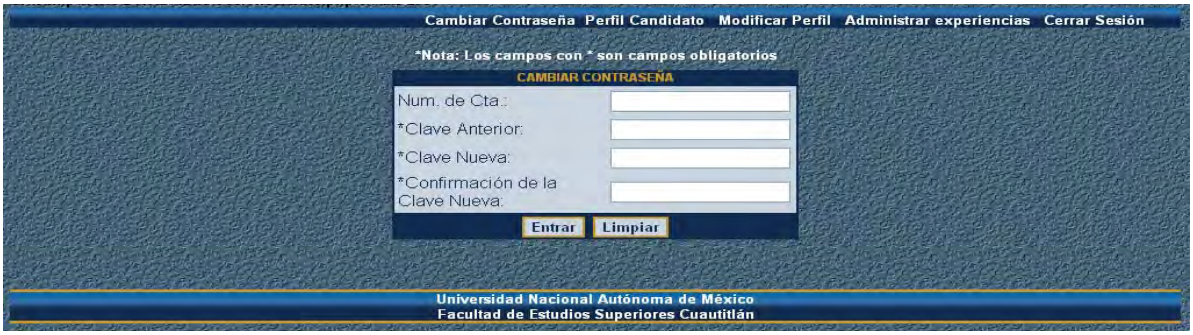


Fig. No. 14: Formulario para cambiar contraseña.

- Perfil candidato: Permite mostrar el perfil del candidato Ver Fig. No. 15.

Fig. No. 15: Vista del perfil de candidato desde el área de administración.

- Modificar Perfil: Esta opción permite modificar los datos del perfil del candidato mediante un formulario. Ver Fig. No. 16

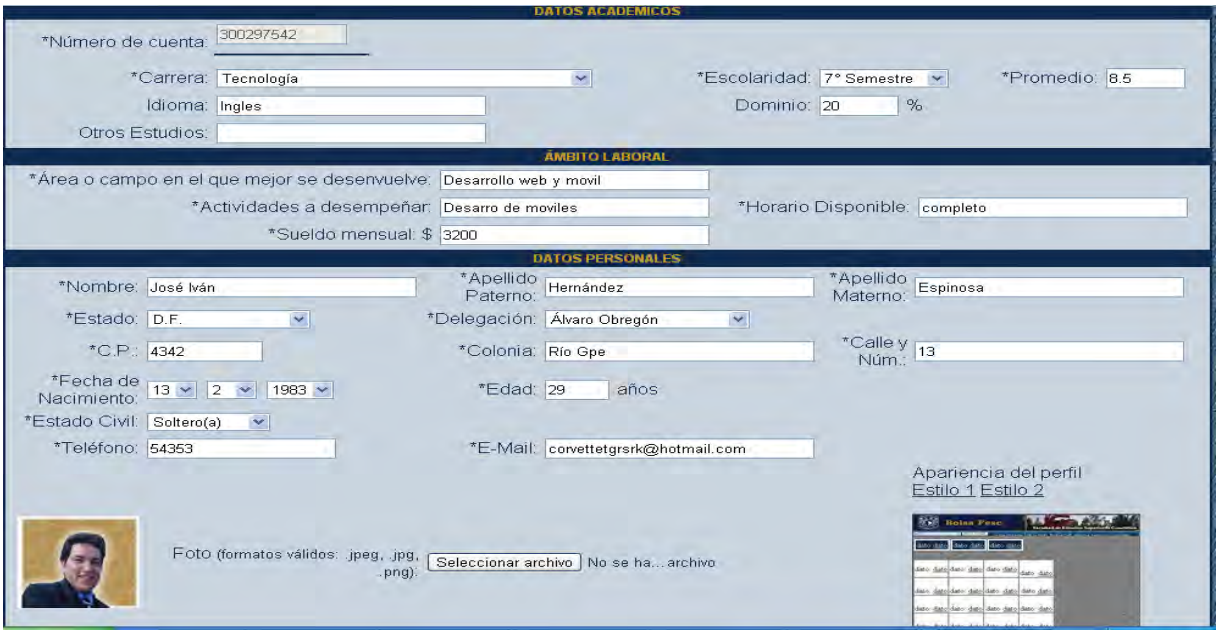


Fig. No. 16: Formulario para modificar los datos del perfil del candidato.

- Administrar Experiencia: Esta opción permite administrar la experiencia laboral del Candidato modificando o eliminando las experiencias disponibles Ver Fig. No. 17. Muestra una lista ordenada por 10 elementos paginados de las experiencias registradas (en caso de ser más de 10 exp) de existir esta sección, es decir, si el candidato tiene experiencias registradas, cuenta con una barra específica de navegación que permite: crear Ver Fig. No. 18 o mostrar las experiencias y buscar Ver Fig. No. 19 y regresar al perfil del candidato.

Fig. No. 17: Vista Principal de la sección Administrar Experiencias.

Fig. No. 18: Vista Previa del formulario crear experiencias.

- Es Posible realizar búsqueda de experiencias.

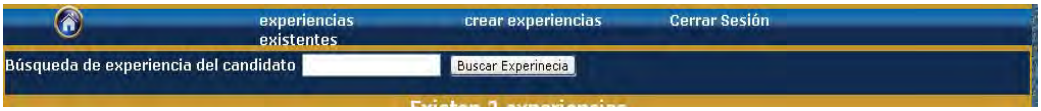


Fig. No. 19: Vista del formulario para buscar experiencias.

Perfil Empresa

Este perfil muestra los datos de la empresa y las vacantes publicadas Ver Fig. No. 20, podrán ser consultadas por los candidatos.

Fig. No. 20: Vista del perfil de la empresa.

Barra de navegación

La barra de navegación Ver Fig. No. 21 permite a la empresa realizar las acciones: Cambiar Contraseña, Perfil Empresa, Modificar Perfil, Administrar Vacantes, Cerrar Sesión.

Fig. No. 21: Vista de la barra de navegación de la empresa

- Buscar candidatos: Esta opción permite realizar búsquedas de candidatos Ver Fig. No. 22, que se mostrarán ordenados en una lista de 10 elementos que pueden ser consultados dando click sobre algún elemento, una vez desplegado el elemento es posible consultar el perfil del candidato.

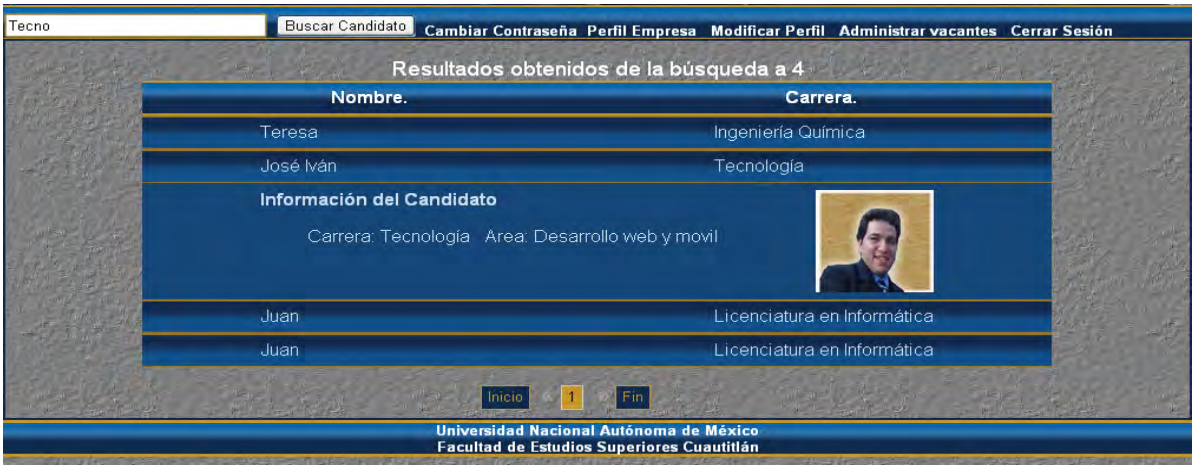


Fig. No. 22: Vista de los resultados obtenidos por el buscador de candidatos.

- Cambiar Contraseña: Permite modificar la contraseña al Usuario. Ver Fig. No. 23.

Fig. No. 23: Formulario para cambiar la contraseña de la empresa.

- Vista del Perfil Empresa: Permite mostrar el perfil de la empresa Ver Fig. No. 24.

Fig. No. 24: Vista del perfil empresa

- Modificar Perfil: Esta opción permite modificar los datos del perfil de la empresa mediante un formulario. Ver Fig. No. 25.

Fig. No. 25: Formulario para modificar los datos de la empresa.

- Administrar Vacantes: Esta opción permite administrar las vacantes publicadas por la empresa, modificándolas o eliminándolas ver Fig. No. 26 y 27. Muestra una lista ordenada por 10 elementos paginados de las vacantes registradas si es que hay más de 10, de existir esta sección, es decir si la empresa ha registrado vacantes, cuenta con una barra específica de navegación que permite crear o mostrar las vacantes y regresar al perfil de la empresa.

Fig. No. 26: Vista Previa de la sección Administrar Vacantes.

Fig. No. 27: Vista previa de la sección crear vacante.

- Buscar Vacante: Es Posible realizar búsqueda de vacantes. Ver Fig. No. 28

Fig. No. 28: Vista para buscar vacantes.

.

Perfil Administrador

En caso de ser el administrador del sistema de la bolsa de trabajo, se puede monitorear todo lo que se ha registrado. El administrador tiene como primera pantalla la lista de los candidatos registrados ver Fig. No. 29, puede ver sus datos, editarlos e incluso eliminarlos en caso de mal uso. Esto con la finalidad de tener un control de las personas que se registran, de saber que realmente son alumnos o exalumnos de la FES Cuautitlán.

Activación de las cuentas de usuario

Entre la información que destaca del candidato es el estado, es decir, si esta activo o inactivo. Un candidato puede estar inactivo por que se acaba de registrar y no se ha puesto en contacto con el administrador del sistema para que lo active, o también porque ya tiene tiempo sin visitar el sistema de la bolsa de trabajo, y al no actualizar la información ni buscar vacantes se puede entender que el candidato ya fué colocado en alguna empresa.

Si el administrador desea activar la cuenta, deberá dar click en el estado del candidato que muestra la palabra Inactivo, con ello el estado del candidato será Activo y ya podrá hacer uso de su cuenta.

Fig. No. 29: Vista del área de administrador para consultar candidatos.

El administrador del sistema, puede también revisar la información de la empresa ver Fig. No. 30, puede editarla y borrarla en caso de descubrir alguna irregularidad.

Fig. No. 30: Vista del área del administrador para consultar empresas.

El administrador puede a su vez revisar, modificar y eliminar las experiencias de los candidatos y vacantes de las empresas. Ver Fig. No.31

Hay una sección donde se puede consultar los candidatos que entraron al sistema más recientemente.

El administrador puede también marcar la fecha que desee para conocer quiénes entraron a consultar o actualizar información. Esto con la finalidad de saber que tan visitado es el sitio, si es de utilidad, etc.

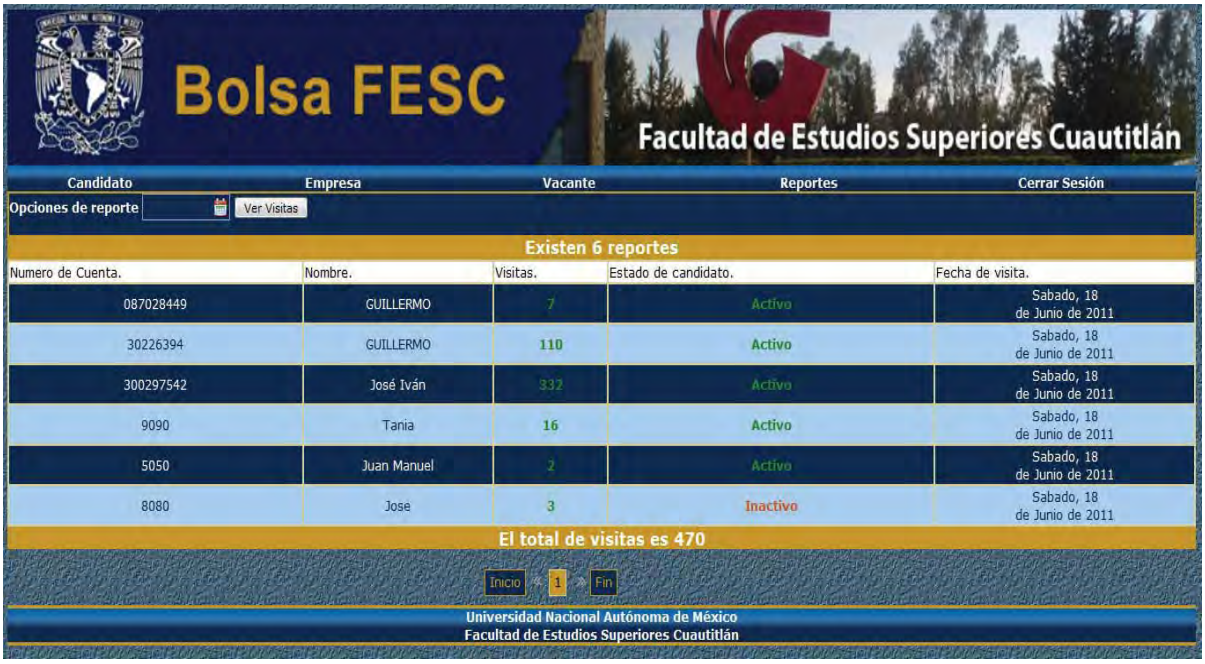


Fig. No. 31: Vista del área de administrador para consultar las visitas al sitio.

Por último, en los 3 casos esta la opción de Cerrar Sesión, donde con solo dar click sale de la sesión actualmente activa.

4.2. Manual Técnico

4.2.1. Presentación

Este documento integra las características técnicas que requiere el Sistema de la Bolsa de trabajo de la Facultad de Estudios Superiores Cuautitlán, para ejecutarse adecuadamente.

Asimismo se definen los componentes que lo integran, a fin de que quede establecido cómo se interrelacionan y la gente de Informática conozca la estructura del Sistema y pueda hacerle las adecuaciones que permitan su evolución de acuerdo al crecimiento de dicho Sistema.

4.2.2. Requerimientos del sistema

Hardware

Se recomienda que el equipo donde se vaya a instalar la aplicación cuente con: 50 megas libres de disco duro para almacenar la aplicación y una gran capacidad de almacenaje en cuestión de datos, así como cuente con un procesador de 1Ghz como mínimo y 1GB de memoria RAM por lo menos.

Sistema Operativo

El sistema trabaja en las plataformas Linux/Windows.

Linux

Se requiere que el sistema cuente con los comandos: TAR que permite desempaquetar el archivo que contiene la aplicación, el comando GZIP o BZIP2, los cuales sirven para comprimir archivos. Estos programas se utilizaran para comprimir los archivos de respaldo de la aplicación.

Para realizar transferencias por medio de ftp. Se recomienda utilizar el programa filezilla disponible en http://filezilla-project.org/download.php.

Si se desea aumentar la seguridad para abrir o transferir archivos de la aplicación, se recomienda utilizar el programa secure shell (ssh) en lugar de filezilla u otro programa de ftp, el programa se encuentra disponible en las distintas versiones del sistema operativo Linux. Este programa posiblemente será utilizado por el administrador del sitio (Webmaster) para que pueda conectarse al servidor y pueda hacer cambios en los programas del Sistema o pueda conectarse a la base de datos.

Windows

Se requiere que el sistema cuente con el programa winzip, winrar o 7zip. Que permiten desempaquetar el archivo que contiene la aplicación, así mismo los programas para desempaquetar también funcionan para comprimir los archivos de respaldo de la aplicación.

Para realizar transferencias por medio de ftp. Se recomienda utilizar el programa filezilla disponible en http://filezilla-project.org/download.php.

Si se desea aumentar la seguridad para abrir o transferir archivos de la aplicación, se recomienda utilizar el programa secure shell (ssh) en lugar de filezilla u otro programa de ftp, el programa se encuentra disponible en http://www.openssh.org/. Este programa posiblemente será utilizado por el administrador (Webmaster) para que pueda conectarse al servidor y pueda hacer cambios en los programas del Sistema o pueda conectarse a la base de datos.

4.2.3. Manejador de base de datos

Linux

El Sistema funciona con el manejador de base de datos MySQL 5.x.

Windows

El sistema funciona con el software wamp que incluye el manejador de base de datos MySQL. Se ha probado con las versiones 1.6.6 y 2.2 de wamp.

4.2.4. Lenguaje de programación PHP

Linux

Se requiere que se instale la versión 5.2.* del lenguaje de programación PHP.

PHP debe ser configurado para que pueda conectarse a MySQL. Para realizar esto, al momento de instalar PHP, se deberá de agregar desde la consola la siguiente opción al comando configure de PHP con las respectivas rutas del software especificado:

--with-mysql.

Windows

El sistema funciona con el software wamp que incluye el lenguaje de programación PHP. Se ha probado con las versiones 1.6.6 y 2.2 de wamp Server.

4.2.5. Servidor Web

Linux

El servidor web utilizado con el Sistema deberá ser Apache. Se ha probado con las versiones 2.059 y 2.2.19. Este servidor puede descargarse de http://www.apache.org .

Se deberá configurar para que Apache interprete además de los archivos .php como programas de php, también los .html. Esto se hace agregado un AddType en el archivo de configuración de Apache (httpd.conf) de la siguiente manera: AddType application/x-httpd-php .php .html.

Windows

El sistema funciona con el software wamp que incluye el servidor web Apache. Se ha probado con las versiones 1.6.6 y 2.2 de wamp.

4.2.6. Servidor de correo electrónico

Se requiere que se instale un servidor de correo para que pueda enviar correos electrónicos, ya que algunos módulos del Sistema requieren que se envíen correos electrónicos.

4.2.7. Software integrado y utilizado en el Sistema

El software mencionado a continuación se encuentra integrado en el Sistema dentro del directorio /calendario:

- Calendario.

Y las bibliotecas de JavaScript se encuentran en el directorio js:

- JQuery.
- JQuery paginate.
- Shadowbox.

La instalación que se describirá a continuación, parte de la base de que ya se encuentra configurado correctamente el servidor donde estará funcionando el Sistema.

Antes de iniciar la instalación del Sistema, es importante decidir que archivo de respaldo del Sistema se desea transferir. Si el servidor tiene instalado el programa bzip2, entonces se deberá transferir el archivo con extensión .bz2, en caso de que se tenga instalado el programa gzip, se deberá transferir el archivo con extensión .gz y si no se tienen instalado ninguno de estos programas deberá transferirse el archivo .tar.

NOTA: Para ejemplificar la instalación, se nombrará al archivo de respaldo del Sistema: sistemaBolsadetrabajo.tar o sistemaBolsadetrabajo.tar.gz o sistemaBolsadetrabajo.tar.bz2. La cuenta de usuario del sistema operativo que se utilizará será webmaster. El HOME de la cuenta de usuario webmaster estará situado en /var/www/site. El usuario de la base de datos será bolsafes_bolsa y su contraseña será bolsa4F24. El nombre de la base de datos será dbbolsa.

/*prompt$ Indica el prompt de la sesión del usuario en el sistema operativo.

promptMysql$ Indica el prompt de la sesión del usuario en la base de datos.

/ruta/comando/tarGNU/ Se deberá poner la ruta donde se encuentra instalado el binario del comando tar en su versión GNU, NO LA VERSIÓN NATIVA DEL COMANDO TAR DE SOLARIS. Esto es con el fin de que si se desean hacer pruebas en equipos que tienen instalados el sistema GNU/Linux no les cause conflictos, además que es un programa más recomendado que el tar nativo de Solaris.

/ruta/comando/bzip2/ Se deberá poner la ruta donde se encuentra instalado el binario del comando bzip2.

/ruta/comando/gzip/ Se deberá poner la ruta donde se encuentra instalado el binario del comando gzip.

4.2.8. Instalación del sistema

Linux

Para iniciar la instalación del Sistema, se requiere que sea transferido el archivo de respaldo del Sistema al HOME de la cuenta webmaster.

Una vez transferido el archivo de respaldo del Sistema, se deberá abrir una sesión de la cuenta webmaster.

Ya que se inició la sesión, se deberá proceder a descomprimir el archivo de respaldo:

En caso de que sea el archivo sistemaBolsadetrabajo.tar.bz2 se deberá ejecutar lo siguiente:

prompt$ /ruta/comando/bzip2/bunzip2 sistemaBolsadetrabajo.tar.bz2

Una vez que se descomprimió el archivo, nos generará el archivo sistemaBolsaderabajo.tar, el cual no contiene ya la extensión bz2.

En caso de que sea el archivo sistemaBolsadetrabajo.tar.gz se deberá ejecutar lo siguiente:

prompt$ /ruta/comando/gzip/gunzip sistemaBolsadetrabajo.tar.gz

Una vez que se descomprimió el archivo, nos generará el archivo sistemaBolsadetrabajo.tar, el cual no contiene ya la extensión gz.

En caso de que sea el archivo sistemaBolsadetrabajo.tar o que se haya realizado lo anterior para los archivos .bz2 o gz, se deberá proceder a lo siguiente:

prompt$ /ruta/comando/tarGNU/tar -xvf sistemaBolsadetrabajo.tar

La consola mostrará la salida de los archivos y directorios que se están desempaquetando del archivo de respaldo el cual generará el directorio **BolsaPublic,** Este directorio contienen todos los programas y archivos que utiliza el Sistema.

El siguiente paso será configurar los valores de la conexión a la base de datos, para ello, se deberá editar el archivo **Cnx.class.php**, cambiar los datos de las variables de para poder conectar a la base de datos.

```
class Cnx
{
   private $host = 'localhost';
   private $user = 'bolsafes_bolsa';
   private $password = ' bolsa4F24';
   private $base = ‘bolsafes_bolsa';
}
```

Se deberán guardar los cambios del archivo

Una vez instalado

Si todo está correctamente funcionando, la instalación del Sistema a finalizado.

Windows

La instalación que se describirá a continuación, parte de la base de que ya se encuentra configurado correctamente el servidor Wamp donde estará funcionando el Sistema.

