

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA EN SISTEMAS**

**Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS**

**TEMA:**

**ANÁLISIS, DISEÑO Y DESARROLLO DE UNA BOLSA DE TRABAJO EN  
AMBIENTE WEB PARA EL DEPARTAMENTO DE BIENESTAR  
ESTUDIANTIL DE LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE  
QUITO - CAMPUS EL GIRÓN.**

**AUTORES:**

**ANGELO ALEXANDER ANDY ACOSTA  
ANDRÉS RICARDO VARGAS MANTILLA**

**DIRECTOR:**

**DANIEL GEOVANY DÍAZ ORTIZ**

**Quito, julio del 2014**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE TITULACIÓN**

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, julio 2014

---

Angelo Alexander Andy Acosta

C.C: 1719099788

---

Andrés Ricardo Vargas Mantilla

C.C: 1717535874

## **AGRADECIMIENTOS**

Debemos agradecer a la Universidad Politécnica Salesiana por las facilidades que ha dado para llevar a cabo esta investigación. A nuestro tutor Ing. Daniel Díaz por haber respondido a todas nuestras dudas y consultas siempre que lo hemos necesitado. Al Departamento de Bienestar Estudiantil, especialmente a su directora Irene Lema, por su disponibilidad y apoyo durante el proceso de elaboración de este trabajo de titulación.

**Andrés Vargas Mantilla & Angelo Andy Acosta.**

## **DEDICATORIA**

La concepción de este proyecto está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos y familia en general.

A mis sobrinos que sin duda alguna son la alegría del hogar y simplemente con sus sonrisas borran todas las penas: Micaela, Sebastián, Fernanda y Camila.

**Andrés Vargas Mantilla.**

La culminación de este proyecto está dedicada a mis padres, quienes han sido todo en mi vida los que me apoyaron en todas las decisiones que tome, hasta conseguir lo más importante que es mi graduación. A mi madre que ha sido mi mayor ejemplo como persona a su perseverancia y lucha por hacernos a mis hermanos y a mí, personas de bien, a mi padre ejemplo de hombre trabajador, que ha estado siempre respaldándome en todo momento y lugar.

A mi esposa ejemplo de hermana, madre y mujer, gracias a ella por estar siempre apoyándome en los buenos y malos momentos, a mi querida hija que ha sido la alegría de mi vida y por quien seguiré luchando y saliendo adelante en esta vida.

**Angelo Andy Acosta.**

# ÍNDICE

|  |    |
|--|----|
| <b>INTRODUCCIÓN</b> .....  | 1  |
| <b>CAPÍTULO 1</b> .....  | 3  |
| <b>PRELIMINARES</b> .....  | 3  |
| 1.1.Objetivo general .....   | 3  |
| 1.2.Objetivos específicos .....  | 3  |
| 1.3.Justificación.....   | 3  |
| 1.4.Alcance.....   | 4  |
| 1.4.1. Perfiles. ....  | 5  |
| 1.4.2. Módulos. ....   | 6  |
| <b>CAPÍTULO 2</b> .....  | 9  |
| <b>MARCO TEÓRICO Y CONCEPTUAL</b> .....                                      | 9  |
| 2.1.Definiciones generales .....   | 9  |
| 2.1.1. Bolsa de trabajo.....   | 9  |
| 2.1.1.1. Tipos.....  | 9  |
| 2.1.1.2. Bolsas de trabajo en el mercado ecuatoriano. ....                   | 10 |
| 2.1.1.3. Análisis del impacto de las bolsas de trabajo en el Ecuador.....    | 13 |
| 2.1.1.4. Criterios generales sobre la gestión de las bolsas de trabajo. .... | 16 |
| 2.1.2. Hoja de vida o Curriculum Vitae. ....                                 | 16 |
| 2.1.2.1. Definición.....   | 16 |
| 2.1.2.2. Elementos.....  | 16 |
| 2.1.2.3. Tipos.....  | 17 |
| 2.1.3. Entornos y herramientas.....  | 18 |
| 2.1.3.1. Introducción. ....  | 18 |
| 2.1.3.2. Herramienta.....  | 19 |
| 2.1.3.3. Desarrollo de software. ....  | 19 |
| 2.1.3.4. Entorno de desarrollo. ....   | 20 |

|   |    |
|---|----|
| 2.1.3.5. Base de datos.....                               | 20 |
| 2.1.3.6. Servidor web. ....                               | 20 |
| 2.1.3.7. Herramienta CASE. ....                           | 21 |
| 2.1.4. Requerimiento funcional.....                       | 21 |
| 2.2.Lenguaje Unificado de Modelado (UML) .....            | 22 |
| 2.2.1. Definición.....                                    | 22 |
| 2.2.2. Diagramas de casos de uso.....                     | 22 |
| 2.2.2.1. Definición.....                                  | 22 |
| 2.2.2.2. Elementos básicos.....                           | 23 |
| 2.2.3. Diagrama de clases.....                            | 23 |
| 2.2.3.1. Definición.....                                  | 23 |
| 2.2.3.2. Elementos básicos.....                           | 23 |
| 2.2.4. Diagrama de estados. ....                          | 24 |
| 2.2.4.1. Definición.....                                  | 24 |
| 2.2.4.2. Elementos básicos.....                           | 24 |
| 2.3.Metodología Técnica de Modelado de Objetos (OMT)..... | 25 |
| 2.3.1. Introducción. ....                                 | 25 |
| 2.3.2. Modelos.....                                       | 25 |
| 2.3.3. Fases.....   | 26 |
| 2.3.3.1. Conceptualización.....                           | 27 |
| 2.3.3.2. Análisis.....                                    | 28 |
| 2.3.3.3. Diseño del sistema.....                          | 28 |
| 2.3.3.4. Diseño de objetos.....                           | 29 |
| 2.4.Herramientas de desarrollo .....                      | 29 |
| 2.4.1. Netbeans IDE. ....                                 | 29 |
| 2.4.1.1. Java Server Faces (JSF). ....                    | 30 |
| 2.4.1.2. Primefaces.....                                  | 30 |

|  |           |
|--|-----------|
| 2.4.1.3. Java Persistence Api (JPA) .....                      | 31        |
| 2.4.2. Base de datos oracle.....                               | 31        |
| 2.4.3. Apache Tomcat. ....                                     | 32        |
| 2.4.4. Sybase power designer.....                              | 32        |
| <b>CAPÍTULO 3.....</b>   | <b>34</b> |
| <b>REQUERIMIENTOS FUNCIONALES Y MODELAMIENTO BAJO OMT ..</b>   | <b>34</b> |
| 3.1.Análisis de requerimientos funcionales .....               | 34        |
| 3.1.1. Identificación de actores y escenarios del sistema..... | 34        |
| 3.2.Modelamiento OMT.....                                      | 36        |
| 3.2.1. Conceptualización. ....                                 | 36        |
| 3.2.2. Modelo de análisis del sistema.....                     | 42        |
| 3.2.2.1. Modelo de objetos.....                                | 42        |
| 3.2.2.2. Modelo dinámico. ....                                 | 52        |
| 3.2.2.3. Modelo funcional .....                                | 56        |
| <b>CAPÍTULO 4.....</b>   | <b>62</b> |
| <b>DESARROLLO DEL PORTAL WEB Y PRUEBAS .....</b>               | <b>62</b> |
| 4.1.Programación de módulos.....                               | 62        |
| 4.1.1. Método login. ....                                      | 62        |
| 4.1.2. Método para la creación de menú de usuario.....         | 63        |
| 4.1.3. Método dar de alta a un usuario. ....                   | 64        |
| 4.1.4. Método creación de anuncios.....                        | 64        |
| 4.1.5. Método login empresa registrada. ....                   | 65        |
| 4.1.6. Método recuperar contraseña. ....                       | 65        |
| 4.1.7. Método generar reporte empresas vs ofertas.....         | 66        |
| 4.1.8. Método crear nueva oferta laboral. ....                 | 67        |
| 4.1.9. Método búsqueda de postulantes (hojas de vida).....     | 67        |
| 4.1.10. Método crear sugerencias.....                          | 68        |

|  |    |
|--|----|
| 4.2.Pruebas de software.....                           | 69 |
| 4.2.1. Pruebas de caja negra. ....                     | 69 |
| 4.2.1.1. Caso de prueba ingreso al sistema. ....       | 70 |
| 4.2.1.2. Caso de prueba cambiar contraseña. ....       | 71 |
| 4.2.1.3. Caso de prueba recuperar contraseña.....      | 72 |
| 4.2.1.4. Caso de prueba validar menú por usuario. .... | 73 |
| 4.2.2. Pruebas de carga y estrés. ....                 | 74 |
| 4.2.2.1. Primera fase de pruebas. ....                 | 74 |
| 4.2.2.2. Segunda fase de pruebas. ....                 | 77 |
| <b>CONCLUSIONES</b> .....                              | 81 |
| <b>RECOMENDACIONES</b> .....                           | 82 |
| <b>LISTA DE REFERENCIAS</b> .....                      | 84 |
| <b>ANEXOS:</b> En digital.                             |    |

## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1. Módulos del sistema .....                              | 6  |
| Figura 2. Fases de microprocesos de análisis de OMT .....        | 26 |
| Figura 3. Fases de OMT .....                                     | 27 |
| Figura 4. Actores del sistema .....                              | 34 |
| Figura 5. Diagrama caso de uso autenticar .....                  | 36 |
| Figura 6. Diagrama caso de uso cambiar contraseña .....          | 37 |
| Figura 7. Diagrama caso de uso recuperar contraseña .....        | 38 |
| Figura 8. Diagrama caso de uso usuario administrador .....       | 39 |
| Figura 9. Diagrama caso de uso usuario postulante .....          | 40 |
| Figura 10. Diagrama caso de uso usuario empresa .....            | 41 |
| Figura 11. Diagrama caso de uso usuario reportes .....           | 42 |
| Figura 12. Diagrama modelo de la base de datos .....             | 43 |
| Figura 13. Diagrama de clases .....                              | 44 |
| Figura 14. Diagrama de estado ingreso al sistema .....           | 52 |
| Figura 15. Diagrama de estado cambiar contraseña .....           | 53 |
| Figura 16. Diagrama de estado recuperar contraseña .....         | 53 |
| Figura 17. Diagrama de secuencia ingreso al sistema .....        | 54 |
| Figura 18. Diagrama de secuencia cambiar contraseña .....        | 55 |
| Figura 19. Diagrama de secuencia recuperar contraseña .....      | 55 |
| Figura 20. Diagrama de flujo de datos ingresar al sistema .....  | 56 |
| Figura 21. Diagrama de flujo de datos cambiar contraseña .....   | 57 |
| Figura 22. Diagrama de flujo de datos recuperar contraseña ..... | 57 |
| Figura 23: Diagrama de flujo de datos administrador .....        | 58 |
| Figura 24. Diagrama de flujo de datos postulante .....           | 59 |
| Figura 25. Diagrama de flujo de datos empresa .....              | 60 |
| Figura 26. Diagrama de flujo de datos usuario reportes .....     | 61 |
| Figura 27. Código fuente método login .....                      | 62 |
| Figura 28. Código fuente crear menú .....                        | 63 |
| Figura 29. Código fuente dar de alta usuario .....               | 64 |
| Figura 30. Código fuente nuevo anuncio .....                     | 64 |
| Figura 31. Código fuente login empresa registrada .....          | 65 |
| Figura 32. Código fuente recuperar contraseña .....              | 65 |

|   |    |
|---|----|
| Figura 33. Código fuente enviar mail .....                          | 66 |
| Figura 34. Código fuente generar reporte .....                      | 66 |
| Figura 35. Código fuente nueva oferta laboral.....                  | 67 |
| Figura 36. Código fuente buscar hojas de vida .....                 | 67 |
| Figura 37. Código fuente nueva sugerencia .....                     | 68 |
| Figura 38. Uso de memoria del servidor (Primera fase) .....         | 75 |
| Figura 39. Peticiones realizadas (Primera fase) .....               | 75 |
| Figura 40. Peticiones HTTP correctas (Primera fase).....            | 76 |
| Figura 41. Peticiones HTTP incorrectas (Primera fase).....          | 77 |
| Figura 42. Uso de memoria (Segunda fase) .....                      | 78 |
| Figura 43. Peticiones realizadas (Segunda fase) .....               | 79 |
| Figura 44. Peticiones realizadas correctamente (Segunda fase) ..... | 79 |
| Figura 45. Resultado peticiones http request y java request .....   | 80 |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| Tabla 1. Bolsas de trabajo virtuales en el Ecuador .....               | 11 |
| Tabla 2. Análisis del impacto de las bolsas de trabajo virtuales ..... | 13 |
| Tabla 3. Especificación caso de uso: Autenticar .....                  | 36 |
| Tabla 4. Especificación caso de uso: Cambiar contraseña.....           | 37 |
| Tabla 5. Especificación caso de uso: Recuperar contraseña .....        | 38 |
| Tabla 6. Especificación del diccionario de clases .....                | 45 |
| Tabla 7. Especificación clase persona.....                             | 46 |
| Tabla 8. Especificación clase empresa.....                             | 47 |
| Tabla 9. Especificación clase postulante.....                          | 47 |
| Tabla 10. Especificación clase ofertas .....                           | 48 |
| Tabla 11. Especificación clase anuncios .....                          | 48 |
| Tabla 12. Especificación clase sugerencias .....                       | 49 |
| Tabla 13. Especificación clase hoja de vida.....                       | 49 |
| Tabla 14. Especificación clase idiomas .....                           | 49 |
| Tabla 15. Especificación clase conocimientos.....                      | 50 |
| Tabla 16. Especificación clase informática.....                        | 50 |
| Tabla 17. Especificación clase experiencia.....                        | 50 |
| Tabla 18. Especificación clase referencias.....                        | 51 |
| Tabla 19. Requerimiento de prueba: Ingreso al sistema .....            | 70 |
| Tabla 20. Caso de prueba: Ingreso al sistema .....                     | 70 |
| Tabla 21. Requerimiento de prueba: Cambiar contraseña .....            | 71 |
| Tabla 22. Caso de prueba: Cambiar contraseña .....                     | 71 |
| Tabla 23. Requerimiento de prueba: Recuperar contraseña.....           | 72 |
| Tabla 24. Caso de prueba: Recuperar contraseña .....                   | 72 |
| Tabla 25. Requerimiento de prueba: Validar menú por usuario .....      | 73 |
| Tabla 26. Caso de prueba: Visualizar menú correcto por usuario .....   | 73 |

## **RESUMEN**

En el documento se analizan los principales inconvenientes que tiene el Departamento de Bienestar Estudiantil de la Universidad Politécnica Salesiana, sede Quito, campus “El Girón”, en cuanto al manejo de los Curriculum Vitae de estudiantes y la publicación de las ofertas laborales por parte de las empresas, una vez analizados los inconvenientes y principales requisitos se propone desarrollar un sistema web que permita automatizar y agilizar las tareas de dicho departamento.

Para el desarrollo del sistema es necesario comprender varios conceptos, es por esta razón que se abordará las definiciones básicas para el análisis y creación de un sistema web, de igual manera los conceptos generales sobre los Curriculum Vitae y las bolsas de empleo, en estas últimas se realiza también un análisis de impacto que tienen las bolsas de empleo virtuales en la actualidad.

Se indica también los esquemas que representan a cada uno de los usuarios del sistema con sus respectivas tareas y roles dentro del mismo, permitiendo de esta manera la fácil comprensión y a su vez aportando la información necesaria para su posterior manejo, todo esto mediante el uso de la metodología OMT.

Se incluye los fragmentos de código fuente de los métodos más importantes con su respectiva descripción para tener un ejemplo claro de cómo se lleva a cabo la programación de los módulos del sistema. También se realiza pruebas de rendimiento mediante el uso de software especializado el cual permite, corregir posibles errores y a su vez medir la robustez del sistema en un ambiente real.

## **ABSTRACT**

In this paper the main drawbacks of the “Departamento de Bienestar Estudiantil”, Quito headquarters campus "El Girón" are analyzed in terms of managing the Curriculum Vitae of students and the publication or dissemination of the job offers by the companies, once the main drawbacks and discussed requirements aims to develop a web system that automates tasks and expedite the department.

For the development of the system is necessary to understand and be clear about several concepts, it is for this reason that the basic and necessary definitions for analysis and creation of a web system, just as it will address the general concepts of the Curriculum Vitae and bags employment, the latter is also performed an analysis of impact bags virtual job today.

Schemes representing each of the system users with their tasks and roles within it, thus allowing easy understanding and in turn providing the necessary information for further management is also indicated, all using the OMT methodology.

Fragments of source code the most important methods with description for a clear example of how to carry out the programming of the system modules are included. Performance tests are also performed by using specialized software which allows correct errors and in turn measure the robustness of the system in a real environment, all these data can then display and compare with each other, with different stages performed.

## INTRODUCCIÓN

La Universidad Politécnica Salesiana es una universidad ecuatoriana perteneciente a la Congregación Salesiana, fue fundada en el año de 1994 en la ciudad de Cuenca y tiene sedes en Cuenca, Quito y Guayaquil, se caracteriza principalmente por la educación técnica y cristiana que imparte a sus estudiantes.

La sede Quito cuenta actualmente con tres campus: Sur, El Girón y Kennedy. El proyecto se centrará en el campus Girón ubicado en la Av. 12 de Octubre 2422 y Wilson, y en su Departamento de Bienestar Estudiantil.

El Departamento de Bienestar Estudiantil tiene como objetivo promover actividades, proyectos, programas, políticas y normativas específicas que favorezcan el desarrollo y bienestar de la población estudiantil de la Universidad Politécnica Salesiana. Es así que el Departamento de Bienestar Estudiantil decide centrar su atención en los estudiantes de los niveles superiores, específicamente apoyándolos en la búsqueda de pasantías laborales y fuentes de trabajo.

Una de las principales dificultades para los estudiantes que culminan la etapa universitaria es conseguir información acerca de vacantes o puestos de trabajo de acuerdo a su perfil académico, por este motivo el profesional debe perder tiempo, dinero y en ciertas ocasiones arriesgar su integridad física en busca de un puesto laboral.

Además, los estudiantes de niveles intermedios deben iniciar la búsqueda de sus pasantías laborales, y al ser esta una actividad nueva en su vida estudiantil no cuentan con la orientación e información necesaria para comenzar la búsqueda de una empresa que brinde esta facilidad.

Otra dificultad es el manejo de la información, los profesionales tienen que imprimir varias hojas de vida para los diferentes puestos laborales, esto conlleva problemas como: desperdicio de hojas y pérdida de documentos.

Por otro lado el Departamento de Bienestar Estudiantil campus El Girón, realiza todas sus tareas de forma manual, dentro de las cuales se encuentran:

- Recepción de hojas de vida de estudiantes y egresados.
- Recolección de ofertas por parte de las empresas.
- Comparación y verificación de requisitos.
- Selección de postulados.
- Concretar citas entre las empresas y los estudiantes o egresados.

El resultado de realizar estas tareas manualmente ha hecho que el departamento tenga las siguientes dificultades:

- Pérdida de información; al no tener una base de datos específica para esta tarea.
- Pérdida de tiempo; en la clasificación de hojas de vida por facultad, pasante, egresado, edad y género.

# **CAPÍTULO 1**

## **PRELIMINARES**

### **1.1. Objetivo general**

Analizar, diseñar y desarrollar una aplicación para gestionar una bolsa de trabajo en ambiente web para el Departamento de Bienestar Estudiantil de la Universidad Politécnica Salesiana sede Quito Campus El Girón.

### **1.2. Objetivos específicos**

1. Analizar requerimientos y necesidades del Departamento de Bienestar Estudiantil de la Universidad Politécnica Salesiana sede Quito campus El Girón, para determinar las principales tareas, roles y usuarios del sistema web.
2. Analizar y diseñar los diagramas necesarios para el sistema informático, en base a la metodología OMT.
3. Diseñar el modelo de la base de datos para el sistema web, basándose en los diagramas obtenidos al realizar los procesos de la metodología OMT, la cual tendrá en su diseño la información de: alumnos, empresas, ofertas laborales y la parte administrativa del sistema web.
4. Automatizar los procesos administrativos de selección de postulantes para vacantes u ofertas de trabajo para agilizar el trabajo de la persona que lo administra.
5. Realizar diferentes tipos de pruebas para verificar el funcionamiento y robustez del sistema.

### **1.3. Justificación**

La búsqueda de vacantes para realizar las pasantías de los alumnos de niveles intermedios y la falta de ofertas laborales para los alumnos graduados o egresados, ha provocado incertidumbre en quienes llegan a culminar las distintas carreras universitarias, motivo por el cual se decide desarrollar un sistema con arquitectura web que permitirá de forma ágil, segura y eficiente, ayudar en la búsqueda de una pasantía o puesto de trabajo en las empresas que se encuentren en contacto con la Universidad Politécnica Salesiana.

Con este proyecto se pretende diseñar una bolsa de trabajo, orientado a la web, para los estudiantes de la Universidad Politécnica Salesiana sede Quito, el cual permitirá gestionar y optimizar las ofertas que se generen a diario y guardarlos en un orden cronológico, esto ayudará en la búsqueda, organización y actualización de esta información.

Mediante el sistema se facilitará a los estudiantes la búsqueda de vacantes laborales para realizar sus pasantías estudiantiles y de ser factible una posición estable de trabajo. También proporcionará a las empresas un sitio en Internet, en el cual se pueda buscar personal mediante un filtrado de datos tales como: sueldo, tipo de empleo (por horas, tiempo completo, medio tiempo), título o carrera universitaria, entre otras, y de esta manera encontrar un perfil laboral acorde a sus necesidades.

El sistema también automatizará la mayor parte de los procesos que realiza el personal de Bienestar Estudiantil. Además, se podrán realizar reportes para presentar datos estadísticos de interés para el administrador del sistema.

Al mantener toda la información organizada de forma digital permitirá optimizar tiempo y recursos sin afectar los procesos que realizan cada uno de los departamentos de la Universidad. También permitirá viabilizar la comunicación entre la empresa que oferta la vacante y el estudiante, egresado o graduado de la Universidad.

La automatización de los procesos de administración de la Bolsa de Trabajo para el Departamento de Bienestar Estudiantil es fundamental, para el mejor manejo de la información sobre las empresas que ofertan pasantías o puestos de trabajo fijos en las diferentes áreas que posee.

El manejo de la información de forma automática ayudará al personal encargado de administrar estos procesos a tener una mayor fluidez en su trabajo y poder obtener información de las empresas y de los posibles postulantes mucho más rápido y seguro.

#### **1.4. Alcance**

El proyecto se desarrollará en base al análisis y recolección de los requerimientos específicos de cada uno de los usuarios, para esto se realizará una entrevista directa con el personal del Departamento de Bienestar Estudiantil de la Universidad

Politécnica Salesiana Campus El Girón, lo cual permitirá obtener la mayor cantidad de requisitos, necesarios para proceder con la construcción del sistema.

Para la construcción del sistema se tomará en cuenta varias tecnologías existentes como es el caso de Java Server Faces (JSF) la cual es una herramienta diseñada para el lenguaje de programación Java, y permitirá ofrecer un sistema de calidad y agradable para los usuarios del sistema.

Además, de una base de datos diseñada en Oracle, la cual tendrá toda la información necesaria acerca del sistema y los datos para el proceso dinámico del portal web, proporcionando así la información necesaria para el administrador.

El sistema también permitirá realizar reportes estadísticos previamente definidos por filtros de datos tales como: periodos de fechas, facultad o carrera universitaria, entre otros, con la opción de exportar en formato PDF o de imprimirlos para su posterior uso.

Una vez realizada la construcción del proyecto se continuará con la fase de pruebas del sistema, para esto se implementará un servidor web Apache Tomcat, ya que es una herramienta muy robusta y ofrece gran soporte para nuevas tecnologías.

#### **1.4.1. Perfiles.**

El sistema web contará con los siguientes perfiles:

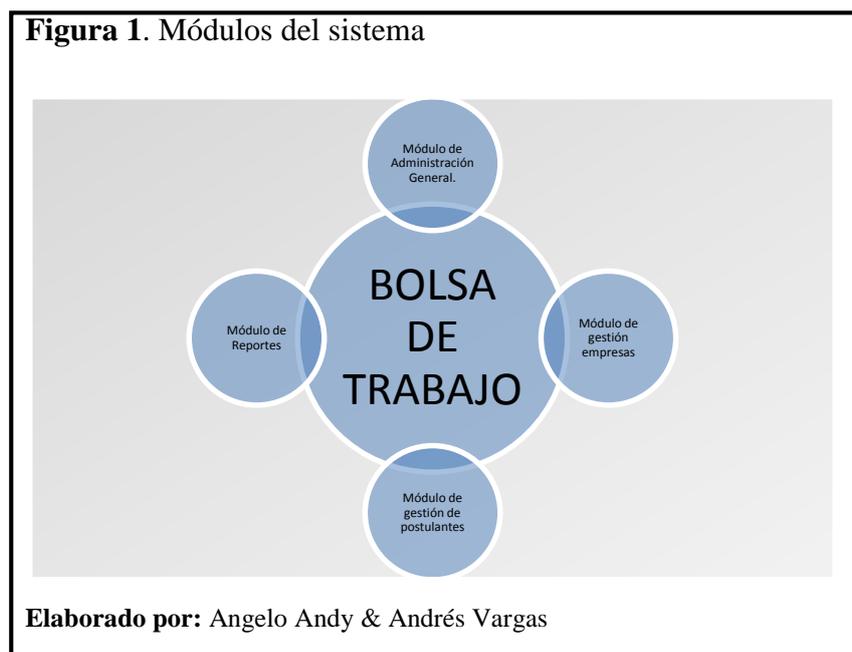
- **Administrador:** Es la persona encargada de gestionar las cuentas de usuarios es decir, dar de alta o baja una cuenta, aceptar solicitudes de registro por parte de las empresas y usuarios de tipo Reportes, crear nuevos usuarios administradores, crear diferentes reportes estadísticos, gestionar la publicidad de la página principal mediante comunicados y banners de información la cual será vista por los usuarios del sistema.

