



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN.**

TESIS

**DISEÑO Y DESARROLLO DE UN SISTEMA EN LÍNEA PARA LA
BOLSA DE TRABAJO DE LA FES CUAUTITLÁN CON ZEND
FRAMEWORK 2.**

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA.

PRESENTA:

MUÑOZ MACÍAS GUILLERMO.

JULIÁN MENDOZA SILVIA.

ASESOR: M.G.T.I. LEONEL GUALBERTO LÓPEZ SALAZAR.

CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO 2017.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

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

FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN

ASUNTO: VOTO APROBATORIO

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Exámenes Profesionales
de la FES Cuautitlán.**

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: Trabajo de Tesis

Diseño y Desarrollo de un sistema en línea para la bolsa de trabajo FES Cuautitlán con zend framework 2

Que presenta el pasante: GUILLERMO MUÑOZ MACÍAS

Con número de cuenta: 41207601-5 para obtener el Título de la carrera: Licenciatura 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 HABLARÁ EL ESPÍRITU"

Cuautitlán Izcalli, Méx. a 13 de marzo de 2017.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	M.C.C. Valentín Roldán Vázquez	
VOCAL	M.C.C. María Araceli Nivon Zaghi	
SECRETARIO	M.G.T.I. Leonel Gualberto López Salazar	
1er. SUPLENTE	L.I. Elizabeth Barrera Romero	
2do. SUPLENTE	I.E. María Guadalupe Vázquez Salazar	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127).

LMCF/ntm*



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

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

U. N. A. M.
FACULTAD DE ESTUDIOS

ASUNTO: VOTO APROBATORIO

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Exámenes Profesionales
de la FES Cuautitlán.**

DEPARTAMENTO DE
EXÁMENES PROFESIONALES

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: Trabajo de Tesis

Diseño y Desarrollo de un sistema en línea para la bolsa de trabajo FES Cuautitlán con zend framework 2

Que presenta la pasante: SILVIA JULIÁN MENDOZA

Con número de cuenta: 41110092-0 para obtener el Título de la carrera: Licenciatura 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 HABLARÁ EL ESPÍRITU"

Cuautitlán Izcalli, Méx. a 13 de marzo de 2017.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	M.C.C. Valentín Roldán Vázquez	
VOCAL	M.C.C. María Araceli Nivon Zaghi	
SECRETARIO	M.G.T.I. Leonel Gualberto López Salazar	
1er. SUPLENTE	L.I. Elizabeth Barrera Romero	
2do. SUPLENTE	I.E. María Guadalupe Vázquez Salazar	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127).

LMCF/ntm*

Dedicatoria.

A Dios, por darme el don de la vida y apoyarme y darme lo que se requiere en la vida.

A mis padres, José y Angélica por ser un ejemplo a seguir de vida

Índice

Introducción.....	7
Objetivos.....	9
Hipótesis.....	9
Justificación.....	10
Capítulo 1- Antecedentes de las aplicaciones Web.....	12
1.1- Definición.....	12
1.1.1-Acontecimientos relacionados al desarrollo de las Aplicaciones Web. .	13
1.2- Tecnologías para la programación de aplicaciones Web.....	19
1.2.1- Técnicas de diseño Web.....	20
1.2.2- Tecnologías del lado del cliente.....	22
1.2.2.1- HTML.....	22
1.2.2.2- CSS.....	24
1.2.2.3- JavaScript.....	24
1.2.2.4- jQuery.....	25
1.2.3- Tecnologías del lado del servidor.....	26
1.2.3.1- PHP.....	26
1.3- Bases de datos.....	28
1.3.1- El ciclo de vida del desarrollo del sistema de bases de datos.....	29
1.3.2- Diseño de la base de datos.....	32
1.3.2.1- Diseño conceptual de la base de datos.....	33
1.3.2.2- Diseño lógico de la base de datos.....	34
1.3.2.3- Diseño físico de la base de datos.....	35
1.3.3- Selección SGBD.....	35
1.3.4- Optimización de consultas.....	37
1.3.5- Seguridad de la información.....	38
1.3.6- Diagramas de entidad relación.....	39
1.4- Diagramas UML.....	41
1.4.1- Modelo Vista Controlador (MVC).....	44
1.4.2- Zend Framework 2.....	46
1.4.3- PHTML.....	48
1.4.5- Composer.....	48
1.5- CamelCase.....	49

1.6- Singleton	49
1.7- Pruebas funcionales.	51
Capítulo 2- Bolsa de trabajo FES Cuautitlán.	53
2.1- Misión, Visión y Objetivo.	54
2.2- Organigrama.	55
2.3- Responsabilidades.....	56
2.4- Funcionamiento de la bolsa de trabajo.	57
2.5- Análisis FODA.....	66
Capítulo 3- Análisis, diseño y desarrollo del sistema.....	69
3.1- Análisis y definición de los requerimientos del sistema.	69
3.2- Diseño del sistema.	75
3.3- Modelado del sistema.	80
3.3.1- Diagrama de casos de uso.....	80
3.3.2- Diagrama de actividades.....	85
3.3.3- Diagrama de clases.	102
3.4- Diseño de la base de datos.	103
3.4.1- Diagrama de E-R.	103
3.4.2- Diccionario de datos.....	115
3.5- Modelo Prototipado.....	131
3.6- Programación y depuración.	134
3.6.1- Módulo Inicio.....	134
3.6.2- Módulo Alumno.....	140
3.6.3- Módulo Empresa	144
3.6.4- Módulo Administrador.....	146
3.6.5.- Modelo de clase	148
Conclusiones.	151
Anexo1.Tecnologías usadas	151
Bibliografía.....	155
Referencias electrónicas.	156

Introducción.

En la presente tesis se realiza la propuesta de un sistema informático (SI) para la Bolsa de trabajo de la Facultad de Estudios Superiores Cuautitlán (FESC) ya que hasta el año 2015 no se cuenta con un SI que automatice los procedimientos y que garantice la gestión de almacenamiento de información. La bolsa de trabajo de la FESC tiene como objetivo vincular la comunidad estudiantil y egresados con el sector laboral, además trabaja conjuntamente con la bolsa de trabajo de otras escuelas y facultades de la UNAM. Para el semestre 2016-II la FESC cuenta con una matrícula de 14,526 alumnos y 634 egresados de 17 carreras profesionales de dos campus, para dichos alumnos FESC solo les proporciona una página informativa (http://www.cuautitlan.unam.mx/alumnos/bolsa_trabajo.html), en donde se muestra los beneficios de la bolsa de trabajo, requisitos de inscripción y pasos para registrarse. (Facultad de Estudios Superiores Cuautitlán, 2015)

Hoy en día las comunicaciones y plataformas informáticas tienen un peso vital en todas las empresas debido a las ventajas que estas ofrecen en diferentes áreas tales como producción, administración, toma de decisiones, tiempos y movimientos, etc. Por ello es imprescindible adecuar estas ventajas a las reglas del negocio disminuyendo costos, tiempos y recursos humanos favoreciendo la interacción entre usuario y sistema.

Las ventajas que los usuarios tienen al usar más sistemas son entre otros: realizar operaciones con mayor facilidad, eficacia, de forma intuitiva, con seguridad, menor complejidad y disminuyendo costos evitando transportarse a un lugar para realizar trámites o pagos. Disminuyendo errores, tiempo y recursos superfluos ayudando a incrementar la atención a más personas con efectividad en las operaciones que se realicen. Por tales motivos hay instituciones de educación que están optando por implementar las nuevas tecnologías de información y comunicación (TIC), ya que manejan grandes cantidades de información por semestre.

Por tal motivo, se requiere un método eficiente del servicio de bolsa de trabajo FESC con el cual se pueda atender a la comunidad universitaria, egresados, empresas, instituciones y candidatos a realizar prácticas profesionales. Con el cual se pueda tener información verídica, precisa, significativa, coherente, etc. Para poder generar reportes que no carezcan de ninguna de estas cualidades que son importantes para la institución.

El sistema de informático que se propone para la Bolsa de Trabajo tiene como propósito ayudar a crear procesos más eficientes, organizados, y siempre disponibles para todas las personas que se ven involucradas en el registro de empresas, alumnos y vacantes para ser más eficiente la administración del área de Bolsa de Trabajo.

Dicho sistema implementa diversas secciones donde el administrador de bolsa de trabajo posee un control sobre la base de datos de los alumnos y empresas registradas, así como la ubicación de alumnos en las diversas vacantes ocupadas.

Se pretende tener un mayor control de las instituciones interesadas y de los alumnos registrados por medio de filtros que permitan clasificar sólo a los alumnos de la Facultad de Estudios Superiores Cuautitlán y aquellas instituciones serias que serán verificadas por el administrador del sistema, el cual tiene la posibilidad de aprobar a las empresas después del análisis que el dictaminé. La organización de la tesis se divide en tres capítulos;

En el capítulo uno se abordan los datos contextuales de la investigación: los antecedentes de las aplicaciones Web, tecnología para la programación y base de datos; con la finalidad de presentar una descripción del escenario, de las tecnologías Web y condiciones en que se llevó a cabo este estudio; de tal manera que permita al lector ubicarse en el tema de la investigación y conocer los aspectos que lo rodean al sistema planteado.

En el capítulo dos se analiza la bolsa de trabajo desde la perspectiva de la FES Cuautitlán, cuáles son sus funciones dentro de la institución, misión, visión, objetivos y se genera a partir de la información recolectada un análisis FODA, así como las responsabilidades que tiene esta con la institución.

En el capítulo tres se presenta el análisis, diseño y desarrollo del sistema, la manera en que se fue llevando la investigación para crear el modelado del sistema actual a través de diagramas de flujo con el modelo de negocios. Se muestra la estructura de la base de datos del proyecto a partir de la cual se establece el diseño y la programación. Además, se da a conocer el tipo de programación que se llevó a cabo para el desarrollo del sistema de bolsa de trabajo FES Cuautitlán.

Objetivos.

Generales:

Analizar, diseñar y desarrollar un sistema informático en línea para la bolsa de trabajo de la Facultad de Estudios Superiores Cuautitlán a partir de las necesidades planteadas por el área, que permita proporcionar un servicio eficiente para los usuarios con información confiable y actualizada referente a las ofertas laborales.

Específicos:

- Sistematizar y automatizar los procesos que se llevan a cabo en el área de bolsa de trabajo, tales como la administración y el registro de alumnos y empresas.
- Proporcionar un sistema informático que agilice la incorporación de los alumnos de los diferentes semestres y egresados de la FES Cuautitlán al campo laboral.
- Ofrecer metodologías confiables que permita a las empresas llevar a cabo el reclutamiento de su personal.
- Implementar la estructura del modelo vista controlador de Zend Framework 2 para el diseño y desarrollo del sistema.

Hipótesis.

Hipótesis Nula.

Implementar un sistema informático en línea para la bolsa de trabajo en la Facultad de Estudios Superiores Cuautitlán, no proporcionará eficiencia a los procesos realizados, tampoco incrementará el grado de accesibilidad a las vacantes y no se presenta una mejora significativa en la calidad de la información consistente, confiable y actualizada.

Hipótesis alternativa.

Implementar un sistema informático en línea para la bolsa de trabajo en la Facultad de Estudios Superiores Cuautitlán, proporcionará mayor eficiencia y eficacia a los procesos realizados, además de incrementar el grado de accesibilidad a las vacantes a través de información consistente, confiable y actualizada.

Justificación.

La creación del sistema en línea para la bolsa de trabajo en la FES Cuautitlán tiene como finalidad cubrir las necesidades que han sido planteadas por el área, así como crear nuevas reglas de negocio que eviten la información errónea. Actualmente la bolsa de trabajo cuenta con 2 oficinas en campo 4 en las que se atiende los trámites relacionados al departamento, este cuenta con una sección de página Web en la cual se da una breve información a los alumnos sobre los beneficios que ofrece, los requisitos de inscripción y los pasos a seguir para el registro, esta página se encuentra con el sitio de la facultad con el siguiente enlace: http://www.cuautitlan.unam.mx/alumnos/bolsa_trabajo.html .Este sitio carece de funcionalidades

Con la tecnología implementada se proporcionará a la comunidad estudiantil un sitio Web exclusivo para su información académica y laboral donde podrán encontrar información precisa y oportuna sobre las empresas y sus vacantes disponibles, incorporándolos al mundo laboral de una manera más ágil y rápida.

El sistema en línea permitirá a los usuarios tener acceso a información actualizada y confiable de las empresas que se encuentran interesadas en contratar alumnos de la FES Cuautitlán, además podrán visualizar las propuestas de trabajo de acuerdo a su formación profesional, teniendo la oportunidad de visualizar las vacantes que fueron publicadas para su carrera que ha cursado el candidato de tal manera que cubra sus expectativas. El alumno podrá difundir sus datos personales y su perfil profesional para incrementar la posibilidad de contratación de no ser así el alumno solo visualizará las vacantes que sean aprobadas por el administrador y que estén vigentes. Se pretende eliminar que el alumno llene formatos tediosos y que tenga que acudir a la oficina de bolsa de trabajo a solicitar información acerca de nuevas vacantes disponibles.

Las empresas tendrán un sistema disponible en cualquier momento para poder publicar sus vacantes disponibles que vayan acorde a los perfiles profesionales que forma la facultad. Se brindará a las empresas la oportunidad de visualizar información específica de los alumnos que cumplan tengan un perfil público también se pretende aumentar el número de empresas interesadas en contratar alumnos de la FES Cuautitlán y eliminar la espera innecesaria de elaboración y difusión de propaganda dentro de la Institución educativa.

La información proporcionada por los usuarios se salvaguardará en todo momento para que no haya pérdidas de información, respetando la integridad de la misma y solo se difundirá con previa autorización del usuario. La gestión de la información quedará a cargo del administrador, a quien se le desarrollará un módulo específico donde podrá realizar estadísticas, visualizar datos específicos de cada usuario y es quien se encargará que los datos proporcionados de las empresas sean verídicos.

El sistema en línea para la bolsa de trabajo pretende ser el medio principal en donde los alumnos y empresas puedan interactuar de manera confiable y contribuir a la automatización de los procesos que se llevan a cabo dentro de la FES Cuautitlán.

Capítulo 1- Antecedentes de las aplicaciones Web.

1.1- Definición.

Una aplicación nos permite realizar un conjunto de tareas o actividades específicas a través de una interfaz, la cual proporciona herramientas que hacen más sencilla su utilización. Las aplicaciones en informática son programas que cumplen ciertas expectativas y se ejecutan dentro de un contenedor, el cual se encarga de llevar a cabo su administración, un ejemplo de contenedor sería los sistemas operativos. Las aplicaciones Web son programas que permiten difundir información o proporcionar un servicio a través de la Web, que es el contenedor encargado de permitir la ejecución de dichas aplicaciones. La Web es solo un servicio que proporciona Internet el cual se define como una gran red mundial de ordenadores que engloba a las tecnologías que hacen posible que dichos ordenadores ubicados en los diferentes lugares del mundo se conecten entre sí y puedan compartir información a través de las aplicaciones Web. (David Megías Jiménez, 2004).

Una aplicación Web puede definirse como una extensión dinámica de un servidor Web o un servidor de aplicaciones. (David Roldán Martínez, 2010). Sin embargo, en la ingeniería del software se denomina aplicación Web al software que los usuarios pueden utilizar accediendo a un servidor Web, bien vía internet, extranet o intranet, utilizando para ello un navegador Web. (Miguel, 2013). En general podemos decir que una aplicación Web se encuentra inmersa dentro de un servidor Web que es el encargado de llevar a cabo su ejecución cuando es solicitado por el usuario a través de un navegador Web. Las aplicaciones Web tienen dos objetivos la de comunicar información específica y la de brindar un servicio que interactúe con el usuario. Existen dos tipos de aplicaciones Web las estáticas y las dinámicas, las estáticas son aquellas que solo permiten mostrar información específica y las aplicaciones dinámicas responden a peticiones que el usuario solicita en tiempo real, la diferencia entre cada tipo de aplicaciones, así como los lenguajes de programación se abordará en el capítulo Tecnologías para la programación de aplicaciones Web.

1.1.1-Acontecimientos relacionados al desarrollo de las Aplicaciones Web.

La Web fue creada en 1989 por Tim Berners Lee trabajador de CERN (Organización Europea para la Investigación Nuclear) y consistía en una forma de organizar la información usando como medio físico de comunicación la red Internet y el protocolo HTTP (Hypertext Transference Protocol) (Alicia Ramos Martín, 2011). Este acontecimiento abrió la posibilidad de compartir información entre equipos heterogéneos a través de redes informáticas.

En un principio la Web fue diseñada para suministrar un conjunto de páginas estáticas en donde el usuario solo podía visualizar determinada información, su diseño era simple ya que no se contaba con estándares ni con un marco de trabajo que normalizaran su contenido y diseño. Su funcionalidad consistía en el envío de consultas o peticiones mediante de un navegador el cual se comunicaba con el servidor Web a través del protocolo de transferencia de hipertexto (HTTP) permitiendo extraer los archivos que se encontraban almacenados en el servidor y devolviendo como respuesta la visualización de la página solicitada.

En el año de 1991 se publicó el primer documento que llevó por nombre HTML Tags creado por el físico Tim Berners-Lee, quien propuso el sistema de hipertexto para compartir información, en dicho documento describía la estructura de cómo elaborar una página Web a través del lenguaje de etiquetas, implementado en años posteriores pero dado a conocer de manera oficial hasta el año de 1993.

Sin embargo, lo que devuelve el servidor Web no tiene por qué ser siempre una página estática, sino que puede tratarse de la salida de un programa o aplicación que muestre información específica solicitada por el usuario. Para poder hacer posible que las aplicaciones se convirtieran en dinámicas fue necesario desarrollar nuevos métodos que permitieran al servidor Web responder a dichas peticiones. Para ello se desarrolló el método conocido como CGI (Common Gateway Interface) que definía un mecanismo a través del cual se podía pasar información entre el servidor HTTP y programas externos. Pero los CGI tenían un inconveniente, ya que la mayoría de los programas se encontraban escritos en algún lenguaje interpretado (Perl, Python, etc.) los cuales requerían de un entorno en tiempo de ejecución (run-time environment) como Java o Visual Basic, esto producía una gran carga para la máquina del servidor provocando

problemas de rendimiento. Para poder solucionar este problema se desarrollaron dos alternativas la primera fue diseñar un sistema de ejecución con módulos más integrados con el servidor, que evitará que este instanciará y ejecutará multitud de programas. La segunda alternativa consistió en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que permitiera incluir las páginas en el código de manera de que el servidor fuera quien la ejecutará, reduciendo así el tiempo de respuesta. (David Megías Jiménez, 2004)

El desarrollo de nuevas aplicaciones Web que permitían este dinamismo significo para las organizaciones la oportunidad de ofrecer sus servicios o productos, rompiendo barreras geográficas y de comunicaciones, abriendo enormes posibilidades al crecimiento. La utilización de nuevas tecnologías para el desarrollo de aplicaciones Web ha permitido agilizar los procesos, mejorar la productividad y aumentar la eficiencia, además de abrir nuevos horizontes de negocio en el mercado global que proporciona Internet.

A partir de entonces se ha ido implementado y desarrollando múltiples aplicaciones y tecnologías que han permitido el crecimiento de la Web. Las aplicaciones Web interactivas poco a poco han revolucionado la forma de utilizar internet, aumentando el contenido de las páginas con texto estático basado solo en HTML a un contenido informativo e interactivo, por lo tanto, escalable, creando en el servicio Web la necesidad de incorporar nuevos lenguajes de programación que permitieran a las aplicaciones Web, ofrecer el dinamismo exigido por los usuarios.

Inicialmente en la Web 1.0 solo se podían apreciar páginas estáticas y el usuario no podía interactuar con ellas. El contenido y el diseño se encontraban juntos con el código HTML lo que hacía difícil su modificación, no existía normas de diseño por lo que existían colores estrafalarios y las paginas creaban desconcierto a los usuarios. Uno de los navegadores existentes eran Yahoo! y AltaVista que eran navegadores muy complicados de manejar.

Posteriormente en la Web 2.0 se comenzó el uso de bases de datos, aparecieron las redes sociales en donde los usuarios pudieron crear perfiles e insertar comentarios para poder difundirlos, logrando almacenar y compartir información de una forma más sencilla. (Alicia Ramos Martín, 2011).

Desde la aparición de las primeras aplicaciones Web hasta la actualidad, han surgido acontecimientos muy importantes que han permitido revolucionar la forma de implementar y desarrollar las aplicaciones la Web, entre las aplicaciones Web más importantes podemos mencionar:

En el año de 1995 se crea Yahoo!, por Jerry Yang y David Filo, como una guía de contenidos en Internet, nace el lenguaje Java, primer lenguaje pensado para integrar dinámicamente programas en las páginas Web (los programas Java que integran esas páginas se denominaron applets), el programador Rasmus Lerdorf puso a disposición el lenguaje PHP que se convirtió en la estructura base para el desarrollo de aplicaciones Web, Netscape el navegador Web más popular, incorporó una nueva tecnología llamada JavaScript, permitiendo a las aplicaciones Web ser más interactivas con el usuario dándole un nuevo enfoque al desarrollo de las aplicaciones Web, esa tecnología fue implementada por grandes organizaciones como Google y Microsoft.

Surge Hotmail en el año de 1996 por los desarrolladores, Sabeer Bhatia y Jack Smith, Hotmail permitió por primera vez al público en general acceder y consultar el correo electrónico.

En 1997 apareció Flash utilizada para añadir contenido interactivo en los sitios Web, convirtiéndose en una plataforma para desarrollar aplicaciones Web interactivas.

En el año 2004 en una conferencia de la Web 2.0 a cargo de John Battelle y Tim O'Reilly, el concepto de "Web como plataforma" fue mencionado por primera vez, dándole un nuevo enfoque de las aplicaciones Web en donde se aprovecha las ventajas de la conexión a internet y que se desvían del uso tradicional del escritorio. En ese mismo año surge Facebook que se convirtió en el medio de comunicación más utilizado en todo el mundo.

En el 2005, YouTube fue lanzado oficialmente, permitiendo a los usuarios compartir videos en línea y finalmente en el año 2006 Twitter fue puesto en marcha. (Asensio, 2012).

1.1.2- Ingeniería Web.

La ingeniería del software se encarga de analizar, diseñar y organizar aquellos elementos de un sistema que se convierte en un producto, servicio o tecnología para el control o transformación de la información, teniendo como objetivo identificar, analizar, especificar, modelar, validar y gestionar los requisitos operacionales del sistema a desarrollar.

La ingeniería Web incorpora modelos de procesos especializados y aspectos fundamentales de la ingeniería del software adaptados a las características de desarrollo de sistemas y aplicaciones basados en Web. La ingeniería Web implementa procesos, métodos y tecnologías (herramientas) enfocados a la calidad del software.

Para el desarrollo de una aplicación Web se debe construir primero un enfoque genérico de los requerimientos del sistema, aplicando fundamentos de la ingeniería del software como los métodos y técnicas que permiten crear la estructura de sistema, para esto es necesario identificar los objetivos y requerimientos operacionales más detallados de la información. Para formular un concepto general se debe identificar y definir los elementos del sistema, comenzando con un entendimiento claro del contexto y después de manera progresiva el enfoque se delimita hasta la comprensión los detalles técnicos. La figura 1.1 muestra la jerarquía de la ingeniería de sistemas que abarca una serie de métodos para navegar de arriba hacia abajo y de abajo hacia arriba, que permiten delimitar cada elemento que integra el sistema.

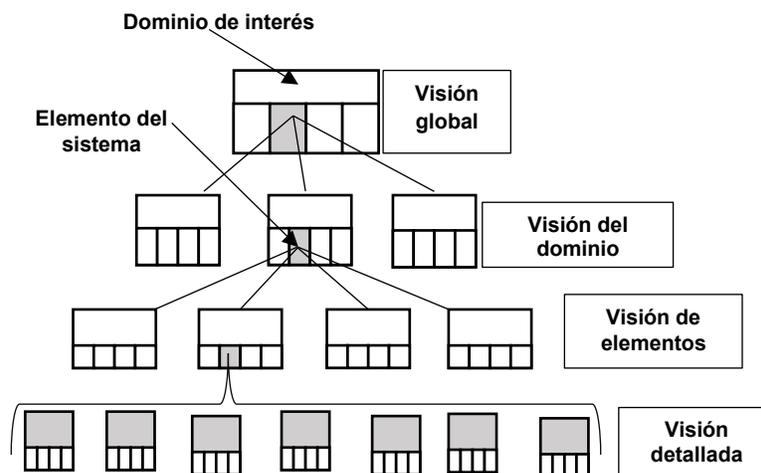


Figura 1.1- La jerarquía de la ingeniería de sistemas permite delimitar los elementos del sistema generando un enfoque

La jerarquía de la ingeniería de sistemas comienza teniendo una visualización global del sistema en donde se conocen los dominios con los que se conformará el sistema y se delimita el dominio de interés, de acuerdo al dominio de interés se analizan las necesidades de los elementos del sistema por ejemplo la información, hardware, software, etc. y después de manera progresiva se lleva a cabo el análisis, diseño y construcción de los elementos deseados de forma detallada. (Roger S. Pressman, 2010)

Procesos.

La implementación de procesos en la ingeniería Web define un marco de trabajo y forma la base para la gestión del proyecto en donde se identifican las actividades principales que se llevarán a cabo durante el desarrollo del sistema. La mayor parte de los modelos de procesos del software se aplican en la ingeniería Web y se basan en uno de los tres modelos generales o paradigmas de desarrollo de software los cuales son el enfoque en cascada, el desarrollo iterativo y la ingeniería del software basada en componentes. El proceso de ingeniería Web adopta el enfoque de desarrollo ágil implementado un proceso genérico del marco de trabajo que se adapta a las necesidades de cada proyecto. En base a un marco de trabajo genérico de ingeniería del software se establece el proceso de ingeniería Web que se muestra en la figura 1.2, el cual se basa en la Garantía de Calidad de Software (SQA) y en la Gestión de Configuración de Software (SCM) los cuales garantizan un buen desarrollo del proyecto y la escalabilidad del software. Siendo un marco de trabajo genérico implementa un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar sin necesidad de volver a resolver el problema. Siendo estos los principales que lo convierten en una de las mejores opciones para el desarrollo de la ingeniería del Web con respecto a los ya mencionados.

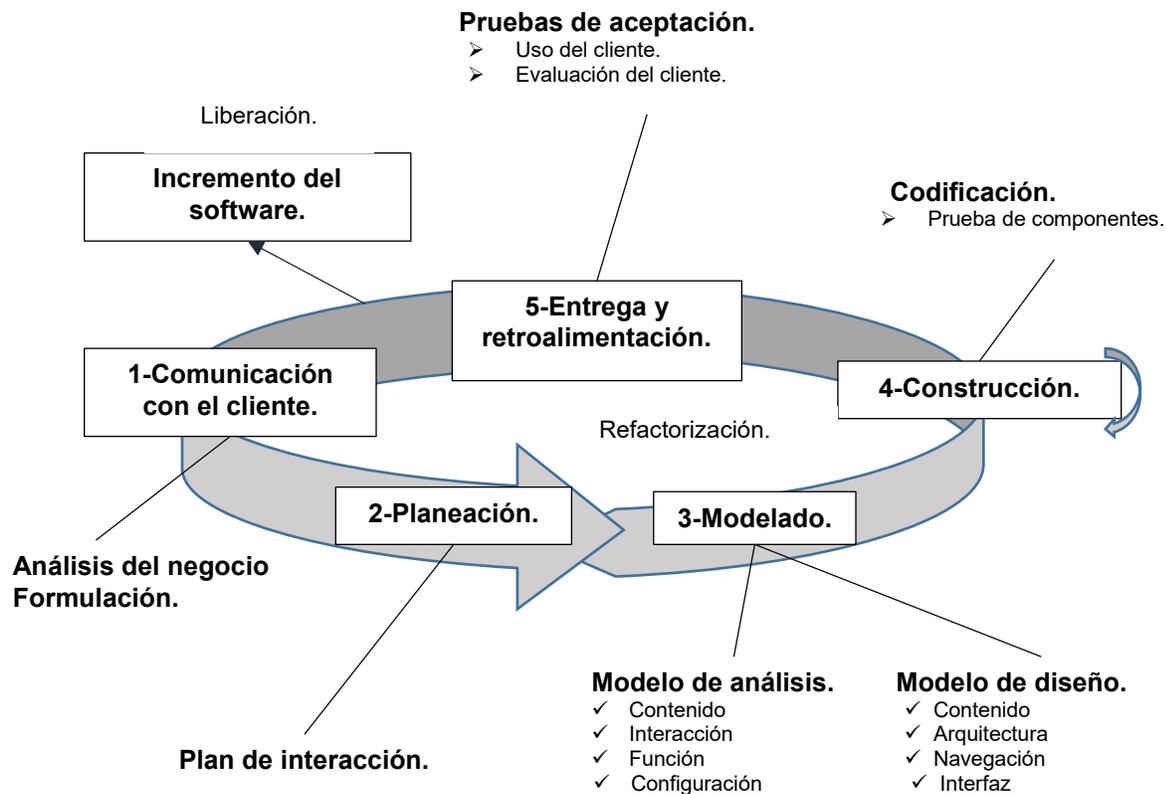


Figura 1.2-El proceso de ingeniería Web se basa en un marco de trabajo en el que se define el conjunto de actividades que se adoptan a las necesidades del proyecto. (Roger S. Pressman, 2010).

Métodos.

Los métodos es el conjunto de labores técnicas que permiten comprender, caracterizar y construir el proyecto, entre los métodos que se implementan para el desarrollo de aplicaciones Web se encuentran.

- Métodos de comunicación. Define la comunicación entre los integrantes del proyecto, así como los usuarios que manipulan la información, este tipo técnicas se utiliza durante la recopilación de información
- Método de análisis de requisitos. Estable el contenido que se mostrará en la interfaz, así como la funcionalidad de los módulos con los que contará la aplicación Web.
- Método de diseño. Comprende la aplicación de técnicas de diseño para establecer el contenido, la aplicación y la arquitectura de la información, así como la estructura de navegación y la interfaz.

- Método de prueba. Incorpora la revisión de técnicas formales del contenido, diseño y de la funcionalidad de la aplicación Web que garanticen la calidad y la seguridad. (Roger S. Pressman, 2010)

Herramientas y tecnologías.

La implementación de herramientas y tecnologías se refiere al conjunto de elementos que se integran a las aplicaciones Web de acuerdo a las necesidades del desarrollo del proyecto y generalmente son herramientas multimedia, de audio, de conectividad a bases de datos, de seguridad, referentes a los tipos de navegadores, de servidores, análisis de sitio, etc.

1.2- Tecnologías para la programación de aplicaciones Web.

La forma de estructurar los elementos gráficos que se encontrarán inmersos, así como las funcionalidades que se implementarán en una aplicación Web deberán encontrarse orientados a las necesidades de la organización, a los requerimientos solicitados y al público al que se encuentra dirigido los servicios. Existen diferentes tecnologías vinculadas al desarrollo de las aplicaciones Web, las relacionadas al diseño del front-end en donde se implementarán tecnologías del lado del cliente y las relacionadas al back-end donde se incorporan tecnologías del lado del servidor que permiten la funcionalidad de las aplicaciones.

Sin embargo, para que se establezca un servicio en donde un cliente solicite peticiones al servidor y obtenga respuesta de este, es necesario de establecer una comunicación entre las tecnologías del lado del cliente y del servidor. La arquitectura cliente servidor es un modelo en donde el servidor gestiona y procesa un conjunto de instrucciones que recibe del cliente y devuelve como respuesta una acción, la cual es reflejada hacia el cliente como una funcionalidad de la aplicación, el cliente es el medio encargado de

interactuar con el usuario. La figura 1.3, ilustra la arquitectura cliente servidor y el medio de comunicación.

1.2.1- Técnicas de diseño Web.

El diseño Web establece la vista donde se distribuyen de manera estética los componentes y es donde se plantea la navegación que tendrá la interfaz. Los elementos visuales, como el color, el diseño, la tipografía y las imágenes, forman la personalidad de cada sitio y provocan un impacto emocional que determina el éxito de cada sitio Web.

Entre las características esenciales para el diseño de un sitio Web se encuentran los siguientes componentes:

- **Fácil de mantener:** Debe de poseer flexibilidad y capacidad de cambio de tamaño, para que las actualizaciones habituales del sitio, durante toda su vida, pueda

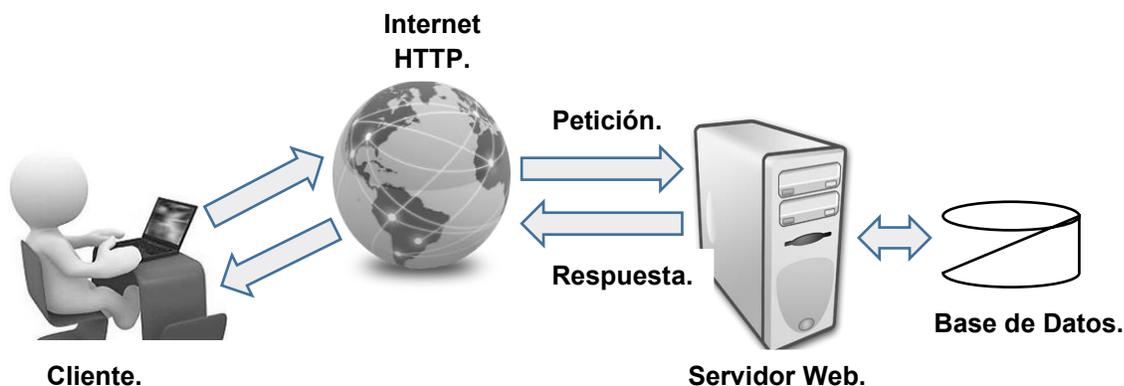


Figura 1.3- Arquitectura cliente servidor

llevarse a cabo invirtiendo un esfuerzo y tiempo mínimos.

- **Atractivo visualmente:** debe ser una experiencia atractiva e interesante para su público objetivo.
- **Fácil de utilizar:** Es decir, el manejo sencillo, efectiva y que se descargue rápidamente. Los visitantes han de poder encontrar fácilmente lo que están buscando, para conseguir sus propósitos sin perder tiempo.
- **Técnicamente sólido:** Debe de comportarse como esperamos, sin que haya vínculos rotos. (McIntire, 2008)

Para diseñar un sitio Web deberá tomarse en cuenta que en ciertos casos no se puede tener el control del diseño ya que depende de la forma en que el usuario configure su ordenador. Entre en los aspectos a considerar son los siguientes:

- Los colores y las fuentes se ven de forma distinta en los diferentes navegadores, monitores y sistemas.
- No podemos garantizar que los elementos de una página se muestren con un tamaño determinado, ya que los diferentes sistemas tienen resoluciones distintas.
- Muchos ordenadores recién salidos de fábrica no están configurados para mostrar sus mejores ajustes. Los fabricantes suelen configurar los ordenadores por debajo de sus capacidades máximas. (McIntire, 2008).

Otro de los componentes a considerar para el diseño de una aplicación Web es la jerarquía visual de los elementos que se mostraran en cada vista, ya que a través de la jerarquía visual se determina la forma en la que será leída la página. La figura 1.4 ejemplifica la jerarquía de importancia de cada elemento.



Figura 1.4- La jerarquía visual normalmente se lleva a cabo de izquierda a derecha y de arriba abajo. El tamaño, el contraste y el color modifica la importancia de los elementos, es por eso que lo elementos de mayor importancia se sitúan en la parte superior izquierda. (Campos, 2005)

1.2.2- Tecnologías del lado del cliente.

En un entorno Web, las tecnologías del lado cliente se refiere al conjunto de programas que permite el diseño y el desarrollo del front-end de una aplicación Web. El navegador es el medio que permite la ejecución de dichas tecnologías para interactuar con el usuario. Existen numerosos navegadores y versiones de navegadores en uso, lo que significa que muchos elementos se interpretarán de forma distinta en cada navegador o incluso en las diferentes versiones de un mismo navegador. Por lo tanto, las tecnologías de lado del cliente se encuentran estrechamente ligadas con el navegador que se implemente, esto significa que se podrá manejar navegadores y plataformas específicos para la ejecución de la aplicación Web, los cuales se encontrarán limpios de errores, aunque podrían aceptarse que algunas funcionalidades no se mostrasen en algunos casos, pero de tal forma que el usuario nunca llegase a notar que falta algo. Se deberá probar las aplicaciones Web en múltiples navegadores y en diferentes plataformas con el propósito de disminuir el número de errores que puedan presentarse durante la ejecución.

Entre las tecnologías utilizadas por los navegadores para mostrar una aplicación Web se encuentra el lenguaje de maquetación de hipertexto HTML, hoja de estilos en cascada CSS, el lenguaje JavaScript, jQuery y en algunos casos Flash y multimedia para el audio y vídeo, etc. estas tecnologías permiten desarrollar y diseñar del front-end de una aplicación Web por lo que su implementación dependerá del criterio de cada desarrollador.

1.2.2.1- HTML.

HTML (Hypertext Markup Language, Lenguaje de maquetación de hipertexto) es un lenguaje basado en etiquetas utilizado por navegadores Web. Su origen se remonta a 1980, cuando el físico Tim Berners-Lee, trabajador de CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de hipertexto para compartir documentos en internet. El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre "HTML Tags" etiquetas HTML. (McIntire, 2008)

El lenguaje de etiquetas que establece HTML propone un conjunto de etiquetas para organizar el contenido de una página Web, estableciendo una estructura semántica que especifica la forma correcta de construir una página Web. En la figura 1.5 se muestra la estructura semántica que maneja HTML5.

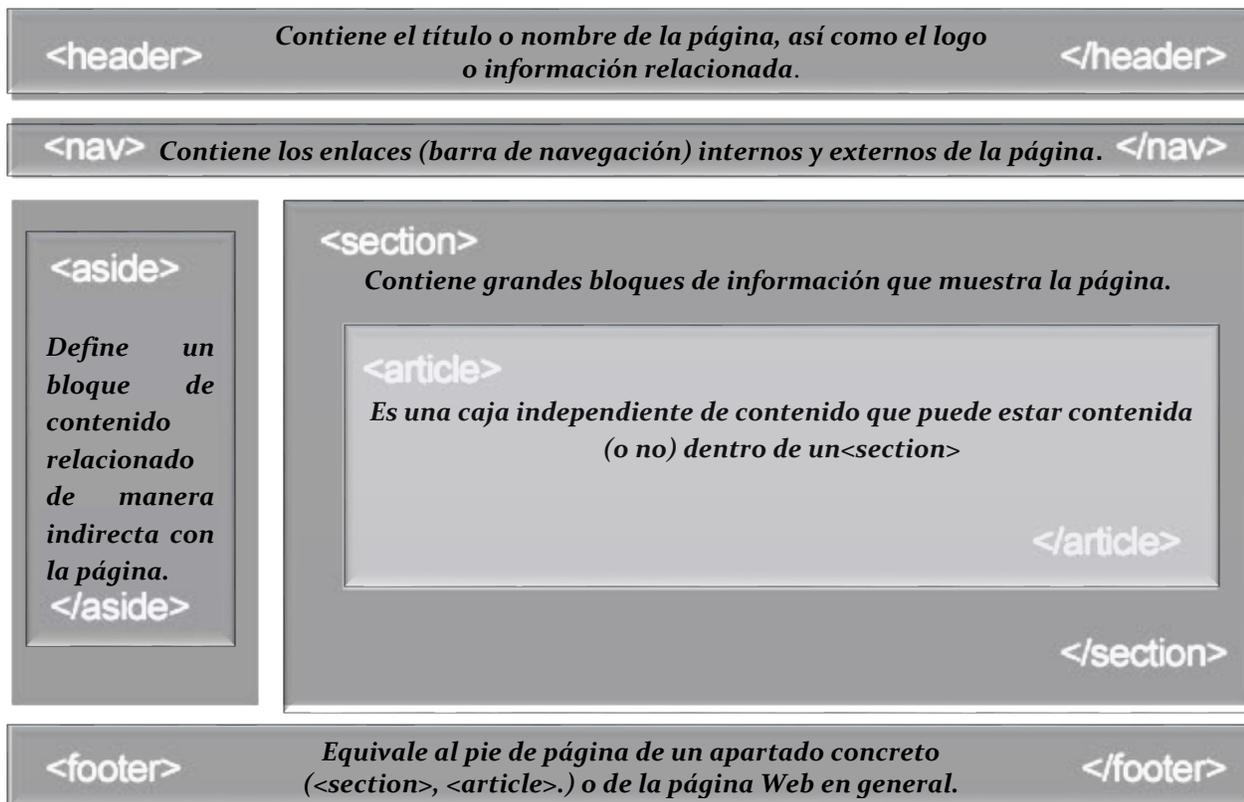


Figura 1.5- Un documento HTML es un conjunto de etiquetas que generalmente funcionan en pares y pueden trabajar de manera anidada. Las etiquetas que se muestran en la figura son propiamente de HTML5 ya que cada versión de HTML maneja diferentes elementos.

HTML5 es el resultado de las diferentes adaptaciones que ha tenido HTML, sus actualizaciones han ido incorporando nuevas tecnologías de desarrollo como CSS3, JavaScript, jQuery, etc. que han permitido crear aplicaciones Web con un concepto de diseño diferente. Existen diversas limitaciones que hacen a HTML solo un lenguaje puramente descriptivo que solo permite definir páginas Web estáticas, por lo que no es considerado como un lenguaje de programación ya que no permite generar estructuras iterativas o condicionales, definir funciones que sean utilizadas en distintos puntos del documento, declarar variables, realizar cálculos matemáticos, etc. Los únicos elementos de HTML que podrían de alguna forma considerarse interactivos son los formularios a través de los cuales se solicita información al usuario. (Ángel Cobo, 2005).

1.2.2.2- CSS.

Hojas de Estilo en Cascada (Cascading Style Sheets) es un lenguaje de hojas de estilos que surge para superar las limitaciones y satisfacer los nuevos requerimientos de los diseñadores que buscaban mejorar la presentación de los documentos mediante estándares de diseño y estilos reduciendo el uso de código JavaScript, ya que este prolongaba el tiempo que el navegador tardaba en ejecutar una página Web, e implementaron nuevas formas y movimientos con CSS. Sin embargo, el motor de los diferentes navegadores que se encarga de interpretar el lenguaje HTML y CSS limita el diseño de los documentos ya que cada navegador cuenta con diferentes componentes. La finalidad de CSS es de crear unos estilos físicos, separados de las etiquetas HTML (en lugar de aparecer como atributos de éstas), y aplicarlas en bloques de texto o cualquier elemento que se desee cambiar al estilo. Esto facilita el diseño y la revisión de páginas, ya que se puede variar la presentación de una página, o de todo el conjunto de ellas, sin cambiar una sola línea de código HTML.

CSS integra su lenguaje de estilos al documento HTML de dos formas, la primera es como código embebido que consiste en agregar el grupo de los estilos en la cabecera del documento o en la línea donde se encuentra el elemento y la segunda forma es a través de archivos externos, la cual es la forma recomendable porque cumple la finalidad del lenguaje CSS de crear estilos en forma separada del documento HTML. CSS maneja su propia sintaxis y permite dar estilos al elemento a través de diversas maneras.

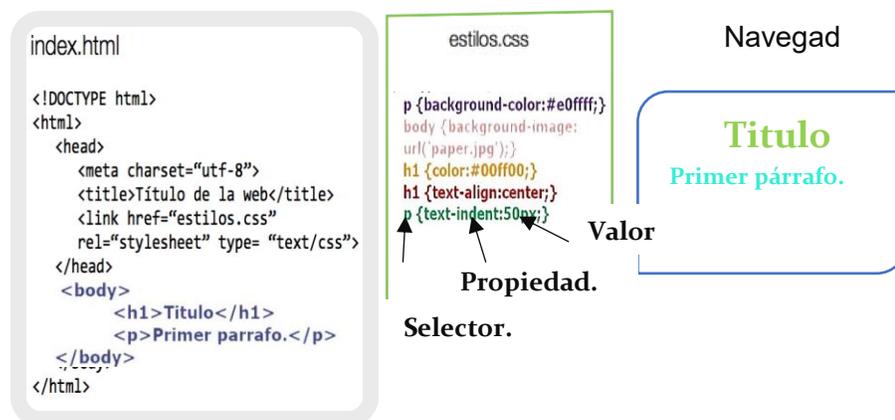


Figura 1.6 CSS permite definir los estilos a través de diferentes selectores (tipo, id, clase) que se definen en el documento HTML.

1.2.2.3- JavaScript.

JavaScript se presenta como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de internet que fue creado por los desarrolladores de Netscape Communications quienes incorporaron el lenguaje Live Script para solucionar la falta de interactividad que le hacía falta a las páginas Web, poco tiempo después Sun Microsystems (creador del lenguaje Java) se unió con Netscape para conseguir que Live Script fuese adaptado como lenguaje estándar de internet, el cual por su semejanza al lenguaje Java fue renombrado a JavaScript. (Maza, 2001)

JavaScript es un lenguaje compacto que permite personalizarse a través de guiones cortos llamados snippets del programa, estos son mini programas que no se compilan, sino que se interpretan línea por línea mientras que la aplicación sigue en ejecución, la sintaxis es similar C, C++, Pascal o Delphi y se encuentra basada en un lenguaje orientada a objetos, JavaScript maneja un sin número de eventos que se ejecutan durante el tiempo que el usuario visualiza la página y se ve limitado por la versión y el tipo de navegador que ejecute el script.

El código JavaScript puede insertarse de la misma manera que lo hace el lenguaje CSS ya sea incluido o embebido en el documento o como archivo externo con extensión .js. Se debe insertar el link del archivo o el código que se va ejecutar en la cabecera del documento HTML, de esta manera se cargará el script antes de ser llamado dentro del cuerpo del documento.

