



**Universidad del Azuay**

**Facultad de Administración**

**Escuela de Ingeniería de Sistemas**

*Sistema web para el manejo de la bolsa de trabajo de la Universidad  
del Azuay*

Trabajo de graduación previo a la obtención del título de  
Ingeniero de Sistemas.

**Autores:** Mónica Durán Carrasco.

Diana Torres Guzmán.

**Director:** Ing. Pablo Esquivel.

**Cuenca, Ecuador**

**2007**

La presente va dedicada a mis padres y hermanas y a quienes de una u otra forma me han apoyado a lo largo de mi formación académica.

Diana Torres Guzmán.

A Dios por permitirme llegar hasta aquí, a mi madre y mi hermano que fueron testigos de mis largas jornadas de estudio, a mi padre y mi hermana que acortaron las distancias teniéndome presente en sus pensamientos, a mis amigas por su apoyo incondicional.

Mónica Durán Carrasco.

## **Agradecimientos**

Este trabajo se realizó gracias al apoyo y confianza que nos brindó la Ing. Katherine Ortiz y el Ing. Pablo Esquivel, quienes nos facilitaron todo el material necesario para el desarrollo de este proyecto, así como también nos ayudaron con sus conocimientos los mismos que nos permitieron continuar con el desarrollo normal de nuestro trabajo.

Todas las ideas vertidas en esta monografía son exclusiva responsabilidad de sus autoras.

Mónica Durán Carrasco.

Diana Torres Guzmán.

## Índice de Contenidos

Dedicatoria .....	II
Agradecimientos .....	IV
Índice de Contenidos .....	VI
Índice de Ilustraciones y Cuadros .....	IX
Resumen .....	X
Abstract .....	XII
INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	2
ASPECTOS TEÓRICOS .....	2
1.1 HTML .....	2
1.2 Páginas estáticas vs. Páginas dinámicas .....	3
1.2.1 Páginas estáticas .....	3
1.2.2 Páginas dinámicas .....	3
1.3 Lenguajes de lado servidor y lado cliente .....	3
1.3.1 Lenguajes de lado servidor .....	4
1.3.1.1 PHP .....	4
1.3.2 Lenguajes de lado cliente .....	5
1.3.2.1 JavaScript .....	5
1.4 Dreamweaver .....	5
CAPÍTULO 2 .....	7
ANÁLISIS DEL SISTEMA .....	7
2.1 Ámbito del Sistema .....	7
2.2 Análisis de Requisitos .....	7
2.2.1 Requisitos Específicos .....	7
2.2.2 Requisitos Técnicos .....	8
2.2.3 Requisitos Funcionales .....	8
2.2.3.1 Interfaces de Software .....	8
2.2.3.2 Interfaces de Usuario .....	8
2.3 Diagnóstico del sistema actual .....	9
2.4 Modelo Entidad Relación de la aplicación propuesta .....	9
2.4.1 Base de Datos “bolsatrabajo” .....	9
2.4.2 Base de Datos Oracle de la Universidad del Azuay .....	11
2.5 Flujograma .....	12
2.5.1 Estudiante .....	12
2.5.2 Usuarios Validadores .....	13
2.5.3 Administrador .....	14
2.6 Diccionario de datos .....	15
2.7 Mapa del Sitio .....	31
CAPÍTULO 3 .....	32
IMPLEMENTACIÓN Y PRUEBAS .....	32
3.1 Creación de Tablas .....	32
3.2 Creación del Sitio Web .....	36
3.2.1 Acceso de usuarios finales .....	37
3.2.2 Acceso de usuarios validadores .....	39
3.2.3 Acceso del usuario administrador .....	40
3.3 Seguridades .....	42
3.3.1 Seguridades para el ingreso a la hoja de vida .....	42
3.3.2 Seguridades para el acceso al sitio web .....	42

3.4 Pruebas y Corrección de errores. ....	43
CAPÍTULO 4 .....	46
MANUAL DE PROGRAMADOR .....	46
4.1 Introducción. ....	46
4.1.1 Requerimientos de Software. ....	46
4.1.2 Estructura del Sistema. ....	46
4.2 Desarrollo .....	47
4.2.1 Función de los archivos. ....	47
4.2.2 Archivo “activar.php” .....	47
4.2.3 Archivo “activar_todos.php” .....	48
4.2.4 Archivo “administrador.php” .....	49
4.2.5 Archivo “bolsa2.php” .....	51
4.2.6 Archivo “buscar.php” .....	52
4.2.7 Archivo “cabecera.php” .....	53
4.2.8 Archivo “conexion_mysql.php” .....	55
4.2.9 Archivo “consulta_pendientes.php” .....	56
4.2.10 Archivo “consulta_sugerencias.php” .....	57
4.2.11 Archivo “datos_usuario.php” .....	60
4.2.12 Archivo “desactivar.php” .....	60
4.2.13 Archivo “desactivar_todos.php” .....	61
4.2.14 Archivo “eliminacion_usuario.php” .....	64
4.2.15 Archivo “fin.htm” .....	65
4.2.16 Archivo “funciones.php” .....	67
4.2.17 Archivo “guardar_usuario.php” .....	72
4.2.18 Archivo “home.php” .....	73
4.2.19 Archivo “ingreso.php” .....	74
4.2.20 Archivo “ingreso_usuario.php” .....	75
4.2.21 Archivo “ingresoalsistema.php” .....	76
4.2.22 Archivo “inicio.php” .....	77
4.2.23 Archivo “listado_final.php” .....	77
4.2.24 Archivo “listaerrores.htm” .....	78
4.2.25 Archivo “matrices.php” .....	79
4.2.26 Archivo “mensaje.htm” .....	80
4.2.27 Archivo “modificacion_usuario.php” .....	80
4.2.28 Archivo “mostar_hojavida.php” .....	82
4.2.29 Archivo “oracle_conexion.php” .....	84
4.2.30 Archivo “oracle_consulta.php” .....	84
4.2.31 Archivo “oracle_cierra.php” .....	85
4.2.32 Archivo “quitar_validacion.php” .....	86
4.2.33 Archivo “redireccionusuario.php” .....	89
4.2.34 Archivo “scripts.js” .....	89
4.2.35 Archivo “usuarios_pendientes.php” .....	91
4.2.36 Archivo “val.js” .....	92
4.2.37 Archivo “val_usuarios.js” .....	94
4.2.38 Archivo “verblob.php” .....	95
4.2.39 Archivo “verificar_estudiantes.php” .....	96
CAPÍTULO 5 .....	98
MANUAL DEL USUARIO .....	98
5.1 Introducción. ....	98
5.2 Ingreso al Sistema. ....	98

5.3 Manejo del sistema por un usuario validador.....	99
5.4 Manejo del sistema por el usuario administrador. ....	103
5.4.1 Mantenimiento de usuarios.....	104
5.4.2 Mantenimiento de hojas de vida. ....	110
5.4.3 Consultas.....	122
CAPÍTULO 6 .....	124
CONCLUSIONES Y RECOMENDACIONES .....	124
6.1 Conclusiones.....	124
6.2 Recomendaciones. ....	124
BIBLIOGRAFÍA .....	125



## Índice de Ilustraciones y Cuadros

Gráfico 2.4.1 Modelo Entidad Relación de la base de datos “bolsatrabajo” .....	10
Gráfico 2.4.2 Modelo E-R de la base de datos Oracle de la Universidad del Azuay .	11
Gráfico 2.5 Flujograma del Sitio Web.....	12
Gráfico 2.7 Mapa del Sitio Web.....	31
Gráfico 3.2.1 Interfaz de usuarios finales .....	37
Gráfico 3.2.2 Interfaz de usuarios validadores.....	39
Gráfico 3.2.3 Interfaz de usuario administrador.....	40
Gráfico 3.3.2 Ejemplo de uso de variables de sesión. ....	43
Gráfico 5.2 Ingreso al sistema. ....	98
Gráfico 5.3 Validación por un usuario validador. ....	99
Gráfico 5.3.1 Mensaje de error en validación por usuario validador. ....	100
Gráfico 5.3.2 Inicio formulario validación de hojas de vida.....	100
Gráfico 5.3.3 Final formulario validación de hojas de vida.....	101
Gráfico 5.3.4 Pantalla de errores. ....	102
Gráfico 5.3.5 Mensaje de confirmación de datos guardados. ....	102
Gráfico 5.4 Página principal para del usuario administrador.....	103
Gráfico 5.4.1 Mantenimiento de usuarios. ....	104
Gráfico 5.4.1.1 Ingreso de usuarios. ....	104
Gráfico 5.4.1.1.2 Pantalla de errores .....	105
Gráfico 5.4.1.1.3 Mensajes de error en ingreso de usuarios. ....	106
Gráfico 5.4.1.2 Modificación de usuarios.....	106
Gráfico 5.4.1.2.1 Presentación de la información del usuario a modificar.....	107
Gráfico 5.4.1.2.2 Presentación de mensaje en modificación de usuarios. ....	108
Gráfico 5.4.1.3 Eliminación de usuarios.....	108
Gráfico 5.4.1.3.1 Presentación de información del usuario a eliminar.....	109
Gráfico 5.4.1.3.2 Presentación de mensajes en caso de eliminación.....	109
Gráfico 5.4.1.4 Presentación de información en consulta de usuarios. ....	110
Gráfico 5.4.2 Mantenimiento de hojas de vida .....	111
Gráfico 5.4.2.1 Activar por estudiante.....	111
Gráfico 5.4.2.1.2 Presentación de mensaje en caso de activación de hojas de vida.	112
Gráfico 5.4.2.1.3 Inicio formulario para la activación de hojas de vida.....	113
Gráfico 5.4.2.1.4 Final del formulario para la activación de hojas de vida. ....	114
Gráfico 5.4.2.1.5 Presentación de listado de errores. ....	114
Gráfico 5.4.2.1.6 Presentación de mensaje para activación de hojas de vida .....	115
Gráfico 5.4.2.2 Listado de hojas de vida para activar.....	115
Gráfico 5.4.2.2.1 Presentación de mensaje en caso de activación.....	116
Gráfico 5.4.2.3 Desactivación de hojas de vida. ....	117
Gráfico 5.4.2.3.1 Presentación de mensaje en caso de desactivación. ....	118
Gráfico 5.4.2.4 Validación de hojas de vida por el administrador. ....	118
Gráfico 5.4.2.4.1 Inicio de formulario para validación de hojas de vida por el administrador.....	119
Gráfico 5.4.2.4.2 Final de formulario para validación de hojas de vida por el administrador.....	120
Gráfico 5.4.2.5 Quitar validación de hojas de vida. ....	121
Gráfico 5.4.3 Consultas.....	122
Gráfico 5.4.3.1 Presentación de la consulta de hojas de vida. ....	122
Gráfico 5.4.3.2 Presentación de la consulta de sugerencias.....	123

## **Resumen**

La realización de la web de la Bolsa de Trabajo de la Universidad del Azuay permitirá tanto a estudiantes como egresados y/o graduados exclusivamente ingresar o modificar su hoja de vida a través de la página web de la institución; luego será necesario que en un lapso máximo de 72 horas, entregue la documentación que respalde dicha información.

Un grupo de validadores, comprobarán la información ingresada al sistema, contra la documentación existente, y si es necesario podrán realizar cambios en algunos de los datos de la hoja de vida.

Una vez validada la información de la hoja de vida, la persona encargada de la bolsa de trabajo o administrador del sistema, procederá a cambiar el estado de la hoja de vida de inactivo a activo. De esta manera únicamente las hojas de vida que tengan estado activo podrán ser mostrados en las consultas a través de la web.

Los ofertantes de trabajo e interesados, podrán acceder a un enlace para realizar la búsqueda del personal. Dichas consultas o búsquedas se realizan de manera abierta, es decir permitiendo al usuario escoger uno o varios requisitos que se ajusten al perfil necesario.

El administrador del sistema por medio del sitio web, estará en capacidad de ingresar, modificar, eliminar y consultar fácilmente los usuarios validadores; además encontrará las opciones para validar, quitar validación, activar y desactivar hojas de vida, así como la obtención de reportes de interés.





