



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

---

---

SISTEMA ADMINISTRADOR DE LA BOLSA DE TRABAJO  
FACULTAD DE PSICOLOGÍA, UNAM

TESIS PROFESIONAL  
QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

BERMUDEZ VELASCO PABLO

MILCHORENA MENDOZA ALBERTO



DIRECTOR DE TESIS:  
ING. ORLANDO ZALDÍVAR ZAMORATEGUI

CIUDAD UNIVERSITARIA, MÉXICO, D.F. 2008



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.

# *Agradecimientos*

## **A Dios.**

Porque me has permitido la vida y me has dejado llegar hasta donde estoy, pero sobre todo porque me has bendecido con una gran familia y me has dado la esperanza para continuar cuando más lo he necesitado.

## **A mi madre Alicia.**

Pues me has brindado tu amor y comprensión y te has sacrificado grandemente por mí desde siempre. Te agradezco porque siempre he contado contigo, eres una persona excepcional y un ejemplo a seguir. Sin ti nunca hubiera cumplido mis sueños, por lo que tienes todo el mérito de mi trabajo.

## **A mi padre Esteban.**

Pues me quieres y apoyas en todo momento y me has dado tu ejemplo. Siempre has procurado guiarme por el camino más justo. Te agradezco también porque te has sacrificado y esforzado por sacar adelante a nuestra familia.

## **A mis hermanas Alma, Farah y Alejandra.**

Por el todo el amor que me han brindado y por permitirme crecer junto a ustedes. Les agradezco por estar siempre unidos y compartir grandes momentos junto a mí. Me hacen sentir muy orgulloso de ser su hermano.

## **A mi amigo Ely Daniel.**

Por compartir conmigo algunos de los mejores momentos de mi vida y por brindarme un apoyo fraterno en algunos de los peores. Siempre te agradeceré por todo lo que has hecho por mí y sabes que siempre podrás contar conmigo hermano.

## **A mis amigos Erika y Rene.**

Porque trabajando junto a ustedes he vivido una de las etapas más felices de mi vida, y me han brindado su amistad de forma sincera. Les agradezco por sus enseñanzas y consejos y por todas las experiencias halladas en nuestro andar. Siempre los tengo presentes en mi pensamiento.

### **A la Universidad Nacional Autónoma de México.**

Pues me ha permitido crecer de forma personal y profesional. Le agradezco porque me ha dado la satisfacción de sentirme orgullosamente universitario.

### **A la Facultad de Ingeniería.**

Le agradezco por las enseñanzas y satisfacciones que me ha dado, tanto dentro como fuera de sus aulas. También porque me brinda la oportunidad de poder ser un profesionalista útil a la sociedad mexicana.

### **Al Ing. Orlando Zaldívar Zamorategui.**

Por su paciencia y consejos brindados en el desarrollo del presente trabajo de tesis. Le agradezco por tratar de conferirnos parte de su conocimiento y guiarnos para culminar correcta y exitosamente este esfuerzo conjunto.

### **Finalmente.**

Quiero agradecer a todas aquellas personas que han significado algo muy especial para mí y con quienes he compartido mi vida en algún momento. Aunque no estén físicamente conmigo, siempre las llevo en mi pensamiento y en mi corazón pues en algún instante fueron una fuente de apoyo e inspiración para concluir este gran sueño.

**Gracias.**

**Pablo Bermudez Velasco.**

## *Agradecimientos*

A *Dios*, por acompañarme siempre en todo momento de dificultad, por ser mi guía y por todas las bendiciones que me ha concebido.

A mis padres *Jesús* y *Aleja*, por darme la vida, por todo su apoyo incondicional, amor y ternura, por una maravillosa formación y ser un ejemplo para mi.

A mis hermanos *Alejandra* y *José Miguel*, por creer en mi, por todos sus consejos, amor y ser para mí dos grandes bendiciones que me otorgó Dios.

A mis familiares: *abuelos, tíos, primos y sobrinos*, por haber compartido conmigo mis triunfos y fracasos, por todo su amor, cariño y apoyo.

A la *Facultad de Ingeniería*, por haberme formado profesionalmente y hecho lo que ahora soy.

A mis *profesores* de la *Facultad de Ingeniería*, por transmitirme sus conocimientos, experiencias y su entusiasmo de seguir siempre adelante.

A mi *director de tesis*, el *Ing. Orlando Zaldivar*, por su dedicación y empeño para que se llevara a cabo este trabajo.

Y a todas las personas que me brindaron su apoyo para poder realizar con éxito esta tesis.

A todos ellos, ¡gracias!.

Con mucha admiración y respeto.

*Alberto Milchorena Mendoza*

## *Pensamientos*

### No te quejes

Nunca te quejes de nadie, ni de nada, porque fundamentalmente tu has hecho lo que querías en tu vida. Acepta la dificultad de edificarte a ti mismo y el valor de empezar corrigiéndote. El triunfo de verdadero hombre surge de las cenizas de su error.

Nunca te quejes de tu soledad o de tu suerte, enfréntala con valor y acéptala. De una manera u otra es el resultado de tus actos y prueba que tu siempre has de ganar.

No te amargues de tu propio fracaso ni se lo cargues a otro, acéptate ahora o seguirás justificándote como un niño. Recuerda que cualquier momento es bueno para comenzar y que ninguno es tan terrible para claudicar.

No olvides que la causa de tu presente es tu pasado así como la causa de tu futuro será tu presente.

Aprende de los audaces, de los fuertes, de quien no acepta situaciones, de quien vivirá a pesar de todo, piensa menos en tus problemas y más en tu trabajo y tus problemas sin eliminarlos morirán.

Aprende a nacer desde el dolor y a ser más grande que el más grande de los obstáculos, mírate en el espejo de ti mismo y serás libre y fuerte y dejarás de ser un títere de las circunstancias porque tu mismo eres tu destino.

Levántate y mira el sol por las mañanas y respira la luz de amanecer. Tú eres parte de la fuerza de tu vida, ahora despiértate, lucha, camina, decídete y triunfarás en la vida; nunca pienses en la suerte, porque la suerte es: el pretexto de los fracasados.

**Pablo Neruda**

## Índice de contenido

<b>Capítulo 1 Planteamiento del problema</b>	<b>1</b>
1.1 Marco teórico de referencia institucional	3
<b>Capítulo 2 Marco teórico conceptual</b>	<b>9</b>
2.1 La ingeniería de software	11
2.2 Redes de computadoras	21
2.3 Lenguajes de programación	24
2.4 Bases de datos	28
2.5 UML	32
2.6 Gestión de riesgos	40
2.7 Mantenimiento del software	42
<b>Capítulo 3 Análisis y diseño</b>	<b>45</b>
3.1 Análisis de factibilidad	47
3.2 Diagramas de casos de uso	53
3.3 Diagramas de clases	56
3.4 Diagramas de secuencia	58
3.5 Diagramas de colaboración	59
3.6 Diagramas de flujo	60
3.7 Normalización	62

<b>Capítulo 4 Construcción del sistema</b>	<b>67</b>
4.1 Administración del sistema	69
4.1.1 Secuencia de pantallas	69
4.1.1.1 Descripción de las pantallas comunes para los diferentes usuarios	69
4.1.1.2 Flujo de pantallas para el Administrador del sistema	70
4.1.1.3 Flujo de pantallas para las Empresas	86
4.1.1.4 Flujo de pantallas para los Egresados	90
4.1.1.5 Flujo de pantallas para las Secretarias	94
4.1.1.6 Flujo de pantallas para los Asesores	96
4.2 Realización de pruebas	99
<b>Capítulo 5 Resultados y conclusiones</b>	<b>107</b>
5.1 Resultados	109
5.2 Mantenimiento	110
5.3 Conclusiones	111
<b>Bibliografía</b>	<b>113</b>



# Capítulo 1

## Planteamiento del problema



## **1.1 Marco teórico de referencia institucional**

Dentro de la organización de la Universidad Nacional Autónoma de México (UNAM) existen diversas facultades y escuelas en donde se han forjado millones de profesionistas, quienes día con día contribuyen al progreso del país realizando múltiples actividades tanto de índole laboral, así como prácticas profesionales y de servicio social.

### **La Facultad de Psicología**

El Colegio de Psicología se formó originalmente como parte de la Facultad de Filosofía y Letras (FFyL), donde se elaboró el primer plan de estudios que se impartió desde 1938.

Después, el 27 de febrero de 1973, y tras un gran esfuerzo de un grupo de psicólogos mexicanos que lucharon inicialmente por incluir a la psicología dentro de los estudios de bachillerato y posteriormente para incorporar su enseñanza en los estudios superiores, el Colegio de Psicología fue transformado en la actual Facultad de Psicología (FP), logrando además de crear sus propias maestrías y doctorados

La currícula de psicología ha tenido algunos cambios a través del tiempo, siendo los más importantes los establecidos en 1960, 1967 y 1971.

En 1960, la currícula tuvo su inicio junto con la profesionalización de la carrera y no difería de la orientación clínica y especulativa que tenía el plan anterior; en 1967 fue modificado el plan de estudios, tomando como base el impartido por la Universidad Veracruzana, en donde por primera vez en México se dio la enseñanza de la psicología con una orientación netamente experimental y fueron enfatizados diversos aspectos metodológicos con el propósito de hacer de la psicología una ciencia más formal, aunque lamentablemente no contó con las condiciones necesarias para aplicarla exitosamente. No fue sino hasta 1971 cuando quedó establecida la currícula que hasta el momento sigue vigente, la cual consiste en estudiar un tronco básico a cursarse dentro de los seis primeros meses de la carrera y seis áreas de semi-especialización, que son cubiertas en los últimos tres semestres.

### **La Coordinación de los Centros de Servicios a la Comunidad Universitaria y al Sector Social**

Inició sus labores a partir de julio de 2001 y brinda sus servicios a la sociedad a través de diversos centros y programas especializados, ubicados tanto dentro de la FP, así como fuera de las instalaciones de Ciudad Universitaria.

Tiene como principal objetivo el planear, organizar, supervisar y colaborar en diversas actividades académicas y administrativas a fin de cumplir con los objetivos de servicio, docencia, investigación,

difusión y extensión universitaria, así como promover el desarrollo de cada uno de los centros y programas que lo integran, como son:

#### Centro comunitario “Dr. Julián MacGregor y Sánchez Navarro”

Se fundó en mayo de 1981; sus objetivos son participar en la profesionalización de estudiantes de las distintas áreas de la carrera de psicología y del posgrado, así como contribuir al conocimiento de las situaciones y problemáticas de la comunidad.

Otorga atención psicológica a través de distintos programas dirigidos a la niñez, adolescencia, adultez y tercera edad, en áreas como adicciones, violencia, atención educativa, salud sexual y reproductiva, relaciones familiares y trastornos de la alimentación, entre otras; todo esto, a nivel preventivo y de interpretación psicoterapéutica.

#### Centro de investigación y servicios de educación especial (CISEE)

Se creó en febrero de 1977 con la finalidad de promover el beneficio social y la mejora de la calidad de vida a nivel personal, familiar y social de los individuos con discapacidad intelectual y/o necesidades de apoyo especial. Actualmente ofrece atención educativa y psicológica a este grupo de personas y sus familiares, además de asesorar a los interesados en el tema.

#### Centro de servicios psicológicos “Dr. Guillermo Dávila”

Inició sus servicios en 1981 y tiene como objetivo apoyar y fortalecer la formación teórico-práctica de los estudiantes de licenciatura y posgrado, con la supervisión de especialistas. El centro proporciona servicio psicoterapéutico, bajo diferentes aproximaciones teóricas, a niños, adolescentes y adultos en forma individual, grupal, de pareja y familiar. Se atienden problemas emocionales, trastornos de la alimentación, sexualidad, dolor crónico, estrés, entre otros.

#### Centro comunitario de atención psicológica “Los Volcanes”

Surgió por iniciativa del Movimiento Popular de Pueblos y Colonias del Sur, ante la necesidad de atender el incremento de problemas emocionales, conductuales, de aprendizaje, violencia intrafamiliar y adicciones, con la población del sur de Tlalpan. A partir del convenio establecido con la FP, este centro fue inaugurado en noviembre de 2004. Son otorgados diversos servicios de atención psicoterapéutica a niños, adolescentes y adultos y se desarrolla un trabajo preventivo a través de conferencias, talleres y otras acciones comunitarias. La formación en la práctica y el servicio social es llevada a cabo bajo una estrecha supervisión por parte de los responsables académicos.

## Programas de conductas adictivas

Fue creado en 1989, brinda servicios de diagnóstico, prevención, orientación, tratamiento, rehabilitación, evaluación y seguimiento en conductas adictivas.

Sus servicios son caracterizados por un abordaje científico y humano de alta calidad y están dirigidos a la comunidad universitaria y al público en general, que presenten o estén en riesgo de presentar algún problema de alcoholismo, tabaquismo o farmacodependencia.

## Programa de sexualidad humana

Este programa fue creado para contribuir desde el quehacer universitario en la transformación de condiciones individuales y sociales, y así favorecer y fortalecer la salud sexual integral de la población desde la equidad de género y el respeto a los derechos humanos. El trabajo es realizado bajo un enfoque multidisciplinario, para lo cual se actúa a nivel de docencia, investigación y servicio con el fin de difundir, conocer y tratar de resolver los diferentes aspectos que contempla la sexualidad humana.

## **El Departamento de Servicio Social y Bolsa de Trabajo (DSSyBT)**

Está encargado de controlar, administrar y regular diversos procesos administrativos relativos a la realización del servicio social, tal y como se enmarca en la Normatividad Administrativa de la UNAM, específicamente en el Reglamento General de Servicio Social de la Universidad Nacional Autónoma de México, Capítulo III y a su vez regula los procesos concernientes a la Bolsa de Trabajo para los egresados de esa facultad.

Los objetivos del departamento en cuanto a la coordinación y realización del servicio social son:

- Coordinar a los alumnos que deseen prestar su servicio social.
- Informarles acerca de las instituciones donde pueden realizar su servicio social.
- Controlar la documentación y papeleo de los alumnos.
- Regular y controlar el listado instituciones receptoras de alumnos de servicio social, tanto internas (pertenecientes a la UNAM), como externas (no pertenecientes a la UNAM).
- Informar de cualquier trámite y procedimiento que sea llevado a cabo dentro del departamento.

Como parte de una organización bien estructurada, la FP también debe contar con un mecanismo para vincular a los estudiantes con el mercado laboral y a su vez, facilitar a las empresas interesadas en contratar personal altamente capacitado para realizar funciones específicas.

Para ello son difundidas diversas ofertas de trabajo de los sectores público y privado registradas en el programa de Bolsa de Trabajo, el cual sirve como enlace entre los estudiantes y egresados de la FP y el mercado de trabajo dentro del ámbito de su especialidad.

Los objetivos de la Bolsa de Trabajo son:

- Coordinar a los egresados de la FP que deseen integrarse al mercado laboral actual.
- Informar a los egresados acerca de las empresas o instituciones donde sea solicitado personal capacitado y egresado de la FP, de acuerdo con su perfil personal.
- Regular y controlar el listado empresas o instituciones, tanto internas (pertenecientes a la UNAM), como externas (no pertenecientes a la UNAM) que deseen inscribirse en la Bolsa de Trabajo de la FP.

### **Necesidad del Sistema de Administración de la Bolsa de Trabajo**

El presente trabajo surge por la necesidad de contar con procesos administrativos más eficientes dentro de ese departamento, debido por un lado a la carencia de recursos tanto humanos como económicos que existe, y por otro lado, a la creciente demanda de los estudiantes por obtener servicios e información fiables y oportunos, pues día con día, el número de interesados en la búsqueda de empleo, se incrementa de forma considerable.

En recientes fechas el DSSyBT se ha enfrentado con un problema de administración tanto de procesos como de usuarios, el cual tiende a sobrepasar sus recursos humanos disponibles y en tiempo, así como incrementar la complejidad en la toma de decisiones debido a la gran cantidad de variables que son manejadas. Para solventar este problema es necesario un sistema robusto, capaz de automatizar gran parte de las tareas a fin de descargar todo este trabajo y permitir aprovechar el tiempo en las actividades paralelas que desarrolla el departamento, además de tener la posibilidad de administrarlo de forma segura y de contar con una gestión vía web.

Así mismo, se busca un planteamiento y desarrollo de la página web del DSSyBT, a fin de aumentar la eficiencia en la divulgación de actividades del área. Es necesario contar con mayor facilidad de acceso a la información de interés para el usuario, como son ferias de empleo, seminarios, etc., pues la que se encuentra actualmente mostrada al público en general no cumple con las expectativas descritas.

Como referencia, los requisitos solicitados por el DSSyBT son mostrados en la tabla 1:

<b>Requerimientos</b>
Automatizar el proceso de inscripción de empresas e instituciones en la Bolsa de Trabajo (BT).
Automatizar el proceso de inscripción de egresados en la BT.
Proponer un sistema para automatizar y controlar la puesta en línea de ofertas de trabajo brindadas por las empresas e instituciones.
Agilizar los trámites que se llevan a cabo en la búsqueda de una oferta para determinado candidato (anteriormente 15 días como mínimo).
Mejorar el proceso de seguimiento de las empresas o instituciones a fin de que cumplan con lo establecido en sus ofertas.
Disminuir las cargas de trabajo dentro del departamento.
Dar a conocer el departamento a empresas e instituciones con el fin de que lo identifiquen y lo tomen en cuenta.
Construir una base de datos de los alumnos inscritos la BT.
Construir una base de datos de las empresas e instituciones que desean contratar personal.
Disponer de información que coadyuve a alcanzar los objetivos dentro del área de BT en el momento oportuno (consultas).

Tabla 1. Requerimientos solicitados por el DSSyBT

Es de suma importancia un sistema computacional integral, que ejecute la mayoría de las actividades de gestión referentes a la BT, que actualmente desempeñan en su mayor parte secretarías, asesores y el jefe del departamento.

Nosotros como futuros ingenieros en computación plantearemos la solución óptima para el problema mediante los conocimientos adquiridos a lo largo de nuestra estadía en la Facultad de Ingeniería, para ello haremos uso de las herramientas, métodos y procesos que nos brinda la ingeniería de software, a fin de contar con un sistema de calidad, funcional y eficiente, que se adapte a las necesidades específicas del DSSyBT.





# Capítulo 2

## Marco teórico conceptual



## 2.1 La ingeniería de software

Para poder llevar cabo el presente proyecto, necesitamos hacer uso de la ingeniería de software (IS), la cual es la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software.

La IS es una tecnología multicapa, como muestra la figura 2.1.1.



Figura 2.1.1 Capas de la ingeniería de software

Cualquier proyecto de ingeniería (incluida la IS) debe apoyarse siempre en criterios y normas de calidad a fin de obtener productos que cumplan siempre con las expectativas del cliente.

Por otro lado, el fundamento de la IS de forma específica es la capa de proceso.

*El proceso de la Ingeniería de software es el elemento que mantiene juntos los estratos de la tecnología y que permite el desarrollo racional y a tiempo del software de computadora. El proceso del software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen los fundamentos, se asegura la calidad y el cambio se maneja de forma apropiada.<sup>1</sup>*

Los métodos de la ingeniería de software nos dicen cómo implementar el desarrollo de un proyecto ya que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte.

Por otro lado, las herramientas de la IS integran un proceso automático o semi-automático para el proceso y para los métodos.

---

<sup>1</sup> Pressman, Roger S. *Ingeniería del software, Un enfoque práctico*, McGraw-Hill, 6ª Edición, Madrid España, 2005. Pág. 24.

## Modelos de proceso de software

Para resolver los problemas en la industria, el ingeniero de software o el equipo encargado de un proyecto debe trabajar con un modelo de proceso (o paradigma) de ingeniería del software. El modelo de proceso de software se selecciona con base en la naturaleza del proyecto, la aplicación, los métodos, las herramientas a utilizarse, los controles y entregas que se requieren.

Existen cuatro etapas en todo desarrollo de software: el *status quo* o estado actual, la definición de problemas, el desarrollo técnico y la integración de soluciones, las cuales las podemos ver como un ciclo de resolución de problemas.

El *status quo* representa el estado actual de sucesos; la definición de problemas identifica el problema específico a resolverse; el desarrollo técnico resuelve el problema con la ayuda de alguna tecnología y la integración de soluciones ofrece los resultados solicitados.

A medida que el trabajo avanza hacia un sistema completo podemos implementar las cuatro etapas descritas anteriormente de manera recursiva dependiendo de las necesidades del usuario y especificaciones técnicas del desarrollador del software.

### **El modelo lineal secuencial**

Llamado también “ciclo de vida clásico” o “modelo en cascada” sugiere un enfoque sistemático y secuencial para el desarrollo del software. La figura 2.1.2 ilustra el modelo, el cual comprende las siguientes actividades:

#### Ingeniería y modelado de sistemas

Comprende los requisitos que nos proporciona el cliente con una pequeña parte de análisis y de diseño.

#### Análisis de los requisitos del software

Los requisitos se clasifican y se estudian a fin de comprender la información para poder construir el programa deseado en función del comportamiento, rendimiento e interconexión del software.

#### Diseño

Se traducen los requisitos en una representación del software donde se puede evaluar su calidad antes de que comience la codificación.

### Generación de código

Se lleva a cabo la traducción del diseño a una forma entendible para la computadora. Es importante mencionar que si se efectúa el diseño de una forma detallada, la generación de código se realiza mecánicamente.

### Pruebas

Se realizan pruebas para la detección de errores y se asegura que la entrada definida produce los resultados esperados.

### Mantenimiento

Son los cambios que se realizan para mejorar o ajustar el software después que se haya entregado al cliente.

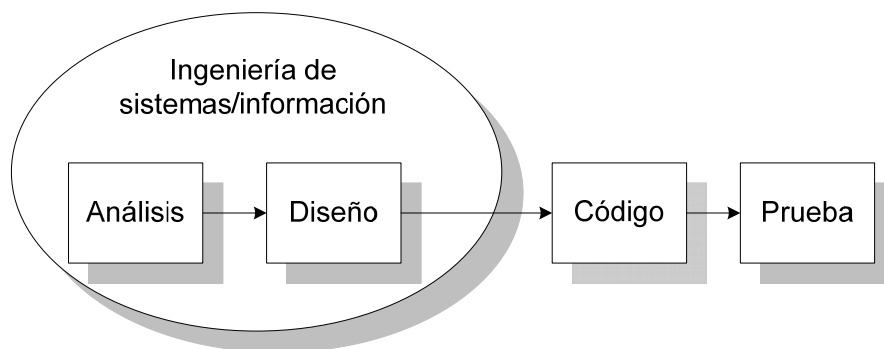


Figura 2.1.2 Ingeniería y modelado de sistemas/información

El modelo lineal secuencial es el paradigma más antiguo y más usado en la ingeniería del software, sin embargo entre los problemas que se encuentran algunas veces en el modelo lineal secuencial son:

1. Los proyectos reales raras veces siguen el modelo secuencial que propone el paradigma.
2. Es muy difícil que el cliente nos proporcione explícitamente todos los requisitos.
3. El cliente deberá ser paciente pues una primera versión del trabajo estará disponible hasta que el proyecto esté en una fase muy avanzada.

### **El modelo de construcción de prototipos**

El modelo de construcción de prototipos lo podemos utilizar cuando el cliente define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados que necesitamos.