JavaScript presentó diversas limitaciones entre las cuales no podía comunicarse con recursos que no pertenecieran al mismo dominio, no se podía cerrar ventanas que no se hayan abierto con el mismo script, no se podía acceder a los archivos del ordenador del usuario y el tiempo de ejecución era demasiado. Sin embargo, la incorporación de AJAX (Asynchronous JavaScript And XML, JavaScript asíncrono y XML) ha resuelto ciertas limitaciones dándole un nuevo enfoque a la programación con JavaScript.

1.2.2.4- jQuery.

jQuery es una biblioteca de JavaScript cross-browser desarrollada para simplificar los scripts client-side que interactúan con HTML. Esta fue lanzada en enero 2006 en el Bar Camp de Nueva York por John Resig. jQuery es una biblioteca de código abierto y posee licencia dual, haciendo uso de la licencia MIT (Massachusetts Institute of Technology) o GNU (General Public License) versión 2.4. (Castillo A. A., 2015)

jQuery consiste en una serie de funciones y métodos de JavaScript que mejoran el desarrollo de aplicaciones Web ya que muestran mayor compatibilidad con los navegadores además permite crear animaciones, manipular eventos y desarrollar aplicaciones AJAX y DOM. Los desarrolladores pueden crear capas de abstracción para interacciones de bajo nivel, simplificando el desarrollo de las aplicaciones Web dinámicas de gran complejidad, además por ser de código abierto existen gran variedad de plugins que pueden ser implementadas libremente y que reducen gran cantidad de código.

jQuery:	<pre>\$(document).ready(function() { \$('#clickedElement').click(function() { \$('#fadedElement').fadeOut(); }); });</pre>
Javascript:	<pre>function fadeThisElement(elm) { for (var i=10; i>0; i--) { var opacity = i/10; setTimeout(function(opacity) { elm.setStyle("-moz-opacity", opacity); elm.setStyle("opacity", opacity); elm.setStyle("filter", "alpha(opacity=" + (opacity*100).toString()); }, 100); } } window.onload = function() { document.getElementById("clickedElement").onclick = function() { fadeThisElement(document.getElementById("fadedElement")); } }</pre>

Figura 1.7 jQuery permite reducir el número de líneas de código obteniendo la misma funcionalidad y con un mayor rendimiento a comparación de JavaScript.

1.2.3- Tecnologías del lado del servidor.

Se entiende como tecnologías del lado servidor al conjunto de lenguajes de programación que son interpretados por un servidor, el cual lleva a cabo el procesamiento de peticiones que se realizan del lado del cliente. Existen diferentes necesidades de recursos a la hora de ejecutarse: velocidad de procesador, memoria, velocidad y capacidad de los discos duros, dispositivos de entrada/ salida, etc. Este tipo de tecnologías permite a las aplicaciones Web tener un dinamismo porque tienen acceso a información centralizada que se encuentra almacenada una base de datos utilizando programas de aplicación escritos en lenguajes de programación como PHP, ASP, CGI/Perl, ColdFusion, JSP (JavaServerPages).

1.2.3.1- PHP.

PHP o Hypertext Pre Processor, conocido en sus inicios como PHP Tools o Profesional, fue creado en 1994 por Rasmus Lerdorf e implementado actualmente por The PHP Group, responsable del desarrollo vigente. Este programa se encuentra licenciado bajo la PHP License, lo que lo hace un software libre y de código abierto. (Biaggini, 2013)

PHP es un lenguaje de programación procedural con una sintaxis similar a la del lenguaje C y en sus últimas versiones hace uso de programación orientada a objetos similar a la de Java y su implementación se lleva a cabo de una forma muy sencilla, ofrece mejor rendimiento en tiempo de ejecución, contiene un conjunto muy amplio de funciones creadas por desarrolladores de forma libre, tiene soporte para conectarse a una gran variedad de bases de datos como MySQL, PostgreSQL, Oracle, dBM, FilePro, HyperWave, Informix, InterBase, Sybase entre otras y además es posible realizar aplicaciones con interfaz gráfica mediante librerías y extensiones como QT y GTK. La forma en la que se lleva a cabo el intercambio de peticiones se ilustra en la figura 1.8.

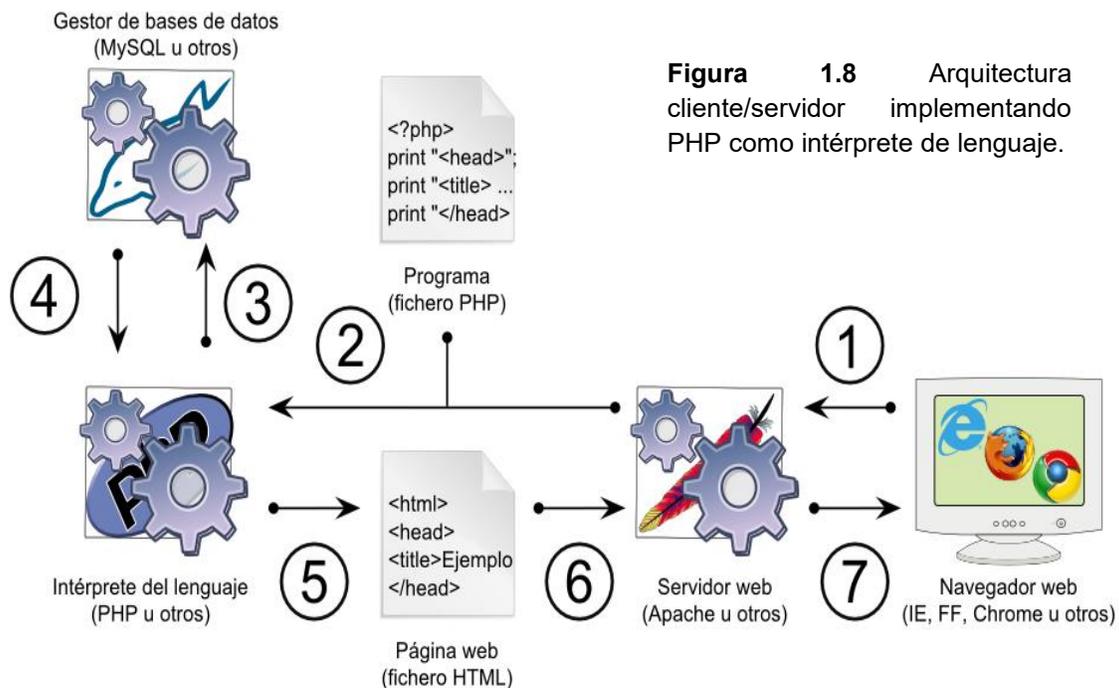


Figura 1.8 Arquitectura cliente/servidor implementando PHP como intérprete de lenguaje.

1. El cliente envía una petición de una página a un servidor Web. Esta solicitud suele incluir variables que le proporcionan al servidor información adicional sobre la petición.
2. El servidor Web pasa el control a un servidor de aplicación (el software y el ordenador en que se ejecuta), indicándole que ejecute el programa de aplicación

adecuado. En este caso se trata de un programa de PHP y lo envía al intérprete del lenguaje.

3. En muchos casos el programa de aplicación extrae y/o actualiza registros de una base de datos.
4. La base de datos entrega al intérprete el resultado de las consultas.
5. El intérprete del lenguaje completa la ejecución del programa. El servidor de aplicación le pasa el control de nuevo al servidor Web, junto con toda la información que necesite éste (como registros de la base de datos) para ensamblar la página de respuesta.
6. EL servidor Web combina los datos del problema de aplicación con el archivo HTML acuerdo y luego envía el nuevo archivo HTML creado, junto con todos los archivos de imágenes, CSS y JavaScript asociados, al cliente.
7. El navegador Web del cliente combina los archivos recibidos y muestra la página completa.
8. El servidor Web se desconecta del cliente cualquier solicitud futura del cliente se tratará independientemente de las peticiones previas (McIntire, 2008)

1.3- Bases de datos.

Un sistema de gestión de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada Bases de Datos contiene información relativa para una organización. (Abraham Silberschatz, 2002). Los SGBD actúan como interfaz entre los programas de aplicación y el sistema operativo teniendo como objetivo principal proporcionar un entorno eficiente a la hora de almacenar y recuperar información. La gestión implica tanto la definición de estructuras para almacenar información como para la provisión de mecanismos para la manipulación de la información.

El diseño de un entorno de aplicaciones de bases de datos, ha ido sustituyendo de manera gradual a los sistemas basados en archivos como parte de la infraestructura de sistemas de información de las organizaciones. Los sistemas de información son recursos que permiten la recopilación, gestión, control y diseminación de la información de la organización, de modo que un sistema de información incluye la base de datos, el software de bases de datos, el software de aplicación, el hardware informático y el

personal que utiliza y desarrolla el sistema. Las bases de datos es un componente fundamental de los sistemas de información, su desarrollo y utilización deben contemplarse desde la perspectiva de los requisitos globales de la organización. (Thomas M. Connolly, 2005)

1.3.1- El ciclo de vida del desarrollo del sistema de bases de datos.

Los sistemas de bases de datos constituyen la estructura principal y fundamental para la organización ya que contienen grandes volúmenes de información y su correcta gestión permitirá a la organización una escalabilidad. Las bases de datos cumplen un ciclo de vida que contiene diferentes etapas que no son estrictamente secuenciales, sino que existe una cierta repetición de las etapas anteriores a través de los que se denominan bucles de realimentación. (Thomas M. Connolly, 2005).

Etapa.	Actividades principales.
Planificación de las bases de datos.	Planificación del modo en que pueden llevarse a cabo las distintas etapas del ciclo de vida de la forma más eficiente y efectiva.
Definición del sistema.	Especificación del ámbito y los límites del sistema de base de datos, incluyendo las principales vistas de usuario, los tipos de usuario y áreas de aplicación.
Recopilación y análisis de requerimientos	Recopilación y análisis de los requerimientos del nuevo sistema de base de datos.
Diseño de las bases de datos.	Diseño conceptual lógico y físico de la base de datos.
Selección del SGBD.	Selección de un SGBD adecuado para el sistema de base de datos.
Diseño de la aplicación.	Diseño de la interfaz de usuario y los programas de aplicación que sirvan para utilizar y procesar los datos de la base de datos.
Prototipado.	Construcción de un modelo funcional del sistema de base de datos que permitan a los diseñadores o usuarios visualizar y evaluar el aspecto y la función del sistema final.
Implementación	Creación de las definiciones físicas de la base de datos y de los programas de aplicación.
Conversión y carga de datos.	Carga de los datos del antiguo sistema en el nuevo y es posible. Conversión de sistemas existentes para que se ejecuten sobre la nueva base de datos.
Pruebas.	Prueba de las bases de datos en busca de errores y validación de la misma con respecto a los requisitos especificados por el usuario.
Mantenimiento operativo.	El sistema de base de datos está completamente implementado, después del cual se monitoriza y mantiene de manera continua.

Cuadro 1.1 El ciclo de vida del desarrollo del sistema de bases de datos. (Thomas M. Connolly, 2005)

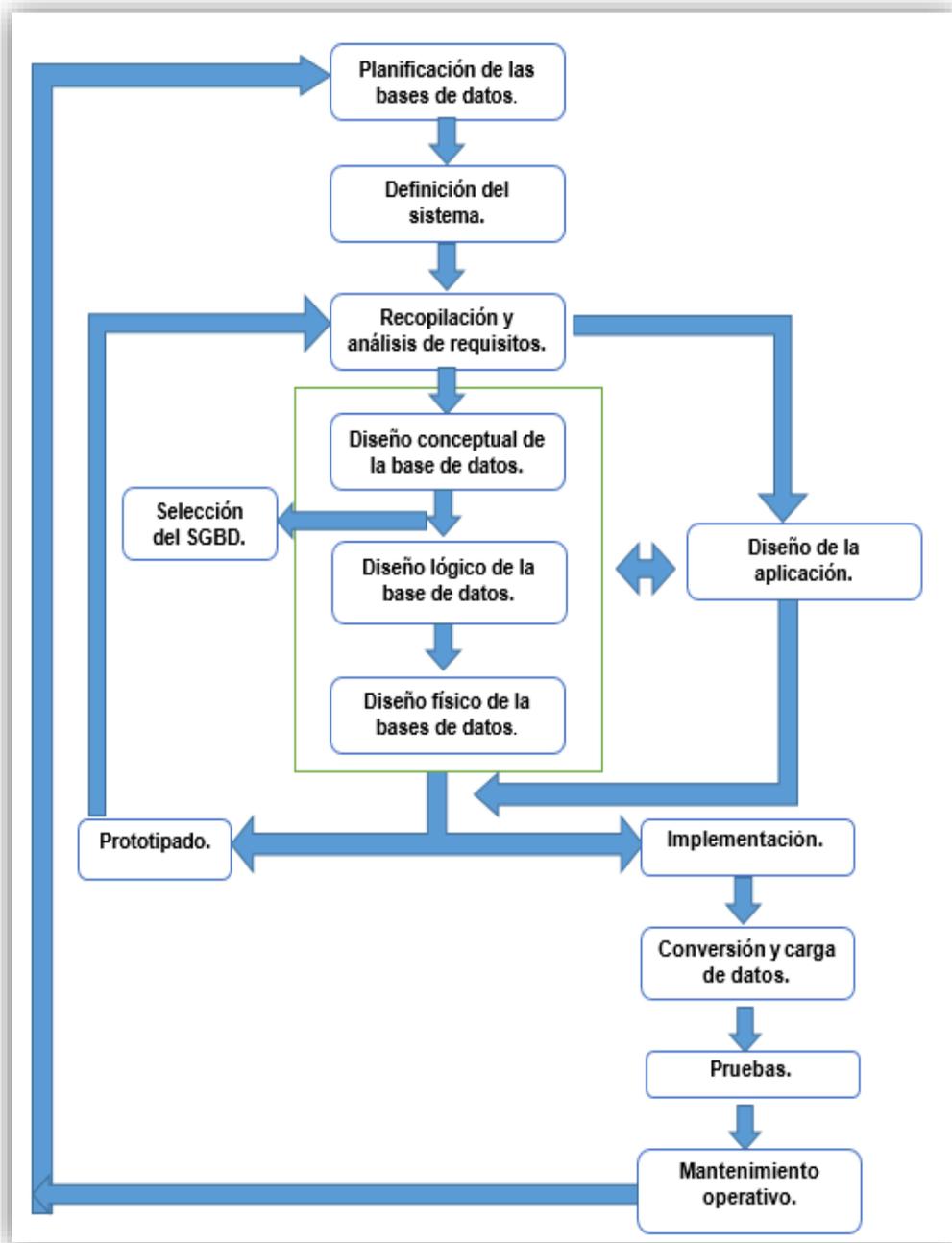


Figura 1.9-El ciclo de vida del desarrollo del sistema de bases de datos. (Thomas M. Connolly, 2005)

1.3.2- Diseño de la base de datos.

El diseño de la base de datos juega un papel muy importante dentro de la organización ya que a través de la cual se lleva a cabo la toma de decisiones y se logra dar soluciones a diversos problemas. Un diseño de bases de datos deberá brindar seguridad, escalabilidad, estabilidad a los sistemas de información, es por eso que se debe de implementar de una metodología que garantice su diseño y que cumplan con los objetivos de la organización.

La creación de aplicaciones de bases de datos implica varias fases, como es el diseño del esquema de la base de datos, el diseño del esquema de seguridad para controlar el acceso a los datos y el diseño de los programas que tienen acceso a los datos y los actualiza. La implementación de una metodología para el diseño de bases de datos nos ayuda a planificar, gestionar, controlar y evaluar el desarrollo de la base de datos.

Para comprender los requerimientos y necesidades de la información es necesario de realizar un modelado de datos, en el que se describe el conjunto de herramientas implementadas para describir los datos, las relaciones y restricciones que integran un sistema de base de datos. La construcción de un modelo de datos debe de representar el flujo de información y la forma de utilizarse ya que se elabora en base a los requisitos de la empresa.

Un modelo de datos es óptimo si satisface los siguientes criterios mostrados en el cuadro 1.2, sin embargo, debe tenerse en cuenta que dichos criterios no son compatibles entre sí, por lo que se hace necesario llegar a ciertos compromisos, como es el caso de tratar de mejorar la velocidad de expresión, podemos llegar a perder simplicidad. (Thomas M. Connolly, 2005).

Criterios para generar un modelo de datos óptimo.	
Validez de estructura.	Coherencia con la forma en que la empresa define y organiza la información.
Simplicidad.	Facilidad de comprensión por parte de los profesionales de los sistemas de información y por parte de los usuarios.
Capacidad de expresión.	Capacidad de distinguir entre diferentes datos, relaciones entre los datos y restricciones.
No redundancia.	Exclusión de información no pertinente; en particular, la representación de cada elemento de información una única vez.
Capacidad de compartición.	No específico de alguna aplicación o tecnología concretas y, como consecuencia, utilizable por muchas aplicaciones o tecnologías.
Ampliabilidad.	Capacidad de evolucionar para satisfacer nuevos requisitos con un efecto mínimo sobre los usuarios existentes.
Integridad.	Coherencia con la forma en que la empresa utiliza y gestiona la información.
Representación diagramática.	Capacidad de representar un modelo utilizando una notación diagramática fácilmente comprensible.

Cuadro 1.2 Criterios para generar un modelo de datos óptimo. (Thomas M. Connolly, 2005)

1.3.2.1- Diseño conceptual de la base de datos.

El diseño conceptual de la base de datos es el proceso de construir de un modelo de los datos en base a la información documentada, las necesidades de la organización y los requisitos de los usuarios que manejan dicha información. El modelo conceptual es un lenguaje que describe el contenido de la información de la base de datos es enteramente independiente de los detalles de implementación tales como el software

SGBD de destino, los programas de aplicación los lenguajes de programación, la plataforma hardware o cualquier otra consideración física.

El modelo conceptual de los datos está soportado en un conjunto de documentos que incluye el diagrama Entidad-Relación (ER) y en el diccionario de datos, que se generan al desarrollar el modelo.

- Identificar los tipos de entidad.
- Identificar los tipos de relación.
- Identificar y asociar los atributos con los tipos de entidad y de relación.
- Determinar los dominios de los atributos.
- Determinar los atributos de la clave candidata, primaria y alternativa. (Thomas M. Connolly, 2005)

El diccionario de datos es una herramienta con la que cuenta el administrador de base de datos para el manejo de los datos. Un diccionario debe de almacenar las descripciones internas, conceptuales y externas de la base de datos, permitiendo la descripción de campos, registros y referencias cruzadas entre los registros, los códigos de autorización y de seguridad, etc.

1.3.2.2- Diseño lógico de la base de datos.

La fase de diseño lógico de la base de datos establece una correspondencia entre el modelo conceptual y un modelo lógico, en donde se describe la estructura de la base de datos en términos al SGB seleccionado. El modelo lógico de los datos es una fuente de información para la fase de diseño físico. Esta fase consiste en traducir el esquema conceptual de alto nivel definido mediante el modelo Entidad-Relación en un esquema de relación.

Para poder traducir el modelo de datos conceptual en un modelo lógico de los datos y después validar el modelo para comprobar que sea estructurado correcto y capaz de soportarlas transacciones requeridas es necesario llevar a cabo los siguientes pasos:

- Determinar las relaciones para el modelo lógico de los datos.
- Validar las relaciones mediante técnicas de normalización.
- Validar las relaciones comprobando las transacciones de los usuarios.
- Comprobar las restricciones de integridad.
- Repasar el modelo lógico de los datos con los usuarios.

- Combinar los modelos lógicos de los datos en un modelo global.
- Verificar las consideraciones derivadas del crecimiento futuro. (Thomas M. Connolly, 2005).

1.3.2.3- Diseño físico de la base de datos.

Para realizar el diseño físico de la base de datos se debe conocer las características específicas sobre el sistema informático que alberga el Sistema de Gestión de Bases de Datos (SGBD) elegido ya que se busca optimizar y asegurar la integridad de los datos. Entre esas características están la forma de organizar los archivos, la forma de implementación de la base de datos, las relaciones base, los índices utilizados para conseguir un acceso eficiente a los datos, las estructuras de almacenamiento interno y secundario, así como las restricciones de integridad y medidas de seguridad asociadas.

El esquema físico de las bases de datos puede modificarse teniendo en cuenta que se encuentra estrechamente relacionado con el diseño lógico ya que las decisiones tomadas durante el diseño físico para mejorar las presentaciones pueden afectar al modelo lógico de los datos.

Los pasos de la metodología física del diseño de datos son los siguientes:

- Traducir el modelo lógico de los datos al SGBD seleccionado.
 - Diseñar las relaciones base.
 - Diseñar la representación de los datos variados.
 - Diseñar las restricciones generales.
- Diseñar la organización de los archivos.
- Analizar las transacciones.
- Seleccionar la organización de los archivos.
- Seleccionar los índices.
- Estimar los requisitos de espacio de disco.
- Diseñar las vistas de usuario.
- Diseñar los mecanismos de seguridad.
- Considerar la introducción de una cantidad controlada de redundancia.
- Monitorizar y ajustar el sistema final. (Thomas M. Connolly, 2005).

1.3.3- Selección SGBD.

El sistema gestor de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener las bases de datos, y proporcionar acceso controlado a la misma. El SGBD tiene como objetivo primordial proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

El proceso de selección SGBD tiene como objetivo seleccionar un sistema que satisfaga los requisitos actuales y futuros de la empresa, equilibrando dichos requisitos con respecto a los costos relevantes, que influyen la compra de productos de mismo SGBD y de cualquier hardware/software adicional requerido para soportar el sistema base de datos, los costos asociados con los cambios y con la formación del personal. Una técnica de selección consiste en comparar las características de cada SGBD con los requisitos, ya que se tiene la oportunidad de garantizar que el sistema proporcione beneficios reales a la empresa. Los pasos principales para seleccionar un SGBD son los siguientes:

- Definición de los términos de referencia de estudio.
- Selección de dos o tres de productos candidatos.
- Evaluar el producto.
- Recomendación de un producto y generación de informes. (Thomas M. Connolly, 2005)

Existen distintos tipos de SGBD libres como PostgreSQL, MySQL, Firebird, SQLite, DB2 Express-C, Apache Derby etc. que ofrecen escalabilidad, soporte, seguridad y un rendimiento óptimo a un menor costo que los SGBD comerciales. En el caso de MySQL que es un SGBD muy conocido y ampliamente usado por su simplicidad y notable rendimiento, se encuentra basado en un sistema de administración de bases de datos relacionales, es rápido, robusto, sólido y flexible, el cual se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor permitiendo la creación de sistemas de transacciones on-line. (Thibaud, 2006). A pesar que MySQL posee menos características que otros SGBD, contiene herramientas que le otorgan una alta escalabilidad. A continuación, se hace mención de algunas de las características que tiene:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.

- La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Es seguro ya que su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC. (Luis Alberto Casillas Santillán, 2007)

1.3.4- Optimización de consultas.

Cuando implantamos el SGBD bajo una máquina esta se encontrará íntimamente ligada con los recursos de la misma. Los problemas de rendimiento de un servidor pueden ser debido a varias causas. La mayoría de los problemas que se generan en el gestor son por causa de un mal diseño de las aplicaciones que atacan la información de las bases de datos, aunque no es solo la única causa. Existen tres puntos relacionados a la optimización del gestor de bases de datos, la primera es la optimización de las sentencias del Lenguaje de Consulta Estructurada (SQL) que forma parte de las aplicaciones, la optimización de las bases de datos y la optimización de los sistemas operativos.

Un mal diseño de una aplicación tiene consecuencias muy negativas para el correcto funcionamiento del gestor, lo cual redundará en una mala utilización de recursos de la máquina y en los tiempos de respuesta. Por lo tanto, para un buen rendimiento del SGBD se debe tener optimizado al máximo las sentencias SQL que forman parte de las aplicaciones, para después realizar una optimización de los recursos del sistema y de las bases de datos con el objetivo de adecuarlos a las necesidades de las aplicaciones. Si se tiene optimizada al máximo las aplicaciones, se podrá decir que un mal rendimiento del gestor se está produciendo por defecto de algún recurso de la máquina y no por un mal diseño de las sentencias SQL.

La optimización se llevará a cabo dependiendo a los sistemas de información y sus requerimientos, tomando en cuenta que el rendimiento dependerá del número de

consultas realizadas, del manejo de la información y de la cantidad de información que se tenga que tratar tanto en operaciones como en actualizaciones de datos.

Según el tipo de operaciones podemos hablar de:

- Aplicaciones OLTP (On Line Transaction Process), son aquellas donde se manejan un mayor número de operaciones de actualización de información
- Aplicaciones DSS (Decision Support System), son aquellas donde el tipo de operaciones se lanzan contra las bases de datos principalmente en la consulta de información. (Euzo, 1998)

1.3.5- Seguridad de la información.

La seguridad de la base de datos son los mecanismos que protegen a la base de datos frente a las amenazas intencionadas o accidentales. Las amenazas a la seguridad son cualquier situación o suceso, que pueda afectar adversamente y, consecuentemente, a la organización. Las consideraciones de seguridad no se aplican únicamente a los datos almacenados en la base de datos, los problemas de seguridad pueden afectar a otras partes del sistema, lo cual puede a su vez afectar a la propia base de datos. Entre los controles de seguridad de información para entorno multiusuario se incluyen los mecanismos de autorización, los controles de acceso, las vistas, los sistemas de copiado de seguridad y de recuperación, los mecanismos de integridad, los sistemas de cifrado.

La seguridad de los datos se socia frecuentemente con la integridad de los mismos. La seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizada, mientras que la integridad se refiere a la precisión o validez de esos datos.

- Seguridad: protección de los datos ante usuarios no autorizados.
- Integridad: protección de datos de usuarios autorizados. (Faudón, 2000)

La seguridad a la base de datos se ocupa de evitar el robo, fraude, pérdida de confidencialidad, pérdida de privacidad, pérdida de integridad y pérdida de disponibilidad. El desafío consiste en poder transmitir y recibir información a través de internet al mismo tiempo que se garantiza que:

- Esa información es inaccesible para todo el mundo salvo para el transmisor y el receptor (confidencialidad).
- La información no ha sido modificada durante la transmisión (integridad).
- El receptor puede estar seguro de que la información proviene del transmisor (autenticidad).
- El transmisor puede estar seguro que el receptor es el adecuado (no simulación).
- El transmisor no puede posteriormente negar que ha enviado la información (no repudio). (Thomas M. Connolly, 2005).

1.3.6- Diagramas de entidad relación.

El diagrama de entidad-relación (E-R) es un modelo de red que describe la distribución de datos almacenados en el sistema a través de un esquema gráfico empleando técnicas de entidades, que son objetos que se relacionan entre sí y poseen características particulares llamadas atributos. EL modelo E-R es una forma de representar la estructura y la relación los elementos de la información que se almacenarán en la base de datos. La figura 1.11 se muestra los elementos utilizados para crear un diagrama E-R.

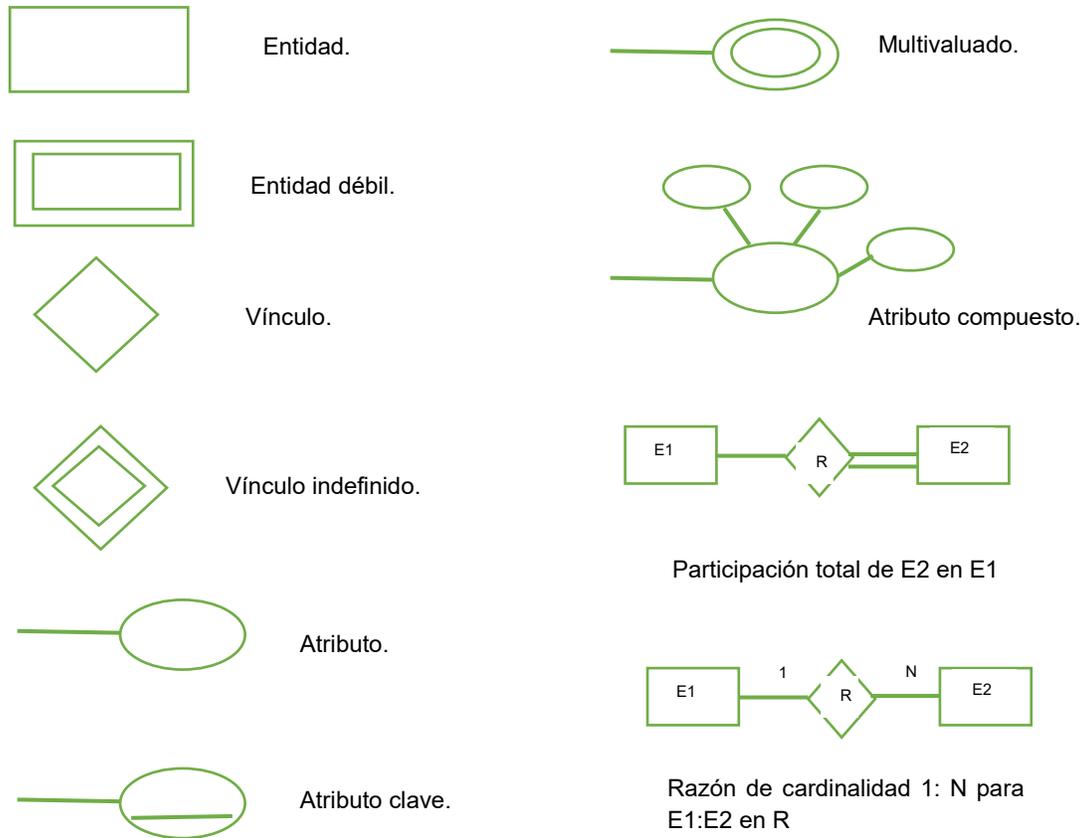


Figura 1.11- Los diagrama E-R implementan diferentes tipos de símbolos para modelar las entidades y las relaciones. (Elmasri, 2000)

1.4- Diagramas UML.

El lenguaje unificado de modelado (UML) nace en la década de los 90' con el fin de unificar y crear una representación estándar. Fue creada por Grady Booch, James Rumbaugh e Ivar Jacobson quienes en enero de 1997 dieron a conocer la representación estándar para los planos de software que pueden ser utilizadas o adaptadas a muchas metodologías de software. (Castillo E. G., 2010)

UML está compuesto por diversos elementos gráficos que se implementan para representar o modelar las fases del ciclo de vida del software, desde el análisis representando los procesos de negocio y requisitos funcionales que tendrá el software, hasta la implementación de la aplicación en sí. Los diagramas UML no definen un proceso de desarrollo específico solo son una representación gráfica a través de modelos que nos dan perspectivas específicas de distintos aspectos del software.

El lenguaje unificado comprende cierto número de diagramas interrelacionados que pueden ser implementados durante las fases de desarrollo del software y se agrupan en tres modelos:

- Estático: Este modelo se implementa durante todo el ciclo de vida del software y consta del diagrama de clases y de objetos.
- Dinámico: Los diagramas dinámicos describen las interacciones entre los objetos y son conocidos como diagramas de comportamiento. Los diagramas que se utilizan para modelar los aspectos dinámicos son los siguientes: diagrama de caso de uso, diagrama de estado y transiciones, diagrama de actividad y los diagramas de interacción que comprende a los diagramas de secuencia y colaboración.
- Implementación: Describen la estructura del software en cuanto a los componentes de que consta y su ubicación. Los diagramas utilizados para modelar la implementación son: el diagrama de componente y el de despliegue. (Falgueras, 2003)

Los diagramas UML tienen numerosos tipos de diagramas y, por lo tanto soporta la creación de muchos diferentes tipos de modelos de sistemas. Sin embargo se considera que solo cinco tipos de diagramas podrían representar lo esencial de un sistema los cuales son:

1. Diagramas de actividad que muestran las actividades incluidas en un proceso o en el procesamiento de datos, así como el flujo de control de una actividad a otra.
2. Diagramas de caso de uso, que expone las interacciones entre un sistema y su entorno.
3. Diagramas de secuencias, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema.
4. Diagramas de clase, que revelan las clases de objetos en el sistema y las asociaciones entre esas clases.
5. Diagramas de estado, que explican cómo reacciona el sistema frente a eventos internos y externos. (Sommerville, 2007)

Existen tipos de modelos de sistemas que se desarrollan como parte de la ingeniería de requerimientos y los procesos de diseño del sistema, los cuales son representados a través de un tipo de notación gráfica como es el lenguaje de modelado unificado, donde cada modelo representa una visión o perspectiva diferente del sistema. En la tabla 1.1 muestra los diagramas UML implementados en cada modelo del sistema.

Tipos de modelos de sistemas.	UML
Modelos de contexto	Diagrama de actividades.
Modelos de interacción	Diagrama de caso de uso. Diagrama de secuencia.
Modelos estructurados	Diagrama de clases.
Modelos de comportamiento	Diagramas de estado.

Tabla 1.1- Diagramas UML implementados en el modelado del sistema.

Patrones de diseño del software.

La fase de diseño del software es la actividad del ciclo de vida de ingeniería del software en donde se llevan a cabo el análisis de los requerimientos del software y se plantea la estructura que servirá como base para el desarrollo, es en esta fase donde se plantean las problemáticas que usualmente se generan durante el desarrollo del software y se crea una solución para cada ocasión, lo cual representa pérdida de tiempo. Sin embargo, la solución a dichas problemáticas pudiera ya existir, solo consistiría en aplicar la solución adecuada las veces que sean necesarias.

En 1979 el arquitecto Christopher Alexander propuso el aprendizaje y uso de series de patrones para la construcción de edificios de una mayor calidad, mencionando que “cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensar otra vez”. (Castillo E. G., 2010) Un patrón define una posible solución para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones.

Los patrones de diseño del software son recomendaciones aprobadas, estandarizadas y documentadas que sirven como modelo para dar soluciones a problemas comunes en el desarrollo de software y son considerados como buena práctica en la ingeniería del software. Existen varios patrones que permiten uniformar y diseñar soluciones estándar del software, para una aplicación Web los patrones que suelen aplicarse son el arquitectónico que describen problemas de diseño y que se resuelven con empleo de un enfoque estructural, los patrones de datos que describen problemáticas orientados a datos y se resuelven mediante modelado de datos, los patrones de componentes resuelven problemáticas asociados con el desarrollo de subsistemas y los patrones del diseño de la interfaz que plantea soluciones para el usuario final.

Sin embargo, uno de los patrones de diseño más conocido y utilizado es el Modelo Vista Controlador (MVC) que ofrece soluciones a problemas de arquitectura de software en ingeniería del software. Los patrones arquitectónicos son patrones de software que resuelven aspectos de adaptabilidad a requerimientos cambiantes, modularidad, acoplamiento, etc. estableciendo fundamentos para los analistas, diseñadores y programadores.

1.4.1- Modelo Vista Controlador (MVC).

El modelo vista controlador es un marco de trabajo usado para el diseño de la Interfaz Gráfica de Usuarios (GUIs) permite múltiples presentaciones de un objeto y estilos, soporta la presentación de los datos de diferentes formas e instrucciones independientes con cada uno de las presentaciones. MVC es el más común en el desarrollo Web ya que permite separar la lógica del control (el qué hacer), la lógica de negocio (cómo hacerlo) y la lógica de presentación (interfaces).

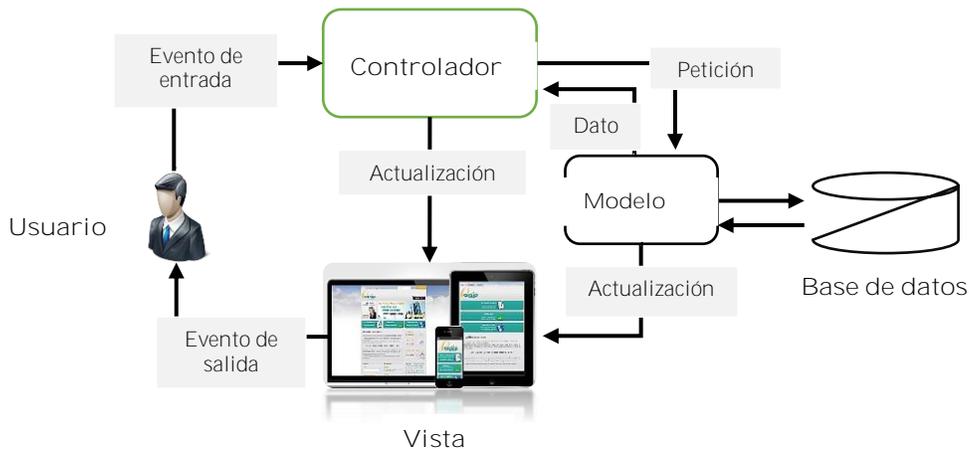


Figura 1.12- El patrón MVC es un paradigma que permite la implementación de sus elementos por separado, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo.

Modelo.

Representa la información sobre la cual el sistema opera gestionando todos los accesos dicha información tanto consultas, como actualizaciones y privilegios de acceso. La lógica de datos permite asegurar que los mismos mantengan su integridad a través de reglas definidas como, por ejemplo, realizando acciones, controlando sucesos, generando resultados, entre otras. Se encarga de enviar a la vista la información que se solicita en cada petición, dichas peticiones llegan al modelo a través del controlador. El modelo es responsable de:

- Proveer acceso a las capas de almacenamiento de datos, preferiblemente siendo independiente del sistema administración del mismo.
- Define las reglas del negocio que actuarán en el sistema
- Conservar el registro de las visitas y los controladores de la misma.

- Controlar los cambios de vistas ante cambios que se presenten frente a sucesos externos. (Jiménez, 2012)

Controlador.

Es el responsable de la respuesta ante los eventos que se presentan en el sistema, generalmente provocados por el usuario, invoca peticiones al modelo cuando se hace alguna solicitud sobre la información ya sea editar documentos o registro a un base de datos, también puede enviar eventos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo por ejemplo la actualización de la vista cuando se realiza una petición a la base de datos. El controlador es el intermediario de la vista y el modelo y es responsable de:

- Recibir los eventos de entrada.
- Contener y procesar reglas de gestión de eventos de la forma: “si evento=A entonces acción=B”. Puede representar soluciones al modelo o las vistas. (Jiménez, 2012)

Vista.

Es la representación visual mediante interfaces de la presentación del modelo (información y lógica del negocio) en un formato adecuado para interactuar con los usuarios finales, por tanto, requiere obtener la información de los datos que administra el modelo para ser presentados como salida y es responsable de:

- Recibir los datos del modelo y los presenta visualmente al usuario.
- Se asocia a un controlador.
- Ejecuta acciones en el modelo a través de un controlador. (Jiménez, 2012).

1.4.2- Zend Framework 2.

Zend Framework 2 es un marco de trabajo (Framework) de código abierto hecho para desarrollar aplicaciones Web y servicios Web con PHP, implementa código orientado a objetos, se encuentra construido mediante módulos haciendo así que cada componente posea una baja dependencia de otros componentes, por ese motivo es de fácil acoplamiento sin utilizar grandes recursos en la hora de la implantación e implementación de un módulo. Ofrece un alto desempeño al adaptar un MVC permitiendo ser más simple la abstracción de base de datos, utiliza HTML5 para la renderización del proyecto además de añadir validaciones y filtros de apoyo para los desarrolladores.

Zend Framework se basa en tecnologías hechas por Zend Technologies y es el principal patrocinador del Framework, a pesar de esto existen compañías que se encuentran involucradas como son Google, Microsoft y Strikelron que se han asociado con Zend para proporcionar interfaces de servicios Web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework. Zend no se encarga de todo el soporte que otorga, sino que hay una comunidad de miembros de Zend los cuales crean enlaces de módulos fáciles de implementar que pueden ser usados por todos.

Zend Framework 2 utiliza un sistema de módulos para organizar el código. El módulo application que provee la aplicación esqueleto se utiliza para proveer la configuración de inicialización, error y enrutamiento para toda la aplicación. Se utiliza habitualmente para proveer controladores de nivel de aplicación, se podría decir, la página de inicio de una aplicación.

Zend utiliza controladores los cuales coordinan las acciones del módulo, siendo ahí donde se recopila la información del MVC, el modelo de datos se hace por medio de objetos permitiendo a los controladores interactuar con las vistas las cuales solo recopilan información que se muestra al cliente. (Zend Technologies Ltd, 2015)

Actions o Acciones son los métodos que Zend utilizadas para responder a peticiones del cliente por GET y HTTP, estas acciones están contenidas en los controladores que son los que encargados de manipularlas, las acciones a su vez se encargan de comunicarse con la capa de datos o modelos para poder mostrar una vista, dichos modelos son instanciados desde la vista y debe de ser llamada igual que el nombre de la acción. Su nomenclatura es <nombre de la acción>Acción ()

Rutas en Zend son utilizadas por cada módulo para poder direccionar a cualquier controlador, puede existir más de un controlador, por lo que las rutas deben ser configuradas por módulo de forma única, no se debe de repetir el nombre de una ruta y estas deben ser definidas en el module.config.php. Hay varios tipos de rutas, tales como; Segment, literal, defaults, esta última se aplica cuando no se encuentra ninguna ruta valida dentro del módulo, cargándose un controlador y una acción predeterminada, la nomenclatura establecida es la siguiente: 'route' => '/' [: controller [/: action][/:id][/:id2][/:id3]]', en tal ejemplo se puede notar que la ruta tendrá un llamado de un controlador y una acción, y podrá pasar variables mediante URL sin que Zend lo reconozca como otra ruta y mande un mensaje de error ya que no ha encontrado la ruta determinada.

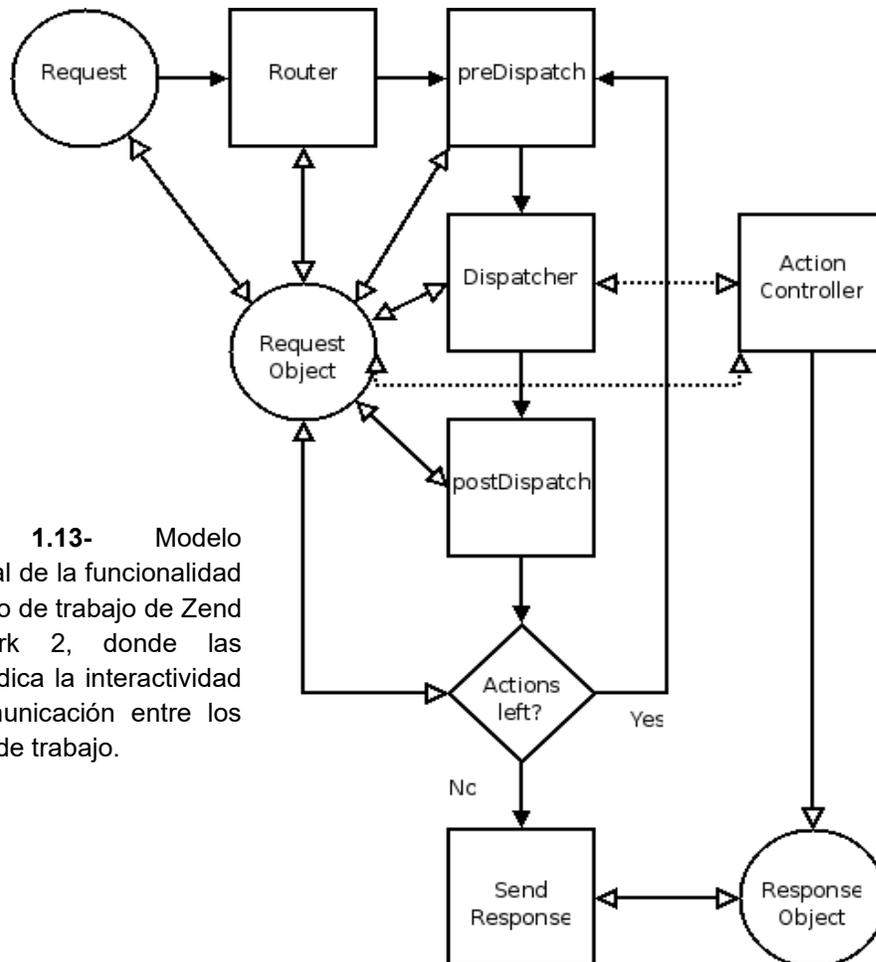


Figura 1.13- Modelo estructural de la funcionalidad del cuadro de trabajo de Zend Framework 2, donde las flechas indica la interactividad y la comunicación entre los módulos de trabajo.

1.4.3- PHTML

Es una extensión utilizada para un tipo de páginas Web, es más conocida y asociada con páginas que utilizan PHP y HTML. Este tipo de extensión es usado para acceder a las bases de datos de forma segura y comúnmente es asociado como un archivo de PHP.

1.4.4 Instalar ZEND FRAMEWORK 2

Pasos para instalar Zend mediante composer

1. Descargar composer en la dirección: <https://getcomposer.org/download/>
2. Instalar composer
3. Ir a la carpeta donde será instalado Zend
4. Abrir la línea de comandos y seguir los siguientes comandos
 - a. `curl -sS https://getcomposer.org/installer | php`
 - b. `curl -sS https://getcomposer.org/installer | /usr/bin/php5.5-cli`
 - c. `/usr/bin/php5.5-cli composer.phar selfupdate`
 - d. `/usr/bin/php5.5-cli ../composer.phar create-project --stability="dev" ZendFramework/skeleton-application ZendBolsa`
 - e. `require ZendFramework/zftool:dev-master`

1.4.5- Composer

Composer es una herramienta que permite gestionar las dependencias a nivel de aplicación desarrolladas en PHP. Nos permite declarar librerías necesarias para nuestro proyecto manteniéndolas actualizadas.

Composer no es un administrador de paquetes, si trata con “paquetes” o librerías, pero las gestiona en función de cada proyecto, instalándolas en un directorio dentro del mismo proyecto. Por defecto nunca instalara algo globalmente y en consecuencia, es un administrador de dependencias.

Composer está fuertemente basado en el gestor de paquetes predeterminado (Npm) para el entorno de ejecución de JavaScript Node.js. o bundler para Ruby. Resuelve todas las dependencias indirectas y descarga automáticamente la versión correcta de cada paquete.