## INTRODUCCIÓN

¿Qué es Bolsa de trabajo?

“Lugar físico o virtual donde se ordenan ofertas para que tanto oferentes como demandantes de empleo puedan encontrar rápida y organizada la información que requieren. En el caso de los oferentes candidatos de todos los niveles y profesiones; mientras que el de los demandantes las ofertas más pertinentes.”<sup>1</sup>

La Universidad del Azuay no solo pensando en el ámbito académico sino también en el bienestar económico de sus estudiantes y/o egresados, vio la necesidad de crear La Bolsa de Trabajo, la misma que pretende ayudar a sus alumnos a conseguir un trabajo que le permita mantener una estabilidad económica; así como también poner en práctica los conocimientos adquiridos a lo largo de su carrera académica; por otro lado brindar a las empresas interesadas un servicio que les permita conseguir personal capacitado y preparado que llene todas sus expectativas.

La Universidad del Azuay contaba con una bolsa de trabajo pero esta tuvo que ser suspendida ya que no cumplía con algunos requerimientos fundamentales tales como seguridad y validaciones, los mismos que se pretenden implementar con el presente proyecto.

Tomando como referencia este preámbulo planteamos como objetivo principal de esta monografía la re-estructuración de La Bolsa de Trabajo de la Universidad del Azuay, tomando en cuenta las deficiencias existentes anteriormente.

---

<sup>1</sup> Fuente: <http://es.wikipedia.org/wiki/Bolsa>

# CAPÍTULO 1

## ASPECTOS TEÓRICOS

### Marco Teórico

A continuación detallamos de manera teórica las herramientas que hemos utilizado para el desarrollo de nuestro proyecto, sus funciones, aplicaciones, ventajas y desventajas.

#### 1.1 HTML.

HTML<sup>1</sup> es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web, y es el formato más popular que existe para la construcción de documentos web.

HTML no es un lenguaje de programación, es un lenguaje descriptivo que permite visualizar texto e imágenes con formato, introducir enlaces, entre otros; todo esto de una manera predeterminada y en ningún caso de modo dinámico o inteligente; es decir HTML no permite realizar ningún cálculo matemático por más simple que éste sea, ni crear una página utilizando bases de datos.

Por esta razón es necesario emplear lenguajes de programación que nos brinden versatilidad, es decir que permitan responder de una manera inteligente a las solicitudes del navegador y además, permitan automatizar tareas como actualizaciones, búsquedas, compras y pedidos por la web, y otros.

---

<sup>1</sup> Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto).

## **1.2 Páginas estáticas vs. Páginas dinámicas.**

En la web podemos encontrar o construir, dos tipos de páginas:

- Páginas estáticas.
- Páginas dinámicas.

### **1.2.1 Páginas estáticas.**

Son aquellas que se presentan sin ninguna funcionalidad mas allá de los enlaces, es decir no permite una amplia interacción con el usuario. Se construyen por lo general con el lenguaje HTML, no permiten grandes efectos, ni funcionalidades.

Son páginas muy sencillas de crear, pero están dirigidas a sitios web pequeños, que no ofrecen a sus usuarios mas que textos planos, imágenes y a lo sumo contenidos multimedia como videos o sonidos.

### **1.2.2 Páginas dinámicas.**

Una página dinámica es aquella que precisa la utilización de lenguajes de programación para producir algún efecto especial o funcionalidad extra a las acciones realizadas por HTML. Estas páginas son más complejas y versátiles.

## **1.3 Lenguajes de lado servidor y lado cliente.**

Cuando se accede a una página, se establece una petición de un archivo HTML, el mismo que reside en un servidor, y que es enviado e interpretado por el navegador<sup>1</sup>.

Sin embargo, no todas las páginas a las que se accede son archivos HTML, por lo tanto el navegador es incapaz de interpretar la orden. Por esta razón cuando se desea emplear algún lenguaje para realizar un sitio web, es preciso que el propio servidor sea quien ejecute e interprete los comandos y órdenes para luego enviar el resultado al navegador en forma de archivo HTML.

---

<sup>1</sup> Aplicación que interpreta órdenes recibidas en forma de código HTML y devuelve como resultado la visualización de la página web.

Dado este preámbulo, podemos definir dos tipos de lenguaje:

- Lenguajes de lado servidor.
- Lenguajes de lado cliente.

### **1.3.1 Lenguajes de lado servidor.**

Son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor en donde la página se encuentra alojada y que se envían al cliente en un formato comprensible para él.

Un lenguaje de lado servidor es independiente del cliente por lo que la programación es mucho menos estricta cuando se trata de la utilización de diferentes navegadores o de diferentes versiones del mismo.

Por otra parte, los scripts que se ejecutarán son almacenados en el servidor y éstos se traducen a HTML, por lo que permanecen ocultos para el cliente, protegiendo así el trabajo realizado. Estos scripts tienen como finalidad el procesamiento de formularios, envío de correo electrónico, administración de base de datos, y otros.

#### **1.3.1.1 PHP.**

Es un lenguaje del lado del servidor, que se utiliza para generar páginas dinámicas y que facilita el acceso y utilización de diferentes bases de datos.

Presenta algunas características interesantes como que es libre y abierto, es decir su código fuente está disponible para cualquier usuario, y además es gratuito. Es multiplataforma aunque inicialmente fue diseñado para entornos UNIX, ahora es compatible con otras plataformas como Windows. Permite un fácil acceso a varias bases de datos. Cuenta con una gran variedad de documentación en la web como manuales, tutoriales, y ejemplos. Presenta una integración perfecta con Apache y MySQL. Es de fácil aprendizaje, seguro y está orientado a objetos.



### **1.3.2 Lenguajes de lado cliente.**

Son aquellos que pueden ser directamente ejecutados por el navegador y no necesitan un tratamiento previo. Para lograr esto, el navegador del visitante debe tener las herramientas necesarias (o plug-ins) para ejecutar el script.

Un lenguaje de lado cliente es HTML, así como Java y el JavaScript los cuales son simplemente incluidos en el código HTML.

Un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de altos costos de alojamiento web.

#### **1.3.2.1 JavaScript.**

JavaScript es un lenguaje de lado cliente muy popular, que lo soportan la mayoría de navegadores. Este lenguaje es muy útil al momento de generar efectos visuales, realizar cálculos, etc.

### **1.4 Dreamweaver.**

Es un editor WYSIWYG<sup>1</sup> y uno de los programas más utilizados en todo el mundo para el diseño y programación web. Es considerado por muchos como un programa superior a otros similares.

El programa Dreamweaver es una aplicación muy completa con un gran número de funcionalidades muy potentes, pero a la vez está diseñada para hacer más cómodo el trabajo del usuario.

El éxito de este programa se basa en su gran poder de ampliación y personalización, sus funcionalidades, integración con otras herramientas, también permite la conexión

---

<sup>1</sup> **WYSIWYG.** Acrónimo de *What You See Is What You Get* ("lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como editores HTML) que permiten a la vez escribir un documento y visualizar directamente el resultado final.

a Bases de Datos como MySQL y Microsoft Access, y además filtra y muestra el contenido utilizando tecnología de script como por ejemplo: ASP, .NET, ColdFusion, JSP y PHP sin necesidad de tener una gran experiencia previa en programación.

Dreamweaver cuenta con un administrador de sitios para agrupar los archivos según el proyecto al que pertenezcan, un cliente FTP integrado que permite subir los archivos editados inmediatamente al sitio en Internet, y funciones de auto completar y resaltar la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP.

## **CAPÍTULO 2**

### **ANÁLISIS DEL SISTEMA**

#### **2.1 Ámbito del Sistema**

La realización del sitio web de la Bolsa de Trabajo de la Universidad del Azuay tiene como fin brindar a los alumnos la posibilidad de llenar su hoja de vida y además, permitir a las empresas interesadas buscar entre los aspirantes disponibles el personal necesario según sus requerimientos.

Se podrá realizar lo mencionado anteriormente accediendo al vínculo “Bolsa de trabajo” que se encuentra en la página principal de la Universidad del Azuay en los enlaces “Ingresar hoja de vida” y “Consultar hoja de vida”.

Por otro lado el sistema permitirá a usuarios validar la información introducida por los estudiantes, y al administrador encargado de la bolsa de trabajo validar, activar hojas de vida validadas, realizar varios tipos de consultas y tener un control de usuarios.

De esta manera se pretende brindar ayuda tanto a los estudiantes como a las empresas que hagan uso de esta página, cubriendo en ambos casos las necesidades planteadas.

#### **2.2 Análisis de Requisitos**

##### **2.2.1 Requisitos Específicos.**

Para determinar los requerimientos del proyecto se realizaron varias reuniones en la Universidad del Azuay entre las personas interesadas María Cecilia Alvarado y la Ing. Katherine Ortiz en donde se llegaron a establecer en gran parte requisitos tales como la estructura del sitio web y sus detalles, validaciones necesarias y seguridades.

Este sitio web será desarrollado de manera que los estudiantes, usuarios interesados, usuarios validadores y administrador de la bolsa de trabajo puedan utilizar el sistema de la manera más fácil posible.

### **2.2.2 Requisitos Técnicos.**

Son los requisitos de software y herramientas necesarias para el desarrollo de este proyecto.

Se ha establecido que para el desarrollo normal de nuestro trabajo los lenguajes PHP, HTML y JavaScript son suficientes. La base de datos seleccionada es MySQL, por ser una herramienta confiable y de libre distribución. Además se trabajará sobre un servidor Apache con el que actualmente cuenta la universidad.

### **2.2.3 Requisitos Funcionales.**

A continuación se describen aquellos requisitos que pueden afectar a interfaces externas tales como interfaces con otros sistemas de hardware y software así como también interfaces con el usuario.

#### **2.2.3.1 Interfaces de Software**

Para el desarrollo de este proyecto se hace uso del servidor Apache, que se asienta sobre el sistema operativo Linux y que accederá a la base de datos Oracle que utiliza la Universidad del Azuay.

#### **2.2.3.2 Interfaces de Usuario**

En este apartado se hace referencia que usuario son todas aquellas personas que van a acceder al sitio web: estudiantes, personas o empresas interesadas, administrador y validadores a quienes se ofrecerá una interfaz fácil de manejar e intuitiva permitiendo de esta manera que el usuario se familiarice en poco tiempo.

### **2.3 Diagnóstico del sistema actual.**

La bolsa de trabajo con la que contaba la Universidad del Azuay permitía el ingreso de hojas de vida no solo de estudiantes sino también de personas ajenas a la institución ya que no se requería ingresar el código de estudiante, ni su cédula de identidad.

La información ingresada en las hojas de vida no era verificada con su documentación correspondiente, y se visualizaba inmediatamente en las búsquedas a las cuales solamente podían acceder empresas que tenían convenios con la universidad, a las cuales se les proporcionaba un usuario y contraseña.

### **2.4 Modelo Entidad Relación de la aplicación propuesta.**

#### **2.4.1 Base de Datos “bolsatrabajo”**

El siguiente gráfico indica las relaciones existentes, cardinalidad, llaves primarias y campos que se relacionan entre las diferentes tablas que conforman la base de datos “bolsatrabajo”.