Antes de iniciar la instalación del Sistema, es importante decidir que archivo de respaldo del Sistema se desea transferir. Si el servidor tiene instalado el programa winrar, winzip o 7zip.

NOTA: Para ejemplificar la instalación, se nombrará al archivo de respaldo del Sistema: sistemaBolsadetrabajo.zip o sistemaBolsadetrabajo.rar. El directorio de instalación del sistema deberá ser desempaquetado en el directorio C:\wamp\www, el usuario de la base de datos será dbbolsa y su password será bolsa4F24.

El siguiente paso es crear la base de datos, para ello se deberá iniciar el servidor wamp desde menú Inicio, Todos los programas y WampServer.

Una vez iniciado deberá seleccionarse Wamp, MySQL. Esto mostrará la consola de MySQL, se deberá teclear el password de usuario de la base de datos. Una vez ingresado el password se deberá crear la base de datos con el comando: create database bolsafes_bolsa. Posteriormente se debe usar la base de datos con el comando:

use bolsafes_bolsa.

Lo siguiente es ingresar a la consola de símbolo de sistema de windows, desde menú Inicio, ejecutar, teclear el comando cmd. En la consola debe ejecutarse el comando MySQL desde la ruta C:\wamp\mysql\bin > mysql -u bolsafes_bolsa -p bolsafes_bolsa < C:\bolsafes_bolsa\bolsafes_bolsa.dump. Lo que cargara el contenido y estructura de la base de datos.

NOTA: Si es la primera vez que se utilizará la aplicación deberá crearse un usuario con permisos de administrador desde la aplicación phpMyAdmin.

En WAMP, phpMyAdmin, seleccionar la tabla usuarios, seleccionar la opción Insertar, ingresar el id del usuario se recomienda sea el rfc o número de cuenta, el tipo de usuario debe ser Administrador, contraseña y nivel de usuario deberá ser 1.

Una vez instalada la base de datos se puede iniciar el sistema desde la dirección http://localhost/BolsaFESC/ que desplegará la página de login del sistema Bolsa de trabajo de la Facultad de Estudios Superiores Cuautitlán.

4.2.9. Respaldo del Sistema

Una tarea que es importante realizar preferentemente diario, es el respaldo de la información tanto de la base de datos, como del sistema de archivos del directorio www. Es recomendable que dicho respaldo se realice en los horarios donde el Sistema casi no tenga demanda, por ejemplo, en la noche o muy temprano.

NOTA: Para ejemplificar el respaldo de la información del Sistema, se nombrará al archivo de respaldo del Sistema: sistemaBolsaDeTrabajo_2013-02-01.zip o sistemaBolsaDeTrabajo_2013-02-01.gzip. El archivo de respaldo de la base de datos que se utilizará será bolsafes_bolsa.dump. El archivo de respaldo bolsafes_bolsa.sql generado por la aplicación phpMyAdmin funciona de la misma manera que el archivo con extensión .dump.

Linux

Se debe cambiar al usuario webmaster, también se puede realizar el proceso con el usuario root, Posicionarse en el directorio respaldos y teclear el comando: **mysqldump -u bolsafes_bolsa -p bolsafes_bolsa > \respaldos\bolsafes_bolsa.dump**.

mysqldump -u bolsafes_bolsa -p bolsafes_bolsa > c:\ bolsafes_bolsa.dump.

Windows

El respaldo se puede hacer de 2 maneras: desde la aplicación phpMyAdmin o desde la consola del símbolo del sistema de Windows.

phpMyAdmin

Desde la aplicación wamp seleccionar phpMyAdmin, seleccionar **bolsafes,** seleccionar la pestaña exportar, seleccionar comprimido con zip y el botón continuar.

Si todo se seleccionó de manera correcta la aplicación mostrará una ventana que permite descargar el archivo **bolsafes_bolsa.sql** de la base de datos.

Consola símbolo del sistema de Windows

Desde la consola de símbolo de sistema se debe ejecutar el comando:

Posicionarse en el directorio **C:\wamp\mysql\bin> y teclear el comando: mysqldump -u bolsafes_bolsa -p bolsafes_bolsa > c:\ bolsafes_bolsa.dump.**

Si no ocurrió ningún error se debe tener el archivo **bolsafes_bolsa.dump** en la raíz de la unidad c:\.

4.2.10. Migración del Sistema a otro Servidor

Para realizar la migración del Sistema a otro servidor se deberá de realizar lo siguiente:

Realizar lo indicado en el numeral **4.2.2. Requerimientos para el Sistema.**

Se recomienda que el Sistema que está en línea se desactive un día para realizar los respaldos y hacer los movimientos correspondientes en el otro servidor. El tiempo variará dependiendo de las personas encargadas de migrar la aplicación. Aunque la instalación del Sistema no requiere ni un día completo, si todas las herramientas que requiere se encuentran bien configuradas.

Posteriormente realizar lo indicado en el numeral **4.2.9. Respaldo del Sistema.**

Ahora en el nuevo servidor realizar lo indicado en el numeral **4.2.8. Instalación del Sistema**.

4.2.11. Metodología

Los diagramas y documentos que a continuación son agregados, muestran los usuarios que actúan con el Sistema, así como las clases, archivos y esquema de tablas que comprenden el Sistema.

Diagramas de casos de uso

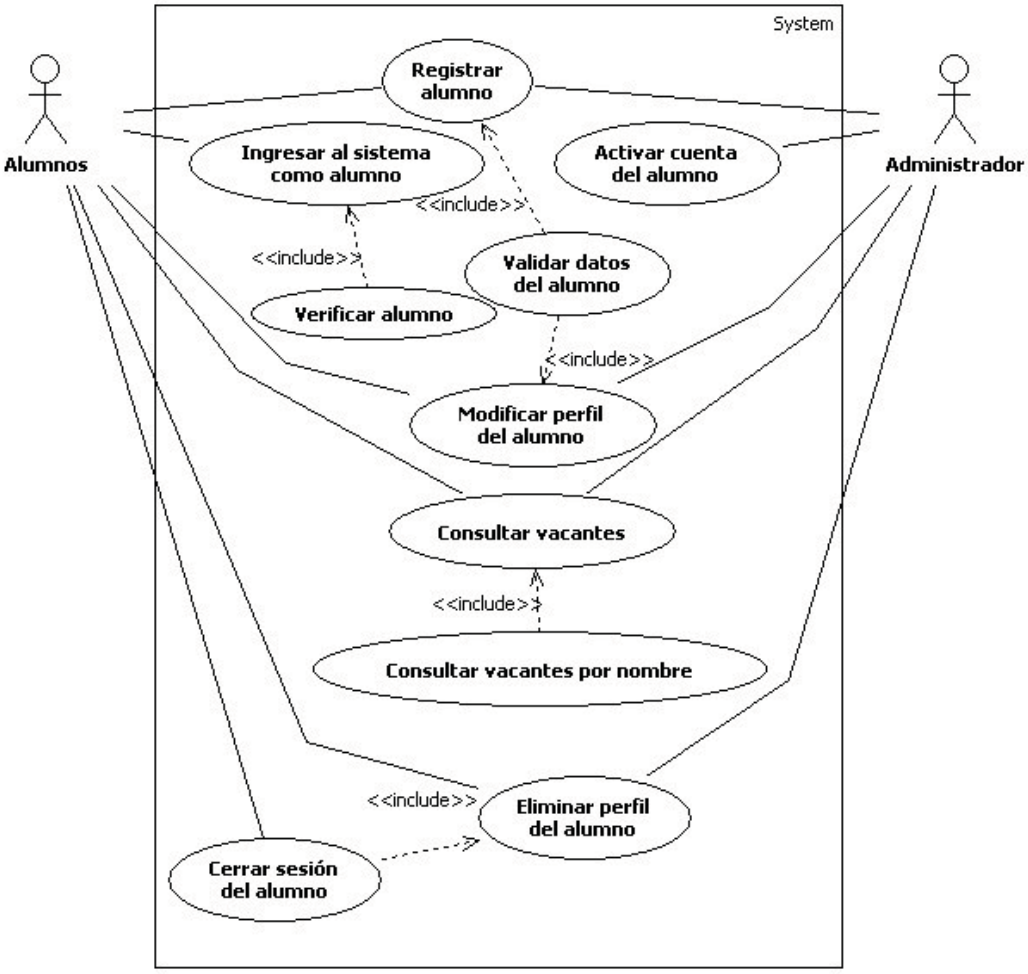


Diagrama 1: Diagrama de caso de uso de los actores: Candidato y Administrador

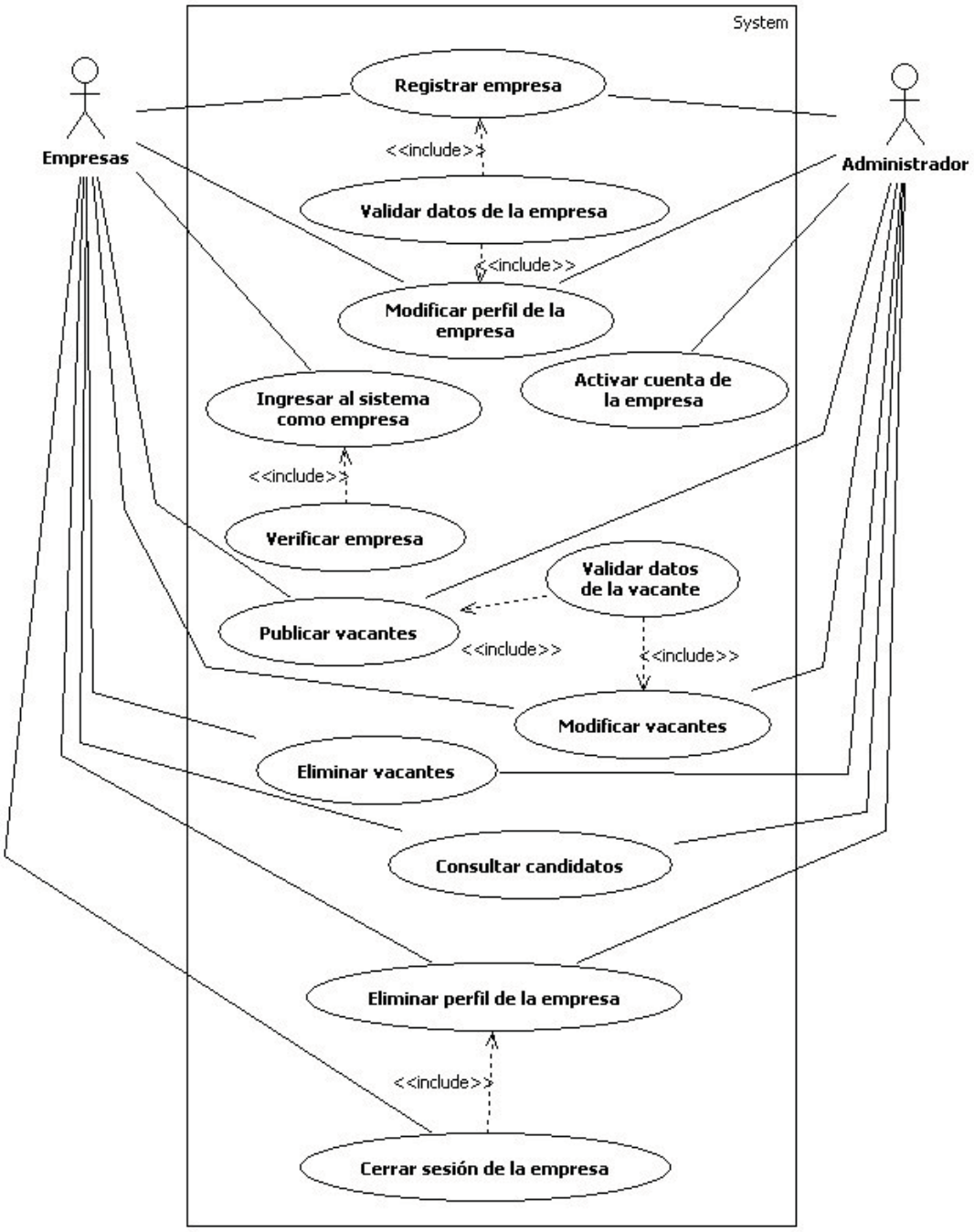


Diagrama 2: Diagrama de caso de uso de los actores: Empresa y Administrador.

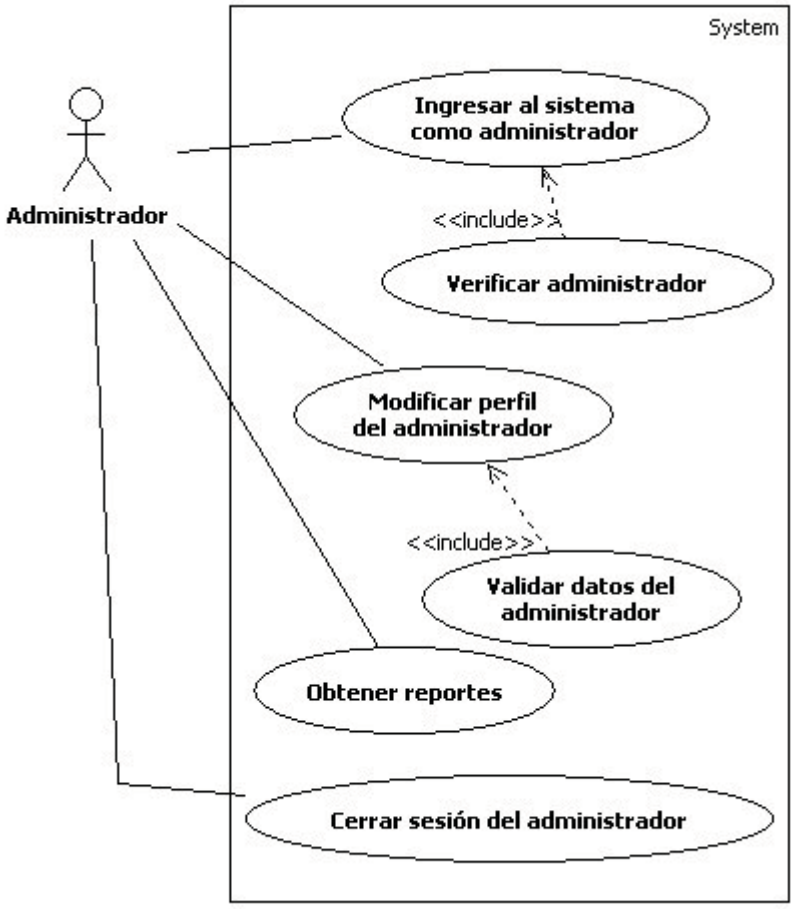


Diagrama 3: Diagrama de caso de uso del actor: Administrador.

Descripción de casos de uso

Descripción	Este Caso de Uso trata acerca del registro de perfiles de los candidatos en el Sistema en línea de bolsa de trabajo para la Facultad de Estudios Superiores Cuautitlán, en lo sucesivo "el sistema."
Flujo básico	El candidato llena el formulario de registro.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.
Precondiciones	*No existen precondiciones.*
Postcondiciones	• Validar datos del candidato.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 4: Descripción del caso de uso Registrar candidato.

Descripción	El sistema validará los datos que ingresó el candidato.
Flujo básico	Una vez que el candidato haya llenado el formulario dará clic en el botón "Guardar" y el sistema validará los datos ingresados.

Flujos alternos	• Datos incorrectos: En caso de que el candidato haya ingresado datos incorrectos en el sistema, este lo regresará al formulario de registro para que lo modifique de tal manera que los datos sean correctos.
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar candidato.
Postcondiciones	• Mensaje de confirmación: Se muestra un mensaje que confirma el registro del candidato al sistema.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 5: Descripción del caso de uso Validar datos del candidato.

Descripción	El administrador del sistema comparará los datos que el candidato ingresó al sistema con los datos reales, para verificar que el candidato es quien dice ser y así activar la cuenta del candidato.
Flujo básico	Al verificar los datos del candidato, el administrador dará clic en activar cuenta, entonces el sistema correrá dos temporizadores, uno durará 30 días, el otro 90 días.
Flujos alternos	• Datos incorrectos: En caso de que los datos del candidato no sean correctos, la cuenta no se activará y se eliminará del sistema.

Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar candidato. • Validar datos del candidato. • Ingresar como administrador. • Verificar administrador.
Postcondiciones	*No existen postcondiciones.*
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 6: Descripción del caso de uso Activar cuenta del candidato.

Descripción	Las empresas y empleadores registran su perfil en el sistema.
Flujo básico	El representante de la empresa ingresa al sistema y llena el formulario de registro.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.
Precondiciones	*No existen precondiciones.*
Postcondiciones	• Validar datos de la empresa.

Puntos de extensión	*No existen puntos de extensión.*

Diagrama 7: Descripción del caso de uso Registrar empresa.

Descripción	Procedimiento para la activación de la cuenta de empresa.
Flujo básico	Al verificar que los datos de la empresa sean correctos, la cuenta de la empresa será activada automáticamente. El administrador verificará que los datos de la empresa sean reales, de no serlo procederá a eliminar la cuenta.
Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Ingresar como administrador. • Verificar administrador.
Postcondiciones	*No existen postcondiciones.*
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 8: Descripción del caso de uso Activar cuenta de la empresa.

Descripción	Este caso de uso describe los pasos necesarios para que el candidato ingrese al sistema.
Flujo básico	El candidato deberá iniciar sesión ingresando su Número de Cta. y la contraseña que estableció cuando se registró en el sistema.
Flujos alternos	*CU Modificar perfil del alumno* *CU Consultar vacantes* *CU Agregar experiencias*
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato.
Postcondiciones	• Verificar candidato.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 9: Descripción del caso de uso Ingresar al sistema como candidato.

Descripción	El sistema validará el nombre de usuario y contraseña que ingresó el candidato.
Flujo básico	Una vez que el candidato haya ingresado su Número de Cta. y contraseña, dará clic en el botón "Entrar", el sistema verificará si la información es correcta.
Flujos alternos	- Datos incorrectos: En caso de que el nombre de usuario y/o contraseña del candidato sean incorrectos, el sistema regresará al candidato a la pantalla de ingresar al sistema como candidato.
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato. • Ingresar al sistema como candidato.
Postcondiciones	• Se muestra la pantalla de la sesión del candidato.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 10: Descripción del caso de uso Verificar candidato.

Descripción	El candidato podrá salir de su sesión en el momento que desee.
Flujo básico	El candidato da clic en el botón "Cerrar sesión."

Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato. • Ingresar al sistema como candidato. • Verificar candidato.
Postcondiciones	*No existen postcondiciones.*
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 11: Descripción del caso de uso Cerrar sesión del candidato.

Descripción	La empresa tecleará su nombre de usuario y su contraseña para entrar al sistema.
Flujo básico	El representante de la empresa tecleará su nombre de usuario, que en este caso es su RFC, y la contraseña que estableció cuando se registró en el sistema.

Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa.
Postcondiciones	• Verificar empresa.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 12: Descripción del caso de uso Ingresar al sistema como empresa.

Descripción	El sistema validará el nombre de usuario y contraseña que ingresó la empresa.
Flujo básico	Una vez que el representante de la empresa haya tecleado su nombre de usuario y contraseña, dará clic en el botón "Entrar", el sistema verificará si la información es correcta.
Flujos alternos	• Datos incorrectos: En caso de que el nombre de usuario y/o contraseña de la empresa sean incorrectos, el sistema regresará al representante de la empresa a la pantalla de ingresar al sistema como empresa.

Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa.
Postcondiciones	• Se muestra la pantalla principal de la sesión de la empresa.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 13: Descripción del caso de uso Verificar empresa.

Descripción	Las empresas podrán salir de su sesión en el momento que deseen.
Flujo básico	El representante de la empresa da clic en el botón “Cerrar sesión.”
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa.
Postcondiciones	*No existen postcondiciones.*
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 14: Descripción del caso de uso Cerrar sesión de empresa.

Descripción	El candidato podrá consultar las vacantes existentes y vigentes que publiquen las empresas y empleadores.
Flujo básico	Este caso de uso comienza cuando el candidato está en su perfil y realiza los siguientes pasos: • El candidato realizará una búsqueda de vacantes ingresando una palabra clave. • El sistema muestra los resultados obtenidos de la búsqueda en una lista de 10 elementos, en caso de tener más, mostrará más páginas. • En caso de encontrar una coincidencia, el candidato podrá revisar los datos de la vacante así como el sitio Web de la empresa que publica dicha vacante.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato. • Ingresar al sistema como candidato. • Verificar candidato.
Postcondiciones	• Se muestra una lista de todas las vacantes que coincidan con la búsqueda del candidato.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 15: Descripción del caso de uso Consultar vacantes.

Descripción	El candidato podrá modificar la información de su perfil, ya sea su información personal como curricular.
Flujo básico	El candidato modifica los datos que aparecen en el formulario, los cuales corresponden a su perfil.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.

Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato. • Ingresar al sistema como candidato. • Verificar candidato.
Postcondiciones	• Validar datos del candidato.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 16: Descripción del caso de uso Modificar perfil del candidato.

Descripción	El administrador podrá eliminar la inscripción del candidato al sistema. Todos los datos de su perfil serán eliminados y para poder acceder nuevamente al sistema el candidato tendrá que registrarse nuevamente.
Flujo básico	El Administrador da clic en el botón de "Eliminar" y confirma su decisión.
Flujos alternos	• No confirma: En caso de que el administrador de clic en el botón de "No eliminar", el sistema mostrará la pantalla de la sesión del administrador.
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar candidato. • Validar datos del candidato. • Activar cuenta del candidato. • Ingresar al sistema como administrador. • Verificar administrador.
Postcondiciones	• Mensaje de confirmación: Mensaje de confirmación de que el perfil ha sido eliminado y finalizará la sesión del administrador.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 17: Descripción del caso de uso Eliminar perfil del candidato.

Descripción	Las empresas publican las vacantes que ofrecen a la comunidad de la Facultad de Estudios Superiores Cuautitlán.
Flujo básico	El representante de la empresa llena el formulario de publicación de vacantes.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa.
Postcondiciones	• Validar datos de la vacante.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 18: Descripción del caso de uso Publicar vacantes.