Es de vital importancia recalcar que para dar de alta a un usuario primero se debe realizar la verificación y validez de la empresa o persona que envié el formulario de registro, siendo esta tarea, única responsabilidad del administrador(Bienestar Estudiantil), ya que el sistema no posee ningún módulo o función para realizar esta validación.

- **Empresas:** La empresa tendrá la opción de registrarse y crear una nueva cuenta en el sistema de bolsa de trabajo, de esta manera puede publicar sus ofertas laborales o pasantías, para que los estudiantes, egresados o graduados apliquen de forma directa a la vacante. La empresa tendrá acceso a revisar las hojas de vida de los postulantes y acordar una cita para una entrevista personal.
- **Postulantes (Estudiantes, Egresados y Graduados):** Tendrán la facilidad de modificar sus cuentas en el sistema, para ingresar sus datos personales, fotografía, datos estudiantiles (cursos, seminarios, etc.), datos de trabajos anteriores e información personal. El postulante podrá revisar las diferentes ofertas realizadas por una empresa, si la oferta es de su interés podrá aplicar y enviar su hoja de vida. También cuenta con un historial en el cual podrá revisar todas sus postulaciones.
- **Reportes:** Podrá registrarse en el sistema mediante el formulario de registro y después de ser dado de alta por parte del administrador, tendrá acceso a la generación de los reportes estadísticos del sistema y exportarlos en formato PDF.

#### 1.4.2. Módulos.

Además, el sistema contará con varios módulos los cuales proporcionarán la funcionalidad necesaria para el usuario, los módulos son los siguientes:



**Módulo de administración general:** Este módulo permitirá realizar las siguientes tareas:

- Publicar anuncios, consejos y publicidad para la página principal.
- Administrar los datos personales de la cuenta.
- Gestionar (crear, modificar y eliminar) las cuentas de los usuarios registrados en el sistema ya sea de las empresas, postulantes y reportes.
- Generar reportes estadísticos.
- Buscar hojas de vida mediante filtro de datos.
- Revisar sugerencias y comentarios de empresas o postulantes.
- Revisar las ofertas laborales publicadas por las empresas.

**Módulo de gestión de postulantes:** Este módulo permitirá realizar las siguientes tareas:

- Gestionar los respectivos formularios de información de su hoja de vida.
- Revisar o aplicar a las distintas ofertas de trabajo publicadas por las empresas.
- Revisar historial de postulaciones.
- Administrar los datos personales de la cuenta.
- Exportar su hoja de vida en formato pdf.
- Enviar sugerencias o comentarios al administrador del sistema.

**Módulo de gestión de empresas:** Este módulo permitirá realizar las siguientes tareas:

- Editar los respectivos formularios de información para registrarse al sistema.
- Publicar ofertas de trabajo ya sea para pasantías, trabajo por horas, tiempo completo y medio tiempo.
- Búsqueda de hojas de vida mediante filtrado de datos tales como: sueldo, tipo de empleo (por horas, tiempo completo, medio tiempo), título o carrera universitaria, entre otras.
- Visualizar el listado de postulantes que aplicaron a una oferta laboral.
- Revisar hojas de vida de los postulantes.
- Exportar la hoja de vida de postulantes en formato pdf.

- Enviar sugerencia o comentario al administrador del sistema.

**Módulo de gestión de reportes:** Este módulo permitirá obtener reportes estadísticos prediseñados, los cuales pueden ser: por el número de ofertas de cada empresa en un periodo de fechas, demanda de facultades o carreras universitarias en un periodo de fechas, usuarios registrados en el sistema por roles (Estudiantes, Empresas, Reportes), número de usuarios registrados en un tiempo determinado, así la Universidad podrá contar con la información verídica y actualizada para su posterior uso.

## CAPÍTULO 2

### MARCO TEÓRICO Y CONCEPTUAL

#### 2.1. Definiciones generales

##### 2.1.1. Bolsa de trabajo.

Es un sitio donde se ordenan las diferentes ofertas de trabajo para que el ofertante como el posible postulante pueda encontrar de manera más ágil, rápida y organizada la información que sea de su necesidad.

En la actualidad, el constante desarrollo de nuevas tecnologías ha hecho que la persona que está en busca de una plaza de trabajo, ya no tenga que acudir personalmente al lugar en donde se oferta una vacante con el objetivo de entregar su hoja de vida.

En la mayoría de empresas, organizaciones, instituciones, fundaciones, universidades e industrias, entre otras, poseen una opción para que la persona que visita su sitio web tenga la oportunidad de cargar su hoja de vida a la base de datos de dicho portal, inclusive, existen portales web de todos los países dedicados específicamente a la difusión de ofertas laborales ya sea por temáticas, habilidades, compañía, cargo y región donde la persona desea y puede trabajar. (deGerencia, 2001)

Las bolsas de trabajo virtuales se presentan como una ventaja para aquellas personas que no tienen tiempo o no saben elaborar un diseño de currículum que sea presentable a la vista de los demás, ya que la mayoría de estos portales posee un tipo de formato donde el visitante sólo tiene que completar un formulario con sus datos más relevantes como: información básica, preferencias, foto, estudios, conocimientos, experiencias y expectativas, entre otros aspectos necesarios para la selección del postulante para la vacante ofertada por la empresa. (deGerencia, 2001)

##### 2.1.1.1. Tipos.

En las Bolsas de trabajo se puede distinguir tres posibilidades de uso:

- **Bolsas de trabajo generales:** son accesibles a toda persona, y las empresas pueden llegar a ofertar vacantes de trabajo de todo tipo.

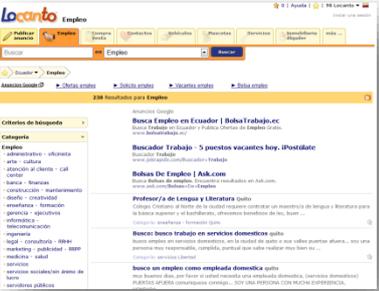
- **Bolsas de trabajo específicas:** son portales o sitios web de empleo, que solo gestionan ofertas de trabajo para sectores de actividad concretos. Se incluyen también en este tipo, los canales de empleo, o webs de determinadas empresas ya sean de gran tamaño o multinacionales que posibilitan que aquellas personas interesadas y que se ajusten a determinados perfiles profesionales puedan introducir su Curriculum Vitae (CV) para acceder o postularse a las distintas ofertas de trabajo que ofrece la empresa. (pisocompartidogranada.blogspot, 2012)
- **Buscadores de ofertas:** Consiste en sitios o portales web que no son especializados en tareas de intermediación sino que tan solo utilizan su propio motor de búsqueda para localizar ofertas laborales publicadas en Internet. El funcionamiento de estos motores es como el de cualquier buscador, de manera que el usuario puede señalar uno o varios términos de búsqueda para que el motor de esta web pueda rastrear cualquier oferta publicada en la red con los criterios de búsqueda que utilizó el usuario. (pisocompartidogranada.blogspot, 2012)

#### *2.1.1.2. Bolsas de trabajo en el mercado ecuatoriano.*

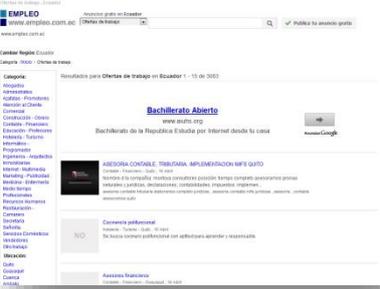
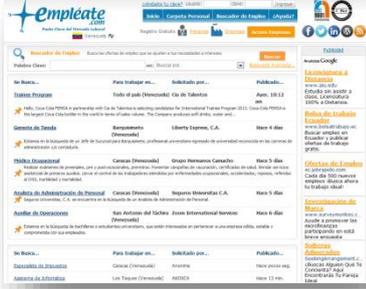
Gracias a los portales web que funcionan como bolsas de trabajo, es posible generar una gran variedad de potenciales candidatos para las vacantes, sin necesidad de hacer una inversión para realizar el proceso de forma manual. El costo de búsqueda virtual de los candidatos se reduce considerablemente, por lo que frente a los medios escritos como el periódico u otras formas de buscar candidatos para las vacantes, es previsible que éste acabe por imponerse.

En nuestro país estos sistemas han tenido una exitosa acogida existiendo una gran variedad dentro del mercado, aquí se cita las principales bolsas de trabajo virtuales del Ecuador.

**Tabla 1.** Bolsas de trabajo virtuales en el Ecuador

| <b><u>BOLSAS DE TRABAJO GENERALES</u></b>   |   |
|---|---|
| <p><b><u>CompuTrabajo</u></b></p>    | <p><b><u>Multitrabajos</u></b></p>    |
| <p><b><u>Características:</u></b> Bolsa de empleo en general.</p>   | <p><b><u>Características:</u></b> Bolsa de empleo en general.</p>   |
| <p><b><u>Postulantes:</u></b> Ingreso gratuito de CV.</p>   | <p><b><u>Postulantes:</u></b> Ingreso gratuito de CV.</p>   |
| <p><b><u>Empleadores avisos:</u></b><br/> <b>Gratis:</b> Publicar avisos es gratis.<br/> <b>Pagos:</b> Avisos destacados y acceso ilimitado a la base de datos desde 100 USD (por 3 meses).</p> | <p><b><u>Empleadores avisos:</u></b><br/> Publicar avisos es gratis.</p>  |
| <p><b><u>PorfinEmpleo</u></b></p>    | <p><b><u>Locanto Empleo</u></b></p>                                       |
| <p><b><u>Características:</u></b> Bolsa de empleo en general.</p>   | <p><b><u>Características:</u></b> Ofertas de trabajo para las distintas ciudades del país, agrupadas en diversas subcategorías que facilitan la búsqueda.</p> |
| <p><b><u>Postulantes:</u></b> Ingreso gratuito de CV.</p>   | <p><b><u>Postulantes:</u></b> Ingreso gratuito de CV.</p>   |
| <p><b><u>Empleadores avisos:</u></b><br/> Publicar avisos es gratis.</p>  | <p><b><u>Empleadores avisos:</u></b><br/> Añadir un aviso clasificado de empleo es gratis y no requiere registro.</p>   |

**Tabla 1. Bolsas de trabajo virtuales en el Ecuador (continuación).**

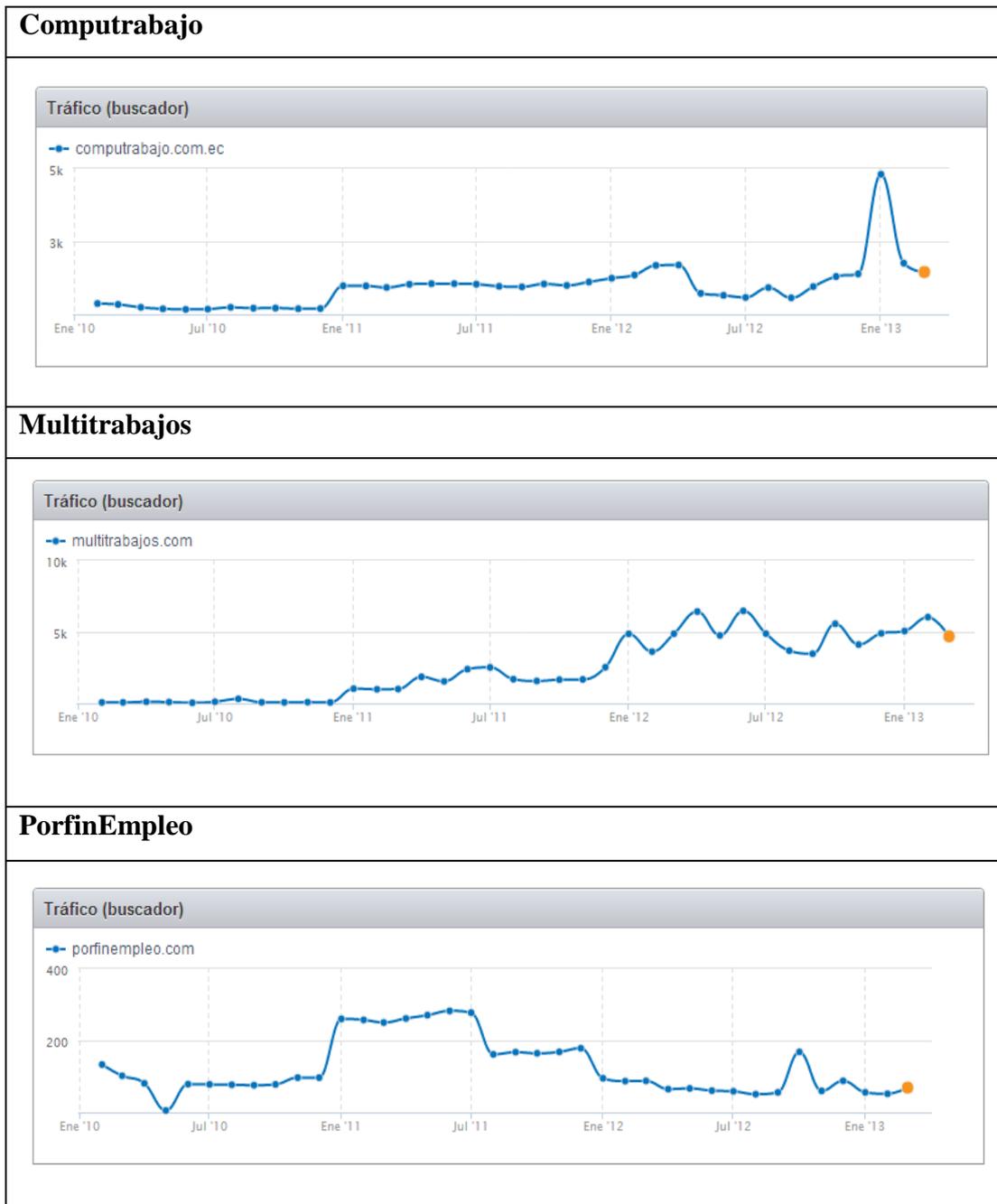
|  |   |
|--|---|
| <p style="text-align: center;"><b>Empleo</b></p>            | <p style="text-align: center;"><b>Empleate</b></p>          |
| <p><b>Características:</b> Bolsa de empleo en general.</p>   | <p><b>Características:</b> Ofertas para trabajos en el exterior.</p>  |
| <p><b>Postulantes:</b> Ingreso gratuito de CV.</p>   | <p><b>Postulantes:</b> Ingreso gratuito de CV.</p>  |
| <p><b>Empleadores avisos:</b><br/>Publicar avisos es gratis.</p>   | <p><b>Empleadores avisos:</b><br/>Publicar avisos es gratis. (Por un mes).</p>  |
| <p style="text-align: center;"><b>Acción Trabajo</b></p>  | <p style="text-align: center;"><b>Info Centro 24</b></p>  |
| <p><b>Características:</b> Bolsa de empleo en general. Última tecnología en e-recruitment.</p>   | <p><b>Características:</b> Publica listados de ofertas y pedidos recientes, al estilo de clasificados.</p>                                    |
| <p><b>Postulantes:</b> Ingreso gratuito de CV.</p>   | <p><b>Postulantes:</b> Ingreso gratuito de CV.</p>  |
| <p><b>Empleadores avisos:</b><br/><b>Gratis:</b> Son menos visibles.<br/><b>Pagos:</b> Desde 39.95 USD (por 30 días)</p>                     | <p><b>Empleadores avisos:</b><br/>Publicar avisos es gratis.</p>  |

Elaborado por: Angelo Andy & Andrés Vargas

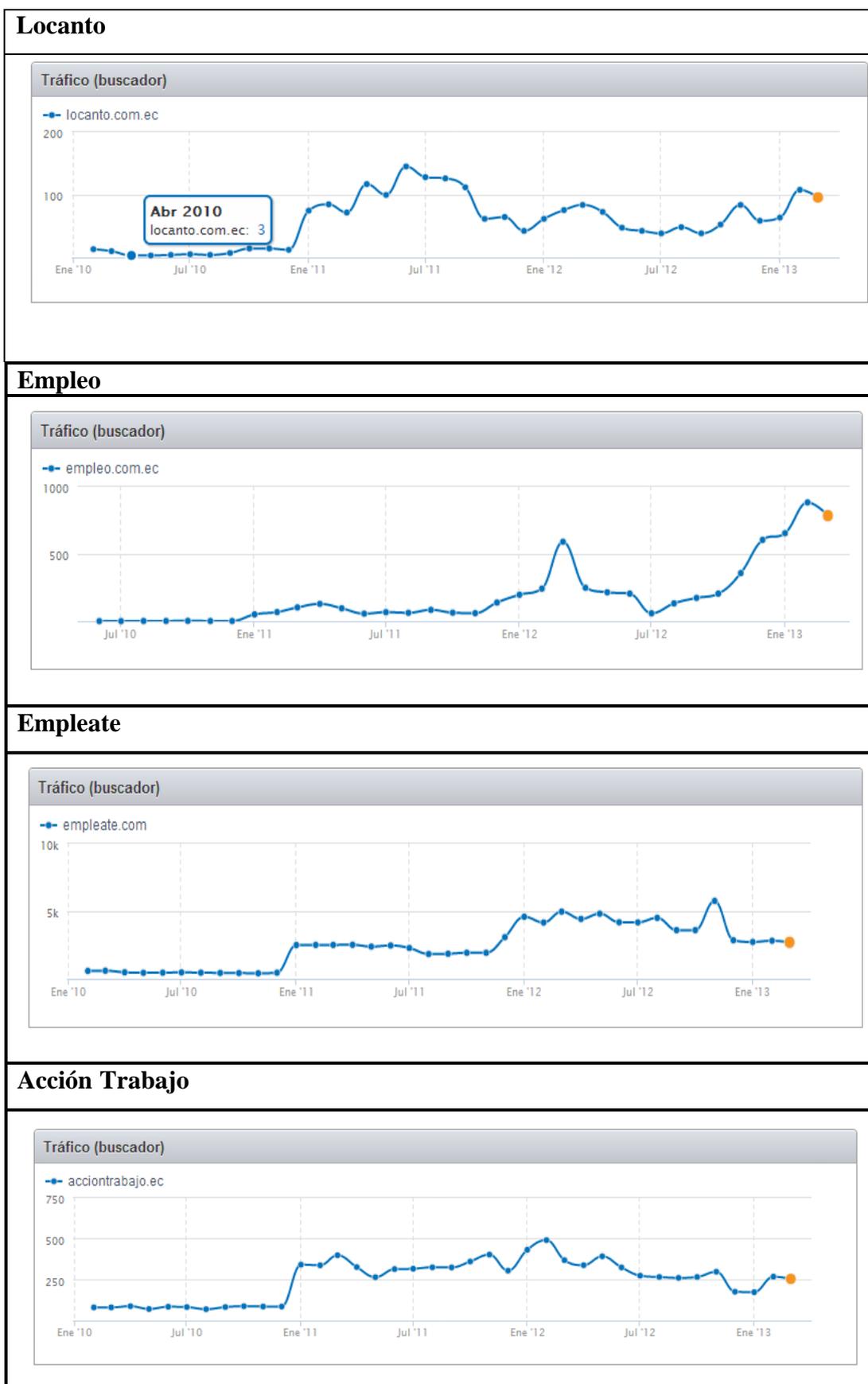
### 2.1.1.3. Análisis del impacto de las bolsas de trabajo en el Ecuador.

Mediante el uso de la tecnología se ha logrado realizar un análisis del impacto y la aceptación que ha tenido cada una de las bolsas de trabajo virtuales anteriormente detalladas, el resultado estadístico de este análisis se lo detalla en la tabla 2.

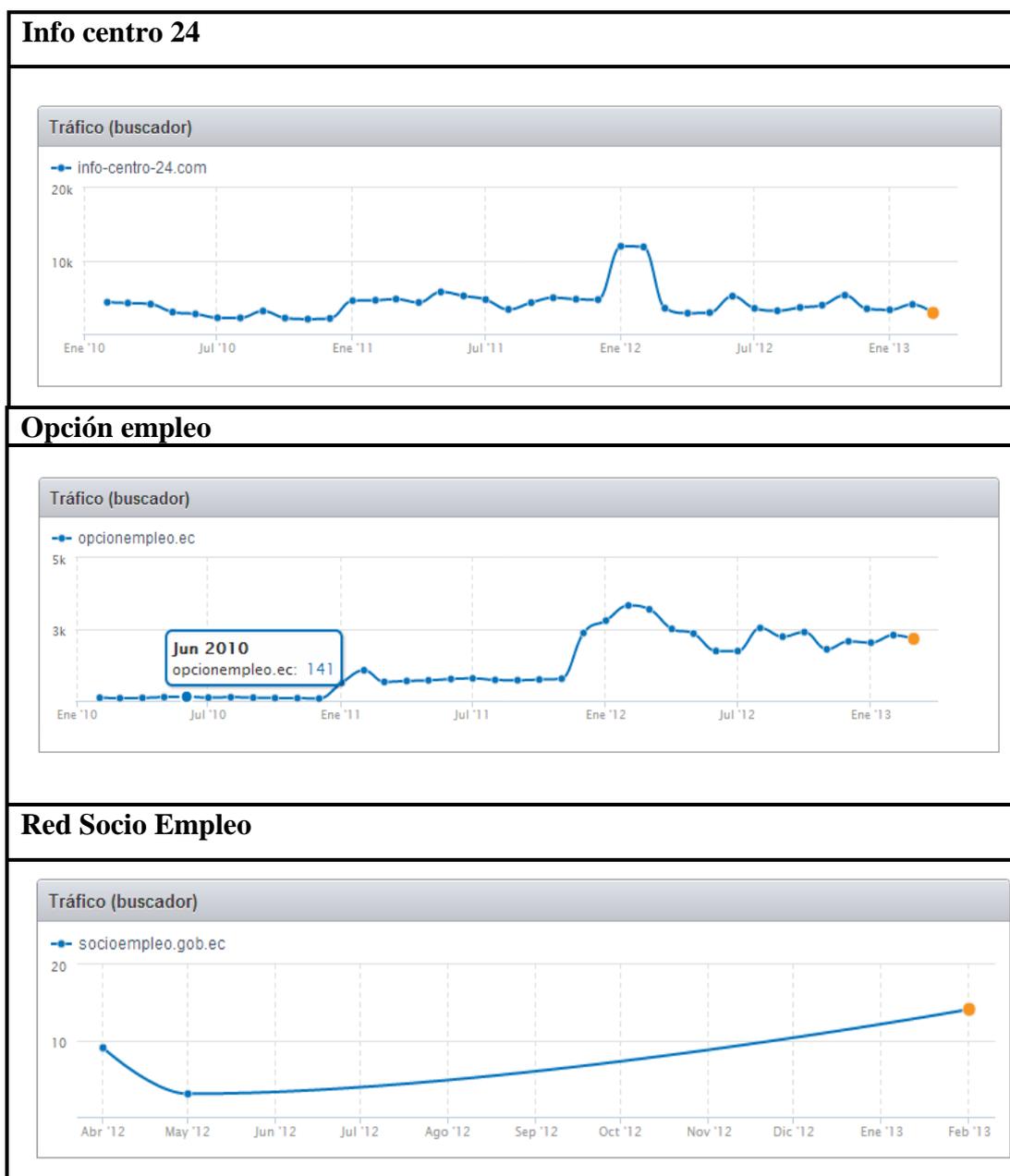
**Tabla 2.** Análisis del impacto de las bolsas de trabajo virtuales



**Tabla 2.** Análisis del impacto de las bolsas de trabajo virtuales (continuación).



**Tabla 2.** Análisis del impacto de las bolsas de trabajo virtuales (continuación).



**Elaborado por:** Angelo Andy & Andrés Vargas.

Como se observa en la tabla 2 las bolsas de trabajo han tenido una buena aceptación dentro del mercado ecuatoriano, especialmente en los últimos años, es por esta razón que se cree conveniente continuar con el proceso de creación del sistema para uso de los estudiantes, egresados, graduados así como también personal de la Universidad Politécnica Salesiana, tales como: Bienestar Estudiantil, delegados de graduados, entre otros.

#### ***2.1.1.4. Criterios generales sobre la gestión de las bolsas de trabajo.***

Las páginas webs o portales de bolsas de trabajo están clasificados por módulos, en la mayoría de casos estos módulos son: de postulantes y de empleadores. Pero el más importante debido a su tarea es el módulo de administración.

Una de las claves para el correcto funcionamiento de una bolsa de trabajo es la gestión o administración del portal. Para efectuar este trabajo se requiere una persona especializada que realice las tareas administrativas que comprenden: visualizar las postulaciones de los postulantes, revisión de las ofertas publicadas por la empresa, mantenimiento y actualización de noticias o información relacionada al portal en general.

La persona encargada de la gestión es la responsable de todo el sitio, ya que es la que tiene por ejemplo, los códigos o contraseñas para hacer modificaciones en la página. Es la única persona autorizada para realizar los contenidos del sitio, puede decidir que se aumenta o se elimina. También puede dar de alta o baja a usuarios en este caso a las empresas, postulantes o los usuarios reportes que no cumplan con un requisito mínimo o que incumplan con alguna política del sitio.

#### **2.1.2. Hoja de vida o Curriculum Vitae.**

##### ***2.1.2.1. Definición.***

La Hoja de Vida es una recopilación de todos los datos académicos y experiencia de una persona a lo largo de su vida independientemente del puesto de trabajo al cual se opta en el proceso de selección. (cv-resume, 2011)

##### ***2.1.2.2. Elementos.***

La estructura y los elementos que debe contener todo Curriculum Vitae son los siguientes:

- **Datos personales:** nombre y apellidos, lugar y fecha de nacimiento, estado civil, dirección personal, número de teléfono de contacto, dirección de correo electrónico, etc.
- **Formación académica:** estudios que se ha realizado, indicando fechas, centro, y lugar donde han sido realizados.