Este paradigma comienza con la recolección de requisitos y la definición de los objetivos globales para el software, se construye un prototipo que es evaluado por cliente/usuario y sirve para refinar

los requisitos del software a desarrollar, permitiendo que el desarrollador comprenda mejor lo que se necesita hacer. Esto es mostrado en la figura 2.1.3.

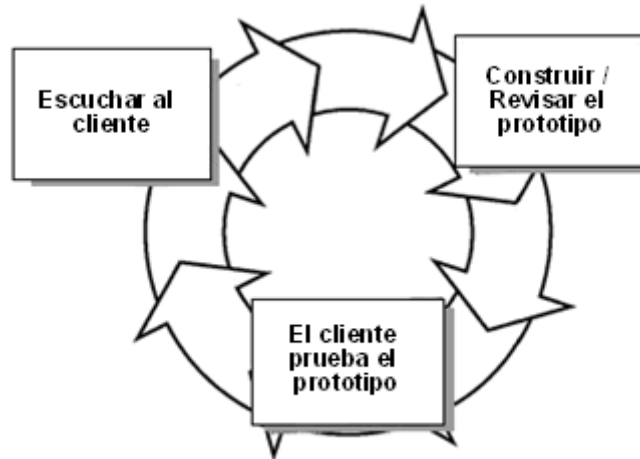


Figura 2.1.3 Modelo de construcción de prototipos

El prototipo nos sirve para identificar los requisitos del software; una vez identificado y comprendido lo que se necesita hacer, el prototipo se deberá desechar y comenzar de nuevo.

La construcción de prototipos también puede ser problemática por las siguientes razones:

1. Cuando el cliente es informado de que el producto debe ser construido otra vez para que puedan mantenerse los niveles de calidad, éste no lo entiende y pide que sólo se le hagan pequeños ajustes para que pueda obtenerse del prototipo un producto final.
2. El desarrollador, a menudo, hace compromisos de implementación para hacer que el prototipo funcione rápidamente.

### **El modelo DRA**

El Desarrollo Rápido de Aplicaciones (DRA) es un modelo de proceso de desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. Se deben conocer y comprender perfectamente los requisitos y limitar el alcance del proyecto, como se muestra en la figura 2.1.4.

Para que el DRA pueda funcionar correctamente y crear un sistema completamente funcional dentro de periodos cortos, deben ser desarrolladas las siguientes fases:

## Modelado de gestión

Obtenemos información para administrarla y modelarla de las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?

## Modelado de datos

Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre los objetos.

## Modelado del proceso

Los objetos de datos definidos en la fase de modelado de datos se transforman para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir o recuperar un objeto de datos.

## Generación de aplicaciones

El DRA se basa en el trabajo realizado para utilizar componentes de programas ya existentes o crear componentes reutilizables.

## Pruebas y entrega

Como el proceso DRA utiliza la reutilización, se comprueban muchos de los componentes de los programas. Sin embargo, se deben comprobar todos los componentes nuevos.

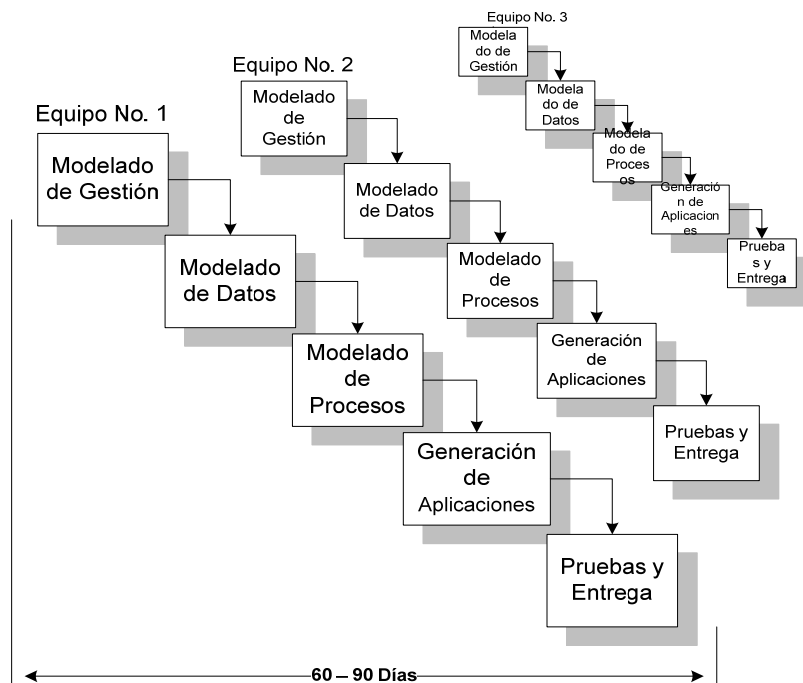


Figura 2.1.4 Modelo de Desarrollo Rápido de Aplicaciones

Al igual que todos los modelos de proceso, el enfoque DRA tiene inconvenientes:

1. Para proyectos grandes, el DRA requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.
2. DRA requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar el sistema.
3. Si un sistema no puede modularizar adecuadamente, la construcción de los componentes necesarios para DRA será problemática.
4. DRA no es adecuado cuando los riesgos técnicos son altos, es decir, cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el software nuevo requiere un alto grado de interoperatividad.

### Modelos evolutivos de proceso de software

Debido a que el software evoluciona con el tiempo, los requisitos de gestión y de productos a menudo cambian para cumplir la presión competitiva, se deben de comprender perfectamente los requisitos y tomar en cuenta las extensiones del sistema dentro de alguna fecha tope. En estas y en otras situaciones similares, los ingenieros del software necesitan un modelo de proceso que se ha diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo.

Los modelos evolutivos son iterativos. Se caracterizan porque ayudan a los ingenieros del software a desarrollar versiones más completas en sus proyectos.

### **El modelo incremental**

El modelo incremental (figura 2.1.5), combina elementos del modelo lineal secuencial con la de construcción de prototipos. Como muestra la figura, el modelo incremental sigue secuencias lineales mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento, éstos son versiones incompletas del producto final pero utilizables; cada incremento se construye sobre aquél que ya ha sido entregado. Es particularmente útil cuando no se cuenta con suficiente personal para una implementación completa en la fecha límite que se haya establecido para el proyecto.



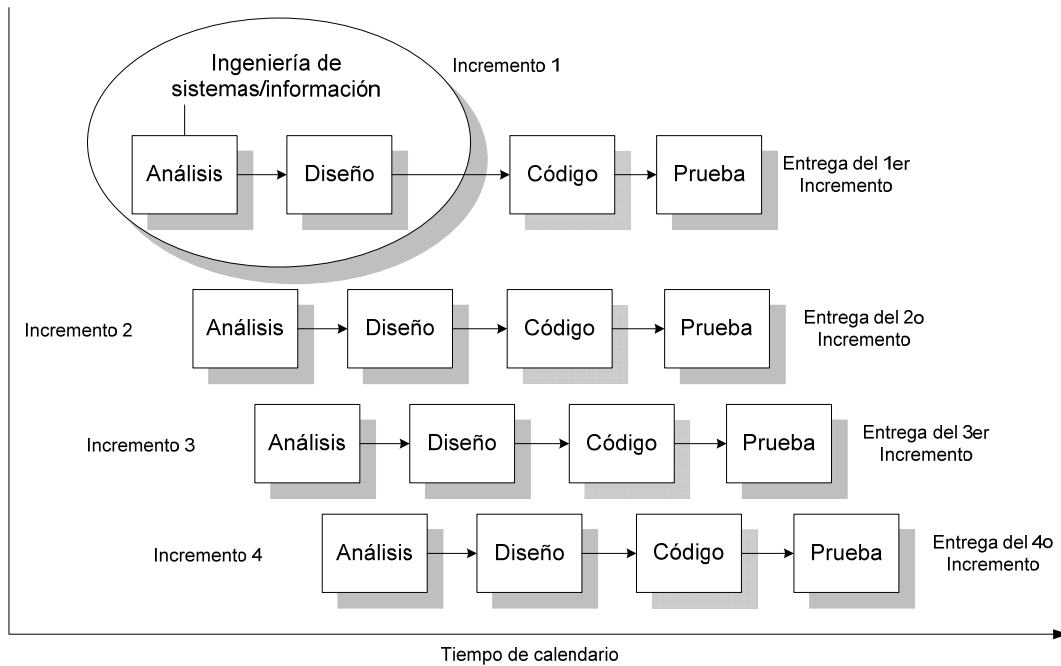


Figura 2.1.5 El modelo incremental

## El modelo en espiral

El modelo en espiral es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa y la reducción de riesgos del modelo de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas. La figura 2.1.6 representa un modelo en espiral que contiene seis regiones de tareas:

### Comunicación con el cliente

Son las tareas requeridas para establecer una comunicación con el cliente.

### Planificación

Son las actividades para establecer un plan para el trabajo de la IS. Describe las tareas técnicas que deben realizarse, los recursos que serán requeridos, los productos de trabajo que han de producirse y un programa de trabajo.

### Análisis de riesgos

Son las tareas requeridas para evaluar riesgos técnicos y de gestión.

### Ingeniería

Son las tareas requeridas para construir una o más representaciones de la aplicación.

## Construcción y adaptación

Son las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

## Evaluación del cliente

Son las evaluaciones realizadas por el cliente al proyecto.

A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.

Al igual que otros paradigmas, el modelo en espiral tiene inconvenientes, ya que requiere una considerable habilidad para la evaluación del riesgo, ya que si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas.



Figura 2.1.6 Modelo en espiral

## El modelo espiral WINWIN

El modelo en espiral WINWIN (figura 2.1.7) define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral, definiéndose las siguientes actividades:

1. Identificación del sistema o subsistemas
2. Determinación de las condiciones de victoria
3. Negociación de las condiciones de victoria tanto para el cliente como para el desarrollador

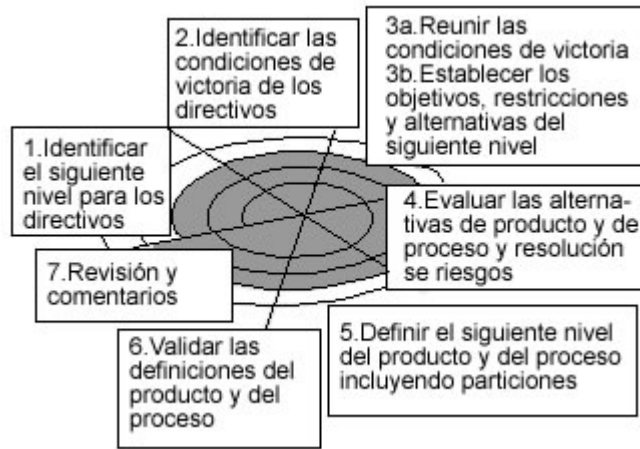


Figura 2.1.7 Modelo en espiral WINWIN

Además de la negociación inicial, el modelo en espiral WINWIN introduce tres hitos, llamados puntos de fijación que ayudan a establecer la completitud de un ciclo alrededor de la espiral. El primer punto de fijación llamado objetivos del ciclo de vida, define un conjunto de objetivos para cada actividad principal de ingeniería del software (por ejemplo un conjunto de objetivos asociados a la definición de los requisitos del producto/sistema del nivel más alto). El segundo punto de fijación, llamado arquitectura del ciclo de vida, establece los objetivos que se deben conocer mientras que se define la arquitectura del software y el sistema (por ejemplo, el equipo del proyecto de software debe mostrar que ha evaluado la funcionalidad de los componentes del software reutilizables y que ha considerado su impacto en las decisiones de arquitectura). La capacidad operativa inicial es el tercer punto de fijación y representa un conjunto de objetivos asociados a la preparación del software para la instalación/distribución, preparación del lugar previamente a la instalación, y la asistencia precisada de todas las partes que utilizará o mantendrá el software.

### **El modelo de desarrollo concurrente**

La mayoría de los modelos de procesos de desarrollo del software son dirigidos por el tiempo; un modelo de proceso concurrente está dirigido por las necesidades del usuario, las decisiones de la gestión y los resultados de las revisiones.

El modelo de proceso concurrente se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas. La figura 2.1.8 proporciona una representación esquemática de una actividad dentro del modelo de proceso concurrente.

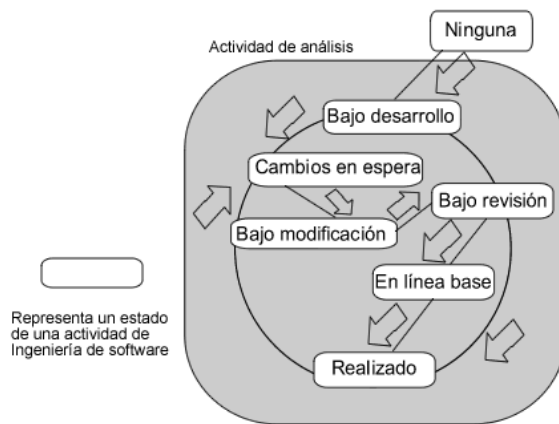


Figura 2.1.8 Modelo de desarrollo concurrente

Todas las actividades existen concurrentemente, pero residen en estados diferentes. El modelo de proceso concurrente define una serie de acontecimientos que dispararán transiciones de estado a estado para cada una de las actividades de la IS.

El modelo de proceso concurrente se utiliza a menudo como el modelo de desarrollo de aplicaciones cliente /servidor.

### El modelo de desarrollo basado en componentes

El modelo de desarrollo basado en componentes (figura 2.1.9) incorpora muchas características del modelo en espiral. Es evolutivo, existe un enfoque iterativo para la creación del software y configura aplicaciones desde componentes preparados de software (llamados clases).

La actividad de la ingeniería comienza con la identificación de clases candidatas. Esto se lleva a cabo examinando los datos que se van a manejar por parte de la aplicación y el algoritmo que se va a aplicar para conseguir el tratamiento adecuado de los datos. Los datos y los algoritmos correspondientes se empaquetan en una clase y se almacenan en una biblioteca de clases o diccionario de datos. Una vez identificadas las clases candidatas, la biblioteca se examina para determinar si estas clases existen; si existen, se extraen de la biblioteca y se vuelven a utilizar, si no es éste el caso, se crean y se almacenan en la biblioteca. Se compone así la primera iteración de la aplicación a construirse, mediante las clases extraídas y las generadas. El flujo del proceso regresa a la espiral y volverá a introducir por último la iteración ensambladora de componentes a través de la actividad de ingeniería.

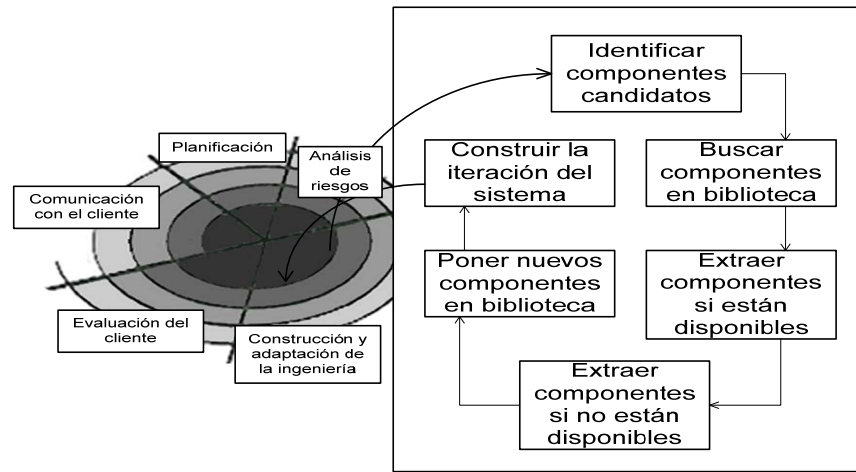


Figura 2.1.9 Modelo de desarrollo basado en componentes

El modelo de desarrollo basado en componentes conduce a la reutilización del software y la reutilización proporciona beneficios a los ingenieros de software.

## 2.2 Redes de computadoras

Una red de computadoras se compone, por un lado, de un conjunto de dispositivos de propósito específico (léase concentradores, unidades de almacenamiento, computadoras, impresoras, etc.), conectados mediante un medio físico (terrestre o aéreo) con la finalidad de intercambiar información y compartir recursos (medios de almacenamiento, periféricos, etc.).

Por otro lado, un aspecto fundamental para formar una red de computadoras es el software utilizado en la administración de hardware mencionado anteriormente, el control de la transmisión sobre el medio, así como en el manejo de protocolos que gobiernan las relaciones entre los equipos y dispositivos. Éste puede ser un sistema operativo como Windows, que integra diversos controladores de hardware de red o software propietario, proporcionado por el fabricante del hardware mencionado.

### Ventajas al utilizar una red de computadoras

Actualmente las empresas o instituciones que no cuenten con una base de información actualizada y confiable están condenadas a la extinción. Las redes de computadoras son una herramienta ideal para trabajar en equipo tanto dentro de una organización así como fuera de ella.

Entre las ventajas más notorias que ofrece la utilización de una red, se encuentran:

- Posibilidad de compartir información entre varios usuarios, permitiendo evitar duplicidad de trabajo.

- Posibilidad de compartir periféricos o equipos costosos entre varios usuarios.
- Posibilidad de mantener información actualizada y disponible entre varios usuarios.
- Posibilidad de administrar equipos de forma remota.
- Posibilidad de conectarse con otras redes.

Elementos que conforman una red de computadoras

Software

Hardware

Estación de trabajo

Concentrador

Medio Físico

Software

Sistema operativo de red. Es el software diseñado específicamente para administrar los recursos y el flujo de información que circula a través de una red, ya sea que esté configurada con una arquitectura cliente/servidor o de pares. A su vez controla el acceso de usuarios a información restringida entre muchas otras características, dependiendo del sistema utilizado.

Clasificación

Las redes de computadoras pueden clasificarse en varios tipos, dependiendo de su alcance geográfico. Las redes que van desde 0.1 hasta 10 Km son conocidas como redes de área local (local area networks o LAN's, por sus siglas en inglés). Las redes mayores a 10 km son conocidas como redes de área amplia (wide area networks o WAN's, por sus siglas en inglés).

Otra clasificación útil en la que nos podemos basar es en la topología que sigue la red. La topología es la organización de los elementos de la red. Las redes tienen dos topologías diferentes, una es física, pues se refiere a la ubicación física de sus elementos y otra es lógica, debido a que determina la ruta a seguir por los datos a fin de que alcancen su destino.

Tipos de topologías físicas:

Estrella

Anillo

Bus

Malla

Modelo Cliente/Servidor

El modelo Cliente/Servidor funciona bajo un esquema de peticiones. En él, cualquier máquina que realice alguna petición de información o de servicio se le nombra cliente, y a la(s) máquina(s) que

proporcionan la información o el recurso solicitado se les llama servidor(es). En una red de computadoras puede haber más de un servidor y más de un cliente.

En el Front End (interfaz), la parte cliente de la aplicación acepta las peticiones del usuario, las traduce en información apta para el procesamiento del servidor y espera una respuesta del mismo. Por otro lado, en el Back End se recibe la petición dada por el cliente, se procesa y se devuelve el servicio solicitado. El cliente a su vez interpreta la información y la traduce en información entendible para el usuario.

#### Modelo Peer to Peer (o Punto a Punto)

Este modelo también se conoce como descentralizado, debido a que permite a cada computadora funcionar como un servidor parcial, no dedicado de forma exclusiva a los servicios que puede ofrecer.

#### Open System Interconnection

OSI (Open Systems Interconnection, Interconexión de sistemas abiertos). Es un lineamiento funcional para tareas de comunicaciones, el cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Nace como una necesidad de homogenizar los elementos que participan en las redes, es decir, ayuda a los fabricantes a crear redes que sean compatibles con otras redes. Está basada en la clasificación por capas o responsabilidades puntuales que debe cumplir cada elemento dentro del sistema de comunicación.

Las capas se definen a continuación y se encuentran en orden descendente:

#### Capa de aplicación

Se entiende directamente con el usuario final, al proporcionarle el servicio de información distribuida para soportar las aplicaciones y administrar las comunicaciones por parte de la capa de presentación.

#### Capa de presentación

Permite a la capa de aplicación interpretar el significado de la información que se intercambia. Ésta realiza las conversiones de formato mediante las cuales se logra la comunicación de dispositivos.

#### Capa de sesión

Administra el diálogo entre las dos aplicaciones en cooperación mediante el suministro de los servicios que se necesitan para establecer la comunicación, flujo de datos y conclusión de la conexión.

#### Capa de transporte

Esta capa proporciona el control de extremo a extremo y el intercambio de información con el nivel que requiere el usuario. Representa el corazón de la jerarquía de los protocolos que permite realizar el transporte de los datos en forma segura y económica.

#### Capa de red

Proporciona los medios para establecer, mantener y concluir las conexiones conmutadas entre los sistemas del usuario final. Por lo tanto, la capa de red es la más baja, que se ocupa de la transmisión de extremo a extremo.

#### Capa de enlace

Asegura con confiabilidad del medio de transmisión, ya que realiza la verificación de errores, retransmisión, control fuera del flujo y la secuenciación de las capacidades que se utilizan en la capa de red.

#### Capa física

Se encarga de las características eléctricas, mecánicas, funcionales y de procedimiento que se requieren para mover los bits de datos entre cada extremo del enlace de la comunicación.

### **2.3 Lenguajes de programación**

Un programa es una secuencia de instrucciones que permiten a una computadora procesar información conocida como datos de entrada para producir una información de salida o resultados.

Un lenguaje de programación consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen la estructura y el significado de sus elementos y expresiones.

Un lenguaje de programación permite al programador crear diversas aplicaciones ya que puede especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.

#### **Clasificación de los lenguajes de programación**

Los lenguajes de programación se clasifican según el nivel de abstracción, según la forma de ejecución y según el paradigma de programación que poseen cada uno de ellos y esos pueden ser:



## **Según su nivel de abstracción**

### Lenguajes de bajo nivel

Los lenguajes de bajo nivel son lenguajes de programación que se acercan al funcionamiento de una computadora. El lenguaje de más bajo nivel es, por excelencia, el código máquina. A éste le sigue el lenguaje ensamblador, ya que al programar en ensamblador se trabajan con los registros de memoria de la computadora de forma directa.

### Lenguajes de medio nivel

Hay lenguajes de programación que son considerados por algunos expertos como lenguajes de medio nivel (como es el caso del lenguaje C) al tener ciertas características que los acercan a los lenguajes de bajo nivel pero teniendo, al mismo tiempo, ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.

### Lenguajes de alto nivel

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés. Se caracterizan por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas.

## **Según la forma de ejecución**

### Lenguajes compilados

Naturalmente, un programa que se escribe en un lenguaje de alto nivel también tiene que traducirse a un código que pueda utilizar la máquina. Los programas traductores que pueden realizar esta operación se llaman compiladores. Éstos, como los programas ensambladores avanzados, pueden generar muchas líneas de código de máquina por cada proposición del programa fuente. Se requiere una corrida de compilación antes de procesar los datos de un problema.

Los compiladores son aquellos cuya función es traducir un programa escrito en un determinado lenguaje a un idioma que la computadora entienda (lenguaje máquina con código binario).

Al usar un lenguaje compilado (como lo son los lenguajes del popular Visual Studio de Microsoft), el programa desarrollado nunca se ejecuta mientras haya errores, sino hasta que luego de haber compilado el programa, ya no aparecen errores en el código.