Los problemas que Composer resuelve son los siguientes:

- a) Tú proyecto depende de un número de librerías.
- b) Algunas de esas librerías dependen de otras librerías.
- c) Tú declaras las cosas de las cuales dependes.
- d) Composer busca las versiones de estos paquetes necesitas instalar, y las instala (las descarga en tu proyecto).

1.5- CamelCase.

Este tipo de escritura, el CamelCase, se caracteriza porque las palabras van unidas entre sí sin espacios; con la peculiaridad de que las primeras letras de cada término se encuentran en mayúscula para hacer más legible el conjunto. Tradicionalmente, se había utilizado para la formulación química; pasando a ser empleado, en la actualidad, como un lenguaje de catalogación o clasificación, utilizado exclusivamente en la Web como lenguaje de programación o simplemente para llamar la atención con fines publicitarios.

El CamelCase admite dos posibles combinaciones entre mayúsculas y minúsculas: la primera palabra en mayúscula y el resto en minúscula (UpperCamelCase) o, por el contrario, cuando la primera está en minúscula y las demás están en mayúscula (lowerCamelCase).

Algunos ejemplos cotidianos sobre el uso de este tipo de escritura, que recibe el nombre de Camel (camello) por su parecido con la joroba de este animal, pueden ser el de McDonald's, McLaren, iPhone, iPod e iPad.

1.6- Singleton

Define una operación de instancia que permite que los clientes accedan a su única instancia mediante una operación de clase. La clase es responsable de crear su única instancia.

Debe haber exactamente una instancia de una clase y ésta debe ser accesible a los clientes desde un punto de acceso conocido.

La única instancia debería ser extensible mediante herencia y los clientes deberían ser capaces de utilizar una instancia extendida sin modificar su código.

Consecuencias

- Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.

Intención

- Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella.

Problema

- Varios clientes distintos precisan referenciar a un mismo elemento y queremos asegurarnos de que no hay más de una instancia de ese elemento.

Solución

- Garantizar una única instancia.

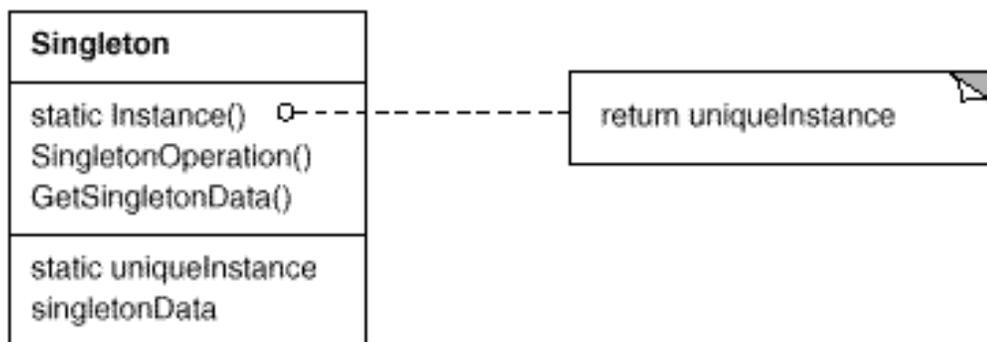


Figura 1.2- Modelo estructural de la funcionalidad del cuadro de trabajo de Singleton

1.7- Pruebas funcionales.

Las pruebas son un conjunto de actividades que deben plantearse y realizarse de manera sistemática, con finalidad de detectar y localizar errores en el contenido, función, utilidad, navegabilidad, capacidad y seguridad durante el desarrollo de una aplicación Web. Es también conocido como el proceso de verificación y validación del software en donde la verificación se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica y la validación es un conjunto de diferentes tareas que aseguran que el software que se construye sigue los requerimientos del cliente. (Roger S. Pressman, 2010)

Las pruebas del software es un proceso fundamental que determina la calidad del software y su confiabilidad. El proceso de pruebas comienza con pruebas que ejecutan la funcionalidad del contenido de la interfaz que son inmediatamente visibles al usuario final y posteriormente se llevan a cabo las pruebas que ejecutan con aspectos relacionados a la arquitectura del diseño y de la navegación, por último, se examinan las capacidades técnicas que no siempre son aparentes para el usuario. El cuadro 1.2 resume las pruebas del software que se implementan en las aplicaciones Web.

Pruebas del software implementadas en aplicaciones Web.	
Pruebas de contenido.	Son aquellas que se aplican a la información que se muestra al usuario a través de la interfaz de la aplicación y se refiere a los errores sintácticos (tipográficos o gramaticales), semánticos (información incorrecta o violación a las leyes de la propiedad intelectual) y de la organización inadecuada de la información.
Pruebas de interfaz de usuario.	Ese tipo de pruebas se verifica y valida durante el proceso del desarrollo de la interfaz. En la etapa de análisis se realiza para confirmar los requerimientos del usuario, en la etapa del diseño para revisar y garantizar los requerimientos establecidos y en la etapa de pruebas para verificar y validar la usabilidad.
Pruebas de componentes.	Son el conjunto de pruebas que se encargan de descubrir los errores de funcionalidad relacionados a la entrada y salida de información. Entre las que se pueden mencionar son las pruebas de nivel de componentes que se derivan de la entrada de información errónea a

	través de formularios y las pruebas de error forzado que consiste en descubrir errores durante la manipulación del error.
Pruebas de navegación.	Se realizan para asegurarse que cada interfaz realiza la función que se le ha encargado, así como los vínculos de navegación, las redirecciones, mapas de sitio, motores de búsqueda, etc.
Pruebas de base de datos	Se prueban la presión y la integridad de los datos almacenados por el servidor. Su finalidad es asegurar que los datos se almacenen, actualicen y se recuperen de una forma adecuada.

Cuadro 1.3- Pruebas del software implementadas en aplicaciones Web. (Roger S. Pressman, 2010)

Además de la etapa de pruebas existe el proceso depuración que consiste en la localización de errores y la corrección de estos durante el proceso de desarrollo del software. La depuración es llevada a cabo por los programadores ya que estos realizan pruebas de código durante el desarrollo del sistema y resuelven estos defectos antes de ser entregado a la fase de pruebas del software.

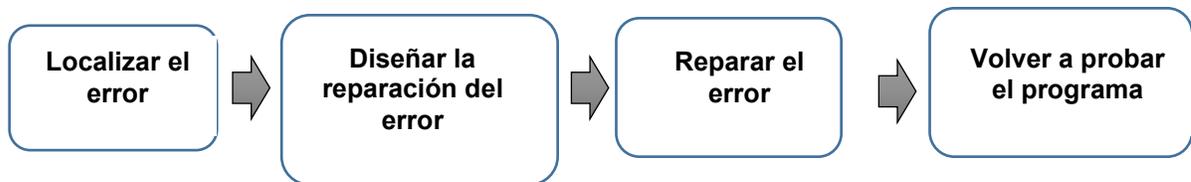


Figura 1.14 – En el proceso de depuración lleva a cabo la formulación de hipótesis para la localización de los errores. (Sommerville, 2007)

Capítulo 2- Bolsa de trabajo FES Cuautitlán.

La bolsa de trabajo cuenta con 2 oficinas una localizada en Campo uno y la otra en Campo cuatro, en las que se atiende los trámites relacionados al departamento, este cuenta con una sección de página Web en la cual se da una breve información a los alumnos de las funciones que realiza, esta página se encuentra con el sitio de la facultad, y se encuentra en el link: http://www.cuautitlan.unam.mx/alumnos/bolsa_trabajo.html. en la cual encontramos la siguiente información:

“SI ERES ALUMNO DE QUINTO SEMESTRE EN ADELANTE O EGRESADO LA BOLSA DE TRABAJO TE OFRECE LO SIGUIENTE:

- *Fungir como intermediario entre tú y el apartado productivo del país, obteniendo ofertas de empleo en las áreas relacionadas con tu carrera.*
- *Promocionar, Boletinar y Canalizar a las empresas tu perfil profesional.*
- *Apoyarte en tu inserción al mercado laboral a través de prácticas profesionales. Orientación y asesoría en la búsqueda de empleo e impartición de los siguientes talleres:*

- I. *"Preparando tu Búsqueda de Empleo"*
- II. *"Elaboración de tu Currículum Vitae".*

El departamento no cuenta con una misión, visión y objetivos tal como lo cuentan otras áreas de la facultad, por tal motivo se tuvo que generar uno para tener una identidad como institución y darle mayor formalidad al proyecto, además de generar análisis FODA, su organigrama y diagramas de datos los cuales nos sirven para tener una mejor representación de los datos y las entidades que interactúan con ella.

Las oficinas de bolsa de trabajo se encuentran ubicadas en Campo 1, Edificio de Servicios Escolares, planta baja con un horario de atención: lunes a viernes de 9:00 a 13:00 horas y 14:00 a 15:00 horas con el teléfono 5623 2016 y en Campo 4, en Edificio de Servicios Escolares, anexo con un horario de atención: lunes a viernes de 9:00 a 15:00 horas y 17:00 a 19:00 horas con el teléfono 5623 1815.

Con base en una entrevista que se tuvo con él encargado del área se pudo obtener información que no es visible en la página de la FESC ni en algún boletín que se dé a conocer a la comunidad estudiantil. Tales como: responsabilidades, nombres de

trabajadores y organigrama interno, los cuales se hace mención en los siguientes apartados.

2.1- Misión, Visión y Objetivo.

La bolsa de trabajo de la Facultad de Estudios Superiores Cuautitlán es un área que tiene como compromiso y responsabilidad la vinculación de los alumnos y egresados con el sector productivo y gubernamental, mediante estrategias de inserción y de desarrollo laboral, estableciendo los siguientes puntos como primordiales.

Fungir como intermediario entre el alumnado y el apartado productivo del país, obteniendo ofertas de empleo en las áreas relacionadas con tu carrera. Promocionar, publicar y canalizar a las empresas tu perfil profesional. Apoyar en la inserción al mercado laboral a través de prácticas profesionales. Orientación y asesoría en la búsqueda de empleo e impartición de los siguientes talleres: "Preparando tu Búsqueda de Empleo" "Elaboración de tu Currículum Vitae".

Misión.

Contribuir a la formación del capital humano que requiere el entorno laboral, mediante estrategias de inserción y desarrollo laboral orientadas a la comunidad universitaria de la FESC, para que con responsabilidad y compromiso social fortalezcan la calidad e innovación académica.

Visión.

Ser un entorno de comunicación funcional y reconocida por su influencia dentro y fuera de la Universidad, un medio efectivo de colaboración entre la FESC y los sectores productivo y gubernamental.

Ser la herramienta estratégica que vincule de manera efectiva, amplia y vigorosa a los estudiantes y egresados de la FESC con el campo profesional.

Constituir una guía permanente y una fuente autorizada de conocimientos e información vigentes para los buscadores de empleo de la comunidad universitaria en todo lo referente a oportunidades de puestos de trabajo, autopromoción profesional, capacitación y actualización.

Objetivo.

Vincular a los estudiantes y egresados de la facultad de Estudios Superiores Cuautitlán con el sector productivo, privado e instituciones gubernamentales, fungiendo como enlace entre los departamentos de recursos humanos y la comunidad universitaria para la obtención de ofertas de empleo, prácticas profesionales y/o becarios.

2.2- Organigrama.

El área de bolsa de trabajo de la FES Cuautitlán se encuentra representada jerárquicamente en el siguiente diagrama que se muestra en la figura 2.1, en donde se aprecia las dependencias y las relaciones con las que cuenta para llevar a cabo la toma de decisiones.

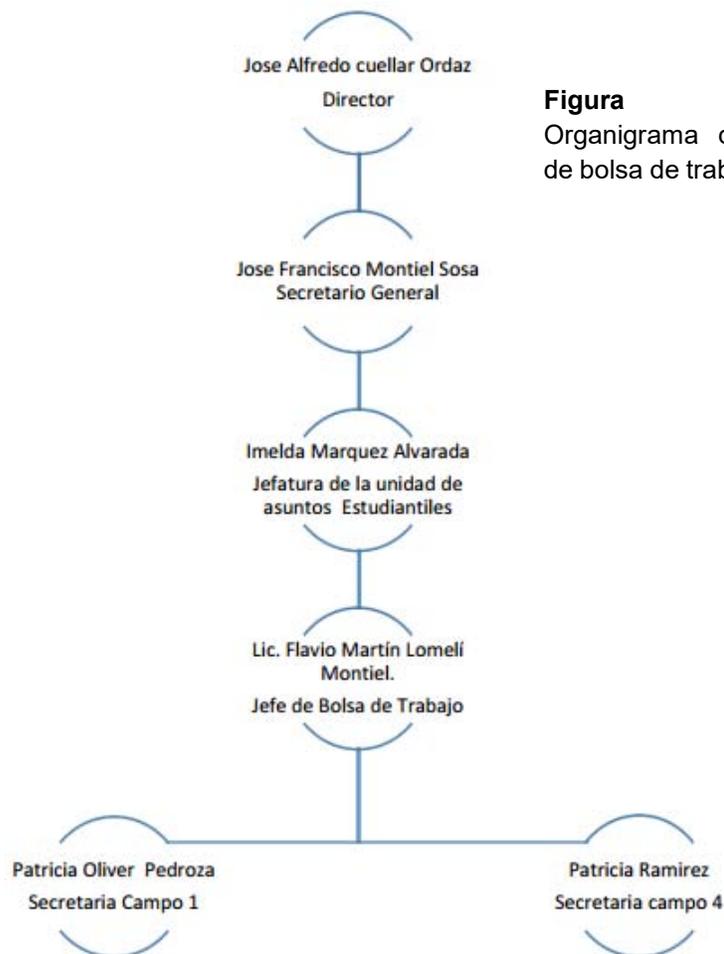


Figura 2.1-
Organigrama del área
de bolsa de trabajo FES

2.3- Responsabilidades.

El área de bolsa de trabajo FES Cuautitlán cuenta con un grupo de trabajo que de forma conjunta desempeñan diversas actividades que permiten el correcto funcionamiento de dicha área. Las funciones que se llevan a cabo dentro del área de bolsa de trabajo se encuentran definidas de acuerdo a la categoría de cada personal y se definen de la siguiente manera.

Secretarias.

- Capturar vacantes.
- Imprimir vacantes.
- Contestar vacantes.
- Atender empresas.

Jefe o encargado de bolsa de trabajo.

- Asistir a juntas con empresas.
- Asistir a reuniones.
- Administrar bolsa de trabajo.
- Realizar las funciones de la secretaria cuando haya terminado la jornada laboral de dicha secretaria o en caso de que el personal no haya asistido a laborar.

2.4- Funcionamiento de la bolsa de trabajo.

El área de bolsa de trabajo de la FESC Campo 4 establece un conjunto de requisitos indispensables para llevar a cabo el registro de un alumno y permitir que este tenga derecho a la obtención de la información. Los requisitos son:

- 1- Ser alumno de la FES Cuautitlán de quinto grado en adelante.
- 2- Ser exalumno de la FES Cuautitlán.
- 3- Contar con una foto reciente tamaño infantil.

Si el candidato cumple con los requisitos ya mencionados podrá asistir o llamar al área de bolsa de trabajo de campo 1 o campo 4, en los horarios establecidos para solicitar información específica. En caso de que el candidato desee realizar su registro deberá acudir a las oficinas correspondientes con una foto reciente tamaño infantil, en donde deberá llenar una “Ficha de registro” con sus datos personales, como la que se muestra en la figura 2.2. La ficha de registro contiene los siguientes campos:

- | | | |
|-----------------|----------------------|-----------------------|
| 1- Carrera | 7- Semestre | 13- Estado civil |
| 2- Área | 8- Pasante (SI/NO) | 14- Apellidos |
| 3- Fecha | 9- Titulado (SI/NO) | 15- Nombre |
| 4- N° Ficha | 10- Posgrado (SI/NO) | 16- Dirección |
| 5- N° de cuenta | 11- Promedio (SI/NO) | 17- Datos adicionales |
| 6- Foto | 12- Edad | 18- Sueldo |

FECHA: _____

FOTO
INFANTIL



FES CUAUTITLAN - UNAM
DEPARTAMENTO BOLSA DE TRABAJO
FICHA DE REGISTRO

No. _____

CARRERA: _____ AREA: _____ No. DE CTA: _____

SEMESTRE: _____ PASANTE () TITULADO () POSGRADO () PROMEDIO: _____

EDAD: _____ EDO CIVIL: _____

APELLIDO PATERNO _____ MATERNO _____ NOMBRE(S) _____

DOMICILIO _____

CALLE _____ NÚMERO _____ COLONIA _____

DELEGACION ó MUNICIPIO _____ C.P. _____ ENTIDAD _____ TELÉFONO _____

OTROS ESTUDIOS: _____

OTRO IDIOMA: _____ % _____ HORARIO DISPONIBLE: _____ SUELDO: _____

ÁREAS DE INTERÉS: _____

FECHA DE NACIMIENTO: _____ CORREO ELECTRONICO: _____

EXPERIENCIA LABORAL:

PUESTO	EMPRESA	AÑOS	SUELDO

Figura 2.2-Formato que lleva por nombre "Ficha de Registro", el cual debe llenarse con información correcta y legible para que el registro sea confiable.

Una vez llenada la ficha de registro de manera legible y correcta, el personal encargado corroborará la información proporcionada, si la ficha de registro es aceptada el candidato procede a llenar el siguiente formato llamado “boletín de candidatos” que se muestra en la figura 2.3, el cual es un documento que contiene datos generales de un currículo. Con el llenado del boletín de candidatos el interesado ha concluido su registro exitosamente y tendrá acceso solo a las vacantes que corresponden a su profesión, podrá tener acceso a dicha información las veces que sean necesarias, de preferencia una vez por semana, en las oficinas correspondientes y en horarios de atención.

**BOLETÍN DE CANDIDATOS
BOLSA DE TRABAJO**

FECHA: Junio 2014
CARRERA: Administración

NÚMERO DE REGISTRO	PUESTO DESIADO	NIL (LTS)	NOMBRE	FECHA DE NACIMIENTO	EDUCACIÓN	EXPERIENCIA	SUELDO DESIADO	EDAD	UBICACIÓN O MUNICIPIO DE RESIDENCIA	DESCRIPTORES DE EXPERIENCIA CONCERNIENTE APTITUDES, HABILIDADES, CONOCIMIENTOS, FORMAS Y OTROS FACTORES PROFESIONALES IMPORTANTES
3655	Administración	35%	Osage Velázquez	01-31-80	Lic. Administración en Desarrollo	Administración de proyectos 5 cuartos a cobrar.	de \$7,000 a \$10,000	34 años	Guadalupe A. Nayarit	Experiencia en actividades propias de un administrador de proyectos de inversión en el sector público de 5 años. Mestrado en el área de control de calidad para el sector público.
3656	Asistente / Analista Administrativo	60	Elio Isabel Lozano Esquivel	25-11-44-41-18	Lic. Administración	Admin. de Ventas, Servicio al Cliente	A partir de \$9,000	31	Quartiles Mexico	5 años de exp. en control de procesos administrativos, servicio al cliente, análisis de info. / Excel y JAP avanzado
3657	Administración	Passo	Rosita Patricia Vázquez	02-11-84	Lic. Administración	Trabajo en Administración	A partir de \$8,000	28 años	Tullihua Mexico	Experiencia en gestión, gestión de recursos de proyectos laborales (en el área de salud)
3658	Administración	70%	Olivero Cucho Norma Inés	02-04-85	Lic. en Administración	Trabajo en Administración	A partir de \$7,000	27 años	San Felipe Durango	Responsable, manejo de procesos, trabajo en equipo, gestión de recursos humanos, gestión de finanzas.
3658	Administración o Recursos Humanos	35%	Guadalupe Macías Yanesa	22-29-80	Admin. de Recursos Humanos	Administración de recursos humanos, control de costos, gestión de proyectos	A partir de \$4,000	20 años	Guadalupe A. Nayarit D.F.	Experiencia en gestión de recursos humanos, control de costos, gestión de proyectos, administración de recursos humanos, gestión de finanzas, gestión de operaciones.
3660	Administración en Mandatos de Recursos Humanos	Passo	Norma Amely Avila Rojas	21-04-83	Lic. Administración	Administración de Recursos Humanos	Menor a \$4,000	25 años	Praxinos de Guzman	Experiencia en el área de gestión de recursos humanos, control de costos, gestión de proyectos, administración de recursos humanos, gestión de finanzas, gestión de operaciones.
3659	Admon	30%	Osage Velázquez	01-31-80	Lic. Administración en Desarrollo	Administración de proyectos 5 cuartos a cobrar.	de \$7,000 a \$10,000	34 años	Guadalupe A. Nayarit	Experiencia en actividades propias de un administrador de proyectos de inversión en el sector público de 5 años. Mestrado en el área de control de calidad para el sector público.

Figura 2.3-Formato que lleva por nombre “Boletín de candidatos”, el cual contiene datos personales e información general de las preferencias laborales de los candidatos.

El registro de ofertantes se lleva a cabo de tres formas, la primera puede ser a través de ferias de empleo, donde el representante del área de bolsa de trabajo tiene oportunidad de difundir el objetivo de la bolsa de trabajo y realizar el registro de ofertantes ahí mismo, la segunda manera es a través de un correo electrónico que envían los ofertantes al encargado de la bolsa de trabajo y el último medio de registro es de manera presencial donde los ofertantes acuden a las oficinas de bolsa de trabajo y se lleva a cabo el registro del ofertante. En los tres casos mencionados el registro de un ofertante no cuenta con un formato establecido, si no que los ofertantes proporcionan datos generales de la organización y de la vacante en base a sus criterios. Finalmente, una vacante es publicada en un pizarrón que se encuentra fuera de las oficinas de la bolsa de trabajo y solo contiene los datos generales de la vacante, las publicaciones son clasificadas por carreras la cuales son identificadas por colores y dichos colores son asignados por el encargado del área. Un ejemplo de estas publicaciones de muestra en la figura 2.4

UNIDAD DE ASUNTOS ESTUDIANTILES
BOLSA DE TRABAJO
I N F O R M A
IMPORTANTE EMPRESA SOLICITA:
FECHA: 23 DE ENERO 2014
NÚMERO DE DEMANDA: 007
JEFE DE PRODUCCION
REQUISITOS:
IM.E: PASANTE O TITULADO, 4 AÑOS DE EXPERIENCIA COMO
JEFE DE PRODUCCION EN EMPRESAS DE GIRO ALIMENTICIO,
EDAD DE 30-40, SEXO MASCULINO, SUELDO DE \$25,000

Figura 2.4- Formato en la que se publica las vacantes que han sido registradas en la bolsa de trabajo. Solo se muestra la información general sobre la vacante ya que solo los usuarios registrados tienen acceso a la información específica de cada vacante. (Información de Bolsa de Trabajo)

El personal del área de bolsa de trabajo realiza el almacenamiento de la información de los candidatos registrados utilizando la herramienta de Word, en donde se ingresa los datos proporcionados en la ficha de registro y son clasificados por carreras para facilitar su consulta.

Para describir los procesos y las actividades que se llevan a cabo en la bolsa de trabajo se implementó el lenguaje unificado de modelado que se hace mención en el capítulo uno en el apartado 1.4, aplicando diagramas de actividades que permitirán representar la información proporcionada. A continuación se muestran los diagramas de actividades que representan como se realiza el registro de usuarios y el proceso para solicitar información.

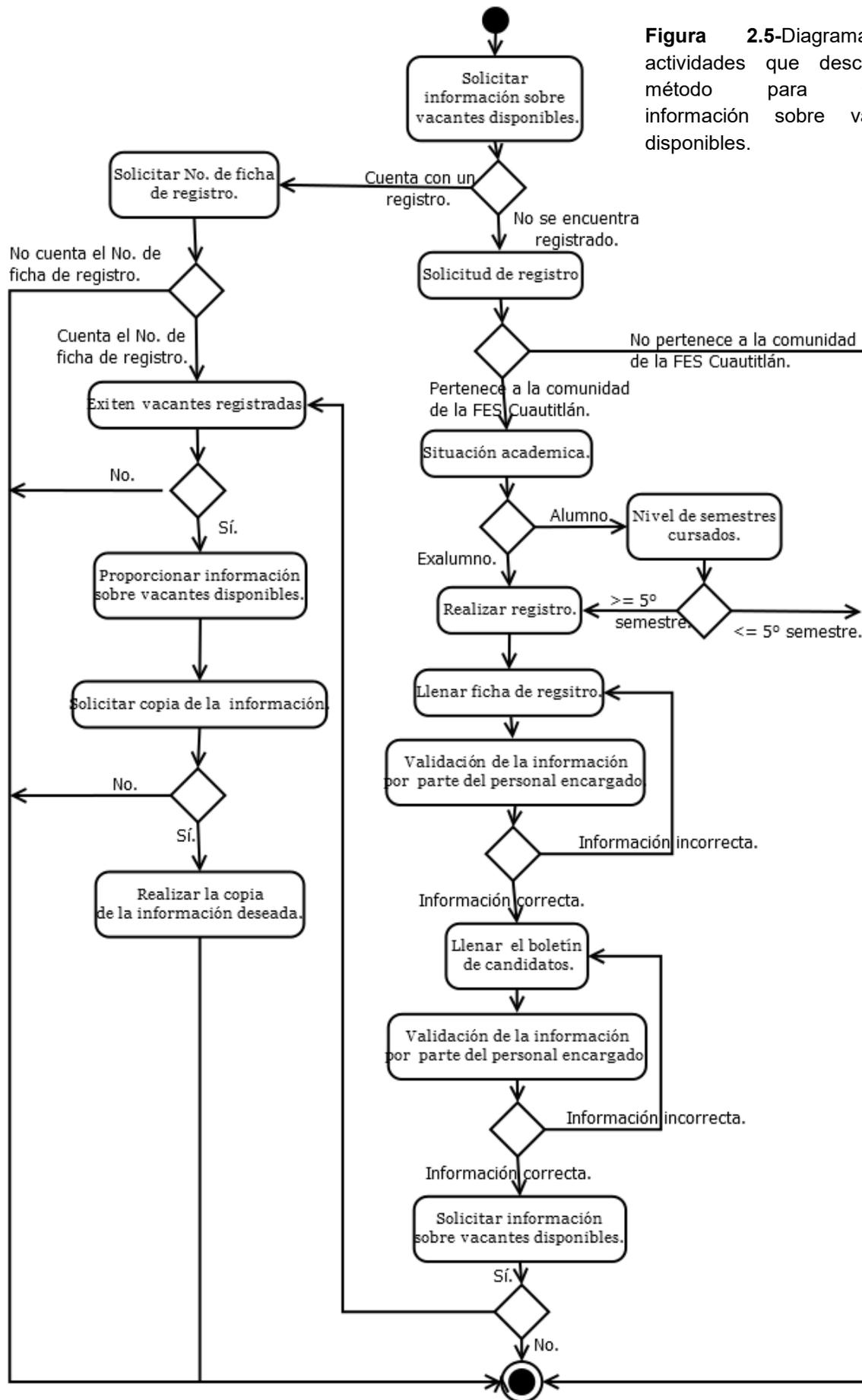


Figura 2.5-Diagrama de actividades que describe el método para solicitar información sobre vacantes disponibles.

En la figura 2.6 representa las actividades que se llevan a cabo para que una empresa realice su registro y publique sus vacantes.

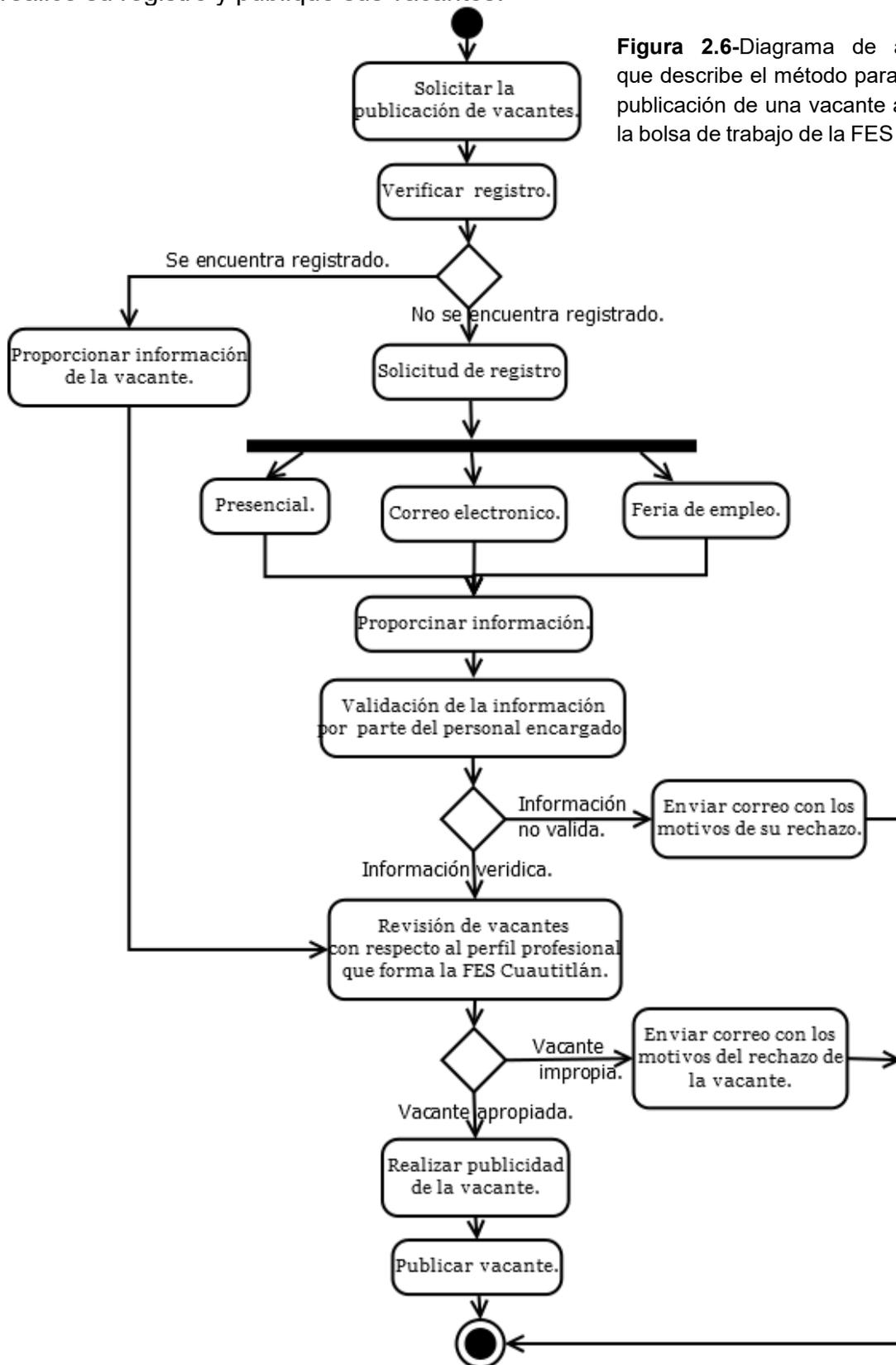
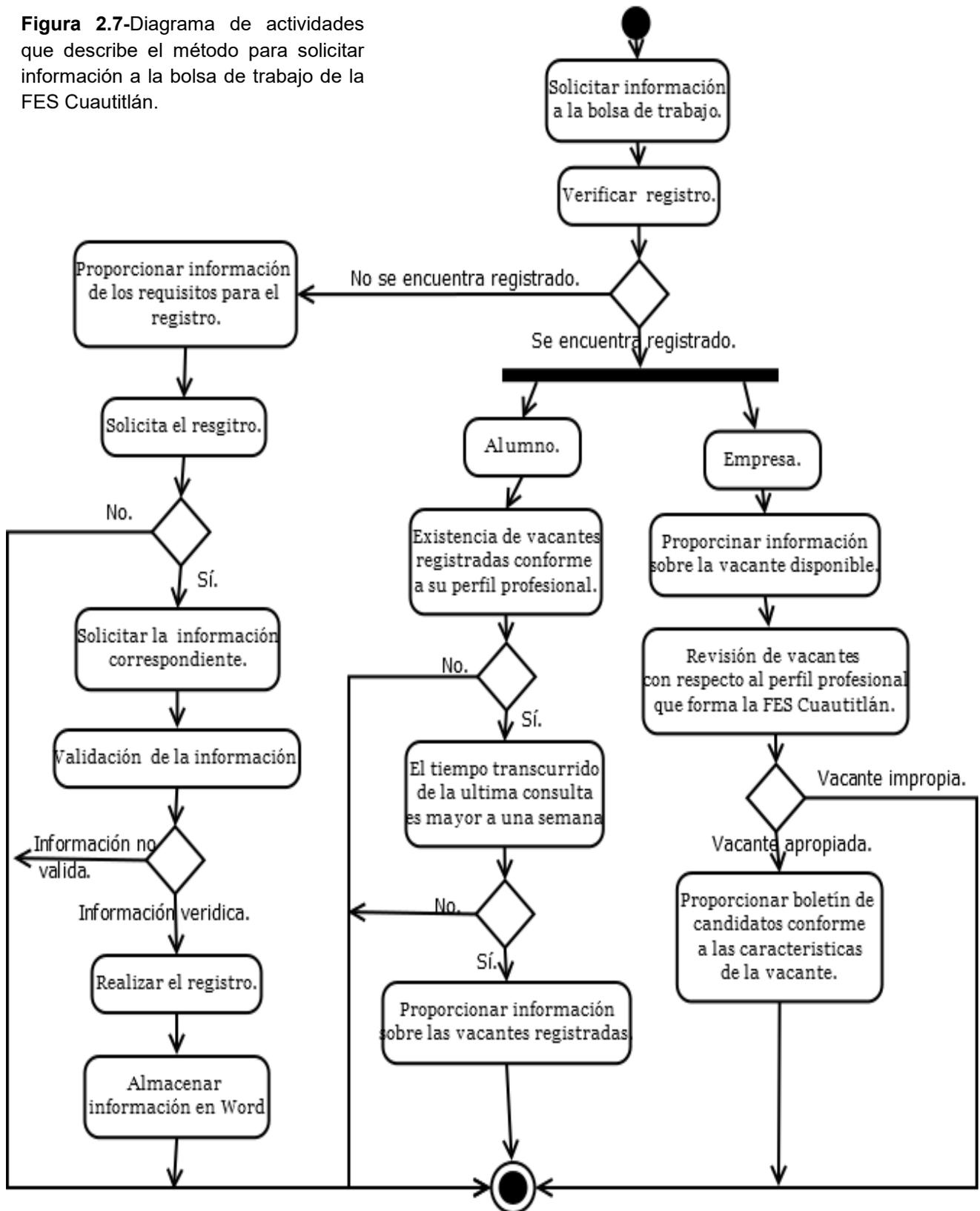


Figura 2.6-Diagrama de actividades que describe el método para solicitar la publicación de una vacante a través de la bolsa de trabajo de la FES Cuautitlán.

En la figura 2.7 representa las actividades que se llevan a cabo para que se proporcione información acerca de las vacantes disponibles y de los posibles candidatos.

Figura 2.7-Diagrama de actividades que describe el método para solicitar información a la bolsa de trabajo de la FES Cuautitlán.



Las figuras 2.8 y 2.9 ejemplifican los procesos con los que se encuentra vinculada la bolsa de trabajo.

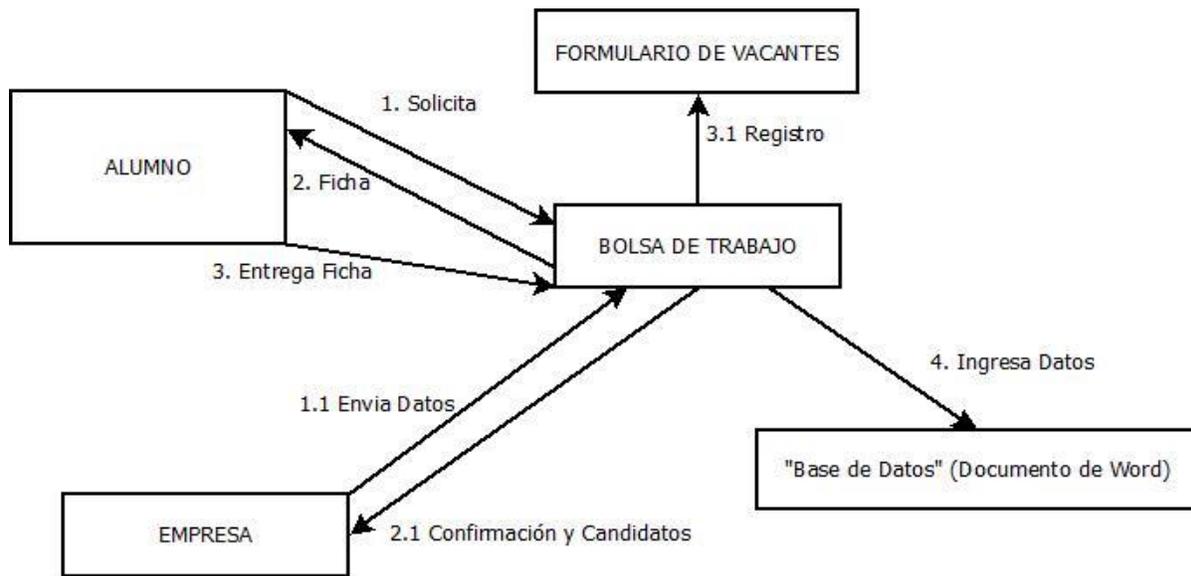


Figura 2.8-El diagrama resume el proceso que se realiza para llevar a cabo el registro de candidatos y ofertantes interesados.

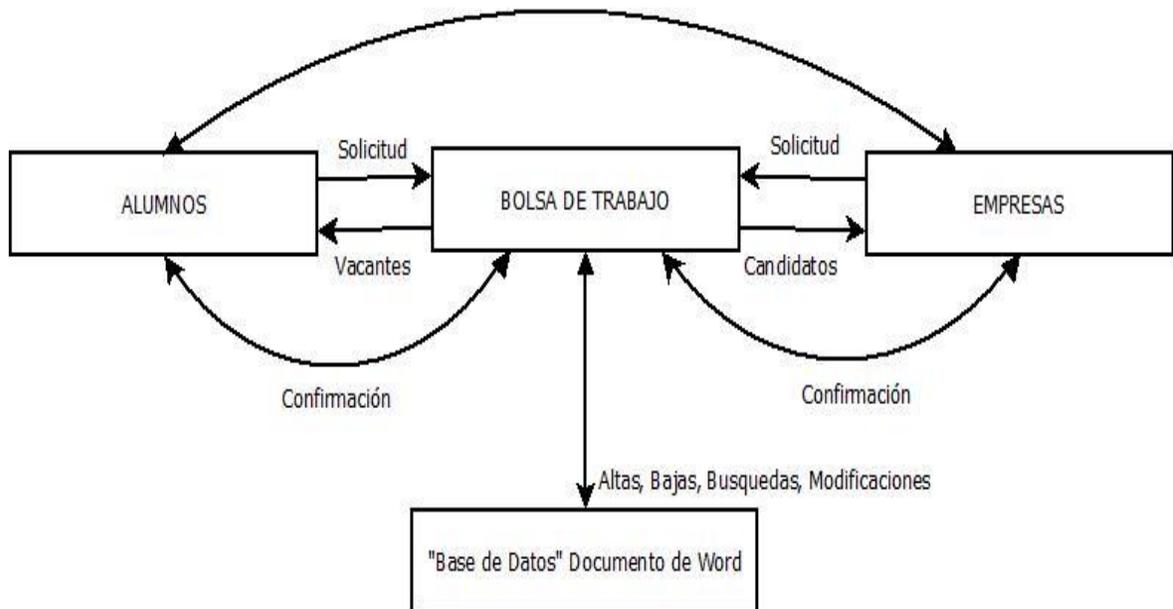


Figura 2.9-En este diagrama se muestra la manera en que interactúa la bolsa de trabajo con los candidatos y ofertantes registrados para coordinar sus actividades.

2.5- Análisis FODA.

Para obtener un panorama general acerca de la funcionalidad y operatividad que tiene la bolsa de trabajo de la FES Cuautitlán, se implementó el análisis FODA para conocer las fortalezas internas que facilitan el logro de objetivos, las limitaciones que impiden el alcance de las metas de una manera eficiente y efectiva, las condiciones o circunstancias ventajosas del entorno que la benefician y las tendencias que pueden ser perjudiciales.

Fortalezas.

- Es el único medio en la FES Cuautitlán que vincula a los alumnos con el sector productivo y gubernamental.
- Implementa estrategias de inserción al campo laboral.
- Contribuye a la calidad académica en la FES Cuautitlán.
- Facilita información sobre los ofertantes y sus vacantes disponibles.
- Difunde información específica de las vacantes disponibles.
- Ofrece una atención personalizada en diferentes horarios.

Oportunidades.

- Incorporar tecnologías más eficientes.
- Mejoras en la comunicación entre candidatos y ofertantes.
- Eliminar la redundancia de la información en los registros de candidatos y ofertantes.
- Proporcionar información actualizada, clara, oportuna y veraz.
- Incorporar un mayor número de candidatos al campo laboral.
- Incrementar el número de ofertantes interesados.
- Incorporar sistemas que mejoren la gestión de la información.
- Proporcionar un servicio más eficaz y eficiente.
- Proporcionar seguridad de la información de los usuarios.
- Mejorar la accesibilidad de la información.
- Incorporar los servicios proporcionados a la Web.

Debilidades.

- Complejidad en el registro. Los formatos que se llenan son tediosos y múltiples.
- La información proporcionada en el boletín de candidatos es parcial y no cumplen con el objetivo de un currículum.
- Información ilegible ya que es escrita por el candidato.
- No existe seguridad de la información ya que los alumnos que se registran en el boletín de candidatos pueden visualizar información de los alumnos que se registraron con anterioridad existiendo el riesgo que sea robada o alterada la información proporcionada.
- Riesgo de pérdida de información por extravío de documentos y porque no existe respaldos de la misma.
- Ambigüedad de la información en los datos solicitados.
- Redundancia de la información ya que se llenan varios formatos con los mismos datos.
- La información es inasequible ya que la consulta de vacantes solo se realiza en horarios de oficina y en el área de bolsa de trabajo lo que significa una dependencia del personal que elabora en dicha área.
- Información obsoleta porque no se garantiza que las vacantes que se consultan aún se encuentren disponibles ya que no se cuenta con una actualización de la información.
- No se garantiza la difusión de los datos del alumno ya que difícilmente una empresa acude al área de bolsa de trabajo a solicitar el boletín de candidatos.
- Se dificulta el acceso a información específica de los candidatos y ofertantes para la toma de decisiones porque el almacenamiento de la información se lleva a cabo en documentos de Word y dicha herramienta de trabajo no permite la gestión de la información.
- Procesos retardados porque no existe una automatización de los procesos que se realizan en el área de bolsa de trabajo por lo que los candidatos y ofertantes registrados dependen de la disponibilidad del personal.
- Se genera grandes cantidades de archivo muerto por la cantidad de documentación que se genera en papel.
- Información imprecisa porque no existe un formato para publicar las vacantes y tampoco para el registro de ofertantes.
- Limitación de vacantes ya que solo se registra una carrera preferente.

- No existe un filtro para el acceso a la información.

Amenazas.

- Existencia de sitios Web automatizados y competitivos que ofrecen un servicio de bolsa de trabajo.
- Preferencia por la competencia por falta de difusión.
- Presupuesto limitado por la FES Cuautitlán.
- Escaso interés por parte de los ofertantes por las limitaciones de la accesibilidad.

Capítulo 3- Análisis, diseño y desarrollo del sistema.

3.1- Análisis y definición de los requerimientos del sistema.

De acuerdo al proceso de ingeniería Web establecido en el capítulo uno el cual se muestra en la figura 1.2 se sigue el siguiente proceso de desarrollo de software.

➤ Comunicación con el cliente.

Para determinar los requisitos del sistema es necesario definir las necesidades que se tiene con respecto a las actividades que se realizan en la bolsa de trabajo, con el objetivo de resolver las problemáticas existentes. La determinación de los requerimientos generales se planteó en una entrevista que se llevó a cabo con los responsables involucrados en el funcionamiento de la bolsa de trabajo de la FES Cuautitlán, como es el jefe de bolsa de trabajo, quien plantea las siguientes necesidades.

La creación de una aplicación Web dedicada a los alumnos y egresados de la FES Cuautitlán que permita la vinculación con el sector productivo y gubernamental, proporcionándoles información precisa y oportuna sobre las vacantes disponibles e incorporándolos al mundo laboral de una manera más ágil y rápida, sin tener que asistir a las oficinas ubicadas en Campo 1 y Campo 4.

La aplicación Web deberá dar a conocer las actividades que cumple la bolsa de trabajo de la FES Cuautitlán, tal como es su misión, visión, objetivos, actividades que se desempeñan, medios de contacto, etc. así como la publicación de eventos importantes o promocionales de importancia. Permitiendo el registro de usuarios a través del sitio sin necesidad de acudir a las oficinas de bolsa de trabajo, mediante el llenado de formularios cortos y precisos evitando la redundancia de información.

El usuario deberá contar con una cuenta personal donde podrá actualizar su información cada vez que sea necesario o cuando el sitio se lo pida, dicha cuenta proporcionará información actualizada dependiendo del tipo de usuario. Para el usuario alumno se le deberá mostrar la información de las vacantes disponibles acerca de su profesión, y permitirá la creación de un documento que exponga su perfil profesional. El usuario empresa deberá poder publicar vacantes, buscar alumnos y establecer contacto con los alumnos.

La gestión de información debe de administrada por un usuario tipo administrador, el cual ingresará los catálogos necesarios para que la aplicación funcione correctamente,

podrá generar reportes de la información que se ha almacenado en el sistema. La aplicación Web deberá validar la entrada de información a fin de que no se publiquen vacantes inadecuadas o que se registren empresas que no cumplan las características que el administrador tenga en las reglas del negocio.

El conjunto de características y de requerimientos planteados anteriormente por los responsables del área permite establecer un enfoque de desarrollo y definir el alcance que tendrá la aplicación Web de bolsa de trabajo.