- **Otros títulos y seminarios:** estudios realizados complementarios a los universitarios que mejoran la formación universitaria, tales como: maestrías, certificaciones, etc, indicando las fechas, el centro y el lugar donde fueron realizados.
- **Experiencia profesional:** Experiencia laboral relacionada con los estudios universitarios o que puedan ser de interés para la empresa que desea contratar los servicios. Se debe señalar las fechas, la empresa dónde se trabajó y las funciones y tareas llevadas a cabo.
- **Idiomas:** En este apartado se menciona los idiomas que se conoce y el nivel. Si se obtuvo algún título reconocido, que acredite los conocimientos en estos ámbitos.
- **Informática:** Señala aquellos conocimientos informáticos que se posea: sistemas operativos, procesadores de texto, hojas de cálculo, bases de datos, diseño gráfico, Internet, etc.
- **Otros datos de interés:** en este último apartado se señala todos aquellos aspectos que no han sido incluidos todavía, tales como: carné de conducir, disponibilidad, etc.

### *2.1.2.3. Tipos.*

Una misma información se puede organizar de formas muy diferentes, dependiendo de la estructura que se elija se dará más importancia a unos aspectos y no a otros. A la hora de redactar un currículum ocurre exactamente lo mismo. Puede organizar la información primando, por ejemplo, la experiencia laboral más reciente, los cursos y las carreras cursadas o bien destacar los logros o metas alcanzadas. (modelocurriculum, 2008)

De esta manera, se puede distinguir fundamentalmente tres tipos de currículum, según la forma en que esté organizada la información:

- **Curriculum cronológico:** “es aquel que organiza la información cronológicamente, partiendo de los logros más antiguos y llegando hasta los más recientes. Esto permite al selector de personal conocer la evolución ascendente de la carrera profesional, aunque no siempre facilita el objetivo

fundamental de cualquier curriculum que es conseguir una entrevista personal. El punto positivo de este tipo de organización es que pone de relieve, si cabe, la estabilidad y la evolución ascendente de la carrera. Destaca la promoción, el aumento de tareas o las subidas en la escala profesional”. (modelocurriculum, 2008)

- **Curriculum funcional o temático:** “A diferencia del curriculum cronológico, el funcional distribuye la información por temas. De esta forma, permite proporcionar un conocimiento rápido de la formación y experiencia en un ámbito determinado. Además, al no seguir una progresión cronológica, permite seleccionar los puntos positivos, destacar las habilidades propias y logros conseguidos”. (modelocurriculum, 2008)
- **Curriculum combinado o mixto:** “Es el más completo de los tres modelos, aunque, por ello, también es el más complejo de elaborar. A grandes rasgos, el curriculum combinado es una mezcla del cronológico y del temático. Parte siempre del modelo funcional, organizando la información por áreas temáticas o profesionales, para llegar después a la organización en el tiempo. De esta forma se destacan las habilidades que se tienen al tiempo que se refleja la experiencia y la formación”. (modelocurriculum, 2008)

### **2.1.3. Entornos y herramientas.**

#### ***2.1.3.1. Introducción.***

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. (Universidad Politécnica de Valencia, 2003, p 3)

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido

a esto, es difícil automatizar todo un proceso de desarrollo de software. Sin embargo, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos: (Universidad Politécnica de Valencia, 2003, p 4)

- **Especificación de software:** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- **Diseño e implementación:** Se diseña y construye el software de acuerdo a la especificación.
- **Validación:** El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- **Evolución:** El software debe evolucionar, para adaptarse a las necesidades del cliente.

#### ***2.1.3.2. Herramienta.***

“En informática, es un subprograma o módulo encargado de funciones específicas y afines entre sí para realizar una tarea. Una aplicación o programa puede contar con múltiples herramientas a su disposición”. (Alegsa, 1998)

#### ***2.1.3.3. Desarrollo de software.***

Es la solución ajustada a los requerimientos de una empresa u otro ente particular, surgiendo de una idea, necesidad o problema. La arquitectura modular en el desarrollo de software, deja abierta la posibilidad de enriquecer el sistema a medida que sea necesario, agregándole nuevas funciones e integrándolo con otras plataformas como la web, dispositivos móviles, etc. (Elecsis, 2011)

Cuando se va desarrollar un software intervienen muchas personas, dentro de ellas se encuentra el cliente quien es el que tiene el problema en su empresa y desea que sea solucionado, aquí interviene el analista de sistema quien es el encargado de hacerle llegar todos los requerimientos y necesidades que tiene el cliente a los programadores quienes son las personas encargadas de realizar la codificación y diseño del sistema para después probarlo e implementarlo. (Ecured, 2010)

#### **2.1.3.4. Entorno de desarrollo.**

“Llamado también entorno integrado de desarrollo (IDE por sus siglas en inglés), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios”. (Prezi, 2014)

El IDE consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. También pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. (Prezi, 2014)

#### **2.1.3.5. Base de datos.**

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos. (EcuRed, 2014)

Existen programas denominados sistemas gestores de bases de datos, abreviados SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática. (EcuRed, 2014)

#### **2.1.3.6. Servidor web.**

La principal función de un servidor web es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso se decide que los servidores web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante. (Duplika, 2014)

Un servidor web tiene mucha importancia ya que sin ellos la Internet tal como se la conoce, no existiría. Los servidores son como la columna vertebral de la estructura de Internet. La industria del web hosting es simplemente la forma de alquilar esos espacios de memoria y administración de datos.

#### **2.1.3.7. Herramienta CASE.**

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (Itescam, 2009, p 1)

El empleo de herramientas Case permiten integrar el proceso de ciclo de vida:

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento.

#### **2.1.4. Requerimiento funcional.**

Define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

Establecen los comportamientos del sistema. Típicamente, un analista de requisitos genera requisitos funcionales luego de diagramar los casos de uso. Sin embargo, esto puede tener excepciones, ya que el desarrollo de software es un proceso iterativo y algunos requisitos son previos al diseño de los casos de uso. Ambos elementos (casos de uso y requisitos) se complementan en un proceso bidireccional. (EcuRed, 2013)

Un requisito funcional típico contiene un nombre y un número de serie único y un resumen. Esta información se utiliza para ayudar al lector a entender por qué el requisito es necesario, y para seguir al mismo durante el desarrollo del producto. (EcuRed, 2013)

## **2.2. Lenguaje Unificado de Modelado (UML)**

### **2.2.1. Definición.**

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (EcuRed, 1997)

Es importante remarcar que UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. (Scholarium, 2014)

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar. (Scholarium, 2014)

### **2.2.2. Diagramas de casos de uso.**

#### **2.2.2.1. Definición.**

Para la descripción de los requerimientos funcionales del sistema de la bolsa de trabajo se utilizará los diagramas de casos de uso, los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. (Cáceres, 2003, p 1)

### ***2.2.2.2. Elementos básicos.***

- **Actores:** Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier cosa externa que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas.
- **Caso de uso:** Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando. Se representan mediante un óvalo. Cada caso de uso debe detallarse, habitualmente mediante una descripción textual.
- **Asociaciones:** Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso.
- **Escenario:** Es una interacción entre el sistema y los actores, que puede ser descrito mediante una secuencia de mensajes. Un caso de uso es una generalización de un escenario. (Cáceres, 2003, p 2)

### **2.2.3. Diagrama de clases.**

#### ***2.2.3.1. Definición.***

Sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia y de uso. El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación. (WikiUml, 2014)

#### ***2.2.3.2. Elementos básicos.***

Un diagrama de clases está compuesto por los siguientes elementos:

**Clase:** Es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase). A través de ella se puede modelar el entorno en estudio. (Universidad de Chile, 2010)

**Asociaciones:** Las clases se relacionan con otras de diferentes maneras:

- **Generalización:** La herencia es uno de los conceptos fundamentales de la programación orientada a objetos, en la que una clase recoge todos los atributos

y operaciones de la clase de la que es heredera, y puede alterar/modificar algunos de ellos, así como añadir más atributos y operaciones propias.

- **Asociación:** representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de conexiones entre objetos. Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describe la conexión entre diferentes clases, las asociaciones pueden tener un papel que especifica el propósito de la asociación y pueden ser unidireccionales o bidireccionales.
- **Composiciones:** son asociaciones que representan acumulaciones muy fuertes. Esto significa que las composiciones también forman relaciones completas, pero dichas relaciones son tan fuertes que las partes no pueden existir por sí mismas. Únicamente existen como parte del conjunto, y si este es destruido las partes también lo son. (KDE, 2012)

#### **2.2.4. Diagrama de estados.**

##### ***2.2.4.1. Definición.***

Es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos. (Martínez, 2013, p 5)

##### ***2.2.4.2. Elementos básicos.***

**Estado:** Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin. (KDE, 2012)

## 2.3. Metodología Técnica de Modelado de Objetos (OMT)

### 2.3.1. Introducción.

Para el desarrollo del proyecto se ha escogido la metodología OMT, ya que es una de las metodologías de análisis y diseño orientada a objetos, más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software. (EsXoops, 2001)

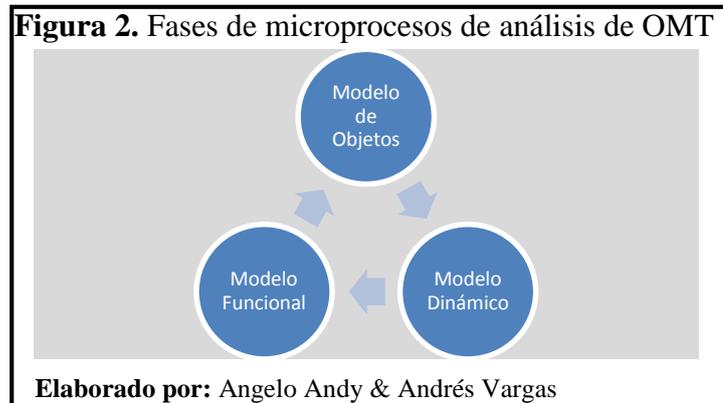
### 2.3.2. Modelos.

“Un modelo es una abstracción de algo, con la finalidad de comprenderlo, antes de construirlo, ya que un modelo omite los detalles no esenciales, es más sencillo manejarlos, que manejar la entidad original”. (Romero, 2001, p 1)

Esta técnica es trilateral, ya que toma en cuenta tres puntos de vista: modelo de objetos, modelo dinámico y modelo funcional.

- **El modelo de objetos:** Es el modelo más importante, ya que en él se identifican las clases dentro del sistema junto con sus relaciones, así como sus atributos y operaciones, lo que representa la estructura estática del sistema. El modelo de objetos se representa mediante un diagrama de clases.
- **El modelo dinámico:** Representa los aspectos temporales de comportamiento "de control" del sistema, mediante la secuencia de operaciones en el tiempo. Se representa por un diagrama de estados. El estado de un objeto es un conjunto de valores para sus atributos, en un cierto momento, el cual cambia al recibir un estímulo llamado evento.
- **El modelo funcional:** Representa los aspectos transformacionales "de función" del sistema, mediante la transformación de valores de los datos. Se utilizan diagramas de flujo de datos para mostrar las dependencias funcionales. (Romero, 2001)

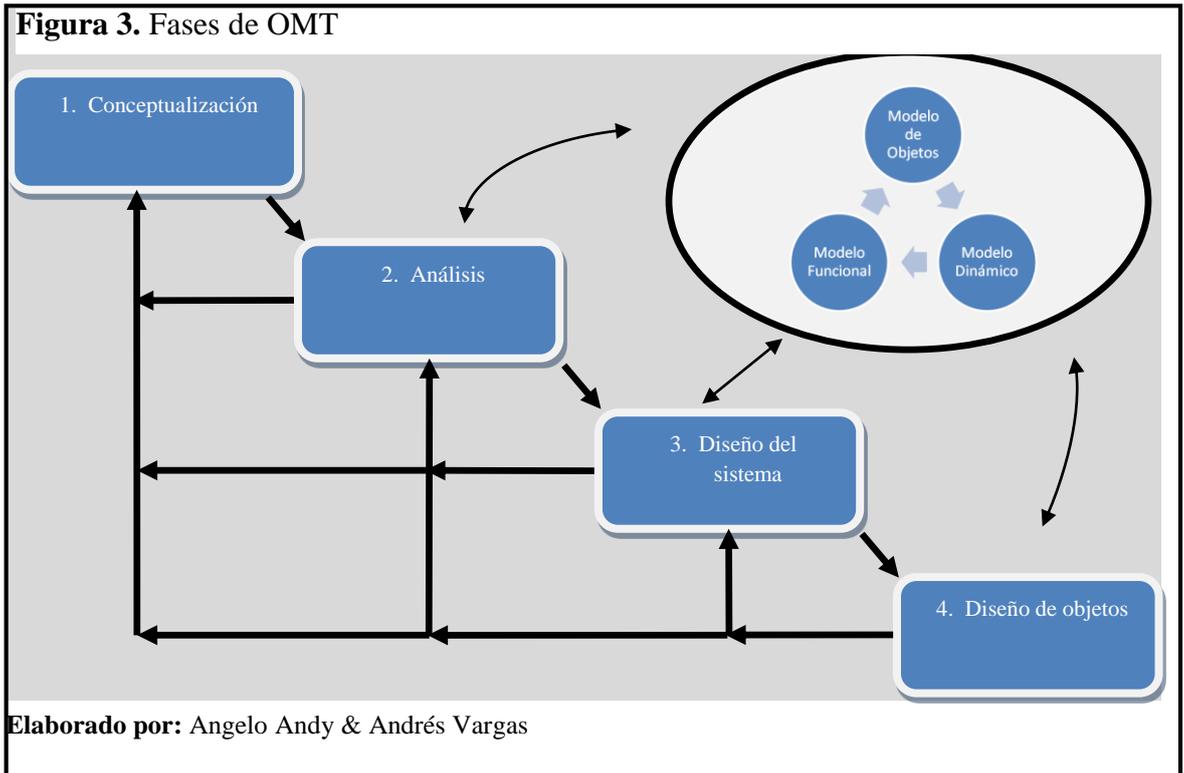
Cada modelo describe un aspecto del sistema pero contiene referencias a los demás modelos. Lo cual indica que los tres no son totalmente independientes.



### 2.3.3. Fases.

La metodología OMT utiliza cinco fases que son las siguientes:

1. Conceptualización
2. Análisis
3. Diseño del sistema
4. Diseño de objetos
5. Implementación



### 2.3.3.1. *Conceptualización.*

La conceptualización de OMT se refiere a definir los requisitos para un sistema en suficiente detalle para permitir el lanzamiento del análisis formal. Las salidas típicas de la actividad incluyen una declaración del problema, una declaración de los requisitos y los análisis de un caso de uso del sistema propuesto. Durante la conceptualización, las necesidades y los requisitos del negocio para un sistema propuesto se identifican junto con un cierto análisis anticipado con respecto a la viabilidad tecnológica y sociológica del sistema que se desarrollará.

A su vez la conceptualización se conforma de varias actividades que permiten identificar y desarrollar de mejor manera una solución, estas actividades se describen a continuación:

**Identificación del problema:** Esta actividad implica establecer los requisitos del negocio y las necesidades del usuario de un sistema propuesto.

**Desarrollo inicial de la solución:** Esta actividad implica definir un sistema propuesto describiendo y demostrando como los usuarios de un sistema lo utilizarán. Para esta

actividad se utiliza un diagrama de casos de uso que defina los límites del sistema e ilustre los usuarios del mismo.

**Definición de los requisitos de sistema:** Esta actividad implica definir los requisitos para el sistema propuesto, basados en el análisis preliminar de los sistemas y el concepto del uso del sistema definidos por el modelo de casos de uso del desarrollo inicial de la solución.

#### ***2.3.3.2. Análisis.***

El análisis orientado a objetos de OMT se refiere a definir los modelos que son una representación de requerimientos formales para el sistema previsto. El análisis comienza con una delimitación del problema, construyendo un modelo que represente una situación del mundo real mostrando sus características o propiedades principales. El analista debe trabajar junto con el usuario para conocer el problema ya que su delimitación exige que sea completa y correcta. (Prezi, 2013)

El análisis orientado a objetos de OMT se realiza en dos fases, análisis del dominio y análisis de aplicación, utilizando los microprocesos de análisis del modelo de objetos, análisis del modelo dinámico y del análisis modelo funcional.

**Análisis del Dominio:** Esta actividad implica identificar y modelar las clases que son relevantes para el problema, se deberá identificar los atributos para cada clase, las operaciones que se realizan y las asociaciones entre las clases.

**Análisis de aplicación:** Esta actividad implica el desarrollo de un modelo de aplicación, donde las clases encontradas en el análisis del dominio se emplean. El modelo de la aplicación es un diseño lógico puesto en práctica-independiente que es capaz de ser rastreado usando las técnicas de papel o computarizadas.

#### ***2.3.3.3. Diseño del sistema.***

Durante el diseño del sistema, se selecciona la estructura de alto nivel del sistema. Existen varias arquitecturas canónicas que pueden servir como un punto de inicio adecuado. El paradigma orientado a objetos no introduce vistas especiales en el diseño del sistema, pero se incluye para tener una cobertura completa del proceso de

desarrollo de software. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta.

Los pasos son:

1. Organizar el sistema en subsistemas.
2. Identificar la concurrencia inherente al problema.
3. Escoger la estrategia básica para implantar los almacenamientos de datos en términos de estructuras de datos, archivos y bases de datos.
4. Identificar recursos globales y determinar los mecanismos para controlar su acceso.

El desarrollo de arquitectura del sistema implica el preparar de la configuración del software del sistema del modelo del análisis, y el realizar de los estudios comerciales para tomar las decisiones del diseño que satisfacen requisitos del funcionamiento y de la capacidad.

#### ***2.3.3.4. Diseño de objetos.***

El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes. (Monografías, 2013)

Durante el diseño de objetos se elabora el modelo de análisis y se proporciona una base detallada para la implantación. Se toman las decisiones necesarias para realizar un sistema sin entrar en los detalles particulares de un lenguaje o base de datos particular.

## **2.4. Herramientas de desarrollo**

### **2.4.1. Netbeans IDE.**

El IDE NetBeans es un entorno de desarrollo integrado, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe

además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (netbeans, 2013)

El proyecto se desarrollará utilizando la versión 7.3 de Netbeans, ya que esta versión posee una gran variedad de módulos y componentes orientados a la web, esto permitirá realizar un trabajo de gran calidad visual y funcional.

#### ***2.4.1.1. Java Server Faces (JSF).***

Java Server Faces es un framework o ambiente de trabajo, destinado a facilitar la construcción y desarrollo de interfaces para aplicaciones basadas en entorno web.

JSF tiene la función de proveer un grupo muy extenso de herramientas pre-desarrolladas para facilitar la creación de interfaces; herramientas como: botones, cajas de texto, tablas de datos, etc. Similar a los objetos empleados para la generación de aplicaciones de escritorio. (nDeveloper, 2009)

Uno de los puntos fuertes hacia los cuales se dirige la arquitectura JSF es la generación de interfaces que puedan ser consumidas por diferentes tipos de dispositivos clientes. De esta manera la misma interface puede verse desde una PC de escritorio, una table PC o un PDA, sin tener que crear nuevas interfaces ni desarrollar diferentes procesos para cada uno de los ambientes antes mencionados. (nDeveloper, 2009)

#### ***2.4.1.2. Primefaces.***

Es una librería de componentes visuales para Java Server Faces (JSF) de código abierto que cuenta con gran cantidad de componentes que facilitan la creación de las aplicaciones web.

Las propiedades de este componente son las siguientes:

- Conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros)
- Soporte de ajax con despliegue parcial, lo que permite controlar cuáles componentes de la página actual se actualizarán y cuáles no
- Componente para desarrollar aplicaciones web para móviles-celulares, especiales para Iphones, Palm, Android y teléfonos móviles Nokia.

### **2.4.1.3. Java Persistence Api (JPA).**

JPA es un framework de persistencia, que abstrae de las bases de datos y brinda un estándar para persistir los datos en java. JPA viene a solucionar el vacío que hay entre utilizar objetos y persistirlos en una base de datos relacional. (slideshare, 2013)

JPA proporciona un modelo de persistencia para mapear bases de datos relacionales en Java. Puede utilizarse directamente en aplicaciones web y aplicaciones clientes.

### **2.4.2. Base de datos oracle.**

Oracle es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones.
- Estabilidad
- Escalabilidad
- Soporte multiplataforma
- Usuarios concurrentes
- Gestión de la seguridad
- Compatibilidad

Dentro de las principales ventajas que se tiene al usar Oracle, se pueden mencionar las siguientes:

- Es el motor de base de datos relacional más usado a nivel mundial.
- Puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Oracle es la base de datos con más orientación hacia INTERNET.

Para el desarrollo del proyecto se utilizará la versión “Oracle 11G Express Edition”, ya que esta versión es de uso libre y exclusivo para fines educativos y no lucrativos.

### 2.4.3. Apache Tomcat.

Tomcat es un servidor web con soporte de servlets y JSPs. A partir de la versión 4.x Tomcat fue lanzado con el contenedor de servlets "Catalina", con el contenedor HTTP "Coyote" y un motor para JSP llamado "Jasper". Las principales características de estos tres componentes son:

**Catalina:** dicho componente implementa las especificaciones de servlets y JSP. Para Apache Tomcat el elemento principal es una base de datos de nombres de usuarios, password y roles a estos usuarios, permitiendo a Catalina integrarse donde ya existe información de autenticación como describe la especificación de servlets. (apachefoundation, 2013)

**Coyote:** Componente conector que admite el protocolo HTTP para el servidor web y que escucha en un puerto TCP especificado por el servidor y envía la solicitud al motor Tomcat para que éste procese la solicitud y envíe una respuesta al cliente. (apachefoundation, 2013)

**Jasper:** Analiza archivos JSP para compilar el código Java y, si se producen cambios, éste los vuelve a compilar. Desde la versión 5.x se usa Jasper 2 que es JavaServer para webs 2.0. (apachefoundation, 2013)

### 2.4.4. Sybase power designer.

Es una poderosa herramienta basada en una tecnología orientada a alinear el negocio y la Tecnología de Información (IT); es una solución de modelado y diseño empresarial que colabora en la implementación efectiva de la arquitectura empresarial y brinda técnicas poderosas de análisis y diseño durante todo el ciclo de vida de desarrollo del proyecto con gestión de meta-datos (datos que describen otros datos, en general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso). (Instituto Politécnico Nacional, 2012)

PowerDesigner permite:

- Aumentar la productividad: Alinea el negocio y el área de TI para mejorar la productividad general.
- Admitir entornos abiertos: Proporciona compatibilidad abierta para entornos heterogéneos.

- Incluir funciones de personalización: PowerDesigner es altamente personalizable para ayudarlo a hacer cumplir los estándares y garantizar el cumplimiento regulatorio.
- Diseñado para la empresa compleja: Facilita las implementaciones de la arquitectura empresarial al capturar intuitivamente las intersecciones entre todas las capas y perspectivas arquitectónicas de la empresa.

Proporciona herramientas de análisis: Aumenta la agilidad del negocio con la tecnología de enlace y sincronización para un verdadero análisis de impacto.

## CAPÍTULO 3

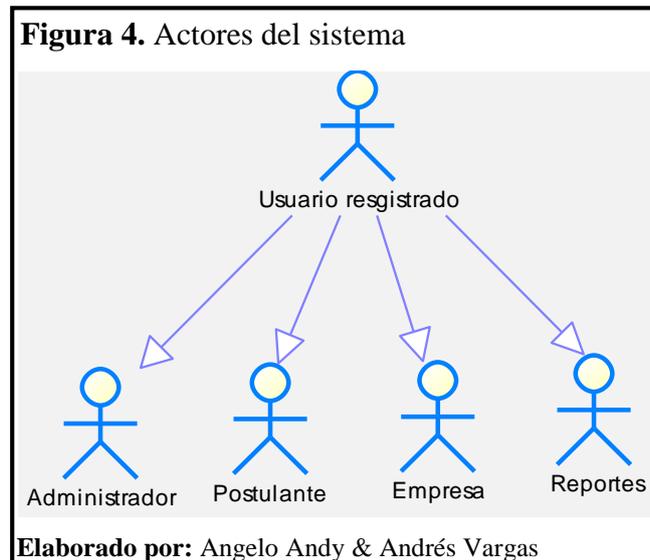
### REQUERIMIENTOS FUNCIONALES Y MODELAMIENTO BAJO OMT

#### 3.1. Análisis de requerimientos funcionales

##### 3.1.1. Identificación de actores y escenarios del sistema.

Para la identificación de los usuarios y sus respectivos perfiles se ha realizado una entrevista con el personal del Departamento de Bienestar Estudiantil de la Universidad Politécnica Salesiana Campus El Girón, específicamente con la señorita Mónica Viturco quien será la encargada de manejar el sistema.

Una vez terminada la entrevista se logró identificar y determinar los siguientes usuarios con sus respectivas tareas:



**Actores con sus respectivos escenarios:**

**Usuario administrador:**

- Autenticar
- Cambiar contraseña
- Recuperar contraseña
- Validar empresas
- Validar usuario reportes.
- Visualizar ofertas laborales.
- Visualizar postulantes.

- Buscar y visualizar hojas de vidas según criterios de búsqueda.
- Generar reportes.
- Historial postulaciones
- Gestionar cuentas de usuarios.
- Gestionar información general.
- Visualizar y gestionar las sugerencias recibidas.