## Lenguajes interpretados

Se puede también utilizar una alternativa diferente de los compiladores para traducir lenguajes de alto nivel. En vez de traducir el programa fuente y grabar en forma permanente el código objeto que se produce durante la corrida de compilación para utilizarlo en una corrida de producción futura, el programador sólo carga el programa fuente en la computadora junto con los datos que se van a procesar. A continuación, un programa intérprete, almacenado en el sistema operativo del disco, o incluido de manera permanente dentro de la máquina, convierte cada proposición del programa fuente en lenguaje de máquina conforme vaya siendo necesario durante el proceso de los datos. No se graba el código objeto para utilizarlo posteriormente.

La siguiente vez que se utilice una instrucción, se le debe interpretar otra vez y traducir a lenguaje máquina. Por ejemplo, durante el procesamiento repetitivo de los pasos de un ciclo, cada instrucción del ciclo tendrá que volver a ser interpretado cada vez que se ejecute el ciclo, lo cual hace que el programa sea más lento en tiempo de ejecución (porque se va revisando el código en tiempo de ejecución) pero más rápido en tiempo de diseño (porque no se tiene que estar compilando a cada momento el código completo). El intérprete elimina la necesidad de realizar una corrida de compilación después de cada modificación del programa cuando se quiere agregar funciones o corregir errores; pero es obvio que un programa objeto compilado con antelación deberá ejecutarse con mucha mayor rapidez que uno que se debe interpretar a cada paso durante una corrida de producción.

### **Según el paradigma de programación**

Un paradigma de programación representa un enfoque particular o filosofía para la construcción del software. No es mejor una filosofía otra sino que cada una tiene ventajas y desventajas. También hay situaciones donde un enfoque resulta más apropiado que otro.

Atendiendo al paradigma de programación, se pueden clasificar los lenguajes en:

#### Lenguajes imperativos

Describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican a la computadora cómo realizar una tarea.

Ejemplos:

- BASIC
- C
- C++

- Java
- C#
- Perl

## Lenguajes funcionales

Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones, entendiendo éstas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas, en las que se verifican ciertas propiedades como la transparencia referencial (el significado de una expresión depende únicamente del significado de sus subexpresiones), y por tanto, la carencia total de efectos laterales.

Otras características propias de estos lenguajes son la no existencia de asignaciones de variables y la falta de construcciones estructuradas como la secuencia o la iteración (lo que obliga en la práctica a que todas las repeticiones de instrucciones se lleven a cabo por medio de funciones recursivas).

Existen dos grandes categorías de lenguajes funcionales: los funcionales puros y los híbridos. La diferencia entre ambos estriba en que los lenguajes funcionales híbridos son menos dogmáticos que los puros, al admitir conceptos tomados de los lenguajes imperativos, como las secuencias de instrucciones o la asignación de variables. En contraste, los lenguajes funcionales puros tienen una mayor potencia expresiva, conservando a la vez su transparencia referencial, algo que no se cumple siempre con un lenguaje funcional híbrido.

Ejemplos:

Puros:

- Haskell
- Miranda

Híbridos:

- Lisp
- Scheme

## Lenguajes lógicos

Consiste en la aplicación del conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional.

La programación lógica comprende dos paradigmas de programación: la programación declarativa y la programación funcional. La programación declarativa gira en torno al concepto de predicado, o relación entre elementos. La programación funcional se basa en el concepto de función (que no es más que una evolución de los predicados), de corte más matemático.

Ejemplo:

- Prolog

Lenguajes orientados a objetos

Este paradigma de programación usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento.

Ejemplos:

- Action Script
- C++
- C#
- VB.NET
- Delphi
- Java
- JavaScript
- Perl (soporta herencia múltiple)
- PHP (en su versión 5)

## **2.4 Bases de datos**

Los sistemas manejadores de bases de datos superan las limitaciones de los sistemas orientados a los archivos, como son la redundancia de los datos, el pobre control de la información, capacidades inadecuadas para manejar la información, así como el esfuerzo de programación excesivo. Al tolerar una estructura de datos centralizada, integrada, los sistemas de bases de datos eliminan los problemas de redundancia y de control de los datos.

En la actualidad, estamos inmersos en varias décadas de un largo esfuerzo por desarrollar sistemas de manejo de bases de datos cada vez más poderosos. Este proceso ha sido testigo del desarrollo evolutivo de los sistemas basados en tres modelos de datos fundamentales, y que no

son más que métodos conceptuales para estructurar los datos. Estos tres modelos son el jerárquico, en red y el relacional.

#### Sistemas de bases de datos de modelos jerárquico y en red

Los primeros sistemas de base de datos, introducidos a mediados de la década de los años 60s estaban basados en el modelo jerárquico, que presume que todas las interrelaciones entre los datos pueden estructurarse como jerarquías.

Estas relaciones eran creadas mediante apuntadores físicos (una dirección física que identifica dónde puede encontrarse un registro sobre un disco) o campos de datos añadidos a los registros individuales.

Este modelo tiene limitaciones importantes debido a que no todas las interrelaciones pueden expresarse fácilmente, en el sentido en que a veces en una jerarquía un hijo (registro “subordinado” en una interrelación jerárquica) puede tener más de un padre (registro “propietario”).

A causa de la necesidad obvia de manejar tales interrelaciones, a finales de los 60s se desarrollaron los sistemas de base de datos en red. Al igual que los sistemas de base de datos jerárquicos, los sistemas de base de datos en red emplearon punteros físicos para enlazar entre sí los registros de diferentes archivos.

El sistema de base de datos jerárquico dominante es el IMS de IBM, desarrollado a mediados de los 60's. Entre mediados de los 60's y principios de los 70's se desarrollaron y comercializaron exitosamente varios sistemas de base de datos en red y este modelo de datos se normalizó.

#### Sistemas de bases de datos relacionales.

El uso de punteros físicos era simultáneamente una fortaleza y una debilidad en los sistemas de base de datos jerárquicos y en red. Los punteros eran una fortaleza debido a que permitieron la rápida recuperación de los datos que tuvieran interrelaciones predeterminadas. La debilidad estaba en el hecho de que estas interrelaciones tenían que definirse antes de que el sistema fuera puesto en explotación. Era difícil, si no imposible, recuperar datos basados en otras interrelaciones. En la medida en que los usuarios se familiarizaron con los sistemas de base de datos y en su potencia para manipular esos datos, rápidamente esas limitaciones se volvieron inaceptables.

En 1970, E.F. Codd publicó un artículo revolucionario (Codd, 1970) que desafió fuertemente el juicio convencional de la “condición” de las bases de datos. Codd argumentó que los datos deberían relacionarse entre sí mediante uniones naturales, lógicas, inherentes a los datos, más que mediante punteros físicos. Es decir, las personas deberían ser capaces de combinar los datos

de fuentes diferentes, si la información lógica necesaria para hacer la combinación estaba presente en los datos. Esto abrió una nueva perspectiva para los sistemas de gestión de información ya que las interrogaciones a las bases de datos no necesitarían, en adelante, limitarse a las interrelaciones indicadas por los punteros físicos.

En su artículo, Codd propuso un modelo simple de datos en el que todos ellos se representarían en tablas constituidas por filas y columnas. A estas tablas se les dio el nombre matemático de relaciones, y por eso el modelo se denominó modelo relacional. El modelo de Codd mejoró enormemente la eficiencia conceptual de la programación de las bases de datos.

La publicación del mencionado artículo causó una conmoción en la actividad de las comunidades de desarrollo de los sistemas de investigación y de sistemas comerciales, en la medida que trabajaban para producir un sistema de gestión de base de datos relacional. El resultado fue la aparición de sistemas relacionales durante la mitad de los 70's que soportaban lenguajes como el Structured Query Language (SQL), el Query Language (Quel) y el Query-By-Example (QBE).<sup>2</sup>

#### Diagrama entidad - relación

Los diagramas entidad - relación son un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza.

Los elementos de dicho lenguaje se describen a continuación, por orden de importancia.

#### Entidades

Una entidad es cualquier "objeto" discreto sobre el que se tiene información. Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un nombre. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, empleado, etc.

Cada ejemplar de una entidad se denomina instancia. Por ejemplo, "Carlos y Doris" pueden ser dos instancias distintas de la entidad "persona". Las instancias no se representan en el diagrama. No obstante, se pueden documentar aparte porque son útiles para inicializar la base de datos resultante. Por ejemplo, los departamentos existentes de una empresa pueden ser relevantes como datos iniciales de la entidad "departamento".

---

<sup>2</sup> Hansen, G.W., Hansen, J.V. *Diseño y administración de bases de datos*. Prentice Hall Internacional, España, 1998. Pág. 570.

## Relaciones

Una relación describe cierta interdependencia (de cualquier tipo) entre entidades. Se representa mediante un rombo etiquetado en su interior mediante un verbo. Además, dicho rombo debe unirse mediante líneas con las entidades que relaciona (es decir, los rectángulos).

Una relación no tiene sentido sin las entidades que relaciona. Por ejemplo: una persona (entidad) trabaja (relación) para un departamento (entidad).

## Atributos

Los atributos son propiedades relevantes propias de una entidad y/o relación. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad - relación, sino que se describen textualmente en otros documentos adjuntos.

Los atributos describen información útil sobre las entidades. En particular, los atributos identificativos son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta. Por ejemplo, el atributo identificativo que distingue a un empleado de otro es su RFC.

Ejemplos de atributos de la entidad "persona":

- Documento nacional de identidad (identificativo)
- Nombre
- Apellidos
- Dirección
- Código postal

## Formas Normales

Las formas normales son aplicadas a las tablas de una base de datos, decir que una base de datos está en la forma normal N es decir que todas sus tablas están en la forma normal N.

En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos.

### Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal sólo si

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una clave primaria.
- La tabla no contiene atributos nulos.

### Segunda Forma Normal (2FN)

Dependencia completa. Una relación está en 2FN si por un lado cumple la 1FN y a su vez los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal.

Es decir que no existen dependencias parciales. Los atributos dependen de la clave. Varía la clave y por ende también los atributos.

### Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si cumple la 2FN y cada atributo que no forma parte de ninguna clave depende directamente y no transitivamente, de la clave primaria.

## 2.5 UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje que permite modelar, construir y documentar los elementos que forman parte de un sistema. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a desarrollar. UML unifica la experiencia obtenida con anterioridad sobre técnicas de modelado e incorpora las mejores prácticas actuales en un lenguaje estándar.

Este lenguaje ha sido ampliamente aceptado debido al prestigio de sus creadores Grady Booch, Ivar Jacobson y Jim Rumbaugh y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE).

UML es el estándar de la industria, no sólo de la del software sino, en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y construcción de prototipos.

UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.



## Diagrama de casos de uso

Un diagrama de casos de uso muestra la relación que existe entre el sistema y los usuarios externos a éste, llamados actores. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. En la figura 2.5.1 se muestra un ejemplo de diagrama de casos de uso para un cajero automático.

Los elementos que pueden aparecer en un diagrama de casos de uso son: actores y casos de uso.

### Actores

Un actor es la idealización de una persona externa, de un proceso o de una cosa que interactúa con un sistema, un subsistema o una clase. Se representa mediante una figura humana dibujada con trazos lineales y el nombre debajo de él. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.

### Casos de uso

Un caso de uso es la descripción de la secuencia de interacciones que se producen entre un actor y el sistema. Esta secuencia se representa en el diagrama de casos de uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

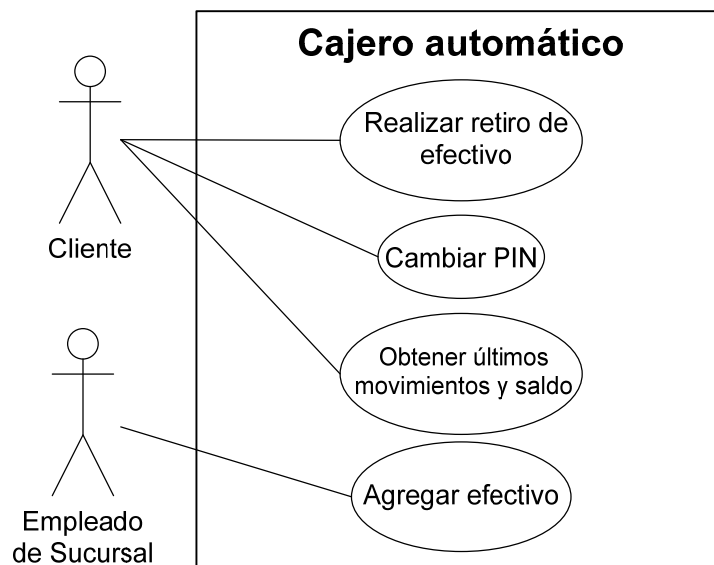


Figura 2.5.1 Diagrama de casos de uso para un cajero automático

## Diagramas de clases

Es un tipo de diagrama que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

**Propiedades.** También llamados atributos o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: marca, tamaño, color y peso.

**Operaciones.** Son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.

**Interfaz.** Es un conjunto de operaciones y/o propiedades que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto.

**Herencia.** Se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre.

Al diseñar una clase se debe pensar en cómo se puede identificar un objeto real (una persona, un transporte, un documento o un paquete). Durante el proceso del diseño de las clases se toman las propiedades que identifican de forma única al objeto y otras propiedades adicionales (datos que corresponden al objeto).

## Diagramas de secuencia

Un diagrama de secuencia muestra un conjunto de mensajes que siguen una secuencia temporal. Muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de

tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren. Esto se observa en la figura 2.5.2.

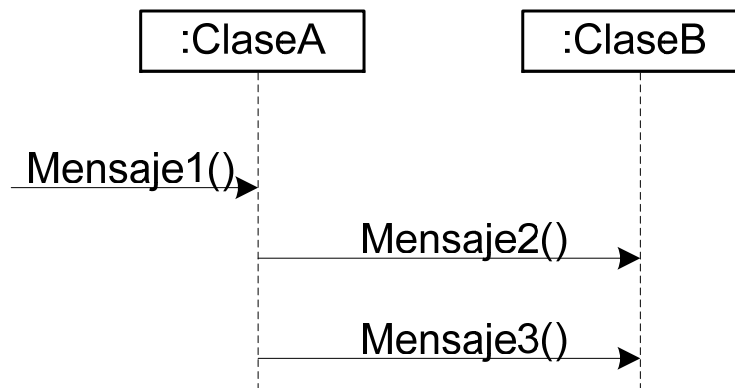


Figura 2.5.2 Diagrama de secuencia

### Diagramas de colaboración

Un diagrama de colaboración muestra un esquema organizado basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia. Véase la figura 2.5.3.

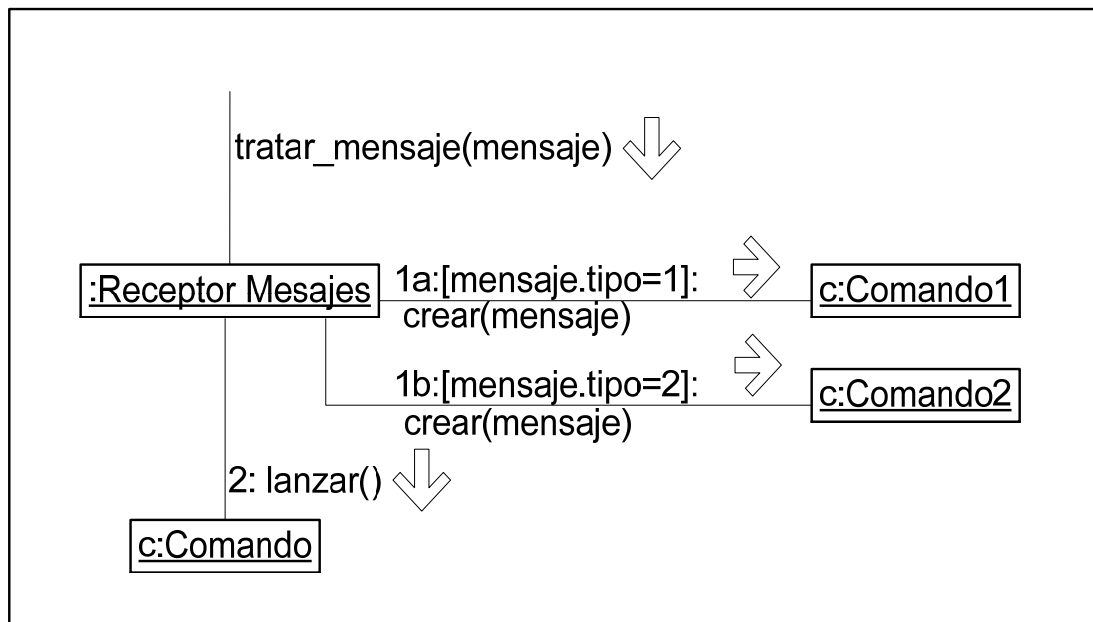


Figura 2.5.3 Diagrama de colaboración

Un diagrama de colaboración muestra una serie de objetos con enlaces entre los mismos objetos y con los mensajes que intercambian dichos objetos.

## **Diagrama de estados**

Un diagrama de estados muestra la secuencia de estados por los que pasa un caso de uso, un objeto a lo largo de su vida o bien el sistema en su totalidad. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

Un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino.

La caja de un estado puede tener 1 ó 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma entrada -> "acción asociada" donde "acción asociada" es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se ingresa al estado por medio de una transición, la acción de entrada se ejecuta.

Una acción de salida aparece en la forma salida -> "acción asociada". Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.

Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma "nombre del evento" -> "acción\_asociada".

Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (finalización del caso de uso o destrucción del objeto).

El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo (véase la figura 2.5.4). En realidad, los estados inicial y final son pseudoestados, pues un objeto no puede "estar" en esos estados, pero se indican en el diagrama, pues nos sirven para saber cuáles son las transiciones inicial y final(es). Lo anterior se muestra en la figura 2.5.4.

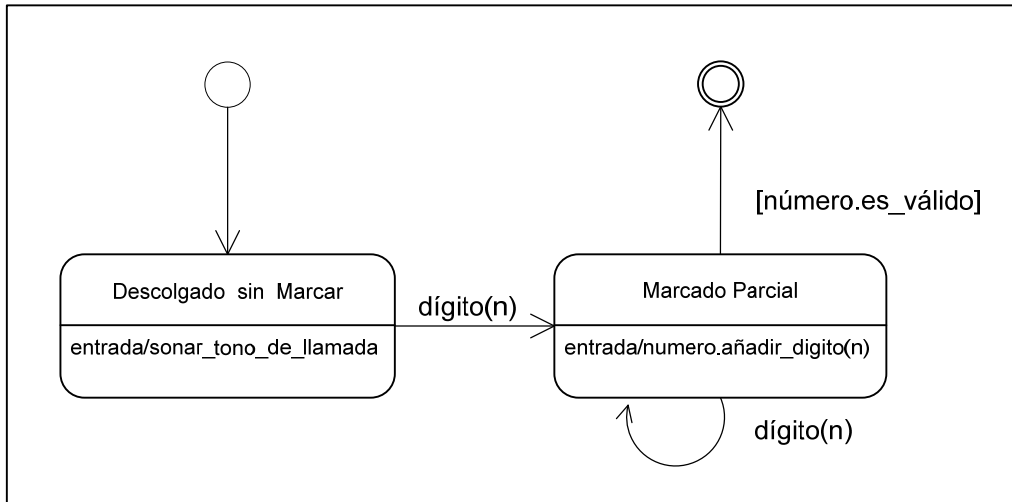


Figura 2.5.4 Diagrama de estados

### Diagrama de actividades

Un diagrama de actividades muestra el flujo de acciones que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. Gráficamente se representa con un conjunto de arcos y nodos. El diagrama de actividades muestra cómo fluye el control de unas clases a otras colaborando entre ellas para conseguir un mismo fin.

Estados de actividad y estados de acción

La representación de ambos es un rectángulo con las puntas redondeadas, en su interior se puede representar una actividad o una acción. Un estado que represente una acción se considera atómico, con ejecución instantánea y no puede ser interrumpida.

En la figura 2.5.5, podemos ver ejemplos de estados de acción.

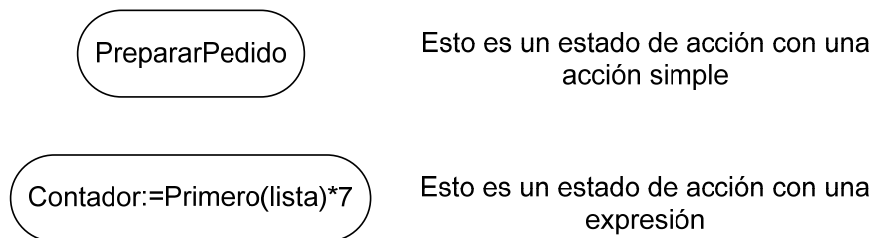


Figura 2.5.5 Estados de acción

A diferencia de los estados de acción, un estado de actividad sí puede descomponerse en más sub-actividades, pueden ser interrumpidos y tardan menor tiempo en completarse.

Un ejemplo de este tipo de estado se muestra en la figura 2.5.6.

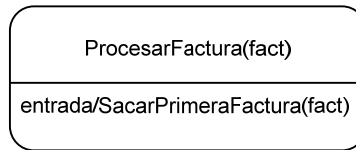


Figura 2.5.6 Estado de actividad

### Transiciones

Las transiciones reflejan el paso de un estado a otro. Para indicar el flujo de control podemos utilizar dos “disparadores” (estados que desencadenan una acción) de inicio y fin tal y como queda reflejado en el ejemplo de la figura 2.5.7.

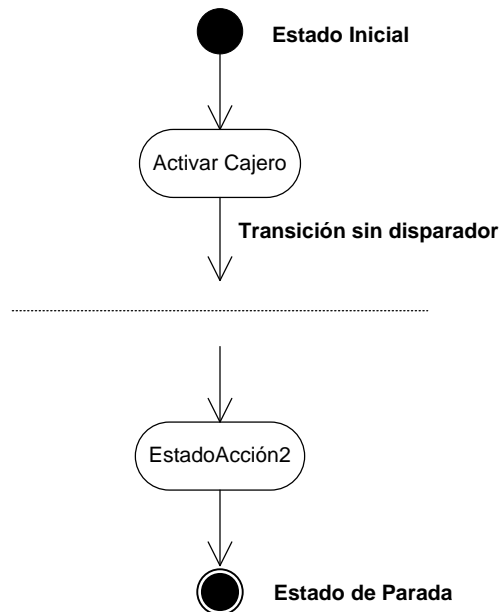


Figura 2.5.7 Transiciones utilizando disparadores

### División y unión

La división y la unión se representan por líneas horizontales anchas. Nos sirven para dividir y organizar el esquema cuando se presentan actividades con concurrencia. En la figura 2.5.8 podemos ver su representación gráfica.