➤ **Planteamiento.**

Con base en el análisis FODA que se plantea en el capítulo 2.5 y los requerimientos propuestos, el desarrollo de la aplicación Web de bolsa de trabajo propone incorporar la automatización de los procesos que se realizan en dicha área, incorporando nuevas tecnologías, mencionadas en el capítulo 1, que permitan ofrecer un servicio de calidad para usuarios a través de internet.

Necesidades generales detectadas.

1. Automatización de procesos.
2. Mayor seguridad de la información.
3. Integridad de la información.
4. Mayor acceso simultaneo a los datos.
5. Reducción del espacio de almacenamiento de la información.
6. Menor redundancia de la información.
7. Menor ambigüedad de la información.
8. Registro de usuario eficiente.
9. Mayor difusión de la información.
10. Mayor accesibilidad de la información.
11. Actualización de la información.
12. Mayor oportunidad de contratación.
13. Mejores oportunidades de búsqueda de candidatos.

En la tabla 3.1. se muestra la propuesta de las funciones generales que contiene cada módulo del sistema.

Módulo.	Funcionalidades generales.
Inicio.	<ul style="list-style-type: none"> ➤ Mostrar información a través de escritos o galería. ➤ Menú general con las secciones más destacadas de la bolsa de trabajo (Inicio, Registro, Acerca de nosotros y Contactos) ➤ Registro de usuarios (alumno y empresa). ➤ Inicio de sesión de usuarios (validación de cuenta de usuario). ➤ Seguridad de la información en el registro de usuarios. ➤ Recuperar contraseña de usuario. ➤ Envío de correo de confirmación del registro.
Alumno.	<ul style="list-style-type: none"> ➤ Menú general con las secciones más destacadas de la bolsa de trabajo (Perfil, Vacantes, Postulaciones, Curriculum y Cerrar sesión) ➤ Actualización de información (personal, académica y preferencias laborales). ➤ Personalizar perfil (editar usuario y contraseña). ➤ Creación del currículo. ➤ Visualizar y buscar empresas registradas. ➤ Visualizar y buscar nuevas vacantes. ➤ Realizar postulaciones. ➤ Seguridad de la cuenta (tiempo de vida de sesión).
Empresa.	<ul style="list-style-type: none"> ➤ Menú general con las secciones más destacadas de la bolsa de trabajo (Perfil, Vacantes, Postulaciones, Curriculum y Cerrar sesión) ➤ Actualización de información personal. ➤ Personalizar perfil (editar usuario y contraseña). ➤ Registrar vacantes. ➤ Ver vacantes (activas, rechazadas, vencidas). ➤ Visualizar notificaciones de nuevas postulaciones. ➤ Realizar postulaciones. ➤ Ver candidatos. ➤ Ver postulaciones. ➤ Seguridad de la cuenta (tiempo de vida de sesión).
Administrador.	<ul style="list-style-type: none"> ➤ Menú general con las secciones más destacadas de la bolsa de trabajo (Empresas, Alumnos, Vacantes, Cuentas de usuarios, Gestión de interfaz, Catálogos y Cerrar sesión). ➤ Visualizar tareas pendientes. ➤ Gestionar el registro de empresas (aceptar o rechazar el registro). ➤ Validar las vacantes registradas. ➤ Filtrar información sensible del alumno. ➤ Gestionar información mostrada en la de interfaz. ➤ Gestionar alta, baja y actualizaciones de los catálogos de base de datos. ➤ Crear cuentas de usuarios. ➤ Realizar reportes.

Tabla-3.1- Tabla de funcionalidades generales del software.

En la página principal se podrán incorporar todos los promocionales que la bolsa de trabajo generé y/o la FESC, a través de una galería que tendrá vínculos con la página de inicio de la aplicación, para proporcionar información específica tal como conferencias y contrataciones masivas organizadas por el área.

Se mostrará un menú general con las secciones más destacadas de la bolsa de trabajo, como es el registro de usuarios, prácticas profesionales. Además, se mostrará los medios de contacto, el personal responsable y se anexará una sección dedicada al ingreso de los usuarios para facilitar el acceso a su cuenta, con esto se eliminará la necesidad de acudir a las oficinas a solicitar información referente al registro o de los eventos próximos a realizar.

El usuario colocará en dicha plataforma datos personales relacionados con para realizar el registro de usuarios a la bolsa de trabajo se contará con un área determinada en la aplicación en la cual solo se solicitarán datos específicos del usuario para evitar el llenado de múltiples documentos, creando una cuenta personal única por cada usuario registrado, de tal manera se garantizará el acceso de la información válida, ya que solo se aceptarán aquellos usuarios que cumplan los requisitos del registro.

El candidato podrá registrar datos personales y datos referentes a su perfil profesional, a través de un formato general para poder eliminar la redundancia de la información que se crea al llenar varios formatos con los mismos requisitos.

Los datos solicitados se utilizarán para poder crear un currículum que será visible a las organizaciones si el usuario así lo definió en su perfil, ofreciéndole una mayor difusión de su perfil profesional. Con esta información se remplazará al boletín de candidatos que contenía información duplicada, ofreciendo seguridad de la información.

Las actualizaciones al perfil serán por el usuario, teniendo la oportunidad de elegir sobre el acceso a su información ya que él decidirá si su perfil será público o privado, además se ofrece un documento digital legible y completo sobre su formación profesional, que facilita su comprensión y le proporciona la información necesaria a los ofertantes interesados. Esto elimina la inseguridad que se tiene al registrarse en el boletín de candidatos ya que el acceso a dicho documento la tienen todos los candidatos interesados, en dicho formato se lleva a cabo el registro de varios alumnos, teniendo acceso a la información de los candidatos registrados con anterioridad, existiendo el

riesgo de que la información pueda ser alterada, borrada, eliminada o robada por otro candidato.

Las organizaciones ya no serán obligadas a tener que acudir a las oficinas de bolsa de trabajo de la FES Cuautitlán a solicitar el boletín de candidatos el cual es impreciso, ilegible, además de tener información repetida y no confiable ya que hay alumnos que ya ingresaron al mercado laboral. Se llevará un control de las personas que ya fueron contratadas para que las empresas no tengan que asistir de nuevo a las oficinas e informar si ya han contratado a alguien de la comunidad FESC.

La información que se le mostrará al aspirante sobre las organizaciones y sus vacantes disponibles tendrán una vigencia de dos meses con el propósito de proporcionar información oportuna y actualizada que sea de utilidad para el usuario. Se podrá visualizar aquellas vacantes que se relacionen a su formación profesional, para evitar mostrarle información que no sea de su interés.

Las vacantes serán mostradas conforme a la fecha de publicación lo que significa que las primeras vacantes que se mostrarán serán las más actuales. Se le proporcionará herramientas para filtrar la información y así facilitar la búsqueda de vacantes, además se le permitirá al usuario alumno realizar postulaciones a las vacantes que sean de su preferencia, estableciendo comunicación con los ofertantes. Con esto se pretende mejorar la comunicación entre ambos usuarios ya que no se cuenta con este tipo de actividades y la bolsa de trabajo no conoce cifras exactas sobre el número de alumnos contactados por los ofertantes.

El registro de las empresas ofertantes dependerá de la aprobación del encargado o administrador de la aplicación Web de bolsa de trabajo, ya que se corroborará si la institución existe y si es formal, para evitar el registro de instituciones ficticias o fraudulentas. Una vez analizada dicha información se le creará una cuenta que podrá ingresar al portal de la bolsa de trabajo. La empresa contará con una cuenta que le permitirá actualizar su información cada vez que sea necesario, estableciendo un formato de registro para todas las empresas ya que no se cuenta con uno.

La publicación de vacantes consistirá en el llenado de un formato general donde se especifican las carreras relacionadas a la vacante y el perfil necesario para cubrirlas, estas también quedaran sujetas a la aprobación del encargado o administrador de la

aplicación Web porque solo se aceptarán las vacantes que cumplan con la formación profesional que ofrece la FES Cuautitlán.

La página principal de la empresa mostrará los datos generales de aquellos candidatos que se encuentran relacionados al perfil de sus vacantes registradas, teniendo la oportunidad de visualizar su currículum si este se encuentra público. Se le proporcionará herramientas que faciliten la búsqueda de candidatos, teniendo oportunidad de elegir los candidatos más óptimos para sus vacantes, en otro módulo del sitio se mostrarán los candidatos postulados y podrá establecer comunicación con ellos si así lo desea.

La aplicación Web de bolsa de trabajo permitirá reducir y simplificar las actividades realizadas por el personal encargado, ya que la difusión de información sobre las actividades más destacadas que se llevan a cabo en la bolsa de trabajo se realizarán a través de la interfaz de la aplicación haciéndolas más accesibles y atractivas al público. agilizará la toma de decisiones ya que permitirá gestionar la información mediante la aplicación de un gestor de bases de datos que proporcionará información detallada acerca de la información manipulada para la elaboración de reportes de acuerdo a las necesidades del área, mejorará la calidad del servicio y aumentará el número de usuarios interesados ya que el acceso a la información es más factible porque se realiza a través de internet y no se tiene que acudir y depender del personal de bolsa de trabajo

Proporcionará seguridad a la información ya que se contará con un respaldo de la información obtenida sin necesidad de acumular papelería que se pueda perder o mezclar con otra información, solo se permitirá la gestión de la información a los usuarios responsables del área, contando con una cuenta personal que tendrá un tiempo de caducidad en caso de inactividad para evitar que sea abierta posteriormente por otros usuarios.

Con la implementación de sitio Web, la bolsa de trabajo de la FES Cuautitlán ofrecerá mejoras en la comunicación entre candidatos y ofertantes, proporcionará información actualizada, clara, oportuna y veraz, incorporará un mayor número de candidatos al campo laboral, será más competitiva ya que tendrá una mejor cobertura de las necesidades del usuario ofreciendo un servicio de calidad y eficiente.

3.2- Diseño del sistema.

La aplicación de Bolsa de Trabajo de la FES Cuautitlán se desarrollará bajo el patrón de diseño Modelo Vista Controlador (MVC) que ofrece Zend Framework 2 (ZF2), ya que es un Framework robusto de alto rendimiento que cuenta con herramientas tecnológicas que permiten desarrollar aplicaciones Web basado en SPL (Estándar PHP Library) con paradigma de programación orientada a objetos facilitando la modulación de procesos y la reutilización de código, además cuenta con servicios de Table Gateway que permite abstracción de bases de datos e implementa el servicio Front Controller con el que logramos un mejor renderizado de formularios HTML, validación y filtración de datos proporcionando mayor seguridad de la información.

Para la implementación de Zend Framework 2 es necesario un servidor, la instalación de Composer y el Framework Zend Skeleton Application el cual consta de un conjunto de carpetas que construyen la arquitectura básica para el desarrollo de una aplicación Web, estas se encuentran ordenadas de acuerdo a su contenido y funcionamiento, contienen diversos archivos relacionados que trabajan conjuntamente. Por ejemplo, el archivo `init_autoloader.php` que permite que el proyecto se cargue a través del archivo `index` que se encuentra almacenado en el directorio `public` en donde se almacenan las hojas de estilos, imágenes y eventos de JavaScript.

El directorio `config` contiene todas las configuraciones básicas para cargar los módulos creados en ZF2. El directorio `vendor` contiene todos los componentes y librerías generales para poder trabajar en el proyecto. El directorio `module` contiene un subconjunto de carpetas en donde se crearán los módulos, vistas, controladores, formularios, etc., que formarán parte de la interfaz y la funcionalidad del proyecto.

La funcionalidad se encontrará coordinada por la arquitectura MVC que controlará los procesos, la entrada y salida de información de la base de datos, modelos, controladores, vistas etc. La estructura base consta de cuatro módulos, los cuales se encuentran definidos de acuerdo al tipo de usuarios que consultan la aplicación Web. Cada módulo contiene un conjunto de características específicas que definen las funcionalidades básicas a implementar y que dependerán del tipo de usuario (alumno, empresa y administrador).

La página principal será el primer módulo llamado Inicio, en donde los usuarios podrán consultar información general acerca de la bolsa de trabajo, tales como son horarios de

atención, eventos relevantes, objetivos, misión, visión, etc. a este módulo tendrán acceso todo tipo de usuarios ya que su objetivo principal es publicar información que se quiera dar a conocer a la comunidad. Su funcionalidad dependerá del menú principal que se encontrará en la parte superior que contiene cinco opciones, las cuales son inicio, regístrate, acerca de nosotros, contactos y prácticas profesionales. La opción inicio es un enlace a la página principal que contiene como cabecera en la parte superior izquierda el logotipo de la UNAM, el cual es un vínculo a la página www.unam.mx, en la parte superior derecha se encuentra una imagen representativa de la FES Cuautitlán que de igual manera contiene un enlace, a la página Web www.cuautitlan.unam.mx, este encabezado será utilizado por todas las interfaces que se desarrollen en el proyecto.

En la página principal se mostrará en la parte central un banner que contiene un conjunto de imágenes que se van a ir cambiando en el tiempo que se permanezca en la página, en la parte superior derecha estará una sección dedicada para que el usuario pueda iniciar sesión. Este consistirá en un formulario donde proporcionarán su usuario y contraseña correspondiente. En la parte inferior derecha se encontrará la información necesaria para que los usuarios puedan establecer contacto con el encargado de bolsa de trabajo. La opción regístrate contiene dos opciones más, la opción alumno y la opción empresa, estas opciones son los tipos de usuarios que pueden registrarse y que tendrán acceso a la aplicación, dichas opciones contarán con un vínculo que mostrarán los formularios que deberán ser llenados de forma correcta para poder llevar a cabo su registro, los formularios dependerán del tipo de opción seleccionada.

Acerca de nosotros es una opción del menú principal que consiste en tres apartados que mostrarán la misión, visión y objetivos de la bolsa de trabajo. La opción contactos mostrará los datos generales del personal encargado de la bolsa de trabajo, así como los medios por los cuales podrán solicitar información específica. La opción prácticas profesionales solo mostrará la información necesaria para realizar el trámite de prácticas profesionales.

En el módulo alumno solo lo tendrán acceso los usuarios registrados que cuenten con un perfil de alumno y con un usuario y contraseña válidos vigentes en el sistema. Este contiene un menú vertical en la parte lateral izquierda, ya que en la parte central se visualizará las vacantes actuales que se encuentran disponibles y que van acorde a su carrera profesional, permitiéndole ver información específica sobre la vacante al dar clic

sobre la misma. Dentro del menú se encontrará en la parte superior la foto del usuario (solo en caso de que el usuario registre alguna), su nombre y la carrera correspondiente, las opciones con las que contará son las siguientes: perfil, vacantes, postulaciones y cerrar sesión.

La opción perfil mostrará al usuario los datos actuales que se encuentren registrados y podrá editarlos en caso de que sea necesario. En esta opción el usuario podrá cambiar su contraseña y actualizar sus datos personales (dirección, datos académicos, referencias laborales, datos complementarios e idiomas, etc.), opciones que se encuentran contenidos en el submenú actualizar y por cada opción al usuario se le proporcionará un formulario con los datos correspondientes para que pueda realizar las actualizaciones.

Vacantes es la opción que le permitirá al alumno visualizar todas las vacantes actualizadas y disponibles que se encuentran relacionadas con su carrera profesional. Para facilitar la búsqueda de las vacantes el usuario contará con cuatro opciones que servirán para filtrar las vacantes y dichas opciones son; edad, tipo de horarios, tipos de géneros y sueldos, cada vacante cuenta con dos opciones:

- Postular: permite al alumno postularse a una vacante, mostrándose como interesado en cubrir dicha vacante notificando a la organización acerca de la postulación realizada por el alumno.
- Ver detalles: consiste en ver los datos específicos de la vacante, como los requisitos solicitados o información acerca de la organización quien solicita la vacante.

La opción postulaciones mostrará las solicitudes que el usuario ha realizado en la opción “vacantes”, además de tener la opción de ver los detalles generales de las mismas. La opción cerrar sesión, finalizará la sesión del usuario regresándolo a la página principal de la aplicación. La sesión de un usuario está sujeta a un límite de tiempo, la cual se cierra cuando se excede el límite de 15 minutos de inactividad.

En el módulo empresa solo lo tendrán acceso los usuarios registrados que cuenten con un perfil de empresas y con un usuario y contraseña válidos y activos que les permitirá acceder a la aplicación. En la parte central habrá una sección donde se muestren las notificaciones acerca de los postulados y de las vacantes pendientes que existan,

también se mostrará un menú vertical en la parte lateral izquierda, en donde se encontrará el nombre de la organización en la parte superior y en la parte inferior las siguientes opciones: perfil, vacantes, postulaciones, candidatos y cerrar sesión. La opción perfil mostrará al usuario los datos actuales que se encuentren registrados, en esta opción el usuario podrá cambiar su contraseña y actualizar los datos generales de la organización.

Vacantes consta de registro vacantes y ver vacantes. Registro de vacantes en donde por medio de un formulario el usuario proporcionará información específica de la vacante, teniendo oportunidad de registrar más de una carrera por vacante, todas las vacantes registradas se encontrarán en un estado de pendientes hasta que sean aprobadas por el encargado de la Bolsa de Trabajo. En la opción ver vacantes, se mostrarán las vacantes que ya fueron revisadas por el administrador y se mostrará como pendientes por validar, vigentes o activas y rechazadas dependiendo de la situación de la vacante, en caso de que se trate de una vacante rechazada la empresa podrá editar la información de la vacante y solicitar la validación una vez más.

La opción candidatos le permitirá al usuario visualizar los candidatos que se encuentran relacionados con la vacante, para facilitar la búsqueda de los candidatos la empresa contará con cuatro opciones que servirán como herramientas para filtrar la información, dichas opciones son edad, tipo de horarios, tipos de géneros y sueldos. El usuario contará con dos opciones, la primera opción es postular la cual al darle clic la empresa estará postulando al alumno a su vacante y la segunda opción es ver detalles que consiste en ver los datos generales del perfil del alumno.

La opción postulaciones mostrará las postulaciones que la empresa ha realizado, así como de los alumnos que han realizado postulaciones a sus vacantes teniendo como opción ver los datos generales del perfil del alumno. Como última opción cerrar sesión finalizará la sesión del usuario regresándolo a la página principal de la aplicación. La sesión de un usuario está sujeta a un límite de tiempo, la cual se cierra cuando se excede el límite de 15 minutos de inactividad.

El módulo administrador será gestionado por el personal encargado de la bolsa de trabajo, este contará con un nombre de usuario y una contraseña que deberá ingresar como los demás usuarios. Este módulo contendrá un contenedor principal en donde se mostrará una notificación acerca de nuevos registros de empresas o vacantes. El menú

se encontrará en la parte lateral izquierda y tendrá las siguientes opciones: empresas, vacantes, alumnos, interfaz, catálogos, administradores, reportes y cerrar sesión.

El apartado de empresas consta de tres opciones: nuevos registros, el cual mostrará una lista de las nuevas empresas que se han registrado, por cada registro se podrá ver los detalles específicos de la empresa, en esta sección el administrador podrá realizar la aceptación o el rechazo de la empresa, a la cual se le notificará a través de un correo electrónico. Las demás opciones mostrarán las empresas que se encuentran rechazadas y activas teniendo elección de poder ver los datos específicos.

El apartado vacantes está formada por cuatro opciones: nuevos registros, el cual mostrará una lista de las nuevas vacantes registradas por las empresas, por cada registro de vacantes se podrá ver los detalles específicos de la empresa quien la realizo y los requerimientos solicitados, aquí él administrador podrá aprobar o rechazar la vacante, en caso de que la vacante sea rechazada el administrador deberá especificarle a la empresa el motivo del rechazo, dándole a la empresa la oportunidad de que pueda modificarla. Las demás opciones mostrarán las vacantes que se encuentran rechazadas, vencidas y activas según sea el caso teniendo elección de poder ver los datos específicos por cada una.

La opción interfaz le permitirá al administrador actualizar las imágenes que se mostrarán en la página principal, se le mostrará una lista de las imágenes actuales y la opción para poder actualizarlas. La opción catálogos consistirá en tres opciones: carreras, idiomas y horarios, en cada opción se mostrará una lista de los registros actuales, teniendo como opción actualizar o realizar un nuevo registro dependiendo la opción elegida.

El apartado de administrador permitirá gestionar los usuarios encargados de la aplicación, esta opción permitirá registrar un nuevo administrador, actualizar información de los administradores, mostrar una lista de los administradores registrados y poder dar de baja a un administrador. La opción reportes le permitirá realizar reportes al administrador por rangos de fechas específicas de acuerdo a sus necesidades, se mostrarán los reportes en tablas. La opción cerrar sesión finalizará la sesión del usuario regresándolo a la página principal de la aplicación. La sesión de un usuario está sujeta a un límite de tiempo, la cual se cierra cuando se excede el límite de 15 minutos de inactividad.

3.3- Modelado del sistema.

Con referencia al modelo de procesos de ingeniería Web planteado en el capítulo 1 el modelado del sistema es la tercera etapa donde se lleva a cabo la representación gráfica de las necesidades y los requisitos que la aplicación Web cumplirá.

Sin embargo durante el desarrollo del modelado del sistema se debe considerar que los modelos dependerán de quiénes los utilizarán, por ello cada modelo tendrá un diferente enfoque y un grado de nivel de detalle. Los modelos se emplean con frecuencia de la siguiente manera:

- Como medio para facilitar la difusión de un sistema existente propuesto.
- Como una forma de documentar un sistema existente.
- Como una descripción detallada del sistema que sirva para generar una implementación de sistema.

El modelado del sistema de bolsa de trabajo se realiza enfocado a la difusión del sistema a través de diagramas que permitan la comprensión de los procesos automatizados, las actividades realizadas y los módulos aplicados, por lo que el nivel de detalle aplicado en los diagramas se hace en forma general con relación a las actividades.

3.3.1- Diagrama de casos de uso.

Los diagramas de caso de uso forman parte del modelo de interacción del sistema, los cuales representan las interacciones del usuario con el sistema y sus componentes externos. En los siguientes apartados se muestra los diagramas de caso de uso que hacen referencia a los cuatro tipos de usuarios o clientes que podrán interactuar con el sistema de bolsa de trabajo, así como las relaciones de dependencia que se tiene entre usuarios.

Página de inicio: En la página inicio los usuarios podrán consultar información general que será proporcionada por el encargado de la bolsa de trabajo, los usuarios podrán registrarse e iniciar sesión.

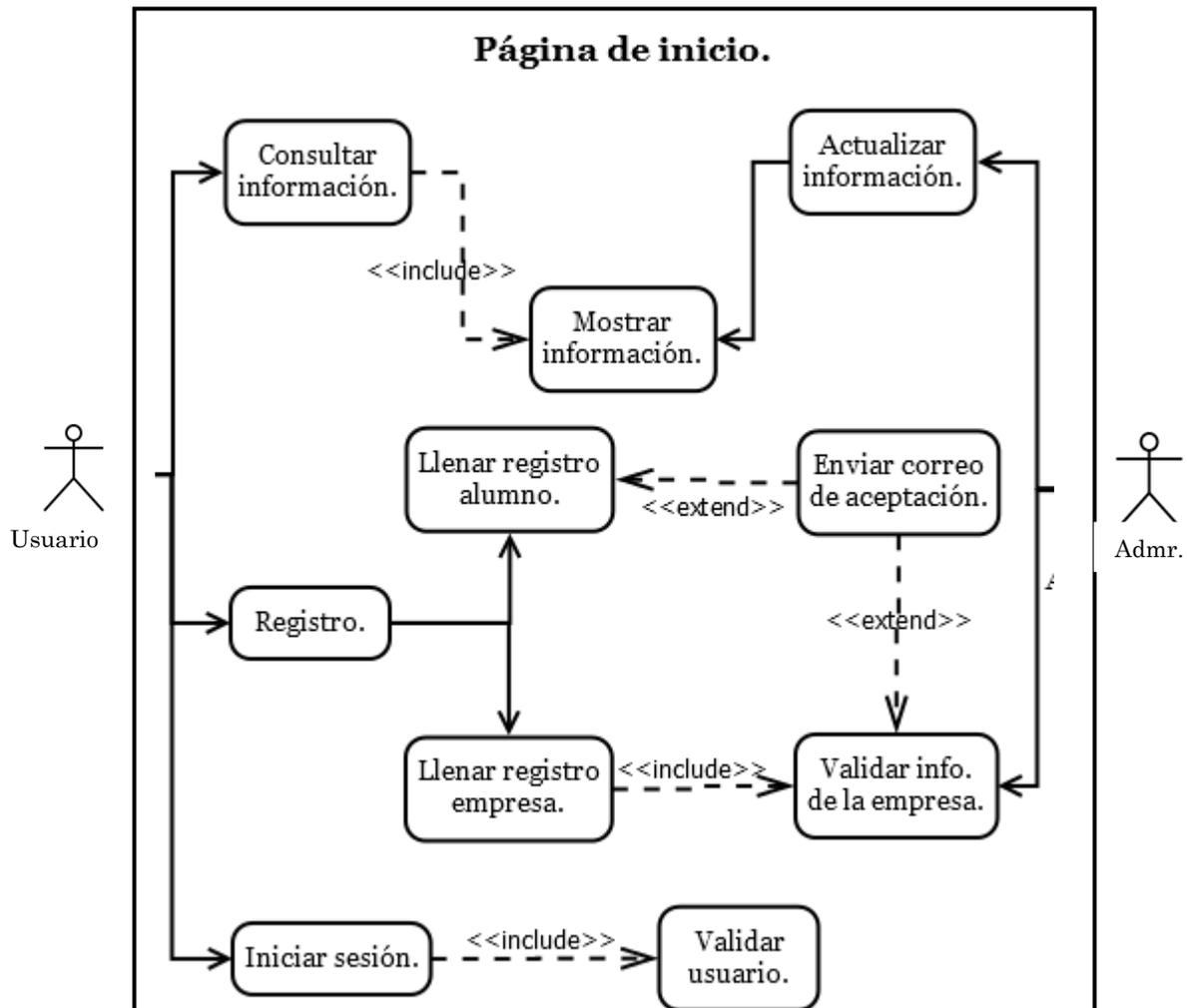


Figura 3.1 – Diagrama de caso de uso del módulo de la página de inicio.

Módulo alumno: Este módulo está dedicado para el usuario alumno ya que podrá actualizar su información personal, ver vacantes relacionadas a su profesión y realizar postulaciones.

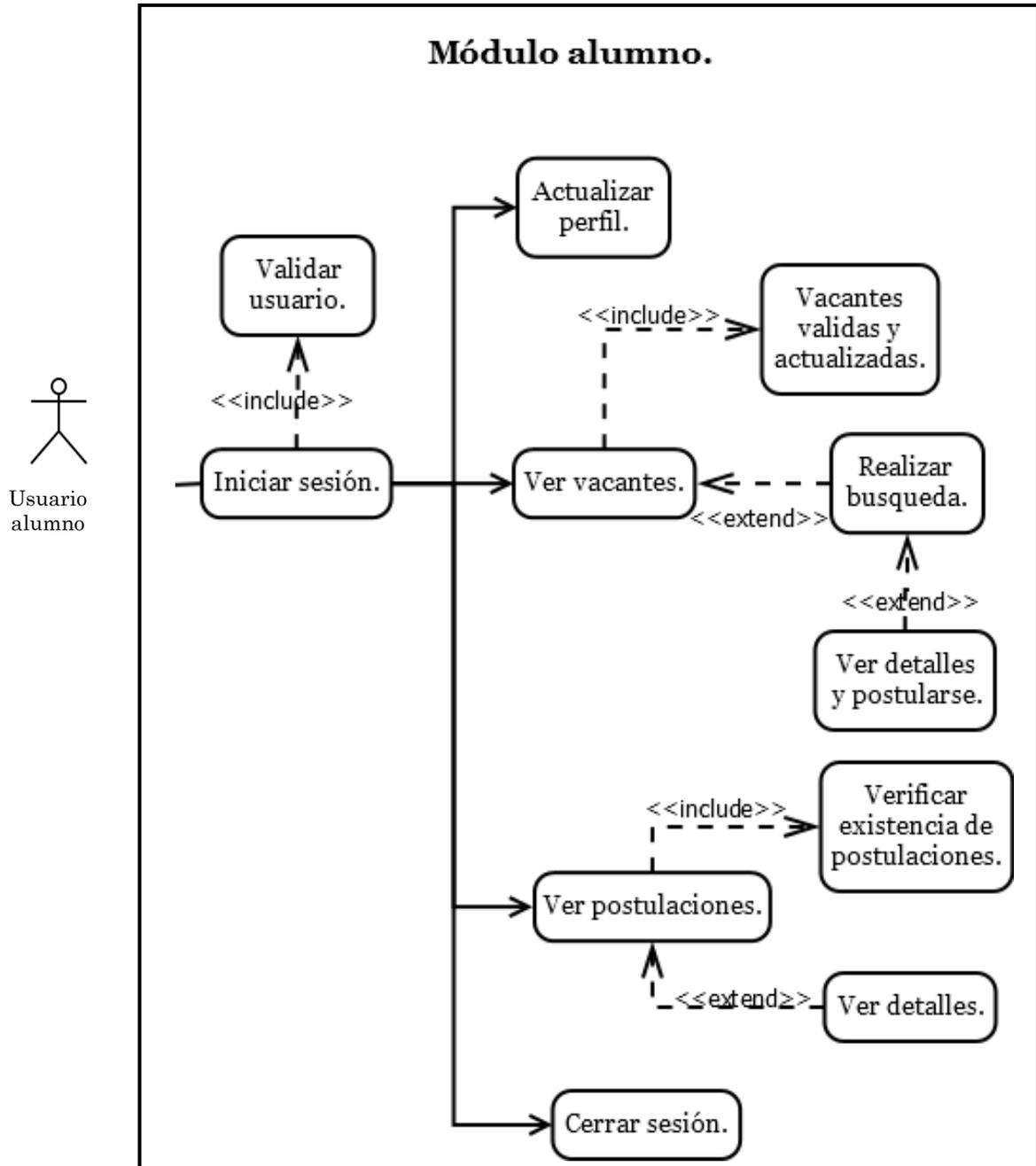


Figura 3.2– Diagrama de caso de uso del módulo alumno.

Módulo empresa: El módulo empresa permitirá actualiza información de la empresa, registrar vacantes, ver los estados de sus vacantes, realizar postulaciones y buscar candidatos.

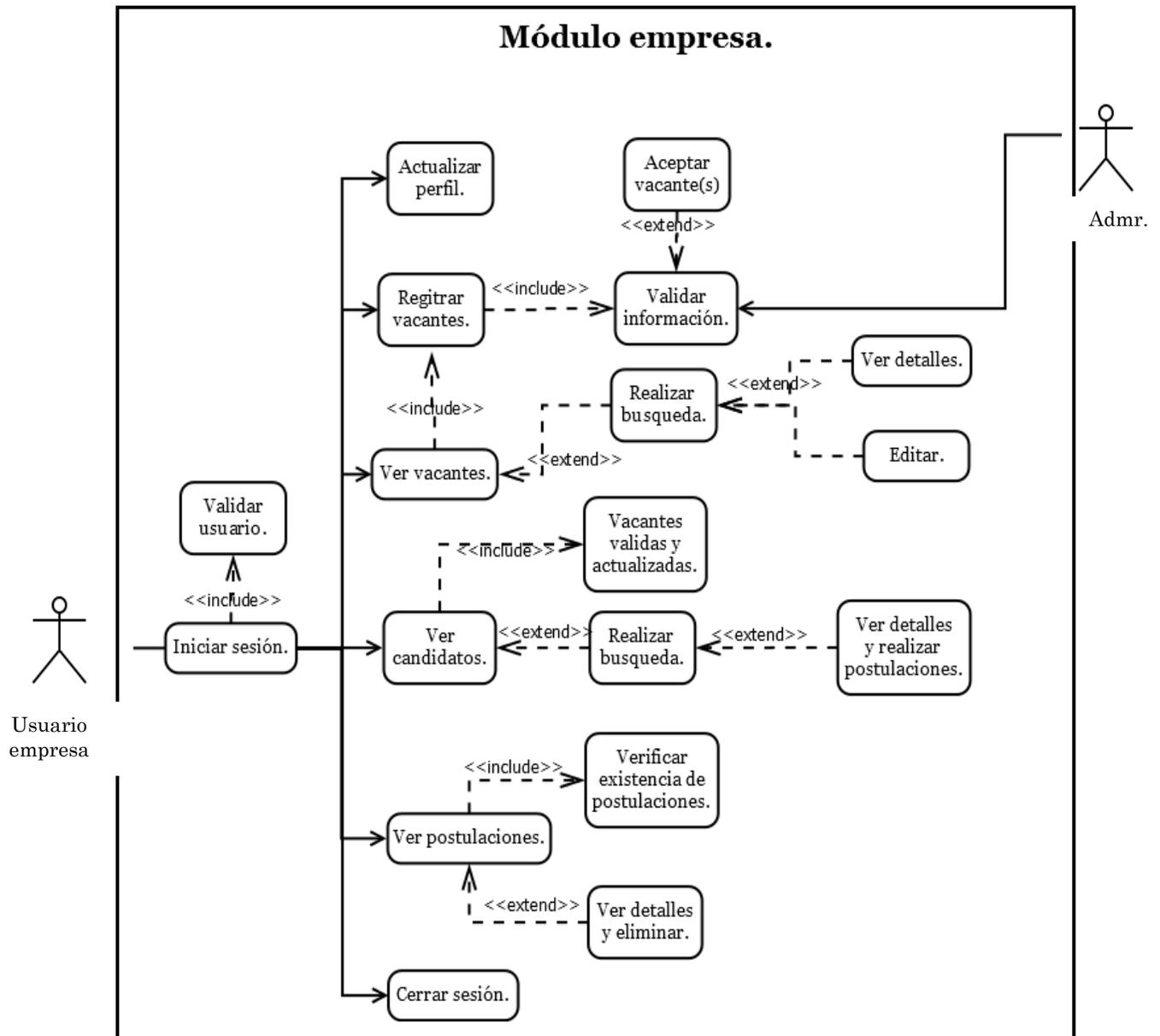


Figura 3.3– Diagrama de caso de uso del módulo empresa.

El módulo administrador: Permitirá aceptar empresas, validar vacantes, gestionar información, registrar nuevos administradores y realizar reportes.

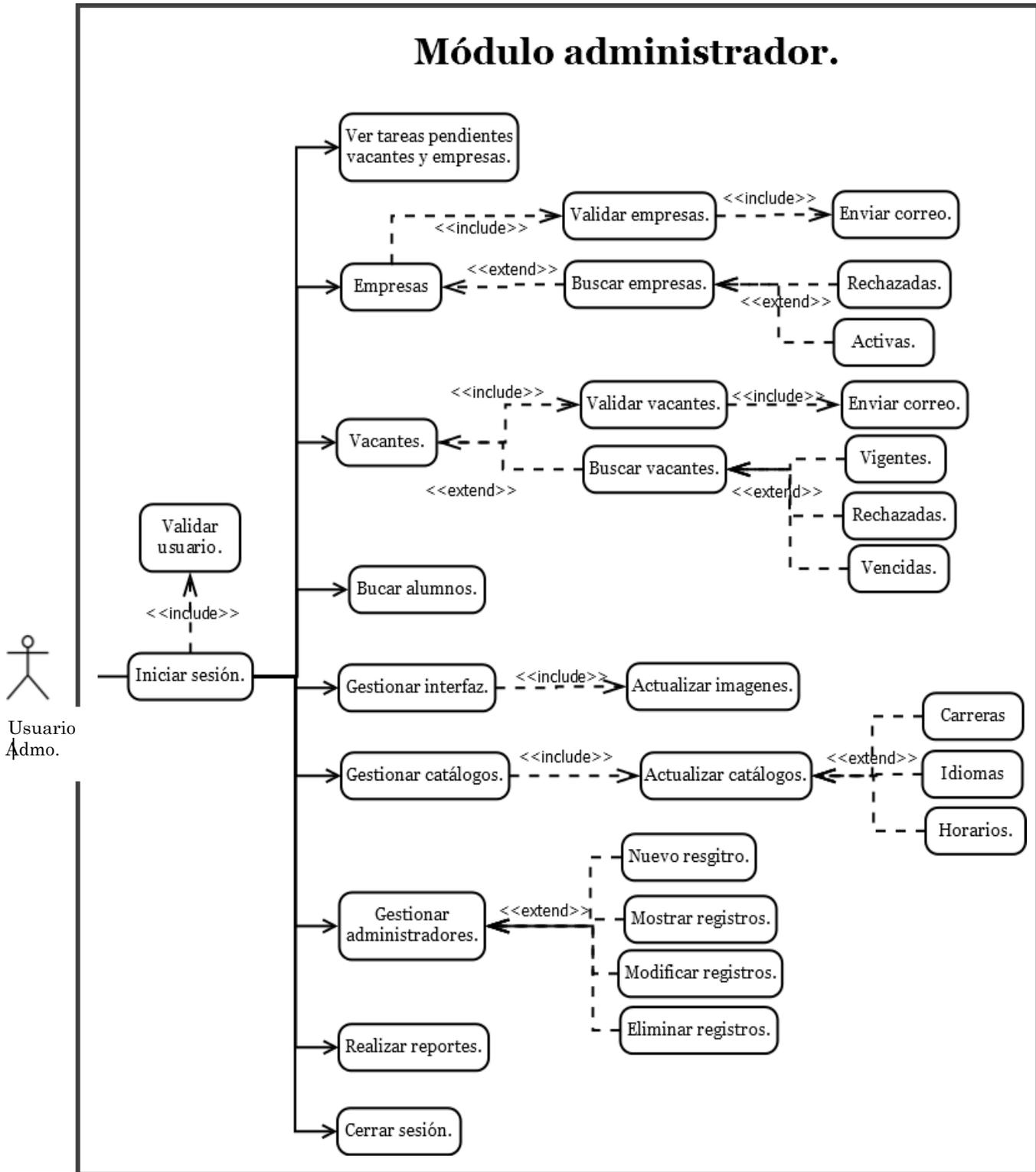


Figura 3.4– Diagrama de caso de uso del módulo administrador.

3.3.2- Diagrama de actividades.

Se incluyen diagramas de actividades para describir las funcionalidades que implementa el sistema de bolsa de trabajo en cada uno de sus módulos.

Módulo inicio.

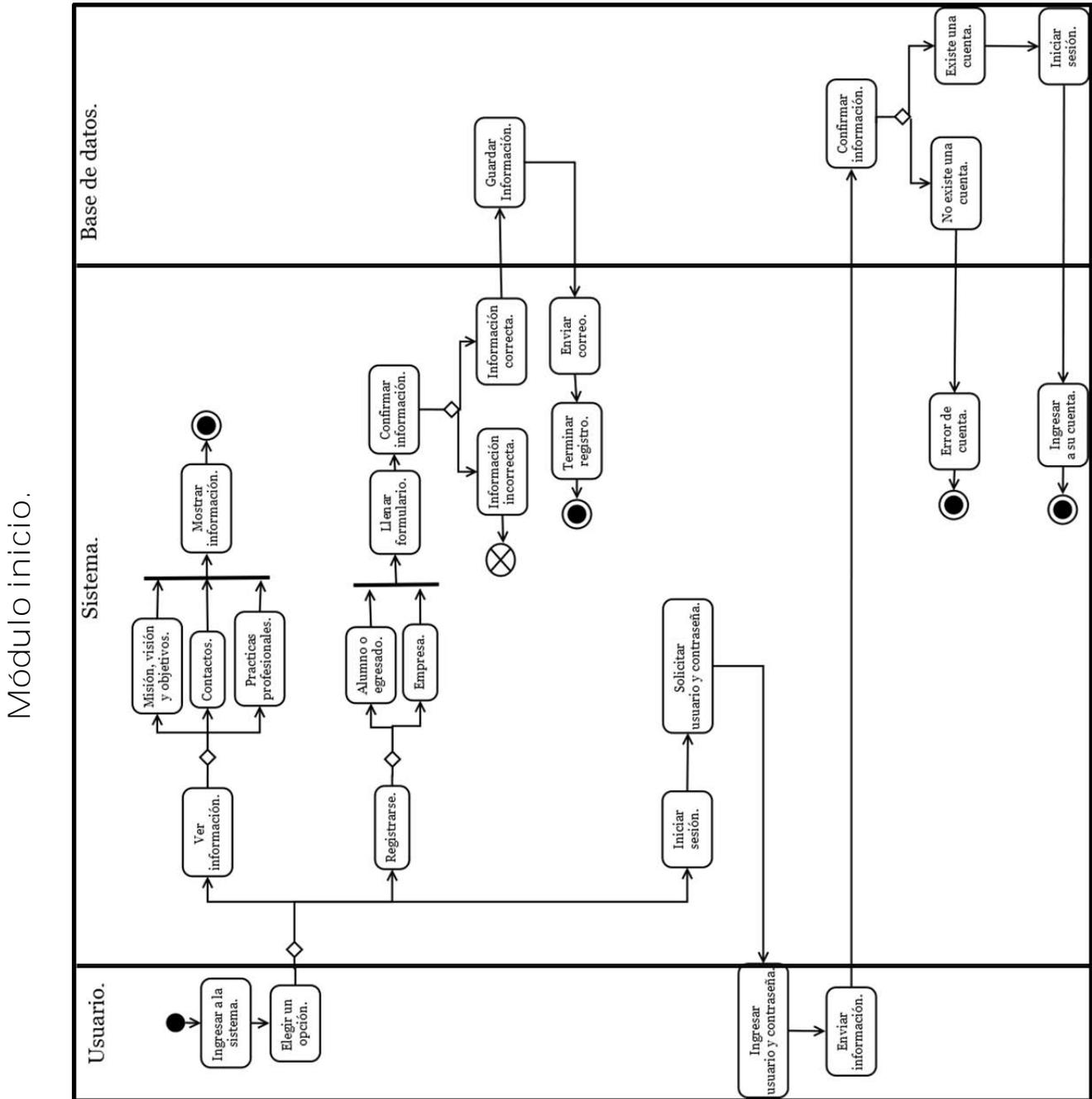


Figura 3.5– Diagrama de caso de uso del módulo inicio.

Empresa-Perfil

Módulo empresa.

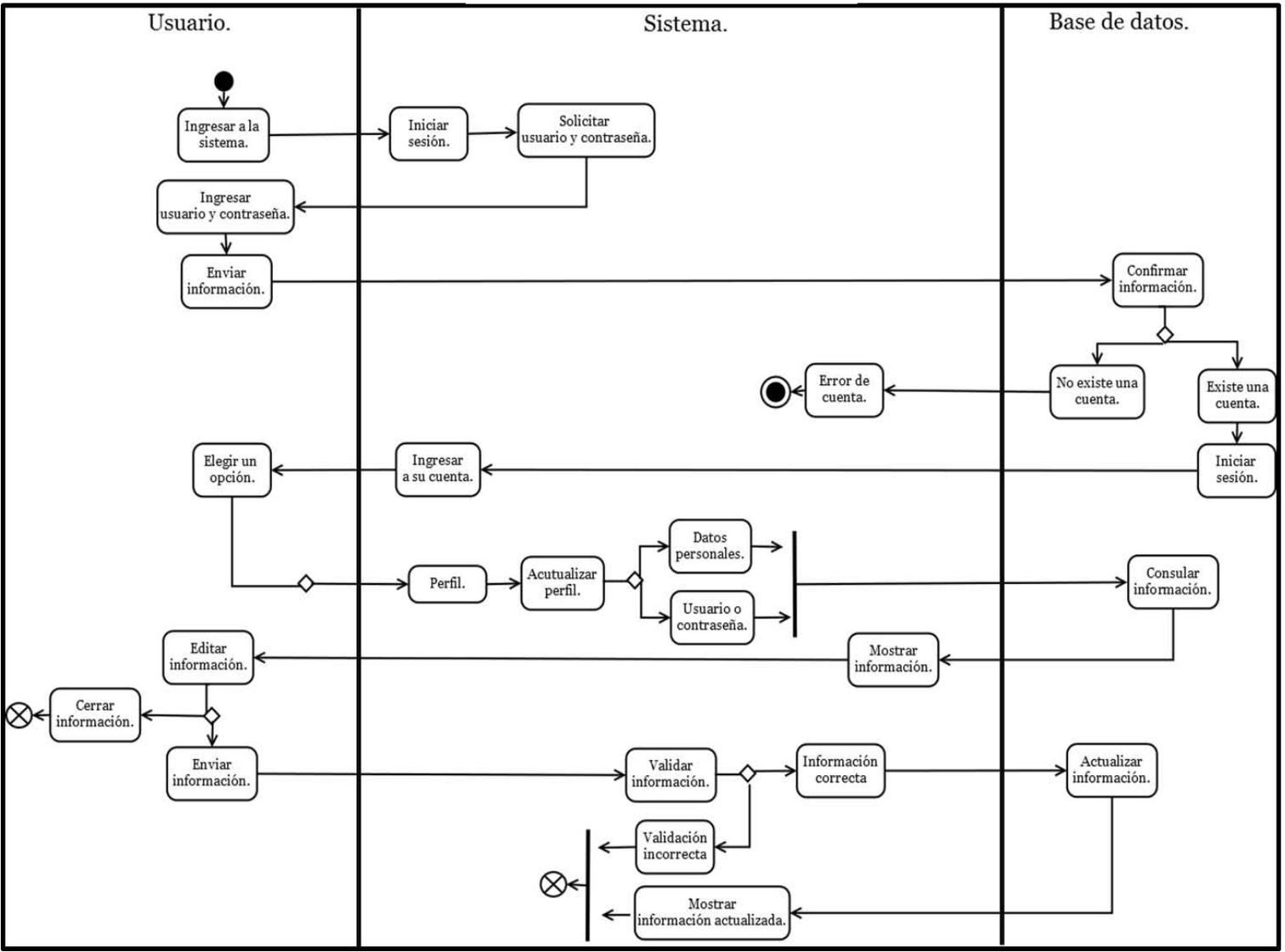


Figura 3.6– Diagrama de actividades del módulo empresa-perfil.