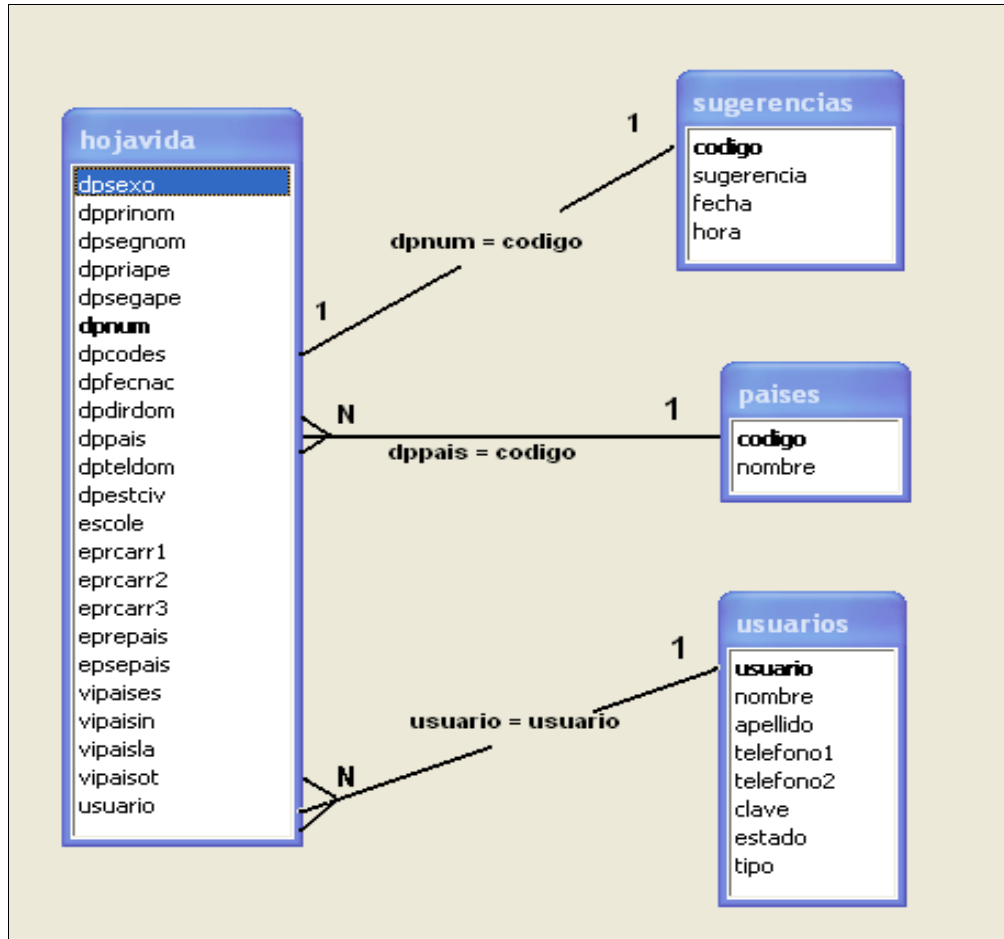


Gráfico 2.4.1 Modelo Entidad Relación de la base de datos “bolsatrabajo”

## 2.4.2 Base de Datos Oracle de la Universidad del Azuay.

El siguiente gráfico indica las relaciones existentes, cardinalidad, llaves primarias y campos que se relacionan entre las diferentes tablas que conforman la base de datos de Oracle de la Universidad del Azuay.

Debido a la cantidad existente de campos en las tablas, solo se visualizan los campos utilizados en la aplicación.

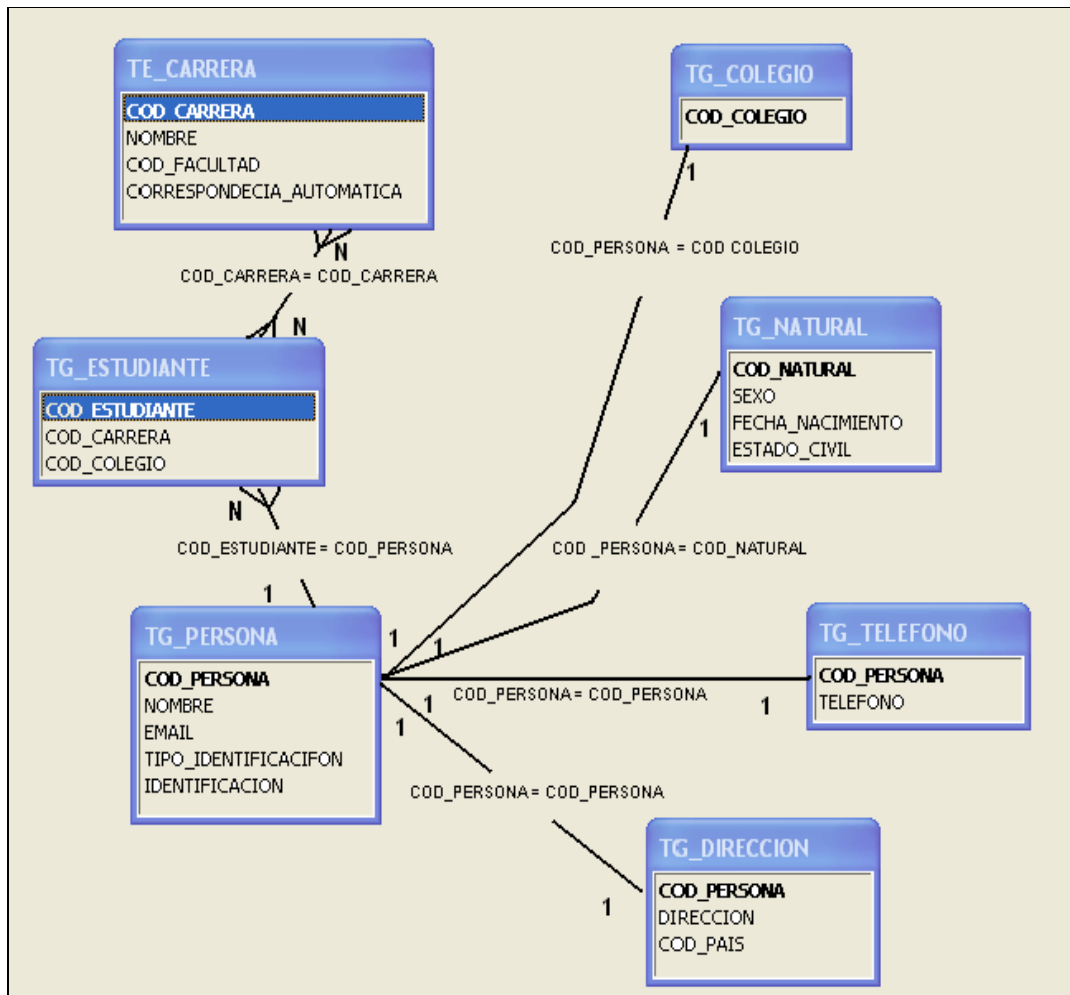


Gráfico 2.4.2 Modelo E-R de la base de datos Oracle de la Universidad del Azuay

## 2.5 Flujograma.

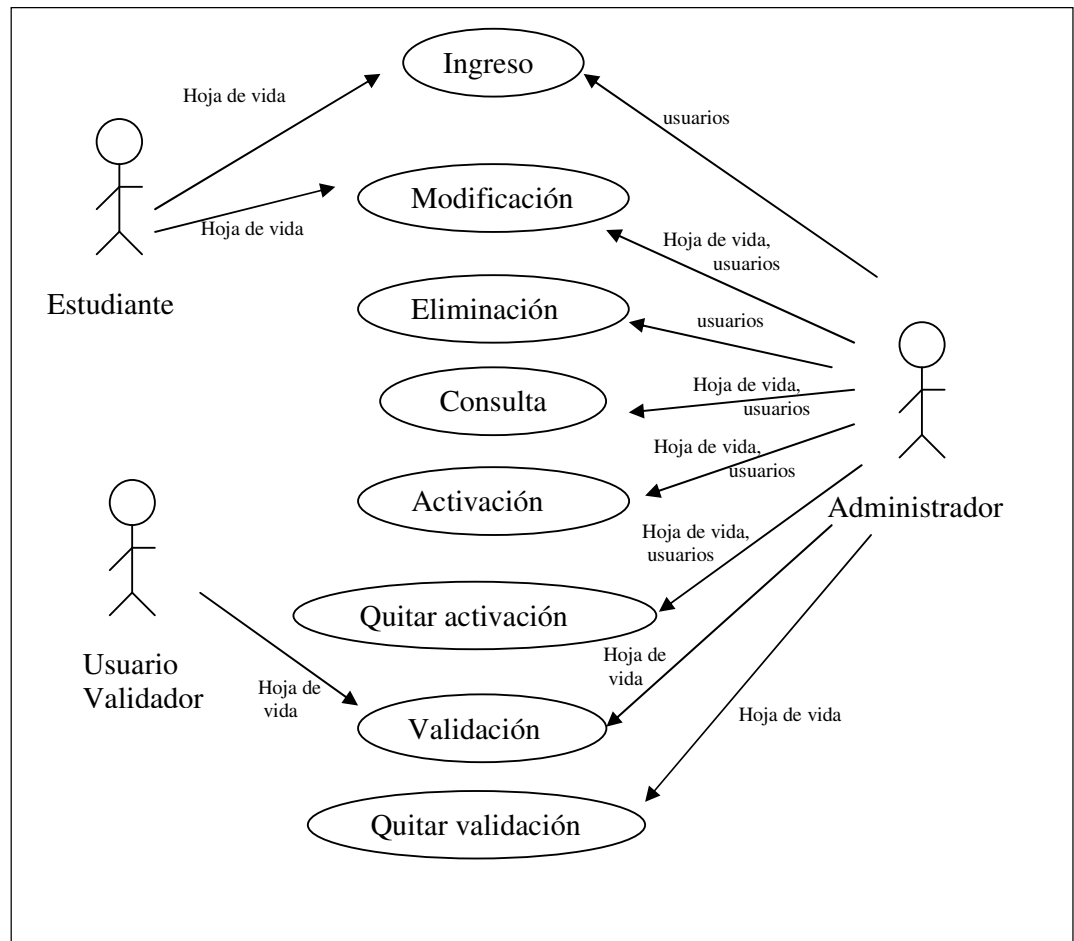


Gráfico 2.5 Flujograma del Sitio Web

### 2.5.1 Estudiante

- El sistema permite a los estudiantes acceder al vínculo “Ingresar Hoja de vida” de la página web de la Bolsa de Trabajo de la Universidad del Azuay y después colocar su código de estudiante y su número de cédula.
- El estudiante puede modificar los datos de su hoja de vida mientras ésta no sea validada por los usuarios validadores o por el administrador.
- Además, de la información introducida por el estudiante se guardará también la hora, fecha y dirección IP de manera automática.



- Al momento de guardar la hoja de vida, los campos “validacion” y “estado” se guardarán con los datos “no” y “0” respectivamente, lo cual indica que la hoja de vida no ha sido validada ni activada, dichos campos cambiarán cuando la información sea verificada con su documentación respectiva.
- Como resultado de las búsquedas realizadas en el enlace “Consultar hoja de Vida”, el sistema visualizará solo aquellas hojas de vida cuyo estado sea “1”, es decir esté activo.

### **2.5.2 Usuarios Validadores**

- El sistema permite ingresar, modificar y eliminar usuarios a los cuales se les asignará un código, contraseña y un estado.
- Solamente los usuarios con estado activo pueden ingresar al sistema.
- Se puede cambiar de estado activo a inactivo al usuario, sin que este sea eliminado, lo que permite saber quien ha realizado ciertas actividades.
- Se permite actualizar los datos del usuario tales como el nombre, apellido, dirección, teléfono, contraseña.
- El usuario puede acceder a las hojas de vida de los estudiantes para verificar que la información ingresada en la hoja de vida se respalde con la documentación entregada.
- Para ingresar al sistema, el usuario debe ingresar su código de usuario y contraseña.
- Para que un usuario pueda validar una hoja de vida, debe ingresar el número de cédula del estudiante, dato que podrá encontrar en las carpetas que le fueron entregadas para su validación.
- El usuario puede cambiar el estado de la hoja de vida del estudiante de no validada a validada.
- Se guardará el código de usuario, hora, fecha y dirección IP de manera automática una vez validada la hoja de vida; lo cual permite saber quien fue la última persona en modificarla.

### **2.5.3 Administrador**

- El administrador puede acceder al sistema luego de ingresar su usuario y contraseña.
- Se permite actualizar los datos del administrador tales como nombres, apellidos, teléfono y contraseña.
- El sistema permite únicamente al administrador realizar el mantenimiento de usuarios, es decir ingreso, modificación, eliminación y consultas.
- El administrador puede acceder a las hojas de vida de los estudiantes para verificar que la información ingresada en la hoja de vida se respalde con la documentación entregada.
- Se guardará el código del administrador, hora, fecha y dirección IP de manera automática una vez validada la hoja de vida; lo que permite saber quien fue la última persona en modificarla.
- El administrador es la única persona que pueda cambiar de estado Inactivo a Activo la hoja de vida de los estudiantes.
- El administrador puede consultar las hojas de vida validadas, pendientes por validar y pendientes por activar.
- El sistema permite al administrador consultar los usuarios activos o inactivos con su respectiva información.