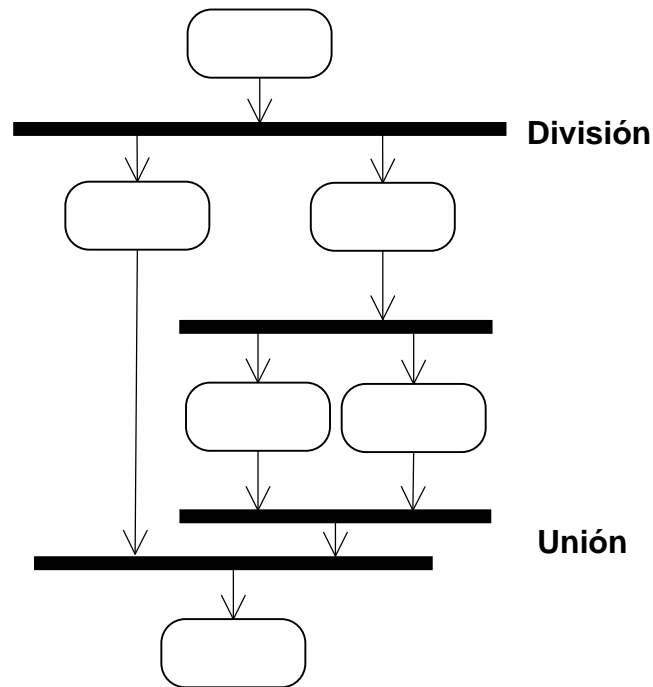


Figura 2.5.8 División y unión

### Diagrama de componentes

Un diagrama de componentes muestra un conjunto de componentes y sus relaciones de manera gráfica a través del uso de nodos y arcos, muestra la organización y las dependencias entre un conjunto de componentes.

Un componente se dibuja como un rectángulo, con dos rectángulos pequeños a un lado, se unen por medio de líneas sólidas a los círculos que representan sus interfaces.

Un nodo es un elemento donde se ejecutan los componentes, son los elementos que participan en la ejecución de un sistema, tienen memoria, capacidad de proceso y se pueden agrupar en paquetes igual que las clases y los componentes.

Un nodo se representa mediante un cubo con su respectivo nombre y, de manera opcional, su clasificación.

La relación entre un nodo y los componentes que despliega se puede representar mediante una relación de dependencia como se indica en la figura 2.5.9.

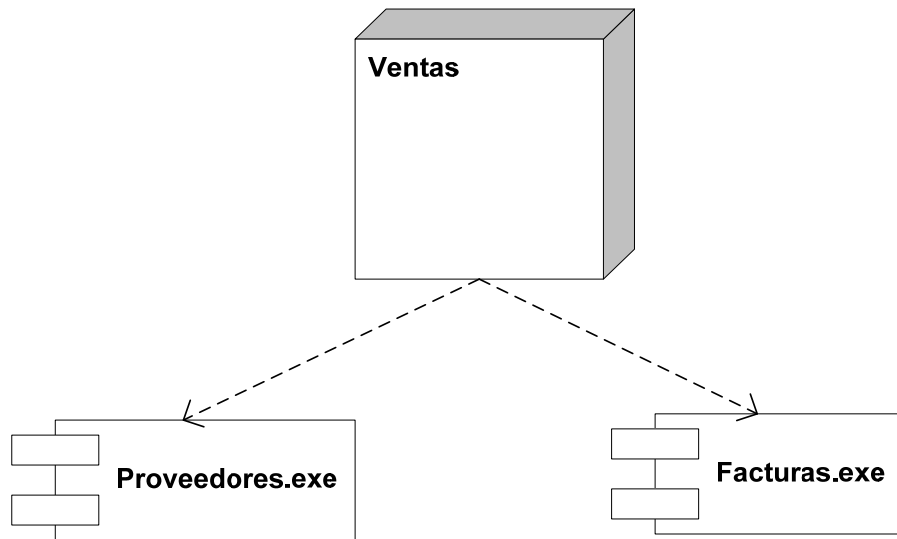


Figura 2.5.9 Relación entre nodos y componentes

El tipo de relación más común entre nodos es la asociación. Una asociación entre nodos viene a representar una conexión física entre nodos como se puede ver en la figura 2.5.10.

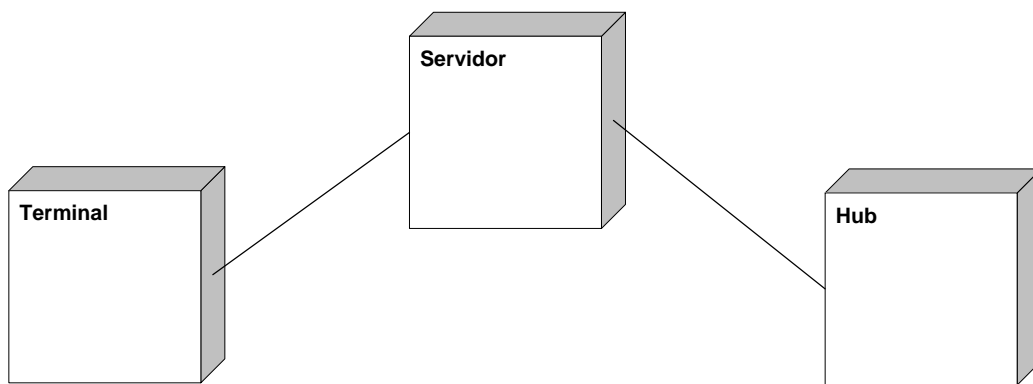


Figura 2.5.10 Conexiones entre nodos

## 2.6 Gestión de riesgos

La gestión de riesgos se convierte en una parte muy importante en el desarrollo de un sistema de software, debido a que la elaboración correcta de este análisis puede absorber una cantidad significativa de esfuerzo en la planificación y desarrollo del proyecto.

El riesgo de software se define a su vez como el factor que impide (o al menos amenaza) llevar a cabo el sistema en la forma programada. De esta manera, podemos decir que el riesgo siempre involucra dos características fundamentales:



1. **Incertidumbre.** El acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100% de probabilidad (ya que un riesgo del 100% es una limitación del proyecto).
2. **Pérdida.** Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

### **Riesgos del proyecto**

Los riesgos del proyecto amenazan al plan del proyecto, es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican:

- Problemas potenciales de presupuesto
- Planificación temporal
- Personas (asignación y organización)
- Recursos
- Cliente, requerimientos y su impacto en el proyecto software
- La complejidad del proyecto
- Tamaño y grado de incertidumbre estructural

### **Riesgos técnicos**

Amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo se convierte en realidad, la implementación puede llegar a ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de:

- Diseño
- Implementación
- De interfaz
- Verificación
- Mantenimiento
- Las ambigüedades de especificaciones
- Tecnologías

### **Identificación y clasificación del riesgo**

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan de proyecto (estimaciones, planificación temporal, cargo de recursos, etc.). Identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario.

Existen dos tipos diferenciados de riesgos para cada categoría:

1. **Riesgos genéricos.** Son una amenaza potencial para todos los proyectos de software.
2. **Riesgos específicos.** Sólo los pueden identificar quienes tienen una clara visión de la tecnología, el personal y el entorno específico del proyecto en cuestión, entre los que se encuentran:
  - **Tamaño del producto.** Riesgos asociados con el tamaño general del software a construir o a modificar
  - **Impacto en el negocio.** Riesgos asociados con las limitaciones impuestas por la gestión o por el mercado
  - **Características del cliente.** Riesgos asociados con la sofisticación del cliente y la habilidad del desarrollador para comunicarse con el cliente en los momentos oportunos
  - **Definición del proceso.** Riesgos asociados con el grado de definición del proceso del software y su seguimiento para la organización de desarrollo
  - **Entorno de desarrollo.** Riesgos asociados con la disponibilidad y calidad de las herramientas que se van a emplear en la construcción del producto
  - **Tecnología a construir.** Riesgos asociados con la complejidad del sistema a construir y la tecnología de punta que contiene el sistema
  - **Tamaño y experiencia de la plantilla.** Riesgos asociados con la experiencia técnica y de proyectos de los ingenieros de software que van a realizar el trabajo

### Componentes y controladores del riesgo

En el contexto de este estudio, los componentes del riesgo se definen de la siguiente manera:

- **Riesgo de rendimiento.** Grado de incertidumbre con el que el producto encontrará sus requerimientos y se adecue para su empleo pretendido
- **Riesgo de costo.** Grado de incertidumbre que mantendrá el presupuesto del proyecto.
- **Riesgo de soporte.** Grado de incertidumbre de la facilidad del software para corregirse, adaptarse y ser mejorado
- **Riesgo de la planificación temporal.** Grado de incertidumbre con que se podrá mantener la planificación temporal y de que el producto se entregue a tiempo

## 2.7 Mantenimiento del software

El mantenimiento de software es el proceso que se sigue para mejorar y optimizar el producto de software después de la entrega, aunque también se utiliza para remediar los defectos, con el fin de asegurar el continuo desarrollo de las funciones programadas para dicho producto.

La fase de mantenimiento de software involucra cambios en el sistema en orden de corregir defectos y dependencias encontradas durante el uso del mismo, por ejemplo en la adición de alguna nueva funcionalidad para mejorar la usabilidad o en la aplicabilidad del software.

Los tipos de mantenimiento que se pueden realizar son:

- Perfectivo
- Adaptativo
- Correctivo
- Preventivo

Una petición de modificación es un término usado para identificar el cambio en un producto de software que está en mantenimiento.

### **Mantenimiento perfectivo**

Se refiere a la modificación del producto de software después de su entrega, para mejorar su desempeño o facilidad de mantenimiento. El mantenimiento perfectivo proporciona mejoras para los usuarios, mejora de la documentación y la recodificación del sistema, a fin de mejorar el rendimiento del software.

Para realizar el mantenimiento perfectivo, se debe presentar alguna de las dos situaciones siguientes:

- A) El usuario hace una petición para incorporar funcionalidades adicionales al sistema o sugiere alguna mejora
- B) Se realiza un trabajo de investigación a los usuarios y se les pregunta su opinión acerca de la manera en que obra el sistema y si satisface sus necesidades, a su vez, se investiga la manera en que trabajan otros sistemas similares y se compara con nuestro propio sistema

De acuerdo a los resultados obtenidos, se realiza el plan de mantenimiento para perfeccionar el sistema y satisfacer las necesidades de los usuarios.

### **Mantenimiento adaptativo**

Es la modificación del producto de software una vez que ha sido entregado, a fin de lograr que sea utilizable en un nuevo entorno tecnológico (nuevo hardware, otro sistema gestor de bases de datos, diferente sistema operativo, etc.).

Estos cambios pueden ser:

1. En el entorno de datos (por ejemplo, el cambio del sistema de archivos por un SMDB)
2. En el entorno de procesos (realizar una versión del sistema para otro navegador como Firefox, Opera o Netscape)

### **Mantenimiento correctivo**

Se refiere a la corrección de defectos y fallas detectadas en la operación del producto de software después de su entrega. Un defecto en un sistema es una característica del sistema con el potencial de producir una falla.

Una falla se produce cuando el comportamiento del sistema difiere con respecto al comportamiento definido en la especificación. Las fallas en un sistema de software pueden ser de:

1. Procesamiento (salidas incorrectas de un programa)
2. Rendimiento (tiempo de respuesta demasiado alto)
3. Programación (inconsistencias en el diseño)
4. Documentación (inconsistencias entre la funcionalidad de un programa y un manual de usuario)

### **Mantenimiento preventivo**

Consiste en la revisión periódica del producto de software una vez entregado, a fin de detectar y corregir fallos latentes, antes de que éstos se conviertan en fallos efectivos. Este tipo de mantenimiento ayuda a reducir los tiempos que pueden generarse por mantenimiento correctivo.

Los conceptos explicados anteriormente fueron estudiados a lo largo de la carrera y del tiempo de desarrollo de este trabajo y sirven como base en el proceso de planteamiento de la solución propuesta, lo cual se presenta en el siguiente capítulo, donde se aborda el análisis y diseño del SABB de una forma práctica.

# Capítulo 3

## Análisis y diseño



### **3.1 Análisis de factibilidad**

Antes de comenzar a realizar el proyecto de software es necesario precisar un plan que defina los procesos y tareas específicas a realizar por parte de los encargados del desarrollo trabajo, así como los mecanismos para evaluar los riesgos que pueden hacer que el proyecto no se complete, y por otro lado valorar los componentes que aseguren la calidad del sistema en general.

#### **Estudios de factibilidad**

Siguiendo con el proceso de gestión, se han elaborado cinco estudios de factibilidad, los cuales nos ayudarán a determinar las posibilidades reales de alcanzar el éxito del sistema en su conjunto. Los estudios realizados se desarrollan a continuación:

#### **Factibilidad técnica**

Este estudio se conforma de la revisión de las herramientas de hardware con las que dispone tanto el Departamento de Cómputo de la FP (DCFP), así como del DSSyBT, a fin de que el sistema opere de forma óptima y no se tengan problemas técnicos.

De acuerdo con ello, dentro del DSSyBT no se cuenta con una infraestructura de red adecuada, pues de las 15 personas que allí laboran, solo 3 trabajadores tienen una conexión a la red que les permite acceder a internet. Cabe mencionar que el DCFP, al momento de realizar este documento, está llevando a cabo las gestiones necesarias para complementar el sistema de red, a fin de que todas las personas que laboran en el DSSyBT posean una conexión en su lugar de trabajo.

Por otro lado, dentro del DCFP se cuenta con varios equipos servidores, los cuales pueden alojar al sistema SABB prácticamente sin problema.

Ahora bien, en la realización del módulo correspondiente a la BT, se requiere la construcción de una interfaz que recabe la información proporcionada por cada usuario, por lo que se procederá a programarla en el lenguaje de programación PHP y se utilizará el manejador de bases de datos MySQL, los cuales se encuentran disponibles en internet.

#### **Evaluación tecnológica**

Para la realización de este proyecto, necesitamos contar con herramientas de programación que faciliten el trabajo de los desarrolladores, aunque a su vez, deben ser lo suficientemente robustas y completas para soportar todas las aplicaciones y peticiones que ha hecho el cliente, y deben garantizar la mejor razón de precio contra rendimiento.

Debido a la gran diversidad de lenguajes de programación y de sistemas manejadores de bases de datos existentes en el mercado, se debe realizar una cuidadosa selección de la herramienta adecuada, a fin de que el desarrollo tenga la mayor calidad posible.

#### Sistema operativo

Actualmente en el mercado, existen diversos sistemas operativos disponibles, los cuales varían en precio, siendo Microsoft Windows el más popular, aunque se debe pagar por la licencia o el derecho a su uso.

Por otro lado, existen también los sistemas operativos basados en GNU/Linux como Red Hat, el cual se utilizará como sistema operativo dentro del proyecto, debido a que el DCFP trabaja ya con ese software y se precisó su interés en no modificarlo por cuestiones de organización interna. La versión de este sistema operativo es Red Hat 9.

#### Sistemas manejadores de bases de datos

Los sistemas manejadores de bases de datos (SMBD) se encuentran disponibles en el mercado en varios costos y configuraciones dependiendo el número de usuarios que se conectarán y del almacenamiento de espacio que pueden utilizar.

Algunos SMBD son:

##### Oracle 10g Database Express Edition (XE)

Esta edición cuenta con una compatibilidad total con Linux, permitiendo que la administración sea eficiente y confiable en equipos con éste sistema operativo instalado.

Una de las principales características de este SMBD es su escalabilidad, pues se puede, por ejemplo, adquirir una nueva versión de Oracle con características básicas y si las necesidades del sistema lo requieren, se pueden hacer crecer los recursos disponibles adquiriendo nuevos módulos, lo cual contribuye favorablemente en su mantenimiento, crecimiento y actualización.

##### Microsoft SQL Server 2005

Este SMBD es un excelente recurso en el almacenamiento, procesamiento y aseguramiento de la información, debido a su acceso controlado a la BD y procesamiento rápido de las transacciones, además de ser compatible con diversas herramientas muy arraigadas en el ámbito laboral, como Microsoft Office. El soporte dado por parte del fabricante es amplio y se encuentra bien definido, lo cual redundo en una mayor productividad. El inconveniente al utilizar esta herramienta es el costo



de utilización, aunado a la necesidad de instalarlo en un sistema operativo del mismo fabricante, como lo es Microsoft Windows Server 2003 o Microsoft Windows NT, el cual a su vez también debe ser adquirido.

## MySQL

A fin de obtener una completa compatibilidad en cuanto al manejo de la información requerida por los módulos que implementaremos, se utilizará el SDBD MySQL.

La utilización de esta herramienta se encuentra muy difundida en todo el mundo, debido a su gran rapidez al realizar diversas consultas, su gran versatilidad de comandos, integración a ambientes como Linux o Windows, adaptabilidad al trabajo con el lenguaje de programación PHP y su amplia documentación disponible en la página de internet del fabricante.

Cabe mencionar que el uso del software no tiene ningún costo, por lo que es muy apropiado para desarrollar el proyecto actual dadas las limitantes mencionadas anteriormente.

## Servidores web

Como muchas otras tecnologías, los servidores web existen en el mercado y tienen diversas características y precios. Antes del desarrollo del proyecto se debe identificar el servidor idóneo para alojar el sistema ya que de él dependerá en gran parte la fiabilidad y seguridad de la información exhibida en línea, como se ha mencionado en el capítulo anterior.

## Servidor HTTP Apache

*El servidor HTTP Apache es un software de distribución libre y gratuita, disponible para aplicaciones Unix, Windows, Macintosh y otras, que implementa la noción de sitio virtual. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de National Center for Supercomputing (NCSA, por sus siglas en inglés). Era pues, en inglés, 'a patchy server' (un servidor "parcheado").*

*Apache presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de su contenido, pero es criticado por su falta de interfaz grafica que ayude en su configuración.<sup>3</sup>*

---

<sup>3</sup> [http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

## Internet Information Server (IIS)

Internet Information Server es una serie de servicios para las computadoras que funcionan bajo el sistema operativo Windows. Originalmente IIS era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP y HTTP/HTTPS.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo los de Active Server Pages (ASP) y ASP.NET, ambos desarrollados por Microsoft.

Versiones anteriores de IIS 5.1 han sido descontinuadas debido a la vulnerabilidad a virus por lo que actualmente se utiliza la versión 6.0.<sup>4</sup>

## Servidor HTTP Cherokee

Cherokee es un servidor de uso gratuito y puede funcionar sobre diversos sistemas operativos, como Unix y también Windows, aunque su instalación es más difícil debido a que la configuración de puertos no está completamente perfeccionada.

La finalidad del programa es facilitar la administración de servicios ofrecidos a través de una interfaz para el usuario. También intenta que la instalación no ocupe tanto espacio en disco como lo hacen Apache o IIS. Un último punto es que trata de ser un servidor novedoso, pues los programas anteriormente descritos tienen más de diez años de desarrollo.

Debido al resultado arrojado por estas comparativas, hemos decidido que la mejor opción para utilizar es el servidor http Apache Versión 2.0 debido a que es muy confiable y gratuito, además de contar con un gran respaldo de personas a nivel mundial, interesadas en solventar cualquier problema de seguridad que surja en un futuro.

## Lenguajes de programación

A raíz de la limitante que implica la noción de un lenguaje de programación por parte del personal del DCFP, se hace necesaria la adopción del lenguaje PHP para desarrollar nuestro proyecto y, por lo tanto, en caso de una posterior etapa de mantenimiento del código, ésta se pueda realizar sin la necesidad de contratar personal externo o demorar en la comprensión del código por parte del personal existente.

---

<sup>4</sup> <http://www.mastermagazine.info/termino/5454.php>

### **Factibilidad económica**

Este estudio se conforma de la revisión de dos aspectos económicos que actualmente influyen en la realización del presente trabajo de tesis, los cuales son el aspecto humano y el aspecto tecnológico.

Dentro del aspecto humano (lo cual concierne al pago por la realización del SBT), no se obtiene ninguna remuneración económica ni de algún otro tipo y en cambio, se lleva a cabo como una forma de retribución a la Universidad Nacional Autónoma de México.

En cuanto al aspecto tecnológico se tiene que ejercer un gasto presupuestario dentro de la FP a fin de instalar la infraestructura de red necesaria para que el sistema pueda funcionar de manera adecuada, pero la asignación del mismo es responsabilidad del DCFP.

A su vez, las herramientas que se utilizarán en el desarrollo del sistema (los lenguajes de programación, el manejador de bases de datos y el sistema operativo), se distribuyen de forma gratuita en internet, lo cual tampoco implica un gasto para ninguna de las partes involucradas.

### **Factibilidad operacional**

Para la factibilidad operacional, se cuenta con un apoyo de todo el personal que labora dentro del DSSyBT, así como del DCFP, por lo que el proyecto no tiene problemas en este sentido para llevarse a cabo.

Se espera que el personal del DSSyBT que utilizará el sistema, obtenga una capacitación en el uso de la herramienta de 2 semanas, y que lo utilice de forma correcta, completa y autónoma en un lapso de 3 semanas.

Los usuarios universitarios, esto es egresados y alumnos de la FP, así como las empresas participantes, no tendrán problemas reales en la utilización del sistema, pues existirá un instructivo de uso en línea donde resolverán todas sus dudas.

### **Factibilidad legal**

La elaboración y uso del sistema no incumple ninguna ley vigente dentro del Reglamento Interno de BT de la Facultad de Psicología, por lo que su implementación y desarrollo es perfectamente válido y legal.

Referente a la pertenencia, se ceden todos los derechos del sistema en favor de la Universidad Nacional Autónoma de México, siempre y cuando solamente sea utilizado para el propósito específico para el cual ha sido diseñado y sólo se use dentro del Departamento de Servicio Social y Bolsa de Trabajo de la Facultad de Psicología.

### Factibilidad de programa

En la realización del sistema, se analizaron varios aspectos en cuanto al riesgo que presenta el realizar el sistema en una fecha de tiempo determinada. Para ello se elaboró el diagrama de Gantt mostrado en la figura 3.1.1, el cual indica las actividades prioritarias y el lapso de tiempo en el que se tiene previsto iniciarlas y finalizarlas:

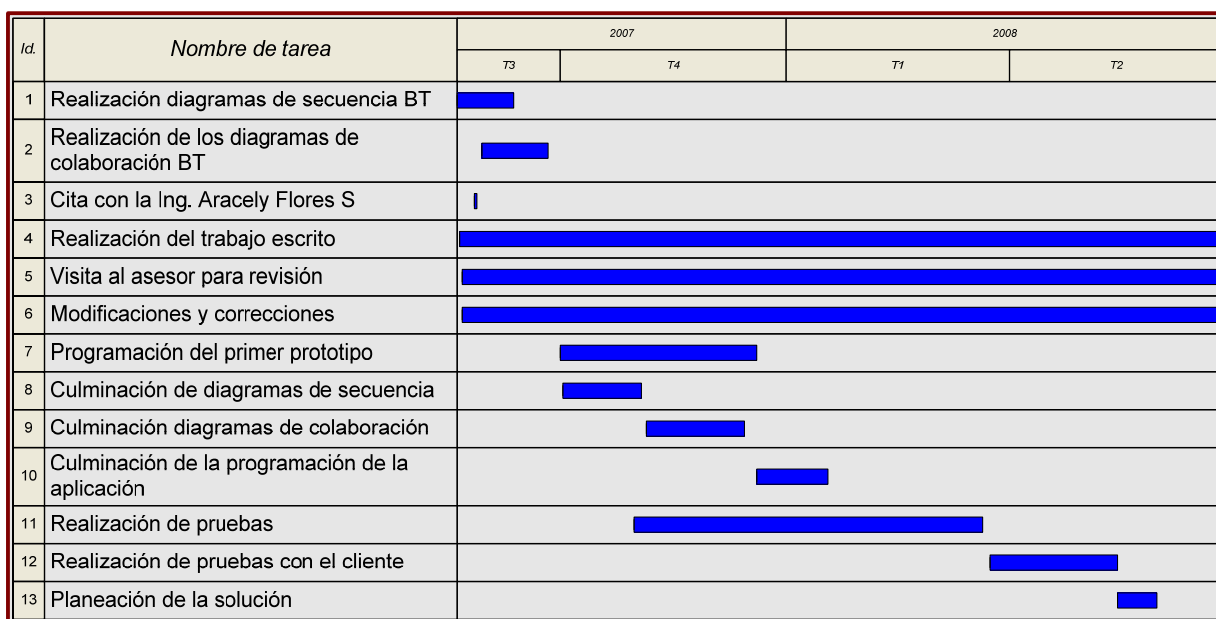


Figura 3.1.1 Diagrama de Gantt para el SABT

En el diagrama de Gantt se definen sobre el eje vertical cada una de las actividades que se realizarán y sobre el eje horizontal el periodo de duración del proyecto, por lo tanto, cada tarea tendrá un lapso bien definido de duración.