**Usuario postulante:**

- Autenticar
- Cambiar contraseña
- Recuperar contraseña
- Gestionar hoja de vida.
- Visualizar, descargar o imprimir hojas de vida.
- Visualizar/postular una oferta laboral
- Postularse a una oferta laboral.
- Enviar sugerencias y observaciones.
- Visualizar historial.

**Usuario empresa:**

- Registrar empresa
- Autenticar
- Cambiar contraseña
- Recuperar contraseña
- Gestionar ofertas laborales.
- Buscar y visualizar hojas de vidas según criterios de búsqueda.
- Visualizar, descargar o imprimir hojas de vida.
- Enviar sugerencias y observaciones.

**Usuario reportes:**

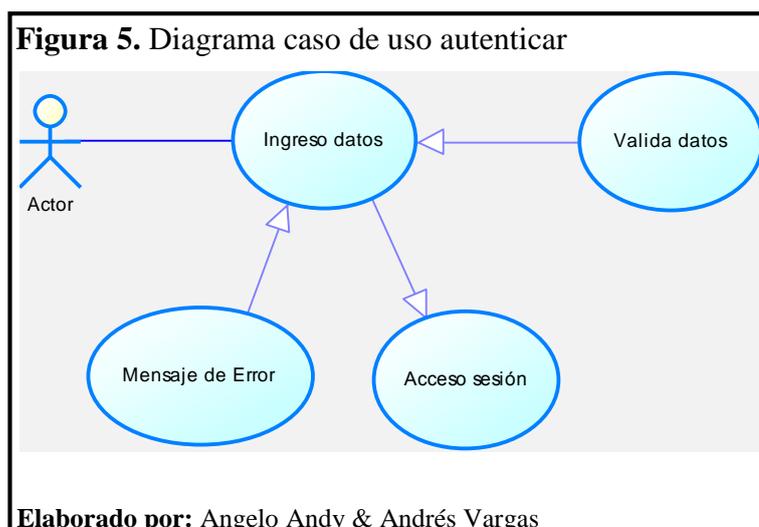
- Autenticar
- Cambiar contraseña
- Recuperar contraseña
- Generar Reportes

Cabe resaltar que las ofertas que se presenten por parte de la empresa en el sistema se contemplaran como vacantes laborales y pasantías.

### 3.2. Modelamiento OMT

#### 3.2.1. Conceptualización.

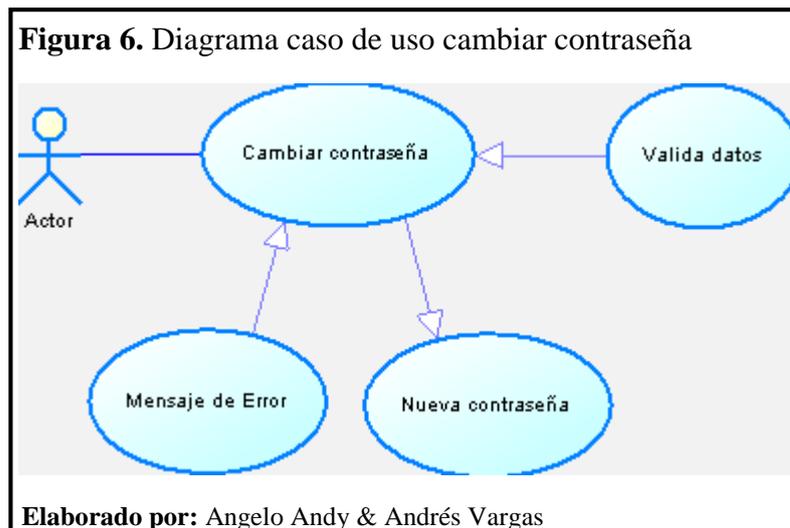
Para realizar los casos de uso del sistema de bolsa de trabajo, se analizará los procesos y subprocesos que manipularán cada uno de los actores del sistema.



**Tabla 3.** Especificación caso de uso: Autenticar

| CASO DE USO AUTENTICAR    |   |
|---------------------------|---|
| <b>Identificación:</b>    | 1   |
| <b>Nombre:</b>            | Autenticar  |
| <b>Descripción:</b>       | Proceso que permite la autenticación al sistema por perfil de usuario.  |
| <b>Actores:</b>           | Administrador, postulante, empresa y reportes   |
| <b>Pre-condiciones:</b>   | El usuario debe estar registrado en el sistema  |
| <b>Flujo normal:</b>      | <ol style="list-style-type: none"> <li>1. El usuario ingresa su login y contraseña correspondientes.</li> <li>2. Se valida sus credenciales en la base de datos, en la tabla: <i>bt_credenciales</i>, si los datos son correctos el usuario ingresa al sistema con su perfil.</li> <li>3. El sistema despliega la página inicial con su correspondiente menú de usuario.</li> </ol> |
| <b>Flujo alternativo:</b> | <ol style="list-style-type: none"> <li>2a. Si las credenciales son incorrectas el sistema genera un mensaje de error de autenticación y regresa a la página inicio de sesión.</li> </ol>  |

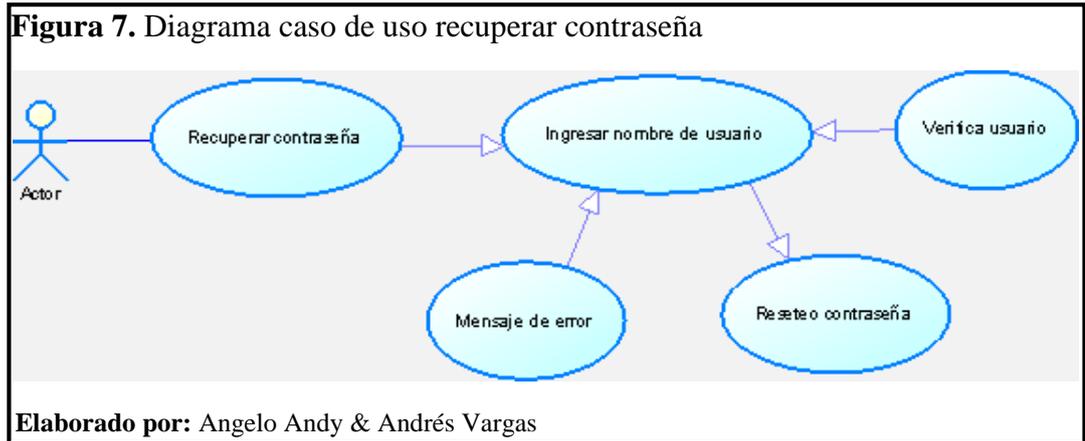
**Elaborado por:** Angelo Andy & Andrés Vargas



**Tabla 4.** Especificación caso de uso: Cambiar contraseña

| CASO DE USO CAMBIAR CONTRASEÑA |  |
|--------------------------------|--|
| <b>Identificación:</b>         | 2  |
| <b>Nombre:</b>                 | Cambiar Contraseña   |
| <b>Descripción:</b>            | Proceso que permite cambiar la contraseña actual por una nueva para autenticarse al sistema.   |
| <b>Actores:</b>                | Administrador, postulante, empresa y reportes  |
| <b>Pre-condiciones:</b>        | El usuario debe estar registrado en el sistema.  |
| <b>Flujo normal:</b>           | <ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema con la contraseña actual.</li> <li>2. Seleccionar la opción <i>Mi cuenta</i>, dentro de esta se escoge la pestaña <i>Cambiar Contraseña</i>.</li> <li>3. Para cambiar la contraseña es necesario seguir son los siguientes pasos:               <ol style="list-style-type: none"> <li>3.1 Ingresar la contraseña actual</li> <li>3.2 Ingresar la contraseña nueva</li> <li>3.3 Confirmar la nueva contraseña</li> </ol> </li> <li>4. El sistema valida la contraseña actual sea correcta, en la tabla: <i>bt_credenciales</i>.</li> <li>5. El sistema confirma que la contraseña nueva y la confirmación sean iguales.</li> <li>6. Se guarda la nueva contraseña válida para el usuario en la tabla: <i>bt_credenciales</i>.</li> <li>7. El sistema confirma mediante un mensaje que la contraseña ha sido cambiada correctamente.</li> </ol> |
| <b>Flujo alternativo:</b>      | <ol style="list-style-type: none"> <li>1.a. Si las credenciales son incorrectas el sistema devuelve un mensaje de error de autenticación.</li> <li>3.a. Si la nueva contraseña no coincide con la confirmación se presentara un mensaje de error.</li> </ol>   |

**Elaborado por:** Angelo Andy & Andrés Vargas

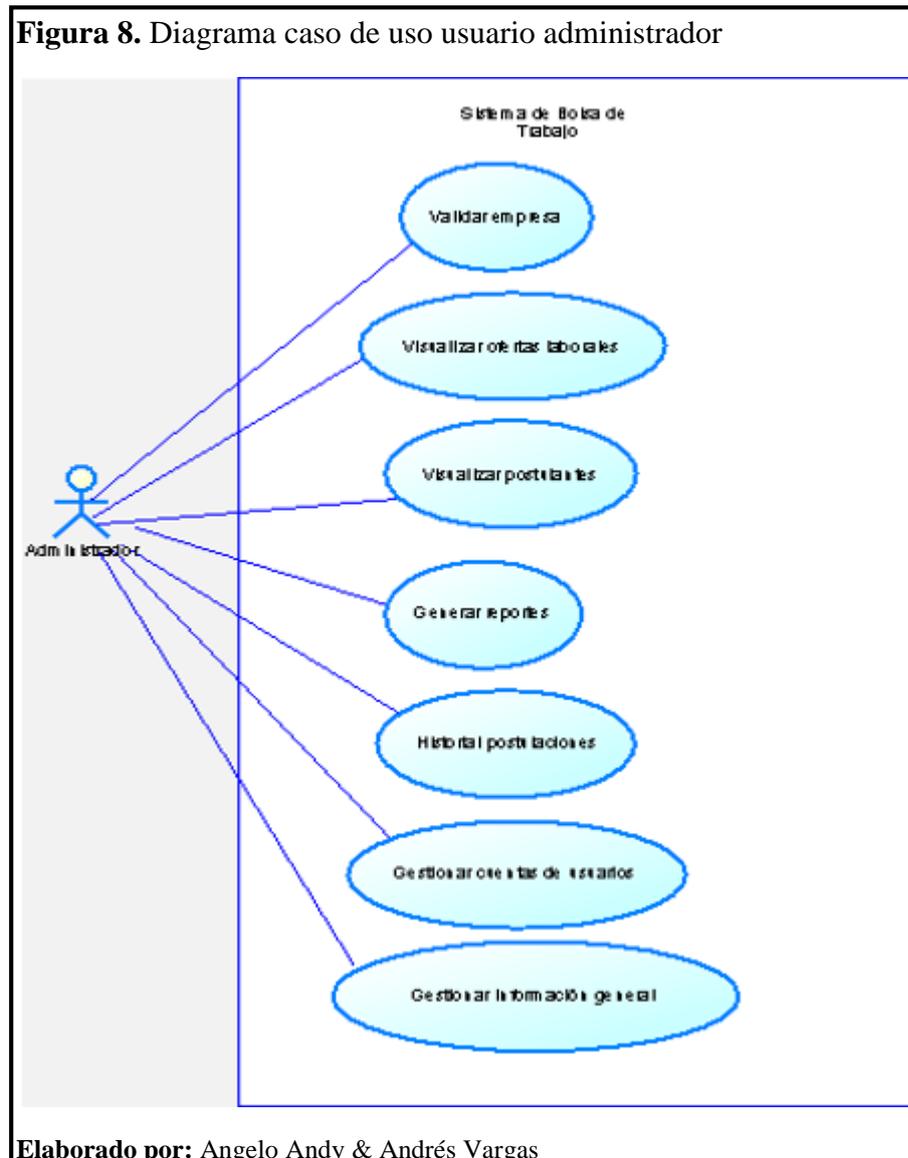


**Tabla 5.** Especificación caso de uso: Recuperar contraseña

| CASO DE USO RECUPERAR CONTRASEÑA |   |
|----------------------------------|---|
| <b>Identificación:</b>           | 3   |
| <b>Nombre:</b>                   | Recuperar Contraseña  |
| <b>Descripción:</b>              | Proceso que permite recuperar la contraseña olvidada.   |
| <b>Actores:</b>                  | Administrador, postulante, empresa y reportes   |
| <b>Pre-condiciones:</b>          | El usuario debe estar registrado en el sistema  |
| <b>Flujo normal:</b>             | <ol style="list-style-type: none"> <li>1. Ingresar a la opción <i>Olvido su contraseña?</i> desde la página principal de la web.</li> <li>2. Ingresar el nombre de usuario.</li> <li>3. El sistema verifica el nombre de usuario y envía un correo con la nueva contraseña a la dirección electrónica asociada con el usuario. Esta búsqueda se realiza en las tablas: <i>bt_credenciales</i> y <i>bt_contactos</i> de la base de datos.</li> </ol> |
| <b>Flujo alternativo:</b>        | <ol style="list-style-type: none"> <li>3.a. Si el nombre de usuario no existe en la base de datos el sistema devuelve un mensaje de error y no se envía ningún dato vía correo.</li> </ol>  |

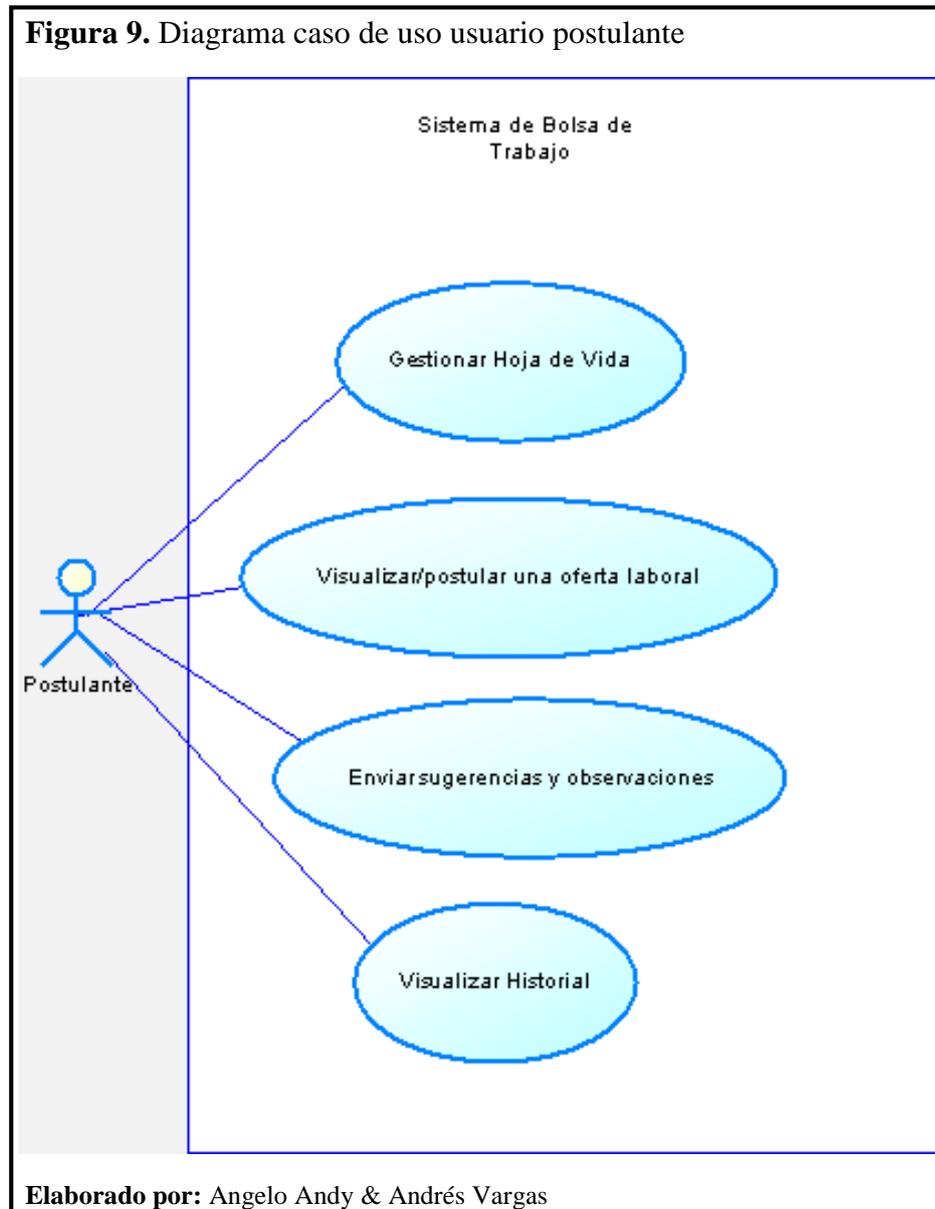
**Elaborado por:** Angelo Andy & Andrés Vargas

En la figura 8 se observa el diagrama de caso de uso de los procesos que puede realizar el usuario administrador en el sistema.



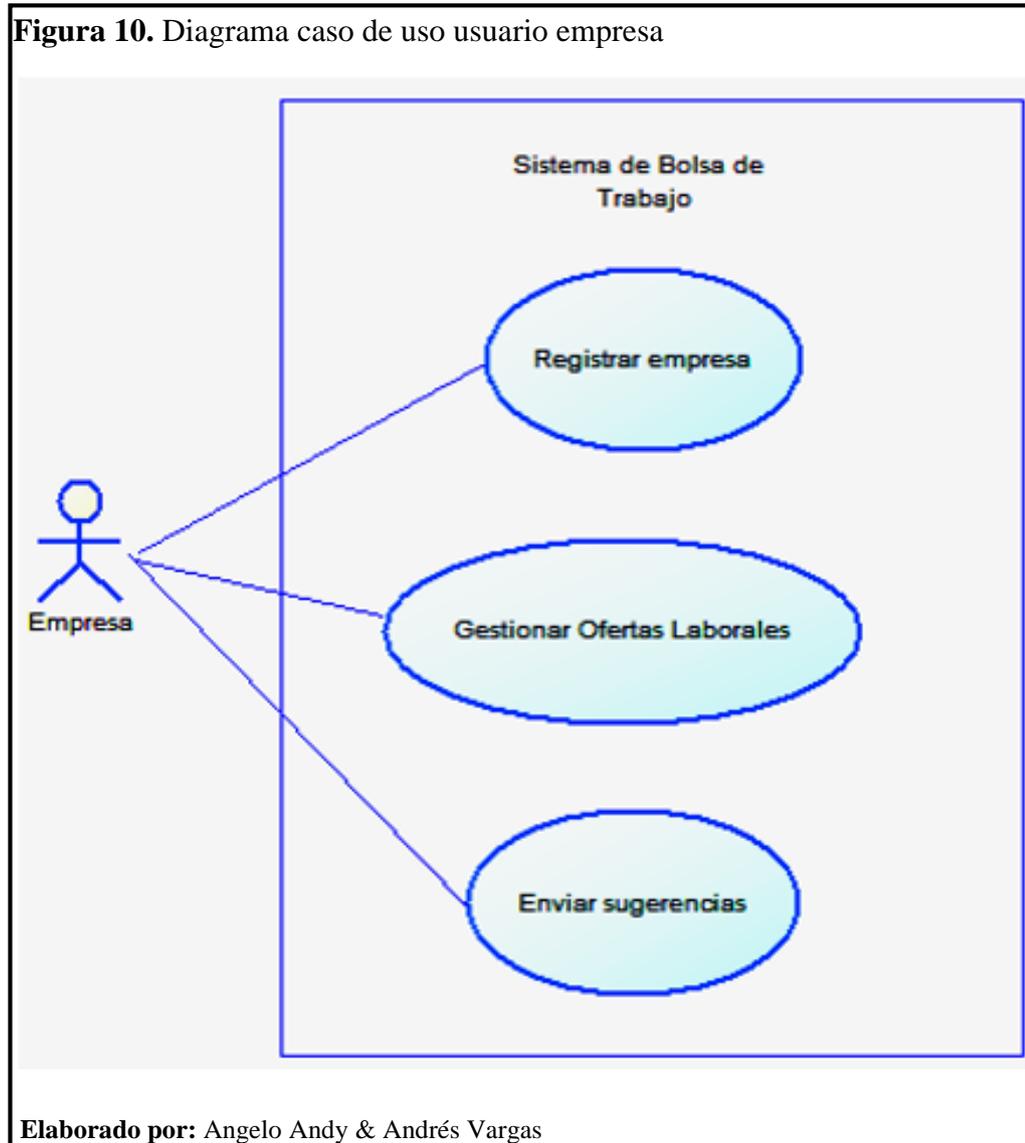
La información detallada de cada caso de uso se lo puede encontrar en el anexo 1: Manual técnico.

En la figura 9 se observa el diagrama de caso de uso de los procesos que puede realizar el usuario postulante en el sistema.



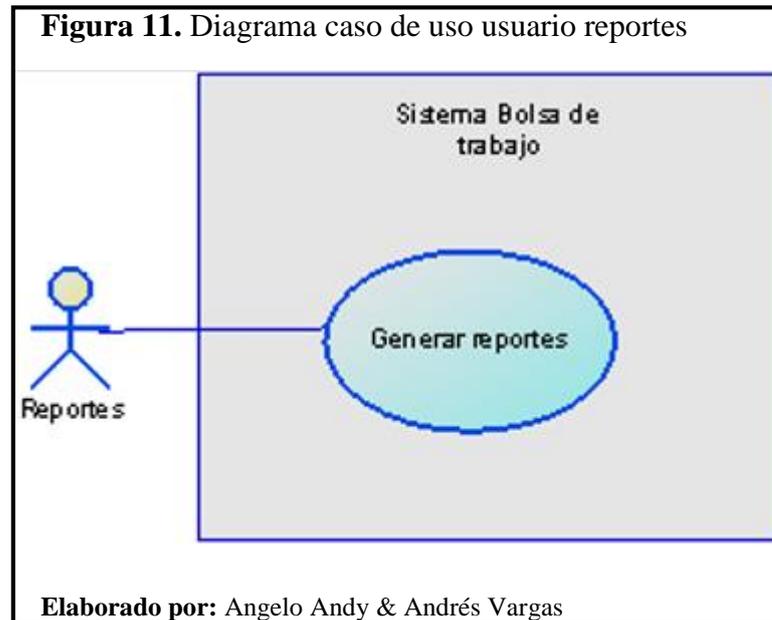
La información detallada de cada caso de uso se lo puede encontrar en el anexo 1: Manual técnico.

En la figura 10 se observa el diagrama de caso de uso de los procesos que puede realizar el usuario empresa en el sistema.



La información detallada de cada caso de uso se lo puede encontrar en el anexo 1: Manual técnico.

En la figura 11 se observa el diagrama de caso de uso de los procesos que puede realizar el usuario reportes en el sistema.



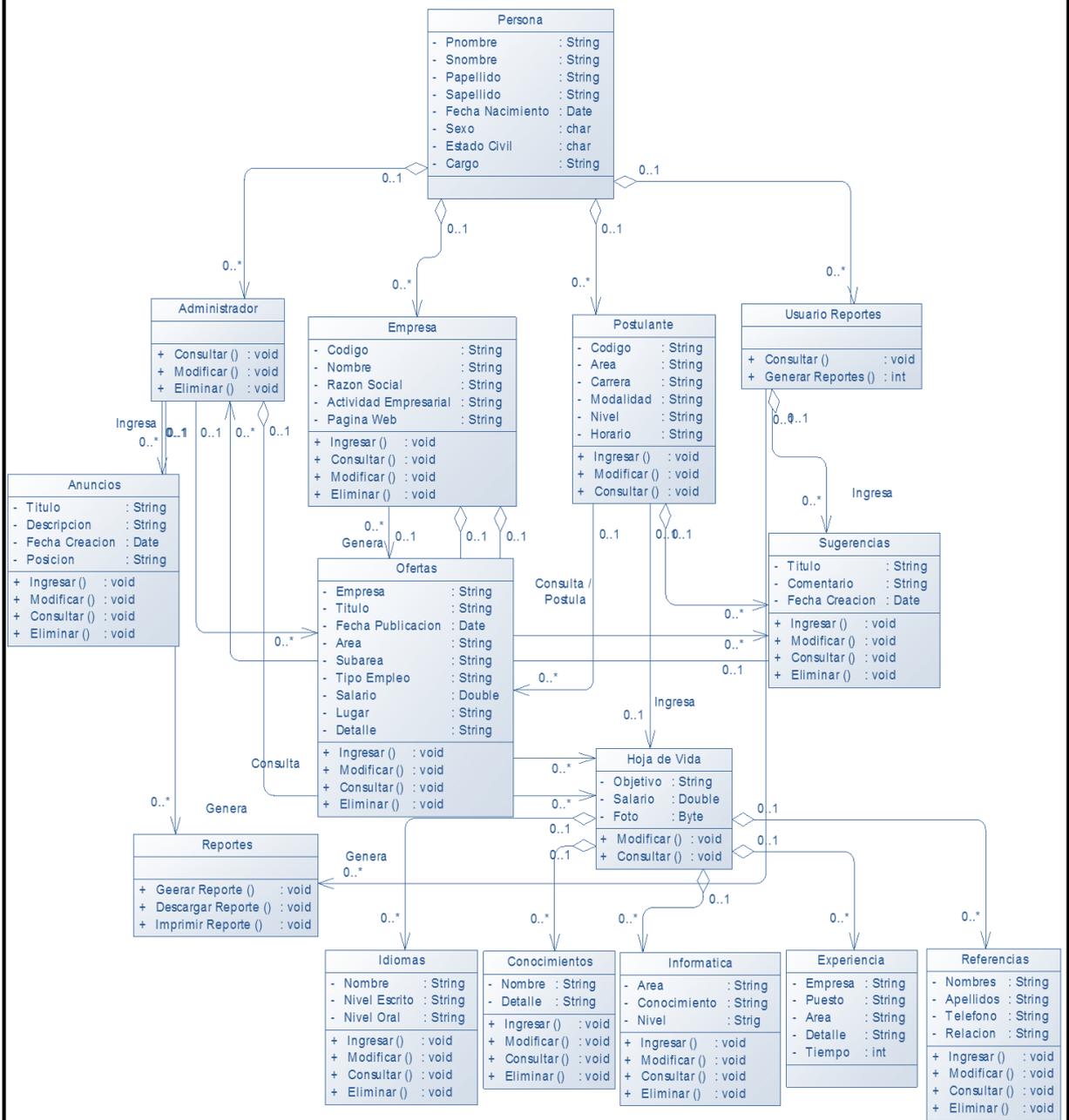
### 3.2.2. Modelo de análisis del sistema.