## 2.6 Diccionario de datos

**Tabla: hojavida**

Atributo	Descripción	Definición	Tipo	Detalle
dpfoto	Fotografía del estudiante		blob	
archivo_nombre	Nombre del archivo que contiene la foto		varchar(20)	
archivo_tipo	Tipo de archivo de la foto		varchar(20)	
archivo_tamano	Tamaño del archivo de la foto en bytes		varchar(20)	
dpsexo	Género del estudiante		varchar(9)	masculino - femenino
dpprinom	Primer nombre del estudiante		varchar(20)	
dpsegnom	Segundo nombre del estudiante		varchar(20)	
dppriape	Primer apellido del estudiante		varchar(20)	
dpsegape	Segundo apellido del estudiante		varchar(20)	
dpnum	Número de cédula del estudiante		varchar(10)	
dpcedpas	Indica el tipo de documento de identidad		varchar(17)	cédula - pasaporte
dpiess	Número de carnet de afiliación del seguro social		varchar(13)	
dpcodes	Código del estudiante	Pk	varchar(6)	
dpfecnac	Fecha de nacimiento del estudiante (aaaa/mm/dd)		date	
dpdirdom	Dirección domiciliaria del estudiante		varchar(60)	
dppaisor	País de origen del estudiante		varchar(4)	Lista de países
dpprov	Provincia de origen del estudiante		varchar(20)	Lista de provincias
dpciu	Ciudad de origen del estudiante		varchar(20)	Lista de ciudades
dpteldom	Teléfono del domicilio del estudiante		varchar(10)	
dpotrtel	Otro teléfono de contacto del estudiante		varchar(10)	

dpemail	Dirección electrónica del estudiante		varchar(50)	
dpcel	Número de celular del estudiante		varchar(10)	
dpcontac	Manera en la que desea ser contactado el estudiante		varchar(20)	teléfono domicilio, email, celular, otro teléfono
dpestciv	Estado civil del estudiante		varchar(15)	Soltero(a), casado(a), divorciado(a), viuda(a)
dpnumhij	Número de hijos del estudiante		varchar(1)	
escole	Nombre del colegio en el que se graduó el estudiante		varchar(100)	Lista de colegios
esarea	Área de especialidad en la que se graduó el estudiante		varchar(100)	Lista de áreas de especialidad
estitbac	Título de bachiller obtenido		varchar(100)	
esanogra	Año de graduación del estudiante		varchar(4)	
esinfrel	Información relevante acerca de los estudios secundarios del estudiante como dignidad estudiantil, participación en eventos académicos, culturales o deportivos, etc.		text	
epcarr1	Nombre de la carrera que se cursa, o haya cursado en la Universidad del Azuay		varchar(100)	Lista de carreras
epresta1	Indica el estado en el que se encuentra la carrera		varchar(10)	graduado - egresado - estudiante
eprciap1	Indica el porcentaje de ciclos aprobados por el estudiante en la Universidad del Azuay		varchar(3)	
epcarr2	Nombre de la carrera que se cursa, o haya cursado en la Universidad del Azuay		varchar(100)	Lista de carreras
epresta2	Indica el estado en el que se encuentra la carrera		varchar(10)	graduado - egresado - estudiante
eprciap2	Indica el número de ciclos aprobados por el estudiante en la Universidad del Azuay		varchar(3)	

eprcarr3	Nombre de la carrera que se cursa, o haya cursado en la Universidad del Azuay		varchar(100)	Lista de carreras
eprest3	Indica el estado en el que se encuentra la carrera		varchar(10)	graduado - egresado - estudiante
eprciap3	Indica el número de ciclos aprobados por el estudiante en la Universidad del Azuay		varchar(3)	
eprouni	Nombre de la universidad donde cursa, o haya cursado estudios de pregrado		varchar(60)	
eprocar	Nombre de la carrera que se cursa, o haya cursado en otra universidad		varchar(60)	
eproest	Indica el estado en el que se encuentra la carrera que estudia o haya estudiado en otra universidad		varchar(10)	graduado - egresado - estudiante
eprociap	Porcentaje de aprobación de la carrera		varchar(3)	
eprepais	País donde haya cursado otros estudios de pregrado		varchar(4)	Lista de países
epreuni	Nombre de la universidad extranjera donde haya cursado estudios de pregrado		varchar(60)	
eprekar	Nombre de la carrera que se cursó en una universidad extranjera		varchar(60)	
epreest	Indica el estado en el que se encuentra la carrera que haya estudiado en una universidad extranjera		varchar(10)	graduado - egresado - estudiante
epreciap	Porcentaje de aprobación de la carrera que haya estudiado en una universidad extranjera		varchar(3)	
epreinfrel	Información relevante acerca de todos los estudios de pregrado que haya realizado		text	
epsare1	Área de especialidad en la que se graduó o está cursando el estudiante		varchar(100)	Lista de especialidades
epsest1	Indica el estado en el que se encuentra el postgrado que haya estudiado en la Universidad del Azuay		varchar(10)	graduado - egresado - estudiante

epsniv1	Tipo de postgrado que esté o haya realizado en la Universidad del Azuay		varchar(15)	PhD - Maestría - Especialidad - Diplomado
epsnum1	Porcentaje de aprobación del postgrado		varchar(3)	
epsare2	Área de especialidad en la que se graduó o está cursando el estudiante		varchar(100)	
epsest2	Indica el estado en el que se encuentra el postgrado que haya estudiado en la Universidad del Azuay		varchar(10)	graduado - egresado - estudiante
epsniv2	Tipo de postgrado que esté o haya realizado en la Universidad del Azuay		varchar(15)	PhD - Maestría - Especialidad - Diplomado
epsnum2	Porcentaje de aprobación del postgrado		varchar(3)	
epsare3	Área de especialidad en la que se graduó o está cursando el estudiante		varchar(100)	
epsest3	Indica el estado en el que se encuentra el postgrado que haya estudiado en la Universidad del Azuay		varchar(10)	graduado - egresado - estudiante
epsniv3	Tipo de postgrado que esté o haya realizado en la Universidad del Azuay		varchar(15)	PhD - Maestría - Especialidad - Diplomado
epsnum3	Porcentaje de aprobación del postgrado		varchar(3)	
epsouni	Nombre de la universidad donde esté o haya realizado estudios de postgrado		varchar(60)	
epsoarea	Área de especialidad en la que se graduó o está cursando el estudiante en otra universidad		varchar(60)	
epsoest	Indica el estado en el que se encuentra el postgrado que haya o esté estudiando en otra universidad		varchar(12)	graduado - egresado - estudiante
epsoniv	Tipo de postgrado que esté o haya realizado en otra universidad		varchar(15)	PhD - Maestría - Especialidad -

				Diplomado
epsonum	Porcentaje de aprobación del postgrado que haya o esté realizando en otra universidad		varchar(3)	
epsepais	Nombre del país en donde haya realizado un postgrado		varchar(4)	Lista de países
epseuni	Nombre de la universidad extranjera donde haya realizado estudios de postgrado		varchar(60)	
epsearea	Área de especialidad en la que haya cursado en otra universidad extranjera		varchar(60)	
epseniv	Indica el estado en el que se encuentra el postgrado que haya estudiado en una universidad extranjera		varchar(15)	graduado - egresado - estudiante
epseest	Tipo de postgrado que ha realizado en una universidad extranjera		varchar(12)	PhD - Maestría - Especialidad - Diplomado
epsenum	Porcentaje de aprobación del postgrado realizado en una universidad extranjera		varchar(3)	
epseinfrel	Información relevante acerca de cualquier postgrado realizado		text	
vivivido	Indica si el estudiante ha vivido en otro país		varchar(2)	si - no
vipaises	Nombre del país en el que vivió por postgrado o pregrado en el extranjero		varchar(4)	Lista de países
videmes4	Mes en el que inició su viaje por postgrado o pregrado en el extranjero		varchar(10)	Lista de meses
videano4	Año en el que inició su viaje por postgrado o pregrado en el extranjero		varchar(4)	Lista de años
vihaemes4	Mes en el que terminó su viaje por postgrado o pregrado en el extranjero		varchar(10)	Lista de meses
vihaano4	Año en el que terminó su viaje por postgrado o pregrado en el extranjero		varchar(4)	Lista de años

vipaisin	Nombre del país en el que vivió por experiencia de intercambio		varchar(4)	Lista de países
videmes1	Mes en el que inició su viaje por experiencia de intercambio		varchar(10)	Lista de meses
videano1	Año en el que inició su viaje por experiencia de intercambio		varchar(4)	Lista de años
vihames1	Mes en el que terminó su viaje por experiencia de intercambio		varchar(10)	Lista de meses
vihaano1	Año en el que terminó su viaje por experiencia de intercambio		varchar(4)	Lista de años
vipaisla	Nombre del país en el que vivió por actividades laborales		varchar(4)	Lista de países
videmes2	Mes en el que inició su viaje por actividades laborales		varchar(10)	Lista de meses
videano2	Año en el que inició su viaje por actividades laborales		varchar(4)	Lista de años
vihames2	Mes en el que terminó su viaje por actividades laborales		varchar(10)	Lista de meses
vihaano2	Año en el que terminó su viaje por actividades laborales		varchar(4)	Lista de años
vipaisot	Nombre del país al que viajó por otros motivos		varchar(4)	Lista de países
videmes3	Mes en el que inició su viaje por otros motivos		varchar(10)	Lista de meses
videano3	Año en el que inició su viaje por otros motivos		varchar(4)	Lista de años
vihames3	Mes en el que terminó su viaje por otros motivos		varchar(10)	Lista de meses
vihaano3	Año en el que terminó su viaje por otros motivos		varchar(4)	Lista de años
famesin1	Indica el mes en el que inició alguna capacitación		varchar(2)	Lista de meses
faanoin1	Indica el año en el que inició alguna capacitación		varchar(4)	Lista de años
fadurac1	Indica el número de acuerdo a la unidad de tiempo que seleccione, que duró la capacitación adicional		varchar(3)	
faunida1	Indica la unidad de tiempo		varchar(7)	horas - días - semanas - meses - años
fatipca1	Indica el tipo de capacitación recibida		varchar(25)	Lista de tipos de capacitación



fanacin1	Indica si la capacitación recibida es nacional o internacional		varchar(13)	Nacional, Internacional
faforpa1	Indica el tipo de participación que realizó en la capacitación		varchar(15)	Organizador - Expositor - Asistente
fainsor1	Nombre de la institución organizadora de la capacitación		varchar(60)	
fatemfo1	Nombre del tema tratado en la capacitación		varchar(60)	
famesin2	Indica el mes en el que inició alguna capacitación		varchar(2)	Lista de meses
faanoin2	Indica el año en el que inició alguna capacitación		varchar(4)	Lista de años
fadurac2	Indica el número de acuerdo a la unidad de tiempo que seleccione, que duró la capacitación adicional		varchar(3)	
faunida2	Indica la unidad de tiempo		varchar(7)	horas - días - semanas - meses - años
fatipca2	Indica el tipo de capacitación recibida		varchar(25)	Lista de tipos de capacitación
fanacin2	Indica si la capacitación recibida es nacional o internacional		varchar(13)	Nacional, Internacional
faforpa2	Indica el tipo de participación que realizó en la capacitación		varchar(15)	Organizador - Expositor - Asistente
fainsor2	Nombre de la institución organizadora de la capacitación		varchar(60)	
fatemfo2	Nombre del tema tratado en la capacitación		varchar(60)	
famesin3	Indica el mes en el que inició alguna capacitación		varchar(2)	Lista de meses
faanoin3	Indica el año en el que inició alguna capacitación		varchar(4)	Lista de años
fadurac3	Indica el número de acuerdo a la unidad de tiempo que seleccione, que duró la capacitación adicional		varchar(3)	
faunida3	Indica la unidad de tiempo		varchar(7)	horas - días - semanas - meses - años
fatipca3	Indica el tipo de capacitación recibida		varchar(25)	lista de tipos de capacitación