### 3.2 Diagramas de casos de uso

A fin de conocer el alcance real del proyecto, se realizaron 25 diagramas de casos de uso, los cuales como se ha explicado anteriormente, ayudan al desarrollador a entender de mejor manera el problema, así como un primer alcance de la solución.

Para poder realizarlos, se elaboró una investigación de los trámites que se siguen actualmente, dando como resultado un proceso para cada actividad.

#### Casos de uso de alto nivel

**Caso:** Validar una Empresa que desee solicitar personal.

**Actores:** Empresa que desea contratar personal.

**Tipo:** Primario.

**Descripción:** El responsable del área de recursos humanos de la Empresa ingresa al sistema e introduce los datos requeridos. El sistema valida el nombre de usuario y contraseña.

Como se puede notar, se indican claramente el nombre del evento, los actores que participan en la acción, así como el tipo de evento. Se menciona que es primario, tomando en cuenta su importancia para el sistema y a su vez se desarrolla una breve descripción.

#### Caso de uso expandido

**Caso:** Validar una Empresa que desee solicitar personal.

**Actores:** Empresa que desea contratar personal.

**Propósito:** Contar con una base de datos con información de empresas que desean contratar personal egresado de la FP.

**Resumen:** El nombre de usuario y la contraseña de la Empresa que desea contratar personal son validados por el sistema.

**Tipo:** Primario y esencial.

## Curso normal de los eventos

Esencial

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el encargado por parte de una Empresa ingresa al sistema para registrar sus datos.	2. El sistema requiere sea proporcionado un nombre de usuario y una contraseña.
3. El Actor introduce los datos solicitados.	4. El sistema determina que los datos proporcionados son correctos.
	5. El sistema muestra la página de opciones de la Empresa.

Tabla 3.2.1 Diagrama de caso de uso básico para validar una Empresa que desee solicitar personal

Curso alternativo 1

Acción de los actores	Respuesta del sistema
	4. El sistema detecta que uno o más datos proporcionados no cumplen la(s) condición(es).
	5. El sistema indica cuál(es) condición(es) no se cumple(n) e indica que no puede continuar.

Tabla 3.2.2 Curso alternativo para el diagrama de caso de uso inicial

En los párrafos anteriores se detalló el caso de uso explicado inicialmente. Como se puede observar, se agregó una parte muy importante, que son los cursos normales y alternos de los eventos. En ellos se detalla paso por paso la secuencia que debe seguir el actor para realizar ese caso de uso, así como las variantes que pueden ocurrir en el proceso. Esto es sumamente útil debido a que nos da una idea aun más clara de la secuencia de las acciones.

**Caso:** Inscribir a una nueva Secretaria.

**Actor:** Administrador General del sistema (Administrador).

**Tipo:** Primario.

**Descripción:** El Administrador ingresa al sistema y asigna una cuenta a la Secretaria para que pueda hacer uso del sistema.

### Caso de uso expandido

**Caso:** Inscribir a una nueva Secretaria.

**Actores:** Administrador General del sistema (Administrador).

**Propósito:** Tener un control del personal del departamento que hace uso del sistema.

**Resumen:** El Administrador es la única persona con facultades de asignar cuentas de usuario.

**Tipo:** Primario y esencial

### Curso normal de los eventos

Esencial

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando el Administrador desea asignar una cuenta de Secretaria.	
2. El Administrador ingresa al sistema.	3. El sistema valida el nombre de usuario y la contraseña y determina que estos datos son correctos.
4. El Administrador selecciona la opción <i>Inscribir a una nueva secretaria.</i>	5. El sistema muestra el formulario para inscribir a una nueva Secretaria.
6. El Administrador llena el formulario.	7. El sistema valida el llenado del formulario y determina que es correcto.
	8. El sistema almacena la información proporcionada y le indica al usuario que sus datos fueron almacenados correctamente.
9. El Administrador indica que desea salir.	10. El sistema muestra las opciones del Administrador.

Tabla 3.2.3 Diagrama de caso de uso extendido para Inscribir a una nueva Secretaria.

#### Curso alternativo 1

Acción de los actores	Respuesta del sistema
	3. El sistema determina que los datos no son correctos.
	4. El sistema determina que no se puede continuar y que se ingresen los datos de nuevo.

Tabla 3.2.4 Curso alternativo para el diagrama de caso de uso Inscribir a una nueva Secretaria

#### Curso alternativo 2

Acción de los actores	Respuesta del sistema
	7. El sistema determina que el formulario no fue llenado correctamente.
	8. El sistema indica los campos que fueron llenados incorrectamente, vuelve a solicitarlos y pasa al punto 5.

Tabla 3.2.5 Segundo curso alternativo para el diagrama de caso de uso Inscribir a una nueva Secretaria

### 3.3 Diagramas de clases

Después de que los casos de uso han sido identificados y clasificados, se procede a desarrollar un diagrama de clases o modelo conceptual basados en objetos, el cual servirá para entender los conceptos significativos del problema.

El paso esencial de un análisis orientado a objetos es descomponer el problema en conceptos u objetos individuales.

El concepto es una idea, cosa u objeto y podemos definirlo a partir de su símbolo, intención y extensión.

Símbolo. Palabras o imágenes que representan un concepto.

Intención. La definición del concepto.

Extensión. El conjunto de ejemplos a que se aplica el concepto.



La figura 3.3.1 muestra el concepto Empresa. Lo designamos con el símbolo *Empresa*. La intención de una *Empresa* cuenta con un Nombre, Giro y Dirección. La extensión de *Empresa* son el conjunto de todas las empresas.

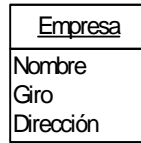


Figura 3.3.1. Diagrama de clase para empresa (no exhaustivo)

El siguiente paso para crear un modelo conceptual consiste en incorporar asociaciones para registrar las relaciones entre conceptos.

La asociación es una relación entre dos conceptos que indica alguna conexión significativa entre ellos.

Como podemos observar en la figura 3.3.2, una vez que hemos incorporado las asociaciones, agregamos los atributos necesarios para cada uno de nuestros conceptos.

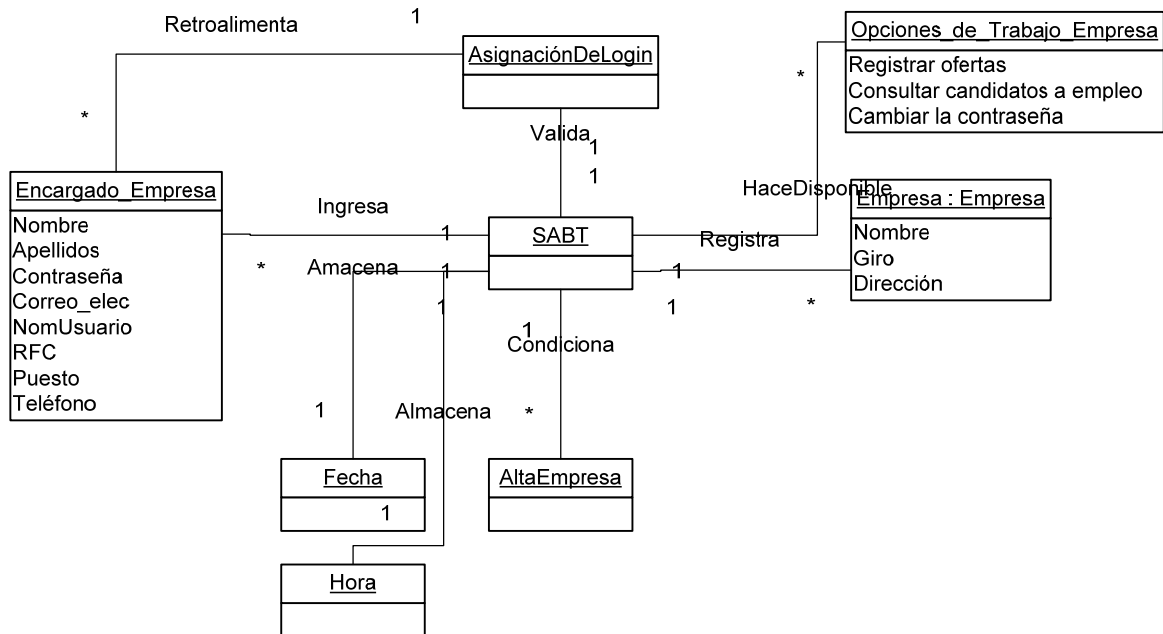


Figura 3.3.2. Diagrama de clases para “Validar una Empresa que desee solicitar personal”

### 3.4 Diagramas de secuencia

Como siguiente paso debemos analizar el comportamiento del sistema, es decir, tener una descripción de lo que hace, sin explicar la manera de cómo lo hace. Para ello nos basaremos en el diagrama de secuencia.

La figura 3.4.1 muestra un diagrama de secuencia típico con mensajes.

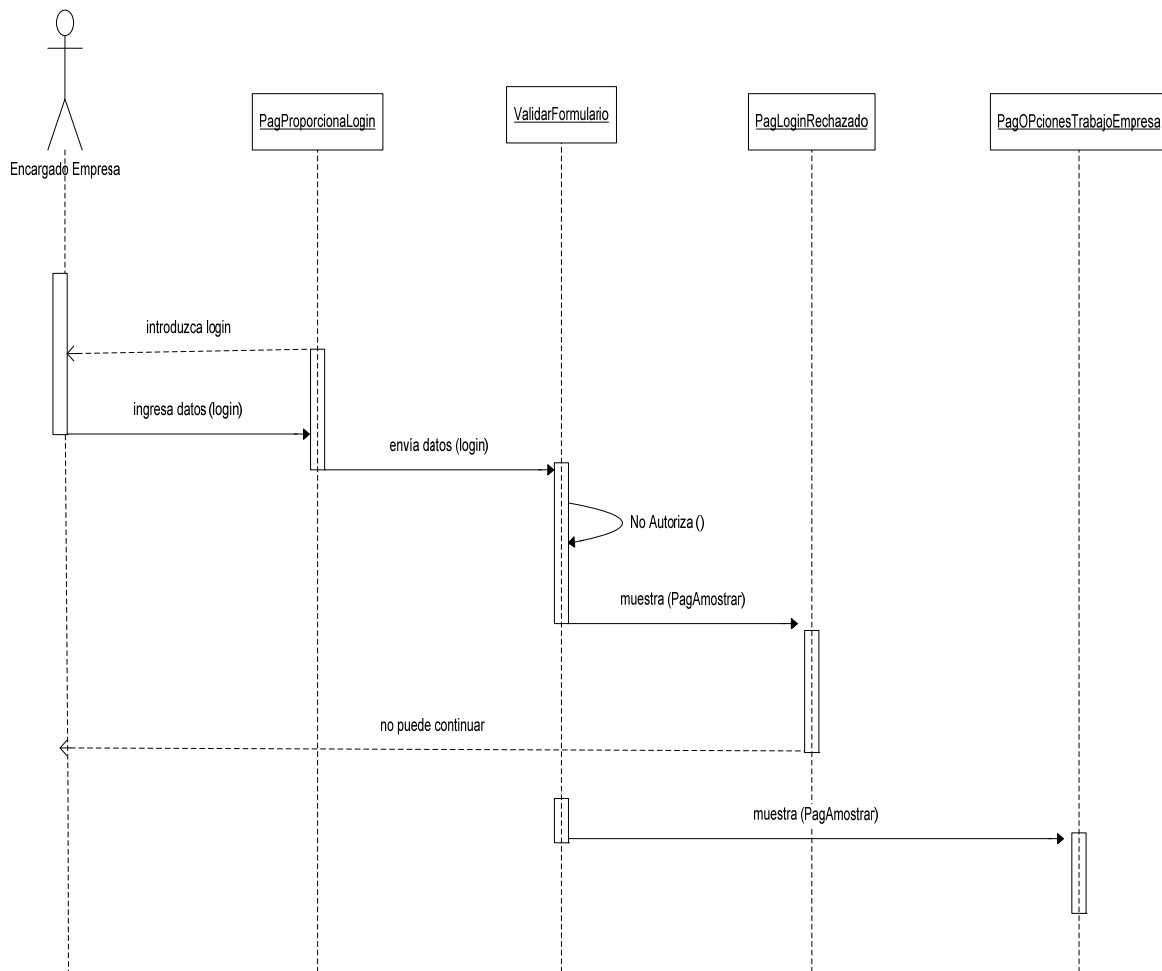


Figura 3.4.1 Diagrama de secuencia para “Validar una Empresa que desee solicitar personal”

El diagrama de secuencia anterior comienza cuando el Actor Empresa ingresa al sistema para poder utilizarlo. El sistema le muestra una pantalla para que se registre, proporcionando un login y una contraseña. El usuario proporciona los datos requeridos y éstos son validados por el sistema. Si los datos son incorrectos o son campos en blanco, el sistema mostrará una página en donde se le indica al usuario que su login o password son incorrectos. Por el contrario, si los datos que proporcionó el usuario son validados correctamente, el sistema mostrará una pantalla con las opciones de la Empresa.

### 3.5 Diagramas de colaboración

Al igual que los diagramas de secuencia, los diagramas de colaboración muestran interacciones. Los diagramas de secuencia muestran sucesiones de acciones en el tiempo y los diagramas de colaboración son útiles para mostrar el diseño detallado de procedimientos.

Una vez que los diagramas de secuencia han sido elaborados, el siguiente paso es construir el diagrama de colaboración.

La figura 3.5.1 muestra el diagrama de colaboración que fue diseñado con la ayuda del diagrama de objetos y diagrama de secuencias.

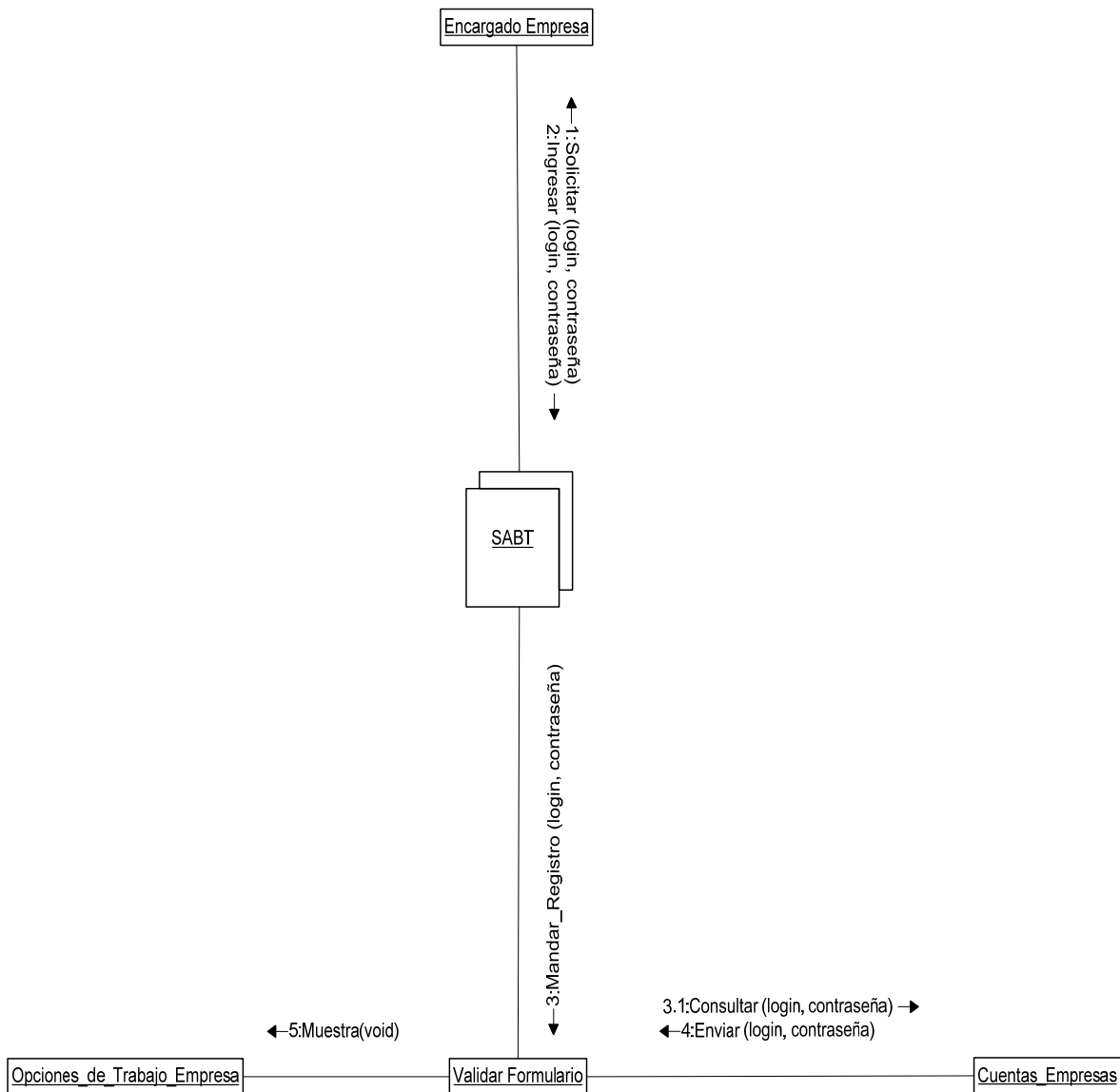


Figura 3.5.1 Diagrama de colaboración “Dar de alta a una empresa”

El diagrama de colaboración muestra la solicitud que hace el Actor Empresa al sistema, éste valida la información y si los datos proporcionados son correctos, el sistema muestra las *opciones de trabajo de la Empresa*.

### 3.6 Diagramas de flujo

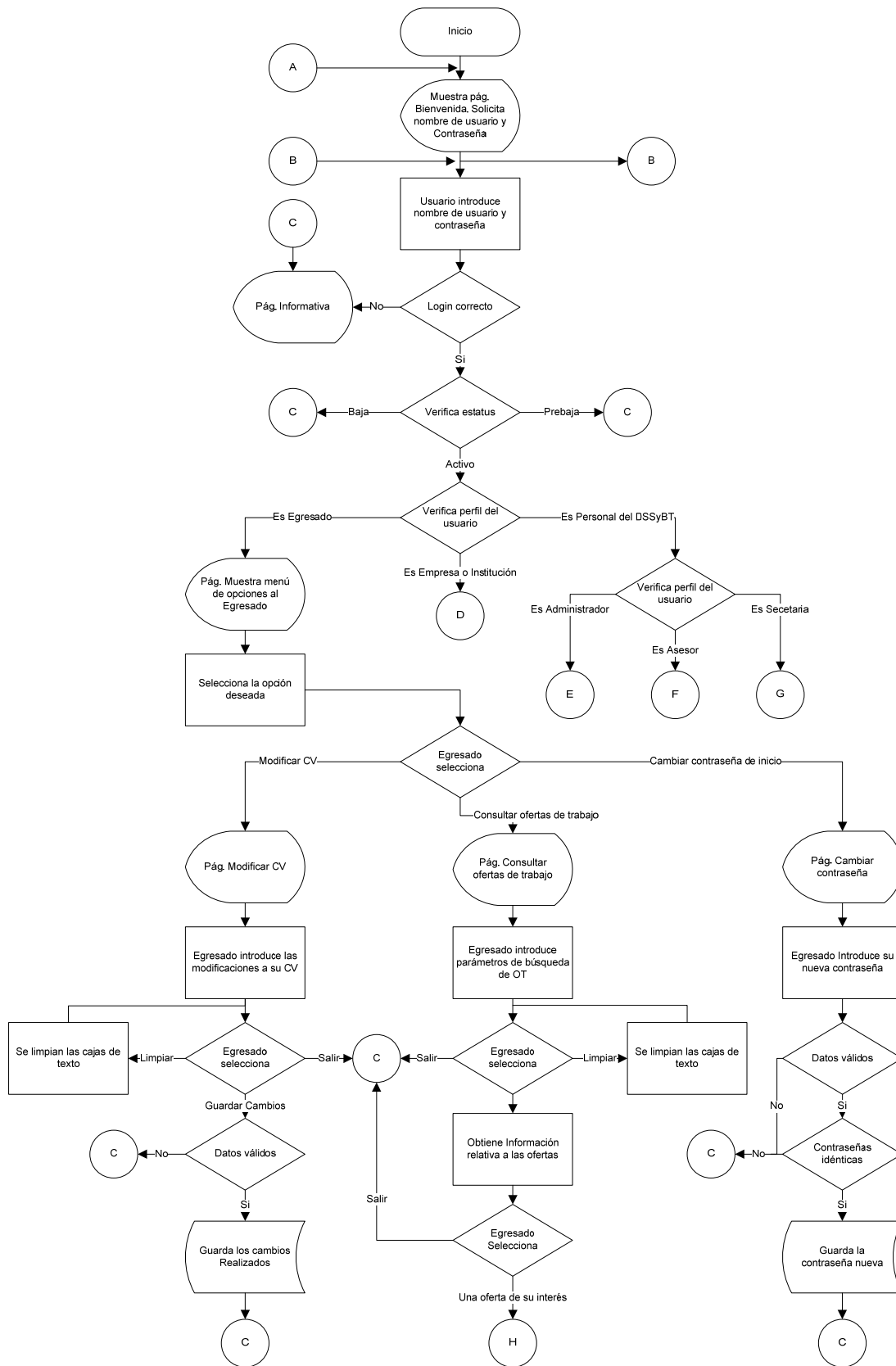


Figura 3.6.1 Diagrama de flujo del SABT (primera parte)