Empresa-Vacantes.

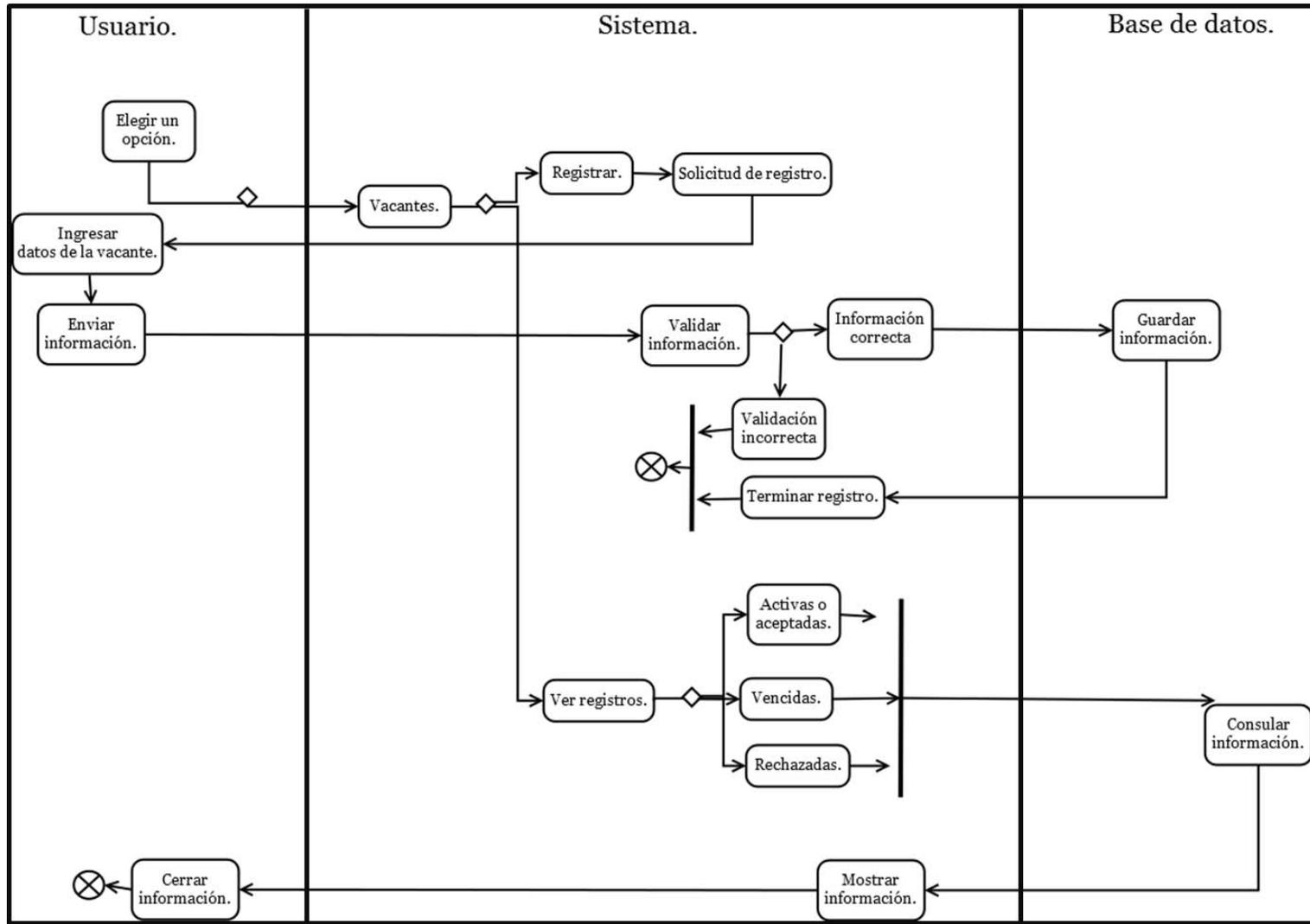


Figura 3.6.1 – Diagrama de actividades del módulo empresa-vacantes.

Empresa-Postulaciones.

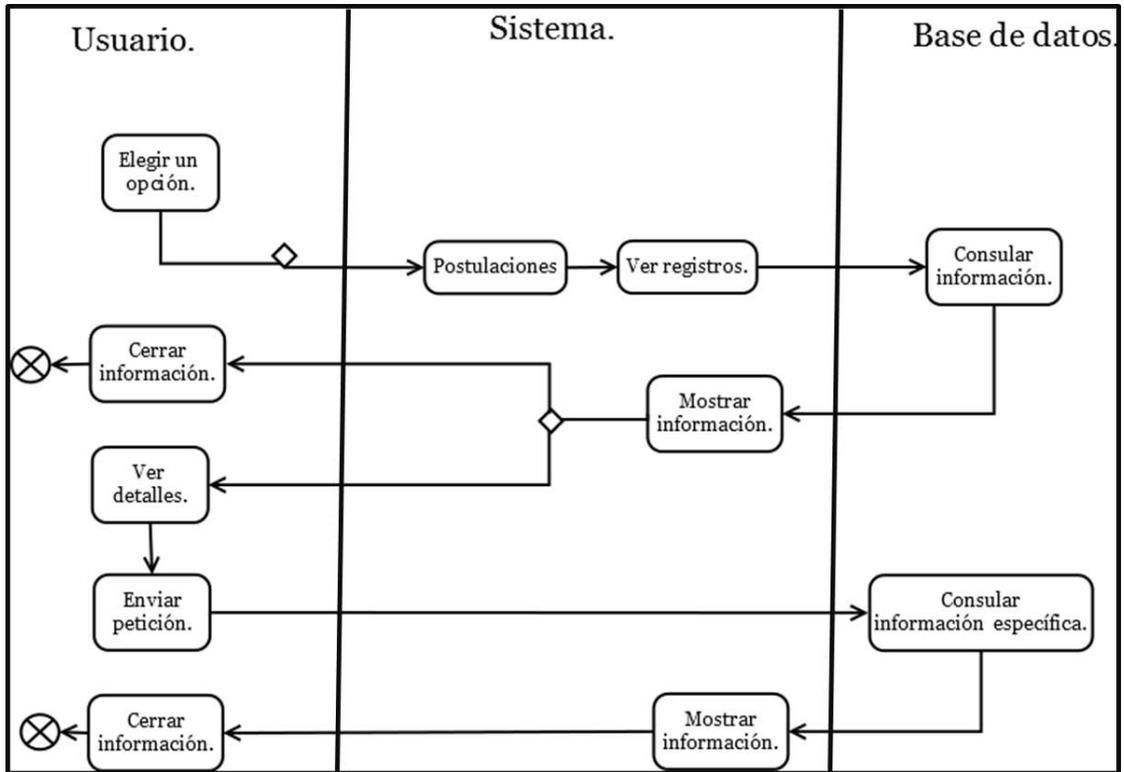


Figura 3.6.2- Diagrama de actividades del módulo empresa-

Empresa-Cerrar sesión.

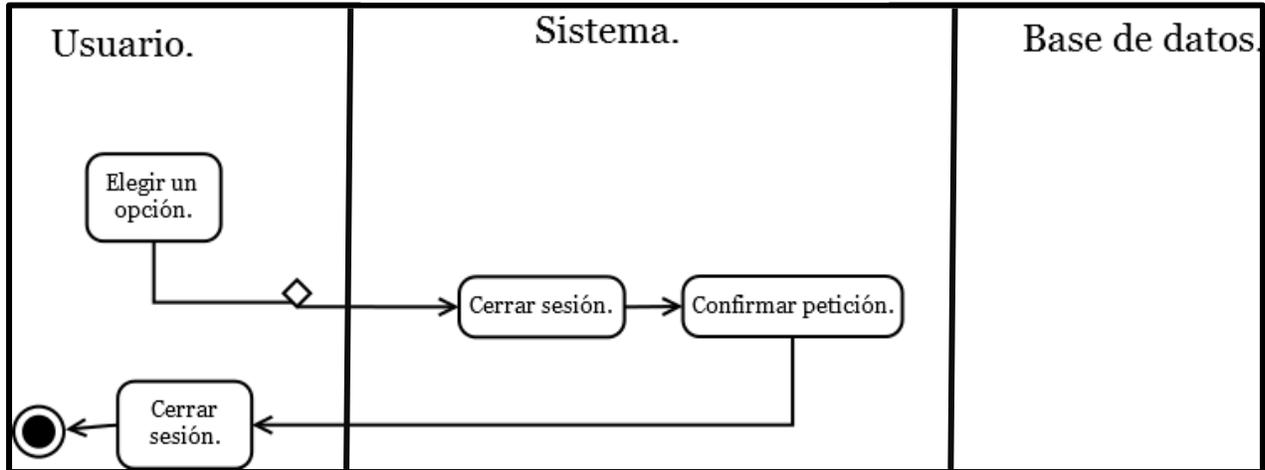


Figura 3.6.4– Diagrama de actividades del módulo empresa-cerrar sesión

Módulo Alumno.

Alumno-Perfil

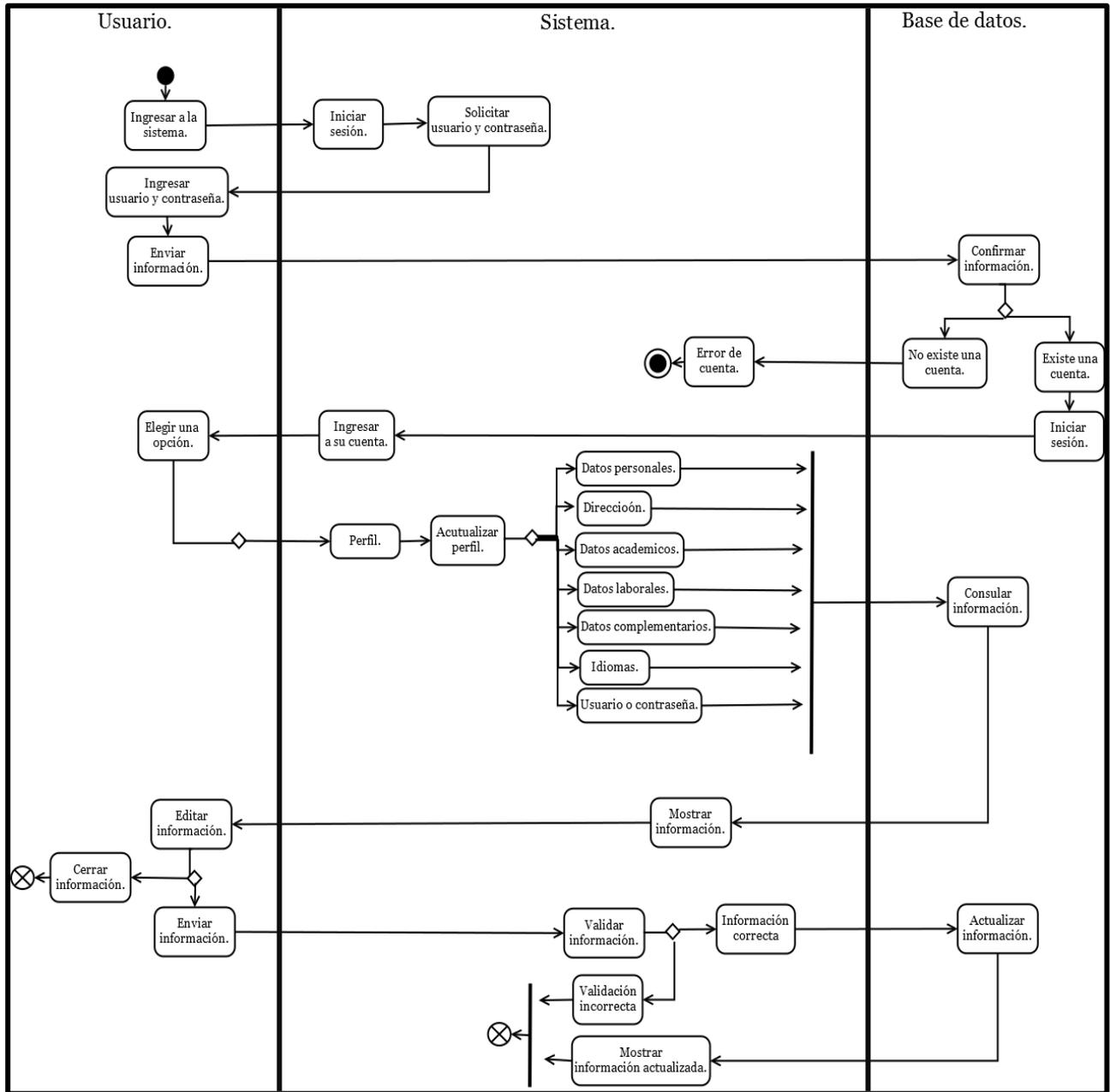


Figura 3.7– Diagrama de actividades del módulo alumno-perfil.

Alumno-Vacantes.

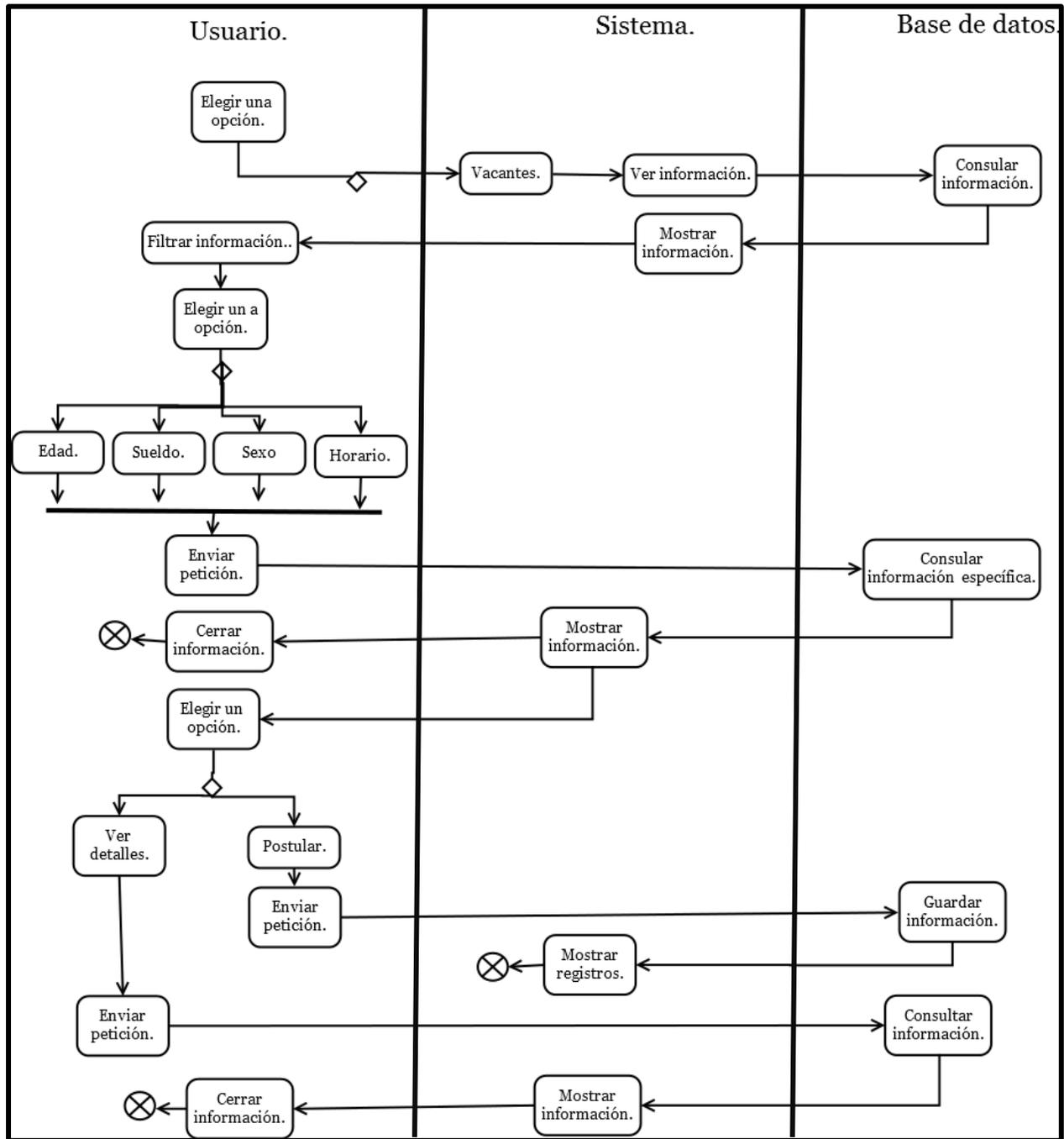


Figura 3.7.1– Diagrama de actividades del módulo alumno-vacantes.

Alumno-Postulaciones.

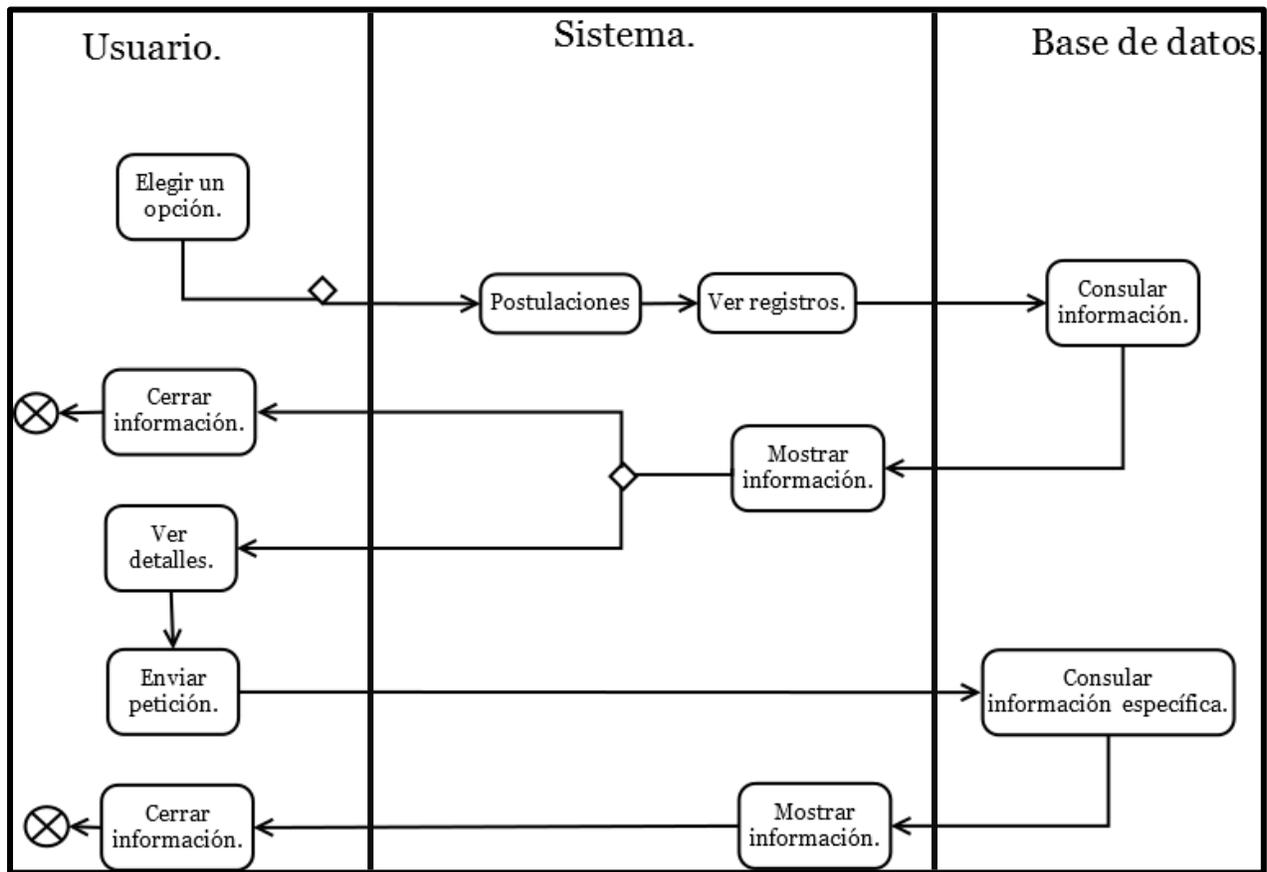


Figura 3.7.2– Diagrama de actividades del módulo alumno-postulaciones.

Alumno-Cerrar sesión.

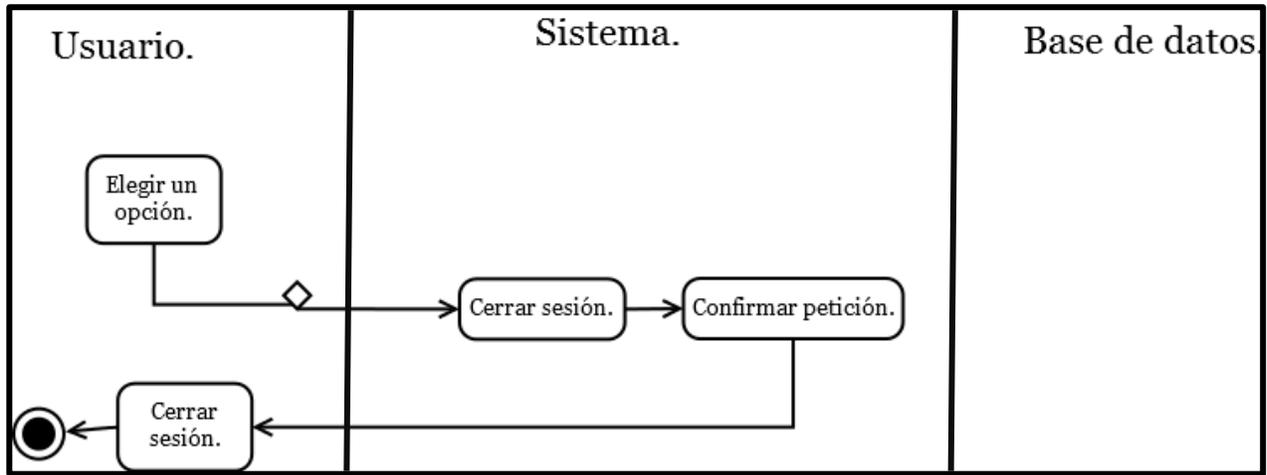


Figura 3.7.3– Diagrama de actividades del módulo alumno-cerrar sesión

Módulo Administrador.

Administrador-Empresas.

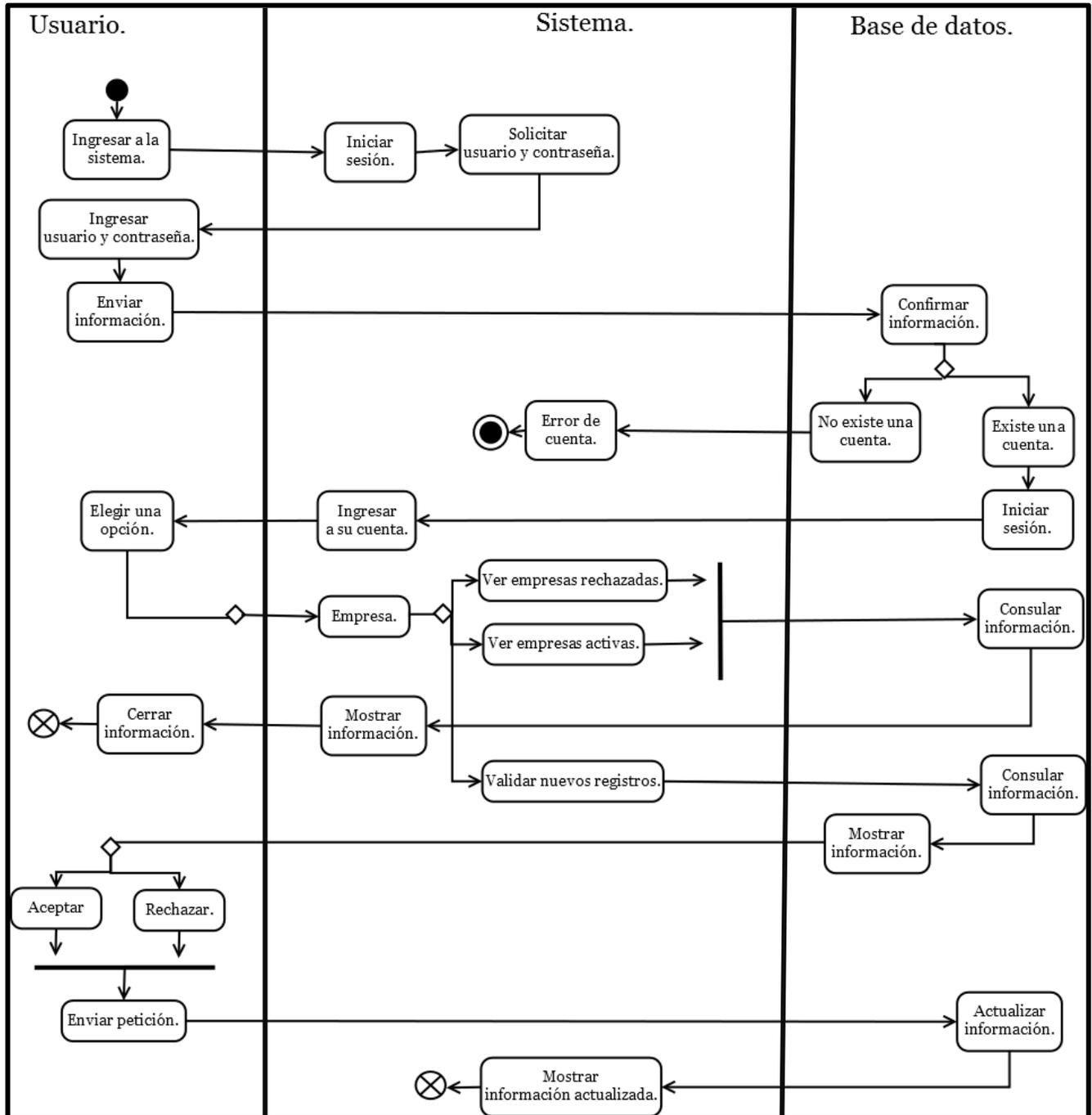


Figura 3.8– Diagrama de actividades del módulo administrador-empresas.

Administrador-Vacantes.

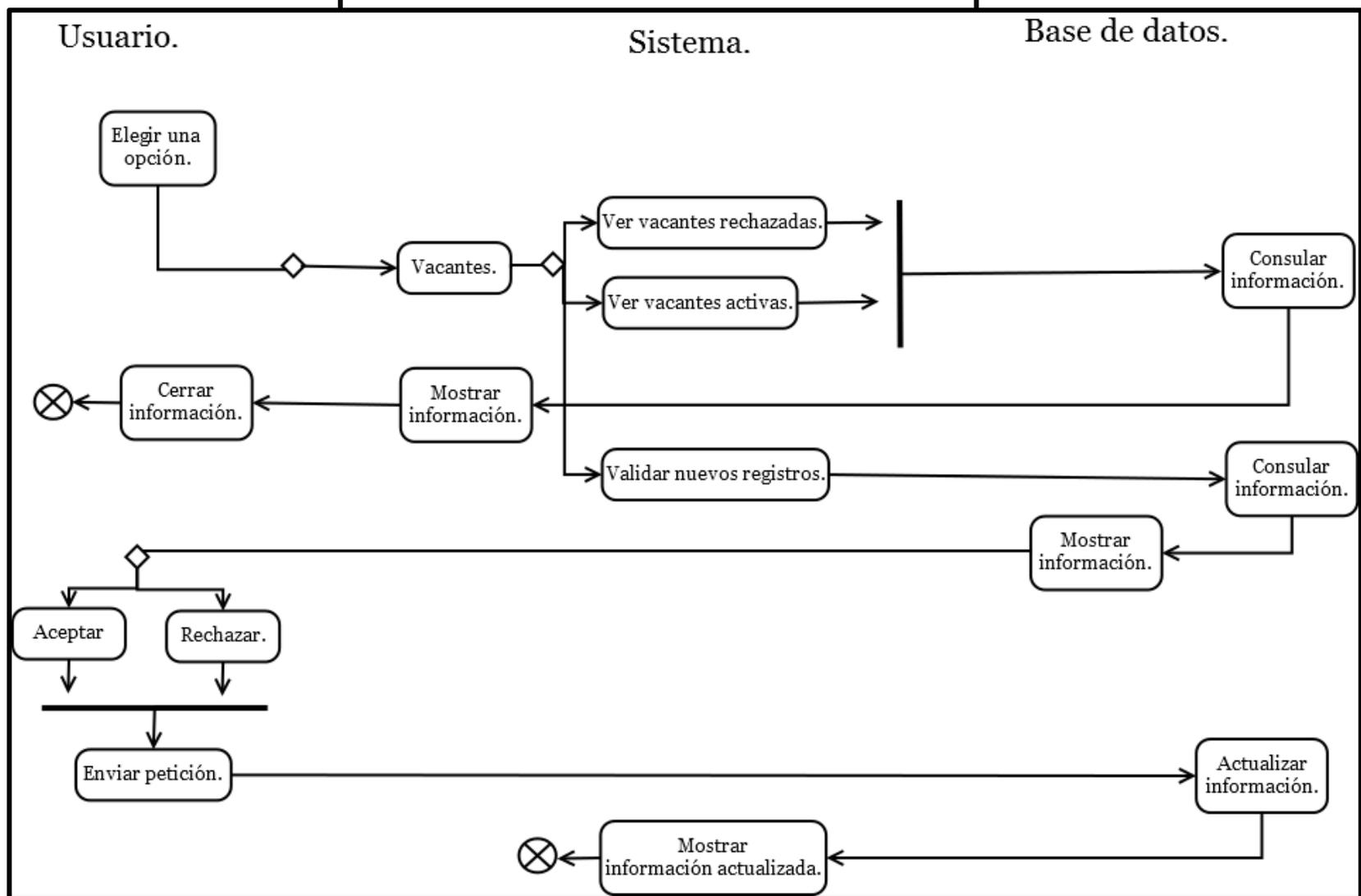


Figura 3.8.1– Diagrama de actividades del módulo administrador-vacantes.

Administrador-Alumnos.

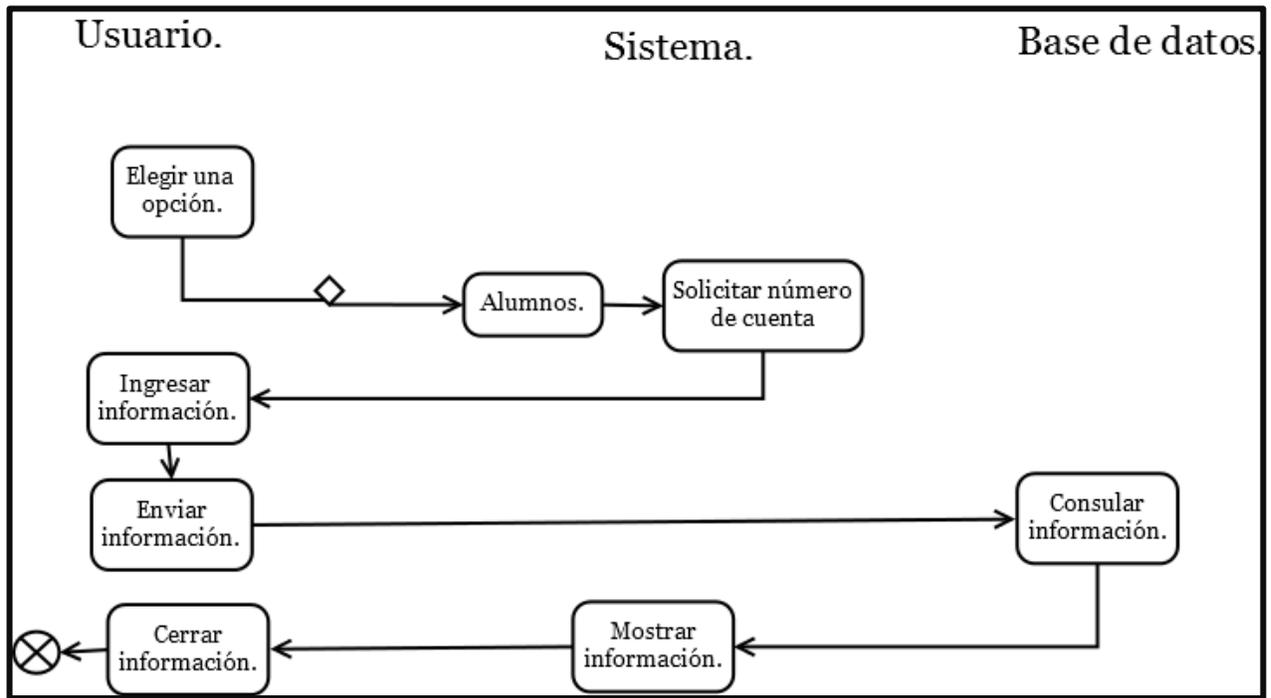


Figura 3.8.2– Diagrama de actividades del módulo administrador-alumnos.

Administrador-Interfaz.

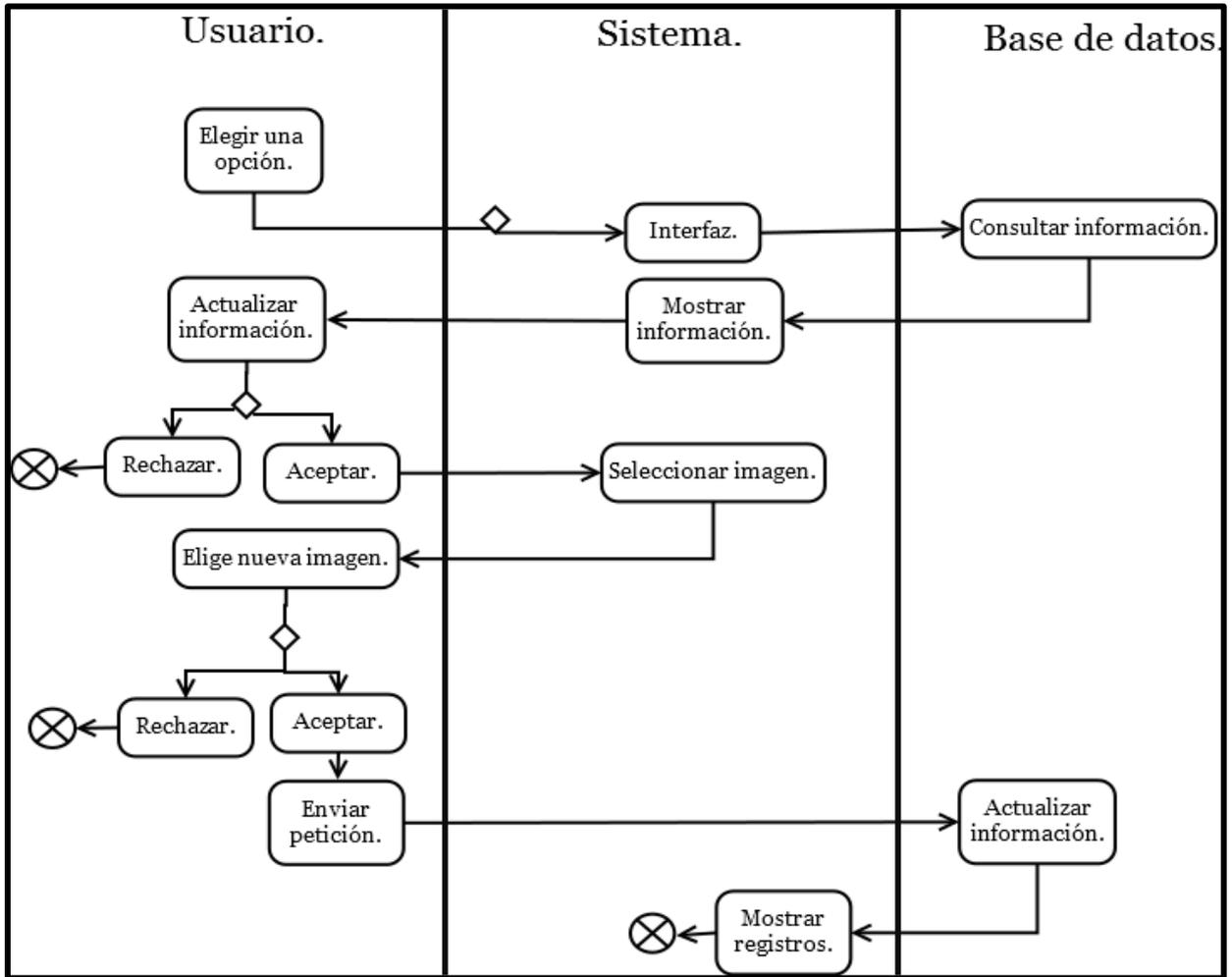


Figura 3.8.3– Diagrama de actividades del módulo administrador-interfaz.

Administrador-Catálogos.

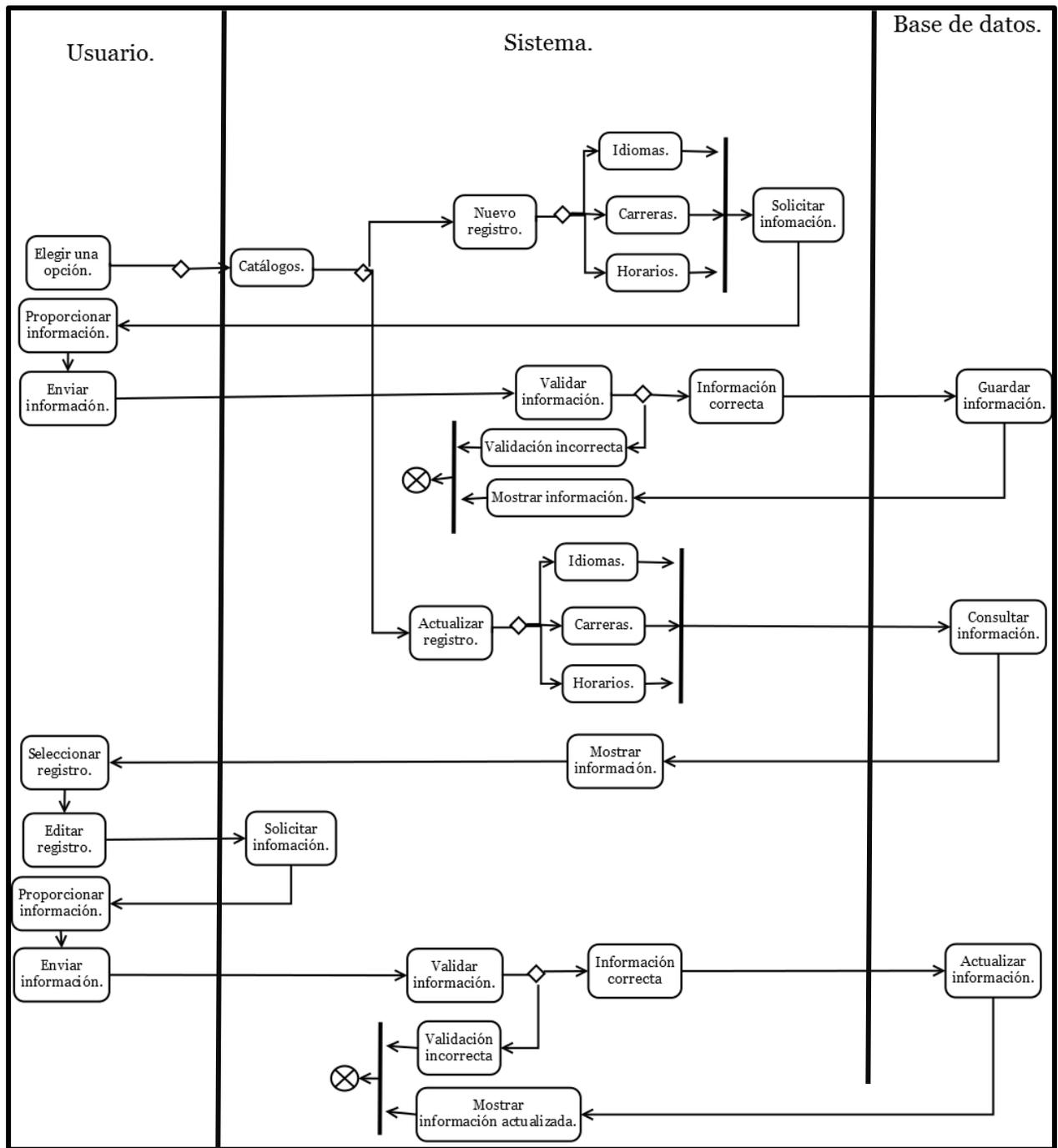


Figura 3.8.4– Diagrama de actividades del módulo administrador-catálogos.

Administrador-Cuentas de usuario.

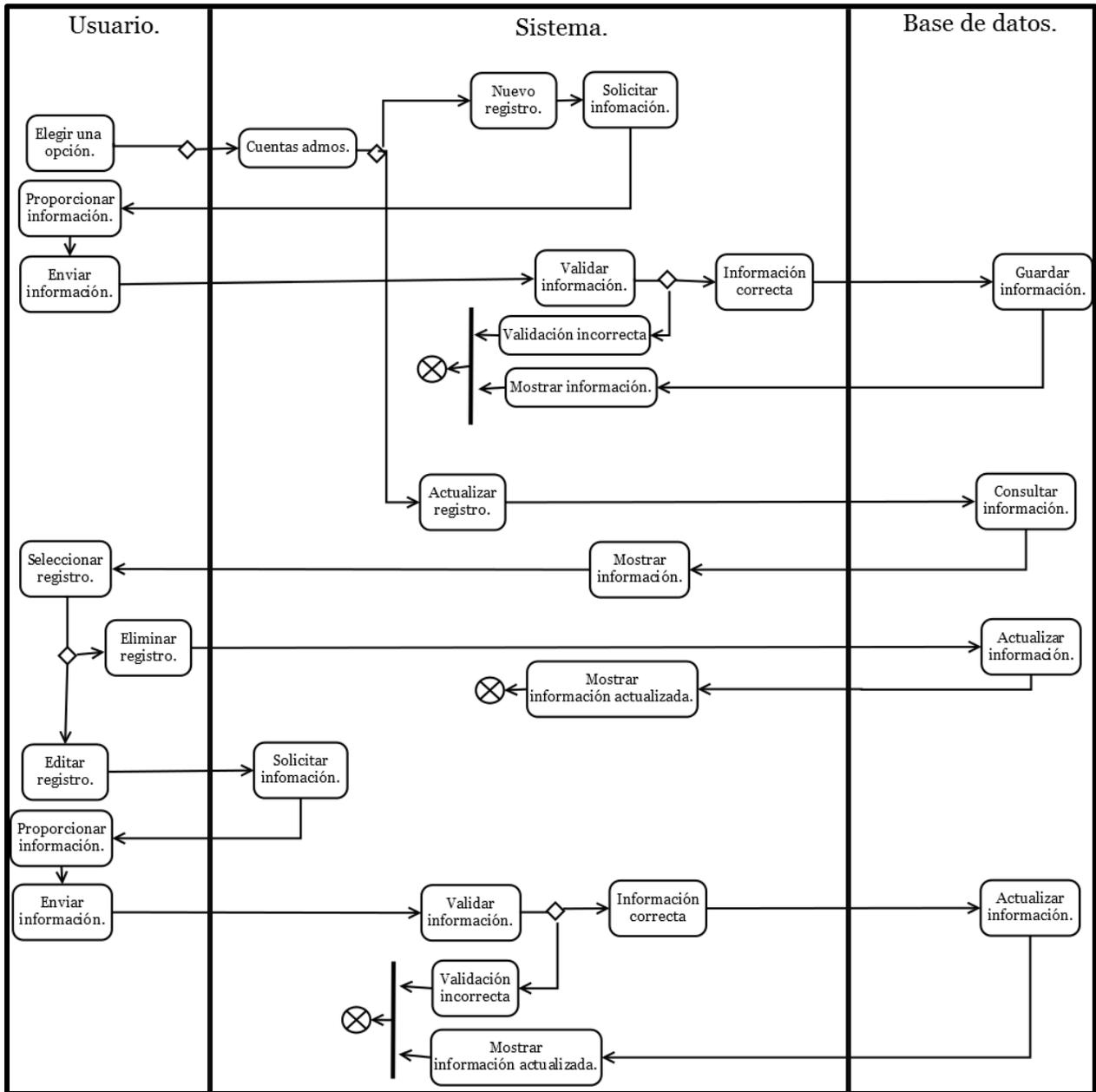


Figura 3.8.5– Diagrama de actividades del módulo administrador-cuentas de usuario.

Administrador-Reportes.

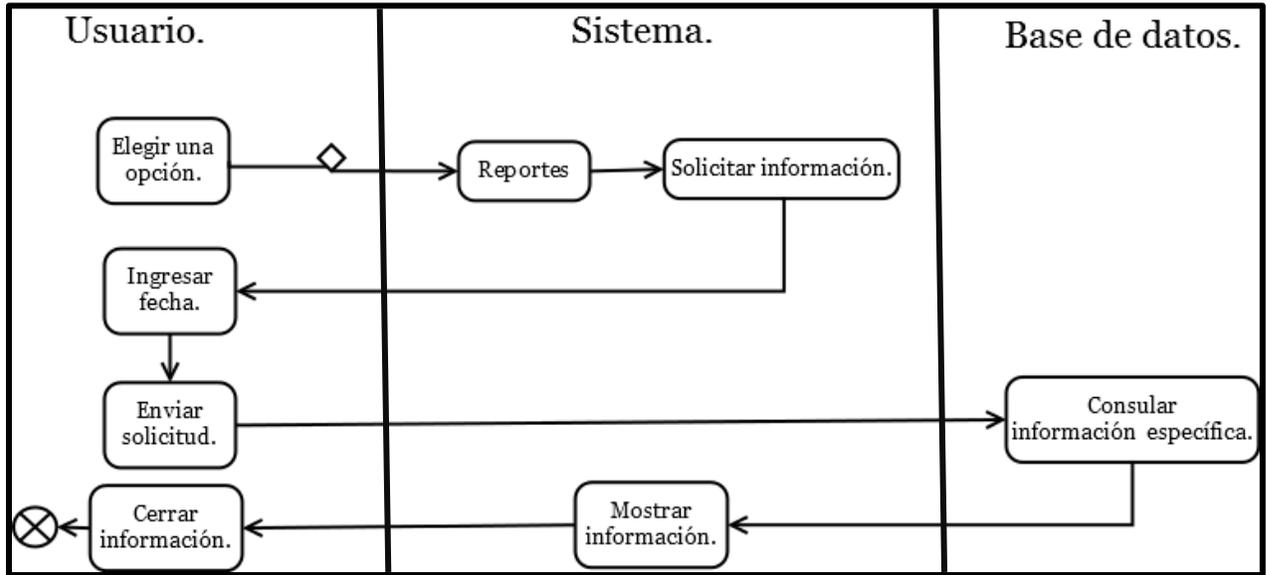


Figura 3.8.6– Diagrama de actividades del módulo administrador-reportes.