Descripción	El sistema validará los datos que ingresó la empresa.
Flujo básico	Una vez que el representante de la empresa haya llenado el formulario dará clic en el botón "Enviar" y el sistema validará los datos ingresados.
Flujos alternos	• Datos incorrectos: En caso de que la empresa haya ingresado datos incorrectos en el sistema, este lo regresará al formulario de registro de vacantes para que lo modifique de tal manera que los datos sean correctos.
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa. • Publicar vacantes.
Postcondiciones	• Mensaje de confirmación: Se muestra un mensaje que confirma el registro de la vacante al sistema.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 19: Descripción del caso de uso Validar datos de la vacante.

Descripción	Las empresas podrán modificar las vacantes que publicaron anteriormente.
Flujo básico	El representante de la empresa selecciona la vacante a modificar y llena el formulario.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	- Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa. • Publicar vacantes. • Validar datos de la vacante.
Postcondiciones	• Validar datos de la vacante.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 20: Descripción del caso de uso Modificar vacante.

Descripción	Las empresas podrán eliminar las vacantes que publicaron anteriormente.
Flujo básico	El representante de la empresa selecciona la vacante a eliminar y presiona el botón "Eliminar vacante" y confirma su decisión.
Flujos alternos	- No confirma: En caso de que el representante de la empresa de clic en el botón de "Cancelar", el sistema mostrará la pantalla de la sesión de la empresa.
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa. • Publicar vacantes. • Validar datos de la vacante.
Postcondiciones	• Mensaje de confirmación: Mensaje de confirmación de que la vacante ha sido eliminada.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 21: Descripción del caso de uso Eliminar vacantes.

Descripción	Las empresas podrán consultar el perfil de los candidatos que se inscribieron al sistema.
Flujo básico	El representante de la empresa teclea el nombre de la carrera o habilidad a buscar y da clic en "Buscar".
Flujos alternos	• Candidato no encontrado: En caso de que no existan vacantes con los criterios de búsqueda, se mostrará un mensaje de que no existen candidatos que cubran ese perfil.

Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa.
Postcondiciones	• Se muestra una lista de todos los candidatos que coincidan con el perfil buscado.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 22: Descripción del caso de uso Consultar candidatos.

Descripción	Las empresas podrán modificar la información de su perfil.
Flujo básico	El representante de la empresa modifica los datos que aparecen en el formulario, los cuales corresponden a su perfil.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como empresa. • Verificar empresa.
Postcondiciones	• Validar datos de la empresa.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 23: Descripción del caso de uso Modificar perfil de la empresa.

Descripción	El administrador podrá eliminar la inscripción de la empresa al sistema. Todos los datos de su perfil y de las vacantes publicadas serán eliminados y para poder acceder nuevamente al sistema las empresas tendrán que registrarse nuevamente.
Flujo básico	El administrador da clic en el botón de "Eliminar perfil" y confirma su decisión.
Flujos alternos	• No confirma: En caso de que el administrador de click en el botón de "No eliminar", el sistema mostrará la pantalla de sesión de la empresa.
Requerimientos especiales	*No existen requerimientos especiales.*

Precondiciones	• Registrar empresa. • Validar datos de la empresa. • Activar cuenta de la empresa. • Ingresar al sistema como administrador. • Verificar administrador.
Postcondiciones	• Mensaje de confirmación: Mensaje de confirmación de que el perfil ha sido eliminado y finalizará la sesión de la empresa.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 24: Descripción del caso de uso Eliminar perfil de la empresa.

Descripción	El administrador ingresará al sistema mediante su nombre de usuario y contraseña proporcionado por los desarrolladores. El administrador del sistema podrá cambiar el nombre de usuario y contraseña cuando lo desee.
Flujo básico	El administrador del sistema ingresará su nombre de usuario y contraseña.
Flujos alternos	*No existen flujos alternos.*
Requerimientos Especiales	*No existen requerimientos especiales.*
Precondiciones	*No existen precondiciones.*
Postcondiciones	• Verificar administrador.

Puntos de Extensión	*No existen puntos de extensión.*

Diagrama 25: Descripción del caso de uso Ingresar al sistema como administrador.

Descripción	El sistema validará el nombre de usuario y contraseña que ingresó el administrador del sistema.
Flujo básico	Una vez que el administrador del sistema haya ingresado su nombre de usuario y contraseña, dará clic en el botón "Entrar", el sistema verificará si la información es correcta.
Flujos alternos	• Datos incorrectos: En caso de que el nombre de usuario y/o contraseña del administrador del sistema sean incorrectos, el sistema regresará al administrador del sistema a la pantalla de inicio de sesión.
Requerimientos Especiales	*No existen requerimientos especiales.*
Precondiciones	• Ingresar al sistema como administrador.
Postcondiciones	• Se muestra la pantalla de la sesión del administrador.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 26: Descripción del caso de uso Verificar administrador.

Descripción	El administrador del sistema podrá salir de su sesión en el momento que desee.

Flujo básico	El administrador del sistema da clic en el botón “Cerrar sesión.”
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Ingresar al sistema como administrador. • Verificar administrador.
Postcondiciones	*No existen postcondiciones.*
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 27: Descripción del caso de uso Cerrar sesión del administrador.

Descripción	El administrador del sistema podrá modificar la información de su perfil.
Flujo básico	El administrador del sistema modifica los datos que aparecen en el formulario, los cuales corresponden a su perfil.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	• Tipo de datos: Llenar los campos del formulario con el tipo de dato correcto.
Precondiciones	• Ingresar al sistema como administrador. • Verificar administrador.
Postcondiciones	• Validar datos del administrador.

Puntos de extensión	*No existen puntos de extensión.*

Diagrama 28: Descripción del caso de uso Modificar perfil del administrador.

Descripción	El sistema validará los datos que ingresó el administrador del sistema.
Flujo básico	Una vez que el administrador del sistema haya llenado el formulario dará clic en el botón "enviar" y el sistema validará los datos ingresados.
Flujos alternos	- Datos incorrectos: En caso de que el administrador del sistema haya ingresado datos incorrectos en el sistema, este lo regresará al formulario de modificar perfil del administrador para que lo modifique de tal manera que los datos sean correctos.
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Ingresar al sistema como administrador • Verificar administrador. • Modificar perfil del administrador.
Postcondiciones	• Mensaje de confirmación: Se muestra un mensaje que confirma la modificación del perfil del administrador.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 29: Descripción del caso de uso Validar datos del administrador.

Descripción	El administrador del sistema podrá obtener reportes del número de candidatos que ingresaron al sistema en los últimos 30 días de la fecha que desee.
Flujo básico	El administrador del sistema da clic en la opción Reportes.
Flujos alternos	*No existen flujos alternos.*
Requerimientos especiales	*No existen requerimientos especiales.*
Precondiciones	• Ingresar al sistema como administrador. • Verificar administrador.
Postcondiciones	• Reporte: Se muestra una lista con el número de candidatos que ingresaron al sistema en los últimos 30 días.
Puntos de extensión	*No existen puntos de extensión.*

Diagrama 30: Descripción del caso de uso Obtener reportes.

Diagrama de Clases

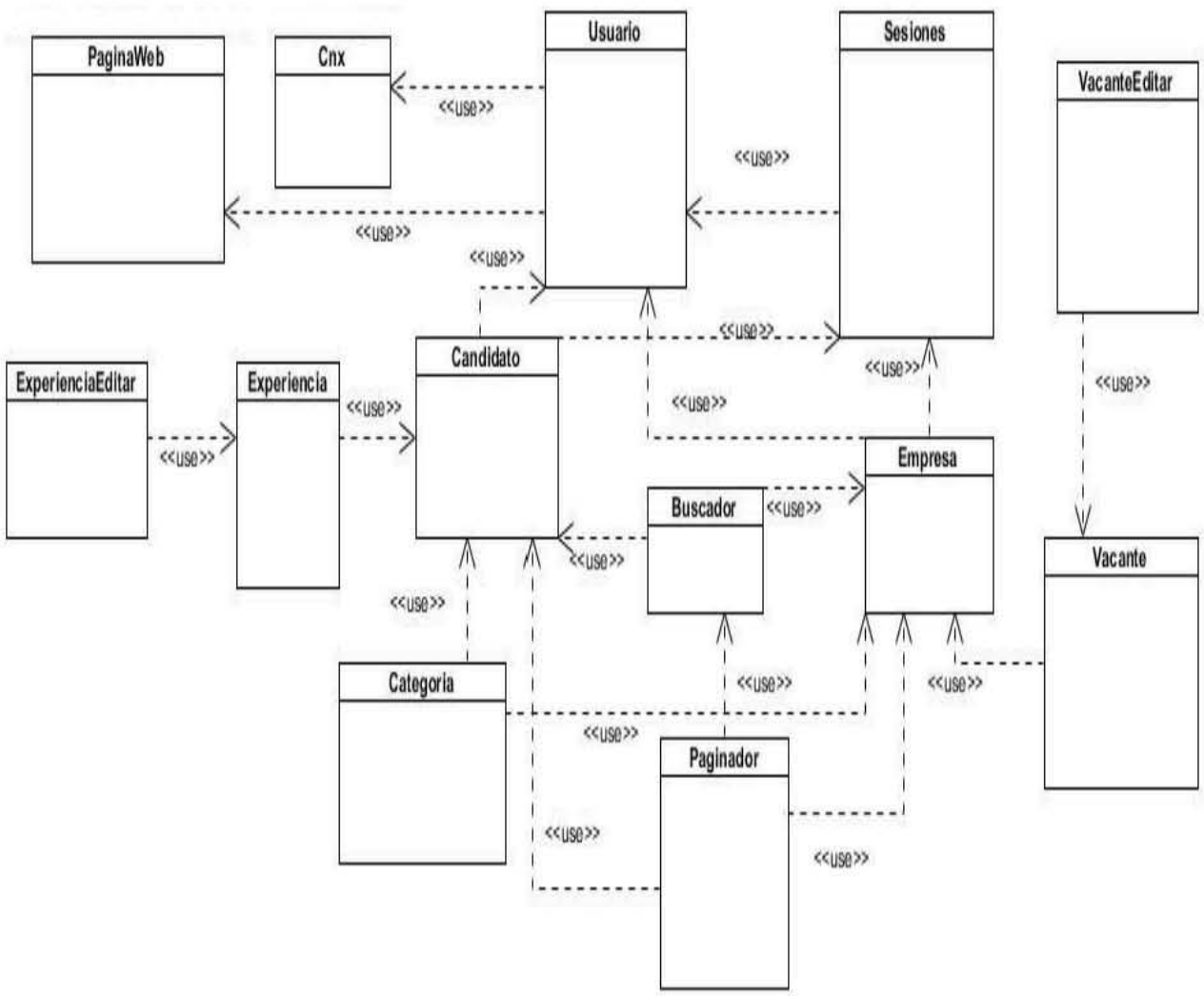


Diagrama 31: Diagrama de clases

Diagrama de Tablas

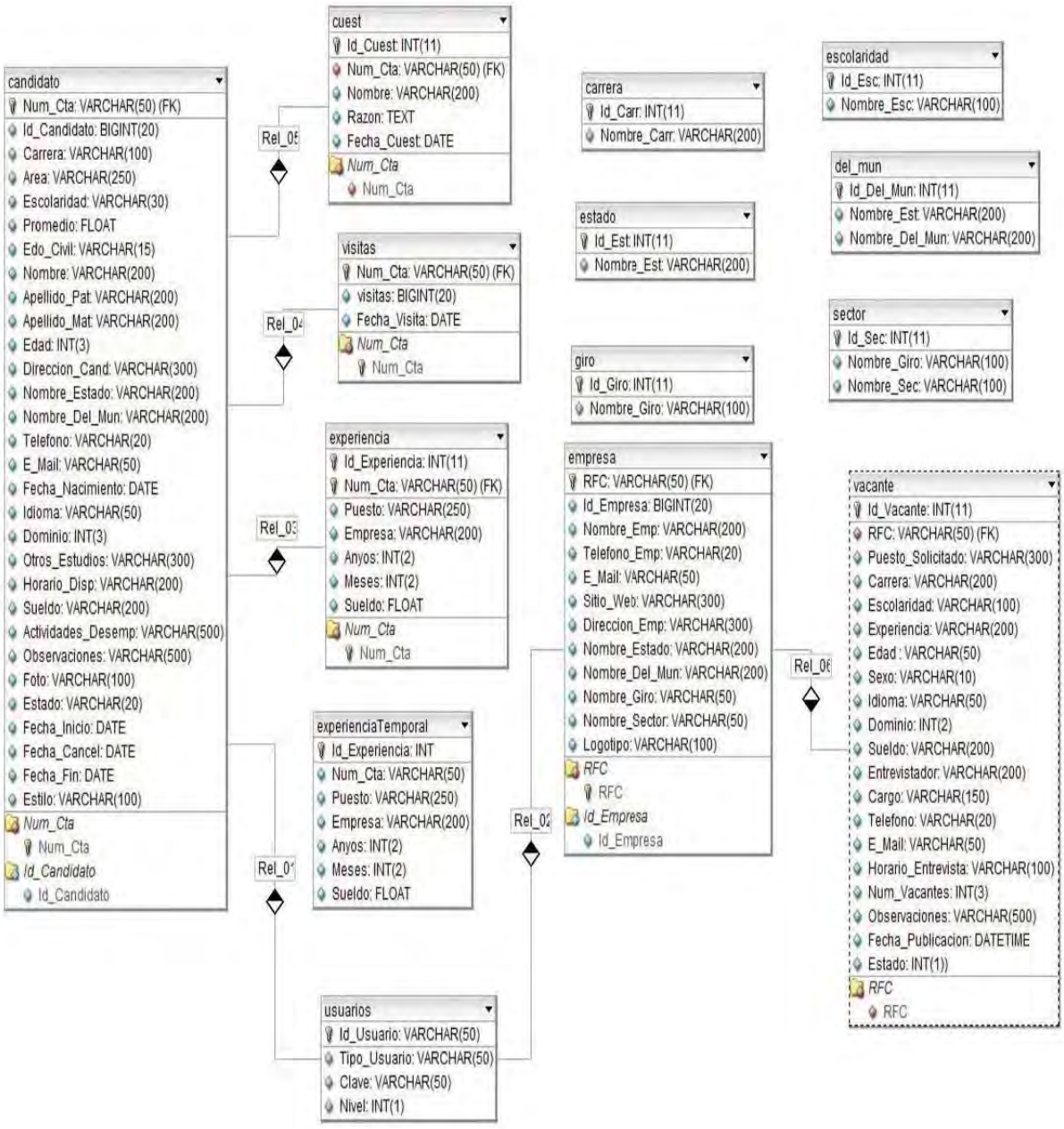


Diagrama 32: Diagrama de tablas de la base de datos

Para ver el diccionario de datos favor de revisar el Anexo A.

Diagramas de secuencia

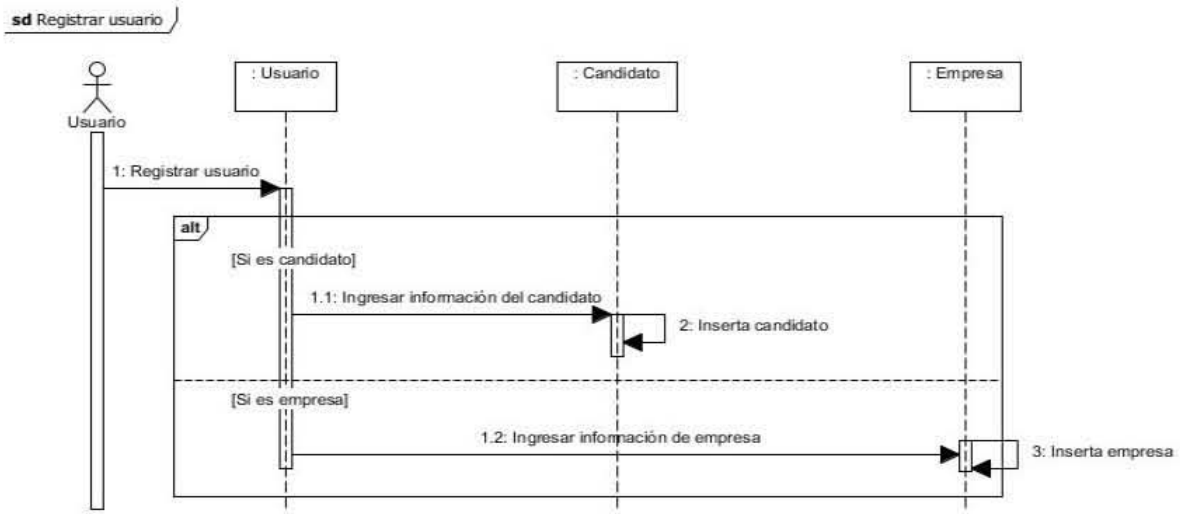


Diagrama 33: Diagrama de secuencia de Usuario.

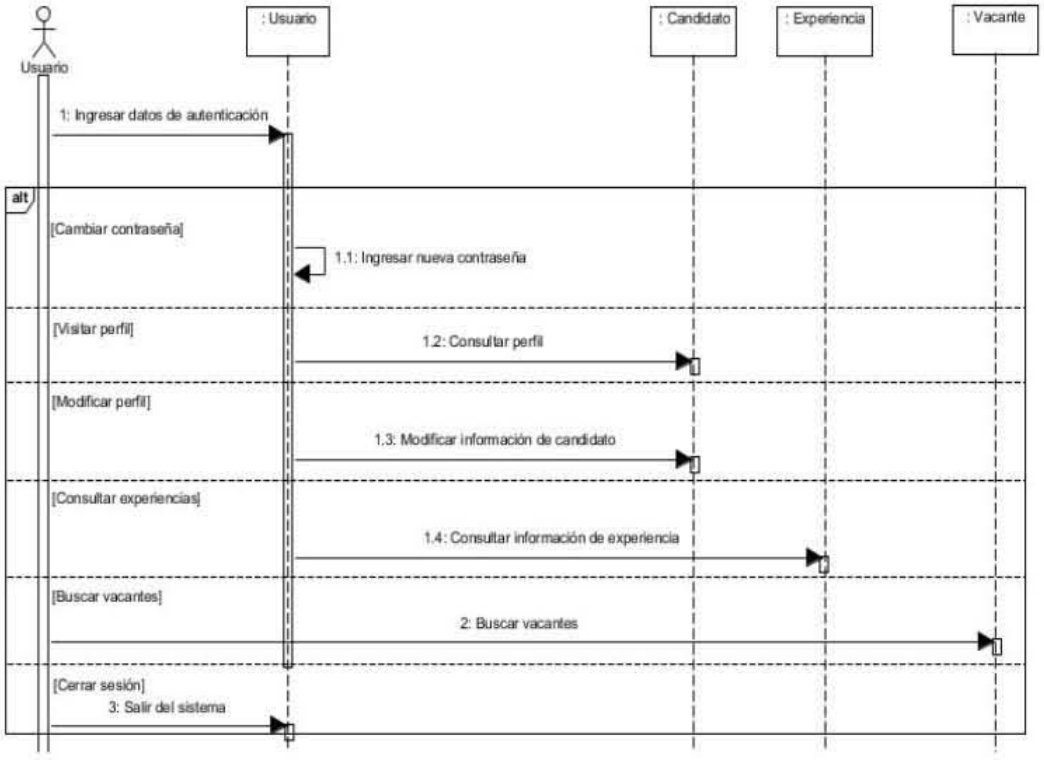


Diagrama 34: Diagrama de secuencia de Candidato.

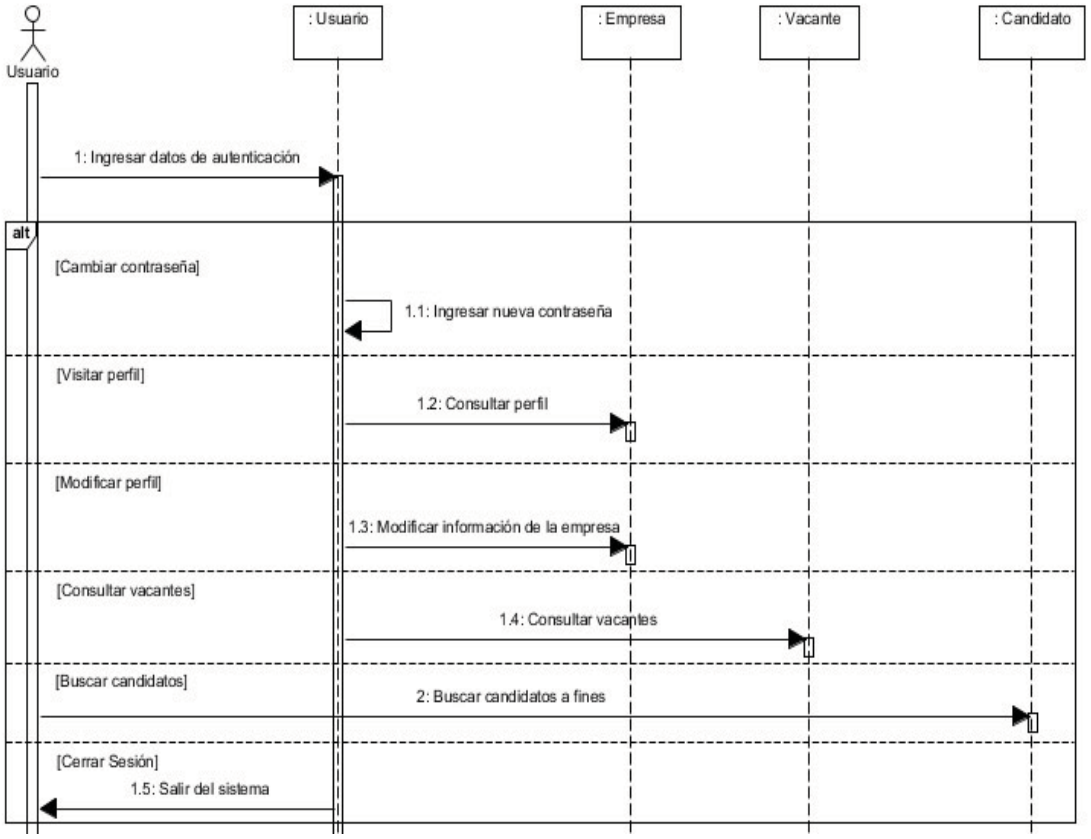


Diagrama 35: Diagrama de secuencia de Empresa.