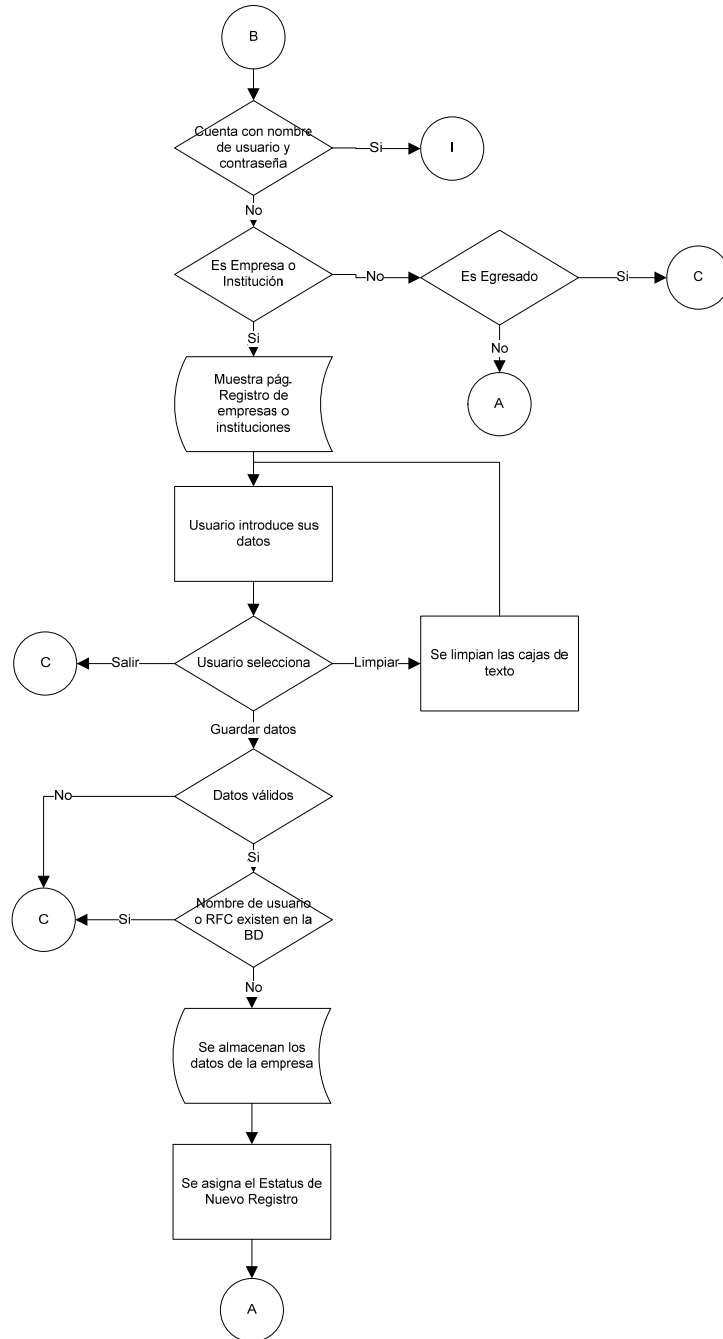
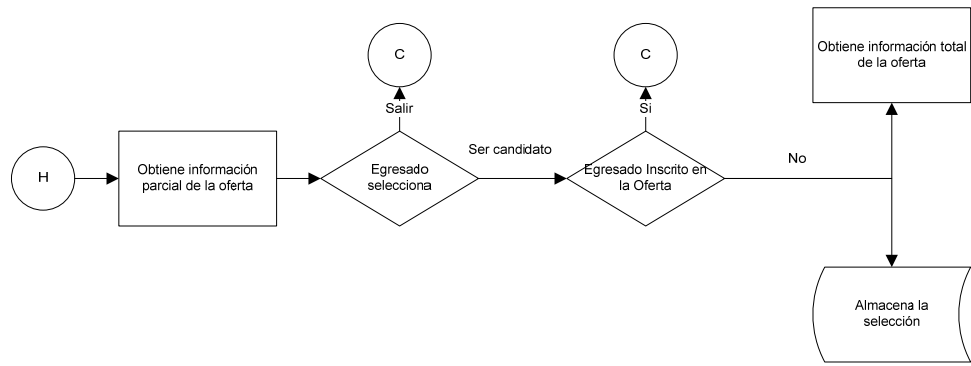


Figura 3.6.2 Diagrama de flujo del SABT (segunda parte)

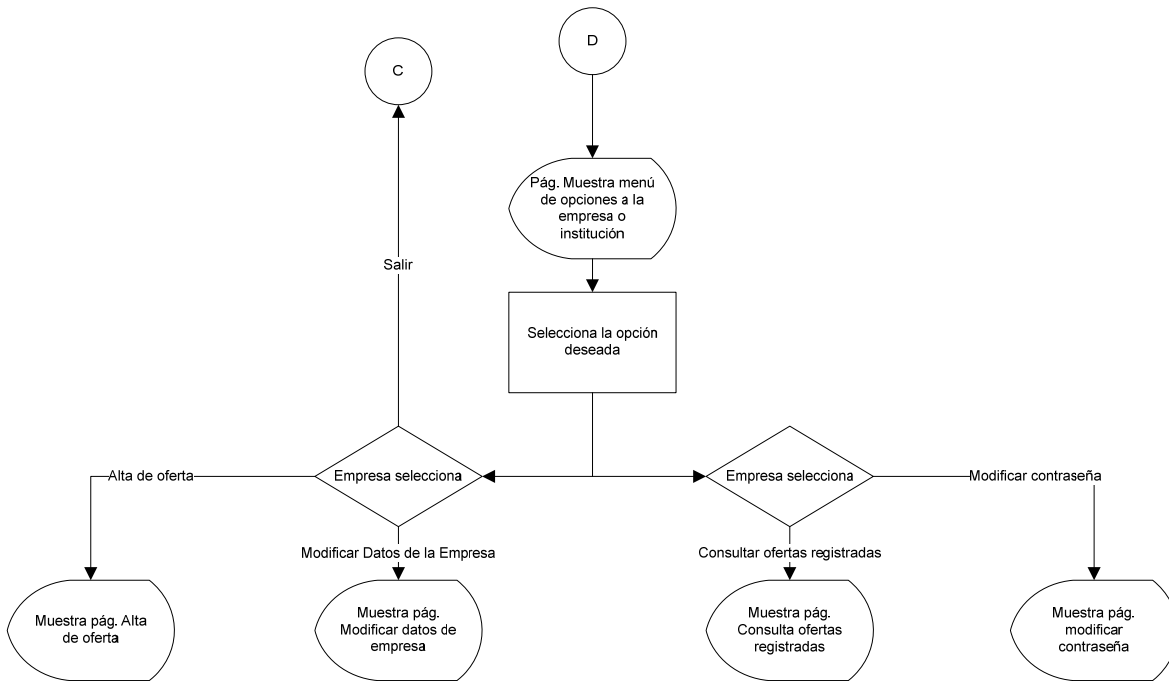


Figura 3.6.3 Diagrama de flujo del SABT (tercera parte)

### 3.7 Normalización

Una vez que contamos con los diagramas de secuencia y colaboración, también procedemos a elaborar la base de datos que utilizará el sistema.

Diseño de la base de datos de la BT

La parte más importante de cualquier base de datos (BD) es su diseño. Si se encuentra bien diseñada, será una herramienta eficiente, la cual nos proporcionará flexibilidad y la información necesaria que requiramos.

El primer paso en el diseño de una BD es adquirir información acerca del proceso (proceso de negocios) que sigue determinada entidad para desempeñar sus tareas y así poder alcanzar sus metas.

Así por ejemplo, nuestro sistema contaría con el siguiente proceso:

1. Mostrar al alumno un mensaje de bienvenida y pedirle que ingrese su nombre de usuario y contraseña

2. Verificar su nombre de usuario y contraseña
  
3. Atender una petición, por ejemplo, la que un alumno realizó en el sitio web del DSSyBT de la Facultad de Psicología

Estos pasos se efectúan cada vez que ingresa un Actor al SABT, por ejemplo la Secretaria, el Administrador del sistema, etc.

El siguiente paso en el proceso de diseño es definir los objetos de negocios. Estos objetos son los componentes que conforman el proceso de negocios, el cual nos servirá para alcanzar un resultado previsto para nuestra aplicación.

Una vez que definimos nuestros objetos, la segunda parte de este paso es la creación de los campos o atributos que describen a los objetos. Podemos pensar en cosas que son utilizadas o asociadas con los objetos.

Definir los objetos de negocios es el inicio de la construcción de la BD. Posteriormente estos objetos serán tablas en la base de datos y la descripción serán los campos de sus tablas.

El tercer paso en el proceso de diseño consiste en establecer las reglas de negocio. Una regla de negocio es un enunciado o un conjunto de enunciados que expresan la forma en que funciona nuestro sistema.

Existen dos tipos de reglas con las cuales vamos a trabajar; las establecidas (reglas impuestas por nuestro sistema) y las implícitas (reglas basadas en sentido común). Para ilustrarlas, nos ayudaremos del ejemplo mencionado en la página anterior, una regla establecida sería que el sistema puede atender a uno o más alumnos. Una regla implícita sería que cada alumno debe tener un nombre de usuario y una contraseña. Definir estas reglas es muy importante para determinar qué tipos de datos debemos utilizar para nuestros campos y si pueden estar vacíos o no.

Las tablas elaboradas se presentan a continuación en la figura 3.7.1.

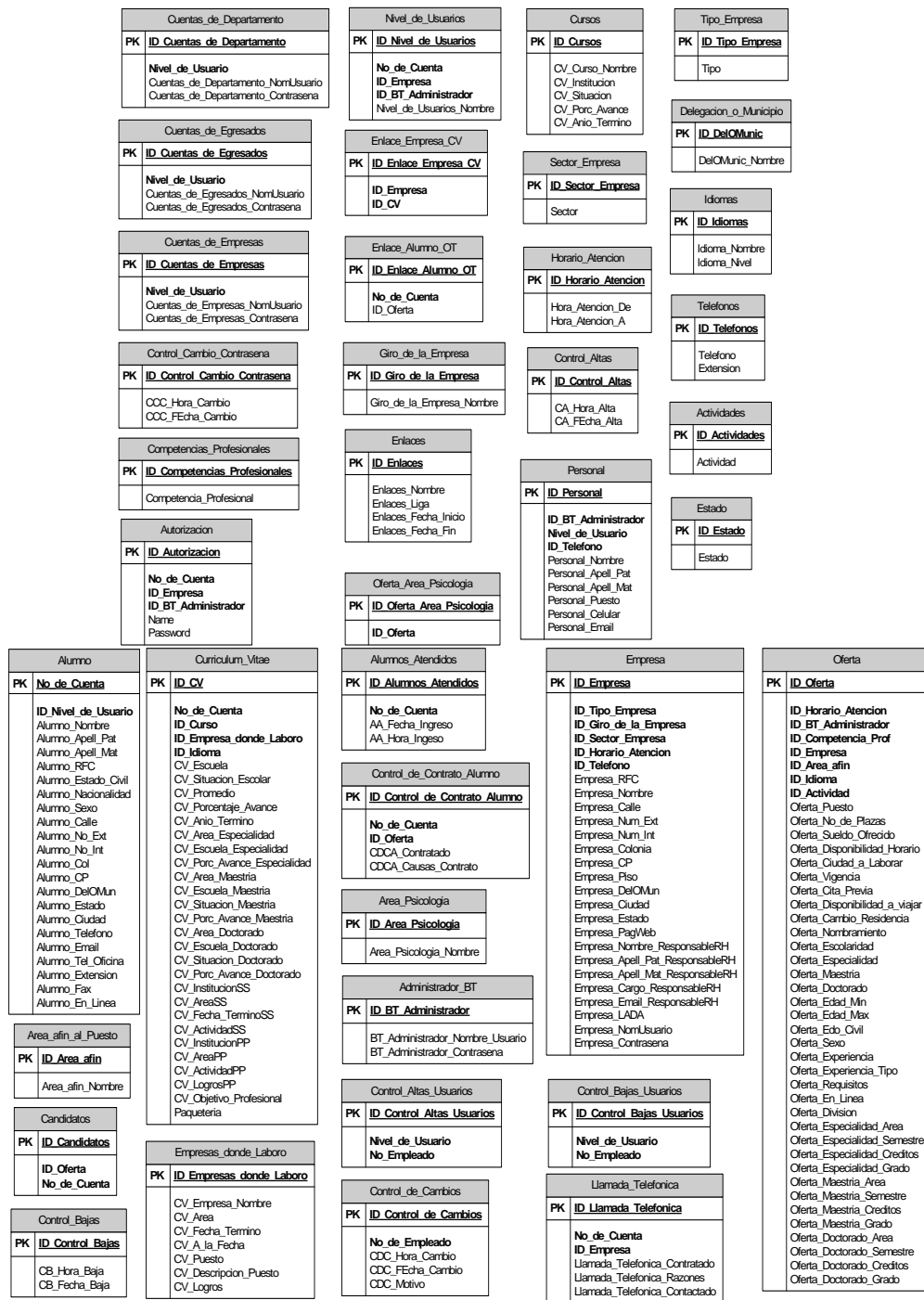


Figura 3.7.1 Estructura de las tablas de la base de datos del SABT

El siguiente paso en el proceso de diseño es hacer un diagrama de nuestro modelo y definir las relaciones que existan entre las tablas. En este momento contamos con un modelo visual de la BD propuesta. Este modelo nos ayuda a revisar nuestro proceso de negocios y los objetos que lo componen. Con el modelo también detectamos si alguna relación no se incluyó o podemos hacer modificaciones en las tablas con mayor facilidad.





A través de la aplicación de las herramientas conceptuales creadas para la gestión de proyectos de software ya descritas en el capítulo anterior, la solución propuesta fue analizada y desarrollada buscando siempre una optimización en los procedimientos ya llevados a cabo dentro del DSSyBT, e inclusive en algunos casos éstos desaparecieron en su realización física, como es el caso de la búsqueda manual de ofertas para los egresados inscritos en la BT.

A continuación se muestra la fase de construcción del sistema, así como el detalle de las opciones que se pueden encontrar dentro del mismo.

# Capítulo 4

## Construcción del sistema



## 4.1 Administración del sistema

En el desarrollo del sistema, surgió la necesidad de hacer las páginas homogéneas en apariencia, además de ser fácilmente identificables y sencillas en su utilización, por lo que se agregó en cada una de ellas la información correspondiente para su buen funcionamiento.

### 4.1.1 Secuencia de pantallas

Las siguientes pantallas describen las características y funcionalidades que se aplican al conjunto de usuarios que pueden utilizar el sistema.

#### 4.1.1.1 Descripción de las pantallas comunes para los diferentes usuarios (Administrador, Asesor, Secretaria, Empresa y Egresado) del sistema

##### Pantalla de bienvenida

Proporciona una breve introducción así como el acceso al sistema a los usuarios. Si no se cuenta con un nombre de usuario para ingresar al sistema, en esta pantalla se podrán registrar los nuevos usuarios.



Figura 4.1.1 Pantalla de bienvenida

- a. El usuario deberá proporcionar su clave de acceso al sistema
- b. El usuario deberá proporcionar su contraseña asociada a su clave
- c. El usuario deberá indicar si es Administrador, Egresado o Empresa
- d. Si la Empresa no cuenta con una clave aquí se registra
- e. Si el Egresado no cuenta con una clave aquí se registra
- f. Enlaces a otras páginas de empleo
- g. Menú de enlaces para ir a la página de inicio del sistema, al directorio, contacto con el DSSyBT y créditos
- h. Contador de visitas

## Pantalla de acceso denegado

Le muestra al usuario que su nombre de usuario y/o contraseña no son válidos y tiene que volver a proporcionarlos.



Figura 4.1.2 Pantalla de acceso denegado

### 4.1.1.2 Flujo de pantallas para el Administrador del sistema

#### Opciones del Administrador

Le permite al Administrador del sistema realizar tareas específicas dentro de 3 campos que son: Empresas, Egresados y Departamento.

El campo Empresas cuenta con un menú que tiene dos opciones: Administración General y Consultas.

Administración General a su vez cuenta con un menú y éstas son las opciones:

- Cambiar estatus de empresas
- Dar de alta empresas recién registradas
- Cambiar estatus de ofertas
- Dar de altas ofertas recién registradas
- Registrar empresas
- Modificar datos de empresas

Consultas cuenta con un menú con las opciones:

- Número de empresas registradas
- Número de ofertas dadas de baja (por área y por división)
- Giro de las empresas
- Puesto ofrecido
- Actividades a desarrollar (por perfil)
- Sueldo ofrecido
- Empresas que tienen más vacantes
- Empresas con mayor éxito
- Empresas que tienen candidatos registrados
- Empresas que no tienen candidatos registrados

El campo de Egresados cuenta con un menú que también tiene dos opciones: Administración General y Consultas.

Administración General cuenta con las siguientes opciones:

- Consultar ofertas
- Registrar Egresado
- Egresados registrados
- Modificar datos del Egresado
- Modificar estatus del Egresado
- Búsqueda de candidatos

Consultas cuenta con las siguientes opciones:

- Número de Egresados registrados (en empresas públicas y privadas)
- Estudiantes atendidos (por sistema)

Finalmente el campo Departamento cuenta con un menú con las siguientes opciones:

Asesores y Secretarias.

Asesores cuenta con las siguientes opciones:

- Inscribir a un nuevo Asesor
- Dar de alta nuevo Asesor
- Modificar estatus del Asesor
- Modificar datos del Asesor

Secretarias cuenta con las siguientes opciones:

- Inscribir a una nueva Secretaria
- Dar de alta nueva Secretaria
- Modificar estatus de la Secretaria
- Modificar datos de la Secretaria

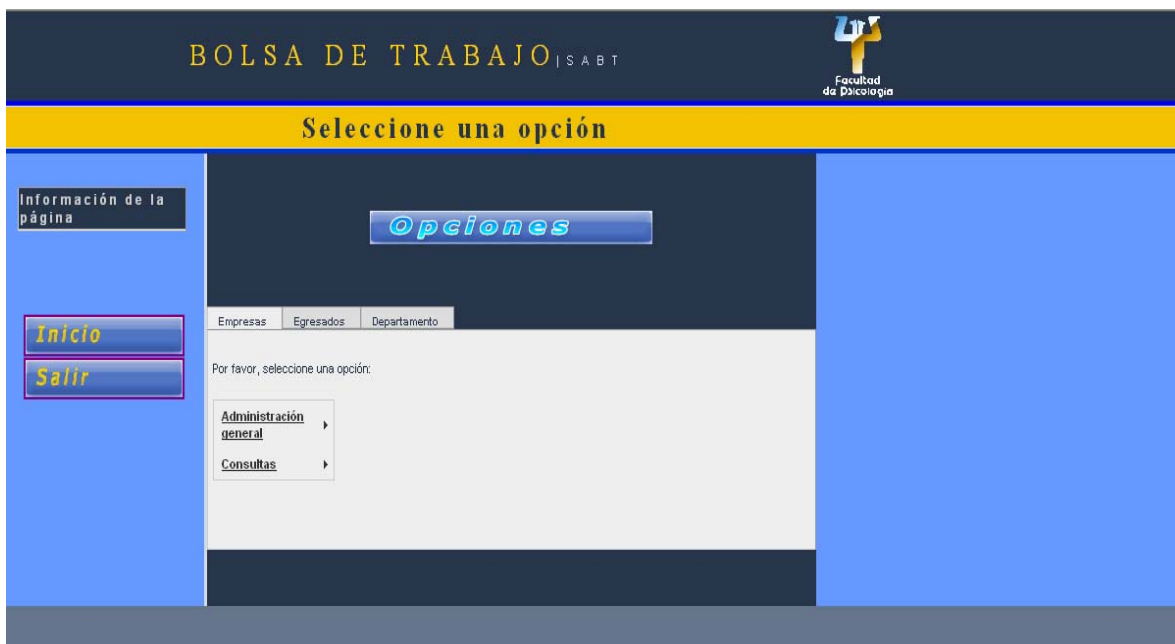


Figura 4.1.3 Pantalla de opciones del Administrador

## Formulario de búsqueda de una Empresa

Le permite al Administrador realizar la búsqueda de una Empresa por tres campos.



BOLSA DE TRABAJO | S A B T Facultad de Psicología

**Proporcione los datos**

Información de la sección

[Inicio](#)

[Salir](#)

**Búsqueda**

RFC:

Nombre de la empresa / institución:

Nombre de usuario:

a

b

c

Figura 4.1.4 Formulario de búsqueda de Empresas

- a. Búsqueda por RFC de la Empresa
- b. Búsqueda por nombre de la Empresa o institución
- c. Búsqueda por nombre de usuario de la Empresa

### Resultado de la búsqueda de una Empresa

Muestra los resultados de las empresas de acuerdo con los campos de búsqueda de la página anterior

BOLSA DE TRABAJO | S A B T Facultad de Psicología

**Seleccione el registro a modificar**

Información de la página

[Inicio](#)

[Salir](#)

[Opciones ADM](#)

**Empresas**

Nombre de la empresa/institución	Nombre de usuario	RFC	Estatus actual
TICKET MASTER	ticketmaster	JUNA811117	Activo
BONAFONT	bonafont oriente	LOYA871211	Activo
UNAM FACULTAD PSICOLOGIA	unamfpsbt	KJUN671231	Activo
UNAM_FP	UNAM FP SSBT	MADI811217	Activo
HERDEZ	alracrobat	FEDC450508	Activo
AKAI	AKAI232323	CONI811217	Activo
GUARAPO	quarapos	FETR435689	Activo
UNAM_FP_01	unam fp ssbt	KAFE671120	Activo
DULCES MIGUELITO	miquelito	DERT567809	Activo
NINTENDO64	Nintendo64	CODI891222	Activo

[Siguiente](#) [Último](#)  
 Registros 1 a 10 de 21

Figura 4.1.5 Resultado de búsqueda de una Empresa

## Cambiar de estatus a una Empresa

Esta pantalla permite cambiar de estatus a una determinada Empresa

Información de la página

[Inicio](#)

[Salir](#)

[Opciones ADM](#)

### Empresas

RFC	Nombre de la empresa	Estatus actual	Seleccione el nuevo estatus
JUNA811117	TICKET MASTER	Activo	Activo <input type="button" value="Enviar"/>

a

Figura 4.1.6 Formulario para cambiar estatus a una Empresa

a. Permite escoger el nuevo estatus (alta, prebaja o baja) para determinada Empresa

## Cambio exitoso

Muestra al usuario que el cambio se efectuó con éxito



Figura 4.1.7 Pantalla de cambio exitoso

## Dar de alta una Empresa recién registrada

Permite visualizar todas las empresas que se han registrado, pero que todavía no las ha dado de alta el Administrador.

Seleccione el nuevo Estatus	
RFC	MAZA651211
Nombre de usuario:	opticas_devlyn
Nombre de la empresa	DEVLYN
Calle	MIRAMONTES
Número exterior	33
Número interior	0
Colonia	SANTO DOMINGO
Código postal	23451
Piso	0
Ciudad	DF
Delegación o municipio	COYOACÁN
Estado	
Correo electrónico	opticas_devlyn_09@devlyn.org
Página web	
Estatus actual	Nuevo Reg
Seleccione el nuevo estatus	Activo <input type="button" value="Enviar"/>
Registros faltantes	1

Figura 4.1.8 Formulario para dar de alta a una Empresa recién registrada

- a. Permite escoger el nuevo estatus (alta, prebaja o baja) para una Empresa recién registrada

## Formulario búsqueda de ofertas de trabajo

Le permite al Administrador realizar la búsqueda de ofertas de trabajo por tres campos.



Figura 4.1.9 Formulario de búsqueda de ofertas de trabajo

- a. Realiza la búsqueda por clave de oferta de trabajo
- b. Realiza la búsqueda por nombre de la Empresa
- c. Realiza la búsqueda por nombre de usuario de la Empresa

### Resultado de la búsqueda de ofertas de trabajo

Muestra al Administrador los resultados de la búsqueda de ofertas de trabajo de acuerdo a los campos de la página anterior.