#### 3.2.2.1. Modelo de objetos.

Se detalla las entidades más importantes que intervienen en el sistema para su desarrollo.



**Figura 13.** Diagrama de clases



Elaborado por: Angelo Andy & Andrés Vargas

## Diccionario de clases

**Tabla 6.** Especificación del diccionario de clases

| No. | Clases                  | Descripción  |
|-----|-------------------------|--|
| 1   | <b>Persona</b>          | Esta clase almacena toda la información personal básica de los usuarios del sistema, que van hacer uso de sus funcionalidades.   |
| 2   | <b>Administrador</b>    | Esta clase almacena toda la información que está a cargo del usuario administrador como publicar anuncios, generar reportes, consultar datos de empresas, postulantes o usuarios reportes.     |
| 3   | <b>Empresa</b>          | Esta clase almacena toda la información de la empresa que desee utilizar el sistema de bolsa de empleo, para publicar ofertas laborales, buscar posibles candidatos a los puestos que ofertan. |
| 4   | <b>Postulante</b>       | Esta clase almacena todos los datos personales de los postulantes ya sean estudiantes, egresados o graduados.  |
| 5   | <b>Usuario reportes</b> | Esta clase almacena los datos personales y permisos de acceso de los usuarios reportes, que solamente podrán generar reportes para su revisión.  |
| 6   | <b>Ofertas</b>          | Esta clase almacena todas las ofertas laborales que han realizado las empresas que están registradas en el sistema.  |
| 7   | <b>Anuncios</b>         | Esta clase almacena todos los anuncios que ha realizado el administrador para el conocimiento de todos los usuarios del sistema.   |
| 8   | <b>Sugerencias</b>      | Esta clase almacena todas las sugerencias realizadas por los usuarios del sistema para la gestión del administrador.   |
| 9   | <b>Reportes</b>         | Esta clase almacena todos los reportes con los que cuenta el sistema y pueden ser generados por los diferentes usuarios del sistema.   |
| 10  | <b>Hoja de vida</b>     | Esta clase almacena la información básica de una hoja de vida del usuario postulantes y puede ser revisada por la empresa y por el administrador.  |
| 11  | <b>Idiomas</b>          | Esta clase almacena los datos de todos los idiomas que habla el postulante y están reflejados en la hoja de vida.  |
| 12  | <b>Conocimientos</b>    | Esta clase almacena todos los conocimientos en distintas áreas y que fueron adquiridos por el postulante y los hace públicos en su hoja de vida.   |
| 13  | <b>Informática</b>      | Esta clase almacena todos los conocimientos en informática que posee el postulante como manejo de sistemas operativos, paquetes de office, diseño gráfico, etc.                                |

**Tabla 6.** Especificación del diccionario de clases (continuación).

|           |                    |   |
|-----------|--------------------|---|
| <b>14</b> | <b>Experiencia</b> | Esta clase almacena los datos de la experiencia adquirida por el postulante en distintos puestos de trabajos en los cuales se ha desempeñado. |
| <b>15</b> | <b>Referencias</b> | Esta clase almacena los datos de todas las referencias que posee el postulante ya sea antiguos jefes, compañeros de trabajo, familiares, etc. |

**Elaborado por:** Angelo Andy & Andrés Vargas

## Diccionario de atributos

### Clase persona

Esta tabla contiene la información de datos personales de los usuarios del sistema Bolsa de Empleo.

**Tabla 7.** Especificación clase persona

| <b>Campo</b>            | <b>Descripción</b>  | <b>Tipo de Dato</b> | <b>Restricción</b> |
|-------------------------|---|---------------------|--------------------|
| <b>Pnombre</b>          | Contiene el primer nombre del usuario que accede al sistema.  | Varchar(15)         | Not Null           |
| <b>Snombre</b>          | Contiene el segundo nombre del usuario que accede al sistema.   | Varchar(15)         | Not Null           |
| <b>Papellido</b>        | Contiene el primer apellido del usuario que accede al sistema.  | Varchar(15)         | Not Null           |
| <b>Sapellido</b>        | Contiene el segundo apellido del usuario que accede al sistema.                                       | Varchar(15)         | Not Null           |
| <b>Fecha Nacimiento</b> | Contiene la fecha de nacimiento del usuario que accede al sistema.                                    | Datetime            | Null               |
| <b>Sexo</b>             | Contiene el género del usuario que accede al sistema.   | Char(10)            | Null               |
| <b>Estado Civil</b>     | Contiene el estado civil del usuario que accede al sistema.   | Char(10)            | Null               |
| <b>Cargo</b>            | Aplica para los usuarios Reporte y Empresa, contiene el cargo que tienen en la universidad y empresa. | Varchar(64)         | Null               |

**Elaborado por:** Angelo Andy & Andrés Vargas

## Clase empresa

Esta tabla contiene la información de la empresa que está registrada en el sistema.

**Tabla 8.** Especificación clase empresa

| Campo                        | Descripción  | Tipo de Dato | Restricción |
|------------------------------|--|--------------|-------------|
| <b>Código</b>                | Contiene el código autogenerated por el sistema para el acceso.  | Varchar(24)  | Not Null    |
| <b>Nombre</b>                | Contiene el nombre de la empresa.                                | Varchar(64)  | Not Null    |
| <b>Razón Social</b>          | Contiene la razón social de la empresa.                          | Varchar(100) | Null        |
| <b>Actividad Empresarial</b> | Contiene la actividad empresarial a la que se dedica la empresa. | Varchar(10)  | Null        |
| <b>Página Web</b>            | Contiene la dirección de la página web de la empresa.            | Varchar(64)  | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

## Clase postulante

Esta tabla contiene la información de los postulantes que existen en el sistema.

**Tabla 9.** Especificación clase postulante

| Campo            | Descripción  | Tipo de Dato | Restricción |
|------------------|--|--------------|-------------|
| <b>Código</b>    | Contiene el código (cedula) con el cual el postulante accede al sistema.                     | Varchar(24)  | Not Null    |
| <b>Área</b>      | Contiene el nombre del área en donde está su carrera, ejemplo: Área Ciencias Administrativas | Varchar(100) | Not Null    |
| <b>Carrera</b>   | Contiene el nombre de la carrera que estudia o estudio el postulante.                        | Varchar(100) | Not Null    |
| <b>Modalidad</b> | Contiene la modalidad en la que estudia el postulante.                                       | Varchar(20)  | Null        |
| <b>Nivel</b>     | Contiene el nivel en el que se encuentra estudiando el postulante.                           | Int          | Null        |
| <b>Horario</b>   | Contiene el horario en el que se encuentra estudiante el postulante, ejemplo: diurno.        | Varchar(15)  | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase ofertas

Esta tabla contiene la información de las ofertas laborales que ofertan las empresas registradas en el sistema.

**Tabla 10.** Especificación clase ofertas

| Campo                    | Descripción  | Tipo de Dato | Restricción |
|--------------------------|--|--------------|-------------|
| <b>Empresa</b>           | Contiene el código de la empresa que público la oferta.                  | Varchar(24)  | Not Null    |
| <b>Título</b>            | Contiene el título de la oferta de trabajo.                              | Varchar(120) | Not Null    |
| <b>Fecha publicación</b> | Contiene la fecha de publicación de la oferta.                           | Datetime     | Not Null    |
| <b>Área</b>              | Contiene el área en que se desea buscar un nuevo empleado de la empresa. | Varchar(100) | Not Null    |
| <b>Subarea</b>           | Contiene la subarea a la que pertenece la oferta laboral.                | Varchar(100) | Not Null    |
| <b>Tipo empleo</b>       | Contiene el tipo de empleo ejemplo: Tiempo completo, por horas, etc.     | Varchar(60)  | Not Null    |
| <b>Salario</b>           | Contiene el salario que ofrece la empresa para la vacante.               | Number       | Null        |
| <b>Lugar</b>             | Contiene el lugar en donde se oferta la vacante.                         | Varchar(250) | Null        |
| <b>Detalle</b>           | Contiene el detalle de la oferta laboral.                                | Varchar(250) | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase anuncios

Esta tabla contiene la información de las publicaciones que realizo el administrador para el conocimiento de los usuarios del sistema.

**Tabla 11.** Especificación clase anuncios

| Campo                 | Descripción   | Tipo de Dato | Restricción |
|-----------------------|---|--------------|-------------|
| <b>Título</b>         | Contiene el título del anuncio  | Varchar(24)  | Not Null    |
| <b>Descripción</b>    | Contiene la descripción del anuncio.  | Varchar(250) | Not Null    |
| <b>Fecha creación</b> | Contiene la fecha en la que fue publicado el anuncio.   | Datetime     | Not Null    |
| <b>Posición</b>       | Contiene la posición en la que va aparecer el anuncio, en la parte principal o tipo banner izquierdo. | Varchar(24)  | Not Null    |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase sugerencias

Esta tabla contiene la información de las sugerencias u observaciones que han hecho los usuarios en el sistema para que sean atendidos por el administrador.

**Tabla 12.** Especificación clase sugerencias

| Campo                 | Descripción   | Tipo de Dato | Restricción |
|-----------------------|---|--------------|-------------|
| <b>Título</b>         | Contiene el título de la sugerencia.                            | Varchar(24)  | Not Null    |
| <b>Comentario</b>     | Contiene el comentario u observación que realizo algún usuario. | Varchar(250) | Not Null    |
| <b>Fecha creación</b> | Contiene la fecha de creación de la sugerencia.                 | Datetime     | Not Null    |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase hoja de vida

Esta tabla contiene la información básica de la hoja de vida del postulante.

**Tabla 13.** Especificación clase hoja de vida

| Campo           | Descripción  | Tipo de Dato | Restricción |
|-----------------|--|--------------|-------------|
| <b>Objetivo</b> | Contiene el objetivo que tiene el postulante para un futuro puesto de trabajo. | Varchar(250) | Not Null    |
| <b>Salario</b>  | Contiene la aspiración salarial que tiene el postulante.                       | Number       | Null        |
| <b>Foto</b>     | Contiene la foto del postulante y que será publicada en la hoja de vida.       | Byte         | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase idiomas

Esta tabla contiene la información de los idiomas que habla el postulante.

**Tabla 14.** Especificación clase idiomas

| Campo                | Descripción   | Tipo de Dato | Restricción |
|----------------------|---|--------------|-------------|
| <b>Nombre</b>        | Contiene el nombre del idioma.                                      | Varchar(24)  | Null        |
| <b>Nivel Escrito</b> | Contiene el nivel que posee el postulante al escribir en un idioma. | Char(12)     | Null        |
| <b>Nivel Oral</b>    | Contiene el nivel que posee el postulante para hablar un idioma.    | Char(12)     | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase conocimientos

Esta tabla contiene la información de los conocimientos que posee el postulante.

**Tabla 15.** Especificación clase conocimientos

| Campo          | Descripción  | Tipo de Dato | Restricción |
|----------------|--|--------------|-------------|
| <b>Nombre</b>  | Contiene el nombre del conocimiento que posee el postulante. | Varchar(64)  | Null        |
| <b>Detalle</b> | Contiene el detalle del conocimiento.                        | Varchar(128) | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase informática

Esta tabla contiene la información de los conocimientos en informática que posee el postulante.

**Tabla 16.** Especificación clase informática

| Campo               | Descripción  | Tipo de Dato | Restricción |
|---------------------|--|--------------|-------------|
| <b>Área</b>         | Contiene el área de la informática en el que tiene conocimiento el postulante. | Varchar(64)  | Null        |
| <b>Conocimiento</b> | Contiene la descripción del conocimiento en informática.                       | Varchar(64)  | Null        |
| <b>Nivel</b>        | Contiene el nivel de conocimiento del postulante.                              | Varchar(824) | Null        |

**Elaborado por:** Angelo Andy & Andrés Vargas

### Clase experiencia

Esta tabla contiene la información de las experiencias laborales que posee el postulante.

**Tabla 17.** Especificación clase experiencia

| Campo          | Descripción   | Tipo de Dato | Restricción |
|----------------|---|--------------|-------------|
| <b>Empresa</b> | Contiene el nombre de la empresa en donde trabajo anteriormente | Varchar(64)  | Null        |
| <b>Puesto</b>  | Contiene el nombre del puesto que tenía en su trabajo anterior. | Varchar(64)  | Null        |

**Tabla 17.** Especificación clase experiencia (continuación).

|                |  |              |      |
|----------------|--|--------------|------|
| <b>Área</b>    | Contiene el nombre del área en la que trabajo.                 | Varchar(64)  | Null |
| <b>Detalle</b> | Contiene el detalle de sus labores realizadas en dicho puesto. | Varchar(128) | Null |
| <b>Tiempo</b>  | Contiene el tiempo que laboró en la empresa.                   | Varchar(20)  | Null |

**Elaborado por:** Angelo Andy & Andrés Vargas

### **Clase referencias**

Esta tabla contiene la información de las referencias del postulante.

**Tabla 18.** Especificación clase referencias

| <b>Campo</b>     | <b>Descripción</b>                               | <b>Tipo de Dato</b> | <b>Restricción</b> |
|------------------|--|---------------------|--------------------|
| <b>Nombres</b>   | Contiene los nombres de la referencia.           | Varchar(24)         | Null               |
| <b>Apellidos</b> | Contiene los apellidos de la referencia.         | Varchar(24)         | Null               |
| <b>Teléfono</b>  | Contiene el teléfono de la referencia.           | Varchar(15)         | Null               |
| <b>Relación</b>  | Contiene la relación que tuvo con la referencia. | Varchar(64)         | Null               |

**Elaborado por:** Angelo Andy & Andrés Vargas

### 3.2.2.2. Modelo dinámico.

Identifica los aspectos que son temporales sobre funcionamiento del sistema, a través de la secuencia de operaciones en el tiempo.

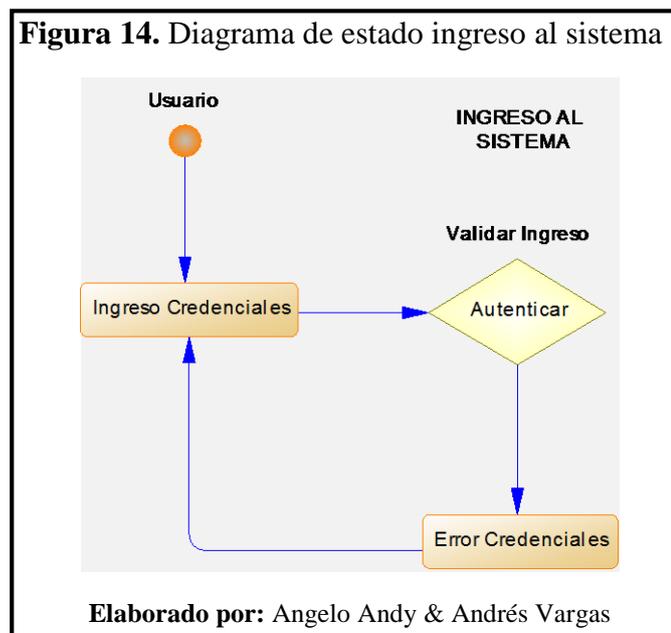
#### Diagramas de estado

Durante el transcurso del desarrollo del sistema los objetos pueden cambiar de estados o mantenerse en un estado inicial, este estado está en función del tiempo con intervalos entre dos sucesos recibidos por un objeto.

#### Diagramas de estados generales

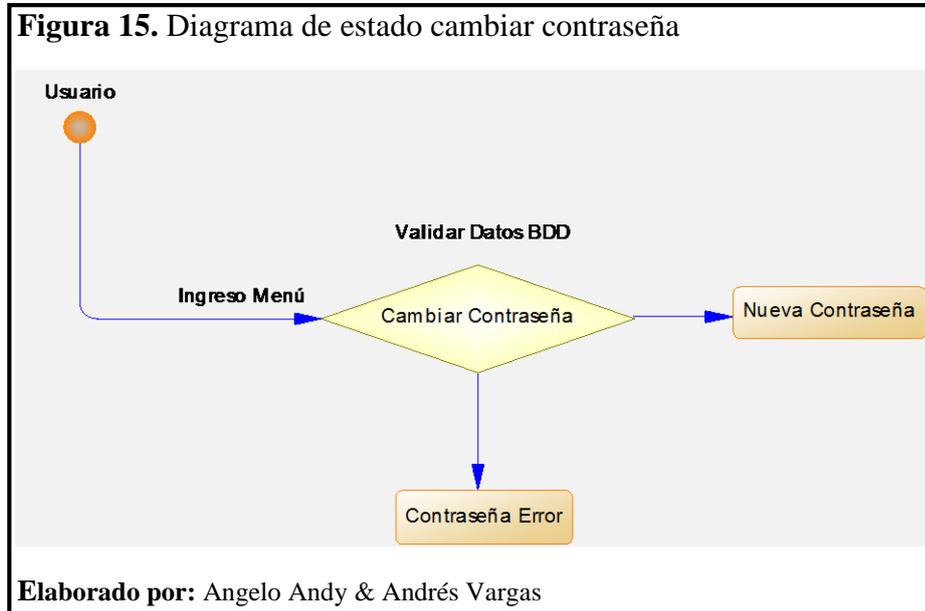
##### Ingreso al sistema

Detalla el proceso que realiza el sistema para autorizar o no el ingreso de un usuario, después de validar los datos proporcionados.



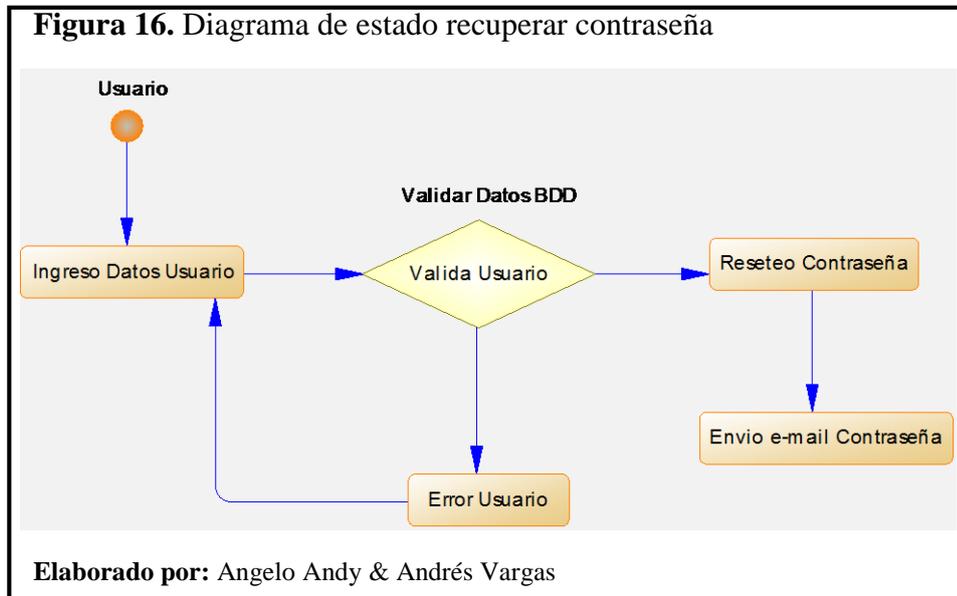
### Cambiar contraseña

Detalla el proceso que realiza el sistema cuando un usuario desea cambiar la contraseña.



### Recuperar contraseña

Detalla el proceso que realiza el sistema cuando un usuario desea recuperar la contraseña y el login para acceder.

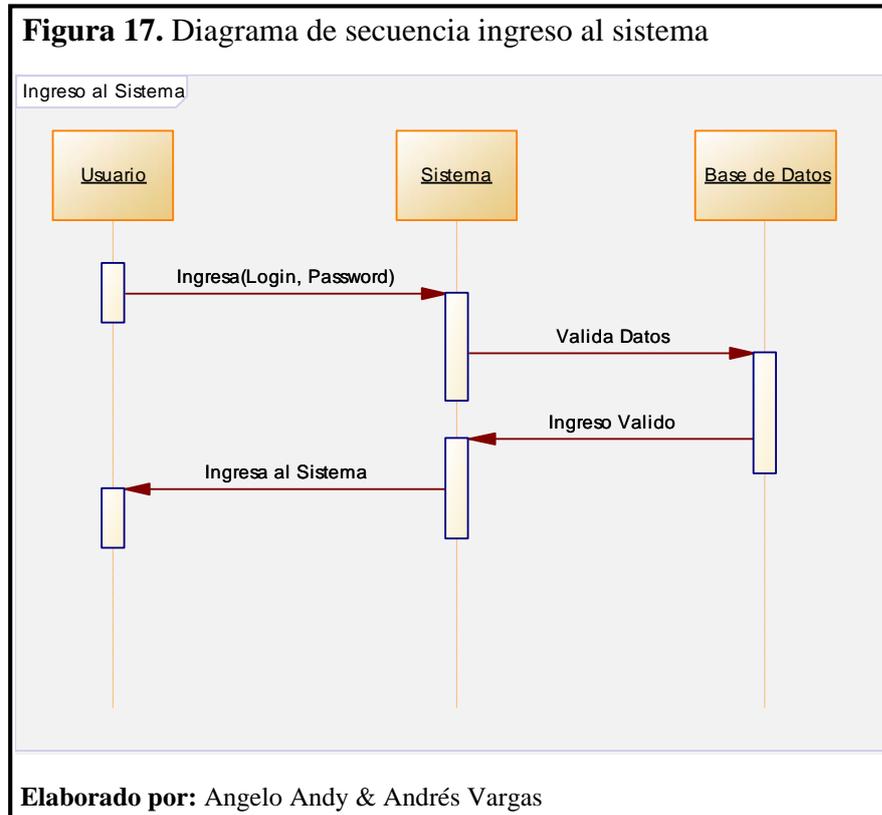


La información detallada de cada diagrama de estado se lo puede encontrar en el anexo 1: Manual técnico.

## Diagramas de secuencia

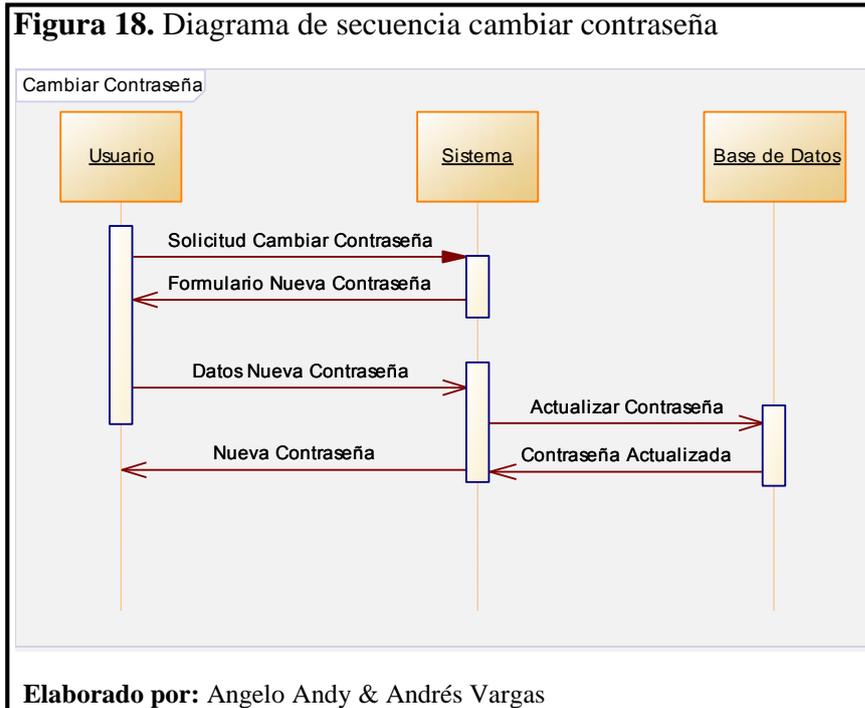
### Ingreso al sistema

Especifica el proceso que realizan los usuarios en el sistema para validar los datos proporcionados y mediante un algoritmo interno verifique que sean los que corresponden a los que se encuentran registrados en la base de datos.