Diagramas de estados

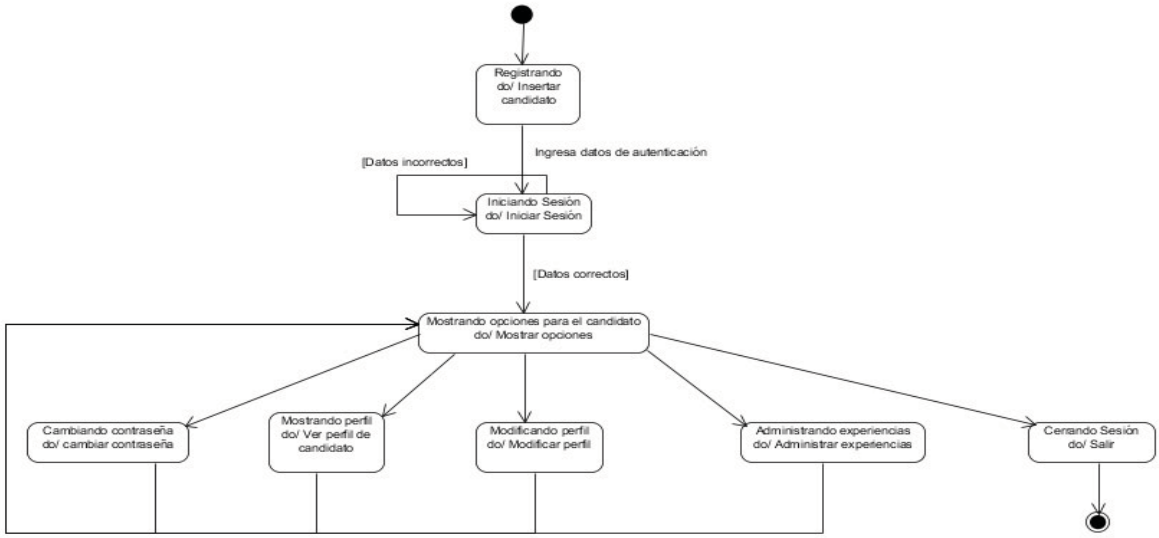


Diagrama 36: Diagrama de estados de Candidato.

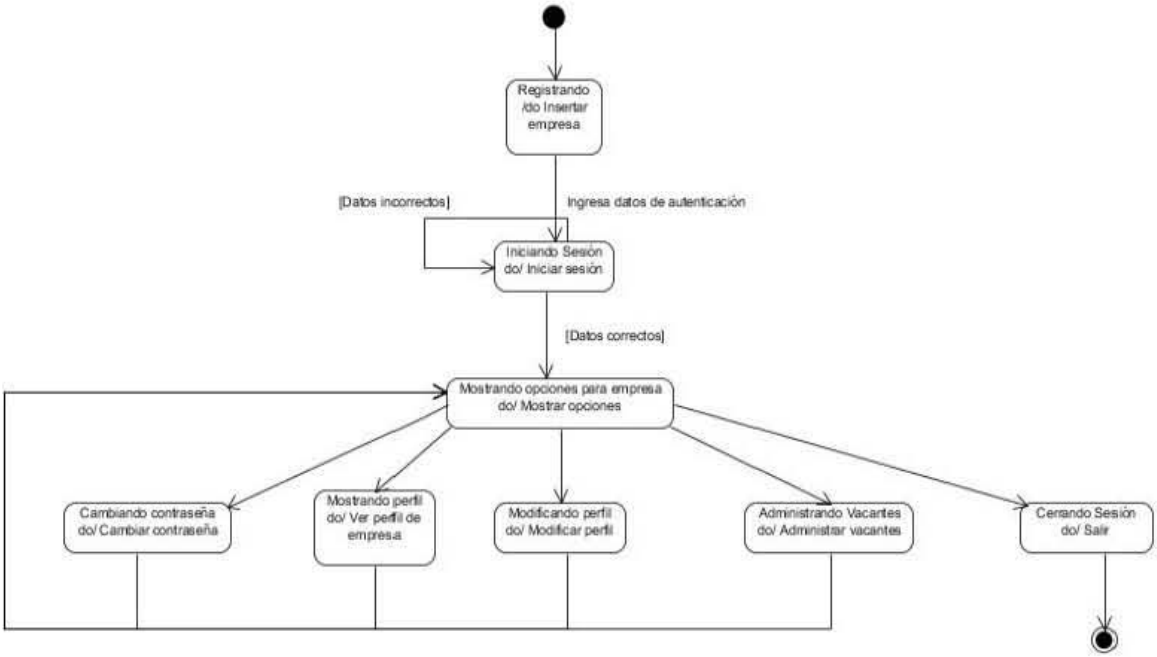


Diagrama 37: Diagrama de estados de Empresa.

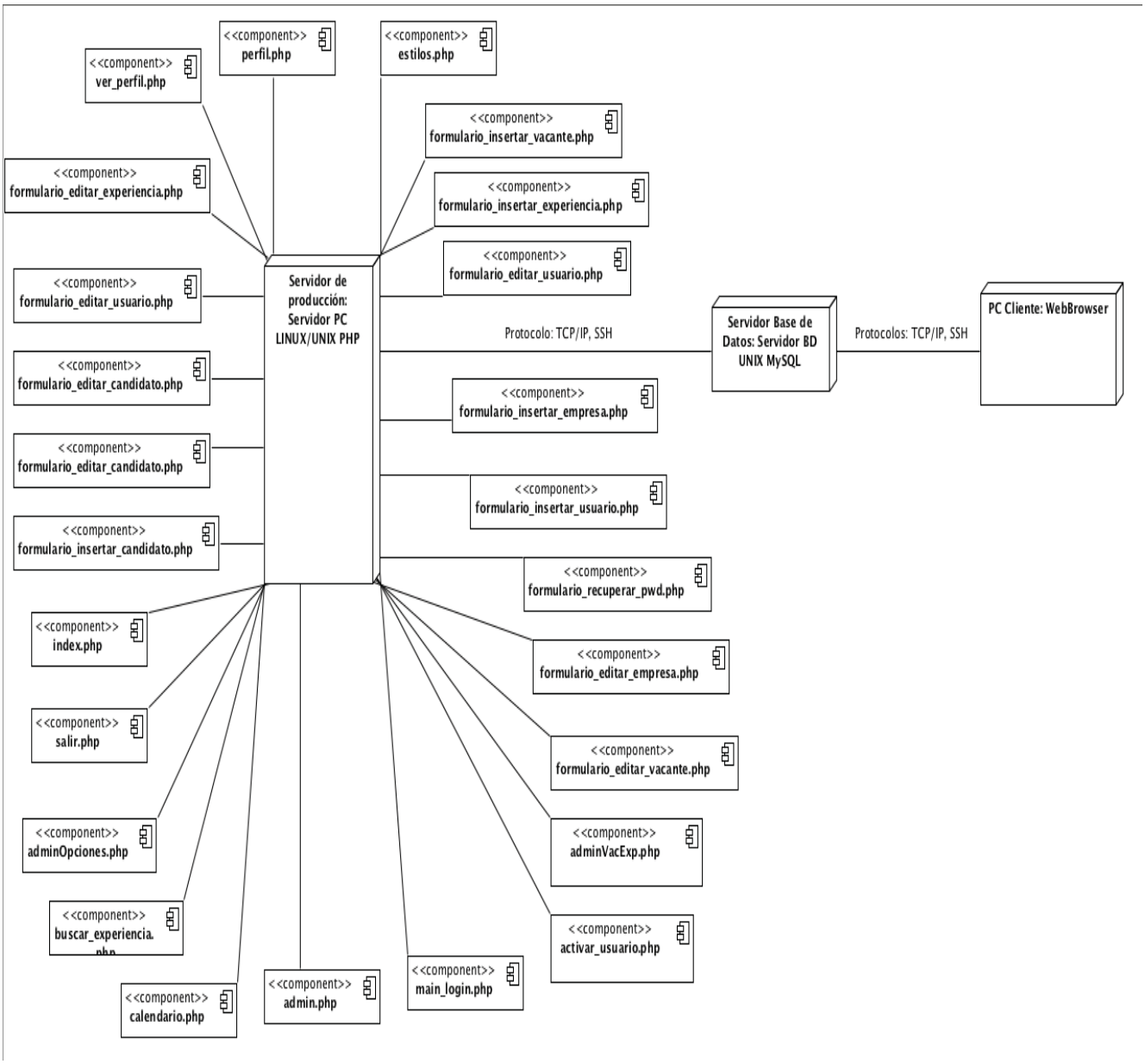


Diagrama 38: Diagrama de distribución

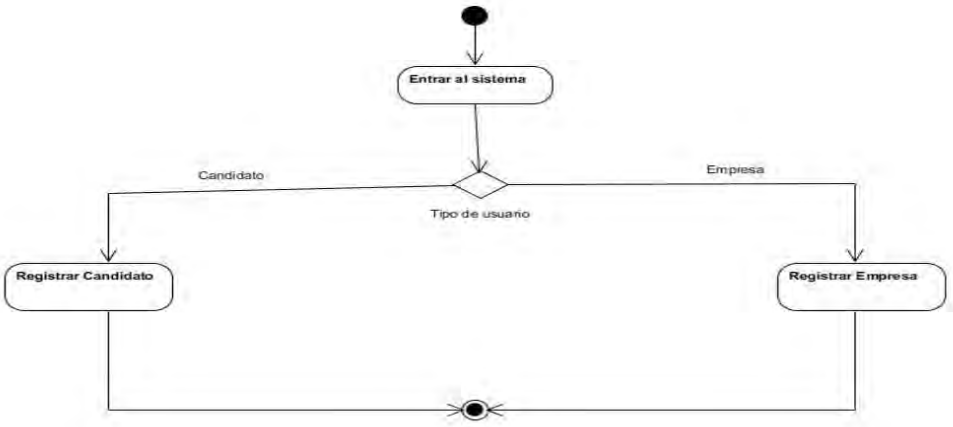


Diagrama 39: Diagrama de actividades de Registrar usuario

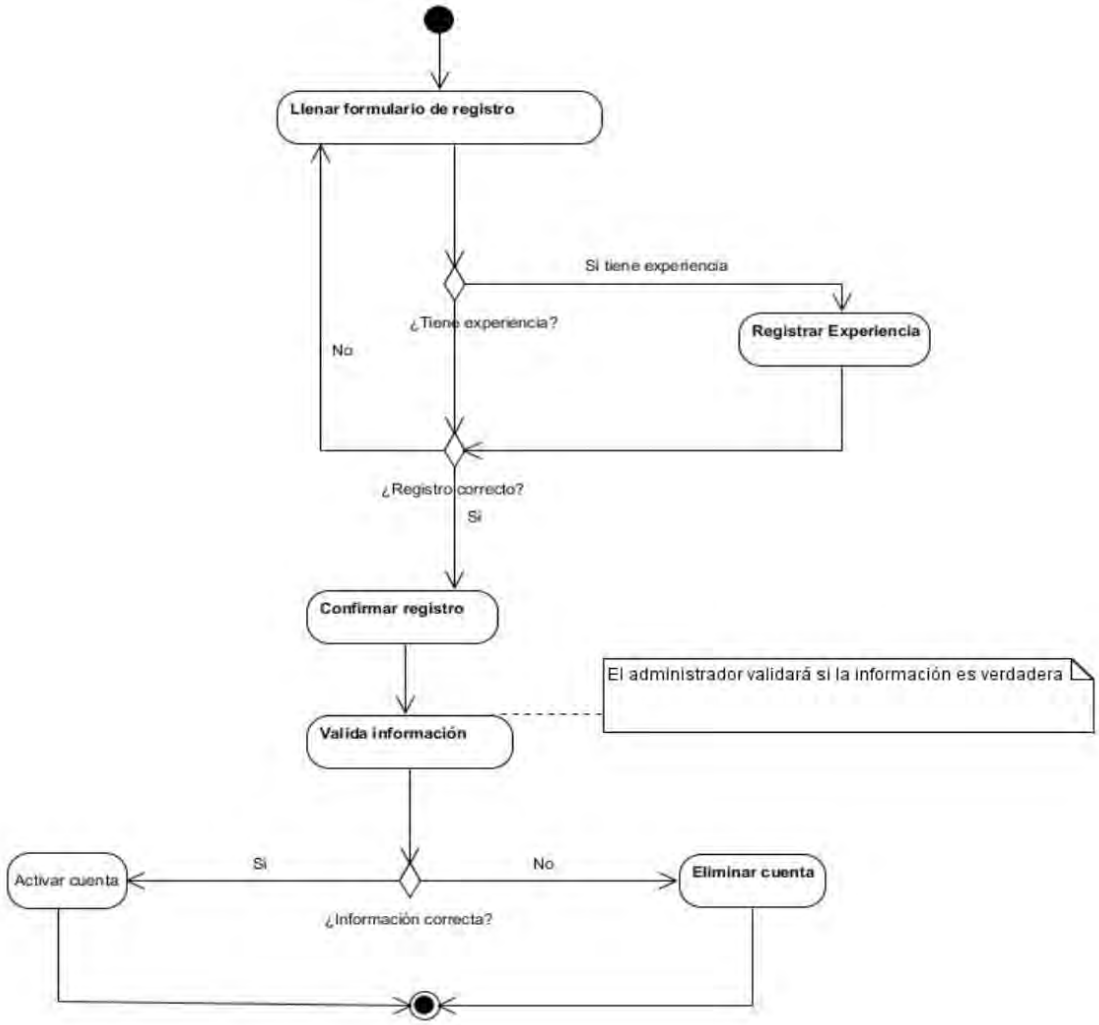


Diagrama 40: Diagrama de actividades de Registrar candidato.

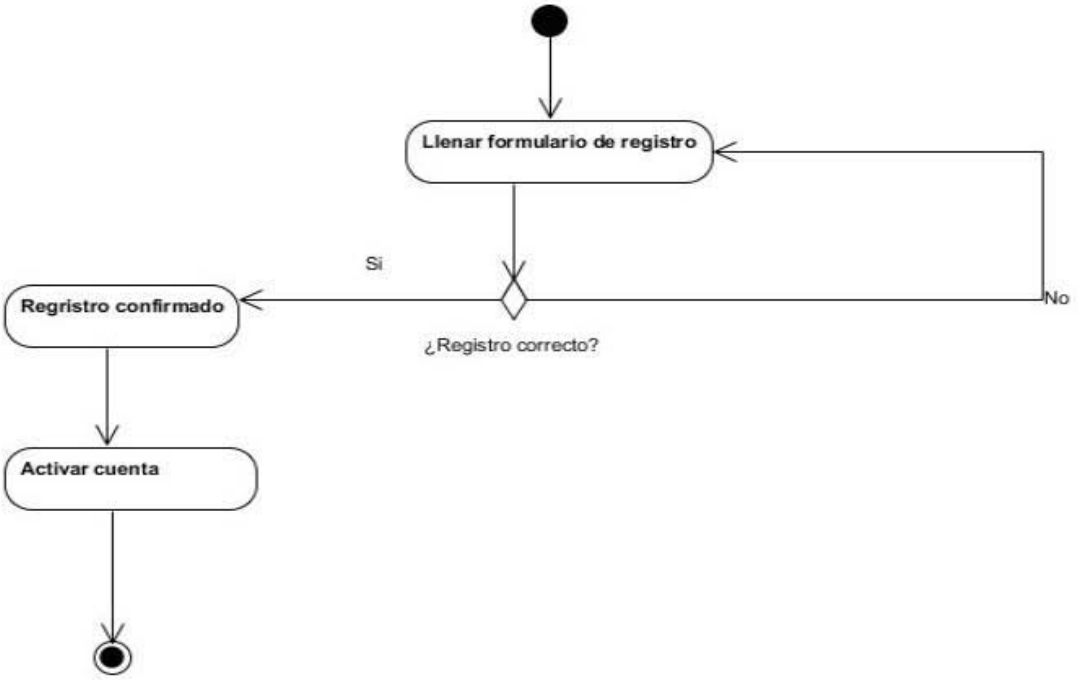


Diagrama 41: Diagrama de actividades de Registrar empresa

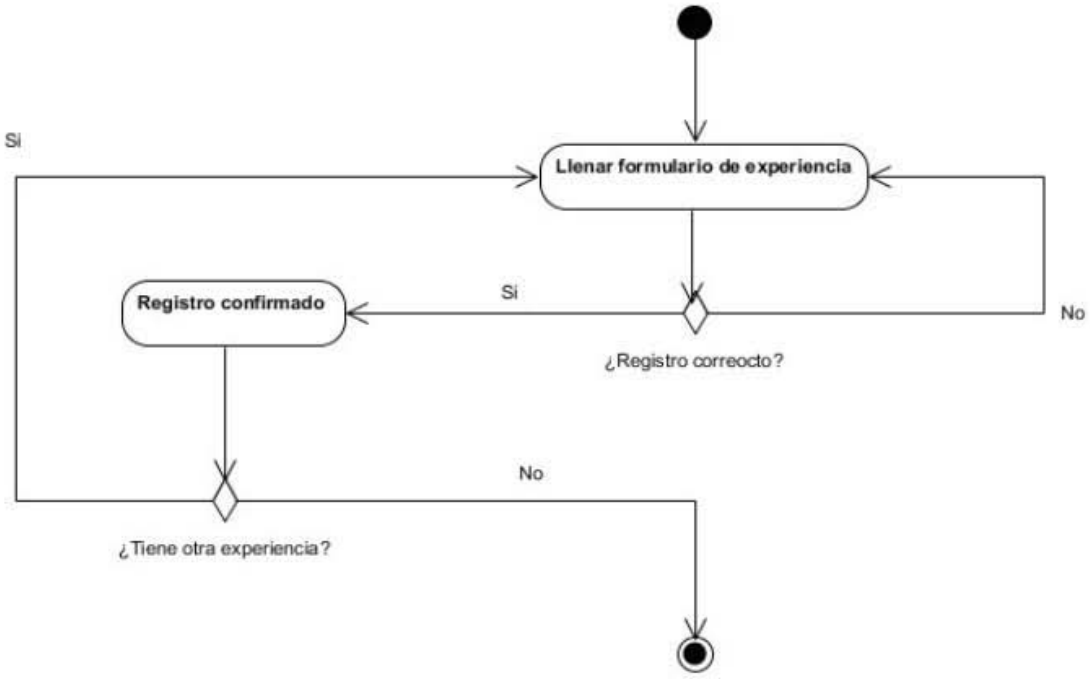


Diagrama 42: Diagrama de actividades de Registrar experiencia

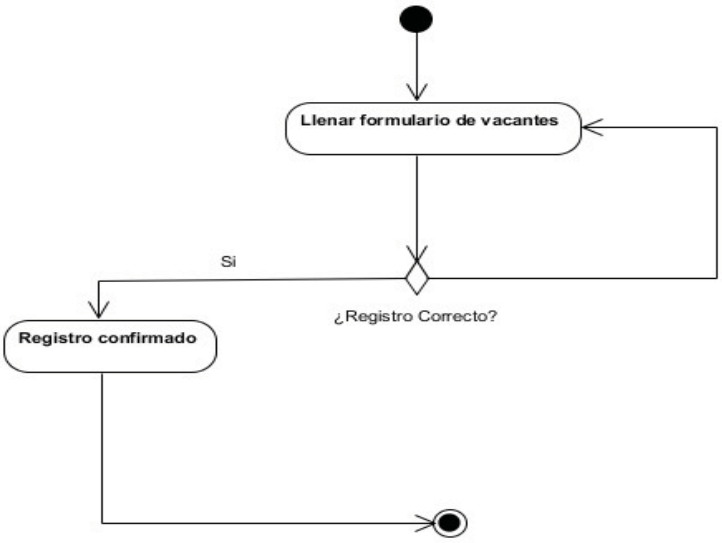


Diagrama 43: Diagrama de actividades de Registrar vacante.

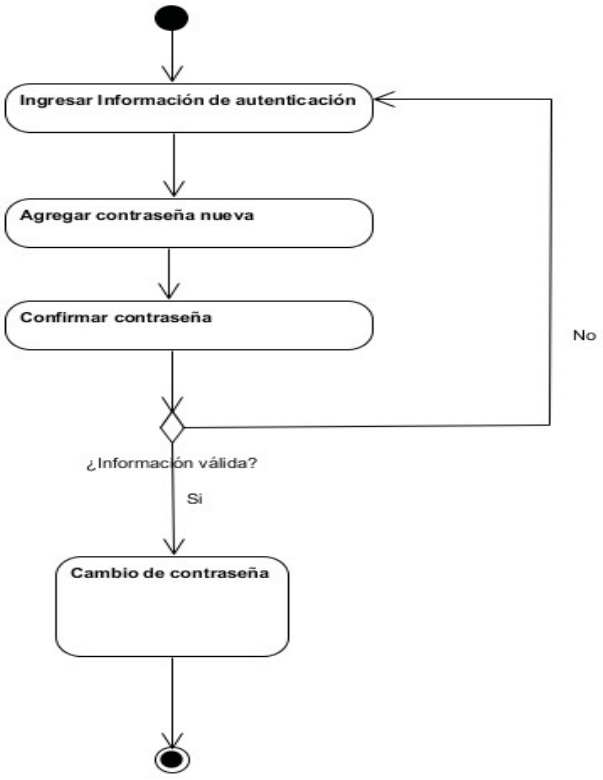


Diagrama 44: Diagrama de actividades de Cambiar contraseña.