Figura 4.1.10 Pantalla de resultado de búsqueda de ofertas de trabajo

## Cambio de estatus a una oferta de trabajo

El Administrador podrá cambiar el estatus que tenga una oferta de trabajo por uno nuevo.



The screenshot shows the 'Ofertas' section of the 'BOLSA DE TRABAJO | S A B T' website. On the left, there is a sidebar with 'Información de la página' and buttons for 'Inicio', 'Salir', and 'Opciones ADM'. The main content area features a table of job offers and a form to change their status.

Id oferta	Nombre de la oferta	Nombre de usuario	Estatus actual	Seleccione el nuevo estatus
37	GERENTE DE RECURSOS HUMANOS	ticketmaster	Baja	Activo <input type="button" value="Enviar"/>

A callout bubble labeled 'a' points to the 'Activo' dropdown menu in the form.

Figura 4.1.11 Formulario para cambiar estatus de una oferta de trabajo

- Permite escoger el nuevo estatus (activo, prebaja o baja) para una oferta de trabajo

## Dar de alta ofertas de trabajo recién registradas

El Administrador podrá dar de alta ofertas recién registradas o darles el estatus de prebaja o baja.



The screenshot shows the 'Ofertas' section of the 'BOLSA DE TRABAJO | S A B T' website. A detailed form for adding a new job offer is displayed. The form includes fields for offer ID, user name, position, salary, and activity.

Seleccione el nuevo Estatus	
Id de la oferta	44
Nombre de usuario:	constructora_roma
Puesto	MONITOR CLÍNICO I
Número de plazas	2
Sueldo ofrecido	10000
Ciudad a laborar	DF
Horario 1 de:	09:00:00
Horario 1 a:	13:00:00
Horario 2 de:	15:00:00
Horario 2 a:	18:00:00
Edad mínima	0
Edad máxima	0
Actividad 1	VERIFICACIÓN EXACTA Y MINUCIOSA DE DOCUMENTOS

Figura 4.1.12a Formulario para dar de alta ofertas de trabajo recién registradas

Actividad 2	REALIZAR VISITAS A LOS SITIOS PARA VERIFICAR QUE LOS PROCESOS SEAN ADECUADOS.
Actividad 3	FACILITAR LA COMUNICACIÓN EFECTIVA ENTRE LOS SITIOS DE INVESTIGACIÓN Y EL CLIENTE.
Experiencia en:	EXPERIENCIA MÍNIMA DE 1 AÑO COMO MONITOR CLÍNICO O COORDINADOR DE ESTUDIO.
Requisitos:	
Estatus actual	Nuevo Reg
Seleccione el nuevo estatus	Activo <input type="button" value="Enviar"/>
Registros faltantes	1

Registros 1 a 1 de 1

Figura 4.1.12b Formulario para dar de alta ofertas de trabajo recién registradas

- a. Seleccionamos el nuevo estatus de la oferta de trabajo recién registrada (activo, prebaja, baja )

### Formulario para registrar una nueva Empresa

El Administrador podrá registrar una nueva Empresa.

BOLSA DE TRABAJO | S A B T 

Proporcione sus datos

Información de la página

Inicio

Salir

## Empresas

Por favor ingrese los siguientes datos

\*RFC:  a

\*Nombre de la empresa:  b

Por favor escoja un Login

\*Nombre de usuario:  c

\*Contraseña:  d

\*Repetir contraseña:  e

Figura 4.1.13 Formulario para registrar una nueva Empresa

- a. RFC de la Empresa
- b. Nombre de la Empresa
- c. Nombre de usuario de la Empresa
- d. Contraseña del usuario
- e. Repetir contraseña del usuario

El sistema validará los datos introducidos por el usuario para poder continuar con el llenado del formulario.



Figura 4.1.14 Pantalla de seguimiento de inserción de nueva Empresa a la BD

En caso de que los datos proporcionados por el usuario existan en el sistema, aparecerá la siguiente pantalla indicando que se proporcionen otros datos.



Figura 4.1.15 Pantalla indicativa de error en el seguimiento de inserción de nueva Empresa a la BD

Si el nombre de usuario es correcto el Administrador podrá seguir llenando el formulario con los datos de la Empresa.

**BOLSA DE TRABAJO ISABT** Facultad de Psicología

**Proporcione sus datos**

**Empresas**

Tipo de empresa:  a

Giro:  b Sector:

**Domicilio**

Calle:  c

Número exterior:  e Número interior:  f

Colonia:  g Piso:  h

Código Postal:  i

Delegación o Municipio:  j

Ciudad:  k

Estado:  l

Correo electrónico:  m

Página web:  n

http://

Figura 4.1.16 Formulario para registrar una nueva Empresa (continuación)

- a. Tipo de Empresa
- b. Sector de la Empresa
- c. Giro de la Empresa
- d. Calle de la Empresa
- e. Número exterior
- f. Número interior
- g. Piso
- h. Colonia
- i. Código postal
- j. Delegación o municipio
- k. Ciudad
- l. Estado
- m. Correo electrónico del encargado de la Empresa
- n. Página web de la Empresa



## Proporcione sus datos

Información de la página

**Empresas**

DATOS DEL ENCARGADO DE LA EMPRESA

\*Nombre:  a

\*Apellido paterno:  b

Apellido materno:  c

\*Cargo:  d

LADA:  e

\*Teléfono:  f

Extensión:  g

Teléfono 2:  h

Extensión 2:  i

Horario de atención:

\*1er turno

De:  a  l

2do turno

De:  a  m

Inicio

Salir

Figura 4.1.17 Formulario para registrar una nueva Empresa (continuación)

- a. Nombre del encargado de la Empresa
- b. Apellido paterno del encargado de la Empresa
- c. Apellido materno del encargado de la Empresa
- d. Cargo del encargado de la Empresa
- e. LADA de la oficina o departamento del encargado de la Empresa
- f. Primer número telefónico del encargado de la Empresa
- g. Primera extensión telefónica del encargado de la Empresa
- h. Segundo número telefónico del encargado de la Empresa
- i. Segunda extensión telefónica del encargado de la Empresa
- j. Hora inicio del primer horario de atención
- k. Hora fin del primer horario de atención
- l. Hora inicio del segundo horario de atención
- m. Hora fin del segundo horario de atención

Al terminar el formulario el sistema mostrará la siguiente pantalla indicando que los datos fueron introducidos satisfactoriamente.



Figura 4.1.18 Pantalla informativa de éxito en la inserción de nueva Empresa en la BD

### Modificar datos de la Empresa

Le permite al Administrador modificar los datos de la Empresa que se requieran.



Figura 4.1.19a Formulario de modificación de los datos de las Empresas

Figura 4.1.19b Formulario de modificación de los datos de las Empresas

- a. Nombre de la Empresa
- b. Calle de la Empresa
- c. Número exterior
- d. Número interior
- e. Colonia
- f. Código postal
- g. Piso
- h. Ciudad
- i. Estado
- j. Correo electrónico del encargado de la Empresa
- k. Página web de la Empresa
- l. Nombre del encargado de la Empresa
- m. Apellido paterno del encargado de la Empresa
- n. Apellido materno del encargado de la Empresa
- o. Cargo del encargado de la Empresa
- p. LADA de la oficina o departamento del encargado de la Empresa
- q. Delegación o municipio
- r. Primer número de telefónico del encargado de la Empresa
- s. Segundo número telefónico del encargado de la Empresa
- t. Primera extensión telefónica del encargado de la Empresa
- u. Segunda extensión telefónica del encargado de la Empresa
- v. Hora inicio del primer horario de atención
- w. Hora fin del primer horario de atención
- x. Hora inicio del segundo horario de atención
- y. Hora fin del segundo horario de atención

## Número de Empresas registradas

Le permite al Administrador consultar las empresas que se han registrado en el sistema.

The screenshot shows a web interface for consulting registered companies. At the top, it says 'BOLSA DE TRABAJO | S A B T' and 'Facultad de Psicología'. The main title is 'Número de empresas registradas'. On the left, there are navigation buttons: 'Inicio', 'Salir', and 'Opciones ADM'. The main area is titled 'Consulta' and contains the following form elements:

- 'Seleccione el tipo de empresa / institución:' with a dropdown menu showing 'Tipo: Pública' (labeled 'a').
- 'Seleccione un periodo de tiempo:' with two date selection fields: 'Fecha de Inicio:' (labeled 'b') and 'Fecha de Fin:' (labeled 'c').
- 'Limpiar' and 'Enviar' buttons.

Figura 4.1.20 Formulario de la consulta del número de Empresas registradas

- a. Tipo de la Empresa
- b. Fecha de inicio
- c. Fecha de fin

Una vez que se haya seleccionado el tipo de Empresa y el intervalo de fechas, el sistema arrojará los registros que cumplan con las especificaciones.

BOLSA DE TRABAJO | S A B T Facultad de Psicología

## Número de empresas registradas

**Inicio**

**Salir**

**Opciones ADM**

**Reporte**

Intervalo de consulta: del 2008-10-08 al 2008-10-23

Existen 2 registros de empresas tipo Pública

Hombre de usuario	RFC de la empresa	Hombre de la empresa	Calle	Correo electrónico del contacto
umbrellacorp	LOLO871211	UMBRELLA CORPORATION	ROSALES	unam_fp@unam.com
UNAM_FP_SSBT	MADI811217	UNAM_FP	ALCANFORES	unam_pf@unam.mx

Figura 4.1.21 Pantalla del resultado de la consulta

### Ofertas dadas de baja

Le permite al Administrador consultar las ofertas de las empresas que se han dado de baja en el sistema.

BOLSA DE TRABAJO | S A B T Facultad de Psicología

## Número de ofertas dadas de baja

Información de la página

**Inicio**

**Salir**

**Opciones ADM**

**Consulta**

Seleccione el tipo de Empresa / Institución:

Area:

Seleccione un periodo de tiempo:

Fecha de Inicio:

Fecha de Fin:

a

b

c

Figura 4.1.22 Formulario de la consulta del número de ofertas dadas de baja

- Área de la oferta de trabajo
- Fecha de inicio
- Fecha de fin

Una vez que se haya seleccionado el área y el intervalo de fechas, el sistema arrojará los registros que cumplan con las especificaciones.

**BOLSA DE TRABAJO** | S A B T

Facultad de Psicología

**Número de ofertas dadas de baja por área**

Información de la página

**Reporte**

Intervalo de consulta: del 2007-10-09 al 2008-10-23

Existen 4 ofertas para el área Clínica

Id oferta	Nombre de usuario	Puesto ofrecido	Vacantes	Estatus actual	Fecha de registro	Área de especialidad
33	dimeranayyosalto	MERCADOLOGO	1	Baja	2008-05-16	CLÍNICA
34	ticketmaster	ENCARGADO DE VENTAS	2	Baja	2008-05-16	CLÍNICA
36	ticketmaster	VENTAS	1	Baja	2008-05-20	CLÍNICA
37	ticketmaster	GERENTE DE RECURSOS HUMANOS	6	Baja	2008-05-20	CLÍNICA

Figura 4.1.23 Pantalla del resultado de la consulta

#### 4.1.1.3 Flujo de pantallas para las Empresas

##### Opciones de la Empresa

Le permite a la Empresa realizar tareas específicas como:

- Registrar ofertas de trabajo
- Consultar ofertas registradas
- Cambiar datos de la Empresa
- Cambiar contraseña

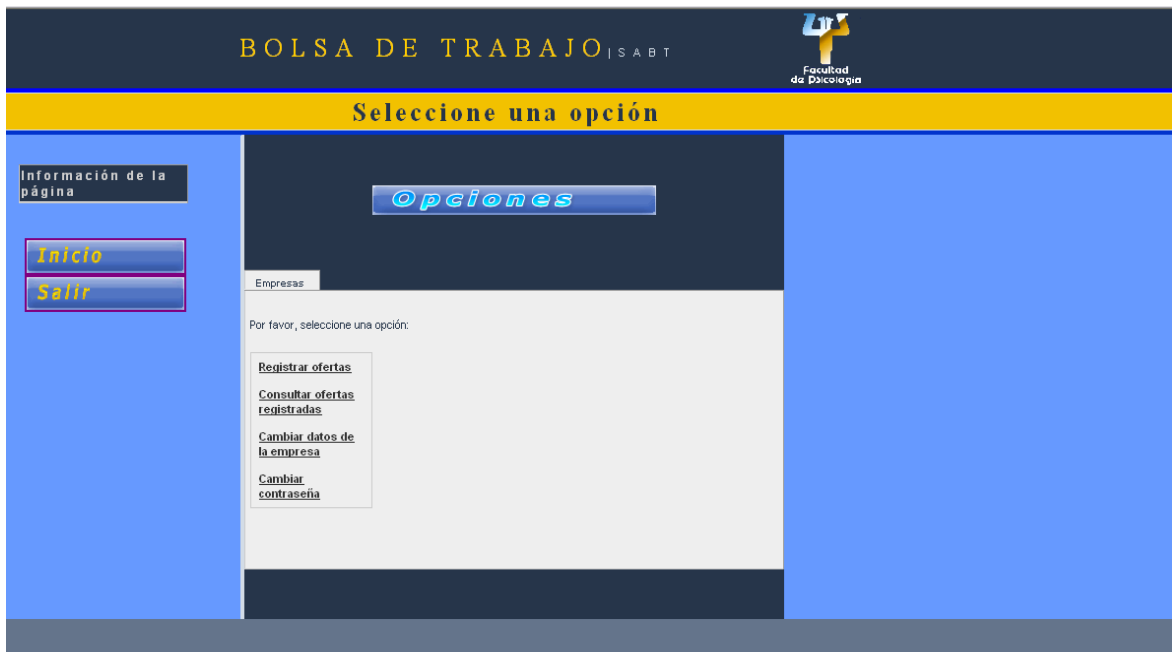


Figura 4.1.24 Pantalla de opciones para las Empresas

## Registrar ofertas de trabajo

Le permite a la Empresa realizar la captura de sus ofertas de trabajo.

Figura 4.1.25a Formulario para registrar ofertas de trabajo

The image shows a dark-themed web form for job offers. At the top, there are two radio button options: "Previa cita:" with "Si" and "No" options, and "Tipo de nombramiento:" with four radio button options: "Planta", "Honorarios", "Confianza", and "Contrato". Below this is a section titled "Actividades a desarrollar" containing six numbered input fields (1-6). At the bottom, there is a section titled "Área afín al puesto" with a dropdown menu labeled "Área: Selecciona:". At the very bottom are two buttons: "Limpiar" and "Continuar".

Callout letters are placed around the form:
 

- m**: Points to the "Tipo de nombramiento:" radio buttons.
- l**: Points to the "Previa cita:" radio buttons.
- n**: Points to the "Actividades a desarrollar" input fields.
- o**: Points to the "Área: Selecciona:" dropdown menu.

Figura 4.1.25b Formulario para registrar ofertas de trabajo

- a. Nombre del puesto ofrecido
- b. Número de plazas
- c. Sueldo
- d. Hora inicio del primer turno de trabajo
- e. Hora fin del primer turno de trabajo
- f. Hora inicio del segundo turno de trabajo
- g. Hora fin del segundo turno de trabajo
- h. Ciudad donde se laborará
- i. Si se necesita disponibilidad de horario
- j. Si se necesita disponibilidad para viajar
- k. Si se necesita cambiar de residencia
- l. Si se necesita previa cita
- m. Tipo de nombramiento
- n. Actividades a desarrollar
- o. Área afín al puesto



## Proporcione sus datos

Información de la página

[Inicio](#)

[Salir](#)

### Ofertas

PERFIL DEL CANDIDATO

Escolaridad  
 egresado  titulado a

	Área	Semestre	Créditos	Grado
<input type="checkbox"/> Especialidad				% <span style="float: right;">b</span>
<input type="checkbox"/> Maestría				% <span style="float: right;">c</span>
<input type="checkbox"/> Doctorado				% <span style="float: right;">d</span>

Estado civil:  
 Soltero  Casado  Indistinto e

Idioma   
 Básico  Medio  Avanzado f

Idioma   
 Básico  Medio  Avanzado g

Edad  
 Mínima:  j  
 Máxima:  k

Sexo  
 Femenino  Masculino  Indistinto l

Tiempo de vigencia de la oferta

Fecha de inicio de publicación:   m

Fecha de fin de la publicación:   n

Figura 4.1.26 Formulario para registrar ofertas de trabajo (continuación)

- a. Escolaridad
- b. Área, semestre, créditos y grado de la especialidad
- c. Área, semestre, créditos y grado de la maestría
- d. Área, semestre, créditos y grado del doctorado
- e. Estado civil
- f. Idioma 1
- g. Nivel idioma 1
- h. Idioma 2
- i. Nivel idioma 2
- j. Edad mínima
- k. Edad máxima
- l. Sexo
- m. Fecha de inicio de publicación de la oferta
- n. Fecha de fin de la publicación de la oferta

BOLSA DE TRABAJO | S A B T Facultad de Psicología

**Proporcione sus datos**

Información de la página

[Inicio](#)

[Salir](#)

Ofertas

PERFIL DEL CANDIDATO

**Competencias Profesionales Personales**

- Adaptabilidad
- Aprendizaje continuo
- Creatividad/Innovación
- Confianza
- Energía
- Impacto
- Impacto
- Iniciativa
- Integridad
- Tolerancia al estrés
- Tenacidad

**De Negocios/Ventas**

- Orientación al cliente
- Habilidad de persuasión

**Interpersonales**

- Asesoría
- Comunicación
- Trabajo en equipo

**De Liderazgo**

- Liderazgo
- Compartir responsabilidades
- Compartir responsabilidades
- Participación en reuniones

**De Toma De Decisiones**

- Toma de decisiones
- Planeación y Organización

Experiencia:  Necesaria  No necesaria

En:

Otros requisitos:

Figura 4.1.27 Formulario para registrar ofertas de trabajo (continuación)

- a. Competencias profesionales
- b. Tipo de experiencia
- c. Otros requisitos

#### 4.1.1.4 Flujo de pantallas para los Egresados

##### Opciones de los Egresados

Le permite al Egresado realizar tareas específicas como:

- Consultar ofertas de trabajo
- Actualizar currículum
- Cambiar contraseña



Figura 4.1.28 Pantalla de opciones para los Egresados

## Búsqueda de ofertas de trabajo

Le permite al Egresado consultar las ofertas de trabajo dadas de alta en el sistema.



Figura 4.1.29 Formulario de búsqueda de ofertas de trabajo

El Egresado tendrá la opción de realizar la búsqueda de ofertas por cuatro campos que son:

- a. Ciudad
- b. Sueldo
- c. Puesto
- d. Área de la psicología

El sistema mostrará las ofertas de trabajo que existen en el sistema de acuerdo con los parámetros de búsqueda solicitados previamente.

Nombre de la oferta	Ciudad	Sueldo	Área de la psicología	Experiencia
TERAPEUTA	GUADALAJARA	12000	EDUCATIVA Y DESARROLLO	

Figura 4.1.30 Resultado de búsqueda de ofertas de trabajo

El Egresado tendrá que elegir una oferta de su interés para que el sistema le proporcione sólo algunos datos de la misma.



Figura 4.1.31 Pantalla con datos parciales de una oferta de trabajo seleccionada

Si después de ver la oferta el Egresado está interesado en que se le proporcionen más datos, tendrá que dar clic en el link “Dar clic aquí si le interesa obtener más información de la oferta”, con esto, el Egresado automáticamente pasaría a ser candidato de esa oferta de trabajo.



Figura 4.1.32a Pantalla con el detalle de una oferta de trabajo seleccionada

Página WEB	www.hermanosquiroz.com.mx
Nombre del responsable de la oferta	RENÉ
Apellido paterno	GARCÍA
Apellido materno	CRUZ
LADA	55
Teléfono	54121710
Extensión	56
Teléfono	0
Extensión	0
Horario de atención de:	09:00:00
Horario de atención a:	12:00:00
Horario de atención de:	15:00:00
Horario de atención a:	19:00:00

Ya eres candidato a esta oferta

Figura 4.1.32b Pantalla con el detalle de una oferta de trabajo seleccionada

#### 4.1.1.5 Flujo de pantallas para las Secretarías

##### Opciones de la Secretaria

Le permite a la Secretaria realizar tareas específicas como:

- Modificar datos
- Cambiar contraseña
- Registrar a un Egresado
- Dar de alta a un Egresado
- Modificar datos de un Egresado



Figura 4.1.33 Pantalla con opciones para la Secretaria

## Cambiar estatus a Egresados inscritos

Le permite a la Secretaria cambiar el estatus (activo, prebaja y baja) de los nuevos Egresados registrados.

La Secretaria elegirá un registro para otorgarle un nuevo estatus.



Figura 4.1.34 Pantalla con registros a modificar

Una vez que se haya seleccionado el registro se selecciona el nuevo estatus.