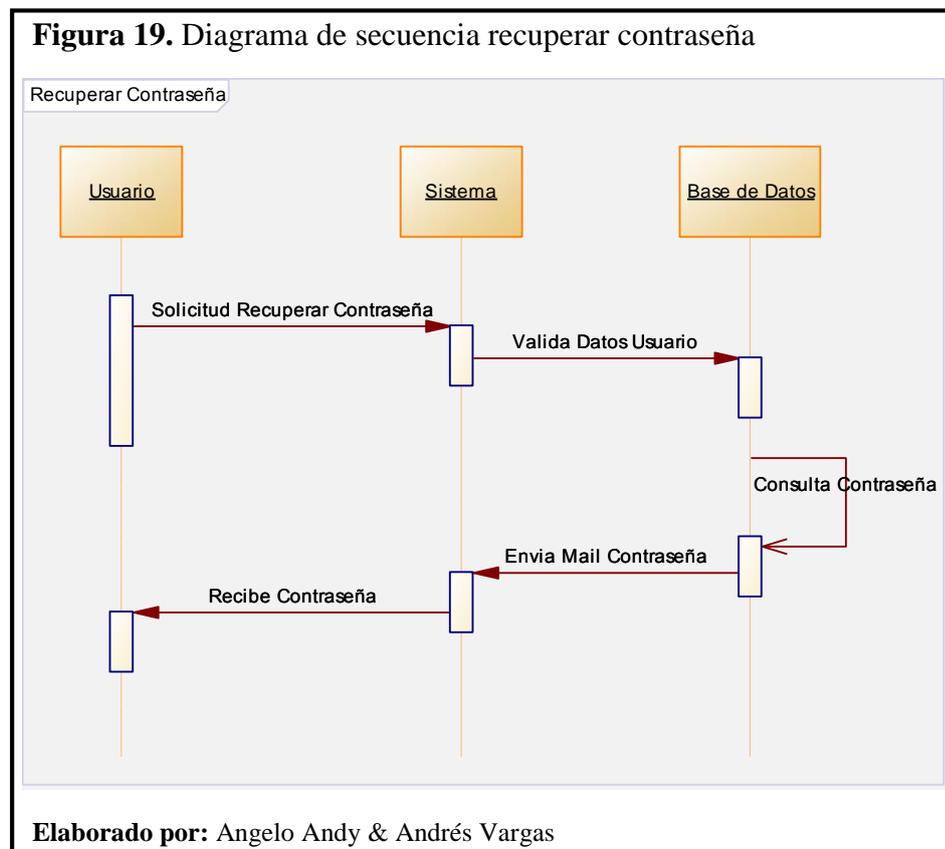
### Cambiar contraseña

Especifica el proceso que realizan los usuarios en el sistema para cambiar la contraseña de su usuario en el sistema.



### Recuperar contraseña

Especifica el proceso que debe realizar el usuario para recuperar su contraseña.



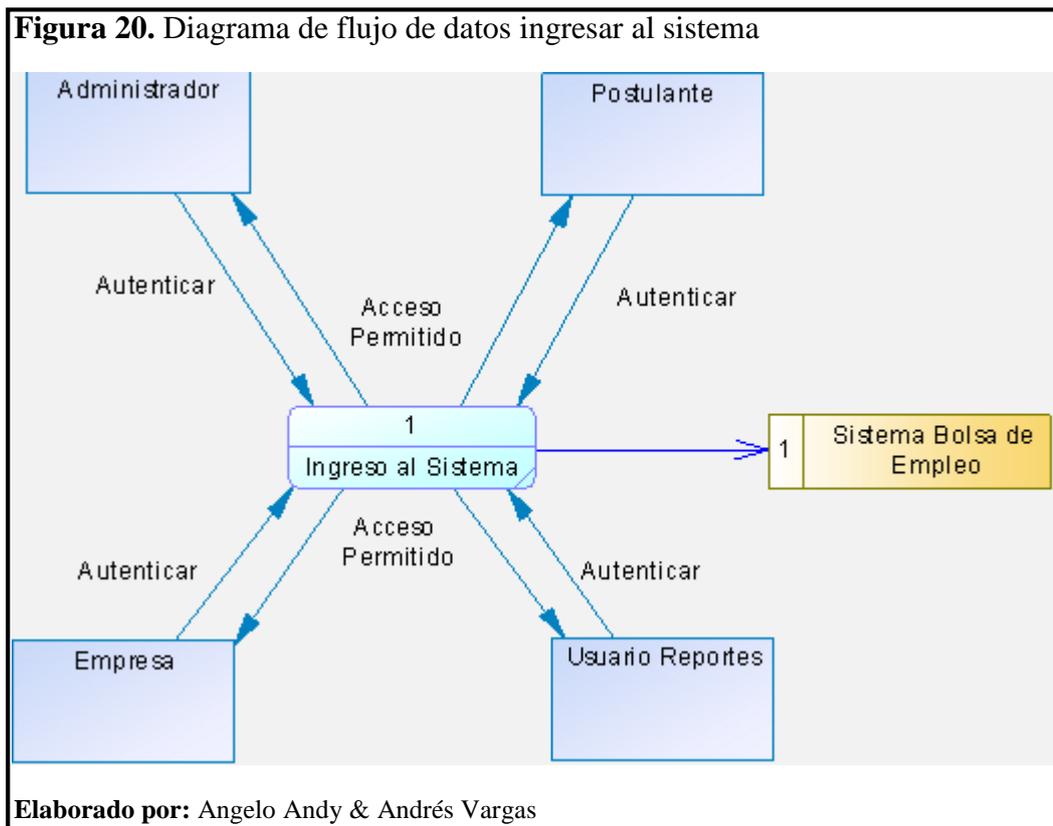
La información detallada de cada diagrama de secuencia se lo puede encontrar en el anexo 1: Manual técnico.

### 3.2.2.3. Modelo funcional

En la metodología OMT este es el último modelo en el cual se representa lo que sucede dentro del funcionamiento del sistema, es decir, la forma en que se transforman sus datos, así como también las dependencias que posee con otros valores y las funciones que los relacionan entre sí.

#### Ingresar al sistema

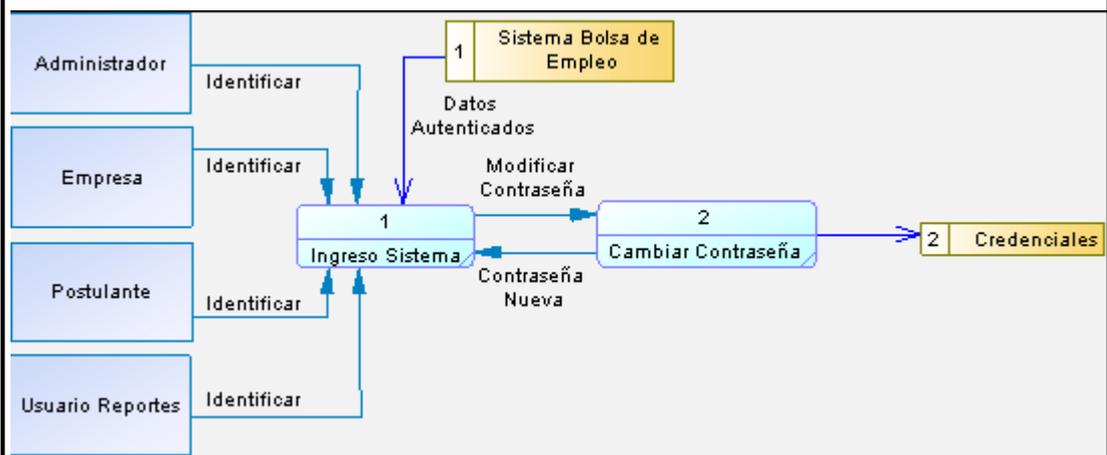
En el diagrama se puede observar el flujo que sigue un usuario para ingresar al sistema.



#### Cambiar contraseña

En el diagrama se puede observar el flujo que sigue un usuario para cambiar la contraseña en el sistema.

**Figura 21.** Diagrama de flujo de datos cambiar contraseña

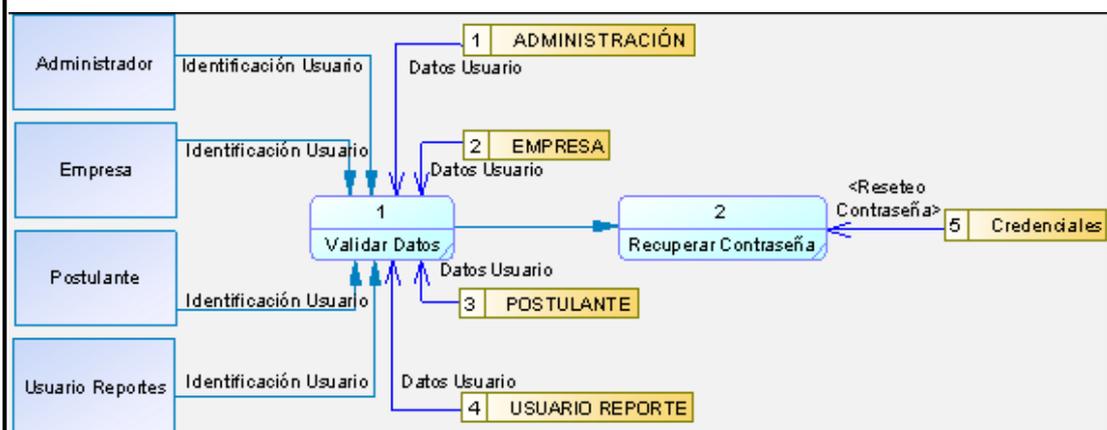


Elaborado por: Angelo Andy & Andrés Vargas

### Diagrama de flujo de datos recuperar contraseña

En el diagrama se puede observar el flujo que sigue un usuario para cambiar su contraseña en el sistema.

**Figura 22.** Diagrama de flujo de datos recuperar contraseña

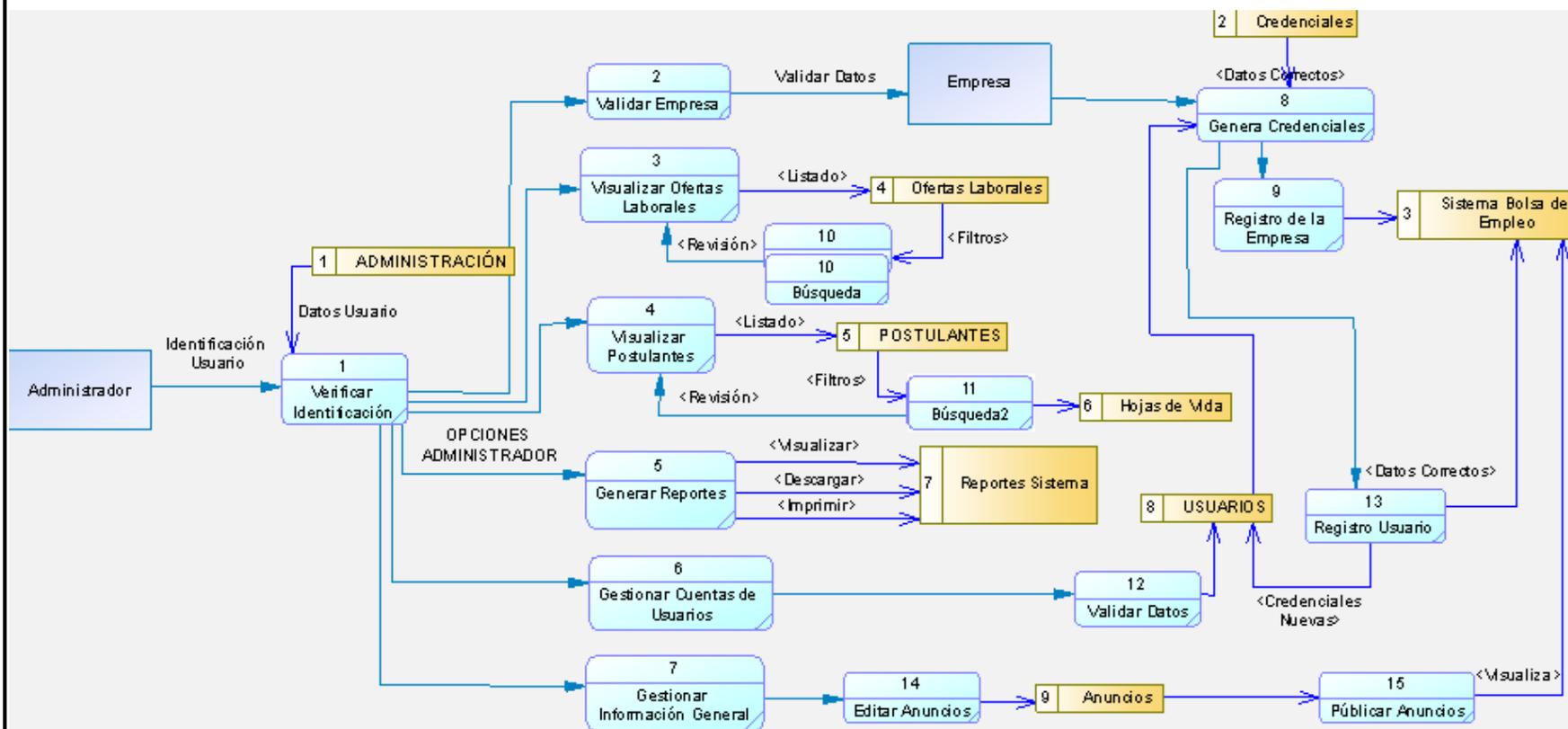


Elaborado por: Angelo Andy & Andrés Vargas

## Usuario administrador

En el diagrama se puede observar el flujo que sigue el administrador para ejecutar todas las opciones con las que cuenta este usuario en el sistema.

**Figura 23:** Diagrama de flujo de datos administrador

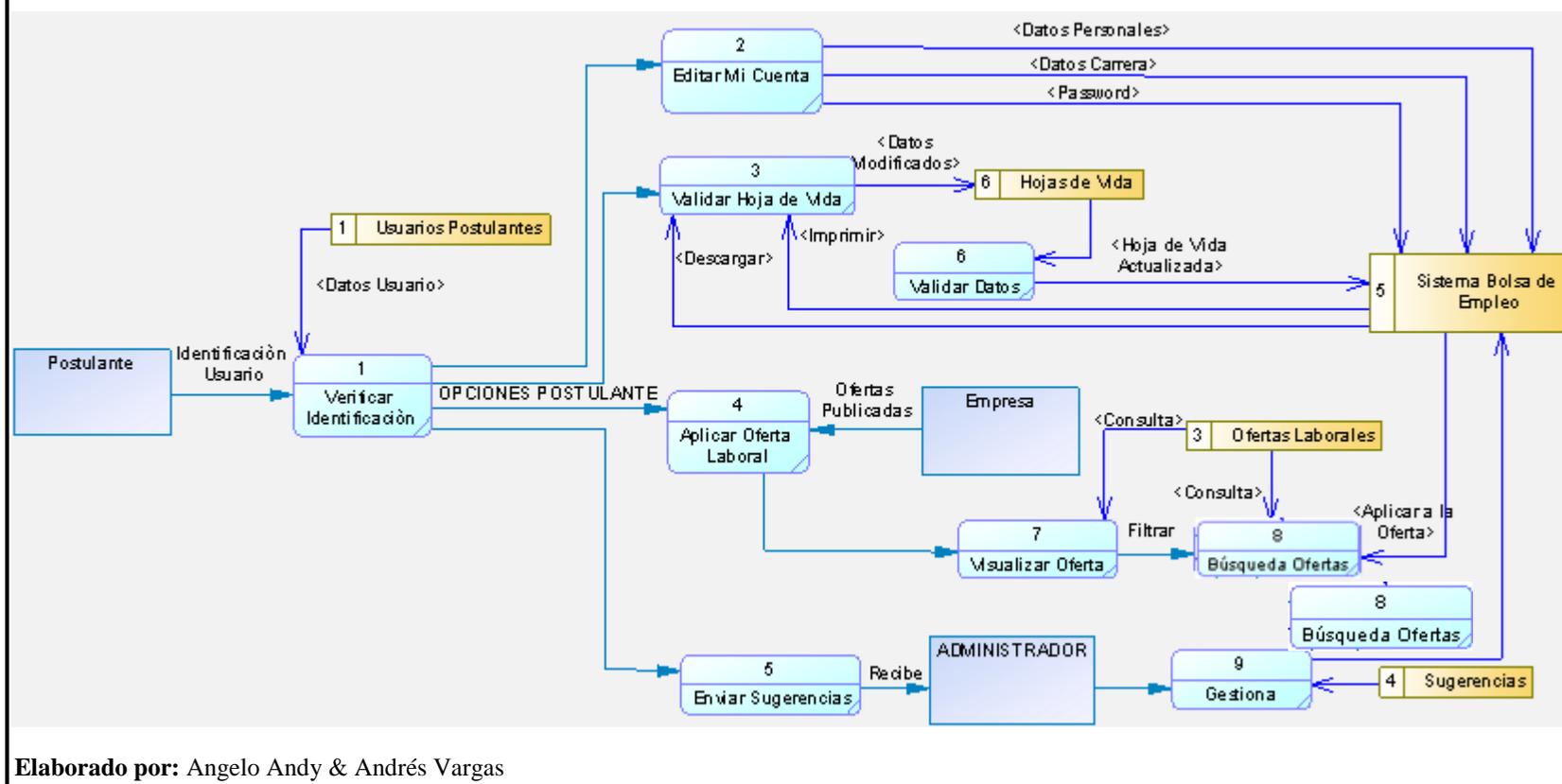


Elaborado por: Angelo Andy & Andrés Vargas

## Usuario postulante

En el diagrama se puede observar el flujo que sigue postulante para ejecutar todas las opciones con las que cuenta el usuario en el sistema.

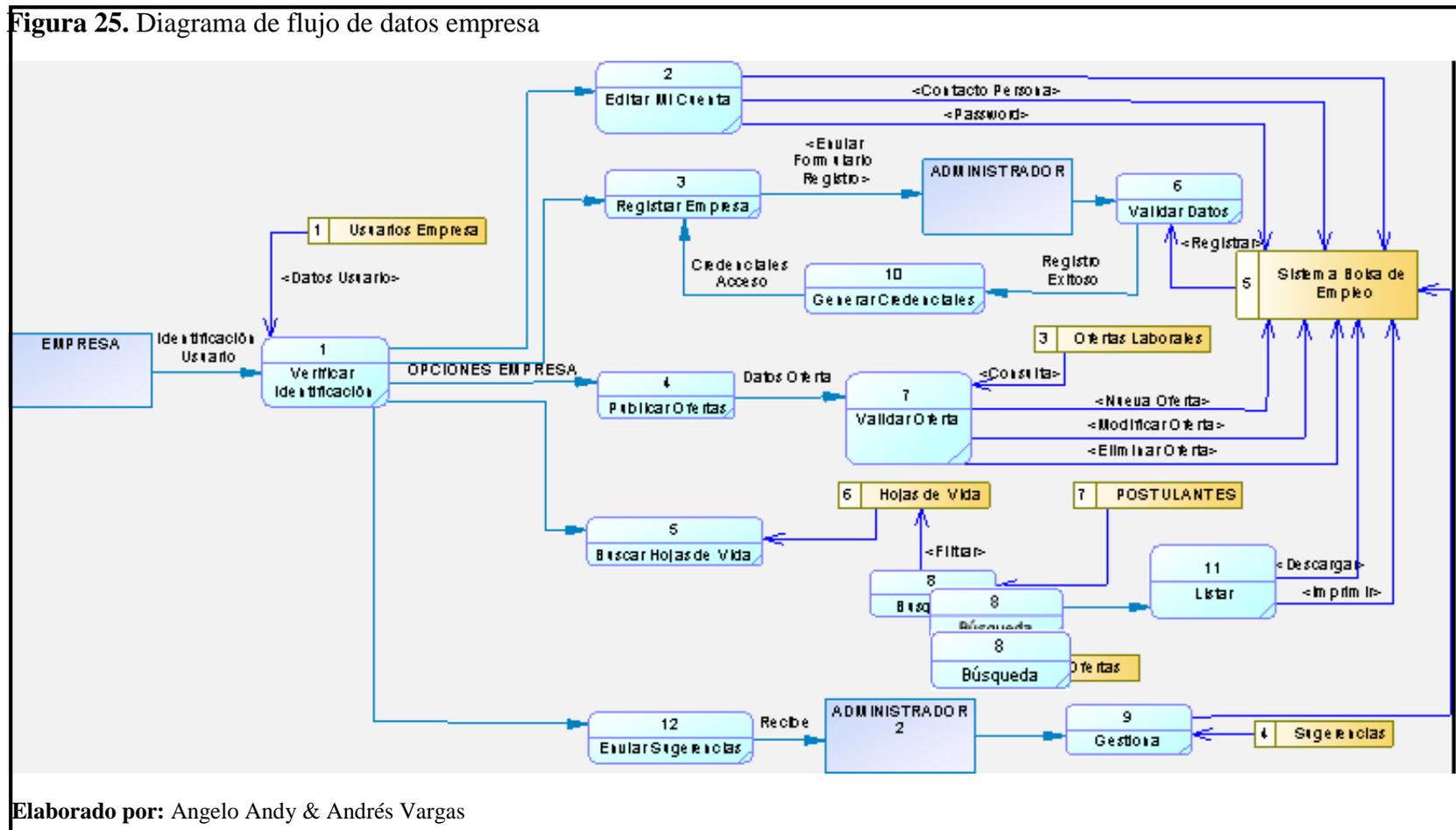
**Figura 24.** Diagrama de flujo de datos postulante



## Usuario empresa

En el diagrama se puede observar el flujo que sigue la empresa para ejecutar todas las opciones con las que cuenta el usuario en el sistema.

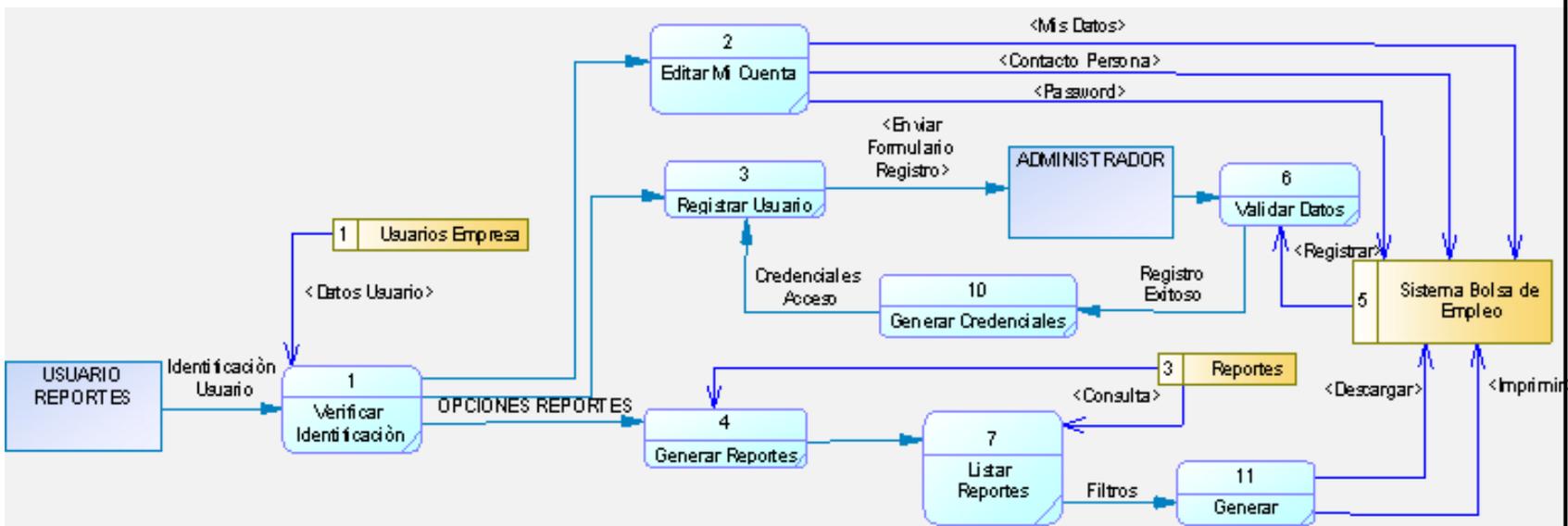
Figura 25. Diagrama de flujo de datos empresa



## Usuario reportes

En el diagrama se puede observar el flujo que sigue el usuario reportes para ejecutar todas las opciones con las que cuenta en el sistema.

**Figura 26.** Diagrama de flujo de datos usuario reportes



Elaborado por: Angelo Andy & Andrés Vargas

## CAPÍTULO 4

### DESARROLLO DEL PORTAL WEB Y PRUEBAS

#### 4.1. Programación de módulos

En este capítulo se detallarán los métodos más importantes dentro del proyecto, a su vez cada método contará con su respectivo código fuente y su breve explicación.

##### 4.1.1. Método login.

**Figura 27.** Código fuente método login

```
public void AutenticarLogin(ActionEvent actionEvent){
    RequestContext context = RequestContext.getCurrentInstance();
    FacesMessage msg;
    LoginServicio service = new LoginServicio();
    cred = service.buscarporLogin(login, pass);
    String ruta = "/WebEmpleo/faces/secured/LoginUser.xhtml";

    if(cred != null && cred.getCrEstado().equals("Activo")){
        setCred(cred);
        loggedIn = true;
        msg = new
    FacesMessage(FacesMessage.SEVERITY_INFO,"Bienvenido","user");

        FacesContext.getCurrentInstance().getExternalContext().getSessionMap().put("user", cred.getIdPersona().getIdPersona());
        cargam();
        CrearMenu();
    }
    else {
        loggedIn = false;
        msg = new FacesMessage(FacesMessage.SEVERITY_ERROR,"Error de
    Logeo","Credenciales Inválidas");
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
    context.addCallbackParam("loggedIn", loggedIn);
    context.addCallbackParam("ruta", ruta);
}
```

**Elaborado por:** Angelo Andy & Andrés Vargas

Esta función permite validar las credenciales que el usuario ingresa en la sección de Login, las compara con las credenciales almacenadas en la base de datos, al igual que el estado de su cuenta "Activo". Si las credenciales son válidas permite el acceso, en caso contrario devuelve un mensaje de "Error de Login".

También invoca al método "CrearMenu()" el cual permite desplegar el menú de usuario dependiendo de su rol dentro del sistema.