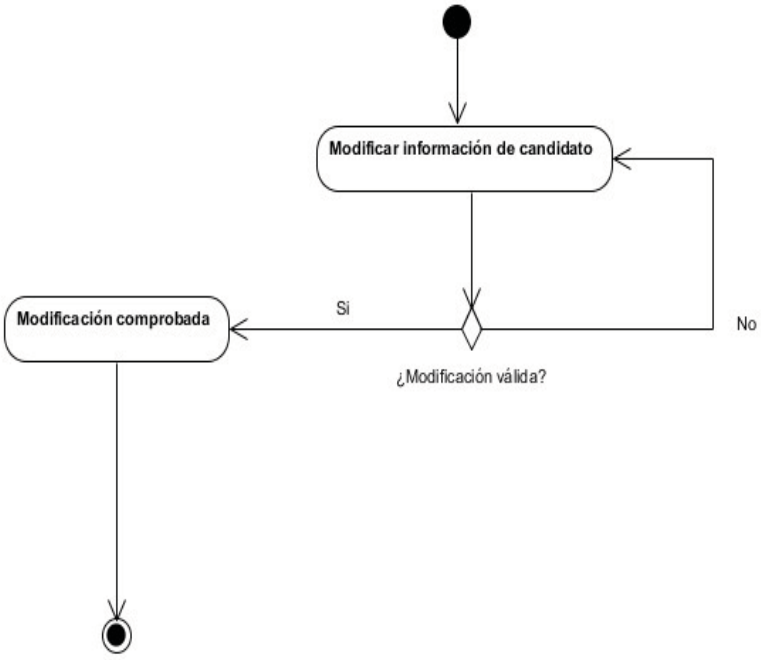


Diagrama 45: Diagrama de actividades de Modificar candidato.

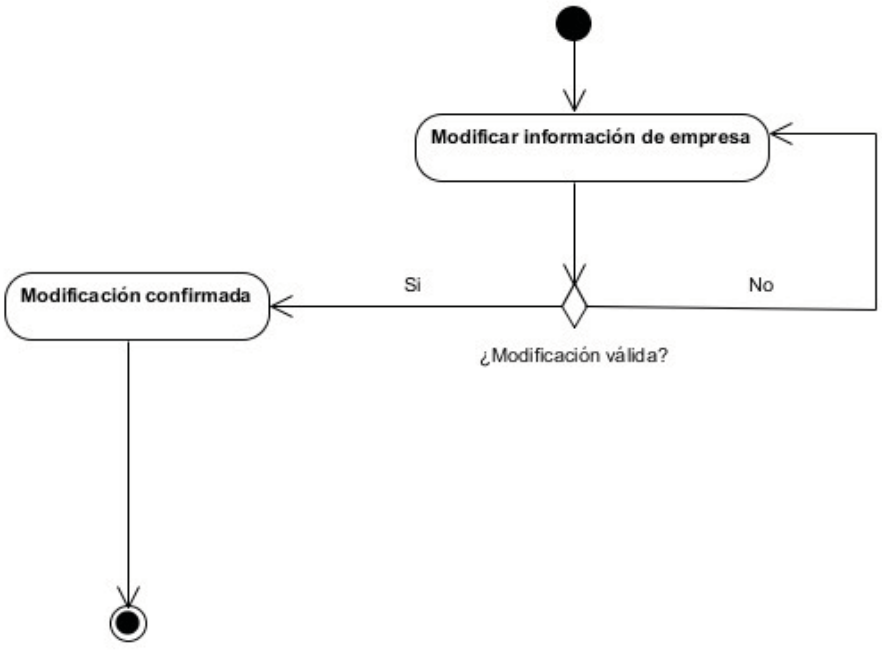


Diagrama 46: Diagrama de actividades de Modificar empresa.

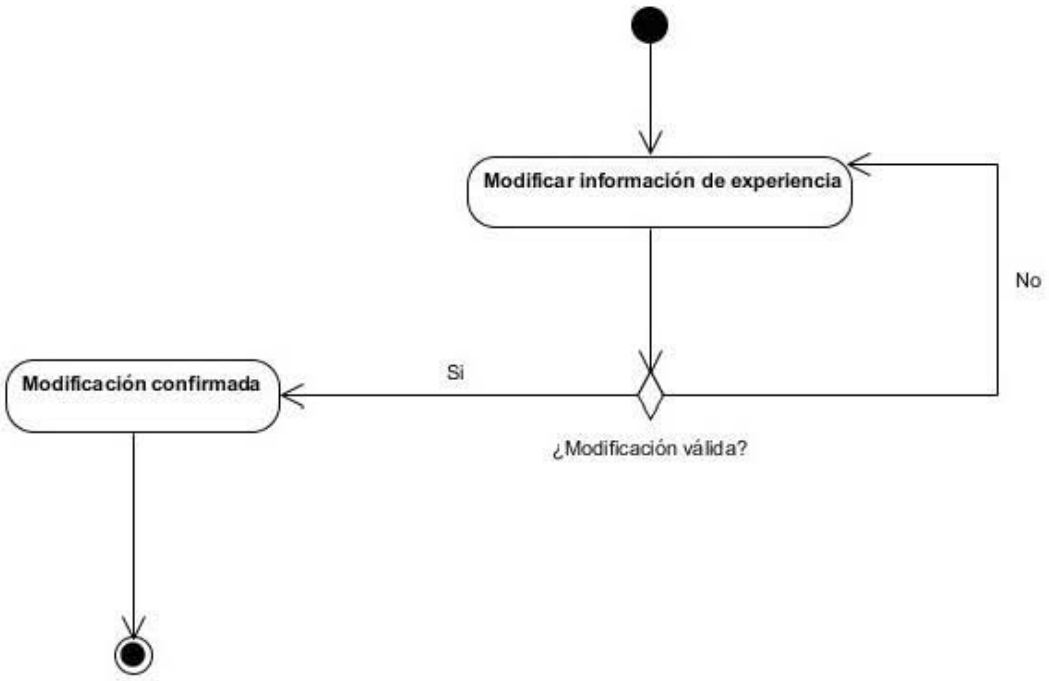


Diagrama 47: Diagrama de actividades de Modificar experiencia.

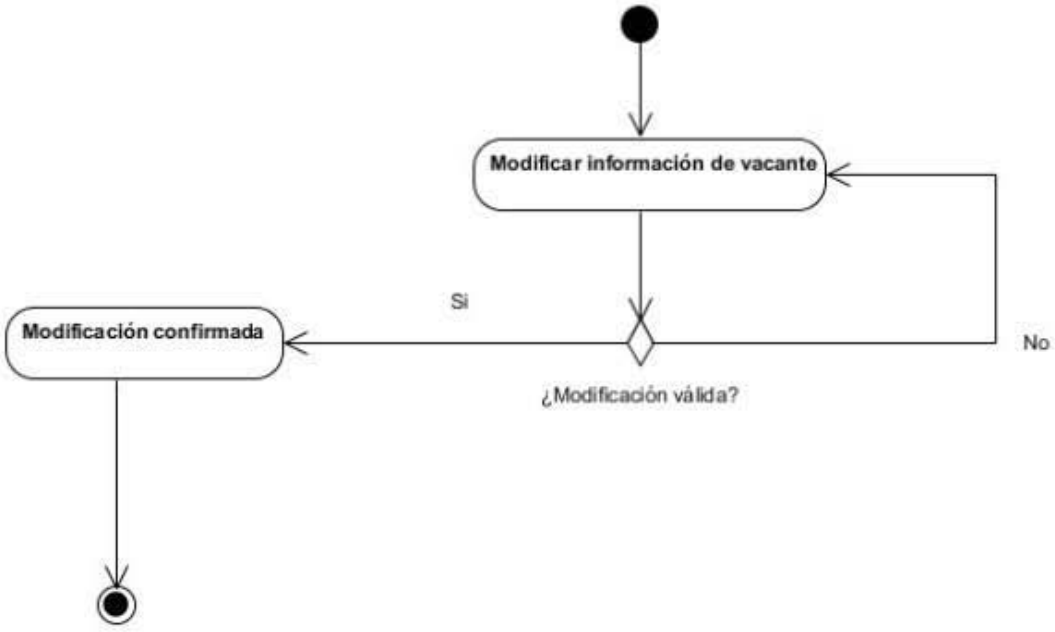


Diagrama 48: Diagrama de actividades de Modificar vacante.

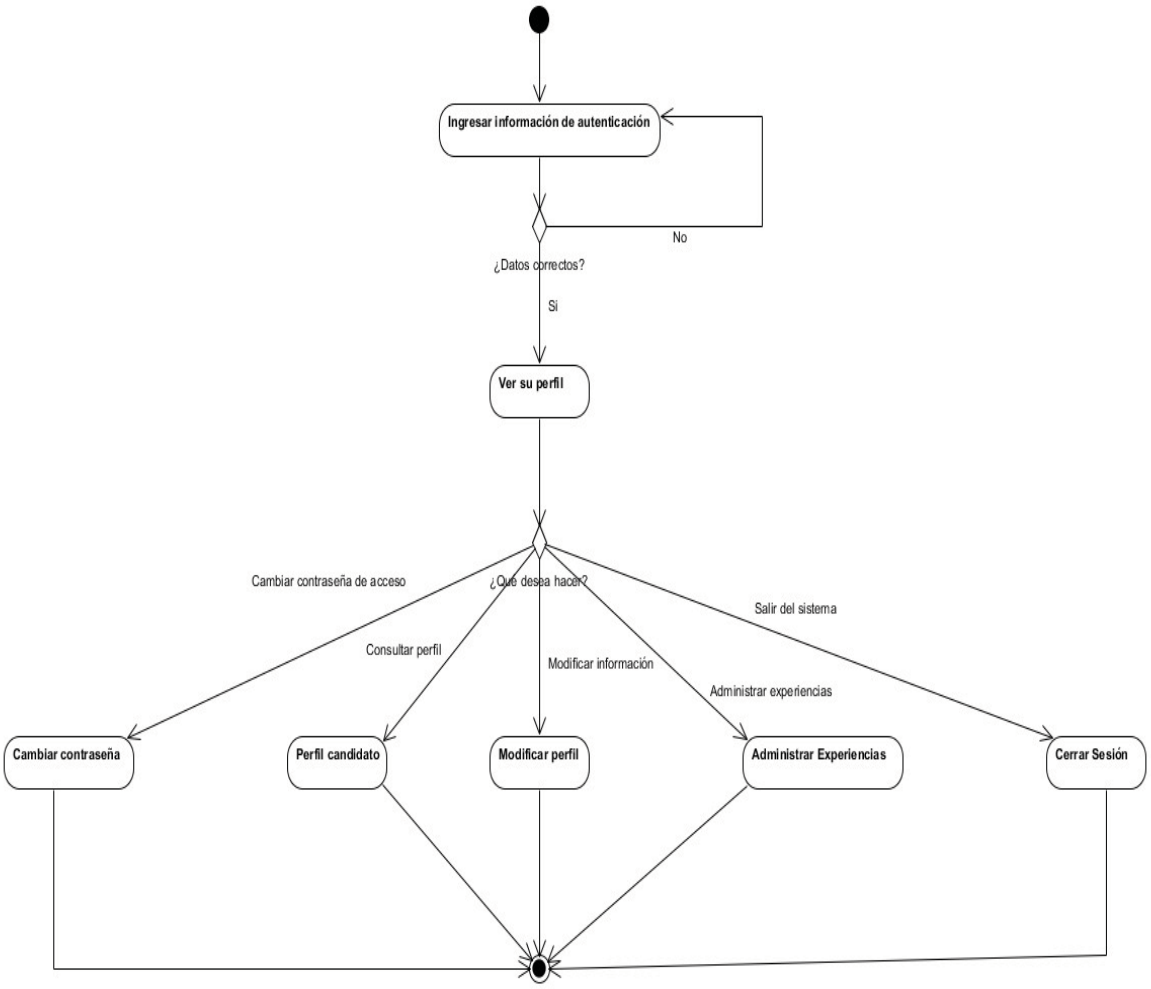


Diagrama 49: Diagrama de actividades de Perfil candidato.

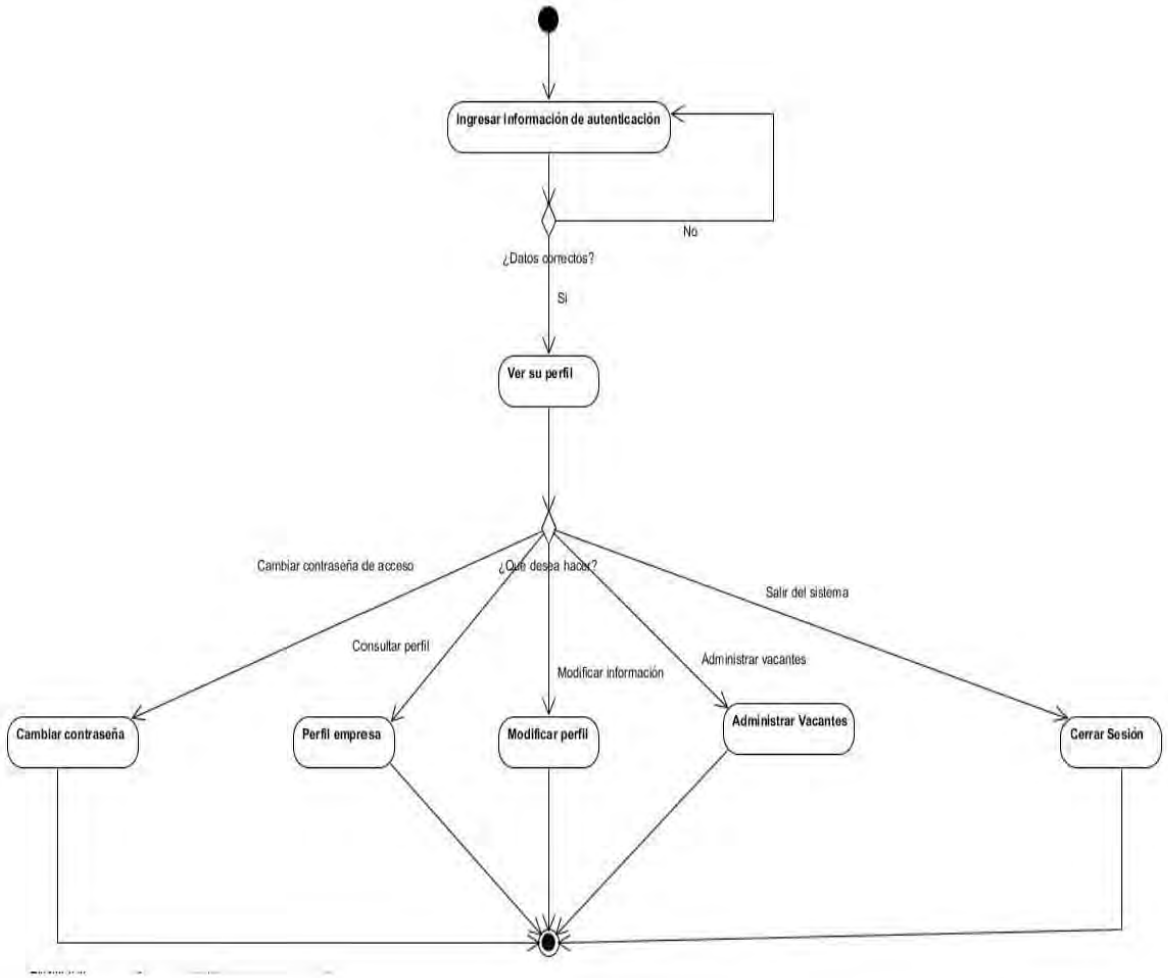


Diagrama 50: Diagrama de actividades de Perfil empresa.

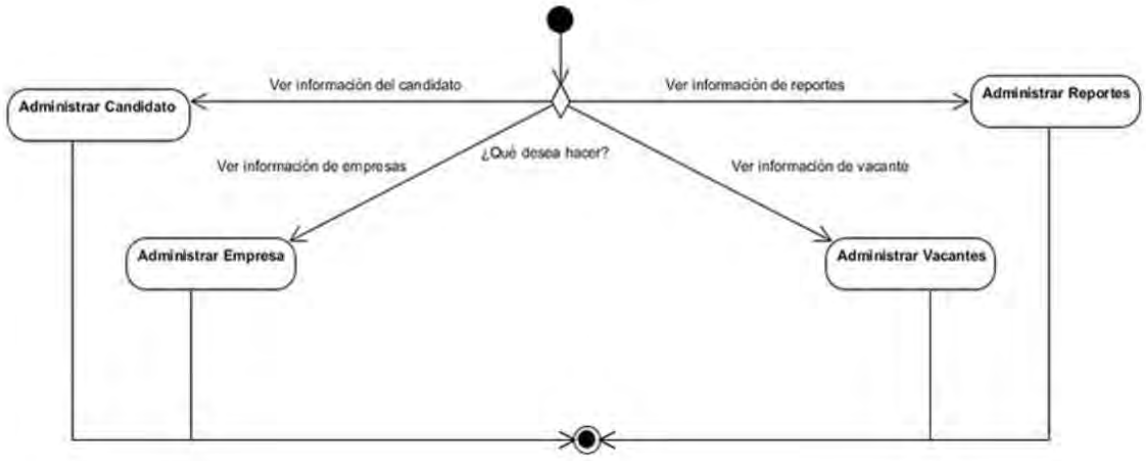


Diagrama 51: Diagrama de actividades de Administrador.

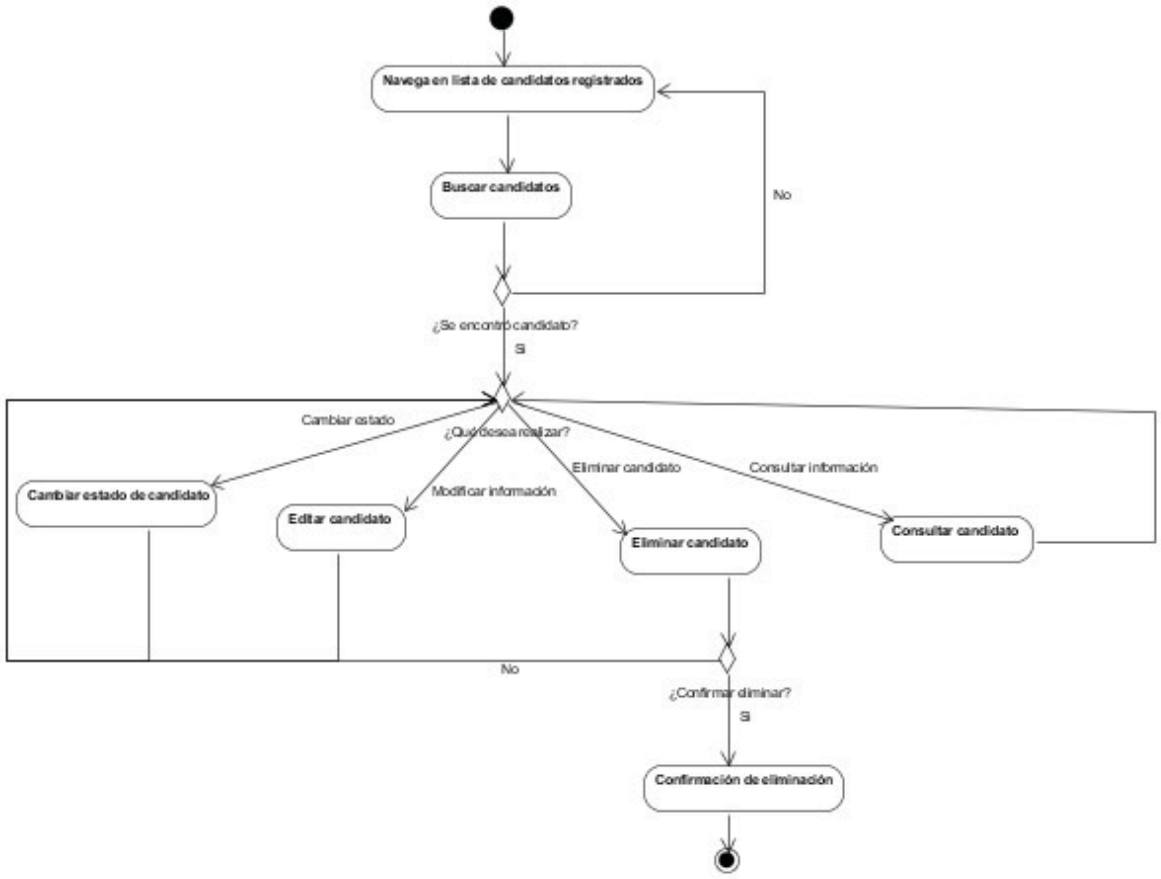


Diagrama 52: Diagrama de actividades de Administrador candidato.