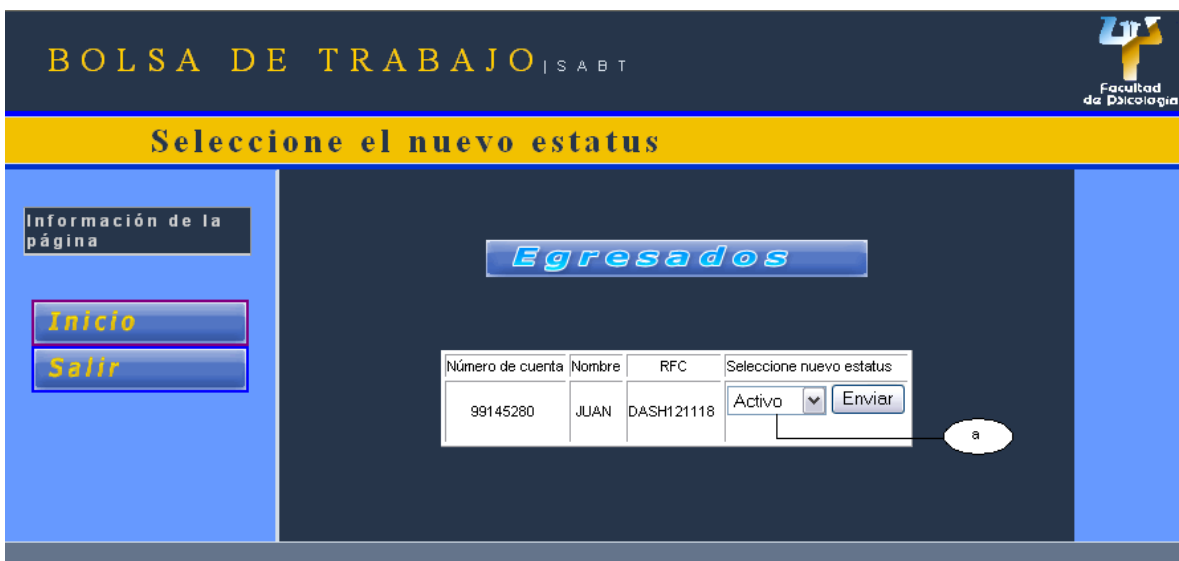


Figura 4.1.35 Formulario para cambiar estatus a Egresados inscritos

- a. Nuevo estatus (activo, prebaja y baja)

#### 4.1.1.6 Flujo de pantallas para los Asesores

##### Opciones del Asesor

Le permite al Asesor realizar tareas específicas como:

- Modificar datos
- Cambiar contraseña
- Llenar formulario de llamada telefónica

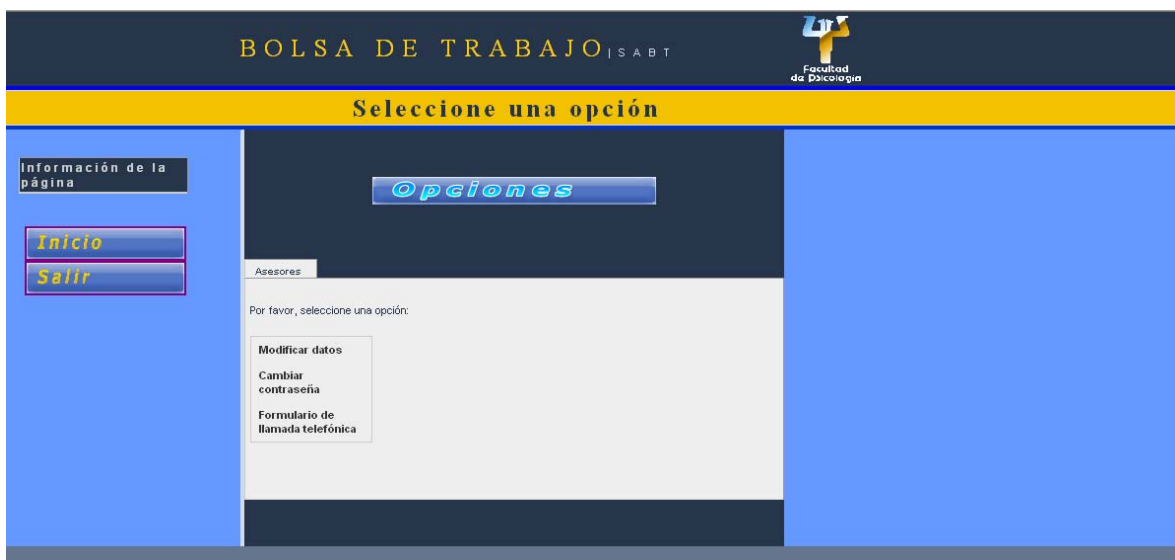


Figura 4.1.36 Pantalla con opciones para el Asesor

##### Formulario de llamada telefónica

Le permite al Asesor llenar un formulario del Egresado que tuvo o no éxito en su entrevista de trabajo.

El Asesor ingresará el número de cuenta del Egresado.





Figura 4.1.37 Formulario de llamada telefónica

- a. Número de cuenta

El sistema validará el número de cuenta ingresado y mostrará la siguiente pantalla en caso de que el dato introducido sea válido.



Figura 4.1.38 Pantalla con información de inserción de registro en la BD

Por el contrario, si el número de cuenta introducido ya se encuentra registrado en la base de datos, el sistema le informará al usuario que esa información ya fue proporcionada.



Figura 4.1.39 Pantalla con información de inserción de registro en la BD

Por último, si el número de cuenta proporcionado no ha sido registrado el sistema mostrará un formulario para que el usuario lo pueda llenar.

Figura 4.1.40 Formulario de llamada telefónica

- a. Si el Egresado fue contactado o no (vía telefónica)
- b. Si el Egresado fue contratado o no
- c. Razones por las cuales el Egresado fue o no contratado

## 4.2 Realización de pruebas

### 4.2.1 Pruebas realizadas por el equipo de trabajo

El equipo realizó diversas pruebas al sistema para verificar la funcionalidad bajo diversas circunstancias.

Dentro del diseño de las pruebas se tomó en cuenta el aspecto funcional, la homogeneidad de la interfaz del usuario así como la seguridad en la carga de información y el acceso a los usuarios.

Se obtuvieron diversos resultados que denotaron algunos errores en varios aspectos, pero se corrigieron en su totalidad. Los resultados de dichas pruebas se muestran a continuación.

#### Pruebas de software

**Caso de uso:** Realizar el registro de un Egresado en el SABB como Administrador.

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Validaciones	<p>Verificar que se cuenten con las validaciones preventivas para que:</p> <ul style="list-style-type: none"><li>• Se llenen de manera obligatoria los campos que se requieran.</li><li>• Los campos numéricos permitan solamente datos numéricos, ya sea como entero o como decimal según se requiera en los formularios.</li><li>• Los campos de texto permitan solamente letras.</li><li>• Los campos alfanuméricos permitan capturar tanto letras como números, teniendo en cuenta algunas veces caracteres especiales (-,;,_, etc.) según lo requiera el campo del formulario.</li><li>• Se limite la longitud de captura en los campos de acuerdo a las especificaciones del diccionario de datos.</li></ul>	<p>Se identifica el error en el campo y se le notifica al usuario para que pueda volver a ingresarlo.</p>

Observaciones	El sistema no permitió la captura de números en el campo de "Calle", por lo tanto la respuesta esperada es incorrecta, pues existen direcciones cuya "Calle" incluye números. Se procedió a realizar las correcciones.	
Controles	Verificar que no se pueda acceder al sistema si no se cuenta con un nombre de usuario y una contraseña. Verificar que el nombre de usuario y la contraseña coincidan con el nivel de usuario que quiera acceder al sistema. Verificar que la fecha de registro de un Egresado se asigne automáticamente. Verificar que se capturen los campos obligatorios en los formularios.	Se accionan los controles dejando pasar sólo información válida garantizando un adecuado registro de información y generando un flujo de pantallas establecido.
Observaciones	El SABT sólo muestra las páginas a los usuarios dependiendo de su nombre de usuario, contraseña y nivel de usuario El sistema permitió la entrada de Egresados sin importar su estatus (alta, prebaja y baja), por lo que se corrigió el comportamiento erróneo.	
Integridad de datos	Verificar que el Número de Cuenta, nombre de usuario y RFC sean únicos. Verificar que los cambios de datos por parte de los usuarios se vean reflejados en las próximas consultas.	Garantizar que la información de la BD sea congruente, evitando tener información innecesaria.
Observaciones	El sistema permitió ingresar registros sólo con el número de cuenta repetido, por lo que se corrigió el comportamiento erróneo.	

Tabla 4.2.1 Pruebas de software para el caso de uso Registro de un Egresado

**Caso de uso:** Autorizar alta de un Egresado en el SABT como Administrador.

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Validaciones	Cuando el usuario seleccione un número de cuenta a dar de alta, el sistema debe mostrar los datos del alumno, así como la opción de validación correspondiente.	Se identifica el error en el campo y se le notifica al usuario para que pueda regresar a la pantalla anterior.
Observaciones	El sistema mostró correctamente los datos del Egresado elegido y se mostró el control para cambiar su estatus a "alta"	
Controles	Verificar que cuando el usuario elige la opción	Se accionan los controles dejando

	de “alta” el sistema debe modificar el campo correspondiente en la base de datos.  Verificar que el nombre de usuario y la contraseña coincidan con el nivel de usuario que quiera acceder al sistema.  Verificar que el sistema muestre sólo a Egresados que estén recién registrados.	pasar sólo información válida garantizando un adecuado registro de información y generando un flujo de pantallas establecido.
Observaciones	No existió ninguna página que permitiera el ingreso de algún usuario sin tener un Nombre de Usuario, una Contraseña y un Nivel de Usuario diferente al del Administrador, definido dentro del SABT.	
Integridad de datos	Verificar que los cambios de datos por parte del usuario se vean reflejados en las próximas consultas.	Garantizar que la información de la BD sea congruente, evitando tener información innecesaria.
Observaciones	El sistema no tuvo problemas al evitar mostrar Egresados con estatus modificado.	

Tabla 4.2.2 Pruebas de software para el caso de uso Autorizar alta de un Egresado en el SABT

### Pruebas interfaz de usuario

**Caso de uso:** Dar de baja una oferta de trabajo como Administrador del sistema.

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Pantallas	Verificar que el diseño de las pantallas es consistente en todos los módulos del sistema. Verificar que el tamaño, color y tipo de letra es la misma para los campos, botones, encabezados y mensajes, cambiando algunas veces en apartados importantes. Verificar que la posición de los botones es la misma en las diferentes pantallas (arriba, abajo, derecha, izquierda o centrado). Verificar que se puede navegar utilizando tanto las teclas del cursor como el tabulador. Verificar que se puede acceder a las opciones mediante el uso del ratón. Verificar que la distribución de los campos sea uniforme y con espacio suficiente para su fácil	Navegar a través de las pantallas de forma fácil.  La ayuda incluida en cada página hace que el usuario pueda operar el sistema de forma rápida y fluida (sin tener que descifrar textos).

	<p>lectura y captura.</p> <p>Verificar que la ayuda para cada sección sea legible y entendible para el usuario en todo momento.</p>	
Menú	<p>Verificar que el menú es el mismo para todos los módulos del sistema:</p> <ul style="list-style-type: none"> <li>• Mismo tipo de letra</li> <li>• Mismo color</li> <li>• Misma forma de acceder a las diferentes opciones</li> </ul> <p>Verificar que el menú agrupe opciones que se relacionan y que en conjunto forman un módulo.</p>	<p>Fácil acceso a los componentes del sistema.</p> <p>Fácil lectura y selección de opciones.</p>
Mensajes	<p>Verificar que el estilo de todos los mensajes es el mismo:</p> <ul style="list-style-type: none"> <li>• Mismo tipo de letra</li> <li>• Mismo color</li> <li>• Misma posición</li> <li>• Misma ubicación de los botones</li> <li>• Mismo estilo de redacción</li> <li>• Texto claro, describiendo cuál es el motivo del mensaje.</li> </ul> <p>Verificar que el texto de los mensajes sea claro para el usuario (evitar abreviaturas).</p>	<p>Fácil comprensión y lectura de mensajes.</p> <p>El texto describa por sí sólo el motivo del mensaje y que se debe hacer.</p>
Botones	<p>Verificar que el estilo de los botones es el mismo:</p> <ul style="list-style-type: none"> <li>• Misma letra</li> <li>• Mismo color</li> <li>• Misma posición</li> <li>• Texto corto y claro</li> </ul> <p>Verificar que no se mezcle la funcionalidad de los botones, un mismo botón no se use para más de una acción.</p> <p>Verificar solamente los botones que cada perfil de usuario puede usar.</p>	<p>Fácil comprensión y lectura de etiquetas.</p> <p>La funcionalidad de cada botón sea única.</p> <p>No se muestren botones que para determinados usuarios no tienen acceso.</p>

Tabla 4.2.3 Pruebas de interfaz de usuario para el caso de uso Dar de baja una oferta de trabajo

## Pruebas de seguridad y control de acceso

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Nombre de usuario y contraseña	Verificar que siempre el nombre de usuario y la contraseña sean codificadas.	Permite mantener de forma segura y confidencial la clave de acceso
Administrador del sistema	Verificar que sólo el Administrador pueda registrar un asesor y una Secretaria. Verificar que tenga acceso a la administración de usuarios: <ul style="list-style-type: none"> <li>• Altas</li> <li>• Prebajas</li> <li>• Bajas</li> <li>• Cambios</li> <li>• Consulta</li> </ul>	El usuario puede aplicar acciones de acuerdo a su nivel de usuario y no puede realizar ninguna otra si su nivel no lo contempla.
Asesor	Verificar que pueda tener acceso a las opciones que le corresponden. Verificar que sólo el asesor pueda acceder al formulario de llamada telefónica.	El usuario puede aplicar acciones de acuerdo a su nivel de usuario y no puede realizar ninguna otra si su nivel no lo contempla.
Secretaria	Verificar que pueda tener acceso a las opciones que le corresponden. Verificar que pueda registrar a un Egresado y modificar sus datos.	El usuario puede aplicar acciones de acuerdo a su nivel de usuario y no puede realizar ninguna otra si su nivel no lo contempla.
Empresa	Verificar que pueda tener acceso a las opciones que le corresponden. Verificar que pueda registrar ofertas. Verificar que sólo pueda ver sus propias ofertas.	El usuario puede aplicar acciones de acuerdo a su nivel de usuario y no puede realizar ninguna otra si su nivel no lo contempla.
Egresado	Verificar que pueda tener acceso a las opciones que le corresponden. Verificar que pueda registrarse como candidato a las ofertas que estén dadas de alta por el administrador solamente.	El usuario puede aplicar acciones de acuerdo a su nivel de usuario y no puede realizar ninguna otra si su nivel no lo contempla.

Tabla 4.2.4 Pruebas de seguridad y control de acceso

## Pruebas de carga de trabajo

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Acceso al sistema	15 usuarios realizan captura de: <ul style="list-style-type: none"> <li>• Registro de empresa</li> <li>• Nombre de usuario y contraseña</li> </ul>	El sistema siga operando.
Consulta	15 usuarios realizan consultas de las páginas de opciones de acuerdo a su nivel: <ul style="list-style-type: none"> <li>• Opciones del Egresado</li> <li>• Opciones de la Empresa</li> <li>• Opciones de la Secretaria</li> <li>• Opciones del Administrador</li> </ul>	El sistema siga operando.
Cambios	15 usuarios realizan cambios en las páginas de opciones de acuerdo a su nivel: <ul style="list-style-type: none"> <li>• Cambio de datos del usuario</li> <li>• Cambio de contraseña</li> </ul>	El sistema siga operando.
Registro de usuarios	15 usuarios realizan captura de: <ul style="list-style-type: none"> <li>• Nueva Empresa</li> <li>• Nueva Secretaria</li> <li>• Nuevo Asesor</li> <li>• Nuevo Egresado</li> </ul>	El sistema siga operando.
Registro de ofertas de trabajo	15 usuarios realizan captura de: <ul style="list-style-type: none"> <li>• Ofertas de trabajo</li> </ul>	El sistema siga operando.

Tabla 4.2.5 Pruebas de carga de trabajo



## Pruebas de tiempo de respuesta

Estas pruebas se realizaron con computadoras de características similares y con una conexión de red de 100 Mbps.

Tipo de prueba	Instrucciones de verificación	Resultado esperado
Carga de la pantalla principal	Con 15 usuarios el tiempo fue de 1.3 seg.	El sistema siga operando.
Carga de pantalla de opciones de usuario	Con 15 usuarios el tiempo fue 0.4 seg.	El sistema siga operando.
Carga de pantalla de cambios	Con 15 usuarios el tiempo fue 0.4 seg.	El sistema siga operando.
Carga de pantalla de consulta	Con 15 usuarios el tiempo fue 0.5 seg.	El sistema siga operando.
Salir del sistema	Con 15 usuarios el tiempo fue 0.4 seg.	El sistema siga operando.

Tabla 4.2.6 Pruebas de tiempo de respuesta

Un sistema creado podría tener errores diversos, por lo tanto con la finalidad de depurar y corregir la mayoría de éstos nos auxiliamos de diferentes tipos de pruebas: pruebas de software, pruebas de carga de trabajo, pruebas de seguridad y control de acceso, pruebas interfaz de usuario y pruebas de tiempo de respuesta.

Con las pruebas de seguridad y control de acceso fueron identificados fallos de implementación y la usabilidad del producto de software.

Las pruebas de interfaz de usuario ayudaron a ver qué tan sencillo era navegar por las pantallas del sistema y qué tan entendible era el módulo de ayuda de cada pantalla para los usuarios.

Gracias a las pruebas de carga de trabajo y de tiempo de respuesta se observó que el sistema tiene un buen desempeño con 15 personas (estimación de la media de clientes para el uso del sistema usándolo al mismo tiempo).

Los resultados arrojados por las pruebas de software permitieron conocer la calidad de nuestro producto y hacer las mejoras respectivas para poder alcanzar la calidad del producto deseado.



# Capítulo 5

## Resultados y conclusiones



## 5.1 Resultados

Con base en los requerimientos indicados al inicio del presente trabajo, fueron obtenidos los siguientes resultados:

Requerimientos	Resultados
Automatizar el proceso de inscripción de empresas e instituciones en la BT	El resultado fue satisfactorio, pues las empresas e Instituciones se inscriben en la BT de forma automatizada. Estas inscripciones son controladas por el Administrador del sistema
Automatizar el proceso de inscripción de egresados en la BT	El resultado fue bueno, pues los egresados se inscriben de forma semi-automática. Estas inscripciones son controladas por el Administrador del sistema
Proponer un sistema para automatizar y controlar la puesta en línea de ofertas de trabajo brindadas por las empresas e instituciones	El resultado fue satisfactorio, pues las empresas e instituciones ingresan sus ofertas, pero la puesta en línea es controlada por el Administrador del sistema
Agilizar los trámites que se llevan a cabo en la búsqueda de una oferta para determinado candidato (anteriormente 15 días como mínimo)	El resultado fue satisfactorio, debido a que la búsqueda de una oferta se realiza de forma ágil y en pocos segundos
Mejorar el proceso de seguimiento de las empresas o instituciones a fin de que cumplan con lo establecido en sus ofertas	El resultado fue satisfactorio, debido a que gracias a la consulta telefónica existente en el sistema, se puede hacer controlar el estatus de las empresas y su permanencia en la BT
Aliviar las cargas de trabajo dentro del departamento	El resultado fue satisfactorio, debido a que el personal no invierte tanto tiempo en atender a los egresados
Dar a conocer el departamento a empresas e instituciones con el fin de que lo identifiquen y lo tomen en cuenta	El resultado fue satisfactorio, debido a que a través de internet el sistema puede ser accedido en cualquier momento y en cualquier lugar del mundo
Construcción de una base de datos de los alumnos inscritos la Bolsa de Trabajo	Actualmente la base de datos se encuentra en crecimiento, debido a que no se contaba con una base general y actualizada de todos los egresados de la facultad

Construcción de una base de datos de las empresas e instituciones que desean contratar personal	Actualmente la base de datos se encuentra en crecimiento, debido a que no se contaba con una base general y actualizada de todas las empresas e instituciones que desean contratar personal egresado de la FP
Disponer de información que coadyuve a alcanzar los objetivos dentro del área de Bolsa de Trabajo en el momento oportuno (consultas)	El resultado fue satisfactorio, debido a que gracias a las diversas consultas diseñadas se pueden tomar decisiones clave para orientar los esfuerzos del DSSyBT
Los resultados obtenidos fueron así catalogados con base en la opinión del cliente y en las pruebas realizadas.	

Tabla 5.1.1 Resultados obtenidos por el SABT, de acuerdo a los requerimientos del DSSyBT

## 5.2 Mantenimiento del software

Una vez concluido el proceso de pruebas, debemos someter el sistema a un plan de mantenimiento, con el fin de dar mayor seguridad al cliente sobre la permanente funcionalidad del mismo, debido a que existe la posibilidad de que en un futuro, las nuevas necesidades del DSSyBT, la introducción de una nueva tecnología o el cambio en el ambiente de red, por mencionar sólo algunos ejemplos, dificulte el uso del SABT o lo haga obsoleto.

Por lo anterior, se ha propuesto un programa de mantenimiento adaptativo, preventivo y perfectivo, el cual estará presente desde el inicio de la utilización del sistema por el cliente.

En cuanto al mantenimiento adaptativo, por el momento no se ha propuesto ninguna modificación al sistema, debido a su capacidad de poder ser utilizado bajo los sistemas operativos más comerciales y novedosos sin realizar ningún cambio en el código.

Respecto al mantenimiento perfectivo, se tendrá que realizar cuando el sistema haya madurado, es decir, cuando posea de un tiempo extenso de utilización y sea conocido por los usuarios finales.

Concerniente al mantenimiento preventivo, la base de datos del SABT debe ser revisada cada mes, al menos durante los primeros seis meses, a fin de detectar de forma oportuna la tasa de crecimiento real que se tiene en cuanto a los registros introducidos en la misma y se tenga noción de cuándo será el momento oportuno para su depuración general.

### 5.3 Conclusiones

Con la realización del SABT se mejoran diversos procesos administrativos, se tiene un mayor control y una mejor organización de los datos de los egresados, rapidez y eficiencia para que los egresados puedan encontrar oportunidades laborales según sus aptitudes y requerimientos de las ofertas de trabajo.

Los sistemas de software son entes evolutivos. Es de suma importancia que un sistema cumpla con los requisitos originales para los que fue construido, aunque el frecuente uso por parte de los clientes da pauta para que se detecten nuevas características que serían deseables incorporar al sistema. El SABT representa el inicio de una aplicación web que puede ir creciendo de acuerdo a las nuevas necesidades e ideas que surjan durante la utilización y conocimiento del mismo.

UML ha sido una herramienta de desarrollo indispensable ya que con ella hemos podido visualizar el sistema desde diversos puntos de vista conceptuales, como son: objetos, clases, secuencias, colaboraciones, etc. Dado que UML es un estándar en la industria del desarrollo del software será fácil para otros desarrolladores comprender los modelos del SABT y de ser necesario, modificarlo o extenderlo de acuerdo a nuevas necesidades del Departamento de Bolsa de Trabajo y Servicio Social de la Facultad de Psicología.

El SABT se desarrolló con herramientas de software libre (PHP, MySql y Apache), pensando en el ahorro en la compra de licencias, además estas herramientas pueden instalarse en gran variedad de plataformas con distintos sistemas operativos, por lo cual el sistema resulta portable, puesto que no es necesario efectuar cambios importantes en el código fuente para poder ejecutarlo en diversos sistemas operativos.

Las diferentes pruebas aplicadas al sistema nos sirvieron para determinar los errores y poderlos corregir satisfactoriamente.

Gracias a la realización de este trabajo teórico y práctico de tesis hemos podido poner en práctica diversos conocimientos adquiridos durante nuestra estadía en la Facultad de Ingeniería, como se ha indicado a lo largo del texto.

Hemos reforzado conocimientos y aprendido numerosos conceptos más en varios temas de nuestra carrera, generando experiencia para nuestro crecimiento profesional.





# Bibliografía



## Bibliografía

Booch, G., Jacobson, I., Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*, Pearson Educación SA, Madrid España, 2000.

Forouzan, Behrouz A. *Transmisión de Datos y Telecomunicaciones*, McGraw-Hill /Interamericana de España S.A.U., Segunda Edición, España, 2002.

Gil Rubio, F., Tejedor Cerbel, J. A., Yagüe Panadero, A., Alonso Villaverde, S., Gutiérrez Rodríguez, A. *Creación de sitios web con PHP 4*, McGraw-Hill, Madrid España, 2001.

Hansen, G.W., Hansen, J.V. *Diseño y gestión de base de datos*, Prentice Hall Internacional, España, 1998.

Larman, C. *UML y Patrones, Introducción al Análisis Orientado a Objetos*, Prentice Hall, México, 1999.

Maslakowsky, M. I. *Aprendiendo MySQL en 21 días*, Prentice Hall, México, 2001.

Pressman, R. S. *Ingeniería del Software, Un enfoque práctico*, McGraw- Hill, 6ª Edición Madrid España, 2005.

Zaldívar Zamorategui, O. *Apuntes de ingeniería de programación*, Universidad Nacional Autónoma de México, Facultad de Ingeniería, México, D.F., 2004.

### Mesografía

García, Miguel Ángel. *Introducción a JavaScript .Comunidad javaHispano*, 2007 y 2008.  
<http://www.javahispano.org/tutorials.type.action?type=js>

[http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

<http://www.mastermagazine.info/termino/5454.php>