#### 4.1.2. Método para la creación de menú de usuario.

Figura 28. Código fuente crear menú

```
public void CrearMenu(){
    model = new DefaultMenuModel();
    Submenu submenu = new Submenu();
    submenu.setLabel("Menú Principal");
    for(int i=0;i<Menu_list.size();i++){
        MenuItem item = new MenuItem();
        item.setIcon("ui-icon ui-icon-triangle-l-e");
        item.setValue(Menu_list.get(i));

        if(cred.getIdRol().getRoNombreRol().equals("ESTUDIANTE")){
            item.setUrl("/faces/secured/postulantes/"+Menu_list.get(i)
                .toString().replaceAll(" ", "").toLowerCase()+"_xhtml");
        }

        if(cred.getIdRol().getRoNombreRol().equals("ADMINISTRADOR") ||
            cred.getIdRol().getRoNombreRol().equals("SUBADMIN")){

            item.setUrl("/faces/secured/admin/"+Menu_list.get(i).toStr
                ing().replaceAll(" ", "").toLowerCase()+"_xhtml");
        }

        if(cred.getIdRol().getRoNombreRol().equals("EMPRESA")){

            item.setUrl("/faces/secured/empresas/"+Menu_list.get(i).to
                String().replaceAll(" ", "").toLowerCase()+"_xhtml");
        }

        if(cred.getIdRol().getRoNombreRol().equals("REPORTE")){

            item.setUrl("/faces/secured/usrepo/"+Menu_list.get(i).toSt
                ring().replaceAll(" ", "").toLowerCase()+"_xhtml");
        }
        submenu.getChildren().add(item);
        model.addSubmenu(submenu);
    }
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite cargar el menú de usuario dependiendo de su rol dentro del sistema, para esto se realiza una consulta a la base de datos utilizando el objeto "Menu\_list".

#### 4.1.3. Método dar de alta a un usuario.

**Figura 29.** Código fuente dar de alta usuario

```
public void cambiaActivo() throws EmailException{

    GestionCuentas ue = new GestionCuentas();
    FacesMessage msg;
    pendSelected.setCrEstado("Activo");
    String estado = "Activo";
    fecha = new Date();
    maild = ue.mailDestino(this.pendSelected.getCrLogin());
    if(ue.editEstadoAlta(this.pendSelected,estado,fecha)){
        enviaMail();
        msg = new FacesMessage(FacesMessage.SEVERITY_INFO,"Cuenta
        Actualizada","La cuenta se ha dado de alta");
        cargaPendientes();
    }
    else{
        msg = new FacesMessage(FacesMessage.SEVERITY_ERROR,"Error al
        Actualizar","La cuenta no se ha modificado");
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite al usuario administrador dar de alta una cuenta que se encuentre en estado "Pendiente". Una vez que el estado cambia a Activo se envía una notificación vía mail mediante el método "enviaMail()", también se almacena la fecha en la cual la cuenta ha sido activada.

#### 4.1.4. Método creación de anuncios.

**Figura 30.** Código fuente nuevo anuncio

```
public void nuevoAnuncio(){
    AnunciosService as = new AnunciosService();
    FacesMessage msg;
    fecha = new Date();
    String id;
    id = (String)
    FacesContext.getCurrentInstance().getExternalContext().getSessionMap(
    ).get("user");

    if(as.NuevoAnuncio(id, titulo, detalle, fecha, posicion1)){...
        titulo = null;
        detalle = null;
        msg = new FacesMessage(FacesMessage.SEVERITY_INFO,"Anuncio
        creado","El anuncio se guardó correctamente");
        cargaAnuncios();
    }
    else{
        msg = new FacesMessage(FacesMessage.SEVERITY_ERROR,"Error
        al Actualizar","La información no se guardó
        correctamente");
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite al usuario administrador crear anuncios o avisos para la página principal, tomando como parámetros: el título, detalle, fecha de creación y posición del anuncio. Si el anuncio fue creado de manera correcta se muestra un mensaje de “Anuncio Creado”, en caso contrario se mostrará un mensaje de “Error al Actualizar”.

#### 4.1.5. Método login empresa registrada.

**Figura 31.** Código fuente login empresa registrada

```
public String creaLogin(String empresa){
    String cadena;
    String login;
    Random r = new Random();
    int valor = r.nextInt(500);
    cadena = empresa.toLowerCase().replaceAll(" ", "");
    login = cadena+""+valor;
    return login;
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite generar automáticamente un login o nombre de usuario para las empresas que se registran en el sistema, para la creación de este login se toma como parámetro el nombre de la empresa, se elimina los espacios en blanco y finalmente se agrega un número randómico.

#### 4.1.6. Método recuperar contraseña.

**Figura 32.** Código fuente recuperar contraseña

```
public String recuperaPass(){
    PassService paSer = new PassService();
    String cuenta=null;
    cred = paSer.buscaPass(login);
    FacesMessage msg=null;
    if(cred == null){
        msg = new
            FacesMessage(FacesMessage.SEVERITY_ERROR, "Error", "No
            existe este Nombre de Usuario");
        login = "";
        FacesContext.getCurrentInstance().addMessage(null, msg);
    }
    else{
        cuenta = cred.getCxLogin();
    }
    return cuenta;
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Método que permite encontrar la cuenta perteneciente al nombre de usuario que se ingresa en el formulario de recuperación de contraseña.

**Figura 33.** Código fuente enviar mail

```
public void enviaMail() throws EmailException{
    conectaEmail();
    FacesMessage msg=null;
    PassService ps = new PassService();
    String maild,cuenta;
    cuenta= recuperaPass();
    maild=ps.mailDestino(cuenta);
    if(maild != null){
        String titulo = "Recuperación Password - Bolsa de Empleo
UPS";
        String mensaje = "Gracias por usar la Bolsa de Empleo de
la UPS. \n Su Nombre de usuario es: "+this.cred.getCrLogin()+"\n Su
Password es: "+this.cred.getCrPassword();
        try {
            Email email = new SimpleEmail();
            email = conectaEmail();
            email.setSubject(titulo);
            email.setMsg(mensaje);
            email.addTo(maild);
            String respuesta = email.send();
            msg = new
FacesMessage(FacesMessage.SEVERITY_INFO,"Password Enviado","La
contraseña se envió correctamente");
        } catch (EmailException ex) {

        }

        Logger.getLogger(GestionCuentasBean.class.getName()).log(Level.SEVERE
, null, ex);
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
}
}
```

**Elaborado por:** Angelo Andy & Andrés Vargas

Método con el cual se envía las credenciales recuperadas vía mail al correo electrónico del usuario para que pueda ingresar nuevamente al sistema.

#### 4.1.7. Método generar reporte empresas vs ofertas.

**Figura 34.** Código fuente generar reporte

```
public void crearReporteEmpresa() {
    categoryModel = new CartesianChartModel();
    ChartSeries empresas = new ChartSeries();
    empresas.setLabel("Empresas");
    ReporteServicio rs = new ReporteServicio();
    ListEmpresa=rs.reportOfEmpresaFecha(fecha_in, fecha_fin);
    ListCount = rs.reportOfEmpresaFechaCount(fecha_in,
fecha_fin);

    for(int i=0;i<ListEmpresa.size();i++){
        for(int j=0;j<ListCount.size();j++)
            empresas.set(ListEmpresa.get(i), (Number)
ListCount.get(j));
    }

    categoryModel.addSeries(empresas);
    cargaEmpOfe();
}
}
```

**Elaborado por:** Angelo Andy & Andrés Vargas

Este método permite crear un reporte estadístico en el cual se muestra las empresas con sus respectivos números de ofertas dentro de un lapso de tiempo, esto quiere decir entre 2 fechas dadas. Para esto se consulta todas las empresas que se encuentran el

lapso de fechas determinado por el usuario, esto mediante el método “*reportOfEmpresaFecha*”, seguidamente se realiza el conteo de ofertas realizadas, mediante el método “*reportOfEmpresaFechaCount*”, finalmente se envía esta información a graficar utilizando la clase “*categoryModel*” que para este ejemplo devolverá un gráfico de barras.

#### 4.1.8. Método crear nueva oferta laboral.

**Figura 35.** Código fuente nueva oferta laboral

```
public void nuevaOferta(){
    ofertasServicio os = new ofertasServicio();
    FacesMessage msg;
    String tipoOf = "completa";
    fecha = new Date();
    if(os.NuevaOferta(empresa.getIdEmpresa(), titulo, fecha,
        area, subarea, tipo, salario, lugar, detalles, tipoOf,
        vigencia)){
        titulo = null;
        fecha = null;
        area = null;
        subarea = null;
        tipo = null;
        salario = null;
        lugar = null;
        detalles = null;
        msg = new FacesMessage(FacesMessage.SEVERITY_INFO, "Oferta
        creada", "La oferta se guardó correctamente");
        cargaOfertas();
    }
    else{
        msg = new FacesMessage(FacesMessage.SEVERITY_ERROR, "Error
        al Actualizar", "La información no se guardó
        correctamente");
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite a las empresas crear nuevas ofertas de empleo, este procedimiento se lo realiza mediante el formulario de nueva Oferta, disponible en el menú del usuario empresa, aquí se toma los valores ingresados en el formulario y se los almacena en la base de datos mediante el uso del método “*NuevaOferta*”.

#### 4.1.9. Método búsqueda de postulantes (hojas de vida).

**Figura 36.** Código fuente buscar hojas de vida

```
public List<Object[]> busquedaI(){
    ofertasServicio os = new ofertasServicio();
    String deta = detalleb.replace(",","").toLowerCase();
    resultado = os.busquedaIntel(salariob, deta, carrera,
    tipoPos);
    return resultado;
}
```

Elaborado por: Angelo Andy & Andrés Vargas

Este método permite a la empresa realizar una búsqueda de posibles postulantes mediante un filtrado de datos. Aquí se toma los datos recogidos por el formulario y se los envía al método “*busquedaIntel*”, el cual se encarga de realizar la búsqueda en la base de datos y devolver las respectivas coincidencias.

#### 4.1.10. Método crear sugerencias.

**Figura 37.** Código fuente nueva sugerencia

```
public void nuevaSugerencia(){
    SugerenciaServicio ss = new SugerenciaServicio();
    FacesMessage msg;
    fecha = new Date();
    String id;
    id = (String)
FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("user");

    if(ss.NuevaSugerencia(id, titulo, comentario, fecha)){
        msg = new
        FacesMessage(FacesMessage.SEVERITY_INFO,"Sugerencia
        enviada","La sugerencia se ha enviado correctamente");
        cargaSugerencias();
        titulo = null;
        comentario = null;
    }
    else{
        msg = new FacesMessage(FacesMessage.SEVERITY_ERROR,"Error
        al Actualizar","La sugerencia no se ha enviado
        correctamente");
    }
    FacesContext.getCurrentInstance().addMessage(null, msg);
}
```

**Elaborado por:** Angelo Andy & Andrés Vargas

Este método permite a los usuarios enviar sugerencias al administrador del sistema en este caso el Departamento de Bienestar Estudiantil, para esto se llena el respectivo formulario, una vez con todos los datos necesarios se envían al método “*NuevaSugerencia*”, el cual es el encargado de almacenarla en la base de datos y posteriormente mostrarla en la cuenta del administrador.

## **4.2. Pruebas de software**

El proceso de pruebas a un software es algo fundamental a la hora de detectar fallas o errores, con este proceso se pone a prueba la calidad del software basado en conceptos como estabilidad, escalabilidad, eficiencia y seguridad, con el fin de ajustar de la mejor manera el funcionamiento del sistema para que el usuario que lo use este satisfecho con el producto final.

En el sistema Bolsa de Empleo se ejecutó las siguientes pruebas:

- Caja negra
- Carga y estrés

### **4.2.1. Pruebas de caja negra.**

Las pruebas de caja negra realizadas proporcionan una visión del funcionamiento del software sin tener en cuenta su funcionamiento interno, con estas pruebas se puede revisar los requerimientos con los que fue concebido el sistema, con el propósito de detectar insatisfacción de los mismos.

Los principales objetivos de las pruebas son:

- Proporcionar un sistema con un alto nivel de calidad y productividad.
- Reducción de costos en mantenimiento, mediante el seguimiento de estándares de programación, calidad en la documentación del sistema.
- Obtener información verdadera acerca de los errores o fallas que tiene el sistema, para que pueda servir como retroalimentación para mejorar los procesos y la parte de programación de los mismos.
- Demostrar que las entradas y salidas del software son totalmente operativas.
- Revisar que los requerimientos funcionales estén completamente implementados en el sistema.

Las pruebas se realizaran con cada usuario siguiendo los procesos que éste ejecuta en el sistema.

#### 4.2.1.1. Caso de prueba ingreso al sistema.

Se ingresa el requerimiento en el cual se validara la autenticación de un usuario en el sistema.

**Tabla 19.** Requerimiento de prueba: Ingreso al sistema

| Id Req | Requerimiento              | Id Caso | Caso de prueba     |
|--------|----------------------------|---------|--------------------|
| Req01  | Validar ingreso al sistema | CP01    | Ingreso al sistema |

**Elaborado por:** Angelo Andy & Andrés Vargas

**Tabla 20.** Caso de prueba: Ingreso al sistema

| Id Caso | Entrada                  | Resultado   | Conclusiones   |
|---------|--------------------------|---|--|
| CP01    | Ingresar usuario válido  | <p>El acceso al sistema es correcto para el usuario administrador.</p>  <p>Se verifica el acceso a las páginas propias del usuario.</p>  | El algoritmo de validación de usuarios en el sistema es correcto, ya que se ingresaron valores válidos para el ingreso |
|         | Ingresar usuario erróneo | <p>La validación de ingreso al sistema con un usuario incorrecto.</p>   | Validación correcta ya que no deja que entre un usuario que no esté habilitado   |

**Elaborado por:** Angelo Andy & Andrés Vargas

#### 4.2.1.2. Caso de prueba cambiar contraseña.

Se ingresa el requerimiento en el cual se valida el cambio de contraseña de un usuario en el sistema.

**Tabla 21.** Requerimiento de prueba: Cambiar contraseña

| Id Req | Requerimiento                   | Id Caso | Caso de prueba     |
|--------|---------------------------------|---------|--------------------|
| Req01  | Validar el cambio de contraseña | CP01    | Cambiar contraseña |

**Elaborado por:** Angelo Andy & Andrés Vargas

**Tabla 22.** Caso de prueba: Cambiar contraseña

| Id Caso | Entrada                        | Resultado  | Conclusiones   |
|---------|--------------------------------|--|--|
| CP01    | Formulario cambiar contraseña  | <p>Cambio de contraseña correcto.</p>     | El proceso para cambiar la contraseña funciona correctamente   |
|         | Contraseña anterior incorrecta | <p>Cambio de contraseña incorrecto.</p>  | La validación del cambio de contraseña es correcta ya que verifica la contraseña anterior para cambiar a una nueva |

**Elaborado por:** Angelo Andy & Andrés Vargas

### 4.2.1.3. Caso de prueba recuperar contraseña.

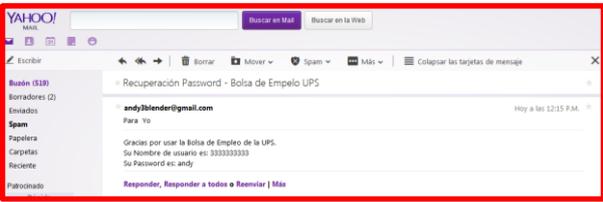
Se ingresa el requerimiento en el cual se valida el proceso de recuperar la contraseña de un usuario, y la recepción del mail en el correo personal con las credenciales recuperadas.

**Tabla 23.** Requerimiento de prueba: Recuperar contraseña

| Id Req | Requerimiento   | Id Caso | Caso de prueba                   |
|--------|---|---------|----------------------------------|
| Req01  | Validar recuperar contraseña  | CP01    | Recuperar contraseña             |
|        | Validar que se envió correctamente el mail con las credenciales al correo electrónico | CP02    | Validar credenciales recuperadas |

Elaborado por: Angelo Andy & Andrés Vargas

**Tabla 24.** Caso de prueba: Recuperar contraseña

| Id Caso | Entrada  | Resultado  | Conclusiones   |
|---------|--|--|--|
| CP01    | Formulario recuperar contraseña  | <p>La validación del usuario es correcta, se envía un mail al correo personal con las credenciales de acceso.</p>  | El funcionamiento del algoritmo de recuperar credenciales y el envío de mail funciona correctamente en el sistema. |
| CP02    | Validar credenciales enviadas al correo electrónico personal del usuario | <p>Se recibe el mail con las credenciales del usuario.</p>   | Las credenciales recibidas en el correo electrónico personal son correctas   |

Elaborado por: Angelo Andy & Andrés Vargas

#### 4.2.1.4. Caso de prueba validar menú por usuario.

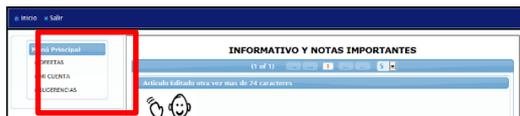
Se ingresa el requerimiento en el cual se valida la visualización correcta del menú y el funcionamiento correcto de los enlaces a cada una de las paginas por cada uno de los usuarios del sistema.

**Tabla 25.** Requerimiento de prueba: Validar menú por usuario

| Id Req | Requerimiento                      | Id Caso | Caso de prueba           |
|--------|------------------------------------|---------|--------------------------|
| Req01  | Validar menú usuario administrador | CP01    | Visualizar menú correcto |
| Req02  | Validar menú usuario postulante    | CP01    | Visualizar menú correcto |
| Req03  | Validar menú usuario empresa       | CP01    | Visualizar menú correcto |
| Req04  | Validar menú usuario reportes      | CP01    | Visualizar menú correcto |

**Elaborado por:** Angelo Andy & Andrés Vargas

**Tabla 26.** Caso de prueba: Visualizar menú correcto por usuario

| Id Caso    | Entrada                    | Resultado   | Conclusiones   |
|------------|----------------------------|---|--|
| Req01-CP01 | Menú usuario administrador | Menú generado por el sistema para el usuario administrador.<br> | El menú con todas las opciones para el usuario Administrador se genera correctamente |
| Req02-CP01 | Menú usuario postulante    | Menú generado por el sistema para el usuario postulante.<br>    | El menú con todas las opciones para el usuario Postulante se genera correctamente    |
| Req3-CP01  | Menú usuario empresa       | Menú generado por el sistema para el usuario empresa.<br>       | El menú con todas las opciones para el usuario empresa                               |

**Tabla 26.** Caso de prueba: Visualizar menú correcto por usuario (continuación).

|            |                       |  |   |
|------------|-----------------------|--|---|
| Req04-CP01 | Menú usuario reportes | Menú generado por el sistema para el usuario reportes.<br> | El menú con todas las opciones para el usuario reportes se genera correctamente |
|------------|-----------------------|--|---|

**Elaborado por:** Angelo Andy & Andrés Vargas

El detalle de todos los casos de prueba se encuentra en el anexo 1 (Manual Técnico)

#### 4.2.2. Pruebas de carga y estrés.

En estas pruebas de carga y estrés se determinara lo rápido que realiza una tarea el sistema de Bolsa de Empleo en condiciones particulares de trabajo, se espera demostrar que el sistema cumple con los criterios de rendimiento óptimos para alcanzar un buen nivel de rendimiento.

##### 4.2.2.1. Primera fase de pruebas.

La primera estuvo conformada por un servidor con las siguientes características:

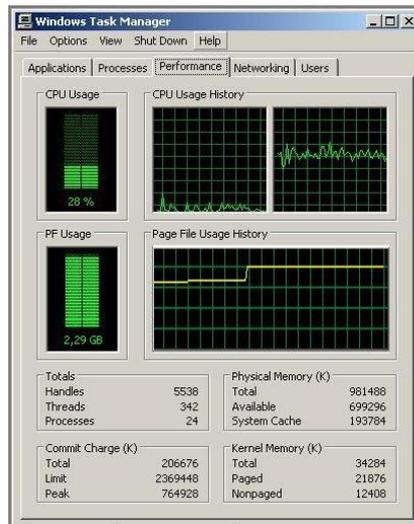
- **Sistema operativo:** Windows Server 2003
- **Memoria RAM:** 2.5GB
- **Procesador:** Intel Core I7 3.4GHz

Esta primera fase se la realizó con un grupo de 15 estudiantes, los resultados obtenidos mediante el software de monitoreo (JMeter), transcurrido 60 minutos desde el inicio de las pruebas, son los que a continuación se detallan:

#### Uso de memoria

La figura 38 muestra que transcurrido el tiempo antes mencionado, la memoria RAM llegó a su máximo valor, esto debido a que su capacidad era muy baja y no soportó varias peticiones al mismo tiempo, es por esta razón que el servidor dejó de responder correctamente lo cual conllevó a una caída y paralización del servicio web.

**Figura 38. Uso de memoria del servidor (Primera fase)**



Elaborado por: Angelo Andy & Andrés Vargas

**Peticiones realizadas**

Como se observa en la figura 39 se tiene la lista de peticiones las cuales fueron atendidas sin ningún problema hasta el instante de la caída del servicio web. Una vez paralizado dicho servicio las peticiones automáticamente fueron rechazadas.

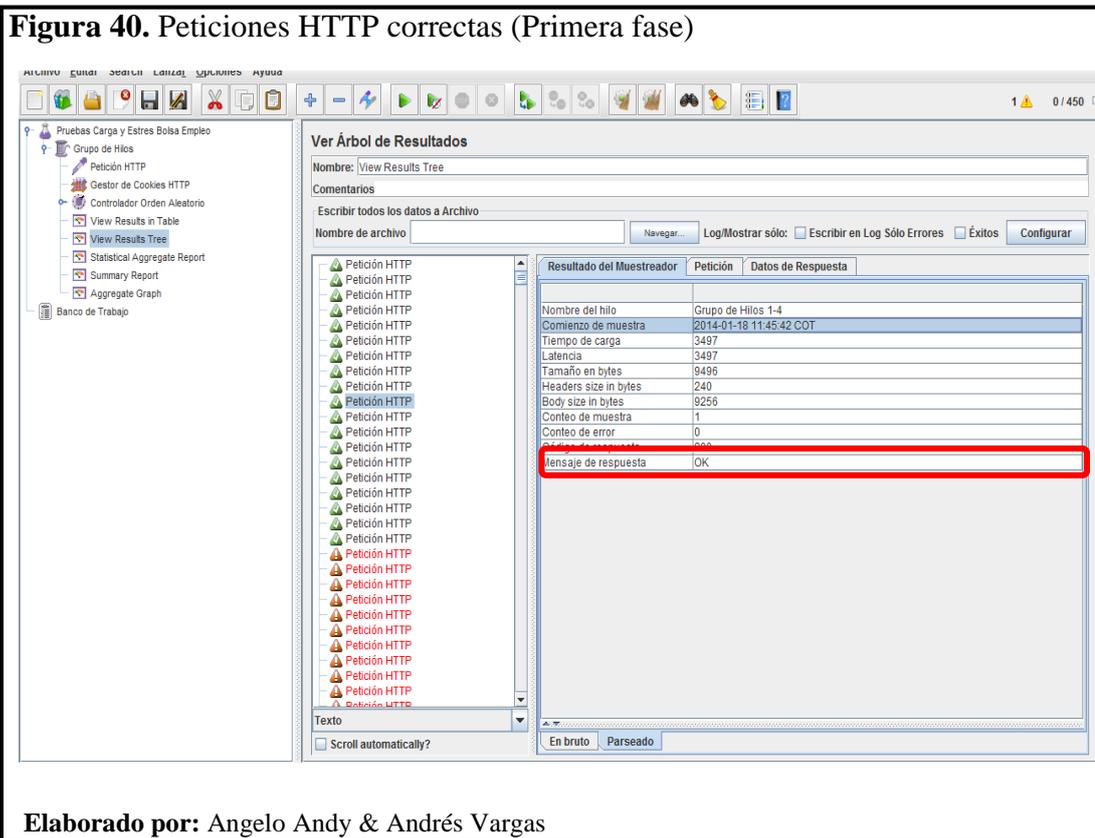
**Figura 39. Peticiones realizadas (Primera fase)**

| Nombre del hilo | Etiqueta             | Tiempo de Muestra (ms) | Estado | Bytes | Latency |
|-----------------|----------------------|------------------------|--------|-------|---------|
| 1               | Grupo de Hilos 1-12  | 1193                   | OK     | 8495  | 1193    |
| 2               | Grupo de Hilos 1-8   | 2142                   | OK     | 8495  | 2142    |
| 3               | Grupo de Hilos 1-1   | 2206                   | OK     | 8495  | 2206    |
| 4               | Grupo de Hilos 1-11  | 2072                   | OK     | 8495  | 1983    |
| 5               | Grupo de Hilos 1-6   | 2330                   | OK     | 8495  | 2330    |
| 6               | Grupo de Hilos 1-3   | 2241                   | OK     | 8495  | 2245    |
| 7               | Grupo de Hilos 1-2   | 2584                   | OK     | 8495  | 2581    |
| 8               | Grupo de Hilos 1-24  | 2364                   | OK     | 8495  | 2359    |
| 9               | Grupo de Hilos 1-16  | 2488                   | OK     | 8495  | 2482    |
| 10              | Grupo de Hilos 1-14  | 3443                   | OK     | 8495  | 3443    |
| 11              | Grupo de Hilos 1-4   | 3497                   | OK     | 8495  | 3497    |
| 12              | Grupo de Hilos 1-10  | 3644                   | OK     | 8495  | 3644    |
| 13              | Grupo de Hilos 1-13  | 3638                   | OK     | 8495  | 3638    |
| 14              | Grupo de Hilos 1-5   | 3652                   | OK     | 8495  | 3651    |
| 15              | Grupo de Hilos 1-9   | 3703                   | OK     | 8495  | 3703    |
| 16              | Grupo de Hilos 1-36  | 3871                   | OK     | 8495  | 3871    |
| 17              | Grupo de Hilos 1-7   | 3768                   | OK     | 8495  | 3768    |
| 18              | Grupo de Hilos 1-20  | 3801                   | OK     | 8495  | 3801    |
| 19              | Grupo de Hilos 1-84  | 3411                   | OK     | 9515  | 3411    |
| 20              | Grupo de Hilos 1-208 | 997                    | OK     | 2131  | 0       |
| 21              | Grupo de Hilos 1-298 | 1037                   | OK     | 2131  | 0       |
| 22              | Grupo de Hilos 1-305 | 1083                   | OK     | 2131  | 0       |
| 23              | Grupo de Hilos 1-312 | 1030                   | OK     | 2131  | 0       |
| 24              | Grupo de Hilos 1-304 | 1064                   | OK     | 2131  | 0       |
| 25              | Grupo de Hilos 1-302 | 1056                   | OK     | 2131  | 0       |
| 26              | Grupo de Hilos 1-307 | 1089                   | OK     | 2131  | 0       |
| 27              | Grupo de Hilos 1-1   | HTTP Request           | 2390   | 3107  | 103     |
| 28              | Grupo de Hilos 1-109 | 1087                   | OK     | 2131  | 0       |
| 29              | Grupo de Hilos 1-110 | 1077                   | OK     | 2131  | 0       |
| 30              | Grupo de Hilos 1-141 | 1022                   | OK     | 2131  | 0       |
| 31              | Grupo de Hilos 1-143 | 1008                   | OK     | 2131  | 0       |
| 32              | Grupo de Hilos 1-80  | 4792                   | OK     | 9515  | 4792    |
| 33              | Grupo de Hilos 1-58  | 5015                   | OK     | 9515  | 5014    |
| 34              | Grupo de Hilos 1-410 | 1102                   | OK     | 2131  | 0       |
| 35              | Grupo de Hilos 1-349 | 1060                   | OK     | 2131  | 0       |
| 36              | Grupo de Hilos 1-412 | 1087                   | OK     | 2131  | 0       |
| 37              | Grupo de Hilos 1-351 | 1047                   | OK     | 2131  | 0       |
| 38              | Grupo de Hilos 1-357 | 1032                   | OK     | 2131  | 0       |
| 39              | Grupo de Hilos 1-414 | 1074                   | OK     | 2131  | 0       |
| 40              | Grupo de Hilos 1-347 | 1091                   | OK     | 2131  | 0       |
| 41              | Grupo de Hilos 1-353 | 1050                   | OK     | 2131  | 0       |
| 42              | Grupo de Hilos 1-355 | 1048                   | OK     | 2131  | 0       |
| 43              | Grupo de Hilos 1-345 | 1102                   | OK     | 2131  | 0       |
| 44              | Grupo de Hilos 1-359 | 1015                   | OK     | 2131  | 0       |
| 45              | Grupo de Hilos 1-208 | HTTP Request           | 1088   | 2131  | 0       |
| 46              | Grupo de Hilos 1-385 | 1054                   | OK     | 2131  | 0       |
| 47              | Grupo de Hilos 1-369 | 1029                   | OK     | 2131  | 0       |
| 48              | Grupo de Hilos 1-373 | 1083                   | OK     | 2131  | 0       |
| 49              | Grupo de Hilos 1-361 | 1002                   | OK     | 2131  | 0       |
| 50              | Grupo de Hilos 1-298 | HTTP Request           | 1011   | 2131  | 0       |
| 51              | Grupo de Hilos 1-363 | 1079                   | OK     | 2131  | 0       |
| 52              | Grupo de Hilos 1-384 | 1010                   | OK     | 2131  | 0       |