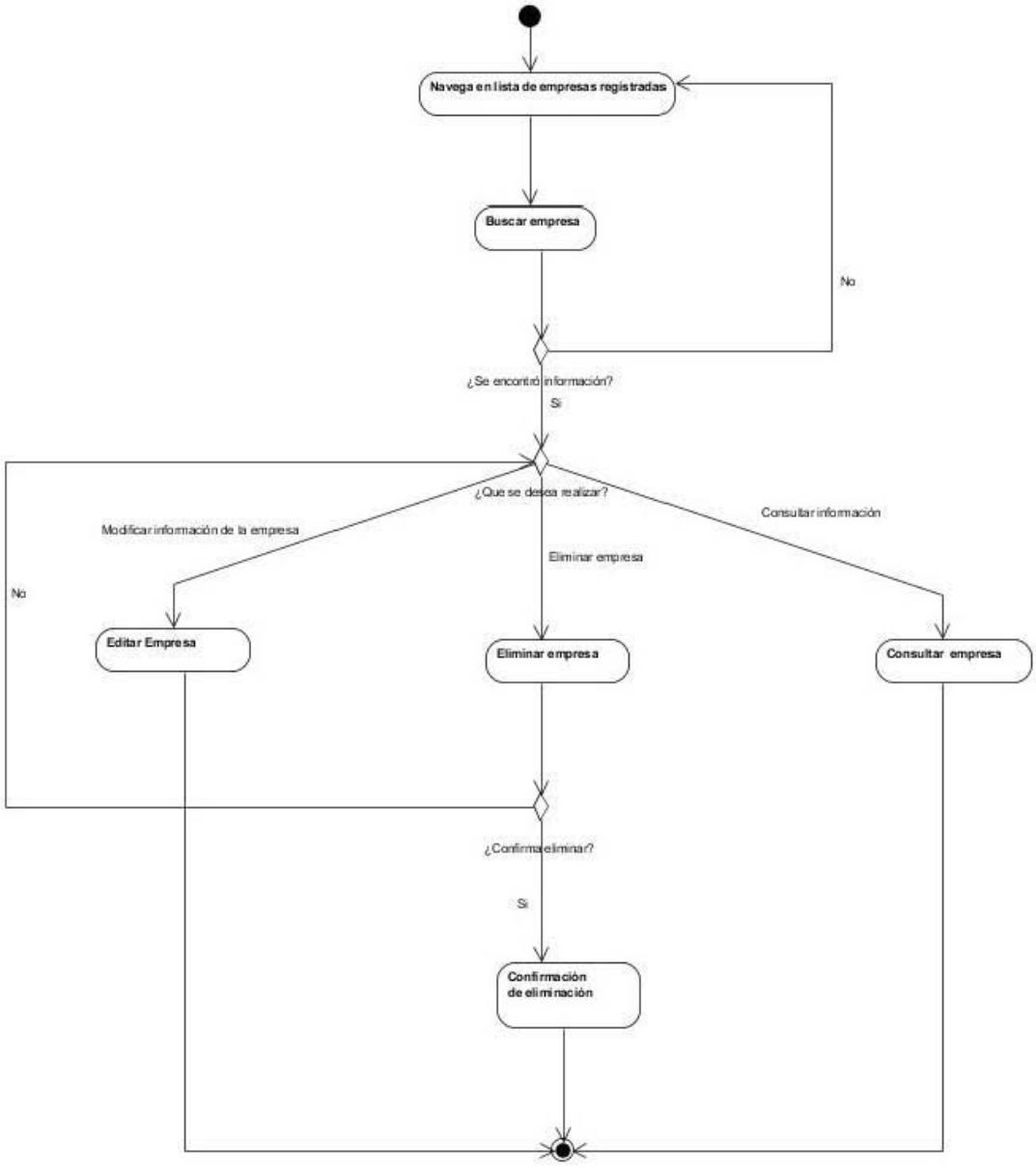


Diagrama 53: Diagrama de actividades de Administrador empresa.

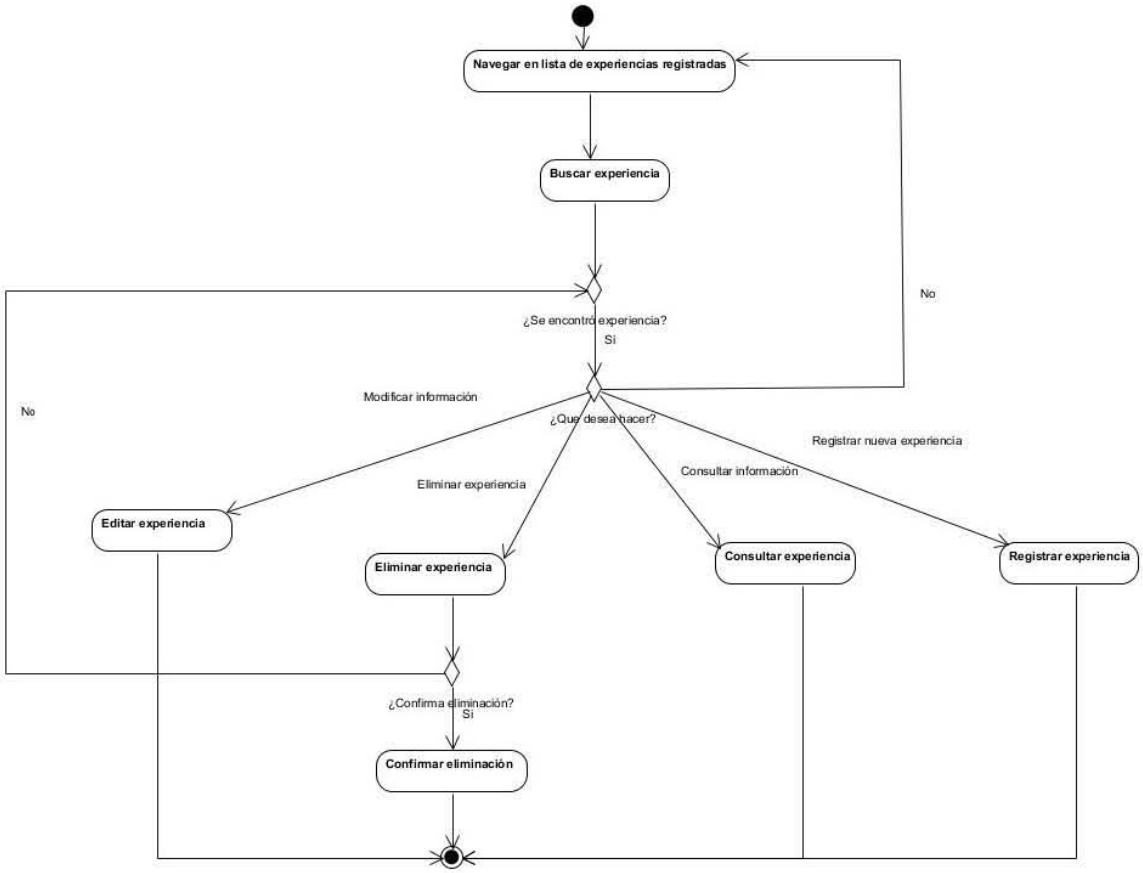


Diagrama 54: Diagrama de actividades de Administrador experiencia.

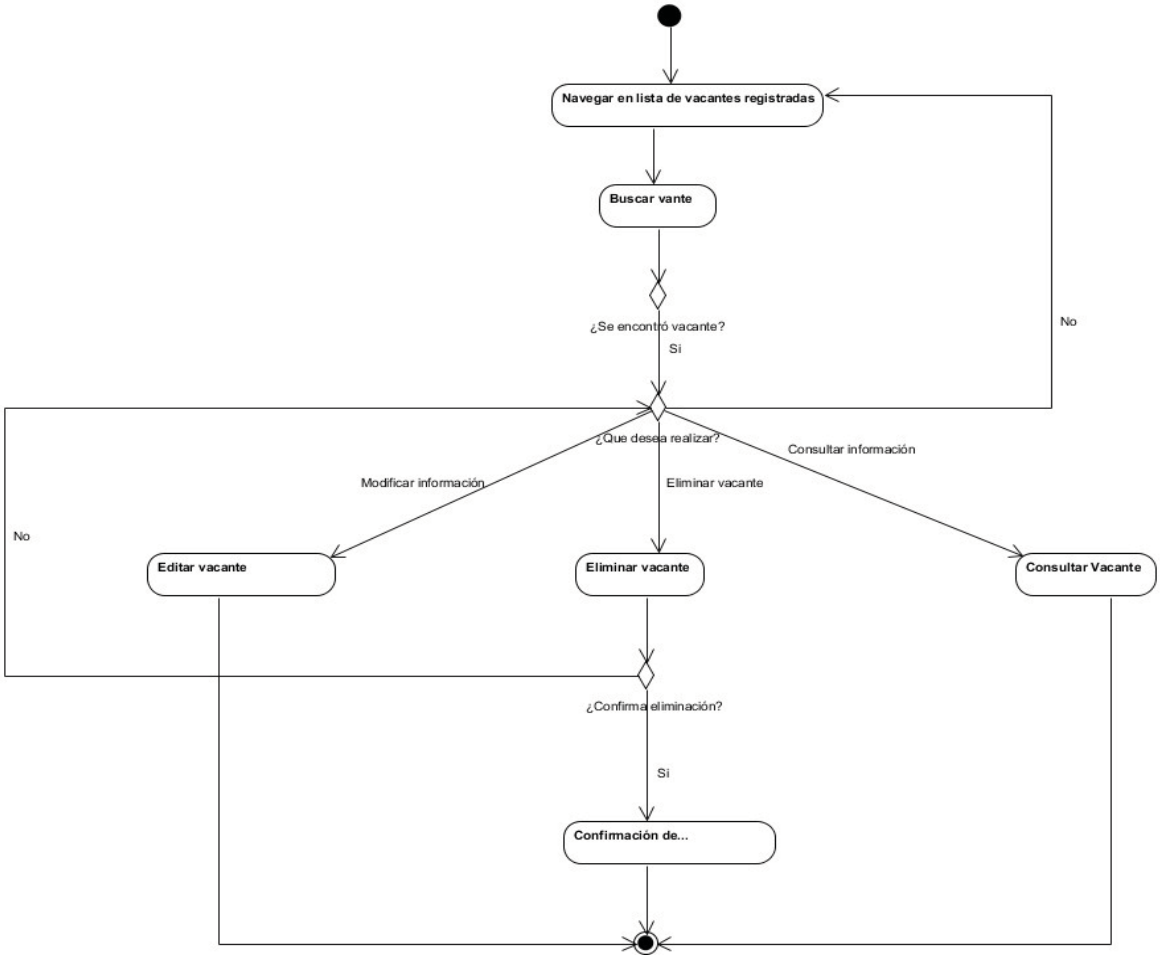


Diagrama 55: Diagrama de actividades de Administrar vacante.

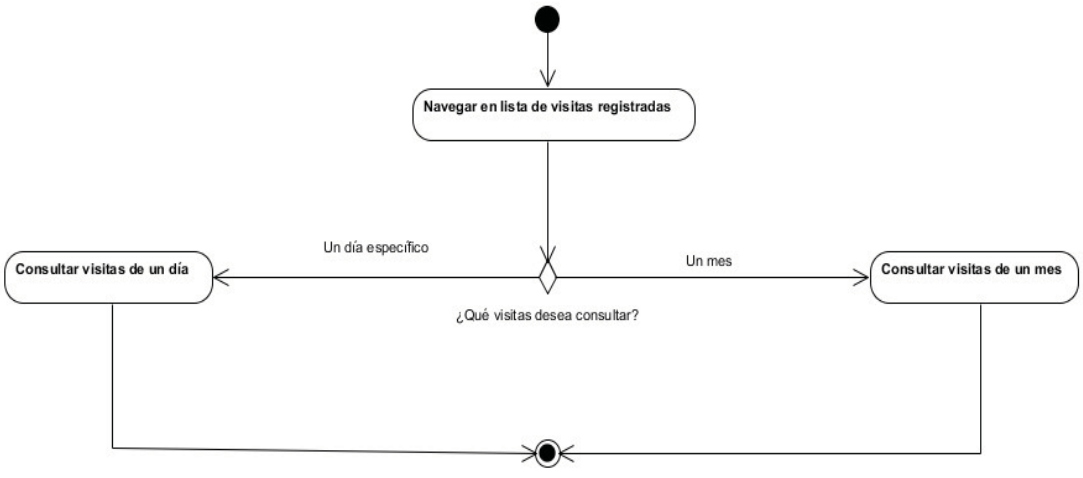


Diagrama 56: Diagrama de actividades de Administrador consultar visitas.

5. ANÁLISIS DE RESULTADOS

El desarrollo del sistema de bolsa de trabajo se realizó en un principio con programación estructurada y nos percatamos que su escalabilidad y mantenimiento sería complejo y muy limitado, en contraste con el paradigma de programación orientación a objetos que permite crear clases que ayudan a escalar y ordenar la programación del sistema. En base a esto fue necesario re diseñarlo con un paradigma de POO.

Una vez desarrollado el sistema se instaló en un ambiente de pruebas donde se registró el administrador en la base de datos.

También se registraron algunas empresas mediante el uso de una computadora personal, ingresando los datos solicitados en los formularios, este proceso no requirió más de 10 minutos para realizarse.

Posteriormente comenzaron a registrar vacantes, este proceso se realizó con no más de 10 minutos para cada vacante.

También se registraron algunos candidatos mediante el uso de una computadora personal, ingresando los datos solicitados en los formularios.

Después de activar su cuenta los candidatos comenzaron a buscar vacantes con ayuda del buscador, lograron encontrar algunas de su interés y consultaron los datos de las empresas que las publicaron.

Al contar con candidatos registrados en el sistema, las empresas comenzaron a buscarlos con ayuda del buscador, encontraron algunos de su interés y consultaron sus datos.

Todos los procesos anteriores se realizaron desde una computadora personal sin la necesidad para las empresas de asistir a la FES Cuautitlán y para el caso de los candidatos solo se requiere que asistan a las instalaciones para activar su cuenta, una vez activa ya no es necesario que se trasladen al plantel.

6. CONCLUSIONES.

Gracias a que se reduce el número de visitas al plantel, se evita el gasto que implica el trasladarse lo que ayuda a la economía de los interesados.

Actualmente el sistema se encuentra implementado en un servidor de pruebas que permite consultas limitadas de usuarios así como un número reducido de registros en la base de datos ya que el espacio de almacenamiento es reducido.

Por estas limitantes es necesario antes de implementarlo en un servidor de producción realizar pruebas con una muestra de usuarios más grande, que sea equivalente a la esperada, así como capturar la información vigente que existe en la bolsa de trabajo para ser consultada.

La disposición del sistema en línea permite que los candidatos y las empresas puedan consultar la información tanto de vacantes como de candiadatos en cualquier momento sin estar sujetos a un horario, esto permite mejorar a corto plazo el servicio de la bolsa de trabajo de la FES Cuautitlán brindando una mayor amplitud de disponibilidad de consulta de información.

La administración de la información de las empresas y candidatos podrá realizarse de una manera más ágil a corto plazo, ya que al estar alamcenada en el sistema de la bolsa de trabajo de la FESC Cuautitlán se reducen los procesos para su consulta como realizar la búsqueda en registros de documentos físicos.

7. TRABAJO FUTURO

El sistema puede escalarse para realizar nuevas funciones que ayuden a mejorar la experiencia de usuario y ampliar las capacidades del sistema.

A corto plazo se puede modificar el diseño gráfico para mejorar la experiencia de usuario.

A Largo plazo se puede implementar una validación que evite que los candidatos vayan a las instalaciones de la FESC para activar su cuenta. Mejorar el buscador optimizando las consultas SQL para realizar búsquedas más precisas. Implementar un servicio que permita notificar vía correo electrónico a los candidatos que se ha publicado una nueva vacante y desarrollar servicios Web basados en XML o JSON para ser consultados por una aplicación móvil.

8. ANEXOS

Anexo A. Diccionario de datos.

TABLA:	**Usuarios**
Descripción:	Contiene los datos de los usuarios que ingresaran al sistema.

PK	**FK**	**Nombre**	**Tipo dato**	**NULL**	**Único**	**Descripción**
Sí	Si	Id_Usuario	varchar(50)	No	Sí	Identificador del usuario.
No	No	Tipo_Usuario	varchar(50)	No	No	Cadena que representa el tipo de usuario.
No	No	Clave	varchar(50)	No	No	Cadena que representa la contraseña del usuario.
No	No	Nivel	Int(1)	No	No	Valor que representa el nivel del usuario.

TABLA:	Candidato
Descripción:	Contiene los datos del candidato.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	Si	Num_Cta	varchar(50)	No	Sí	Número de cuenta del candidato.
No	Sí	Id_Usuario	BIGINT(20)	No	No	Identificador del usuario.
No	No	Carrera	varchar(100)	No	No	Nombre de la carrera al que pertenece el candidato.
No	No	Area	varchar(250)	No	Sí	Cadena que representa el área en que se especializa el candidato.
No	No	Escolaridad	varchar(30)	No	No	Cadena que representa el nivel educativo del candidato.
No	No	Promedio	Float	No	No	Promedio obtenido por el candidato en sus estudios.
No	No	Edo_Civil	varchar(15)	No	No	Cadena que representa el estado civil del candidato.

No	No	Nombre	varchar(200)	No	No	Cadena que representa el(los) nombre(s) del candidato.
No	No	Apellido_Pat	varchar(200)	No	No	Cadena que representa el apellido paterno del candidato.
No	No	Apellido_Mat	varchar(200)	No	No	Cadena que representa el apellido materno del candidato.
No	No	Edad	Int(3)	No	No	Valor que representa la edad del candidato.
No	No	Direccion_Cand	varchar(300)	No	No	Cadena que representa el domicilio del candidato
No	No	Nombre_Estado	varchar(200)	No	No	Cadena que representa el nombre del estado donde radica el candidato.
No	No	Nombre_Del_Mun	varchar(200)	No	No	Cadena que representa el municipio donde radica el candidato
No	No	Telefono	varchar(20)	No	No	Cadena que representa el teléfono del candidato.

No	No	E_Mail	varchar(50)	No	No	Cadena que representa el correo electrónico del candidato.
No	No	Fecha_Nacimiento	Date	No	No	Fecha que representa el día del nacimiento del candidato.
No	No	Idioma	varchar(50)	No	No	Cadena que representa el(los) idiomas que domina el candidato además del español.
No	No	Dominio	Int(3)	No	No	Cantidad en porcentaje del domino de el(los) idiomas que habla el candidato
No	No	Otros_Estudios	varchar(300)	No	No	Cadena que representa de manera breve, otros estudios del candidato
No	No	Horario_Disp	varchar(50)	No	No	Cadena que representa el horario disponible del candidato.
No	No	Sueldo	varchar(200)	No	No	Cadena que representa el sueldo mensual al que aspira el candidato.

No	No	Actividades_Desemp	varchar(500)	No	No	Cadena que representa una breve descripción de las actividades que pretende desempeñar el candidato.
No	No	Observaciones	varchar(500)	No	No	Cadena que representa una breve descripción de las mejores habilidades y objetivos del candidato.
No	No	Foto	varchar(100)	No	No	Cadena que representa la ruta donde se encuentra la fotografía del candidato.
No	No	Estado	varchar(20)	No	No	Cadena que representa el estatus del candidato
No	No	Fecha_Inicio	Date	No	No	Fecha que representa el día que el candidato de dio de alta en el sistema. (Activo o Inactivo)
No	No	Fecha_Fin	Date	No	No	Fecha que representa el día limite que tiene el candidato, para consultar el sitio.

No	No	Fecha_Cancel	Date	No	No	Fecha que representa el día limite que tiene el candidato, para que su cuenta no sea dada de baja.
No	No	Estilo	String	No	No	Representa el estilo con que se mostrará los perfiles.

TABLA:	**Empresa**
Descripción:	Contiene los datos de las empresas que ingresan al sistema.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	Si	RFC	varchar(50)	No	Sí	RFC de la empresa.
No	Sí	Id_Empresa	BIGINT(20)	No	Si	Identificador de la empresa.
No	No	Nombre_Emp	varchar(200)	No	No	Cadena que representa el nombre o razón social de la empresa.
No	No	Telefono	varchar(20)	No	No	Cadena que representa el teléfono de la empresa.
No	No	E_Mail	varchar(50)	No	No	Cadena que representa el correo electrónico de la empresa.

No	No	Sitio_Web	varchar(300)	Si	No	Cadena que representa el sitio Web de la empresa.
No	No	Direccion_Emp	varchar(300)	No	No	Cadena que representa el domicilio de la empresa.
No	No	Nombre_Estado	varchar(200)	No	No	Cadena que representa el estado donde se ubica la empresa.
No	No	Nombre_Del_Mun	varchar(200)	No	No	Cadena que representa el municipio donde se ubica la empresa.
No	No	Giro	varchar(50)	No	No	Cadena que representa el giro de la empresa.
No	No	Sector	varchar(50)	No	No	Cadena que representa el sector al que pertenece la empresa.
No	No	Logotipo	varchar(100)	No	No	Cadena que representa la dirección url de la imagen del logotipo.

TABLA:	Experiencia
Descripción:	Contiene los datos de los lugares donde a trabajado el candidato.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	Si	Id_Experiencia	Int(11)	No	Sí	Identificador de la experiencia del candidato.
No	Sí	Num_Cta	varchar(50)	No	No	Número de cuenta del candidato.
No	No	Puesto	varchar(250)	No	No	Cadena que representa el puesto que desempeñó el candidato.
No	No	Empresa	varchar(200)	No	No	Cadena que representa la empresa donde trabajó el candidato.
No	No	Anyos	Int(2)	Si	No	Cadena que representa la cantidad de años que trabajó el candidato en la empresa.
No	No	Meses	Int(2)	Si	No	Cantidad de meses que trabajó el candidato en la empresa.
No	No	Sueldo	Float	No	No	Sueldo del candidato en la empresa.

TABLA:	Vacante
Descripción:	Contiene los datos de las vacantes publicadas por las empresas.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	Si	Id_Vacante	Int(11)	No	Sí	Identificador de la vacante publicada por la empresa.
No	Sí	RFC	varchar(50)	No	No	RFC de la empresa que publica la vacante.
No	No	Puesto_Solicitado	varchar(300)	No	No	Cadena que representa el puesto que solicita la empresa.
No	No	Carrera	varchar(200)	No	No	Cadena que representa la carrera que solicita la empresa para cubrir la vacante.
No	No	Escolaridad	varchar(100)	No	No	Cadena que representa la escolaridad que requiere la vacante.
No	No	Experiencia	varchar(200)	No	No	Cadena que representa la experiencia que necesario para cubrir la vacante.
No	No	Edad	varchar(50)	Si	No	Valor que representa la edad para cubrir la vacante.

No	No	Sexo	varchar(10)	No	No	Cadena que representa el género, ya sea masculino o femenino. Que debe cubrir la vacante.
No	No	Idioma	varchar(50)	Sí	No	Cadena que representa el(los) idioma(s) que debe dominar el candidato a la vacante.
No	No	Dominio	Int(2)	Si	No	Porcentaje del dominio del idioma
No	No	Sueldo	varchar(200)	Si	No	Cadena que representa el sueldo ofrecido por la empresa.
No	No	Entrevistador	varchar(200)	No	No	Cadena que representa el nombre del entrevistador.
No	No	Cargo	varchar(150)	No	No	Cadena que representa el cargo que desempeña el entrevistador
No	No	Telefono	varchar(20)	No	No	Cadena que representa el teléfono del entrevistador.
No	No	E_Mail	varchar(50)	Si	No	Cadena que representa el correo electrónico del entrevistador.