Administrador-Cerrar sesión.

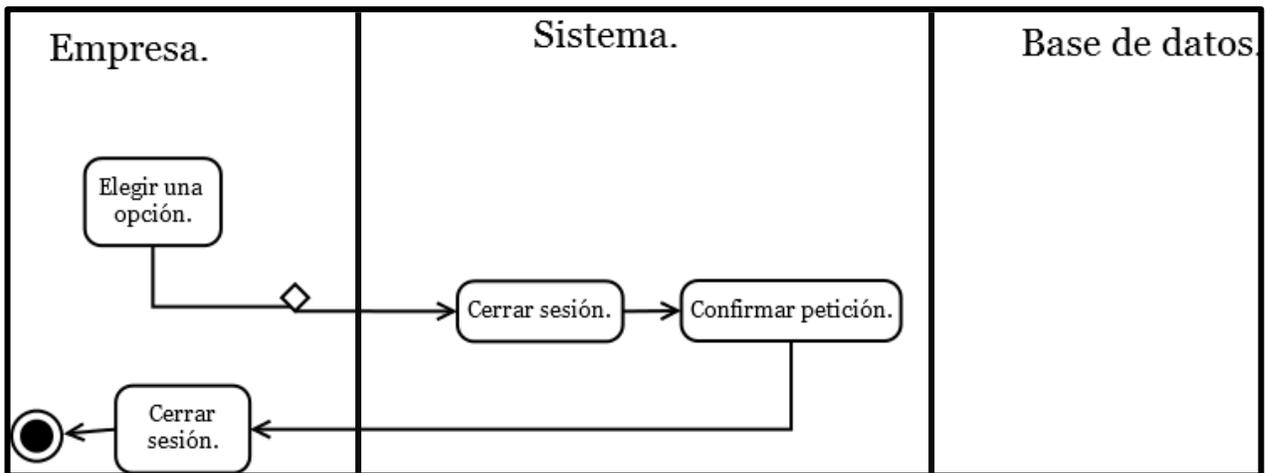


Figura 3.8.7– Diagrama de actividades del módulo administrador-cerrar sesión.

3.3.3- Diagrama de clases.

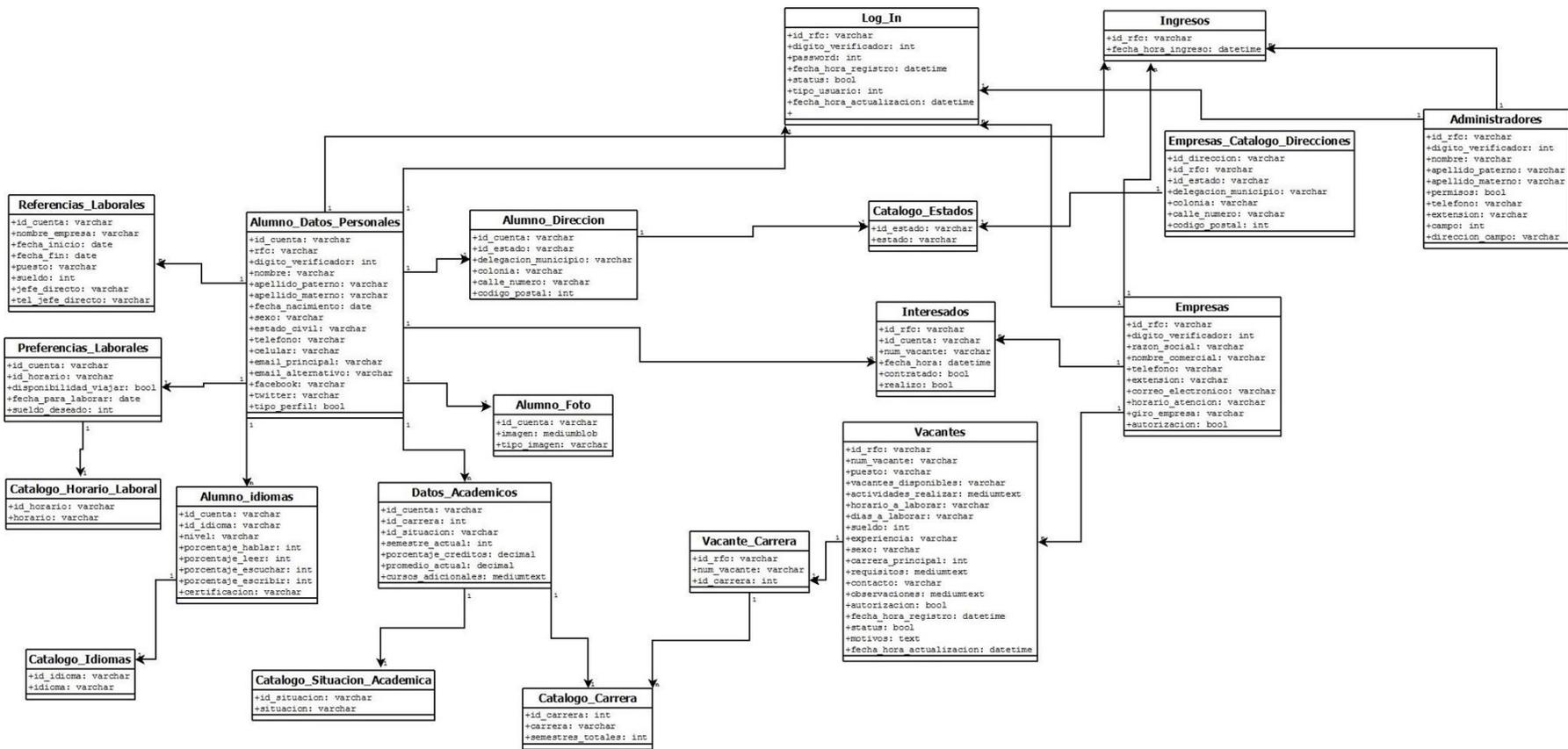


Figura 3.9– Diagrama de clases del sistema bolsa de trabajo.

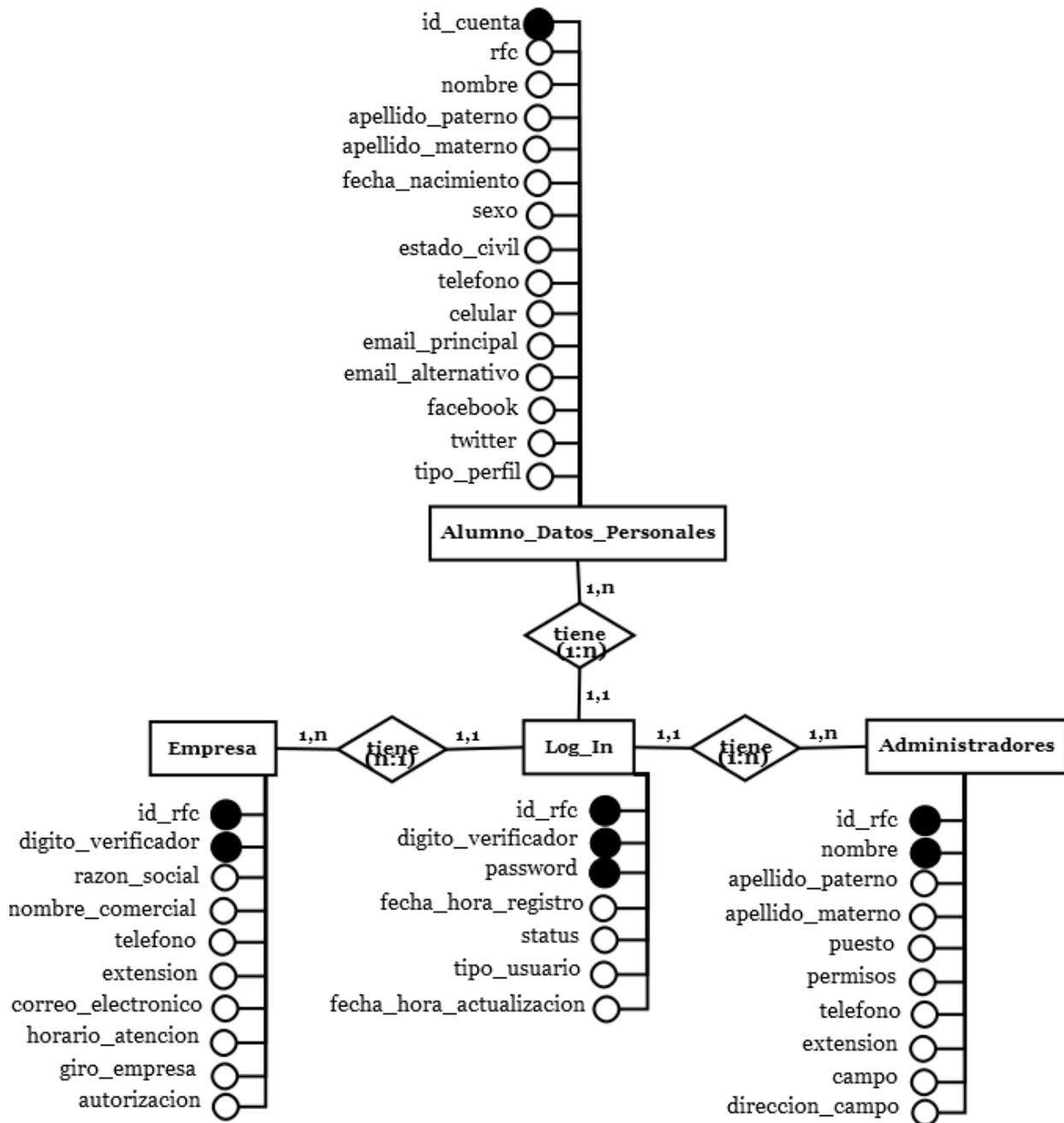
3.4- Diseño de la base de datos.

Siguiendo el modelo de procesos de ingeniería Web planteado en el capítulo 1 la construcción de la arquitectura de software es la cuarta etapa donde definimos y diseñamos la base de datos que definirá el desarrollo del software. En los siguientes capítulos se lleva a cabo la construcción de la base de datos con referencia al apartado 1.3.2 del capítulo 1 donde se establecen los fundamentos para el diseño de la base, el diagrama entidad-relación y el diccionario de datos.

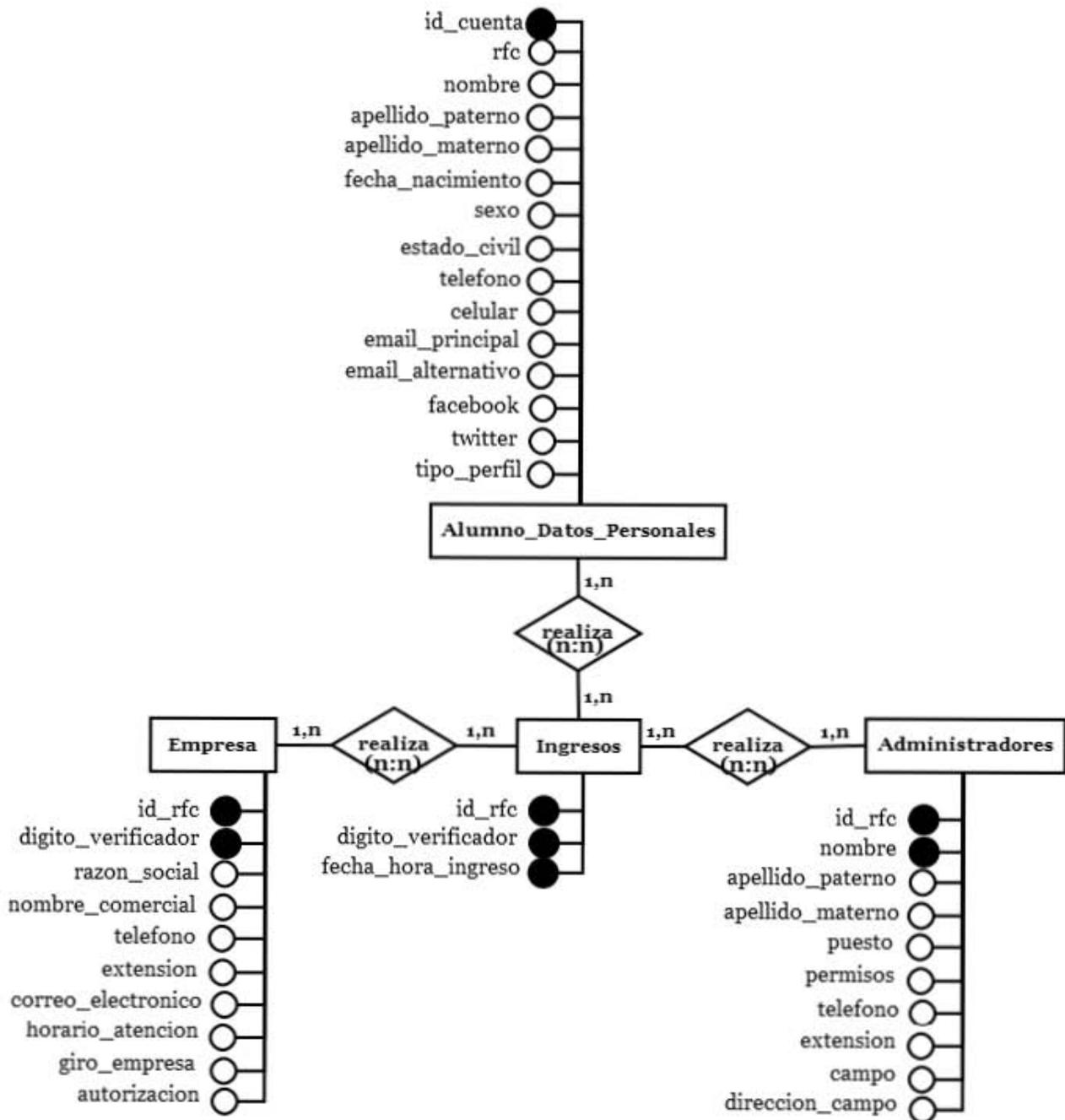
3.4.1- Diagrama de E-R.

De acuerdo al apartado 1.3.2.1 el diseño conceptual de la base de datos comprende la representación gráfica acerca de la información recolectada de los requisitos de la organización y del usuario, utilizando un lenguaje que describe el contenido de la base de datos. A continuación, se muestra los diagramas de E-R que describen la base de datos de bolsa de trabajo.

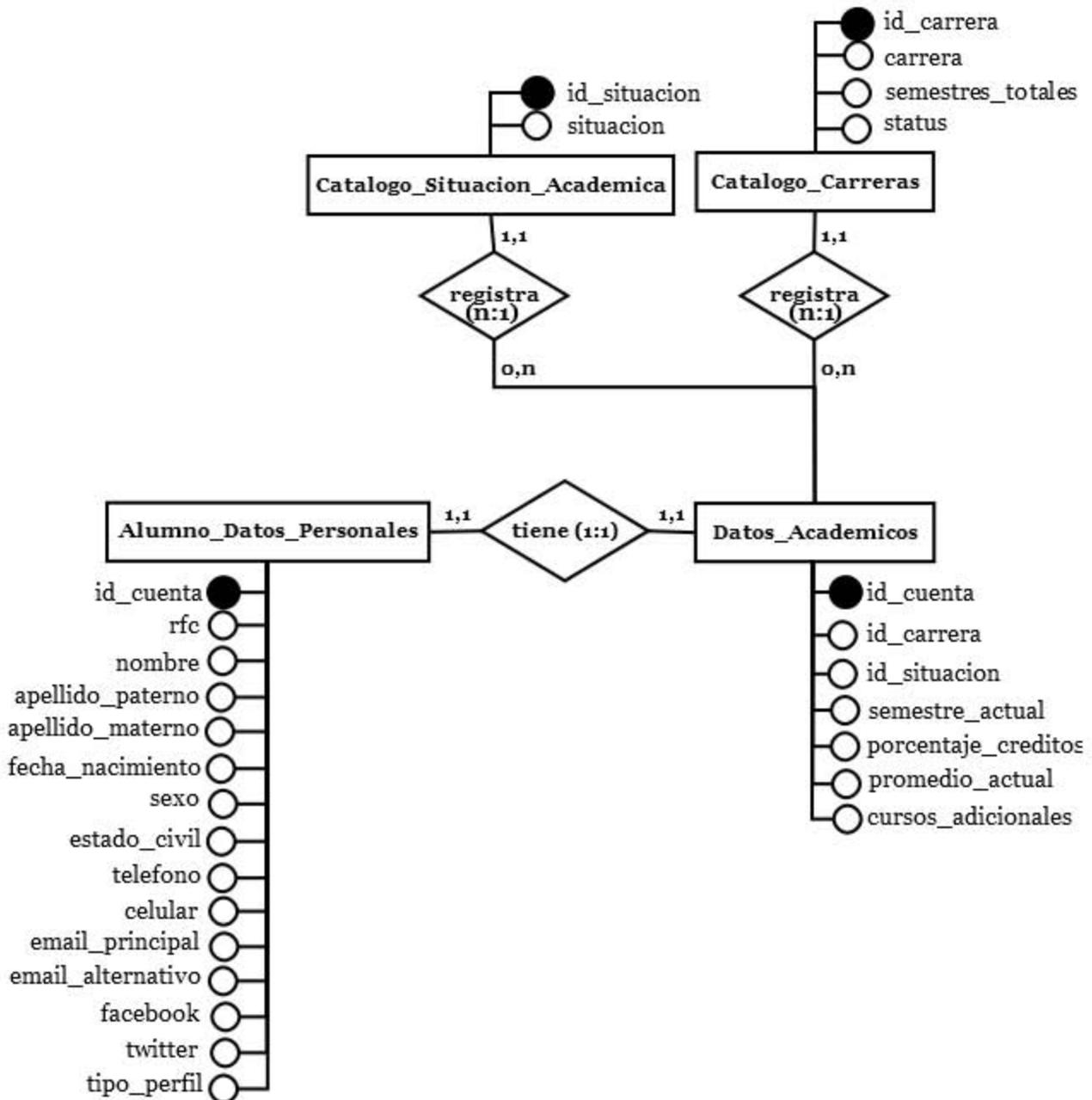
Login: La entidad login se relaciona con las entidades empresa, administradores y alumno_datos_personales ya que son los tipos de usuarios que pueden ingresar a la aplicación.



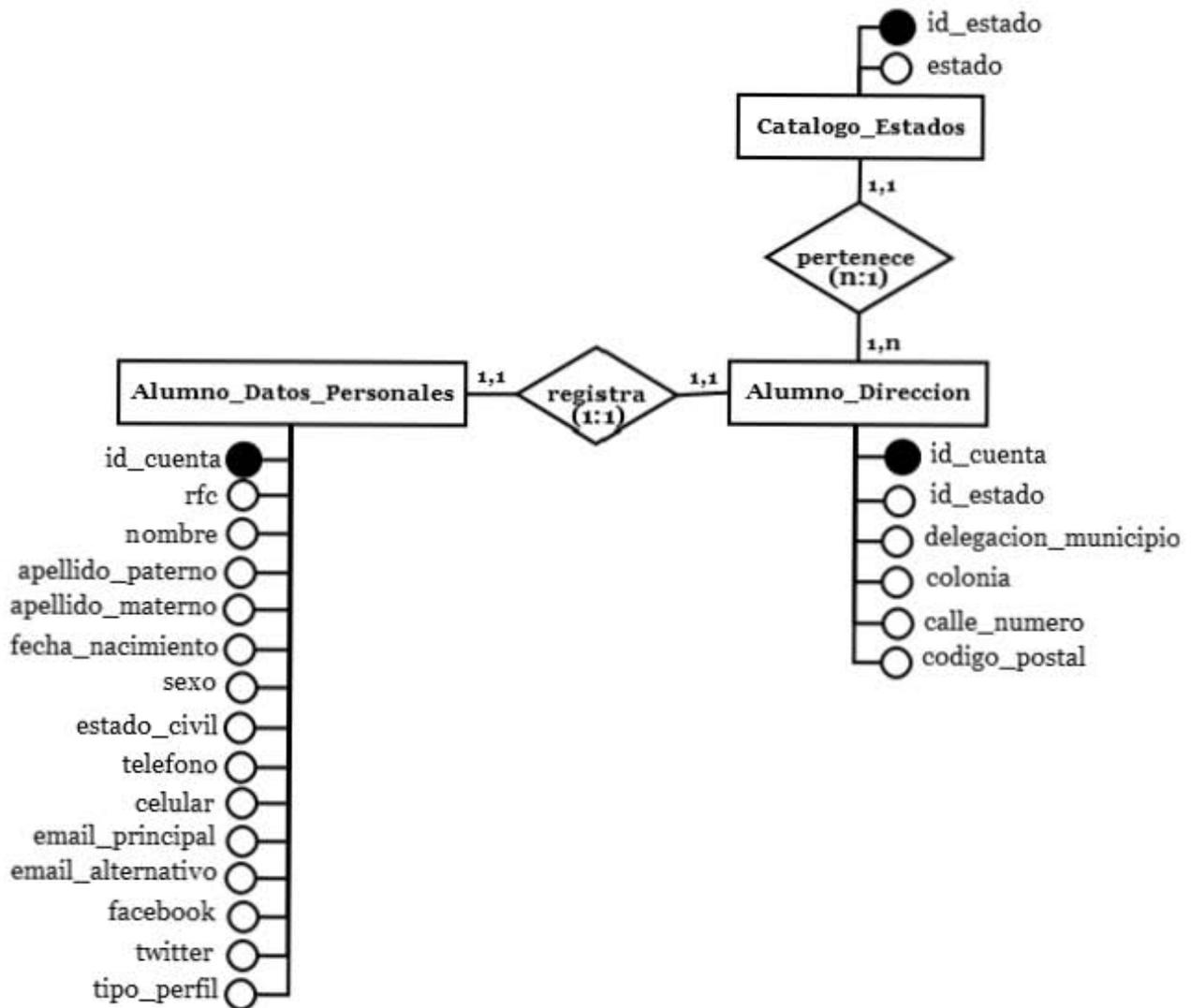
Ingresos: La entidad ingresos se relaciona con las entidades empresa, administradores y alumno_datos_personales ya que lleva el historial de las veces que ingresa un usuario.



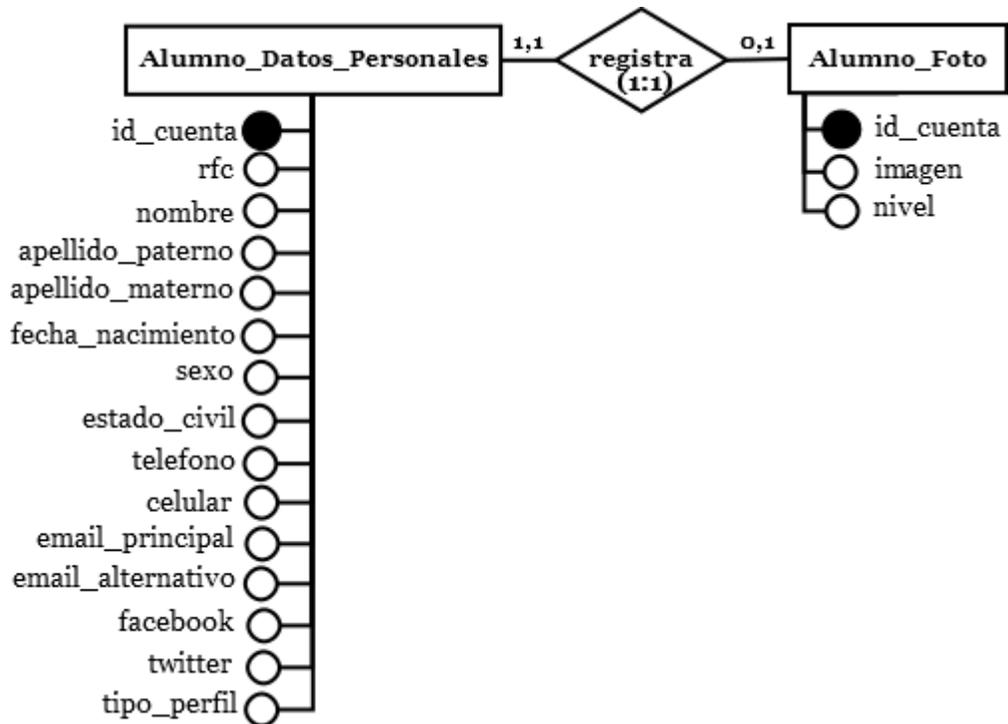
Alumno-Datos_Academicos: La entidad alumno_datos_academicos se relaciona con las entidades alumno_datos_personales, catalogo_carreras y catalogo_situacion_academica porque un alumno puede realizar la búsqueda de su carrera, así como su situación académica.



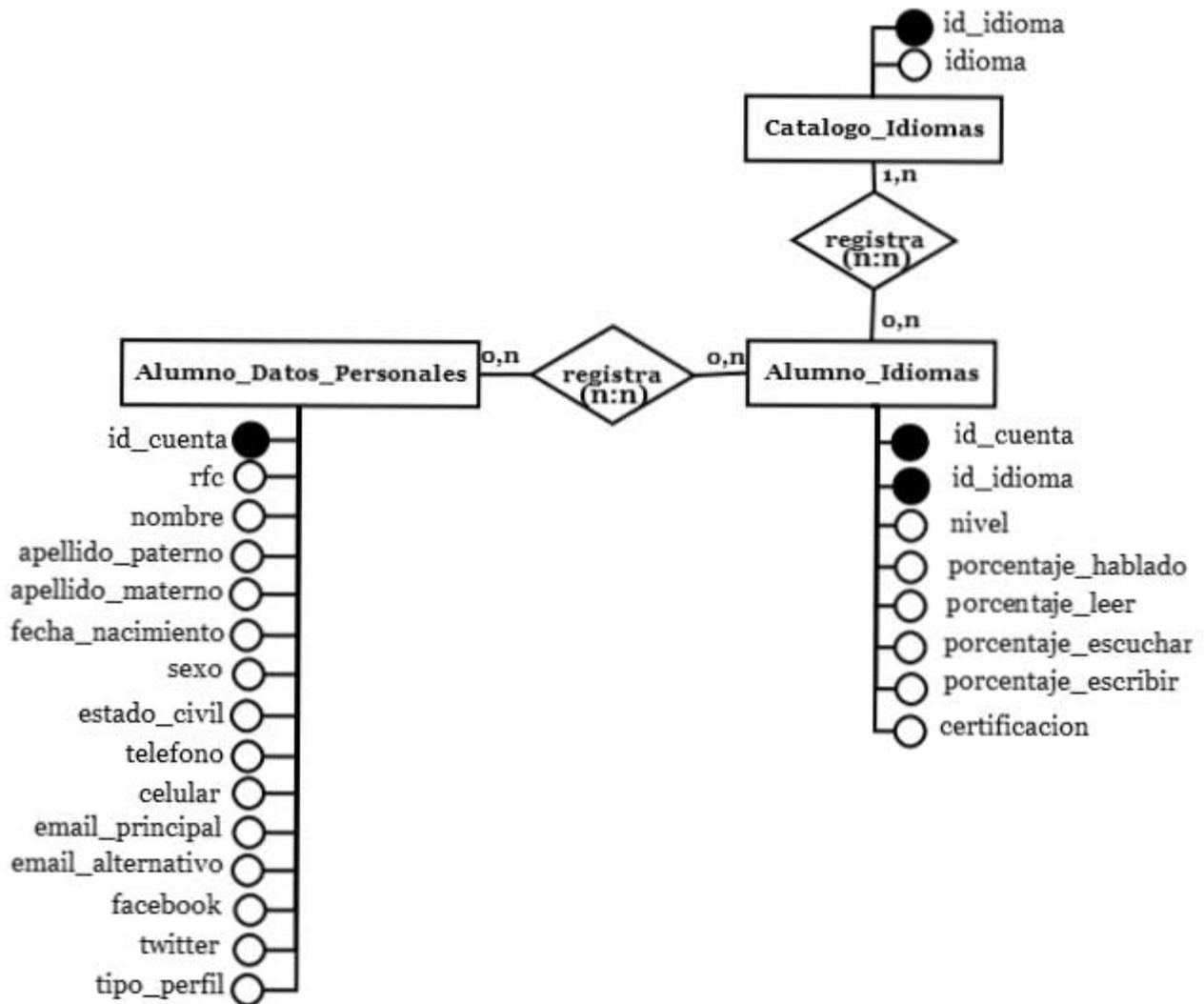
Alumno-Dirección: La entidad alumno_direccion se relaciona con las entidades alumno_datos_personales y catalogo_estados, ya que al estar relacionada con estados podrá pertenecer solo a un estado y ser parte de un alumno.



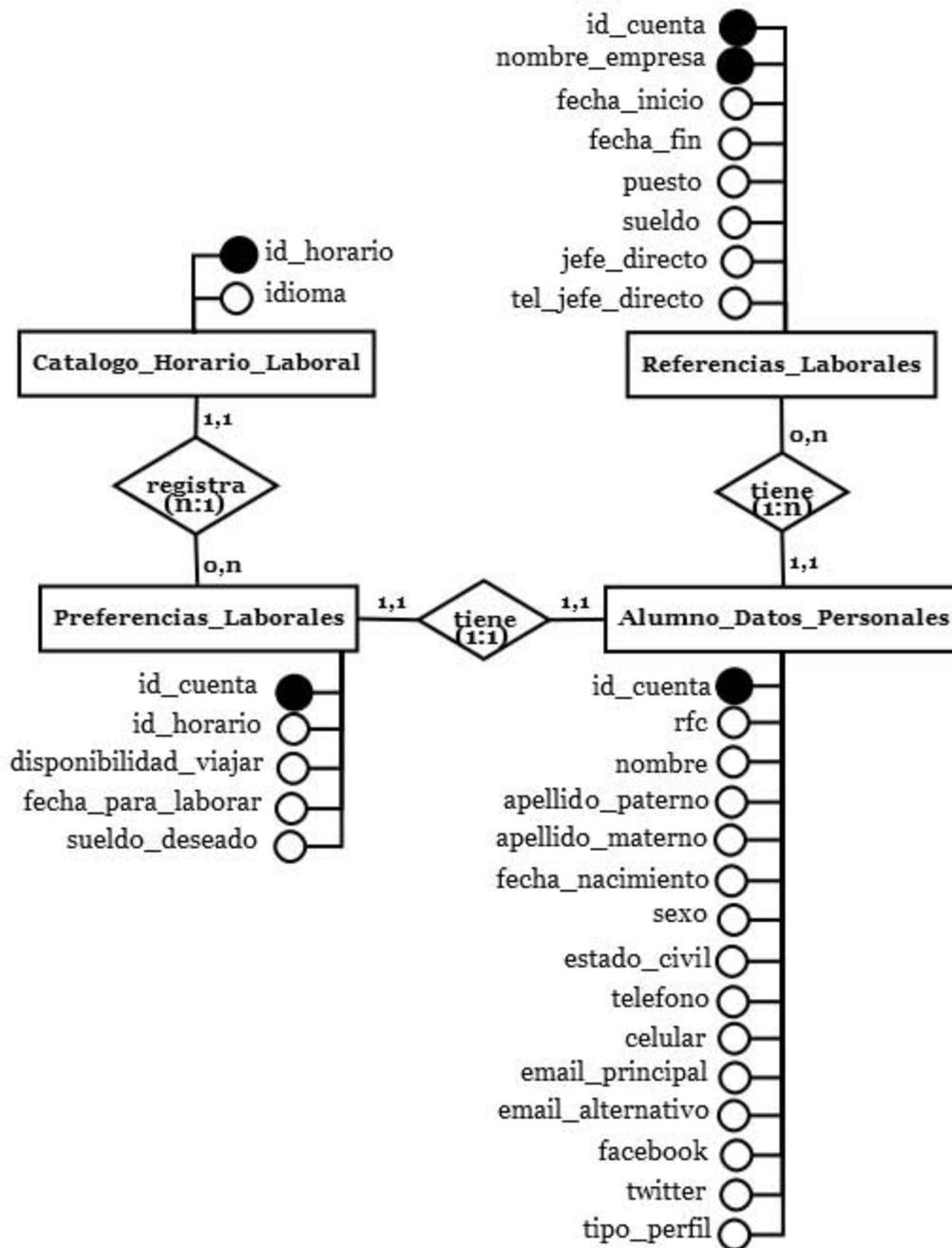
Alumno-Foto: La entidad alumno_foto se relaciona con las entidades alumno_datos_personales, ya que se podrá guardar la información fotográfica del alumno para poder generar su curriculum vitae.



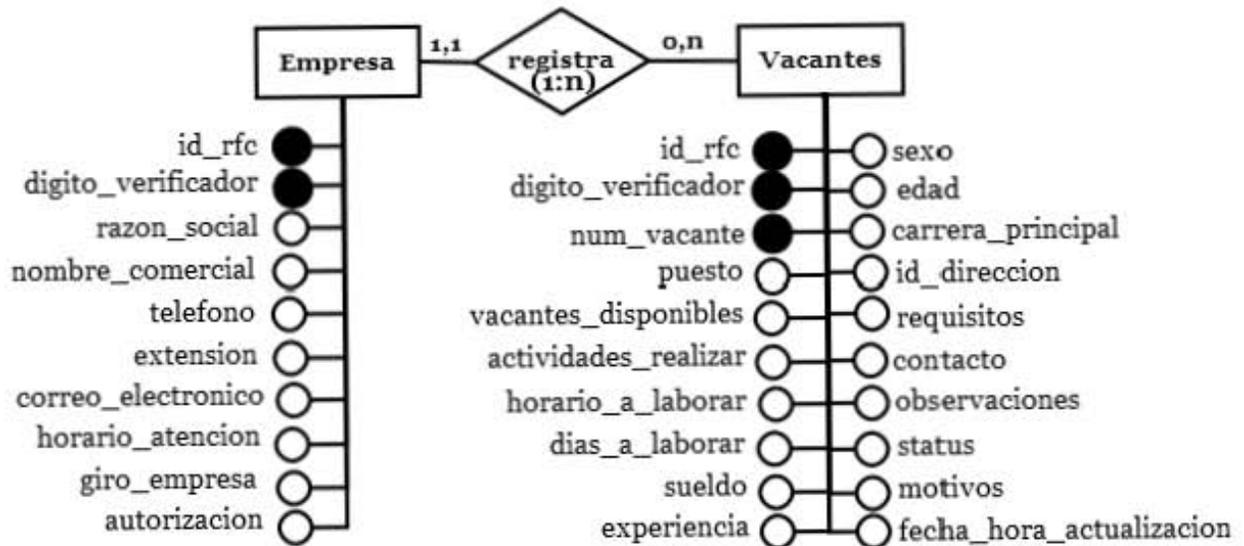
Alumno-Idiomas: La entidad alumno_idiomas se relaciona con las entidades alumno_datos_personales y catalogo_idiomas, porque un alumno puede registrar más de un idioma que se encuentre registrado en el catálogo idiomas.



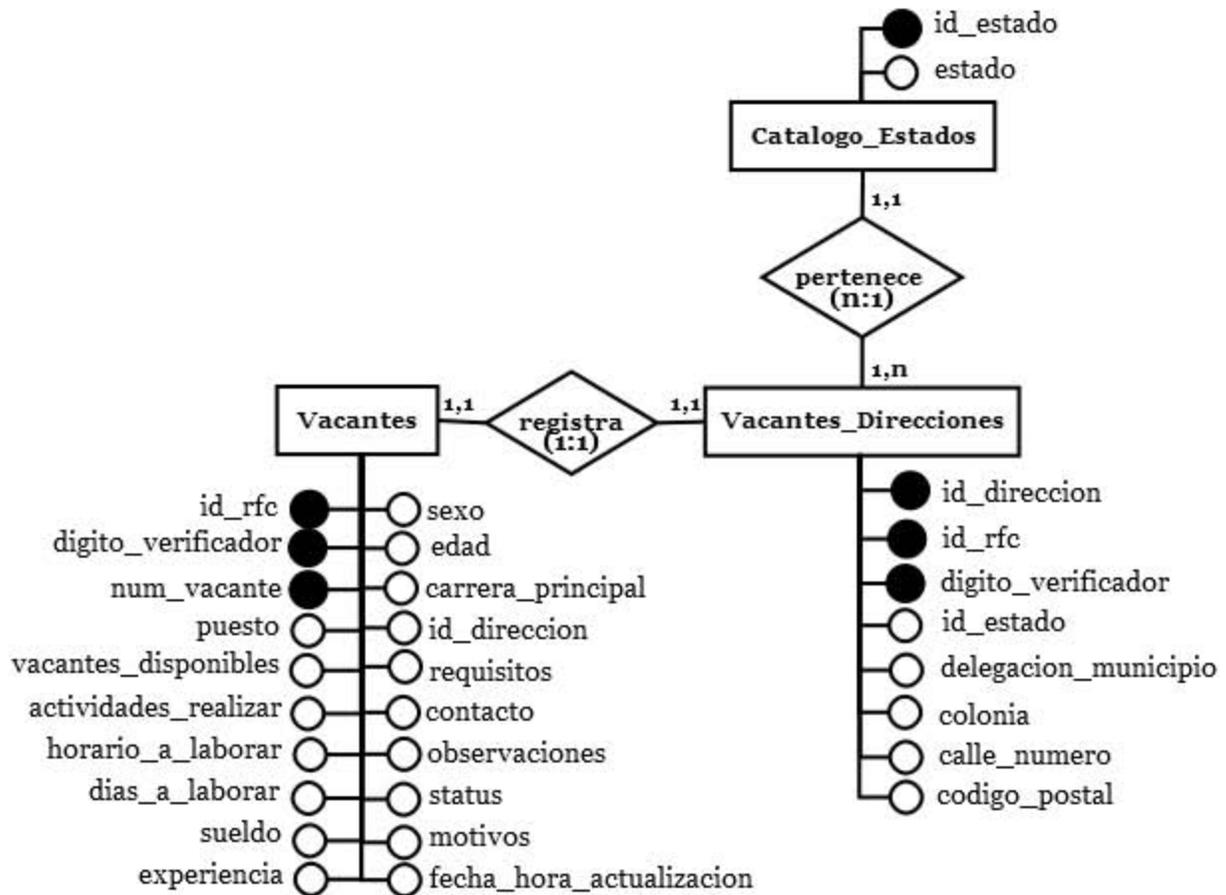
Alumno-Referencias_Laborales y Alumno-Preferencias_Laborales: Estas entidades se relacionan con la entidad alumno_datos_personales, porque un alumno puede registrar su experiencia laboral a través de un registro de referencias laborales sí como sus especificar sus preferencias laborales.



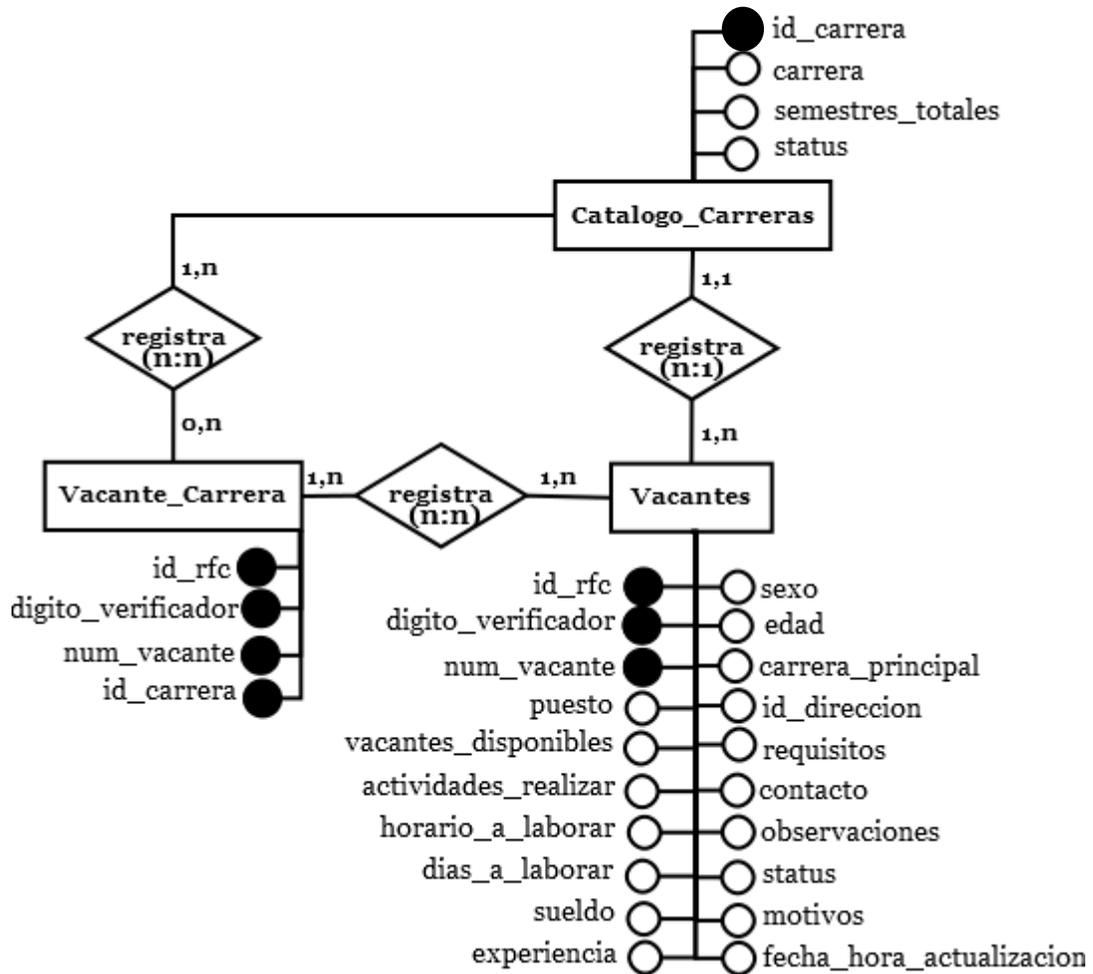
Empresa-Vacantes: La entidad empresa se relaciona con la entidad vacante ya que una vacante solo puede ser realizada por una empresa.



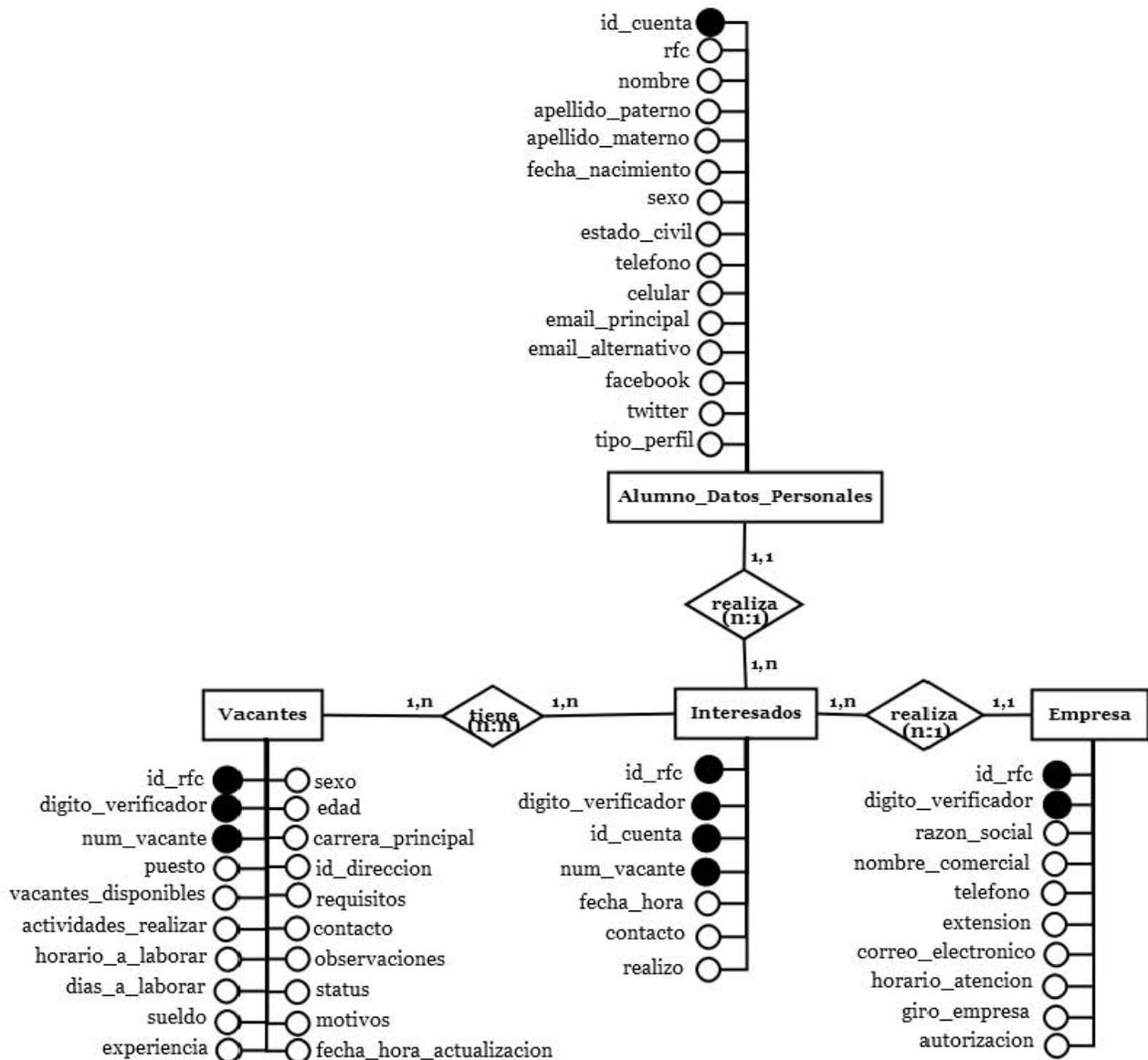
Vacantes-Direcciones: La entidad vacantes_direcciones se encuentra relacionada con las entidades vacantes y catalogo_estados ya que cada vacante debe registrar una dirección, teniendo opción de elegir de un catálogo de estados.