Elaborado por: Angelo Andy & Andrés Vargas

## Peticiones HTTP correctas

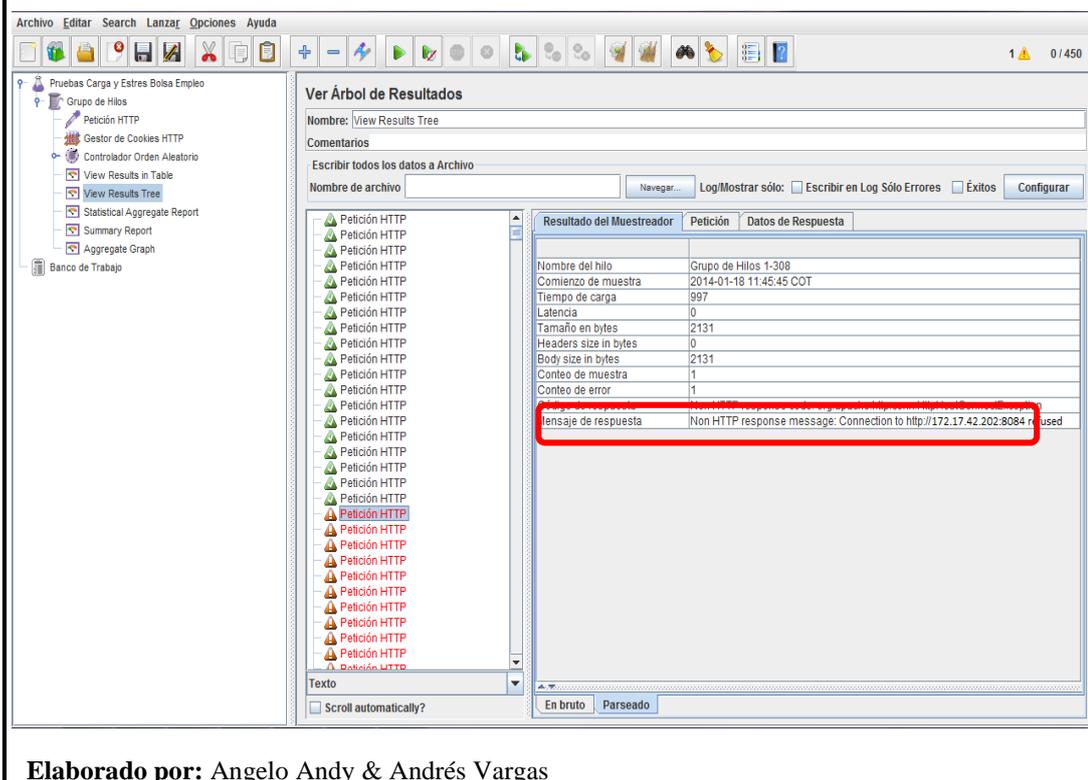
En la figura 40 se puede observar el resultado de una petición HTTP que fue atendida correctamente y obteniendo un mensaje de respuesta de “OK”, esto quiere decir que la petición fue resuelta por el servidor sin ningún problema y devolvió la información que el usuario requería en ese instante.



## Peticiones HTTP incorrectas

En la figura 41 se muestra el resultado de una petición fallida capturada en el instante de la caída del servidor, obteniendo un mensaje de respuesta “Refused”. Esto quiere decir que la información requerida por el usuario no puede ser concluida correctamente por el servidor, teniendo así problemas en la visualización de la información.

**Figura 41.** Peticiones HTTP incorrectas (Primera fase)



**Elaborado por:** Angelo Andy & Andrés Vargas

La conclusión para esta primera fase de pruebas es que la estación que se utilizó como servidor poseía características que no eran las adecuadas para atender la cantidad de usuarios y peticiones con las cuales se decidió realizar las pruebas.

#### ***4.2.2.2. Segunda fase de pruebas.***

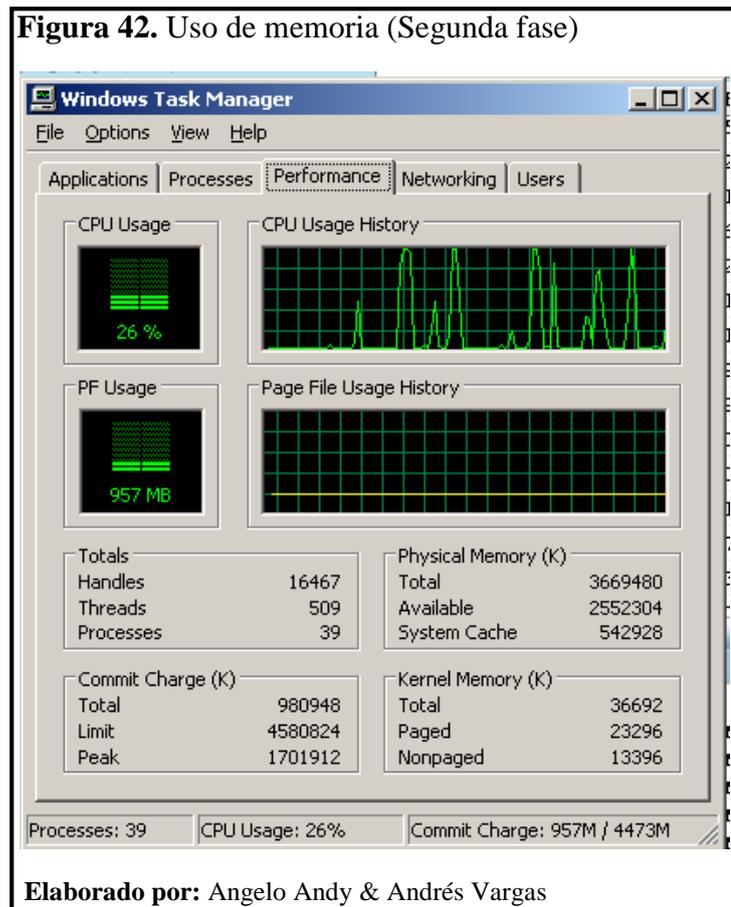
En la segunda etapa de pruebas se tomó muy en cuenta los resultados obtenidos en la primera fase, es por esta razón que se decidió utilizar un equipo con las siguientes características:

- **Sistema operativo:** Windows Server 2003
- **Memoria RAM:** 6GB
- **Procesador:** Intel Core I7 3.4Ghz

La segunda fase se la realizó con un grupo de 15 estudiantes, los resultados obtenidos mediante el software de monitoreo (JMeter), transcurrido 60 minutos desde el inicio de las pruebas, son los que a continuación se detallan:

## Uso de memoria

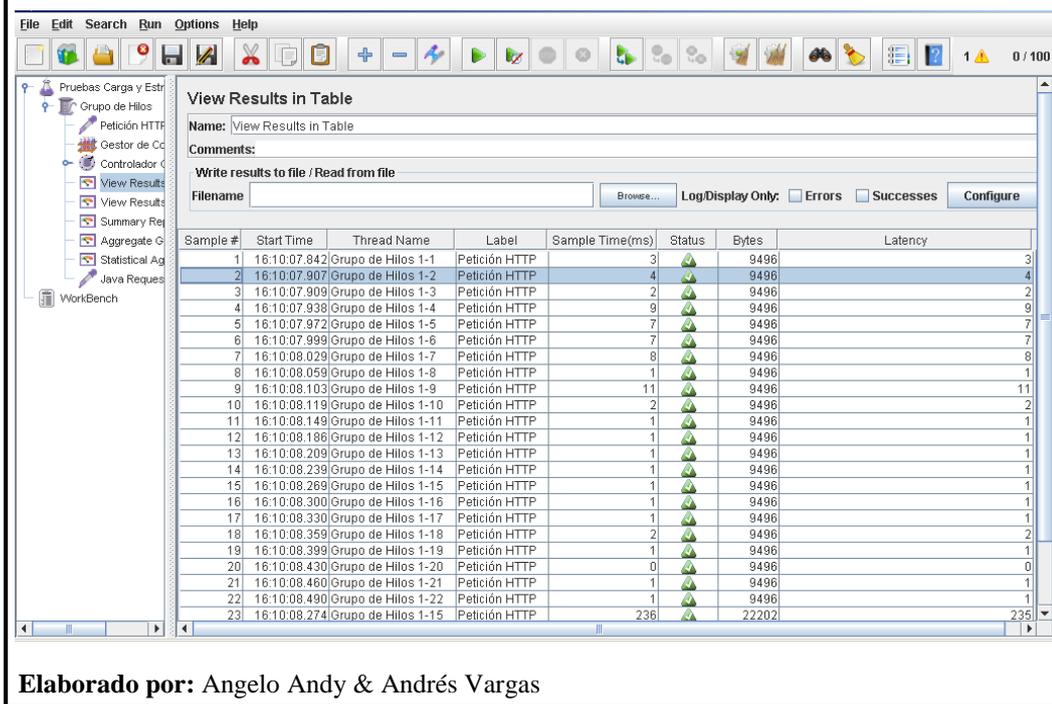
Como se puede observar en la figura 42, y en comparación con la primera fase de pruebas el uso de memoria se mantuvo en un nivel constante y considerable, ya que los procesos estaban consumiendo menos de la mitad de la capacidad máxima de la memoria física, esto permitió que las peticiones de los usuarios sean atendidas de manera normal y efectiva.



## Peticiones realizadas

En la figura 43, se tiene una lista de peticiones las cuales fueron atendidas sin ningún problema. Al contrario de la primera fase las peticiones se atendieron satisfactoriamente en un 90%.

**Figura 43. Peticiones realizadas (Segunda fase)**

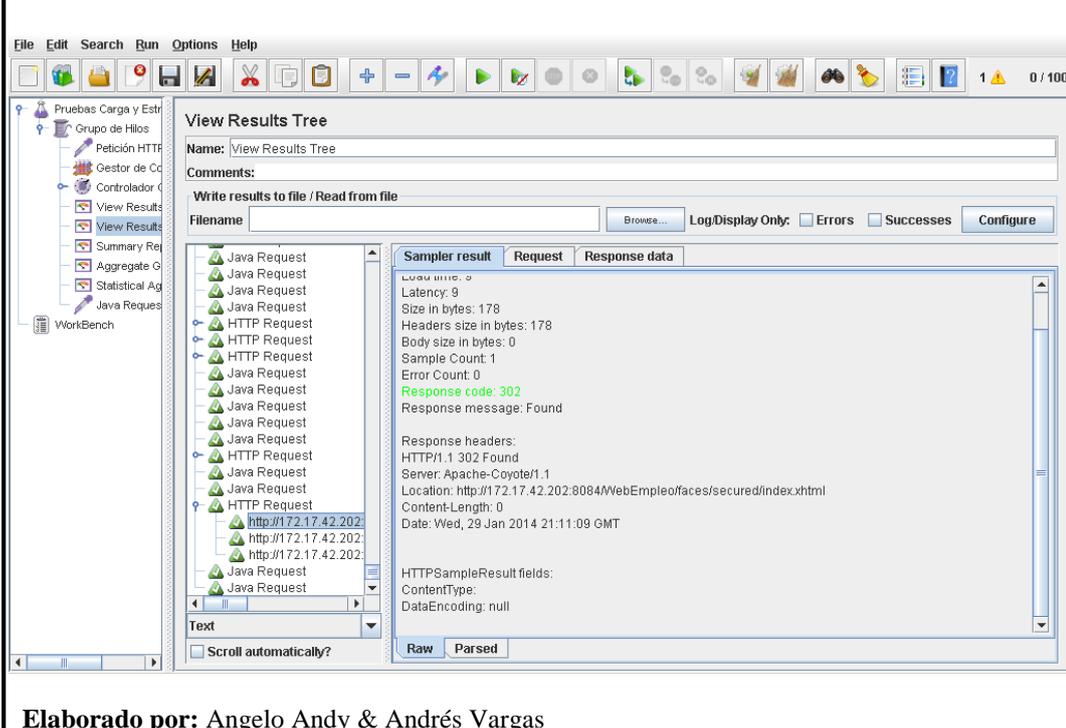


Elaborado por: Angelo Andy & Andrés Vargas

**Peticiones realizadas correctamente**

En la figura 44, se puede observar el resultado de la petición, se atendió satisfactoriamente, dando como resultado el correcto despliegue de la información requerida por el usuario en ese instante.

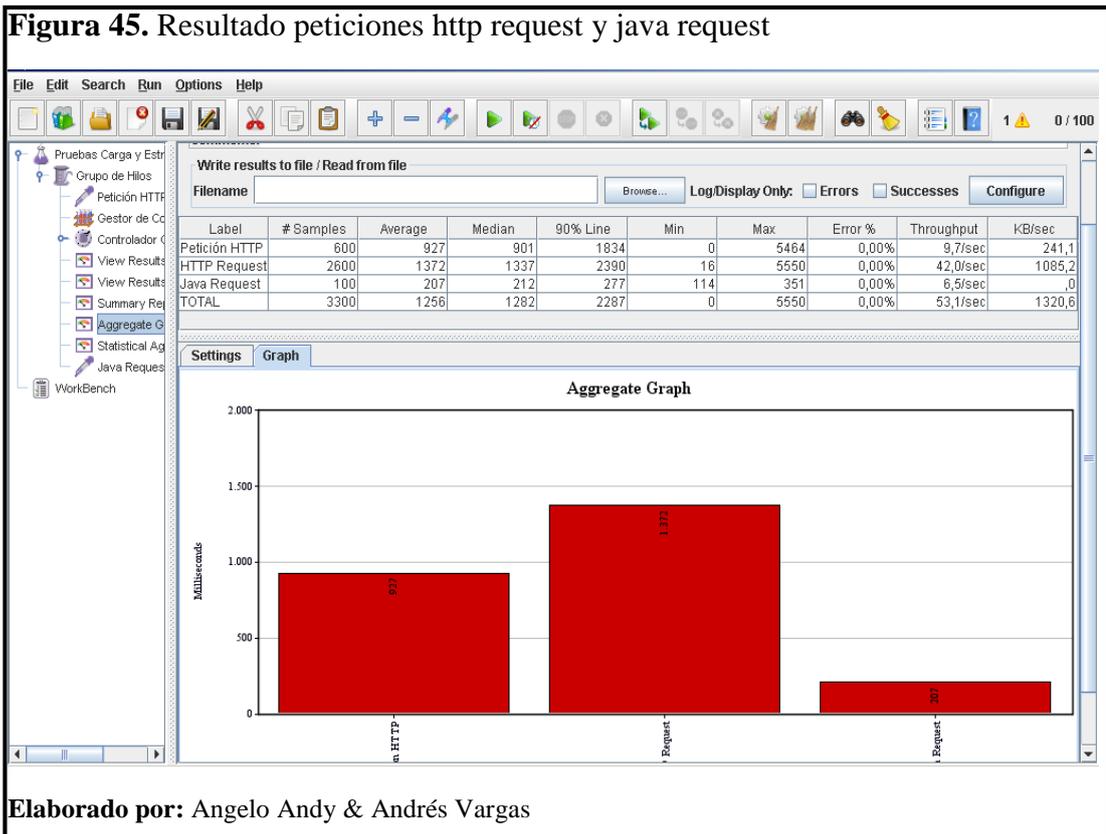
**Figura 44. Peticiones realizadas correctamente (Segunda fase)**



Elaborado por: Angelo Andy & Andrés Vargas

## Resultado de peticiones http request y java request

Como se puede observar en la figura 45, se obtiene una tabla comparativa de datos la cual contiene los tiempos máximos y mínimos de respuesta a las peticiones, el porcentaje de error, entre otros. Todos estos datos dan como resultado una gráfica de barras en la cual se puede observar la cantidad de peticiones realizadas clasificadas por el tipo de petición en este caso se tiene peticiones de tipo: http Request y Java Request.



De esta segunda fase de pruebas se puede concluir que la memoria RAM es un factor importante para tomar en cuenta al momento de ejecutar una aplicación la cual va a contar con una gran cantidad de usuarios y peticiones, finalmente la aplicación desarrollada pasó satisfactoriamente la segunda fase de pruebas.

## CONCLUSIONES

- Al finalizar el sistema se cumplió con los objetivos planteados, automatizando y agilizando los procesos administrativos que anteriormente se los realizaba de forma manual por el personal de Bienestar Estudiantil.
- El uso de la metodología OMT permitió realizar el análisis y diseño del sistema de forma ordenada y coherente, siguiendo cada una sus fases se pudo observar un flujo claro y preciso de cómo diseñar las clases y métodos para la programación de todas las funcionalidades con las que cuenta el sistema de bolsa de empleo.
- El uso de nuevas tecnologías web tales como: jpa, primefaces, JMeter entre otras, permitió desarrollar un sistema atractivo, funcional e intuitivo para el usuario final, y que este no se convierta en una herramienta obsoleta al poco tiempo de uso.
- El diseño de la aplicación en ambiente web, sigue la línea de tendencia de tecnologías nuevas, ya que estas están enfocadas al servicio en línea y permiten estar disponibles las 24 horas del día y los 365 días al año.
- El sistema se desarrolló en su mayoría con herramientas de software libre como es el caso del IDE Netbeans con sus respectivos componentes como: primefaces o jpa, de igual manera el servidor de aplicaciones web Apache Tomcat, y la herramienta JMeter para realizar las pruebas de carga y estrés, con esto se demuestra que no se necesita de herramientas privativas para la construcción de un sistema complejo y funcional.
- Con el desarrollo del sistema se ganó mucha experiencia y conocimiento en temas desconocidos tales como: JPA y Oracle, ya que la continua investigación permitió comprender e implementar las metodologías necesarias para la culminación del proyecto.

## RECOMENDACIONES

- Es aconsejable la automatización de los procesos de selección de personal para una oferta de empleo mediante un sistema informático, ya que minimiza errores por parte de la persona encargada de este proceso en Bienestar Estudiantil, y dejando por parte de los estudiantes y empresas la carga y selección de datos.
- Para el manejo adecuado del sistema y explotar todas sus opciones es recomendable que el Administrador revise el manual que se encuentra adjunto en el presente trabajo, en este se detallan las interfaces gráficas con su respectiva funcionalidad.
- Para la implementación del sistema en la universidad es necesario que se publiquen los manuales de usuario, tanto para la parte de estudiantes como para las empresas, y de esta manera evitar que los datos que se van a cargar sean incorrectos o falsos.
- Para evitar la carga innecesaria de datos se recomienda poblar la base de datos del sistema de Bolsa de Empleo, exclusivamente con los datos de estudiantes de niveles superiores, ya que ellos son los más interesados por obtener experiencia laboral.
- La información que se cargue en el sistema solo deben ser los datos personales del alumno ya que al ingresar éste se encargara de llenar todos los demás datos requeridos como la carrera que sigue en la universidad (*Referirse al anexo 1, Manual Técnico*).
- Se recomienda utilizar la metodología OMT, ya que permite construir un modelo completo del sistema a partir de los requerimientos del usuario, facilitando así la comprensión de cada una de las tareas.
- Se recomienda el uso del sistema para agilizar los procesos de la selección de postulantes en la universidad y tener un seguimiento de los estudiantes que se encuentren trabajando o en busca de nuevas oportunidades laborales.

- Realizar las pruebas de caja negra de todas las funcionalidades del sistema, para identificar los posibles errores ya sea de programación o de diseño y que no fueron identificados en la fase de desarrollo.

## LISTA DE REFERENCIAS

- apachefoundation. (2013). Apache Tomcat. Recuperado el 30 de abril del 2013 de: <http://apachefoundation.wikispaces.com/Apache+Tomcat>
- Burleson D. (2007). Oracle Tuning (Second Edition). United States of America. Elsevier
- CV-Resume. (2011). Definición. Recuperado el 25 de marzo del 2013 de: <http://www.cv-resume.org/curriculumvitae/elaborarcurriculumvitae/definicion.php>
- deGerencia. (2001). Bolsas de empleo. Recuperado el 15 de marzo del 2013 de: <http://degerencia.com/area.php?areaid=10028>
- Deitel & Associates. (2002). Advanced Java 2 Platform How to Program. New Jersey. Prentice Hall
- Geary D. & Horstmann C. (2009). Core Java Server Faces. San Francisco. Prentice Hall
- Jacobi J. & Fallows J. (2006). Pro JSF and Ajax Building Rich Internet Component. United States of America. Apress
- Keith M. & Schincariol M. (2009). Pro JPA 2 Mastering the Java Persistence API. United States of America. Apress
- Modelocurriculum. (2008). El currículum cronológico. Recuperado el 15 de abril del 2013 de: <http://www.modelocurriculum.net/tipos-el-curriculum-cronologico.html>
- Modelocurriculum. (2008). El currículum funcional o temático. Recuperado el 15 de abril del 2013 de: <http://www.modelocurriculum.net/tipos-el-curriculum-funcional-o-tematico.html>
- Modelocurriculum. (2008). El currículum combinado o mixto. Recuperado el 15 de abril del 2013 de: <http://www.modelocurriculum.net/tipos-el-curriculum-combinado-o-mixto.html>
- Netbeans. (2013). Bienvenido a Netbeans. Recuperado el 20 de abril del 2013 de: [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)

- nDeveloper. (2009). JSF (Java Server Faces). Recuperado el 25 de abril del 2013 de: <http://www.ndeveloper.com/ndeveloper/pages/system/public/templates/template1.jsf?detId=3650>
- Perrone P. (2003). Developers Handbook. Indianapolis. Sams Publishing
- Pisocompartidogranada. (2012). Usar la red para buscar trabajo. Recuperado el 22 de marzo del 2013 de: [http://pisocompartidogranada.blogspot.com/2012\\_09\\_01\\_archive.html](http://pisocompartidogranada.blogspot.com/2012_09_01_archive.html)
- Rodríguez M. (2008). La metodología OMT, caso practico
- SlideShare. (2009). Investigación de campo. Recuperado el 20 de abril del 2013 de: <http://www.slideshare.net/combertidor/investigacion-de-campo>
- Vokotic A. & Goodwill J. (2011). Apache Tomcat 7. New York. Apress
- WikiSpaces. (2011). Diagrama de clases. Recuperado el 25 de abril del 2013 de: <http://wikiuml.wikispaces.com/Diagrama+de+Clases>
- WikiSpaces. (2012). Diagramas UML. Recuperado el 16 de junio del 2013 de: <http://iegamarmediatecnica.wikispaces.com/file/view/diagramas%20uml.pdf/429504250/diagramas%20uml.pdf>