No	No	Horario_Entrevista	varchar(100)	Si	No	Cadena que representa el horario para la entrevista.
No	No	Num_Vacantes	Int(3)	No	No	Cantidad de vacantes disponibles para una misma actividad.
No	No	Observaciones	varchar(500)	Si	No	Cadena que representa las observaciones
No	No	Fecha_Publicacion	Date	No	No	Fecha que representa el día que se publicó la vacante.
No	No	Estado	Int(1)	No	No	Entero que representa el estado de la vacante. Ya sea activo/inactivo.

TABLA:	**Cuest**
Descripción:	Contiene los datos del cuestionario del candidato para reactivar su cuenta.

PK	**FK**	**Nombre**	**Tipo dato**	**NULL**	**Único**	**Descripción**
Sí	Si	Id_Cuest	Int(11)	No	Sí	Identificador del cuestionario.
No	Sí	Num_Cta	varchar(50)	No	No	Número de cuenta del candidato.

No	No	Nombre	varchar(200)	No	No	Cadena que representa el nombre del candidato.
No	No	Razon	TEXT	No	No	Cadena que representa la razón por la cuál el candidato no consulta las vacantes con frecuencia.
No	No	Fecha_Cuesta	Date	No	No	Fecha que representa el día en que se realizó el cuestionario.

TABLA:	**Visitas**
Descripción:	Contiene los datos de las consultas de los candidatos al sistema.

PK	**FK**	**Nombre**	**Tipo dato**	**NULL**	**Único**	**Descripción**
Sí	Sí	Num_Cta	varchar(50)	No	Si	Número de cuenta del candidato.
No	No	Visitas	Bigint(20)	No	No	Cantidad de visitas realizadas por los candidatos.
No	No	Fecha_Visita	Date	No	No	Fecha que representa el día que se hicieron las visitas.

TABLA:	Carrera
Descripción:	Contiene el catálogo de las carreras.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Carrera	Int(11)	No	Si	Identificador de la carrera.
No	No	Nombre_Carr	varchar(200)	No	No	Cadena que representa el nombre de la carrera

TABLA:	Escolaridad
Descripción:	Contiene el catálogo de los niveles de escolaridad.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Esc	Int(11)	No	Si	Identificador de la escolaridad.
No	No	Nombre_Esc	varchar(100)	No	No	Cadena que representa el nivel de escolaridad.

TABLA:	Estado
Descripción:	Contiene el catálogo de los estados de la república mexicana.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Est	Int(11)	No	Si	Identificador del estado de la república mexicana.
No	No	Nombre_Est	varchar(200)	No	No	Cadena que representa el nombre del estado de la república mexicana.

TABLA:	Sector
Descripción:	Contiene el catálogo de los sectores a los que pertenecen las empresas.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Sec	Int(11)	No	Si	Identificador del sector.
No	No	Nombre_Giro	varchar(100)	No	No	Cadena que representa el nombre del giro.
No	No	Nombre_Sec	varchar(100)	No	No	Cadena que representa el nombre del sector.

TABLA:	Giro
Descripción:	Contiene el catálogo del giro al que pertenece la empresa.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Giro	Int(11)	No	No	Identificador del giro.
No	No	Nombre_Giro	varchar(100)	No	No	Cadena que representa el giro.

TABLA:	del_mun
Descripción:	Contiene el catálogo de los municipios.

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	No	Id_Del_Mun	Int(11)	No	No	Identificador del municipio.
No	No	Nombre_Est	varchar(200)	No	No	Cadena que representa el nombre del estado de la república mexicana.
No	No	Nombre_Del_Mun	varchar(200)	No	No	Cadena que representa el nombre del municipio.

TABLA:	experienciaTemporal
Descripción:	Es una tabla transitiva que contiene datos la tabla experiencia

PK	FK	Nombre	Tipo dato	NULL	Único	Descripción
Sí	Si	Id_Experiencia	Int(11)	No	Sí	Identificador de la experiencia del candidato.
No	Sí	Num_Cta	varchar(50)	No	No	Número de cuenta del candidato.
No	No	Puesto	varchar(250)	No	No	Cadena que representa el puesto que desempeñó el candidato.
No	No	Empresa	varchar(200)	No	No	Cadena que representa el nombre de la empresa donde trabajó el candidato.
No	No	Anyos	Int(2)	No	No	Cadena que representa la cantidad de años que trabajó.
No	No	Meses	Int(2)	Si	No	Cantidad de meses que trabajó el candidato en la empresa.
No	No	Sueldo	Float	No	No	Sueldo del candidato en la empresa.

Anexo B. Script completo para la creación de la base de datos para el Sistema de Bolsa de Trabajo de la FES Cuautitlán.

```
create database bolsa;

use bolsa;

CREATE TABLE experienciaTemporal (
 Id_Experiencia INT NOT NULL AUTO_INCREMENT,
 Num_Cta VARCHAR(50) NOT NULL,
 Puesto VARCHAR(250) NOT NULL,
 Empresa VARCHAR(200) NOT NULL,
 Anyos INT(2) NULL,
 Meses INT(2) NULL,
 Sueldo FLOAT NOT NULL,
 PRIMARY KEY(Id_Experiencia)
);

CREATE TABLE estado (
 Id_Est INT(11) NOT NULL AUTO_INCREMENT,
 Nombre_Est VARCHAR(200) NOT NULL,
 PRIMARY KEY(Id_Est)
)
TYPE=InnoDB;

CREATE TABLE giro (
 Id_Giro INT(11) NOT NULL AUTO_INCREMENT,
 Nombre_Giro VARCHAR(100) NOT NULL,
 PRIMARY KEY(Id_Giro)
)
TYPE=InnoDB;

CREATE TABLE usuarios (
 Id_Usuario VARCHAR(50) NOT NULL,
 Tipo_Usuario VARCHAR(50) NOT NULL,
 Clave VARCHAR(50) NOT NULL,
 Nivel INT(1) NOT NULL,
 PRIMARY KEY(Id_Usuario)
)
TYPE=InnoDB;

CREATE TABLE sector (
 Id_Sec INT(11) NOT NULL AUTO_INCREMENT,
 Nombre_Giro VARCHAR(100) NOT NULL,
 Nombre_Sec VARCHAR(100) NOT NULL,
 PRIMARY KEY(Id_Sec)
)
TYPE=InnoDB;

CREATE TABLE escolaridad (
 Id_Esc INT(11) NOT NULL AUTO_INCREMENT,
 Nombre_Esc VARCHAR(100) NOT NULL,
 PRIMARY KEY(Id_Esc)
```

```
)
TYPE=InnoDB;

CREATE TABLE del_mun (
  Id_Del_Mun INT(11) NOT NULL AUTO_INCREMENT,
  Nombre_Est VARCHAR(200) NOT NULL,
  Nombre_Del_Mun VARCHAR(200) NOT NULL,
  PRIMARY KEY(Id_Del_Mun)
)
TYPE=InnoDB;

CREATE TABLE carrera (
  Id_Carr INT(11) NOT NULL AUTO_INCREMENT,
  Nombre_Carr VARCHAR(200) NOT NULL,
  PRIMARY KEY(Id_Carr)
)
TYPE=InnoDB;

CREATE TABLE candidato (
  Num_Cta VARCHAR(50) NOT NULL,
  Id_Candidato BIGINT(20) NOT NULL,
  Carrera VARCHAR(100) NOT NULL,
  Area VARCHAR(250) NOT NULL,
  Escolaridad VARCHAR(30) NOT NULL,
  Promedio FLOAT NOT NULL,
  Edo_Civil VARCHAR(15) NOT NULL,
  Nombre VARCHAR(200) NOT NULL,
  Apellido_Pat VARCHAR(200) NOT NULL,
  Apellido_Mat VARCHAR(200) NOT NULL,
  Edad INT(3) NOT NULL,
  Direccion_Cand VARCHAR(300) NOT NULL,
  Nombre_Estado VARCHAR(200) NOT NULL,
  Nombre_Del_Mun VARCHAR(200) NOT NULL,
  Telefono VARCHAR(20) NOT NULL,
  E_Mail VARCHAR(50) NOT NULL,
  Fecha_Nacimiento DATE NOT NULL,
  Idioma VARCHAR(50) NULL,
  Dominio INT(3) NULL,
  Otros_Estudios VARCHAR(300) NULL,
  Horario_Disp VARCHAR(200) NOT NULL,
  Sueldo VARCHAR(200) NOT NULL,
  Actividades_Desemp VARCHAR(500) NOT NULL,
  Observaciones VARCHAR(500) NULL,
  Foto VARCHAR(100) NULL,
  Estado VARCHAR(20) NOT NULL,
  Fecha_Inicio DATE NOT NULL,
  Fecha_Cancel DATE NOT NULL,
  Fecha_Fin DATE NOT NULL,
  Estilo VARCHAR(100) NOT NULL,
PRIMARY KEY(Num_Cta),
  FOREIGN KEY(Num_Cta)
    REFERENCES usuarios(Id_Usuario)
      ON DELETE CASCADE
      ON UPDATE CASCADE
)
TYPE=InnoDB;
```

```
CREATE TABLE empresa (
 RFC VARCHAR(50) NOT NULL,
 Id_Empresa BIGINT(20) NOT NULL,
 Nombre_Emp VARCHAR(200) NOT NULL,
 Telefono_Emp VARCHAR(20) NOT NULL,
 E_Mail VARCHAR(50) NOT NULL,
 Sitio_Web VARCHAR(300) NULL,
 Direccion_Emp VARCHAR(300) NOT NULL,
 Nombre_Estado VARCHAR(200) NOT NULL,
 Nombre_Del_Mun VARCHAR(200) NOT NULL,
 Nombre_Giro VARCHAR(50) NOT NULL,
 Nombre_Sector VARCHAR(50) NOT NULL,
 Logotipo VARCHAR(100) NOT NULL,
 PRIMARY KEY(RFC),
 FOREIGN KEY(RFC)
  REFERENCES usuarios(Id_Usuario)
   ON DELETE CASCADE
   ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE experiencia (
 Id_Experiencia INT(11) NOT NULL,
 Num_Cta VARCHAR(50) NOT NULL,
 Puesto VARCHAR(250) NOT NULL,
 Empresa VARCHAR(200) NOT NULL,
 Anyos INT(2) NULL,
 Meses INT(2) NULL,
 Sueldo FLOAT NOT NULL,
 PRIMARY KEY(Id_Experiencia, Num_Cta),
 FOREIGN KEY(Num_Cta)
  REFERENCES candidato(Num_Cta)
   ON DELETE CASCADE
   ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE cuest (
 Id_Cuest INT(11) NOT NULL AUTO_INCREMENT,
 Num_Cta VARCHAR(50) NOT NULL,
 Nombre VARCHAR(200) NOT NULL,
 Razon TEXT NOT NULL,
 Fecha_Cuest DATE NOT NULL,
 PRIMARY KEY(Id_Cuest),
 FOREIGN KEY(Num_Cta)
  REFERENCES candidato(Num_Cta)
   ON DELETE CASCADE
   ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE visitas (
 Num_Cta VARCHAR(50) NOT NULL,
 visitas BIGINT(20) NOT NULL,
 Fecha_Visita DATE NOT NULL,
```

```
  PRIMARY KEY(Num_Cta),
  FOREIGN KEY(Num_Cta)
   REFERENCES candidato(Num_Cta)
     ON DELETE CASCADE
     ON UPDATE CASCADE
)
TYPE=InnoDB;

CREATE TABLE vacante (
  Id_Vacante INT(11) NOT NULL AUTO_INCREMENT,
  RFC VARCHAR(50) NOT NULL,
  Puesto_Solicitado VARCHAR(300) NOT NULL,
  Carrera VARCHAR(200) NOT NULL,
  Escolaridad VARCHAR(100) NOT NULL,
  Experiencia VARCHAR(200) NOT NULL,
  Edad  VARCHAR(50) NULL,
  Sexo VARCHAR(10) NOT NULL,
  Idioma VARCHAR(50) NULL,
  Dominio INT(2) NULL,
  Sueldo VARCHAR(200) NULL,
  Entrevistador VARCHAR(200) NOT NULL,
  Cargo VARCHAR(150) NOT NULL,
  Telefono VARCHAR(20) NOT NULL,
  E_Mail VARCHAR(50) NULL,
  Horario_Entrevista VARCHAR(100) NULL,
  Num_Vacantes INT(3) NOT NULL,
  Observaciones VARCHAR(500) NULL,
  Fecha_Publicacion DATETIME NOT NULL,
  Estado INT(1) NOT NULL DEFAULT '1',
  PRIMARY KEY(Id_Vacante),
  FOREIGN KEY(RFC)
   REFERENCES empresa(RFC)
     ON DELETE CASCADE
     ON UPDATE CASCADE
)
TYPE=InnoDB;

ALTER TABLE `empresa` ADD UNIQUE (
`Id_Empresa`
);

ALTER TABLE `candidato` ADD UNIQUE (
`Id_Candidato`
);

INSERT INTO `carrera` VALUES(1, 'Bioquimica Diagnostica');
INSERT INTO `carrera` VALUES(2, 'Farmacia');
INSERT INTO `carrera` VALUES(3, 'Ingenier&iacute;a Agr&iacute;cola');
INSERT INTO `carrera` VALUES(4, 'Ingenier&iacute;a de los Alimentos');
INSERT INTO `carrera` VALUES(5, 'Ingenier&iacute;a Mec&aacute;nica El&eacute;ctrica');
INSERT INTO `carrera` VALUES(6, 'Ingenier&iacute;a Qu&iacute;mica');
INSERT INTO `carrera` VALUES(7, 'Licenciatura en Administraci&oacute;n');
INSERT INTO `carrera` VALUES(8, 'Licenciatura en Contadur&iacute;a');
INSERT INTO `carrera` VALUES(9, 'Licenciatura en Dise&ntilde;o y Comunicaci&oacute;n
Visual');
INSERT INTO `carrera` VALUES(10, 'Licenciatura en Inform&aacute;tica');
```

```
INSERT INTO `carrera` VALUES(11, 'Licenciatura en Medicina Veterinaria y Zootecnia');
INSERT INTO `carrera` VALUES(12, 'Licenciatura en Qu&iacute;mica');
INSERT INTO `carrera` VALUES(13, 'Licenciatura en Qu&iacute;mica
Farmac&eacute;utico-Biol&oacute;gica');
INSERT INTO `carrera` VALUES(14, 'Licenciatura en Qu&iacute;mica Industrial');
INSERT INTO `carrera` VALUES(15, 'Tecnolog&iacute;a');

INSERT INTO `escolaridad` VALUES(1, '1&deg; Semestre');
INSERT INTO `escolaridad` VALUES(2, '2&deg; Semestre');
INSERT INTO `escolaridad` VALUES(3, '3&deg; Semestre');
INSERT INTO `escolaridad` VALUES(4, '4&deg; Semestre');
INSERT INTO `escolaridad` VALUES(5, '5&deg; Semestre');
INSERT INTO `escolaridad` VALUES(6, '6&deg; Semestre');
INSERT INTO `escolaridad` VALUES(7, '7&deg; Semestre');
INSERT INTO `escolaridad` VALUES(8, '8&deg; Semestre');
INSERT INTO `escolaridad` VALUES(9, '9&deg; Semestre');
INSERT INTO `escolaridad` VALUES(10, '10&deg; Semestre');
INSERT INTO `escolaridad` VALUES(11, 'Pasante');
INSERT INTO `escolaridad` VALUES(12, 'Titulado');
INSERT INTO `escolaridad` VALUES(13, 'Posgrado');

INSERT INTO `estado` VALUES(1, 'Aguascalientes');
INSERT INTO `estado` VALUES(2, 'Baja California');
INSERT INTO `estado` VALUES(3, 'Baja California Sur');
INSERT INTO `estado` VALUES(4, 'Campeche');
INSERT INTO `estado` VALUES(5, 'Coahuila');
INSERT INTO `estado` VALUES(6, 'Colima');
INSERT INTO `estado` VALUES(7, 'Chiapas');
INSERT INTO `estado` VALUES(8, 'Chihuahua');
INSERT INTO `estado` VALUES(9, 'D.F.');
INSERT INTO `estado` VALUES(10, 'Durango');
INSERT INTO `estado` VALUES(11, 'Estado de M&eacute;xico');
INSERT INTO `estado` VALUES(12, 'Guanajuato');
INSERT INTO `estado` VALUES(13, 'Guerrero');
INSERT INTO `estado` VALUES(14, 'Hidalgo');
INSERT INTO `estado` VALUES(15, 'Jalisco');
INSERT INTO `estado` VALUES(16, 'Michoac&aacute;n');
INSERT INTO `estado` VALUES(17, 'Morelos');
INSERT INTO `estado` VALUES(18, 'Nayarit');
INSERT INTO `estado` VALUES(19, 'Nuevo Le&oacute;n');
INSERT INTO `estado` VALUES(20, 'Oaxaca');
INSERT INTO `estado` VALUES(21, 'Puebla');
INSERT INTO `estado` VALUES(22, 'Quer&eacute;taro');
INSERT INTO `estado` VALUES(23, 'Quintana Roo');
INSERT INTO `estado` VALUES(24, 'San Luis Potos&iacute;');
INSERT INTO `estado` VALUES(25, 'Sinaloa');
INSERT INTO `estado` VALUES(26, 'Sonora');
INSERT INTO `estado` VALUES(27, 'Tabasco');
INSERT INTO `estado` VALUES(28, 'Tamaulipas');
INSERT INTO `estado` VALUES(29, 'Tlaxcala');
INSERT INTO `estado` VALUES(30, 'Veracruz');
INSERT INTO `estado` VALUES(31, 'Yucat&aacute;n');
INSERT INTO `estado` VALUES(32, 'Zacatecas');

INSERT INTO `del_mun` VALUES(1, 'Estado de M&eacute;xico', 'Acambay');
INSERT INTO `del_mun` VALUES(2, 'Estado de M&eacute;xico', 'Acolman');
```

INSERT INTO `del_mun` VALUES(3, 'Estado de México', 'Aculco');
INSERT INTO `del_mun` VALUES(4, 'Estado de México', 'Almoloya de Alquisiras');
INSERT INTO `del_mun` VALUES(5, 'Estado de México', 'Almoloya de Juárez');
INSERT INTO `del_mun` VALUES(6, 'Estado de México', 'Almoloya del Río');
INSERT INTO `del_mun` VALUES(7, 'Estado de México', 'Amanalco');
INSERT INTO `del_mun` VALUES(8, 'Estado de México', 'Amatepec');
INSERT INTO `del_mun` VALUES(9, 'Estado de México', 'Amecameca');
INSERT INTO `del_mun` VALUES(10, 'Estado de México', 'Apaxco');
INSERT INTO `del_mun` VALUES(11, 'Estado de México', 'Atenco');
INSERT INTO `del_mun` VALUES(12, 'Estado de México', 'Atizapán');
INSERT INTO `del_mun` VALUES(13, 'Estado de México', 'Atizapán de Zaragoza');
INSERT INTO `del_mun` VALUES(14, 'Estado de México', 'Atlacomulco');
INSERT INTO `del_mun` VALUES(15, 'Estado de México', 'Atlautla');
INSERT INTO `del_mun` VALUES(16, 'Estado de México', 'Axapusco');
INSERT INTO `del_mun` VALUES(17, 'Estado de México', 'Ayapango');
INSERT INTO `del_mun` VALUES(18, 'Estado de México', 'Calimaya');
INSERT INTO `del_mun` VALUES(19, 'Estado de México', 'Capulhuac');
INSERT INTO `del_mun` VALUES(20, 'Estado de México', 'Coacalco de Berriozábal');
INSERT INTO `del_mun` VALUES(21, 'Estado de México', 'Coatepec Harinas');
INSERT INTO `del_mun` VALUES(22, 'Estado de México', 'Cocotitlán');
INSERT INTO `del_mun` VALUES(23, 'Estado de México', 'Coyotepec');
INSERT INTO `del_mun` VALUES(24, 'Estado de México', 'Cuautitlán');
INSERT INTO `del_mun` VALUES(25, 'Estado de México', 'Chalco');
INSERT INTO `del_mun` VALUES(26, 'Estado de México', 'Chapa de Mota');
INSERT INTO `del_mun` VALUES(27, 'Estado de México', 'Chapultepec');
INSERT INTO `del_mun` VALUES(28, 'Estado de México', 'Chiautla');
INSERT INTO `del_mun` VALUES(29, 'Estado de México', 'Chicoloapan');
INSERT INTO `del_mun` VALUES(30, 'Estado de México', 'Chiconcuac');
INSERT INTO `del_mun` VALUES(31, 'Estado de México', 'Chimalhuacán');
INSERT INTO `del_mun` VALUES(32, 'Estado de México', 'Donato Guerra');
INSERT INTO `del_mun` VALUES(33, 'Estado de México', 'Ecatepec de Morelos');
INSERT INTO `del_mun` VALUES(34, 'Estado de México', 'Ecatzingo');
INSERT INTO `del_mun` VALUES(35, 'Estado de México', 'Huehuetoca');
INSERT INTO `del_mun` VALUES(36, 'Estado de México', 'Hueypoxtla');
INSERT INTO `del_mun` VALUES(37, 'Estado de México', 'Huixquilucan');
INSERT INTO `del_mun` VALUES(38, 'Estado de México', 'Isidro Fabela');
INSERT INTO `del_mun` VALUES(39, 'Estado de México', 'Ixtapaluca');
INSERT INTO `del_mun` VALUES(40, 'Estado de México', 'Ixtapan de la Sal');
INSERT INTO `del_mun` VALUES(41, 'Estado de México', 'Ixtapan del Oro');
INSERT INTO `del_mun` VALUES(42, 'Estado de México', 'Ixtlahuaca');
INSERT INTO `del_mun` VALUES(43, 'Estado de México', 'Xalatlaco');
INSERT INTO `del_mun` VALUES(44, 'Estado de México', 'Jaltenco');
INSERT INTO `del_mun` VALUES(45, 'Estado de México', 'Jilotepec');
INSERT INTO `del_mun` VALUES(46, 'Estado de México', 'Jilotzingo');
INSERT INTO `del_mun` VALUES(47, 'Estado de México', 'Jiquipilco');
INSERT INTO `del_mun` VALUES(48, 'Estado de México', 'Jocotitlán');
INSERT INTO `del_mun` VALUES(49, 'Estado de México', 'Joquicingo');
INSERT INTO `del_mun` VALUES(50, 'Estado de México', 'Juchitepec');
INSERT INTO `del_mun` VALUES(51, 'Estado de México', 'Lerma');
INSERT INTO `del_mun` VALUES(52, 'Estado de México', 'Malinalco');
INSERT INTO `del_mun` VALUES(53, 'Estado de México', 'Melchor Ocampo');
INSERT INTO `del_mun` VALUES(54, 'Estado de México', 'Metepec');