Vacantes-Carreras: La entidad vacantes_carreras se relaciona con las entidades vacantes y catalogo_carreras porque una vacante puede contener más de una carrera que se encuentre en el catálogo de carreras.



Interesados: Esta entidad se encuentra relacionada con las entidades alumno_datos_personales, empresa y vacantes porque un alumno puede mostrarse interesado por una vacante realizada por una empresa o una empresa mostrarse interesada por un alumno a través de una vacante publicada.



3.4.2- Diccionario de datos.

El diccionario de datos es parte elemental para la creación de la base de datos ya que en él se hace mención de las características de las entidades de cada tabla los cuales son de gran apoyo en la etapa de desarrollo del software. En los siguientes apartados describimos el diccionario de datos elaborado para el software de la bolsa de trabajo, así como el diseño lógico de cada tabla.

Tabla 3.1–Administradores.

Tabla: Administradores					No. De campos: 10
Descripción: Contiene datos relevantes sobre los responsables que pueden administrar el sistema.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente del personal responsable.	Valor alfanumérico obligatorio.
nombre	varchar	30		Nombre(s) del encargado.	Cadena de caracteres de valor obligatorio.
apellido_paterno	varchar	30		Apellido paterno del encargado.	Cadena de caracteres de valor obligatorio.
apellido_materno	varchar	30		Apellido materno del encargado.	Cadena de caracteres de valor obligatorio.
puesto	varchar	40		Cargo que desempeña en la institución.	Cadena de caracteres de valor obligatorio.
permisos	tinyint	1		Grado de responsabilidad del administrador.	Valor numérico obligatorio. 0: Permiso restringidos. 1: Todos los permisos.
telefono	varchar	12		Número telefónico de la oficina donde se localiza.	Valor numérico obligatorio con formato (5555555).
extension	varchar	3		Número de extensión existente.	Valor numérico NO obligatorio.
campo	int	1		Número del campus de la institución en donde se encuentra laborando.	Valor numérico obligatorio. 1: Campo Uno. 4: Campo Cuatro.
direccion_campo	varchar	120		Dirección del campus donde se encuentra laborando.	Valor alfanumérico obligatorio.
Relaciones:					
<pre> erDiagram Administradores --} Log_In : tiene (n:1) Administradores --} Ingresos : tiene (n:n) </pre>					

Tabla 3.2- Alumno_Datos_Personales.

Tabla: Alumno_Datos_Personales					No. De campos: 15
Descripción: Contiene los datos principales de los alumnos interesados de la FES Cuautitlán.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
rfc	varchar	13		Código de Registro Federal de Contribuyente del alumno.	Valor alfanumérico obligatorio.
nombre	varchar	30		Nombre(s) del alumno.	Cadena de caracteres de valor obligatorio.
apellido_paterno	varchar	30		Apellido paterno del alumno.	Cadena de caracteres de valor obligatorio.
apellido_materno	varchar	30		Apellido materno del alumno.	Cadena de caracteres de valor obligatorio.
fecha_nacimiento	date			Fecha de nacimiento del alumno.	Fecha de valor obligatorio con formato (aaaa-mm-dd).
sexo	varchar	1		Género del alumno.	Carácter de valor NO obligatorio. F: Femenino. M: Masculino.
estado_civil	varchar	7		Situación del alumno según el registro civil en función de si tiene o no pareja.	Cadena de caracteres de valor NO obligatorio (Soltero o Casado).
telefono	varchar	12		Número telefónico local más lada donde se pueda localizar al alumno.	Valor numérico NO obligatorio con formato (5500000000).
celular	varchar	10		Número telefónico de celular donde se pueda localizar al alumno.	Valor numérico NO obligatorio con formato (55XXXXXXXX).
email_principal	varchar	50		Correo electronico valido del alumno.	Valor alfanumérico obligatorio.
email_alternativo	varchar	50		Correo electronico secundario valido del alumno.	Valor alfanumérico NO obligatorio.
facebook	varchar	50		Nombre del registro en facebook.	Cadena de caracteres de valor NO obligatorio con formato (www.facebook.com/XXXXX XXXX).
twitter	varchar	15		Nombre del registro en twitter.	Cadena de caracteres de valor NO obligatorio con formato (@XXXXXX).
tipo_perfil	tinyint	1		Código de privacidad de la información.	Valor numérico obligatorio. 0: Privado. 1: Público.

Relaciones:

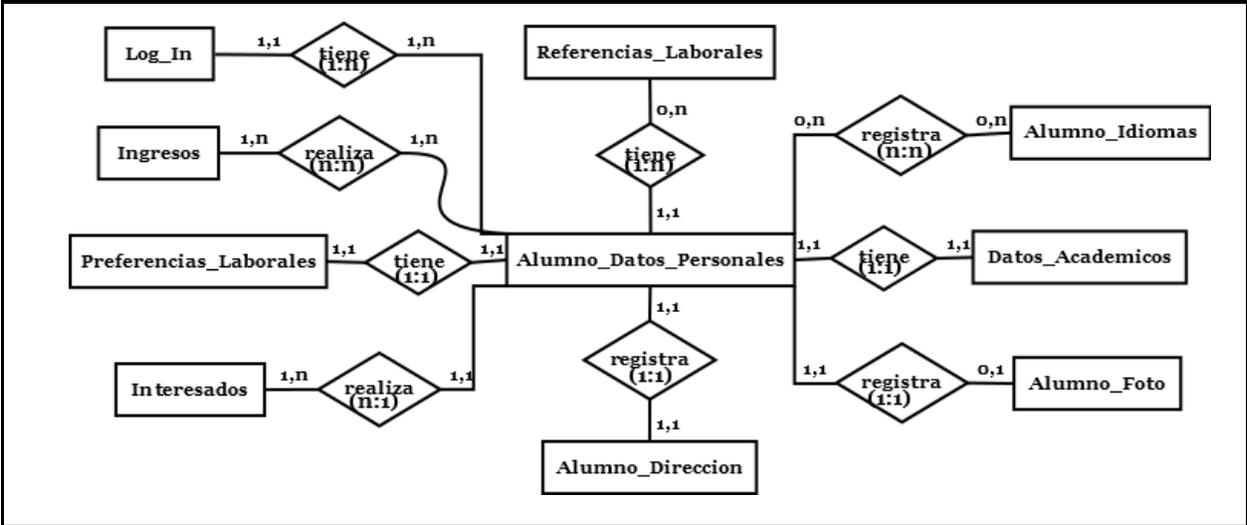


Tabla 3.3–Alumno_Direccion.

Tabla:				Alumno_Direccion	No. De campos: 6
Descripción: Contiene los datos específicos del domicilio del alumno interesado.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
id_estado	varchar	3	foreign key	Código de referencia con la tabla Catalogo_Estados.	Valor alfanumérico obligatorio con formato (Exx).
delegacion_municipipo	varchar	50		Nombre de la delegación o municipio donde radica el alumno.	Cadena de caracteres de valor obligatorio.
colonia	varchar	50		Nombre de la colonia donde radica el alumno.	Cadena de caracteres de valor obligatorio.
calle_numero	varchar	50		Nombre de la calle y número interior y exterior del domicilio del alumno.	Valor alfanumérico obligatorio.
codigo_postal	int	5		Número de identificación de la población donde radica el alumno.	Valor numérico obligatorio.

Relaciones:



Tabla 3.4–Alumno_Foto.

Tabla: Alumno_Foto					No. De campos: 3
Descripción: Contiene las imágenes del perfil de los alumnos en formato binario.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
imagen	mediumblob			Imagen del alumno en formato en binario.	Valor binario NO obligatorio.
tipo_imagen	varchar	30		Tipo de extensión de la imagen.	Cadena de caracteres de valor NO obligatorio.
Relaciones:					
<pre> graph LR ADP[Alumno_Datos_Personales] --- 1,1 registra{registra (1:1)} registra --- 0,1 AF[Alumno_Foto] </pre>					

Tabla 3.5–Alumno_idiomas.

Tabla: Alumno_idiomas					No. De campos: 8
Descripción: Contiene el número de idiomas que maneja el alumno interesado.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
id_idioma	varchar	3	primary key	Código de referencia con la tabla Catalogo_Idiomas.	Valor alfanumérico obligatorio con formato (lxx).
nivel	varchar	10		Nivel general de conocimientos que posee el alumno sobre le idioma.	Cadena de caracteres de valor obligatorio que puede contener (Básico, Intermedio o Avanzado).
porcentaje_hablar	int	3		Porcentaje aproximado del lenguaje hablado que maneja el alumno .	Valor numérico obligatorio de 1 al 100.
porcentaje_leer	int	3		Porcentaje aproximado de la comprensión lectora que maneja el alumno.	Valor numérico obligatorio de 1 al 100.
porcentaje_escuchar	int	3		Porcentaje aproximado de la comprensión oral que maneja el alumno.	Valor numérico obligatorio de 1 al 100.
porcentaje_escribir	int	3		Porcentaje aproximado del lenguaje escrito que maneja el alumno.	Valor numérico obligatorio de 1 al 100.
certificacion	varchar	30		Nombre del documento oficial que avala los conocimientos del alumno.	Valor alfanumérico NO obligatorio.
Relaciones:					

Relaciones:

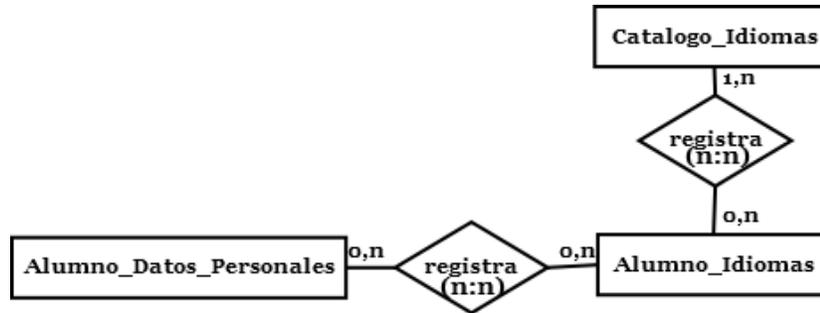


Tabla 3.6–Catalogo_Carreras.

Tabla:				Catalogo_Carreras	No. De campos: 4
Descripción: Contiene las carreras que se imparten en la FES Cuautitlán.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_carrera	int	3	primary key	Clave que establece la UNAM para identificar una carrera.	Valor numérico obligatorio.
carrera	varchar	80		Nombre de la carrera establecida por la UNAM.	Cadena de caracteres de valor obligatorio.
semestre_totales	int	2		Número de semestres totales de duración de la carrera.	Valor numérico obligatorio de 1 -10.
status	tinyint	1		Código de vigencia para mostrar la carrera.	Valor numérico obligatorio. 0: Vigente 1: Revocada

Relaciones:

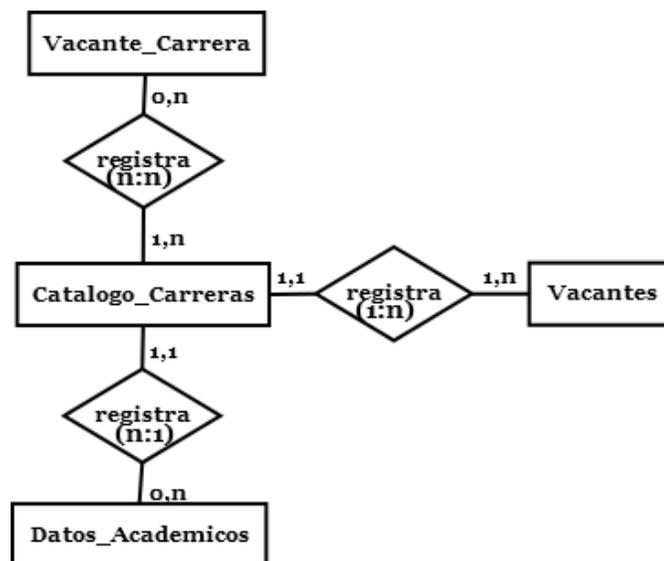


Tabla 3.7–Catalogo_Estados.

Tabla: Catalogo_Estados					No. De campos: 2
Descripción: Contiene los nombres de los estados de México.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_estado	varchar	3	primary key	Código creado para la identificación un estado.	Valor alfanumérico, incremental obligatorio con formato (Exx).
estado	varchar	35		Nombre oficial del estado.	Cadena de caracteres de valor obligatorio.
Relaciones:					
<pre> erDiagram Catalogo_Estados --} Alumno_Direccion : pertenece (1:n) Catalogo_Estados --} Vacantes_Direcciones : pertenece (1:n) </pre>					

Tabla 3.8–Catalogo_Idiomas.

Tabla: Catalogo_Idiomas					No. De campos: 2
Descripción: Contiene los nombres de los idiomas más utilizados.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_idioma	varchar	3	primary key	Código creado para la identificación de un tipo de idioma.	Valor alfanumérico, incremental obligatorio con formato (lxx).
idioma	varchar	15		Nombre del tipo de idioma.	Cadena de caracteres de valor obligatorio.
Relaciones:					
<pre> erDiagram Catalogo_Idiomas --} Alumno_Idiomas : registra (n:n) </pre>					

Tabla 3.9–Catalogo_Situacion_Academica.

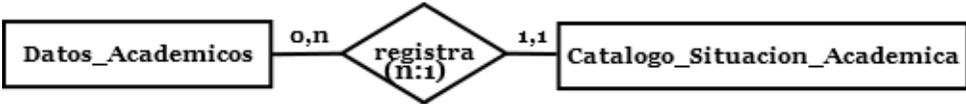
Tabla: Catalogo_Situacion_Academica					No. De campos: 2
Descripción: Contiene el tipo de situación académica al que pertenece un alumno.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_situacion	varchar	4	primary key	Código creado para la identificación de un tipo de situación académica.	Valor alfanumérico, incremental obligatorio con formato (SAxx).
situacion	varchar	15		Nombre del tipo de situación académica.	Cadena de caracteres de valor obligatorio.
Relaciones:					
 <pre> erDiagram Datos_Academicos --o{ Catalogo_Situacion_Academica : registra </pre>					

Tabla 3.10–Datos_Academicos.

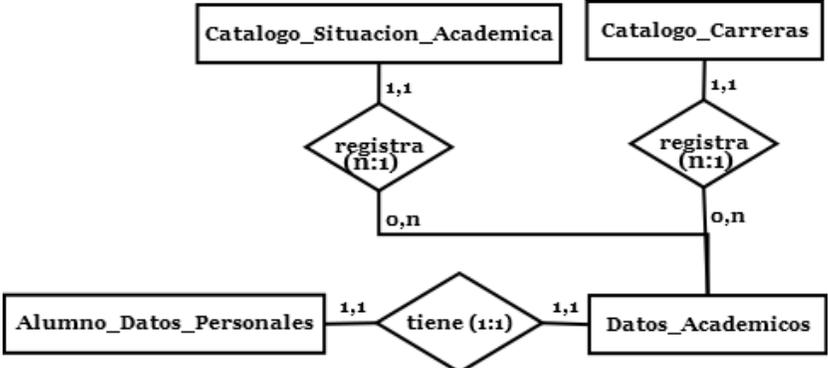
Tabla: Datos_Academicos					No. De campos: 7
Descripción: Contiene los datos referentes a la formación profesional del alumno.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
id_carrera	int	3	foreign key	Código de referencia con la tabla Catalogo_Carreras.	Valor numérico obligatorio.
id_situacion	varchar	5	foreign key	Código de referencia con la tabla Catalogo_Situacion_Academica.	Valor alfanumérico, incremental obligatorio con formato (SAxx).
semestre_actual	int	2		Número de semestres totales cursados.	Valor numérico NO obligatorio de 1 -10.
porcentaje_creditos	decimal	5,2		Porcentaje de creditos actuales del alumno.	Valor decimal obligatorio.
promedio_actual	decimal	4,2		Promedio actual del alumno.	Valor decimal obligatorio.
cursos_adicionales	mediumtext			Nombre y número de cursos adicionales con respecto a su formación profesional .	Cadena de caracteres de valor obligatorio.
Relaciones:					
 <pre> erDiagram Catalogo_Situacion_Academica --o{ Datos_Academicos : registra Catalogo_Carreras --o{ Datos_Academicos : registra Alumno_Datos_Personales -- Datos_Academicos : tiene </pre>					

Tabla 3.11–Catalogo_Horario_Laboral.

Tabla: Catalogo_Horario_Laboral					No. De campos: 2
Descripción: Contiene los tipos de horarios que se manejan en las jornadas laborales.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_horario	varchar	4	primary key	Código creado para la identificación de un tipo de horario.	Valor alfanumérico, incremental obligatorio con formato (HLxx).
horario	varchar	20		Nombre del tipo de horario.	Cadena de caracteres de valor obligatorio.
Relaciones:					
<pre> erDiagram Preferencias_Laborales }o..n "registra" Catalogo_Horario_Laboral : 1..1 </pre>					

Tabla 3.12–Empresas.

Tabla: Empresas					No. De campos: 10
Descripción: Coniene los datos especificos de las empresas interesadas.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente de la empresa.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
razon_social	varchar	70		Nombre oficial y legal de la empresa.	Cadena de caracteres de valor obligatorio.
nombre_comercial	varchar	30		Nombre o denominación que identifica a una empresa.	Cadena de caracteres de valor obligatorio.
telefono	varchar	12		Número telefónico de la oficina donde se podrá proporcionar información.	Valor numérico obligatorio que debe contener lada (5555555555).
extension	varchar	3		Número de extensión existente.	Valor numérico NO obligatorio.
correo_electronico	varchar	50		Correo valido institucional.	Valor alfanumerico obligatorio.
horario_atencion	varchar	50		Especificación de los hoarios disponibles.	Cadena de caracteres de valor obligatorio.
giro_empresa	varchar	30		Nombre del tipo de actividad a la que se dedica la empres.	Cadena de caracteres de valor obligatorio.
autorizacion	tinyint	1		Código de acceso al sistema.	Valor numérico obligatorio. 0: Denegado. 1: Autorizado.
Relaciones:					
<pre> erDiagram Empresa "1..1" "registra" Vacantes : "0..n" </pre>					

Tabla 3.13–Imagen_Banner.

Tabla: Imágenes_Banner					No. De campos: 3
Descripción: Contiene las imágenes que se mostraran en la página principal en formato binario.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_imagen	int	1	primary key	Código creado para la identificación de un tipo de imagen.	Valor numérico, incremental obligatorio.
imagen	mediumblob			Imagen del banner en formato en binario.	Valor binario NO obligatorio.
tipo_imagen	varchar	30		Tipo de extensión de la imagen.	Cadena de caracteres de valor NO obligatorio.

Tabla 3.14–Ingresos.

Tabla: Ingresos					No. De campos: 3
Descripción: Contiene el historial de los usuarios que tienen acceso al sistema.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente del usuario.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
fecha_hora_ingreso	datetime		primary key	Fecha y hora de ingreso al sistema.	Fecha de valor obligatorio con formato (aaaa-mm-dd hh:mm:ss).
Relaciones:					
<pre> erDiagram Alumno_Datos_Personales --} Ingresos : realiza Empresa --} Ingresos : realiza Administradores --} Ingresos : realiza </pre>					

Tabla 3.15–Interesados.

Tabla: Interesados					No. De campos: 7
Descripción: Contiene datos generales del alumno y la empresa, que realizaron una postulación a una vacante ya sea por parte del alumno o la empresa.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rcf	varchar	13	primary key	Código de Registro Federal de Contribuyente de la empresa.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
num_vacante	varchar	9	primary key	Código creado para la identificación de un tipo de vacante.	Valor numérico, incremental obligatorio.
fecha_hora	datetime			Fecha en la que los interesados registran una postulación	Fecha de valor obligatorio con formato (aaaa-mm-dd hh:mm:ss).
contratado	tinyint	1		Código que identifica si se contrato el alumno.	Valor numérico obligatorio. 0: No contratado. 1: Contratado.
realizo	int	1		Código que identifica quien realiza una postulación.	Valor numérico obligatorio. 0: Alumno. 1: Empresa.
Relaciones:					
<pre> erDiagram Alumno_Datos_Personales --o{ Interesados : realiza Vacantes --o{ Interesados : tiene Interesados --o{ Empresa : realiza </pre>					

Tabla 3.16–Log_In.

Tabla:		Log_In			No. De campos: 7	
Descripción: Contiene los datos específicos de los usuarios para ingresar al sistema así como la última fecha de actualización de su información.						
Campo	Tipo	Long.	Llave	Descripción	Validación	
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente del usuario.	Valor alfanumérico obligatorio.	
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.	
password	varchar	10		Contraseña creada por el usuario.	Valor alfanumérico obligatorio.	
fecha_hora_registro	datetime			Fecha de registro en el sistema.	Fecha de valor obligatorio con formato (aaaa-mm-dd hh:mm:ss).	
status	tinyint	1		Código de acceso al sistema.	Valor numérico obligatorio. 0: Denegado. 1: Autorizado.	
tipo_usuario	int	1		Código que identifica al tipo de usuario.	Valor numérico obligatorio. 1: Administrador 2: Alumno. 3: Empresa.	
fecha_hora_actualizacion	datetime			Fecha de actualización de la información en el sistema (esta tiene como vigencia 3 meses).	Fecha de valor NO obligatorio con formato (aaaa-mm-dd hh:mm:ss).	

Relaciones:

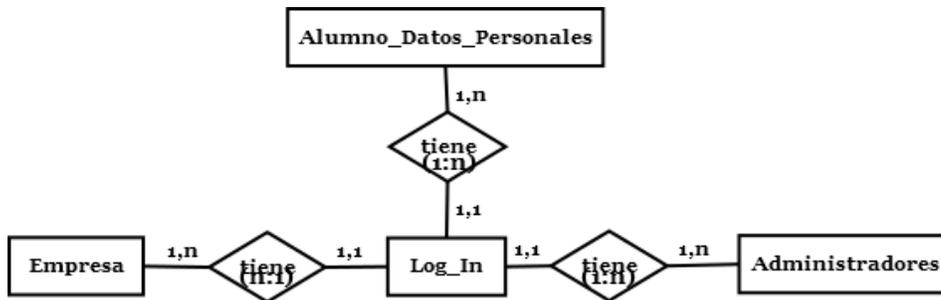


Tabla 3.17–Preferencias_Laboralles.

Tabla: Preferencias_Laborales					No. De campos: 5
Descripción: Contiene los requisitos básicos que el alumno desea sobre un puesto de trabajo.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
id_horario	varchar	5	foreign key	Código de referencia con la tabla Catalogo_Horario_Laboral	Valor alfanumérico, incremental obligatorio con formato (HLxx).
disponibilidad_viajar	tinyint	1		Código que identifica si el alumno desea viajar o no.	Valor numérico obligatorio. 0: No. 1: Sí.
fecha_para_laborar	date			Fecha en la que el alumno desea iniciar a trabajar a partir del día de su registro.	Fecha de valor obligatorio con formato (aaaa-mm-dd).
sueldo_deseado	int	6		Cantida exacta del sueldo deseado.	Valor numérico positivo obligatorio.
Relaciones:					
<pre> erDiagram Catalogo_Horario_Laboral --o{ Preferencias_Laborales : registra Preferencias_Laborales --o{ Alumno_Datos_Personales : tiene </pre>					

Tabla 3.18–Referencias_Laborales.

Tabla: Referencias_Laborales					No. De campos: 8
Descripción: Contiene el historia laboral del alumno.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_cuenta	varchar	9	primary key	Número identificador asignado por la UNAM.	Valor numérico obligatorio.
nombre_empresa	varchar	100	primary key	Nombre comercial de la empresa.	Cadena de caracteres de valor obligatorio.
fecha_inicio	date			Fecha de ingreso.	Fecha de valor obligatorio con formato (aaaa-mm-dd).
fecha_fin	date			Fecha de egredo.	Fecha de valor obligatorio con formato (aaaa-mm-dd).
puesto	varchar	50		Nombre del puesto ocupado.	Cadena de caracteres de valor obligatorio.
sueldo	int	6		Cantida exacta del sueldo cotizado.	Valor numérico positivo obligatorio.
jefe_directo	varchar	150		Nombre completo del jefe directo.	Cadena de caracteres de valor obligatorio.
tel_jefe_directo	varchar	12		Número telefónico de la oficina donde se podrá localizar al jefe directo o pedir información acerca del usuario.	Valor numérico obligatorio que debe contener lada (5555555555).
Relaciones:					
<pre> erDiagram Referencias_Laborales -- Alumno_Datos_Personales : tiene </pre>					

Tabla 3.19–Vacantes_Carreras.

Tabla: Vacante_Carrera					No. De campos: 4
Descripción: Contiene las carreras relacionadas a una vacante.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente de la empresa.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
num_vacante	varchar	9	primary key	Código creado para la identificación de un tipo de vacante.	Valor numérico, incremental obligatorio.
id_carrera	int	3	primary key	Código de referencia con la tabla Catalogo Carreras.	Valor numérico obligatorio.
Relaciones:					
<pre> erDiagram Catalogo_Carreras --} Vacante_Carrera : "registra (n:n)" Catalogo_Carreras --} Vacante_Carrera : "1,n" Vacante_Carrera --} Vacantes : "registra (n:n)" Vacante_Carrera --} Vacantes : "1,n" </pre>					

Tabla 3.20–Vacantes.

Tabla: Vacantes					No. De campos: 20
Descripción: Contiene los requisitos necesarios para publicar una vacante por parte de la empresas.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente de la empresa.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
num_vacante	int	11	primary key	Código creado para la identificación de un tipo de vacante.	Valor numérico, incremental obligatorio.
puesto	varchar	50		Nombre del puesto solicitado.	Cadena de caracteres de valor obligatorio.
vacantes_disponibles	varchar	2		Número de vacantes disponibles.	Valor numérico entero obligatorio.
actividades_realizar	mediumtext			Descripción general de las actividades a desempeñar.	Cadena de caracteres de valor obligatorio.
horario_a_laborar	varchar	30		Especificar el número de horas y el horario a cubrir.	Cadena de caracteres de valor obligatorio.
dias_a_laborar	varchar	30		Descripción de los días a laborar.	Cadena de caracteres de valor obligatorio.
sueldo	int	6		Cantida exacta del sueldo propuesto.	Valor numérico positivo obligatorio.
experiencia	varchar	30		Descripción exacta de la experiencia necesaria.	Cadena de caracteres de valor obligatorio.
sexo	varchar	10		Nombre del genero preferente.	Cadena de caracteres de valor obligatorio.
edad	varchar	15		Rango de edad preferente.	Cadena de caracteres de valor NO obligatorio.
carrera_principal	int	3	foreign key	Código de referencia con la tabla Catalogo Carreras.	Valor numérico obligatorio.
id_direccion	varchar	4	foreign key	Código de referencia con la tabla Vacantes Direcciones.	Valor numérico, incremental obligatorio.
requisitos	mediumtext			Descripción general de los requisitos necesarios para cubrir el puesto.	Cadena de caracteres de valor obligatorio.
contacto	varchar	50		Nombre del responsable o entrevistador.	Cadena de caracteres de valor obligatorio.
observaciones	mediumtext			Aspectos específicos que no han sido mencionados.	Cadena de caracteres de valor NO obligatorio.
status	int	1		Código de acceso para mostrar la carrera.	Valor numérico obligatorio. 0: Revocada. 1: Vigente.
motivos	text			Descripción de los motivos por la cual no se acepta la vacante.	Cadena de caracteres de valor NO obligatorio.
fecha_hora_actualizacion	datetime			Fecha de actualización de la vacante en caso de ser rechazada.	Fecha de valor obligatorio con formato (aaaa-mm-dd hh:mm:ss).

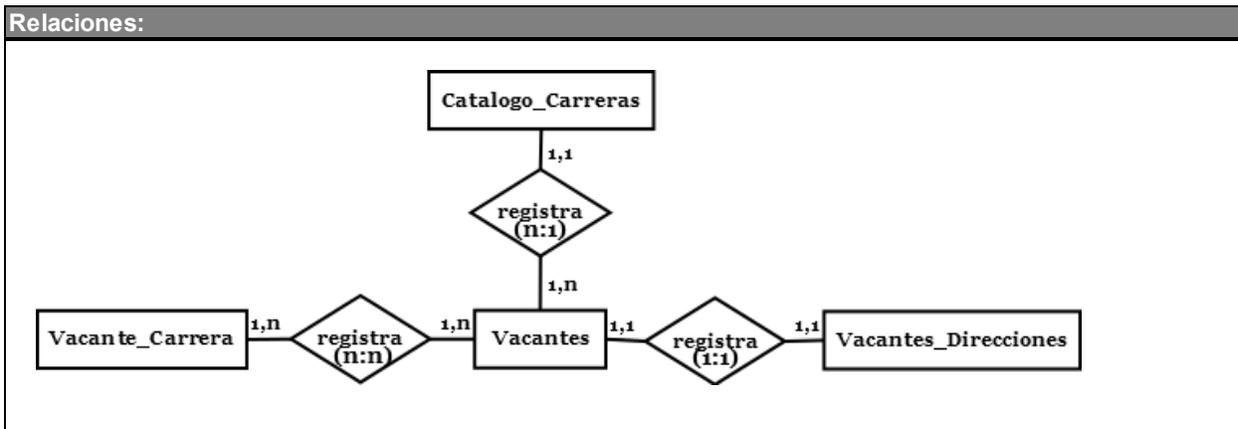
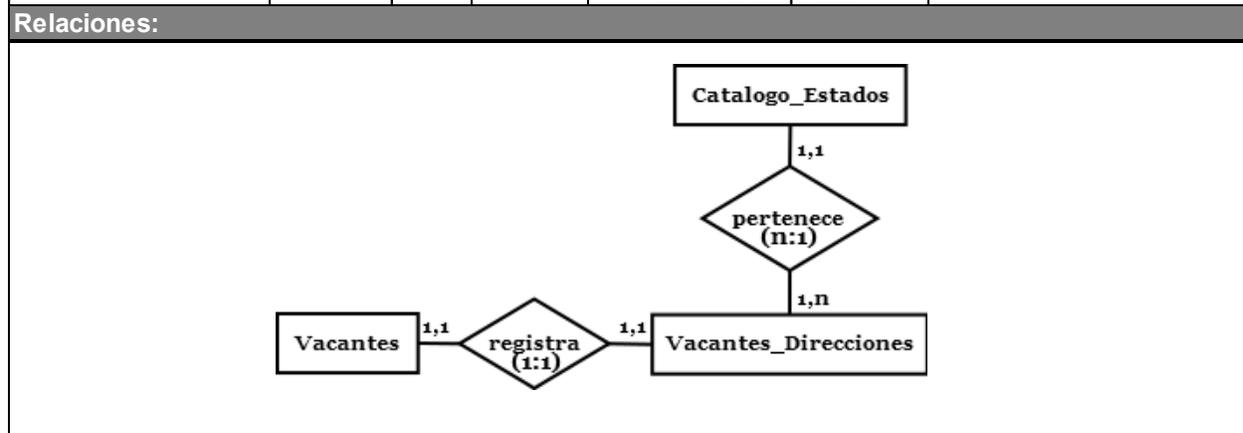


Tabla 3.21–Vacantes_Direcciones.

Tabla: Vacantes_Direcciones					No. De campos: 8
Descripción: Contiene las diferentes direcciones con las que cuenta un vacante.					
Campo	Tipo	Long.	Llave	Descripción	Validación
id_direccion	varchar	4	primary key	Código creado para la identificación de un tipo de dirección de la empresa.	Valor numérico, incremental obligatorio.
id_rfc	varchar	13	primary key	Código de Registro Federal de Contribuyente de la empresa.	Valor alfanumérico obligatorio.
digito_verificador	varchar	4	primary key	Código creado para la identificación específica de una empresa.	Valor numérico, aleatorio obligatorio.
id_estado	varchar	3	foreign key	Código de referencia con la tabla Catalogo_Estados.	Valor alfanumérico, incremental obligatorio con formato (Exx).
delegacion_municipio	varchar	50		Nombre de la delegación o municipio donde se localiza la empresa.	Cadena de caracteres de valor obligatorio.
colonia	varchar	50		Nombre de la colonia donde se localiza la empresa.	Cadena de caracteres de valor obligatorio.
calle_numero	varchar	50		Nombre de la calle y número interior y exterior del domicilio de la empresa.	Cadena de caracteres de valor obligatorio.
codigo_postal	int	5		Número de identificación de la población donde se ubica la empresa.	Cadena de caracteres de valor obligatorio.



3.5- Modelo Prototipado

El diseño rápido del modelo de prototipado permite presentar aquellos aspectos que serán visibles en la capa de presentación o interfaz del sistema. Para el diseño del sistema de bolsa de trabajo se plantea el siguiente modelo de interfaz el cual se divide en cuatro módulos (inicio, empresa, alumno y admo.) principales que se encuentran integrados por un conjunto de vistas que se relacionan con cada funcionalidad del sistema. En las siguientes figuras se muestra el diseño creado para cada módulo.

Módulo Inicio

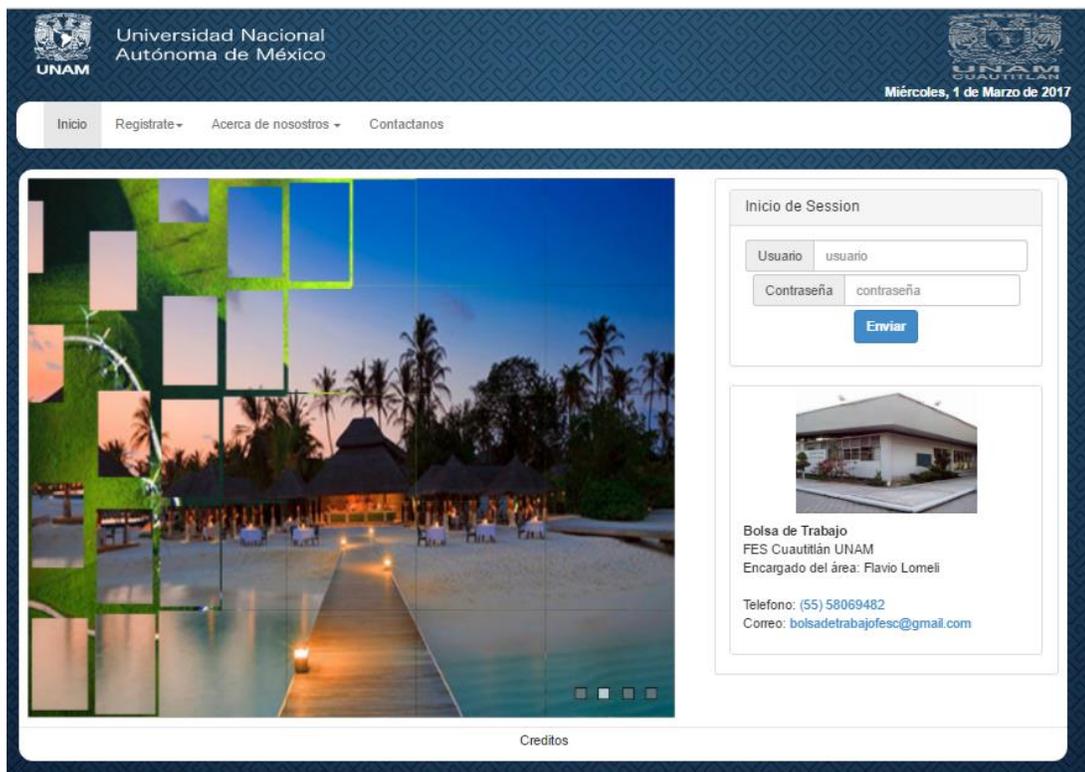


Figura 3.9– Prototipo del módulo inicio.

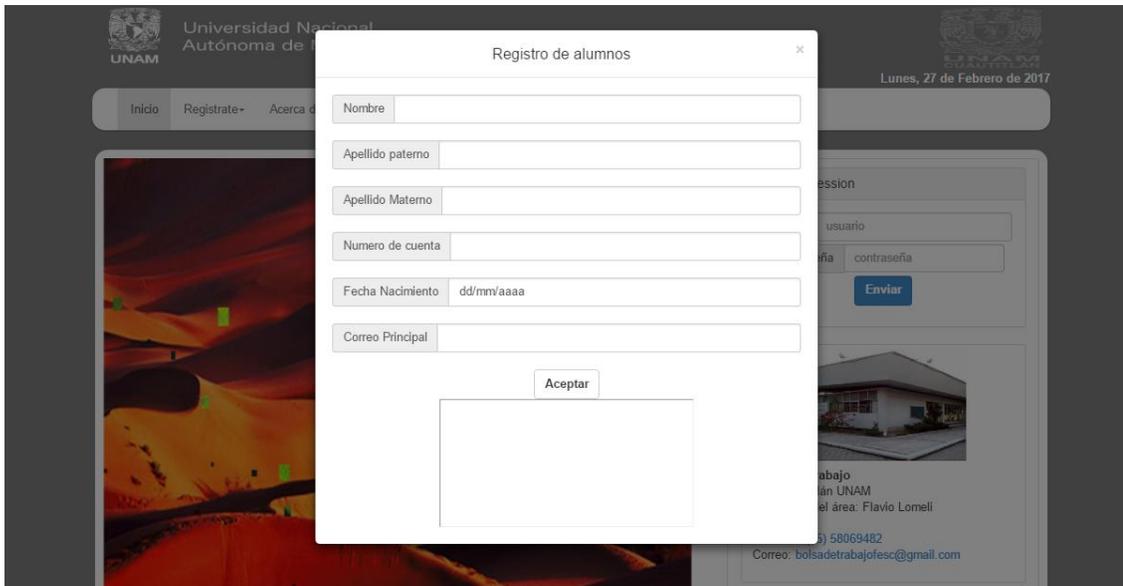


Figura 3.9.1– Prototipo del módulo inicio, registro de alumnos.

Login



Figura 3.10– Prototipo del módulo inicio, Login.

Módulo Alumno.

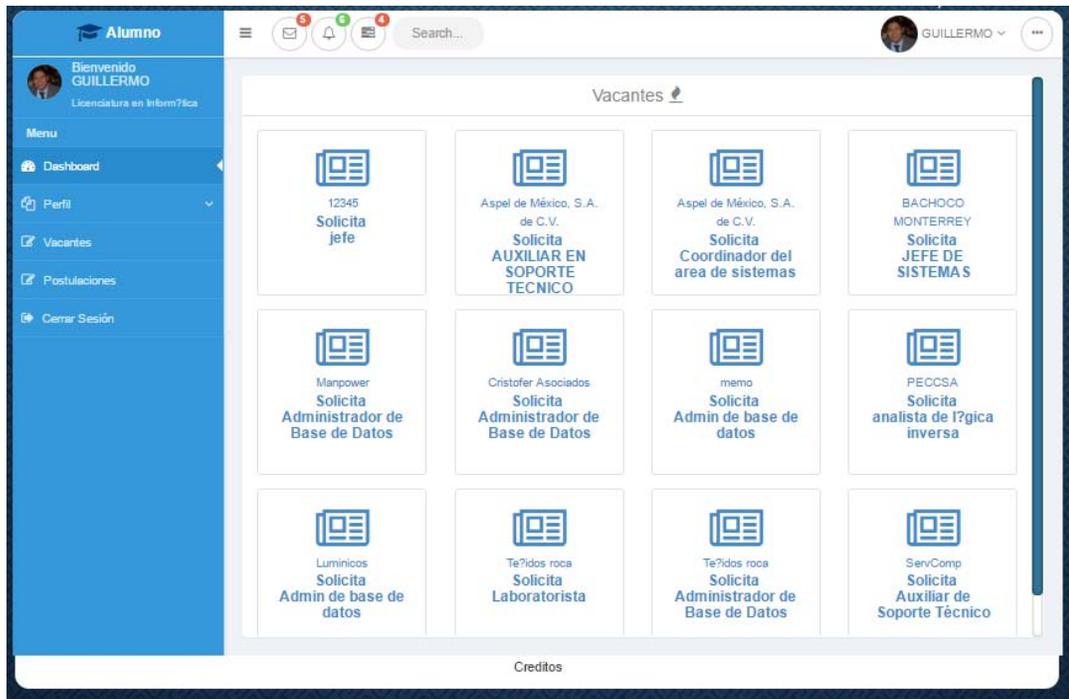


Figura 3.11– Prototipo del módulo alumno.

Módulo Empresa.

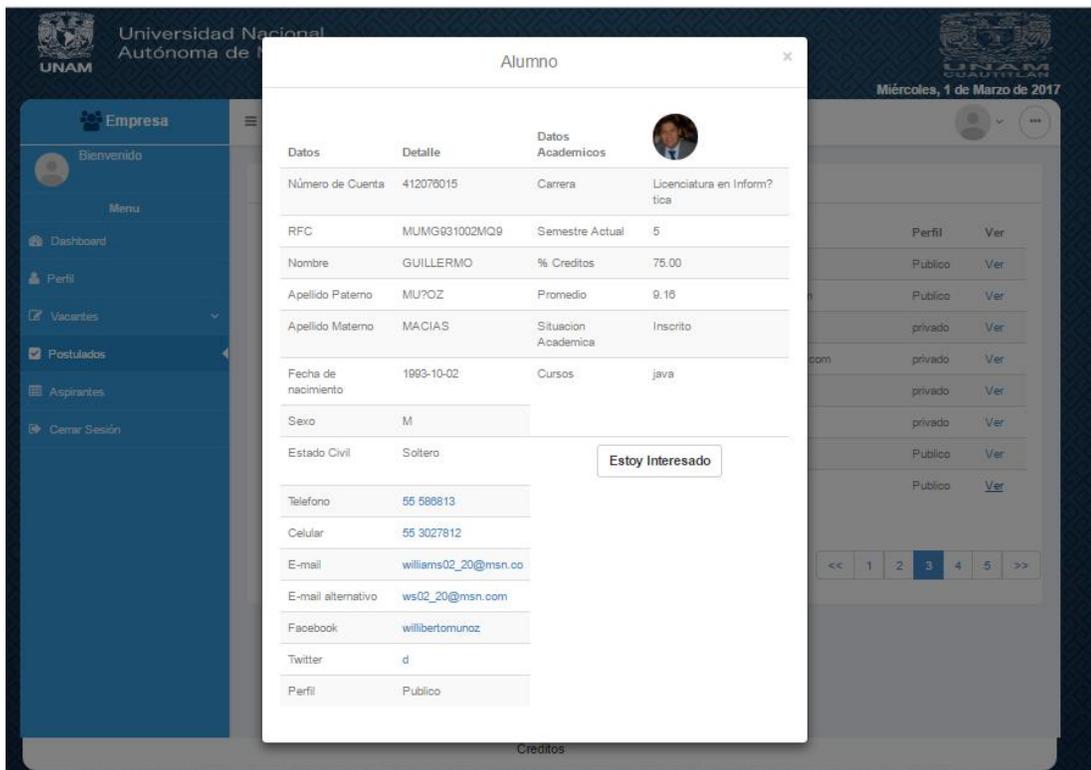


Figura 3.12– Prototipo del módulo Empresa.

Módulo Administrador.

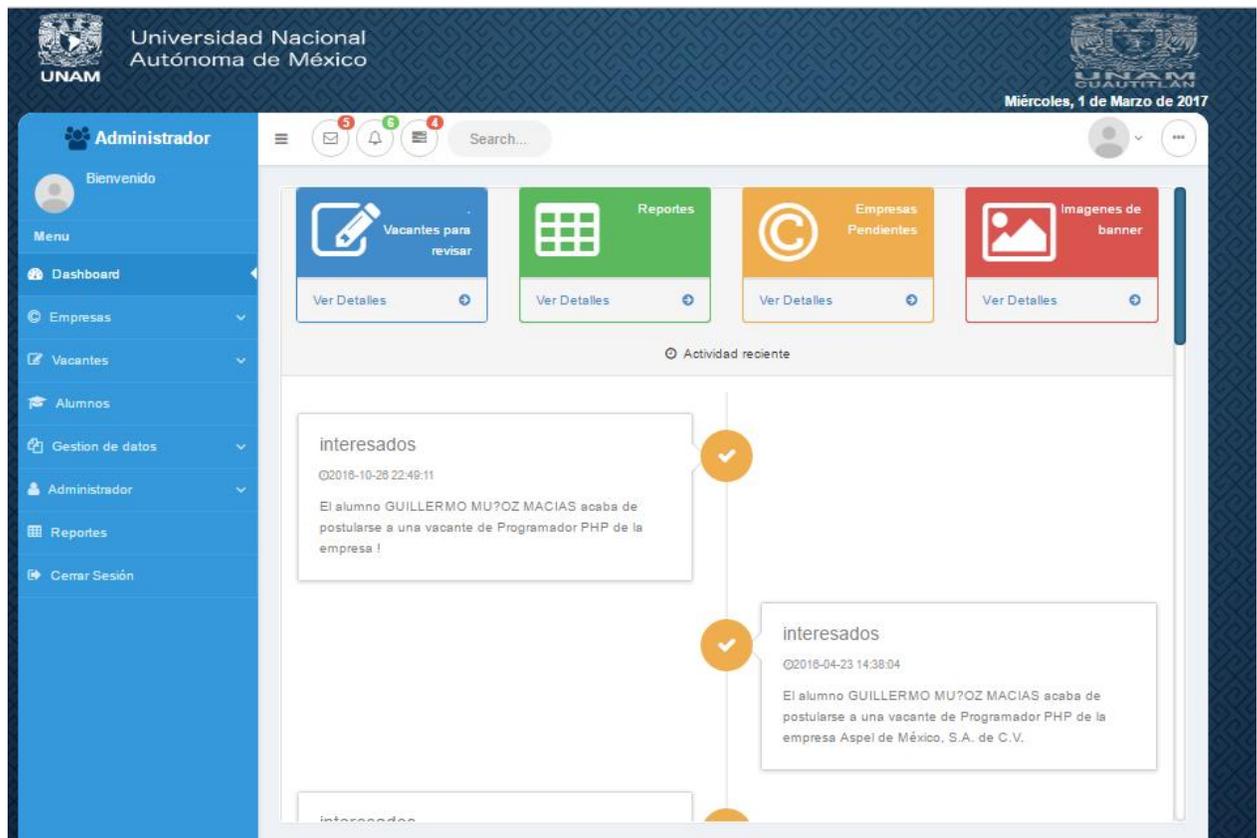


Figura 3.13– Prototipo del módulo Administrador.