fanacin3	Indica si la capacitación recibida es nacional o internacional		varchar(13)	Nacional, Internacional
faforpa3	Indica el tipo de participación que realizó en la capacitación		varchar(15)	Organizador - Expositor - Asistente
fainsor3	Nombre de la institución organizadora de la capacitación		varchar(60)	
fatemfo3	Nombre del tema tratado en la capacitación		varchar(60)	
famesin4	Indica el mes en el que inició alguna capacitación		varchar(2)	Lista de meses
faanoia4	Indica el año en el que inició alguna capacitación		varchar(4)	Lista de años
fadurac4	Indica el número de acuerdo a la unidad de tiempo que seleccione, que duró la capacitación adicional		varchar(3)	
faunida4	Indica la unidad de tiempo		varchar(7)	horas - días - semanas - meses - años
fatipca4	Indica el tipo de capacitación recibida		varchar(25)	Lista de tipos de capacitación
fanacin4	Indica si la capacitación recibida es nacional o internacional		varchar(13)	Nacional, Internacional
faforpa4	Indica el tipo de participación que realizó en la capacitación		varchar(15)	Organizador - Expositor - Asistente
fainsor4	Nombre de la institución organizadora de la capacitación		varchar(60)	
fatemfo4	Nombre del tema tratado en la capacitación		varchar(60)	
fainfrel	Información relevante acerca de cualquier capacitación recibida		text	
ciidiom1	Nombre del idioma en el que se tenga algún conocimiento		varchar(15)	Lista de idiomas
cinivel1	Nivel de desenvolvimiento del estudiante en algún idioma		varchar(12)	técnico - cotidiano - ambos
cilectu1	Porcentaje de lectura sobre un idioma		varchar(3)	
ciescri1	Porcentaje de escritura sobre un idioma		varchar(3)	
ciexpre1	Porcentaje de expresión oral sobre un idioma		varchar(3)	

ciidiom2	Nombre del idioma en el que se tenga algún conocimiento		varchar(15)	Lista de idiomas
cinivel2	Nivel de desenvolvimiento del estudiante en algún idioma		varchar(12)	técnico - cotidiano - ambos
cilectu2	Porcentaje de lectura sobre un idioma		varchar(3)	
ciescri2	Porcentaje de escritura sobre un idioma		varchar(3)	
ciexpre2	Porcentaje de expresión oral sobre un idioma		varchar(3)	
ciidiom3	Nombre del idioma en el que se tenga algún conocimiento		varchar(15)	Lista de idiomas
cinivel3	Nivel de desenvolvimiento del estudiante en algún idioma		varchar(12)	técnico - cotidiano - ambos
cilectu3	Porcentaje de lectura sobre un idioma		varchar(3)	
ciescri3	Porcentaje de escritura sobre un idioma		varchar(3)	
ciexpre3	Porcentaje de expresión oral sobre un idioma		varchar(3)	
ciidiom4	Nombre del idioma en el que se tenga algún conocimiento		varchar(15)	Lista de idiomas
cinivel4	Nivel de desenvolvimiento del estudiante en algún idioma		varchar(12)	técnico - cotidiano - ambos
cilectu4	Porcentaje de lectura sobre un idioma		varchar(3)	
ciescri4	Porcentaje de escritura sobre un idioma		varchar(3)	
ciexpre4	Porcentaje de expresión oral sobre un idioma		varchar(3)	
word	Nivel de conocimientos acerca de la herramienta Microsoft Word		varchar(12)	Experto - Avanzado - Intermedio - Básico
excel	Nivel de conocimientos acerca de la herramienta Microsoft Excel		varchar(12)	Experto - Avanzado - Intermedio - Básico
power	Nivel de conocimientos acerca de la herramienta Microsoft Power Point		varchar(12)	Experto - Avanzado - Intermedio - Básico
cinbusin	Nivel de conocimientos acerca de búsquedas en Internet		varchar(12)	Experto - Avanzado - Intermedio - Básico
cincorel	Nivel de conocimientos acerca de correo electrónico		varchar(12)	Experto - Avanzado - Intermedio - Básico

cinher1	Indica el nombre de otras herramientas en las que tenga conocimiento del estudiante		varchar(20)	Lista de herramientas
cinher2	Indica el nombre de otras herramientas en las que tenga conocimiento del estudiante		varchar(20)	Lista de herramientas
cinher3	Indica el nombre de otras herramientas en las que tenga conocimiento del estudiante		varchar(20)	Lista de herramientas
ciherni1	Nivel de conocimiento acerca de la herramienta uno		varchar(12)	Experto - Avanzado - Intermedio - Básico
ciherni2	Nivel de conocimiento acerca de la herramienta dos		varchar(12)	Experto - Avanzado - Intermedio - Básico
ciherni3	Nivel de conocimiento acerca de la herramienta tres		varchar(12)	Experto - Avanzado - Intermedio - Básico
cininfrel	Información relevante acerca de cualquier conocimiento informático		text	
odtraequ	Indica el porcentaje de capacidad del estudiante para trabajar en equipo		varchar(3)	
odsupjef	Indica el porcentaje de capacidad del estudiante para supervisión		varchar(3)	
odexpven	Indica el porcentaje de experiencia del estudiante en ventas		varchar(3)	
odhabcom	Indica el porcentaje de habilidad del estudiante de comunicación oral y persuasión		varchar(3)	
oddescom	Indica el porcentaje de destrezas del estudiante en comunicación escrita		varchar(3)	
odtrapre	Indica el porcentaje de capacidad del estudiante para trabajar bajo presión		varchar(3)	
elemprel	Nombre de la empresa donde haya trabajado		varchar(60)	
elacteml	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades

elanoin1	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin1	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi1	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses
elanofi1	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel11	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel21	Otro teléfono de la empresa donde haya trabajado		varchar(10)	
eljefdi1	Nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargol	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca1	Detalle de las actividades a cargo del estudiante		text	
elempre2	Nombre de la empresa donde haya trabajado		varchar(60)	
elactem2	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades
elanoin2	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin2	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi2	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses
elanofi2	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel12	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel22	Otro teléfono de la empresa donde haya trabajado		varchar(10)	
eljefdi2	Nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargol2	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca2	Detalle de las actividades a cargo del estudiante		text	
elempre3	Nombre de la empresa donde haya trabajado		varchar(60)	
elactem3	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades
elanoin3	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin3	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi3	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses

elanofi3	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel13	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel23	Otro teléfono de la empresa donde haya trabajado		varchar(10)	
eljefdi3	Nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargos3	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca3	Detalle de las actividades a cargo del estudiante		text	
elempre4	Nombre de la empresa donde haya trabajado		varchar(60)	
elactem4	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades
elanoin4	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin4	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi4	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses
elanofi4	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel14	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel24	Otro teléfono de la empresa donde haya trabajado		varchar(10)	
eljefdi4	nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargos4	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca4	Detalle de las actividades a cargo del estudiante		text	
elempre5	Nombre de la empresa donde haya trabajado		varchar(60)	
elactem5	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades
elanoin5	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin5	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi5	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses
elanofi5	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel15	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel25	Otro teléfono de la empresa donde haya trabajado		varchar(10)	

eljefdi5	Nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargos5	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca5	Detalle de las actividades a cargo del estudiante		text	
elempre6	Nombre de la empresa donde haya trabajado		varchar(60)	
elactem6	Actividad a la que se dedica la empresa en la que haya trabajado		varchar(40)	Lista de actividades
elanoin6	Año de inicio de la actividad laboral		varchar(4)	Lista de años
elmesin6	Mes de inicio de la actividad laboral		varchar(2)	Lista de meses
elmesfi6	Mes de finalización de la actividad laboral		varchar(2)	Lista de meses
elanofi6	Año de finalización de la actividad laboral		varchar(4)	Lista de años
eltel16	Teléfono de la empresa donde haya trabajado		varchar(10)	
eltel26	Otro teléfono de la empresa donde haya trabajado		varchar(10)	
eljefdi6	Nombre del jefe directo con el que trabajó el estudiante		varchar(60)	
elcargos6	Cargo que desempeño en la empresa el estudiante		varchar(30)	
elactca6	Detalle de las actividades a cargo del estudiante		text	
diventas	Indica si el estudiante está dispuesto a trabajar en ventas		varchar(2)	
disecre	Indica si el estudiante está dispuesto a trabajar en secretaría		varchar(2)	
dimerca	Indica si el estudiante está dispuesto a trabajar en mercadeo		varchar(2)	
diproducto	Indica si el estudiante está dispuesto a trabajar en producción		varchar(2)	
diedu	Indica si el estudiante está dispuesto a trabajar en educación		varchar(2)	
diatecli	Indica si el estudiante está dispuesto a trabajar en atención al cliente		varchar(2)	
dicontri	Indica si el estudiante está dispuesto a trabajar en contabilidad - tributaria		varchar(2)	
diaficar	Indica si el estudiante está dispuesto a trabajar en trabajos		varchar(2)	

	afines a su carrera			
ditiecom	Indica si el estudiante está dispuesto a trabajar a tiempo completo		varchar(2)	
dimedtie	Indica si el estudiante está dispuesto a trabajar a medio tiempo		varchar(2)	
diporhor	Indica si el estudiante está dispuesto a trabajar por horas		varchar(2)	
diporpro	Indica si el estudiante está dispuesto a trabajar por proyecto		varchar(2)	
diturrot	Indica si el estudiante está dispuesto a trabajar en turnos rotativos		varchar(2)	
dihornoc	Indica si el estudiante está dispuesto a trabajar en horario nocturno		varchar(2)	
diotrciu	Indica si el estudiante está dispuesto a trabajar en otra ciudad		varchar(2)	
didenreg	Indica si el estudiante está dispuesto a trabajar dentro de la región		varchar(2)	
didenpai	Indica si el estudiante está dispuesto a trabajar dentro del país		varchar(2)	
difuepai	Indica si el estudiante está dispuesto a trabajar fuera del país		varchar(2)	
divehtra	Indica si el estudiante puede disponer su vehículo para el trabajo		varchar(2)	
dicomp	Indica si el estudiante puede disponer de su computadora para el trabajo		varchar(2)	
diinfrel	Información relevante que no haya sido completada		text	
estado	Campo que indica si la hoja de vida se visualizará en una consulta		tinyint(1)	0, 1
Ip	Dirección IP de la máquina en donde se hayan realizado		varchar(15)	



	cambios sobre la hoja de vida correspondiente			
Fecha	Fecha en la que se realizó el último cambio en la hoja de vida		date	
Hora	Hora en la que se realizó el último cambio en la hoja de vida		time	
validación	Indica si la hoja de vida del estudiante fue verificada		varchar(2)	si, no
Usuario	Código del usuario que realizó el último cambio en la hoja de vida		varchar(10)	

**Tabla: usuarios**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>	<b>Detalle</b>
usuario	Código del usuario que manipulará las hojas de vida	Pk	varchar(10)	
nombres	Nombres del usuario que manipulará las hojas de vida		varchar(30)	
apellidos	Apellidos del usuario que manipulará las hojas de vida		varchar(30)	
telefono1	Teléfono del usuario que manipulará las hojas de vida		varchar(10)	
telefono2	Otro teléfono del usuario que manipulará las hojas de vida		varchar(10)	
clave	Contraseña del usuario que manipulará las hojas de vida		varchar(10)	
estado	Estado en el que se encuentra el usuario		varchar(1)	A, D
tipo	Tipo de usuario		varchar(3)	usr, adm, est

**Tabla: sugerencias**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>	<b>Detalle</b>
Código	Código del estudiante que hace la sugerencia	Pk	varchar(6)	
sugerencia	Sugerencia del estudiante		text	
Fecha	Fecha de la sugerencia		date	
Hora	Hora de la sugerencia		time	
Ip	Ip del computador desde donde el estudiante realiza la sugerencia		varchar(15)	