INSERT INTO `del_mun` VALUES(55, 'Estado de México', 'Mexicaltzingo');
INSERT INTO `del_mun` VALUES(56, 'Estado de México', 'Morelos');
INSERT INTO `del_mun` VALUES(57, 'Estado de México', 'Naucalpan de Juárez');
INSERT INTO `del_mun` VALUES(58, 'Estado de México', 'Nezahualcoyótl');
INSERT INTO `del_mun` VALUES(59, 'Estado de México', 'Nextlalpan');
INSERT INTO `del_mun` VALUES(60, 'Estado de México', 'Nicolás Romero');
INSERT INTO `del_mun` VALUES(61, 'Estado de México', 'Nopaltepec');
INSERT INTO `del_mun` VALUES(62, 'Estado de México', 'Ocoyoacac');
INSERT INTO `del_mun` VALUES(63, 'Estado de México', 'Ocuilan');
INSERT INTO `del_mun` VALUES(64, 'Estado de México', 'El Oro');
INSERT INTO `del_mun` VALUES(65, 'Estado de México', 'Otumba');
INSERT INTO `del_mun` VALUES(66, 'Estado de México', 'Otzoloapan');
INSERT INTO `del_mun` VALUES(67, 'Estado de México', 'Otzolotepec');
INSERT INTO `del_mun` VALUES(68, 'Estado de México', 'Ozumba');
INSERT INTO `del_mun` VALUES(69, 'Estado de México', 'Papalotla');
INSERT INTO `del_mun` VALUES(70, 'Estado de México', 'La Paz');
INSERT INTO `del_mun` VALUES(71, 'Estado de México', 'Polotitlán');
INSERT INTO `del_mun` VALUES(72, 'Estado de México', 'Rayón');
INSERT INTO `del_mun` VALUES(73, 'Estado de México', 'San Antonio la isla');
INSERT INTO `del_mun` VALUES(74, 'Estado de México', 'San Felipe del Progreso');
INSERT INTO `del_mun` VALUES(75, 'Estado de México', 'San Martín de las Pirámides');
INSERT INTO `del_mun` VALUES(76, 'Estado de México', 'San Matero Atenco');
INSERT INTO `del_mun` VALUES(77, 'Estado de México', 'San Simón de Guerrero');
INSERT INTO `del_mun` VALUES(78, 'Estado de México', 'Santo Tomás');
INSERT INTO `del_mun` VALUES(79, 'Estado de México', 'Soyaniquilpan de Juárez');
INSERT INTO `del_mun` VALUES(80, 'Estado de México', 'Sultepec');
INSERT INTO `del_mun` VALUES(81, 'Estado de México', 'Tecámac');
INSERT INTO `del_mun` VALUES(82, 'Estado de México', 'Tejupilco');
INSERT INTO `del_mun` VALUES(83, 'Estado de México', 'Temamatla');
INSERT INTO `del_mun` VALUES(84, 'Estado de México', 'Temascalapa');
INSERT INTO `del_mun` VALUES(85, 'Estado de México', 'Temascalcingo');
INSERT INTO `del_mun` VALUES(86, 'Estado de México', 'Temascaltepec');
INSERT INTO `del_mun` VALUES(87, 'Estado de México', 'Temoaya');
INSERT INTO `del_mun` VALUES(88, 'Estado de México', 'Tenancingo');
INSERT INTO `del_mun` VALUES(89, 'Estado de México', 'Tenancingo del Aire');
INSERT INTO `del_mun` VALUES(90, 'Estado de México', 'Tenancingo del Valle');
INSERT INTO `del_mun` VALUES(91, 'Estado de México', 'Teoloyucán');
INSERT INTO `del_mun` VALUES(92, 'Estado de México', 'Teotihuacan');
INSERT INTO `del_mun` VALUES(93, 'Estado de México', 'Tepetlaoxtoc');
INSERT INTO `del_mun` VALUES(94, 'Estado de México', 'Tepetlixpa');
INSERT INTO `del_mun` VALUES(95, 'Estado de México', 'Tepotzotlán');
INSERT INTO `del_mun` VALUES(96, 'Estado de México', 'Tequixquiac');
INSERT INTO `del_mun` VALUES(97, 'Estado de México', 'Texcaltitlán');
INSERT INTO `del_mun` VALUES(98, 'Estado de México', 'Texcalyacac');
INSERT INTO `del_mun` VALUES(99, 'Estado de México', 'Texcoco');
INSERT INTO `del_mun` VALUES(100, 'Estado de México', 'Tezoyuca');
INSERT INTO `del_mun` VALUES(101, 'Estado de México', 'Tianquistenco');
INSERT INTO `del_mun` VALUES(102, 'Estado de México', 'Timilpan');
INSERT INTO `del_mun` VALUES(103, 'Estado de México', 'Tlalmanalco');
INSERT INTO `del_mun` VALUES(104, 'Estado de México', 'Tlalnepantla de Baz');
INSERT INTO `del_mun` VALUES(105, 'Estado de México', 'Tlatlaya');

```
INSERT INTO `del_mun` VALUES(106, 'Estado de M&eacute;xico', 'Toluca');
INSERT INTO `del_mun` VALUES(107, 'Estado de M&eacute;xico', 'Tonatico');
INSERT INTO `del_mun` VALUES(108, 'Estado de M&eacute;xico', 'Tultepec');
INSERT INTO `del_mun` VALUES(109, 'Estado de M&eacute;xico', 'Tultitlan');
INSERT INTO `del_mun` VALUES(110, 'Estado de M&eacute;xico', 'Valle de Bravo');
INSERT INTO `del_mun` VALUES(111, 'Estado de M&eacute;xico', 'Villa de Allende');
INSERT INTO `del_mun` VALUES(112, 'Estado de M&eacute;xico', 'Villa del
Carb&oacute;n');
INSERT INTO `del_mun` VALUES(113, 'Estado de M&eacute;xico', 'Villa Guerrero');
INSERT INTO `del_mun` VALUES(114, 'Estado de M&eacute;xico', 'Villa Victoria');
INSERT INTO `del_mun` VALUES(115, 'Estado de M&eacute;xico', 'Xonacatl&aacute;n');
INSERT INTO `del_mun` VALUES(116, 'Estado de M&eacute;xico', 'Zacazonapan');
INSERT INTO `del_mun` VALUES(117, 'Estado de M&eacute;xico', 'Zacualpan');
INSERT INTO `del_mun` VALUES(118, 'Estado de M&eacute;xico', 'Zinacantepec');
INSERT INTO `del_mun` VALUES(119, 'Estado de M&eacute;xico', 'Zumpahuac&aacute;n');
INSERT INTO `del_mun` VALUES(120, 'Estado de M&eacute;xico', 'Zumpango');
INSERT INTO `del_mun` VALUES(121, 'Estado de M&eacute;xico', 'Cuautitl&aacute;n
Izcalli');
INSERT INTO `del_mun` VALUES(122, 'Estado de M&eacute;xico', 'Valle de Chalco
Solidaridad');
INSERT INTO `del_mun` VALUES(123, 'D.F.', 'Azcapotzalco');
INSERT INTO `del_mun` VALUES(124, 'D.F.', 'Coyoac&aacute;n');
INSERT INTO `del_mun` VALUES(125, 'D.F.', 'Cuajimalpa de Morelos');
INSERT INTO `del_mun` VALUES(126, 'D.F.', 'Gustavo A. Madero');
INSERT INTO `del_mun` VALUES(127, 'D.F.', 'Iztacalco');
INSERT INTO `del_mun` VALUES(128, 'D.F.', 'Iztapalapa');
INSERT INTO `del_mun` VALUES(129, 'D.F.', 'La Magdalena Contreras');
INSERT INTO `del_mun` VALUES(130, 'D.F.', 'Milpa Alta');
INSERT INTO `del_mun` VALUES(131, 'D.F.', '&Aacute;lvaro Obreg&oacute;n');
INSERT INTO `del_mun` VALUES(132, 'D.F.', 'Tl&aacute;huac');
INSERT INTO `del_mun` VALUES(133, 'D.F.', 'Tlalpan');
INSERT INTO `del_mun` VALUES(134, 'D.F.', 'Xochimilco');
INSERT INTO `del_mun` VALUES(135, 'D.F.', 'Benito Ju&aacute;rez');
INSERT INTO `del_mun` VALUES(136, 'D.F.', 'Cuauht&eacute;moc');
INSERT INTO `del_mun` VALUES(137, 'D.F.', 'Miguel Hidalgo');
INSERT INTO `del_mun` VALUES(138, 'D.F.', 'Venustiano Carranza');

INSERT INTO `giro` VALUES(1, 'Industrial');
INSERT INTO `giro` VALUES(2, 'Comercial');
INSERT INTO `giro` VALUES(3, 'Servicios');
INSERT INTO `giro` VALUES(4, 'Agr&iacute;cola');

INSERT INTO `sector` VALUES(1, 'Industrial', 'Extractiva');
INSERT INTO `sector` VALUES(2, 'Industrial', 'Manufacturera');
INSERT INTO `sector` VALUES(3, 'Industrial', 'Agropecuaria');
INSERT INTO `sector` VALUES(4, 'Comercial', 'Mayorista');
INSERT INTO `sector` VALUES(5, 'Comercial', 'Menudeo');
INSERT INTO `sector` VALUES(6, 'Comercial', 'Minorista');
INSERT INTO `sector` VALUES(7, 'Comercial', 'Comisionista');
INSERT INTO `sector` VALUES(8, 'Servicios', 'Servicios P&uacute;blicos');
INSERT INTO `sector` VALUES(9, 'Servicios', 'Servicios Privados');
INSERT INTO `sector` VALUES(10, 'Servicios', 'Transporte');
INSERT INTO `sector` VALUES(11, 'Servicios', 'Turismo');
INSERT INTO `sector` VALUES(12, 'Servicios', 'Instituci&oacute;n Financiera');
INSERT INTO `sector` VALUES(13, 'Servicios', 'Educaci&oacute;n');
INSERT INTO `sector` VALUES(14, 'Servicios', 'Salubridad');
```

```
INSERT INTO `sector` VALUES(15, 'Servicios', 'Finanzas');
INSERT INTO `sector` VALUES(16, 'Servicios', 'Seguros');

INSERT INTO `usuarios` VALUES('admin', 'Administrador', 'flavio', 1);
```

7. BIBLIOGRAFÍA

Álvarez, Miguel Ángel (2001). Breve Historia de PHP. Desarrolloweb. Disponible en: http://www.desarrolloweb.com/articulos/436.php

Álvarez, Miguel Ángel (2008) Qué es el DOM. desarrolloweb.com Disponible en: http://www.desarrolloweb.com/articulos/que-es-el-dom.html

Barcia, Diego (2003). ¿Qué es CSS?. Maestros del Web. Disponible en: http://www.maestrosdelweb.com/editorial/introcss/

Bakken, Stig Sæther; Aulbach, Alexander; Schmid, Egon, Winstead, Jim; Wilson, Lars Torben; Lerdorf, Rasmus; Zmievski Andrei y Jouni Atho (2002). Manual de PHP (1era edición). Disponible en: http://www.librosintinta.in/biblioteca/ver-pdf/www.openboxer.260mb.com/pdf/php/phpCompleto.pdf.htx (pp. 2).

Bobadilla Jesús, Alcocer Alejandro, Alonso Santiago, Gutiérrez Abraham (2001). HTML Dinámico, ASP y Javascript a través de ejmeplos. Alfaomega Rama D.F. 2 p.p.

Camps Paré, Rafael, Castillas Santillán, Luis Alberto, Costal Costa, Dolors, Gilbert Ginestá, Marc, Matín Escofet, Carme, Pérez Mora, Oscar. Sofware libre Base de Datos. Barcelona: Formación de Posgrado. 2005.

Davis, Michele E. y Phillips, Jon A. (2008). PHP y MySQL (Edición Española). Madrid:Anaya Multimedia . 23 p.p.

Eguíluz Pérez, Javier (8 de mayo de 2009). Introducción a CSS. 223 paginas. Disponible en: http://www.librosweb.es/css/

Eguíluz Pérez, Javier, (2009). Introducción a JavaScript. Librosweb.es 5, 140 Disponible en http://www.librosweb.es/javascript/

Eguíluz Pérez, Javier (2011) Introducción a AJAX. Librosweb.es Disponible en: http://www.librosweb.es/ajax/

Flanagan, David (2007) Java Script La Guía Definitiva Madrid: Anaya Multimedia O'REILLY. 328, 329 p.p.

Gilfillan, Ian, (2003). La biblia de MySQL / Ian Gilfillan.Madrid: Anaya Multimedia, c2003. Colección-serie La Biblia. 41 p.p.

Gorostiza, Ignacio (2009). Tutorial Cache Web: cómo gestionar el cacheo de nuestros contenidos. Hello Google Artículos para crecer con su empresa en Internet. Disponible en:

http://www.hellogoogle.com/tutorial-cache-web/

Grupo Eidos (2000), Lenguaje HTML.LaLibreríaDigital. Disponible en: http://www.jargon.com.ar/downloads/Lenguaje.HTML.pdf

González Suárez, Maikel (2011) Introducción a la manipulación del DOM mediante Javascript. Maestros del Web Disponible en: http://www.maestrosdelweb.com/editorial/dom/

Joyanes Aguilar, Luis, (1999). Fundamentos de Programación. McGrawHill Andalucía, 703, 10 p.p.

Libros.es (2011). Breve historia de HTML.librosweb.es. Disponible en:

http://www.librosweb.es/xhtml/capitulo1/breve_historia_de_html.html.

Lanzillotta, Analía (2005) Definición de Cliente/Servidor

http://www.mastermagazine.info/termino/4294.php

López Quijado, José (2007). Domine JavaScript(2ª edición), D.F: Alfaomega Rama. 6, 111 p.p.

Leopoldo, Carlos (14 de Octubre 2010). Compatibilidad de propiedades CSS3 en distintos navegadores. Techtastico tecnología y más que eso. Disponible en: http://techtastico.com/post/compatibilidad-css3-navegadores/

López, Jorge (29 de enero, 2010). Los motores de renderizado de los navegadores web. Circulo de Maquetadores. Comuniidad de Maquetadores Web. Disponible en: http://www.circulodemaquetadores.com/motores-de-renderizado-navegadores

Martín Bailón, Alejandro (2011). Introducción a HTML 5 e Internet Explorer 9. desarrollo web.com. Disponible en: http://www.desarrolloweb.com/articulos/intro-html5-ie9.html

Moraga de la Rubia, M. Ángeles (2001). El modelo de Datos Jerárquico. Ciudad Real: Escuela Superior de Informática, 15 p.

Orós, Juan Carlos (2008). Diseño de páginas Web con XHTML, JavaScript y CSS (2da. edición). D.F: Alfaomega Rama. 51, 347, 356 pp.

Pérez, Damián (2009). ¿Qué es JavaScript? Maestros del Web

Disponible en http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/

Pérez Valdés, Damián (26 oct. 2007). ¿Qué son las bases de datos?. Maestros del Web. Disponible en: http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/

Pérez Valdés, Damián (26 oct. 2007). ¿Qué son las bases de datos?. Maestros del Web. Disponible en: http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-

de-datos/

Rivera Zarate, Omar A. 2007, Base de Datos(II). slideshare. Disponible: http://www.slideshare.net/omarzon/modelo-de-datos-202475

Sánchez Asenjo, Jorge, (2010). Apuntes completos Gestión de Bases de Datos. Centro Don Bosco Villamuriel de Cerrato, 188, 11-12

Soria Momparler, Ramón (2002). Diseño y creación de páginas Web HTML 4 (2ª edición). D.F: Alfaomega

Tejerina, Martín (2011). Programación orientada a Objetos - (OOP, Object Oriented Programming). Ilustrados.com. Disponible en: http://www.ilustrados.com/tema/1283/Programacion-orientada-Objetos----Object.html

Velarde, Peter (2004). Páginas Web estáticas vs dinámica. Sappiens.com Disponible en: http://www.sappiens.com/castellano/articulos.nsf/Marketing/P%C3%A1ginas_Web_est%C3%A1ticas_vs_din%C3%A1micas/992EF9270B53D545C12573D900169C4F!opendocument

Welling, Luke; Thomson, Laura (2005). Desarrollo Web con PHP y MySQL. PHP 5 y MySQL 4.1 y 5 (Edición española). Madrid: Anaya Multimedia. Pp 33. 974 pág.

8. GLOSARIO

Algoritmo: Un algoritmo describe la secuencia ordenada de pasos sin ambigüedades que conducen a la solución de un problema (Joyanes Aguilar, 1999).

Analizador sintáctico: Es una de las partes de un compilador que transforma su entrada en un árbol de derivación. El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada.

API: En inglés significa Application Programming Interface o en español Interfaz de Programación de Aplicaciones. Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas comúnmente "librerías").

Atomicidad: La atomicidad de una transacción garantiza que todas sus acciones sean realizadas o ninguna sea ejecutada, en el caso de la transacción bancaria o se ejecuta tanto el "deposito-deducción" o ninguna acción será realizada.
(http://www.osmosislatina.com/aplicaciones/bases_de_datos.htm).

Backbone: La palabra backbone se refiere a las principales conexiones troncales de Internet. Está compuesta de un gran número de routers comerciales, gubernamentales, universitarios y otros de gran capacidad interconectados que llevan los datos a través de países, continentes y océanos del mundo mediante cables de fibra óptica.

Base de Datos Distribuida: Es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas en diferentes espacios lógicos e interconectados por una red de comunicaciones.

CGI: Significa Interfaz de entrada común (Common Gateway Interface) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos

de un programa ejecutado en un servidor web.

Código ASCII: Es el acrónimo en inglés de American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información) y es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales.

Compilador: Los compiladores son programas o herramientas que toma un texto (código fuente) escrito en un lenguaje de alto nivel y lo traduce a un lenguaje comprensible por las computadoras (código objeto).

Dato: Cualquier elemento informativo que tenga relevancia para el sistema. Desde el inicio de la informática se ha reconocido al dato como al elemento fundamental de trabajo en un ordenador.

DHTML: HTML Dinámico que permite la creación de efectos, por ejemplo, variar el color de fondo de una página, aumentar el tamaño de un texto, variar el color de un botón, hacer aparecer una lista desplegable, mover una figura, entre muchas otras, y todo sin la necesidad de cargar una nueva página.

Escalabilidad: es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para extender el margen de operaciones sin perder calidad, es decir estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

FLASH: Es un conjunto de tecnologías desarrolladas por la empresa Macromedia, para mostrar contenido multimedia en la Web y aplicaciones de escritorio

GPL: Licencia de software libre

Grafo: Los grafos representan conjuntos de objetos que no tienen restricción de relación entre ellos. Un grafo puede representar varias cosas de la realidad cotidiana, tales como

mapas de carreteras, vías férreas, circuitos eléctricos, etc.

GUI: Son las siglas de la Interfaz Gráfica de Usuario, que permite a los usuarios interactuar con los sistemas computacionales de una manera sencilla y ordenada.

Hipertexto: Es el nombre que recibe el texto que conduce a otro relacionado. Si el usuario selecciona un hipervínculo el programa muestra el documento enlazado.

IETF: Internet Engineering Task Force (Grupo Especial sobre Ingeniería de Internet) es una organización internacionalmente reconocida por ser la entidad que regula las propuestas y los estándares de Internet y tiene como objetivos el contribuir a la ingeniería de Intenet, actuando en diversas áreas, como transporte, encaminamiento, seguridad.

Inferencia: Proceso mediante el cual se obtienen conclusiones en base a la información conocida.

Información: Es un conjunto de datos significativos y pertinentes que describen sucesos o entidades.

Lenguaje Binario: Es un lenguaje que utiliza al sistema binario que solo permite 0 y 1, como base para generar códigos que pueden ser interpretados por un circuito programable, por ejemplo los microprocesadores de las computadoras.

Motor Zend: Es un motor de procesamiento para la interpretación y cifrado del código php, desde la versión 4. Desarrollado por Zend Tecnologies para brindar un equipo de soporte y acelerar la carga de aplicaciones realizadas con php.

Nodo: Es un registro que contiene un dato de interés y puede contener la dirección de otro nodo al cual se dirige.

ODBC: Es un estándar de acceso a base de datos desarrollado por SQL Access Group en

1992, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar el sistema de gestión de bases de datos.

Open Source: El software OpenSource se define por la licencia que lo acompaña, que garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente.

Pixel: es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

Plataforma: es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones).

Portabilidad: Característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma.

Redundancia: Repetición innecesaria de información.

Renderizado web: Es cuando se muestra en pantalla el contenido de la página con estilo, estructura y forma de acuerdo a contenido marcado e información del formato.

Script: Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos.

Semántica: Se refiere a los aspectos del significado de una palabra, en los lenguajes de programación existen palabras reservadas que tienen significados específicos.

SGML: Estándar de Lenguaje de Marcado Generalizado (Standard Generalized Markup Language), consiste en un sistema para la organización y etiquetado de documentos y para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

Sintáctica: Se refiere a las reglas que componen la estructura de un lenguaje en general, los lenguajes de programación se basan en una sintaxis específica que es capaz de interpretar una computadora.

SOAP: (Siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

URL: Son las siglas de Uniform Resource Locator (Localizador de Recurso Uniforme), la dirección global de documentos y de otros recursos en la World Wide Web.

XML: Es un lenguaje de Etiquetado Extensible (Extensible Markup Language) muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

WWW: La World Wide Web (WWW) es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.