3.6- Programación y depuración.

El sistema de bolsa de trabajo se encuentra estructurado por módulos los cuales son: Inicio, Alumno, Administrador y Empresa.

3.6.1- Módulo Inicio

Módulo principal el cual funciona como un controlador, este contiene la página principal que se les muestra a los usuarios que ingresen. Este controlador llamado IndexController consta de 13 funciones, estas funciones controlan las vistas que se visualizan en el lado del cliente. Las funciones tienen la siguiente nomenclatura <Public> <function> < nombre de función>Action para ellas que controlan los encabezados URL de la página; todas las funciones que tengan una vista de phtml deben de ser funciones públicas para que puedan ser vistas por cualquier módulo.

Las funciones que no cuentan con esta nomenclatura deberán de ser privadas (private) y utilizar camelCase, ya que son funciones que se utilizarán de uso interno y no deberán de ser asequibles por ningún otro módulo. Redirigir es una función

privada que se utilizara en el módulo y que no afectara el manejo público de las funciones Action. Para utilizar estas funciones se debe de crear un objeto de la clase IndexController, el cual podrá llamar a sus funciones públicas para hacer el manejo de la información.

Inicio es un módulo que será el principal y será inicializado por su constructor llamado desde module.config (véase en capítulo 1), para crear el objeto se mandara a llamar a la clase constructora que creará un servicio de autenticación para que pueda ser llamado desde cualquier parte de su funciones o subclases. Algunas de las funciones que utiliza la clase se mostraran a continuación;

```
public function __construct() {  
    //Cargamos el servicio de autenticación en el constructor  
    $this->auth = new AuthenticationService();  
}
```

La función de redirigir se encarga de administrar la información que fue obtenida en la base de datos, con base en esta se podrá redirigir a los usuarios de los diferentes módulos, sin poder acceder a otro módulo sin autorización de la base de datos.

```
namespace Inicio\Controller;  
  
use Zend\Mvc\Controller\AbstractActionController;  
use Zend\View\Model\ViewModel;  
use Inicio\Form\LoginForm;  
//Componentes de autenticación  
use Zend\Authentication\AuthenticationService;  
//use Zend\Authentication\Storage\Session as SessionStorage;  
use Zend\Session\Container;
```

Figura 3.14—Constructor Inicio controller.

```

private function redirigir($usuario, $status) {
    $auth = $this->auth;
    $identi = $auth->getStorage()->read();
    if ($status == 0) {
        $this->flashMessenger()->addMessage("Usuario esta dada de baja, contacte al administrador");
        return $this->redirect()->toUrl($this->getRequest()->getBaseUrl() . '/login');
    } else {
        switch ($usuario) {
            case 1:
                //Nos redirige a una pagina interior
                $this->registrarIngreso($identi->digito_verificador, $identi->id_rfc);
                return $this->redirect()->toUrl($this->getRequest()->getBaseUrl() . '/Alumno/');
                break;
            case 2:
                $this->registrarIngreso($identi->digito_verificador, $identi->id_rfc);
                return $this->redirect()->toUrl($this->getRequest()->getBaseUrl() . '/Administrador/');
                break;
            case 3:
                return true;
                break;
        }
    }
}

```

Figura 3.14.1– Función Redirigir.

Un layout es creado con una extensión .phtml la cual contiene HTML y PHP con el cual podremos referenciar a cosas que fueron declaradas en el controlador y las cuales no serán mostradas de una manera explícita en las vistas, el layout utiliza funciones en las cuales se hará referencia a archivos de estilos y scripts, todos estos archivos deben de ser incluidos obteniendo la base de la URL, ya que Zend maneja un directorio netamente público no es posible anexar estos archivos sin verificar la base URL.

El módulo utiliza un layout que es utilizado por las funciones action y las vistas de estas para mostrar el contenido de la paginas sin necesidad de estar declarando y haciendo varios archivos HTML que conlleven la misma información, además de que si hubiera un cambio masivo se haría en el layout (front end) sin afectar las reglas del negocio que son los controladores y clases (Back end).

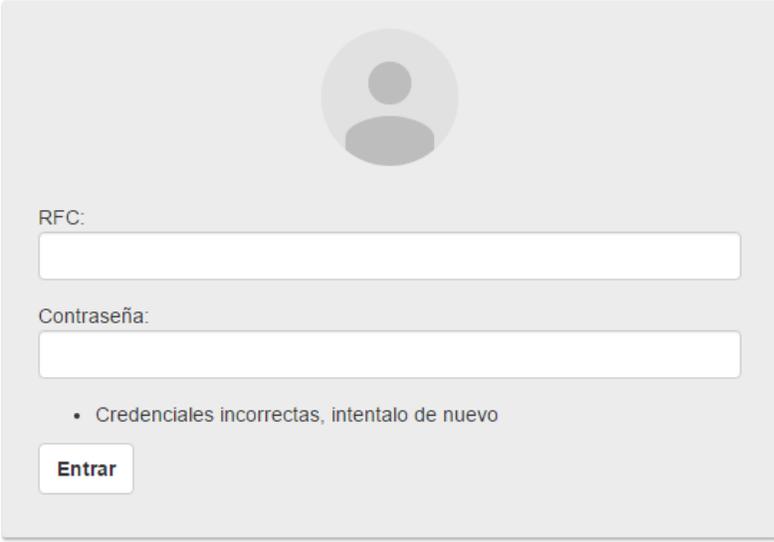
En la página se puede visualizar que se creó un menú horizontal el cual contiene en el parte superior de este como header los logotipos de la FES Cuautitlán y de UNAM, ambos redirigen a sus respectivas paginas oficiales, en el lado derecho se muestra el inicio de sesión (usuario y contraseña) el cual será proporcionado por el sistema y posteriormente podrá ser modificado por el usuario, el inicio de sesión es un formulario hecho en HTML y enlazado a una acción del controlador index del módulo de inicio, esto fue utilizado ya que las acciones que se lleven a cabo por parte

de login se han independientes a la interfaz principal, y de este modo no se tenga manejo de información al iniciar el sistema.

La acción de login controla las entradas de información de este modo la acción login crea un container con el posible usuario a pasar la validación de base de datos, revisa si hay alguna sesión iniciada o creada para no saturar base de datos, de contener alguna sesión verifica a donde tiene que ser dirigido el usuario.

En caso contrario la acción hace una conexión a base de datos creado un objeto authenticate con la identidad del RFC el cual será el usuario de cualquier persona que quiera ingresar al sistema y se encuentre registrado en el sistema, las credenciales será la contraseña que el sistema le género y posteriormente podrá ser modificada. Prosiguiendo con la información si hay una identidad (usuario que exista y cumpla con las características) procederá a la función de redirigir, en caso de ser empresa pedirá el dato de digito verificador.

Bienvenido



A login form with a light gray background. At the top center is a circular placeholder for a user profile picture. Below it are two input fields: the first is labeled 'RFC:' and the second is labeled 'Contraseña:'. Below the password field is a small error message: '• Credenciales incorrectas, intentalo de nuevo'. At the bottom left is a button labeled 'Entrar'.

Figura 3.15- Login.

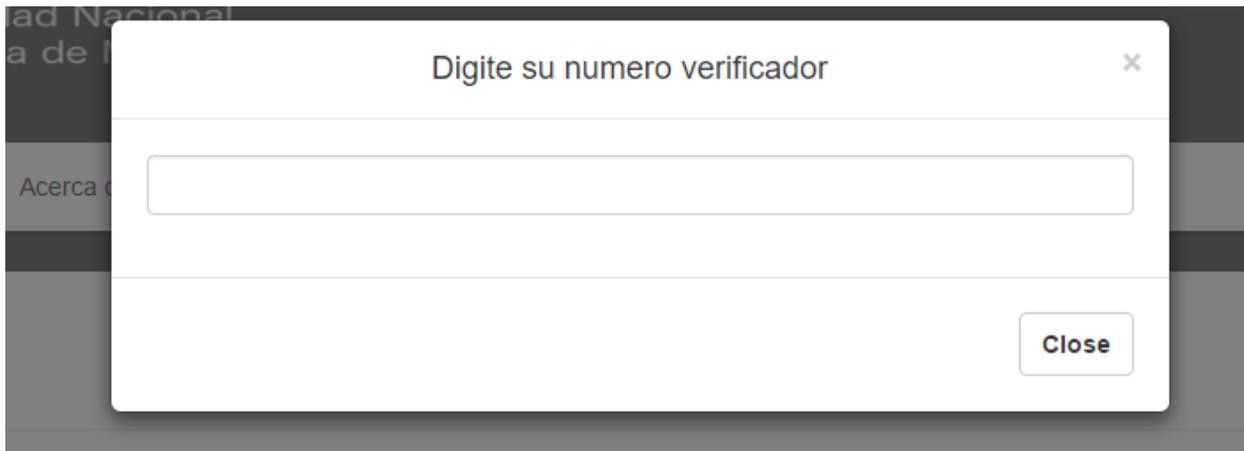


Figura 3.15.1- Dígito verificador

Registrarse como usuario

El sistema está diseñado para poder tener registrado 3 diferentes tipos de usuarios, de los cuales solo dos se podrán auto matricular al sistema, estos son las empresas y los alumnos. A partir de esta premisa se ha creado en el módulo de Inicio en el menú la posibilidad de llenar un registro y generar información con la cual el usuario ingresara. El alumno tendrá que llenar los siguientes datos como: Nombre(s), Apellido Paterno y Materno, número de cuenta, correo electrónico y fecha de nacimiento con esta información el sistema creará un registro para este alumno y generará un PDF el cual será enviado al correo del solicitante para poder ingresar al sistema y completar los registros para poder postularse al sistema y poder visualizar vacantes.

Bienvid@ DAVID

Usuario : **HEPD930730V63**
Contraseña : **1993-07-30**



Objetivo de Bolsa de Trabajo de FESC
Vincular a los estudiantes y egresados de la facultad de Estudios Superiores Cuautitlán con el sector productivo, privado e instituciones gubernamentales, fungiendo como enlace entre los departamentos de recursos humanos y la comunidad universitaria para la obtención de ofertas de empleo, prácticas profesionales y/o becarios.

Visión
Ser un entorno de comunicación funcional y reconocida por su influencia dentro y fuera de la Universidad, un medio efectivo de colaboración entre la FESC y los sectores productivo y gubernamental. Ser la herramienta estratégica que vincule de manera efectiva, amplia y vigorosa a los estudiantes y egresados de la FESC con el campo profesional. Constituir una guía permanente y una fuente autorizada de conocimientos e información vigentes para los buscadores de empleo de la comunidad universitaria en todo lo referente a oportunidades de puestos de trabajo, autopromoción profesional, capacitación y actualización.

Bolsa de Trabajo FES Cuautitlán

Figura 3.16–PDF Bienvenida.

En el caso de la empresa será similar al del alumno a diferencia que la empresa deberá de ingresar el sistema su contraseña y estará a una aprobación del administrador del sistema.

Registro de Empresa ×

RFC

Razón Social

telefono

Extension

Giro Empresa

Correo Principal

Contraseña

Repetir Contraseña

Figura 3.17–Formulario Empresa.

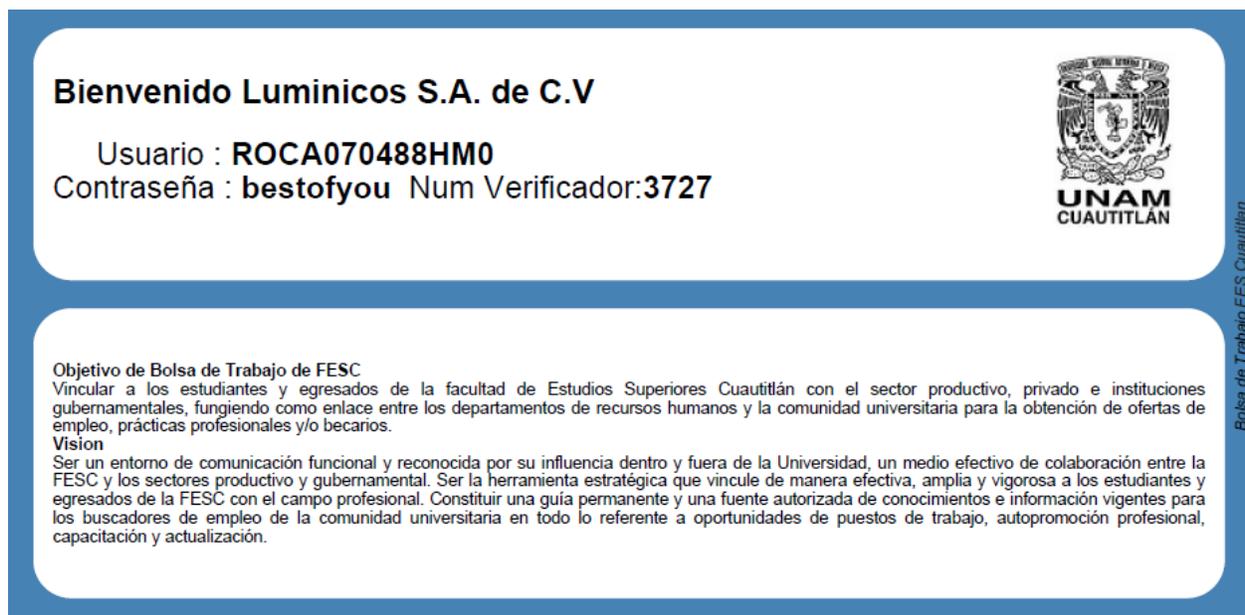


Figura 3.18—PDF Bienvenida alumno.

3.6.2- Módulo Alumno

Este módulo se encargará de procesar la información que el alumno genere cuando se encuentre activo en la página, y se encuentra conformado por un controlador quien administra las acciones que el usuario haga y se comunica con clases, vistas y formularios.

El controlador principal de este módulo tendrá un constructor que revisará al servicio de autenticación y leerá los datos que contiene este servicio, si contiene datos validos los cuales significa que contiene un usuario y si tipo de usuario es 1 (alumno), de contener un usuario, se creará una identidad en Zend, se asignara un layout nuevo, que será el de alumno, y hará que su sesión expire en 800 segundos.

Index action será la acción que por defecto se ejecutará esto en base a la dirección URL ejemplo: `http://< IP del proyecto >/Alumno`, tal y como se muestra en la configuración del módulo en `module.config.php` que en una petición literal por el método `get`, se dirigirá a Inicio controller con la action de `index`. En esta función se hará una comprobación antes de mostrar la vista que si el usuario no es nulo y su tipo de usuario corresponde al número 1 se continuara con la función, en caso de ser negativo se mostrara un flash Messenger en la acción de login del módulo de inicio “Sesión finalizada, inicie de nuevo”.

En el siguiente diagrama se mostrará las funciones que se utilizan en el controller del módulo de alumno, cabe aclarar que el controlador es una clase de PHP que extiende a “AbstractActionController”

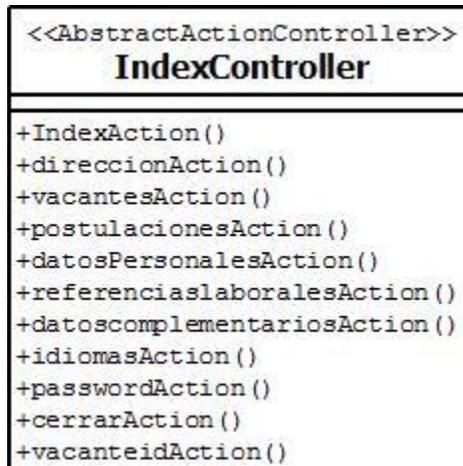


Figura 3.20–Index controller.

Además de esta clase del controlador el módulo de alumno cuenta con otras clases que son fundamentales para su correcto funcionamiento, ya que contienen los modelos que harán relacionar con el controlador y a su vez este con la vista, sin que la vista conozca a los modelos con que se relaciona.

Modelo alumno la cual es una clase que se hizo especialmente para el manejo y uso de los datos del mismo sin afectar ninguna de los modelos que se relacionan con la base de datos. En esta clase se ha implementado el patrón de diseño singleton, véase en el capítulo 1 en patrones de diseño, para que el programa no genere demasiados objetos al momento de hacer las consultas en cada una de las acciones del controlador. Así evitamos que el sistema no tenga demasiada información en memoria RAM y no tener que estar reciclando procesos.

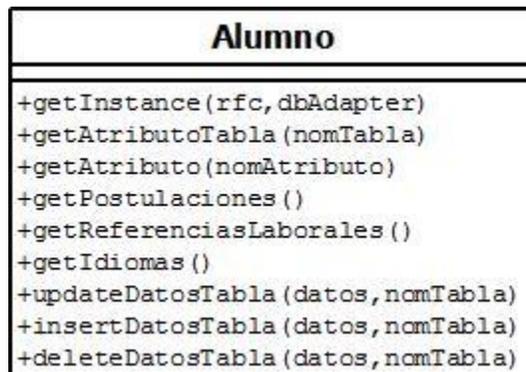


Figura 3.21–Clase alumno.

El modelo de AlumnoDatosAcademicos es una clase que extiende a TableGateway ya que es un modelo de una tabla que se encuentra en la base de datos. Este tipo de clases tendrán un constructor en el cual pidan una conexión a la base de datos que fue configurada en .config y llamada en el controlador (no puede ser declarada en la clase). Contiene 3 funciones elementales para hacer select, update, delete e insert.



Figura 3.22–Clase alumno datos académicos.

AlumnoDatosPersonales es una clase que tiene las definiciones de la tabla. Contiene las funciones esenciales de select, update, delete e insert, además de tener otras como getAlumnoPaginator la que se usa para obtener la consulta de la tabla, pero con un formato de paginator (utilizado en Zend Framework Como paginator). Este formato es usado en la vista para mostrar tablas en HTML sin tener que configurar un paginator especial por el número de filas que genere la consulta hecha.



Figura 3.23—Clase alumno datos personales.

AlumnoDireccion es una clase para obtener los datos de la dirección del alumno contiene las funciones esenciales de select, update, delete y insert.



Figura 3.24—Clase alumno dirección.

Alumnoldioma, PreferenciasLaborales y ReferenciasLaborales son clases para obtener los datos del alumno y contiene las funciones esenciales de select, update, delete e insert y son clases genéricas.

El módulo alumno tiene formularios que deben de ser clases que extienden a Zendform, algunos de los formularios que se han creado en Zend, pero algunos otros se han creado como clases normales que retornan una cadena y tienen una función para generar consultas a la base de datos mediante modelos que extiendan a TableGateway y los select (creados en HTML) sean dinámicos con la base de datos sin modificar programación.

3.6.3- Módulo Empresa

El módulo de empresa es el que procesara la información que genera el usuario con tipo de perfil empresa, y se encuentra conformado por un controlador quien administra las acciones que el usuario haga y se comunica con clases, vistas y formularios.

El controlador de la empresa se encuentra constituida por las acciones que obtiene por el método GET URL. Este controla las vistas que se muestran en el módulo, este igual que todos antes te enviar la vista al cliente crear una vista sin layout (setTerminal (true)) esto es para que la pagina no se esté refrescando con todo el contenido de la página y solo se genere en un div.

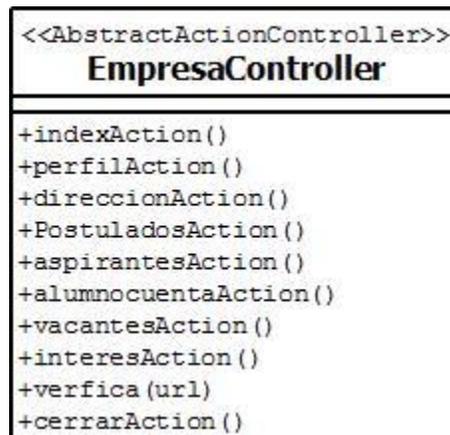


Figura 3.25–Empresa controller.

El módulo empresa es dependiente de los demás módulos, esto se hizo con la finalidad de que si hay un cambio en base de datos o en las reglas de negocio (que deben de estar en la base de datos para evitar tanta programación) solo se deba modificar un solo archivo, aunque la programación por módulos nos recomienda que cada uno sea independiente para que su mejor implementación e implantación de sistema. El módulo contiene 2 modelos, uno es de base de datos y el otro es una implementación propia para generar select de HTML dinámicos con base de datos.

Combo es el modelo creado en la cual al momento de construir el objeto se necesita una conexión a base de datos mediante un adapter. Al momento de utilizar el método llamado “getComboTable” la columna que quiere que se visualice en el combo con su clave que será necesaria como identificador único (PK), la tabla de la que se traerá la información y la condición de la información, con sintaxis de MYSQL. Ejemplo

```
$combo = new Combo($adapter);
$combo->getComboTable("nombre_alumno as nombre,id_cuenta as valor",
    "alumno","1=1");
```

Figura 3.26–Representación de llamada de objeto.

En este ejemplo creamos un objeto de clase combo, con el objeto inicializado con el adaptador de la base los datos mandamos a llamar el método public de la clase llamado

“getComboTable” ingresamos los parámetros en el cual es con sintaxis MYSQL en el primero damos las columnas de la base. Pero adoptamos un alias para que la función genere una consulta con alias, en el segundo parámetro mandamos la tabla en la cual se generará la consulta en caso de querer que se muestren varios datos de varias tablas ahí podemos generar los INNER, LEFT, RIGH, ACROSS y FULL JOIN para que se pueda mostrar en el select.

La clase de vacantesDireccion es una clase y modelo de una tabla de MYSQL tiene los métodos genéricos de cualquier tabla, en su constructor solicita un adaptador de la base de datos, para poder hacer consultas, inserciones , actualizaciones y borrados de la tabla que se declara, también genera una consulta que se convierte en paginator Zend para poder visualizar las vacantes , el método tiene un parámetro “where” donde se le ingresara sintaxis MYSQL en caso de ser nulo el campo se deberá de ingresar “1=1” para evitar una consulta con errores.

```
<<extends TableGateway>>
vacantesDireccion
+getvacantes_direcciones()
+getvacanteId(id)
+getvacantes_direccionesIdwhere(where)
+addvacante(array[])
+updatevacante(rfc,digito,num,data[])
+deletevacante(id)
```

Figura 3.27–Clase vacantes dirección.

3.6.4- Módulo Administrador

Es el cuarto módulo del sistema y se utilizará para que ingresen los usuarios registrados con tipo de perfil de administrador, este módulo administrará la información que el usuario genere, administrará y mantendrá actualizado los catálogos para que el sistema funcione de manera adecuada, dará de alta a usuarios de tipo de administrador, y se encargara de desactivar usuarios cualquier tipo de perfil, autorizarán las vacantes y podrán rechazarlas.

El controlador de la empresa es similar al módulo de alumno ya que contiene un constructor que revisará al servicio de autenticación y leerá los datos que contiene este servicio, si contiene datos validos los cuales significa que contiene un usuario y si tipo de usuario administrador de contener un usuario, se creará una identidad en Zend, se

asignara un layout nuevo, que será el de administrador, y hará que su sesión expire en 800 segundos.

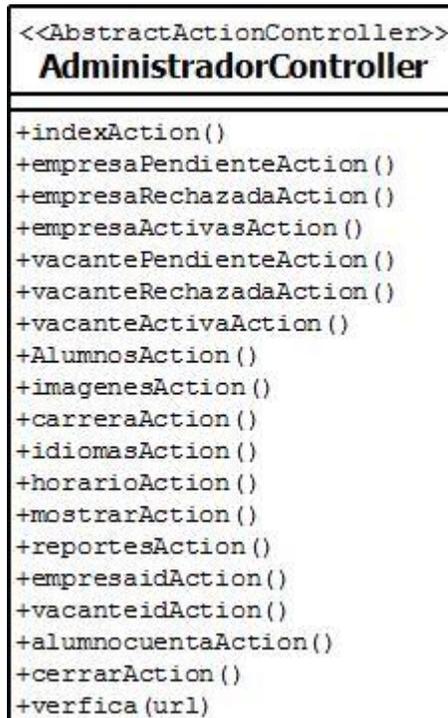


Figura 3.28–Clase Administrador controller

El controlador de este módulo a diferencia de los anteriores tiene muchos métodos (funciones) ya que este contiene muchas acciones que se pueden obtener mediante URL con método GET. La función reportesAction es un método que generará la información que se tiene en la base de datos mediante un PDF, para poder generar este PDF es necesario una librería DOMPdfModule que debe de instalarse con composer.

Con esta función se debe de crear un objeto de tipo PdfModel, creado el objeto se deberá de configurar la página como si se tratara de una impresión, también se pondrá la orientación de la página (“portrait”, “landscape”), se informará al objeto que contiene etiquetas HTML5. Al igual que una acción de ZEND en el controller se deberá de crear la vista que se le mostrará al cliente, pero deberá de ser con etiquetas sencillas de HTML ya que PDFDOM no acepta estilos CSS. Se generará el PDF y se mostrará como un archivo que se esté descargando en el navegador.

Administrador contiene modelos y clases que administra, aunque es dependiente de los modelos de los demás módulos, por lo que es necesario que ocupe los otros módulos mediante NAMESPACE que se llaman en la parte inicial del documento, este

los mandara a llamar, ejemplo: use `DOMPDFModule\View\Model\PdfModel`; Con este llamado podremos hacer objetos de la clase `PdfModel`, la otra manera de utilizar clases que no se encuentran en la clase en la que se está llamando es que cuando se genera el objeto se manda a llamar la ruta de la clase cuando se genera el objeto, ejemplo;

```
public function empresaPendienteAction() {
    $this->verfica("empresapendiente");
    $this->dbAdapter = $this->getServiceLocator()->get('Zend\Db\Adapter');
    $empresa = new \Administrador\Model\Empresa($this->dbAdapter);
    if ($this->getRequest()->isPost()) {
        $post = $this->getRequest()->getPost()->toArray();
        $empresa->updateEmpresa($post["id_rfc"], $post["digito_verificador"], $post);
    }
    $where = "autorizacion =0";
    $paginator = $empresa->getEmpresasWhere($where);
    // set the current page to what has been passed in query string, or to 1 if none set
    $paginator->setCurrentPageNumber((int) $this->params()->fromQuery('page', 1));
    // set the number of items per page to 8
    $paginator->setItemCountPerPage(8);
    $paginator = $alumno->getPostulaciones();
    $view = new ViewModel(array(
        'paginator' => $paginator
    ));
    $view->setTerminal(true);
    return $view;
}
```

Figura 3.29–Clase Empresa.

Los modelos que se tienen declarados en el módulo son: `Administrador`, `Carrera`, `Empresa`, `Estado`, `Horario`, `Idioma`, `Imágenes`, `Ingresos`, `Interesados`, `Login`, `SituacionAcademica` y `vacantes`, las cuales son clases genéricas para insertar, seleccionar, borrar y actualizar.

3.6.5.- Modelo de clase

Ya que se está trabajando con modelo vista controlador, las clases son modelos que utiliza el Framework para poder interactuar con la información sin tener que recurrir a el llamado a base de datos desde una vista de HTML, el controlador se encargará de obtener la información necesaria de los modelos, que a su vez ellos tienen herencias de otras clases que Zend Framework 2 le provee para recibir datos de la base de datos.

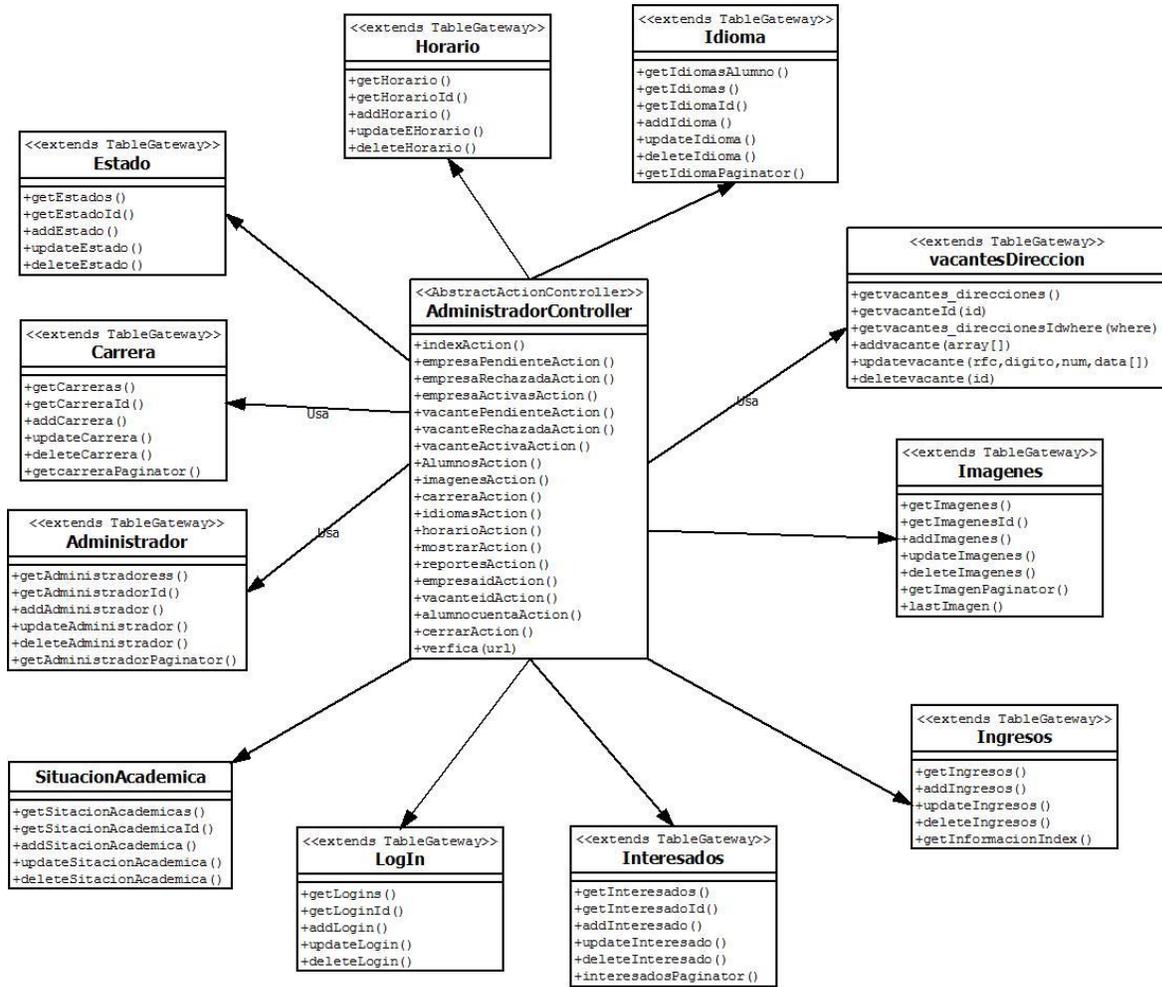


Figura 3.30—Diagrama de clase

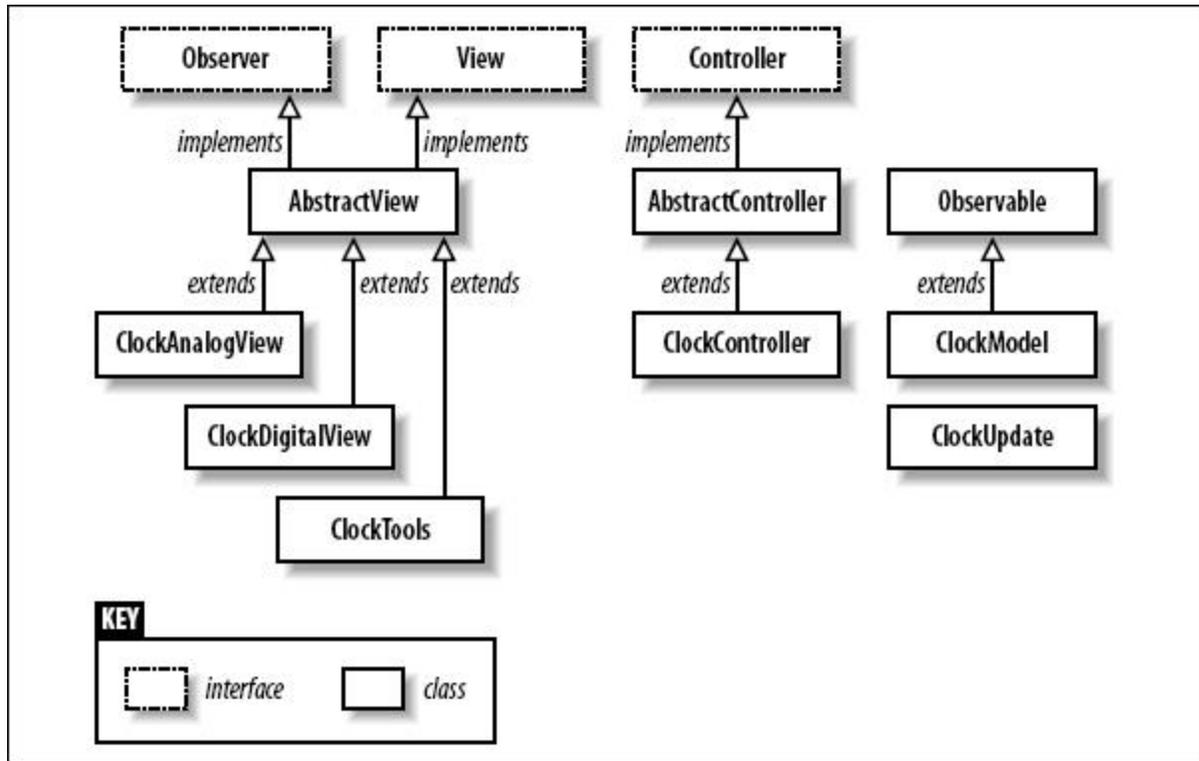


Figura 3.31-Implementación de clases Zend Framework

Conclusiones.

Sistema bolsa de trabajo

- El sistema cubrió las expectativas planteadas al inicio de esta tesis al proporcionar una solución eficaz, rápida, económica y de fácil acceso para los usuarios. Ya que se creó un sistema para la bolsa de trabajo conforme a las necesidades del departamento.
- El sistema diseñado evita los volúmenes excesivos de archivos y documentos físicos, lo que contribuye a un ahorro considerable de papel en beneficio del medio ambiente, que como profesionistas también debemos considerar.
- El sistema de bolsa de trabajo FES Cuautitlán cumplió con los objetivos que se plantearon al inicio de la investigación, aunque debido a deberes burocráticos no se lleva a la implementación e implantación del sistema, quedando con las interrogantes de la máxima capacidad del sistema, si no llevará a colapsos de información en grandes volúmenes.
- El sistema de bolsa de trabajo FES Cuautitlán implemento diversas tecnologías para la automatización de procesos porque es excelente vía para reducir errores, tiempo de ejecución, pérdida de información e incrementar la productividad (ver Anexo1).
- El empleo de bases de datos aumenta la velocidad de registro, recuperación y actualización de datos lo que permite tener información precisa y confiable en cualquier momento.

Anexo1.Tecnologías usadas

MYSQL

Es una tecnología estable que se puede instalar en diferentes sistemas operativos, Windows (con XAMPP y WAMP), LINUX (instalando un paquete mediante terminal), MAC (usando paquete nativo de la página <https://dev.mysql.com> o usando MAMP), es Open Source.

El diseño de base de datos carece de características que no se hicieron presentes en el análisis, diseño y desarrollo de la base de datos ya que se podría tener un mejor uso de los datos y recursos que se generan en el sistema por lo que sugerimos lo siguiente.

1. Se recomienda utilizar tablas en las cuales se tenga un histórico de los datos que se van generando para así poder generar reportes tales como: quien autorizo una vacante, empresa, dar de alta a un usuario, desde donde se conectó la persona, para poder generar estadísticas de quien lo está utilizando y en dónde.
2. La creación de una tabla de permisos utilizada por los administradores del sistema para que se pueda asignar un rol en específico, ejemplo; que un administrador solo tenga permiso para dar de alta vacantes, y que solo pueda visualizar las empresas con permiso.
3. Utilizar una diferente nomenclatura para el nombre de las tablas, <nombre principal del objeto ><nombre secundario> ejemplo, AlumnoDatosAcademicos el cual contiene que el objeto principal es el alumno y esa tabla contiene datos académicos del alumno.
4. Utilizar diferente nomenclatura para el nombre de las columnas de tablas, ya que al generar consultas con la unión de tablas se debe de generar un alias por cada tabla unida, se recomienda utilizar la siguiente nomenclatura <primeras 3 letras de la tabla><nombre de la columna> en caso de ser una tabla secundaria del objeto identificar las jorobas de camello y utilizar la primera letra de la tabla y las 3 últimas de la última palabra, ejemplo; tabla Alumno columna idCuenta, “AluldCuenta” , tabla AlumnoDatosAcademicos columna idCuenta “ADAcaldCuenta”
5. El uso de procedimientos almacenados para la inserción, búsqueda, actualización y borrado de datos, ya que con los procedimientos almacenados podemos administrar las reglas de negocio desde la base de datos, y en caso de cambio en estas poder administrarlas sin modificar programación.
6. Uso y manejo de funciones que puedan reducir el uso de código SQL y aumentar la reutilización de este.
7. El uso de indentación de tablas y columnas donde se genere el mayor uso de consultas a base de datos y contengan o se proyecte gran volumen de información.

A pesar de los puntos antes mencionados MYSQL es una tecnología que fue de gran utilidad en el sistema de bolsa de trabajo por su robustez, el uso de recursos y administración de ellos, velocidad y seguridad de los datos.

ZEND FRAMEWORK

Es una tecnología que se usó en el proyecto para el uso del patrón de diseño MVC, su programación es en PHP y es compatible con MYSQL y SQLSERVER (aunque no son los únicos) se implementa en entorno Web para el uso de navegadores como Firefox, Microsoft Edge, Google Chrome, Safari, etc.

Zend, aunque es utilizado en varias organizaciones tiene algunas limitaciones como son:

1. Requiere de una larga curva de aprendizaje.
2. La documentación que existe la mayoría se encuentra en inglés y es muy escasa.
3. Necesita una correcta configuración para que no se confunda en los módulos y rutas que existen.
4. Se necesita un buen servidor y una buena conexión de internet para que la pagina cargue de manera rápido o el uso de cache en servidor y cliente para su mejor visualización del mismo.
5. Manejo de objetos

Es una herramienta que nos ayuda a que el código este mejor acomodado y distribuido en la aplicación, el acomodo de los archivos ayuda a que si se cambian las reglas de negocio o se añadan más no se deba cambiar la interfaz del usuario y todo se maneje desde un BACK END. También ayuda a generar URLS amigables para el usuario que utiliza el sistema, ya que siempre se mostrará la misma URL en el módulo sin tener que mostrar extensiones de los archivos que se utilizan (index.php, vacantes.aspx).

El Framework provee un repositorio de componentes de alta calidad y que cuenten con soporte activo, cuenta con un sistema completo para el desarrollo de aplicaciones Web elaboradas en PHP5 y tiene una colaboración de la comunidad para una programación avanzada en PHP5.

BOOTSTRAP

Framework que Utiliza componentes y servicios creados por la comunidad Web, tales como: HTML5 shim, Normalize.css, OOCSS, jQuery UI, LESS y GitHub. Contiene La implementación de HTML5 + CSS3, incluye 12 columnas fijas o fluidas.

Uso de LESS, que es una ampliación a las famosas hojas de estilo CSS, pero a diferencia de éstas, funciona como un lenguaje de programación, permitiendo el uso de variables, funciones, operaciones aritméticas, entre otras, para acelerar y enriquecer los estilos en un sitio Web.

Ayuda a la generación del LAYOUT del proyecto para que la página se vea atractiva a los usuarios que la utilicen, ya que el uso de este es administrado por grids se tiene que acomodar todo de manera responsive (para móviles y tablets), de este modo se deberá pensar de forma global ya que cuando se muestre algún objeto que no esté optimizado para las dimensiones del dispositivo se mostrara de una forma no estética.

La última versión de la librería tiene un peso aproximado de 32 Kb minificada el archivo JavaScript (bootstrap.min.js) y de 103 Kb el CSS (bootstrap.min.css). Las versiones sin mifificar pesan todavía más. Por lo que utilizar estas librerías no afecta el rendimiento del sistema al momento de cargar la página. Se puede generar versiones personalizadas con solamente los componentes que se vayan a utilizar en la siguiente página <http://getbootstrap.com/customize>.

Bibliografía.

Abraham Silberschatz, H. F. (2002). *Fundamentos de bases de datos* (4ta ed.). MC Graw Hill.

Alicia Ramos Martín, M. d. (2011). *Aplicaciones Web* (9a ed.). Paraninfo.

Ángel Cobo, P. G. (2005). *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Madrid, España: Días de Santos.

Biaggini, C. C. (2013). *PHP programación web avanzada para profesionales*. Marcombo.

Campos, R. M. (2005). *Diseño de Páginas Web y Diseño Gráfico*. España: Ideaspropias.

Castillo, A. A. (2015). *Curso de programación Web Javascript, AJAX y jQuery*. Compyright.

Castillo, E. G. (2010). *PHP Profesional*. Lima, Perú: Macro E.I.R.L.

David Megías Jiménez, C. M. (2004). *Desarrollo de aplicaciones web*. UOC.

David Roldán Martínez, P. J. (2010). *Aplicaciones Web un enfoque práctico*. Madrid, España: Alfaomega .

Diseño básico de páginas web en HTML. (2009). Madrid, España: Vértice.

Elmasri, R. (2000). *Fundamentos de Bases de Datos* (Tercera ed.). España: Addison Wesley.

Euzo, E. A. (1998). *Oracle 8*. Madrid, España : ANAYA.

Facultad de Estudios Superiores Cuautitlán. (2015). *Informe de actividades*. México: UNAM.

Falgueras, B. C. (2003). *Ingeniería del Software*. Barcelona: UOC.

Falgueras, B. C. (s.f.). *Ingeniería del Software*. UOC.

Faudón, S. L. (2000). *Introducción a los sistemas de bases de datos* (Séptima ed.). México: PEARSON.

Fernando Alonso Amo, L. M. (2005). *Introducción a la Ingeniería del Software modelos de desarrollo de programas* (1er ed.). Madrid, España: Delta Publicaciones.

Hiroshi Maruyama, K. T. (2001). *Creación de sitios Web con XML y Java*. Madrid, España: Pearson Educación.

Jiménez, E. G. (2012). *Desarrollo de Software con NetBeans 7.1* (1er ed.). México: Alfaomega.

Laurent Debrauwer, F. V. (2005). *UML 2 Iniciación ejemplos y ejercicios corregidos*. ENI.

- Luis Alberto Casillas Santillán, M. G. (2007). *Bases de datos en MySQL*. UOC.
- Ma. Carmen Aranda Garrido, A. L. (s.f.). *Fundamentos de informática*. Universidad de Malagan/ Manuales.
- Martin Fowler, K. S. (1999). *UML gota a gota*. México: Pearson.
- Maza, M. Á. (2001). *Internet Javascript*. Madrid, España: Innovación y cualificación S.L.
- McIntire, P. (2008). *Técnicas innovadoras en Diseño Web*. Madrid, España: Anaya Multimedia.
- Miguel, J. V. (2013). *Implantación de aplicaciones web en entorno internet, intranet y extranet*. Paraninfo.
- Morales, R. C. (s.f.). *Introducción al Análisis de sistemas y la ingeniería de software*. EUNED Editorial Universidad Estatal a Distancia.
- Roger S. Pressman, P. (2010). *Ingeniería del software un enfoque práctico* (Séptima ed.). México: Mc Graw Hill.
- Sklar, D. (2005). *Introducción a PHP 5*. Madrid. España: Anaya Multimedia.
- Sommerville, I. (2007). *Ingeniería del Software*. España: Pearson Addison Wesley.
- Thibaud, C. (2006). *MySQL 5 Instalación Administración Programación*. ENI.
- Thomas M. Connolly, C. E. (2005). *Sistemas de bases de datos Un enfoque práctico para diseño, implementación y gestión*. (4a ed.). Madrid: PERSON.

Referencias electrónicas.

- (s.f.). Obtenido (20 de abril 2016) de <https://phpimpact.files.wordpress.com/2008/07/zend-blog-db-form-1.gif>.
- Asensio, R. M. (10 de agosto de 2012). Universidad de Murcia. Obtenido de Artículos. Informática. Servicios en internet Historia del desarrollo de las Aplicaciones Web: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- Zend Technologies Ltd. (14 de Enero de 2015). *Zend Framework*. Obtenido de Zend Framework: <https://packages.zendframework.com/releases/ZendFramework-2.3.4/ZendFramework-2.3.4-manual-en.pdf>