**Tabla: paises**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>	<b>Detalle</b>
Código	Código del país	PK	varchar(4)	
Nombre	Nombre del país	Fk	varchar(38)	Lista de países

## 2.7 Mapa del Sitio.

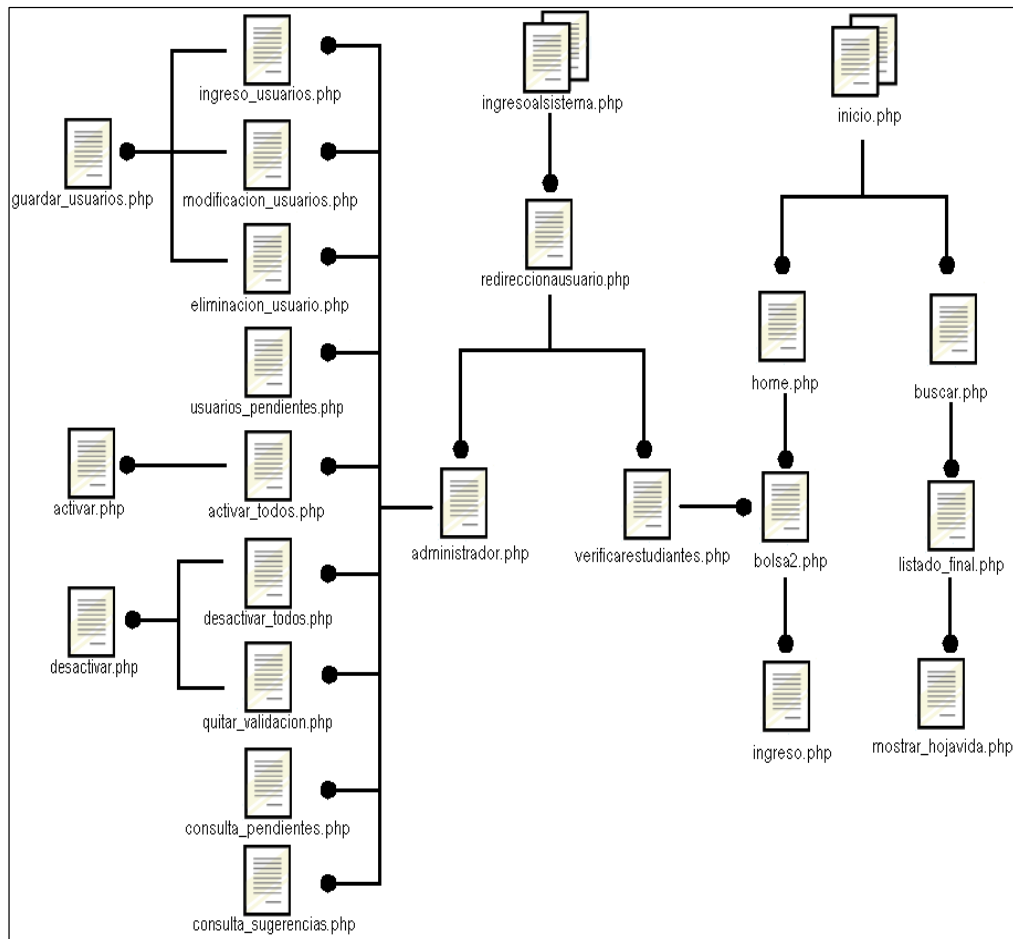


Gráfico 2.7 Mapa del Sitio

## **CAPÍTULO 3**

### **IMPLEMENTACIÓN Y PRUEBAS**

#### **3.1 Creación de Tablas.**

Luego de haber realizado el análisis del sistema se ha procedido a la creación de la base de datos “bolsatrabajo” y sus tablas respectivas.

La creación de la tabla “hoja de vida” se basa en la información brindada inicialmente, a la que se ha creído conveniente completar e incrementar ciertos atributos necesarios.

Como resultado de la tabla hoja de vida se obtuvo alrededor de 250 atributos los mismos que han ido incrementando a lo largo del desarrollo. De la misma manera surgió la necesidad de crear nuevas tablas con sus respectivos atributos tal es el caso de la tabla usuarios, sugerencias y la tabla países.

Además se utilizaron varias tablas de la base de datos Oracle de la universidad que son necesarias y básicas para el desarrollo de este proyecto. A continuación se indica dichas tablas únicamente con los atributos que se han utilizado de cada una de ellas.

**Tabla: TG\_PERSONA**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_PERSONA	Código del estudiante o del colegio	PK	number(13)
NOMBRE	Nombres y apellidos del estudiante		varchar(2)
EMAIL	Email del estudiante		varchar(2)
TIPO_IDENTIFICACION	Indica si es cédula, pasaporte o libreta militar		varchar(2)
IDENTIFICACION	Número de cédula		varchar(2)

**Tabla: TG\_DIRECCION**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_PERSONA	Código del estudiante	PK	number(8)
DIRECCIÓN	Dirección del estudiante		varchar(50)
COD_PAIS	País de origen del estudiante		varchar(3)

**Tabla: TG\_COLEGIO**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_COLEGIO	Código del colegio del que procede el estuiante	PK	number(8)

**Tabla: TG\_NATURAL**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_NATURAL	Código del estudiante	PK	number(13)
SEXO	Dirección del estudiante		varchar(2)
FECHA_NACIMIENTO	País de origen del estudiante		Date
ESTADO_CIVIL	Indica el estado civil del estudiante		Varchar(2)

**Tabla: TG\_TELEFONO**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_PERSONA	Código del estudiante	PK	number(8)
TELEFONO	Teléfono del estudiante		varchar(50)

**Tabla: TE\_CARRERA**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_CARRERA	Código de carrera de la Universidad	PK	number(3)
NOMBRE	Nombre de la Carrera		varchar(100)
COD_FACULTAD	Código de la Facultad		varchar(2)
CORRESPONDECIA_AUTOMATICA	Indica si la carrera esta activa o inactiva		varchar(1)

**Tabla: TG\_ESTUDIANTE**

<b>Atributo</b>	<b>Descripción</b>	<b>Definición</b>	<b>Tipo</b>
COD_ESTUDIANTE	Código del estudiante	PK	number(8)
COD_COLEGIO	Código del colegio		varchar(2)
COD_CARRERA	Código de la carrera		varchar(3)

Las tablas descritas anteriormente fueron creadas en una nueva base de datos denominada “computo”, la cual únicamente servirá para la realización de pruebas en caso de no tener acceso al servidor de la universidad, ya que el servicio FTP está disponible solo en la red de la institución.

Para tener acceso a la base de datos Oracle de la universidad se solicitó ayuda en el centro de cómputo.

### **3.2 Creación del Sitio Web.**

A lo largo del desarrollo del sitio web se ha realizado una serie de modificaciones como por ejemplo agregación y eliminación de atributos en las diferentes tablas, cambios en el diseño del sitio web, forma de visualizar los mensajes, menús, entre otros. Todo esto a partir de reuniones realizadas con las personas involucradas en el proyecto, además de la implementación de nuevas ideas que mejoran la funcionalidad del sistema.

Por otro lado todo el sistema ha sido implementado con variables de sesión para permitir controlar la seguridad del sistema, lo que se expondrá de forma detallada en el siguiente apartado. Además las variables de sesión permiten pasar parámetros que se usan durante toda la sesión de un usuario.

En este apartado se deben tener en cuenta las siguientes definiciones:

**Usuario Final:** Estudiantes egresados y/o graduados o personas que harán uso de la bolsa de trabajo para el ingreso y búsquedas de hojas de vida respectivamente.

**Usuario Validador:** Persona encargada de verificar que la información ingresada en la hoja de vida sea correcta comparándola con la documentación entregada.

**Usuario Administrador:** Persona responsable del manejo del sistema de la bolsa de trabajo; entre sus principales funciones está la creación, modificación y eliminación



de usuarios, además de validación y activación de hojas de vida, así como las respectivas consultas de las transacciones realizadas en el sistema en general.

### 3.2.1 Acceso de usuarios finales.

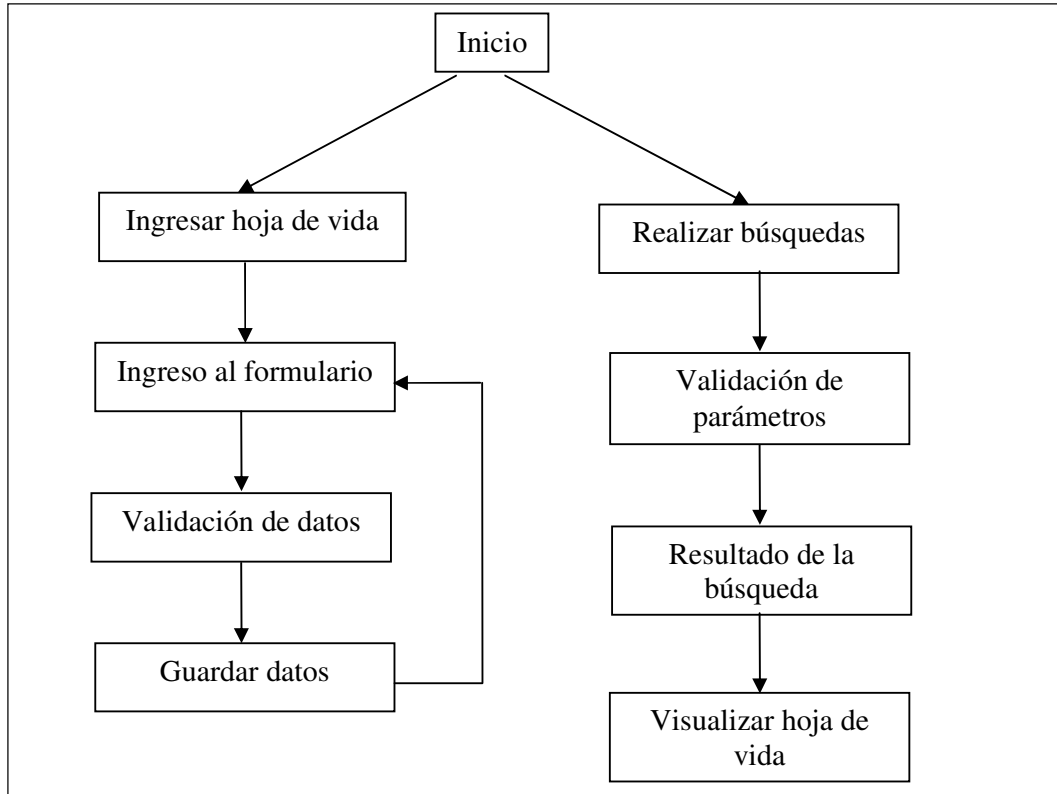


Gráfico 3.2.1 Interfaz de usuarios finales

#### **Inicio:**

- Los usuarios finales pueden elegir entre ingresar una hoja de vida o realizar búsquedas de hojas de vida activadas.

#### **Ingresar hoja de vida.**

- Permite al estudiante ingresar su código y cédula de identidad para poder acceder al formulario de la hoja de vida.

**Ingreso al Formulario:**

- Visualización de formulario en donde el alumno introduce o modifica sus datos. En caso de que sea la primera vez que el estudiante haga uso de la bolsa de trabajo, los datos se cargarán automáticamente de la base de datos Oracle de la universidad, dichos datos pueden ser nombres, apellidos, género, tipo de identificación, número de cédula, teléfono, dirección, estado civil, colegio del cual proviene el alumno y carrera que cursa o cursó en la Universidad del Azuay. Caso contrario, los datos se obtendrán de la base de datos MySQL de la bolsa de trabajo “bolsatrabajo”. En ambos casos se utiliza la base de datos Oracle para cargar lista de colegios, carreras de pregrado y carreras de postgrado de la Universidad del Azuay.

**Validación de datos:**

- Toda la información ingresada será validada; en caso de error se desplegará en el navegador una nueva ventana con la lista de los errores respectivos, indicando el tipo de error, el lugar y el dato erróneo. En caso de no existir error, los datos serán creados o modificados en las tablas “hojajvida” y “sugerencias”, según sea el caso.

**Realizar búsquedas:**

- Los usuarios finales pueden realizar búsquedas abiertas de hojas de vida según sus necesidades, es decir, seleccionar por medio de un check las opciones en las cuales se basará la búsqueda, sin tomar en cuenta las opciones no seleccionadas.

**Validación de parámetros:**

- Se mostrará en pantalla solo los campos que posean información en la base de datos de la bolsa de trabajo, como por ejemplo en los combos se cargaran las listas de colegios, carreras o países existentes.

### Visualización de hoja de vida:

- Visualización de los resultados de búsqueda. Cada hoja de vida mostrada se presenta en forma de vínculo para su acceso y visualización.

### 3.2.2 Acceso de usuarios validadores.

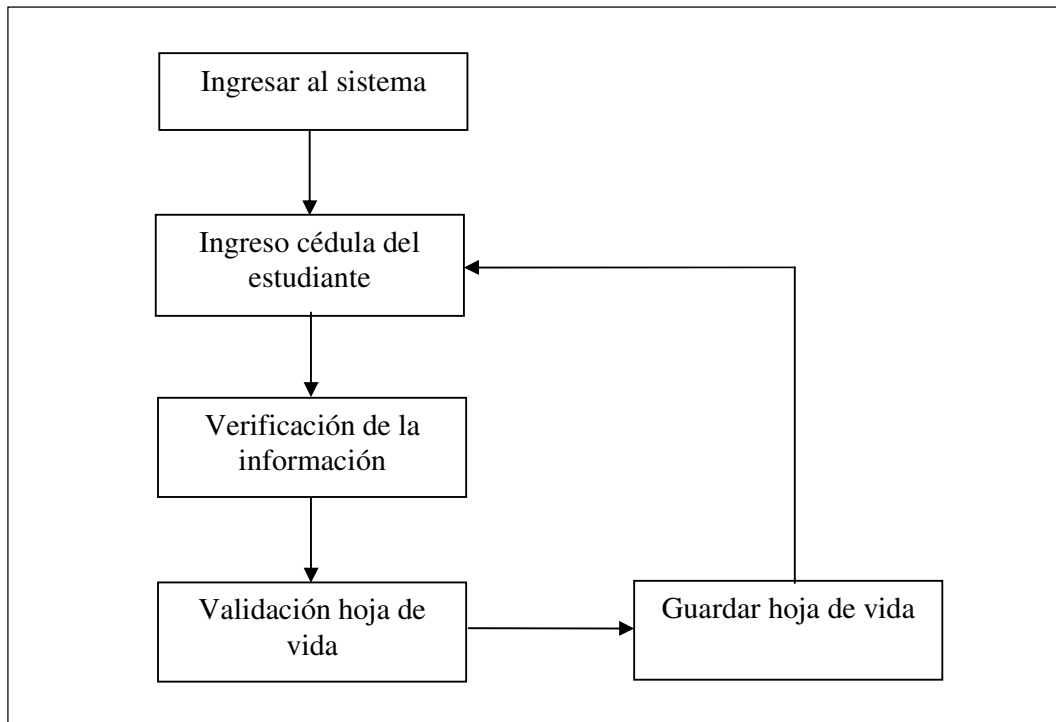


Gráfico 3.2.2 Interfaz de usuarios validadores

### Ingreso al sistema:

- Permite al usuario validador ingresar su código y contraseña para poder acceder a la validación de la hoja de vida.

### Ingreso cédula del estudiante:

- Permite el ingreso del número de cédula del estudiante cuya hoja de vida va a ser validada.

### **Verificación de la información:**

- Verificación de los datos ingresados por los estudiantes en sus respectivas hojas de vida con la documentación que respalde dicha información.

### **Validación hoja de vida:**

- Toda la información modificada será validada; en caso de error se desplegará en el navegador una nueva ventana con la lista de los errores respectivos, indicando el tipo de error, el lugar y el dato erróneo. En caso de no existir error, los datos serán modificados en la tabla “hojavidas”.

### **Guardar hoja de vida:**

- Modificación de los datos manipulados, incluyendo la asignación del valor “si” al campo validación de la tabla “hojavidas”, lo que indica que la información ingresada concuerda con la documentación presentada.

### **3.2.3 Acceso del usuario administrador.**

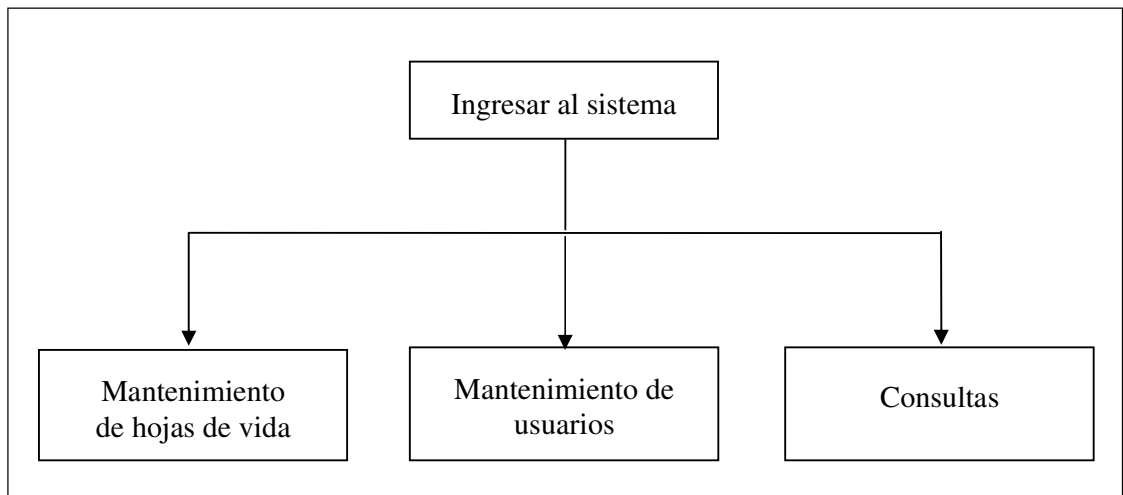


Gráfico 3.2.3 Interfaz de usuario administrador.

**Ingreso al Sistema:**

- Permite al usuario administrador ingresar su código y contraseña para poder acceder al sistema, creando automáticamente al usuario administrador en caso de no existir, siendo usuario “rootadm” y contraseña “rootadm”, lo que puede ser modificado posteriormente.

**Mantenimiento de hojas de vida:**

- Permite al administrador tener acceso a la validación de hojas de vida.
- Permite al administrador tener acceso a la activación de hojas de vida ya sea ingresando la cédula del estudiante o seleccionando de una lista de hojas de vida validadas agrupadas según el usuario validador.
- Quitar validación o desactivar hojas de vida respectivamente, basadas en búsquedas “por alumno” y “por rango de fechas”; el resultado de la búsqueda las hojas de vida se presentan en forma de vínculo para su acceso y visualización y además se presenta la opción de manipular una o varias hojas de vida.

**Mantenimiento de usuarios:**

- Mantenimiento de usuarios (ingreso, modificación, eliminación, consulta). El administrador es la única persona que está en capacidad de cambiar el estado de los usuarios de activo a inactivo o viceversa para impedir u otorgar accesos al sistema, así como la modificación de toda la información incluyendo contraseñas de todos los usuarios.

**Consultas:**

- Realización de consultas de hojas de vida pendientes (de activar o de validar), de las sugerencias realizadas por los estudiantes y de los usuarios validadores ya sean activos o inactivos.
- Como resultado de la consulta de hojas de vida pendientes las hojas de vida se presentan en forma de vínculo para su acceso y visualización.

### **3.3 Seguridades.**

#### **3.3.1 Seguridades para el ingreso a la hoja de vida.**

- El código del estudiante y número de cédula son datos obligatorios, que serán verificados en la base de datos Oracle la primera vez que el estudiante haga uso de la bolsa de trabajo para impedir que personas ajenas a la institución ingresen sus datos en la hoja de vida ya que este servicio es exclusivo para estudiantes de la Universidad del Azuay.
- Si el usuario accede nuevamente a la hoja de vida los datos serán verificados en la tabla “hojavida” de la base de datos MySQL.

#### **3.3.2 Seguridades para el acceso al sitio web.**

- Cualquier página del sitio web es controlada por variables de sesión que dependiendo del usuario restringe accesos a páginas o a cierta información.
- Existen tres tipos de usuarios: administradores (adm), usuarios validadores (usr) y estudiantes (est).
- El administrador tiene permisos de acceso a todas las páginas.
- Los usuarios validadores tienen acceso solamente para validar hojas de vida.
- Los estudiantes únicamente pueden ingresar o modificar su propia hoja de vida mientras esta no se encuentre validada.
- Tanto los usuarios como el administrador tienen permisos para modificar y visualizar todos los datos de la hoja de vida del estudiante a excepción de los campos de información relevante y sugerencias.
- El sistema no permite tener más de una sesión abierta al mismo tiempo en el mismo navegador, si esto sucede se cierra la sesión más antigua y se cambia de sesión automáticamente a la nueva, si el usuario no tiene los permisos correspondientes no podrá continuar con su trabajo.
- A pesar de que el administrador tenga acceso a todas las páginas tiene que seguir un trayecto establecido antes de acceder a determinadas páginas, sino se visualizará el error respectivo como por ejemplo:

Usuario: administrador.

Destino: ingreso.php

Ruta:

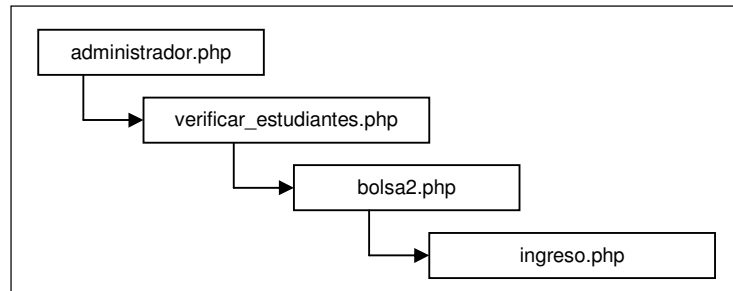


Gráfico 3.3.2 Ejemplo de uso de variables de sesión.

Explicación: Si antes de iniciar sesión cualquier usuario digita en el navegador ruta/administrador.php se visualizará un mensaje de error de acceso porque no ha ingresado su usuario y contraseña.

Si un administrador tiene iniciada su sesión y digita en el navegador ruta/verificar\_estudiantes.php no tendrá ningún problema, por el contrario si digita ruta/bolsa2.php o ruta/ingreso.php se visualizará un mensaje de error de acceso porque la ruta correcta guarda en sus variables de sesión los permisos para poder modificar los datos de bolsa2.php y llegar a ingreso.php.

### 3.4 Pruebas y Corrección de errores.

En todos los procesos de desarrollo del sitio web se fueron realizando continuamente las pruebas pertinentes y de la misma manera corrigiendo los errores encontrados.

Para poder realizar las pruebas inicialmente se guardaron datos ficticios en las base de datos “Computo” que es la base de datos creada para simular a la base de datos de la universidad. A partir de esto se pudieron cargar los combos y mostrar cierta información del estudiante.

Uno de los principales errores cometidos inicialmente fue realizar las pruebas solamente con el navegador “Internet Explorer”; al momento de usar otro navegador

el sitio web no era visible en su totalidad y en otro caso los datos no se cargaba en los combos entre otros detalles. Por ese motivo se decidió que para futuras pruebas también utilizaremos el navegador Opera cuyas normas son mas estrictas en relación con otros navegadores.

Entre las pruebas más importantes están:

- Obviar información de los campos requeridos con el fin de comprobar la validación de campos y visualización de errores.
- Ingreso de fechas incorrectas.
- Ingreso de cadenas en campos numéricos.
- Ingreso de menos caracteres con campos que requieren un número fijo de caracteres. Por ejemplo número de teléfono no puede ser menor a seis dígitos.
- En la hoja de vida existen secciones que deben ser ingresadas en su totalidad o evitar su ingreso, tal es el caso de estudios de postgrado; para esto ingresamos en cada una de las secciones datos incompletos para comprobar validación de datos en conjunto y visualización de errores.
- Validación de rangos de fechas.
- Ingreso de información en minúsculas para luego guardar y recuperar los datos en mayúsculas.
- Ingreso de información con caracteres en blanco, al inicio y final del campo para luego guardar y recuperar los datos sin ellos.
- Ingreso de datos inexistentes en la base de datos como por ejemplo código de estudiante y número de cédula, o código de usuario y contraseña incorrectos.

Realizadas las respectivas pruebas en nuestro computador personal se procedió a subir al servidor de la universidad los archivos realizados hasta el momento y realizar las mismas pruebas para comprobar el normal funcionamiento del sitio web.

Finalmente se hizo una última prueba para la cual se contó con la ayuda de estudiantes de la Universidad del Azuay quienes ingresaron su hoja de vida con datos reales.



Se crearon varios usuarios validadores para que éstos puedan hacer uso de del sitio web y detectar posibles errores. Cabe mencionar que las pruebas no solo fueron efectuadas por nuestra parte sino también la Ing. Katherine Ortiz realizó varias pruebas indicándo los errores encontrados o cambios que se deberían realizar en el sitio web.

Hacemos mención que los estudiantes que nos brindaron su ayuda son ajenos al proyecto por lo cual son más susceptibles a cometer errores de ingreso y a detectarlos.

Los errores encontrados han sido solucionados y se han tomado en cuenta las sugerencias de los estudiantes para permitir que la página tenga una interfaz más amigable con el usuario final.

## **CAPÍTULO 4**

### **MANUAL DE PROGRAMADOR**

#### **4.1 Introducción.**

El propósito de este manual, es simplificar la manipulación y futura modificación del sitio web de la Bolsa de Trabajo de la Universidad del Azuay.

Describiendo el propósito de cada uno de los archivos utilizados, junto con sus variables, funciones, parámetros, entre otros.

##### **4.1.1 Requerimientos de Software.**

El sistema ha sido colocado en el servidor Apache de la Universidad del Azuay. Para su programación se ha utilizado código HTML, PHP y JavaScript.

##### **4.1.2 Estructura del Sistema.**

La estructura del sitio web está dividida en dos partes:

- La primera parte que permite acceder a los estudiantes de la Universidad del Azuay para ingresar sus hojas de vida, y además permite también a las empresas y/o personas interesadas buscar el personal calificado según sus necesidades.
- La segunda parte que permite ingresar al sistema al administrador para tener un completo control sobre usuarios y hojas de vida; y a los usuarios validadores verificar la información introducida por los estudiantes.

## 4.2 Desarrollo

### 4.2.1 Función de los archivos.

A continuación se listan todos los archivos utilizados en el sitio web de la bolsa de trabajo de la Universidad del Azuay. Cada apartado contendrá 5 puntos básicos:

- Descripción del Archivo: Breve descripción de la finalidad del archivo.
- Modo de invocación: Modo en el que se invocará el archivo para poder ser utilizado.
- Lista de archivos dependientes: Lista de todos los archivos que se verán afectados en caso de que se realice un cambio sobre dicho archivo.
- Descripción de funciones que incorpora: Descripción de las funciones que se incorporan en el archivo, su finalidad, parámetros y valores de retorno.
- Descripción de variables relevantes. Descripción de variables

### 4.2.2 Archivo “activar.php”.

**Descripción.** Contiene el código PHP necesario para activar una o más hojas de vida, es decir escribir “1” en el campo “estado” de la tabla “hojavida”.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir algo similar al siguiente ejemplo:

```
<form name="formulario" action="activar.php" method="post">
```

#### **Lista de archivos dependientes.**

- No tiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, los valores de las variables de sesión "\$\_SESSION["accion"]" y "\$\_SESSION["tipo\_usuario]" deben llegar con los valores "guardar" y "adm" respectivamente; caso contrario, se mostrará en la página web un mensaje de error.

### **4.2.3 Archivo “activar\_todos.php”.**

**Descripción.** Contiene el código PHP necesario para que en caso de existir hojas de vida pendientes por activar visualice una lista de ellas, ordenadas por el usuario que validó las mismas. Se pueden seleccionar uno o más registros. Si no existe ningún registro pendiente, se visualiza el mensaje de error correspondiente.

### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

### **Lista de archivos dependientes.**

- activar.php

### **Descripción de funciones que incorpora.**

- `activar_desactivar(elemento)`: Función JavaScript que permite seleccionar o quitar la selección de todos los registros listados en una página web por medio de la selección de sus correspondientes checkbox, de hojas de vida pendientes por activar. Esta función no devuelve ningún valor.  
elemento: Este parámetro debe contener el nombre del formulario en el que se esté trabajando, es decir “document.nombreformulario”.
- `mostrar_listado($sql, $titulo, $error)`: Función PHP que muestra un listado de todas las hojas de vida validadas pendientes por activar, ordenada en grupos según el usuario validador. Cada registro tendrá un checkbox que permitirá al usuario escoger una o más hojas de vida para validar. Esta función no devuelve ningún valor.

\$sql: Este elemento guarda la consulta “sql” que se ejecutará

\$titulo: Este elemento guarda una cadena que define el título que se visualizará antes de mostrar el listado resultado

\$error: Este parámetro deberá llevar la cadena que se mostrará en caso de que la consulta “sql” no devuelva ningún resultado.

#### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION ["tipo\_usuario "]” debe llegar con el valor "adm" caso contrario, se mostrará en la página web el mensaje de error correspondiente.

#### **4.2.4 Archivo “administrador.php”.**

**Descripción.** Contiene el código PHP necesario para poder visualizar los menús que utilizará el administrador.

**Modo de invocación.** No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- activar.php
- activar\_todos.php
- bolsa2.php
- consulta\_pendientes.php
- consulta\_sugerencias.php
- desactivar.php
- desactivar\_todos.php
- eliminacion\_usuario.php
- guardar\_usuarios.php
- ingreso.php
- ingreso\_usuarios.php
- modificacion\_usuarios.php

- quitar\_validacion.php
- usuarios\_pendientes.php
- verificar\_estudiantes.php

### **Descripción de funciones que incorpora.**

- MM\_jumpMenuGo(selName,targ,restore): Función JavaScript creada automáticamente por Dreamweaver al generar un menú, que permite acceder a una página web específica, según la selección del usuario, luego de presionar el botón ir. Esta función va de la mano con las funciones “MM\_findObj(n, d)” y “MM\_jumpMenu(targ,selObj,restore)” creadas también automáticamente por Dreamweaver. Esta función no devuelve ningún valor.

selName: Nombre del combo, en donde se listan las diferentes páginas web a las que puede acceder el usuario.

targ: este parámetro debe contener la cadena “parent”.

restore: Este parámetro debe contener la cadena “0” para que cuando sea cargada la página “administrador.php” no quede seleccionada ninguna opción del menú.

- MM\_findObj(n, d): Función JavaScript creada automáticamente por Dreamweaver al generar un menú y que es llamada por la función JavaScript “MM\_jumpMenuGo(selName,targ,restore)”.
- MM\_jumpMenu(targ,selObj,restore): Esta función JavaScript es creada automáticamente por Dreamweaver al generar un menú y es llamada por la función JavaScript “MM\_jumpMenuGo(selName,targ,restore)”.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión “\$\_SESSION[“tipo\_usuario ”]” debe llegar con el valor “adm” caso contrario se mostrará en la página web el mensaje de error correspondiente.

#### **4.2.5 Archivo “bolsa2.php”.**

**Descripción.** En este archivo se podrán ingresar o modificar todos los datos de la hoja de vida del alumno, ya sea por los propios estudiantes, usuarios validadores o administrador del sistema. Además los usuarios validadores podrán validar en ella la hoja de vida y el administrador activarla.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir, algo similar al siguiente ejemplo:

```
<form name="formulario" action="bolsa2.php" method="post">
```

#### **Lista de archivos dependientes.**

- ingreso.php
- verificar\_estudiantes.php
- administrador.php

#### **Descripción de funciones que incorpora.**

- viajes(): Esta función está desarrollada en JavaScript y permite activar o desactivar todos los campos de la sección viajes dentro del archivo bolsa2.php, dependiendo del “value” del campo “vivivido” del mismo archivo. Si el “value” es igual a “si” todos los campos se activan caso contrario todos se desactivan.

#### **Descripción de variables relevantes.**

Para tener acceso a este archivo, el valor de la variable de sesión “\$\_SESSION[“tipo\_usuario”]” debe llegar con los valores “adm”, “usr”, o “est” caso contrario se mostrará en la página web el mensaje de error correspondiente. Cada tipo de usuario contará con diferentes permisos de visualización en la página.

Cuando es el estudiante quien hace el llamado a esta página, deberá pasar como parámetros el “código de estudiante” y la “cédula de identidad” en las variables o campos “codigo” y “cedula” respectivamente.

Cuando quien accede a esta página es un usuario validador o administrador, tan solo debe pasar el parámetro “cédula de identidad” por el campo “cedula”, además la variable de sesión "\$\_SESSION["nombre\_usuario"]", debe venir con su valor correspondiente.

A la variable de sesión "\$\_SESSION["accion"]", se le debe asignar siempre al inicio del documento la cadena “guardar” para que el momento del “submit” del formulario principal, se puedan realizar las acciones correspondientes del archivo “ingreso.php”. La variable de sesión "\$\_SESSION["tipo"]", deberá venir con sus valores respectivos de “activacion” o “validacion”, dependiendo quien haya llamado a la página, es decir el administrador o validador respectivamente.

La variable de sesión "\$\_SESSION["path"]", guardará la ubicación del servidor, en donde se encuentra la fotografía de un determinado estudiante, dicha información será necesaria en el archivo ingreso.php.

#### **4.2.6 Archivo “buscar.php”.**

**Descripción.** Contiene el código PHP necesario para visualizar un formulario de una búsqueda abierta que permita al usuario final listar alumnos según las condiciones del perfil deseado.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- listado\_final.php
- mostrar\_hojavida.php

#### **Descripción de funciones que incorpora.**

- seleccionar(): Función JavaScript que sirve antes de realizar un “submit” del formulario principal para que sean seleccionados todos los elementos que se hayan agregado en las listas de colegios, estudios de pregrado, y estudios de



postgrado; y puedan ser utilizados en el archivo “listado\_final.php”. Al final de la función, se realiza el “submit” del formulario.

- cargar\_provincias(pais): Función JavaScript que permite cargar las provincias del Ecuador, en caso de que el “value” del país seleccionado sea igual a “1”.  
pais: Este parámetro debe contener el value del combo país que se visualice en el formulario.
- validar(campo): Función que valida el rango de edades ingresados en el formulario de “buscar.php”.  
campo: Nombre del campo que validará.
- activar(campo): Función encargada de activar o desactivar un conjunto de campos cuando se seleccione el “checkbox” correspondiente.  
campo: Nombre del “checkbox”.
- agregar\_a\_lista(dato, lista): Función que agrega a una lista, un elemento que el usuario haya seleccionado en el “combobox” correspondiente a esta lista, luego de haber presionado el botón de agregar(>>).  
dato: Nombre del combo de donde se extraerá algún dato para agregarlo a su lista correspondiente  
lista: Nombre de la lista en donde se agregará el dato seleccionado en el combo.
- quitar\_de\_lista(lista): Función que elimina un elemento que seleccione de una lista luego de presionar el botón eliminar (<<).  
lista: Nombre de la lista en donde se eliminará el dato seleccionado por el usuario.

#### **Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### **4.2.7 Archivo “cabecera.php”.**

**Descripción.** Contiene código HTML que servirá para pintar la cabecera de todas las páginas web de la Bolsa de Trabajo. Esto es el gráfico de la Universidad del Azuay

con su logotipo respectivo, el título de la página (“Universidad del Azuay”) y el título del proyecto (“Bolsa de Trabajo”).

### **Modo de invocación.**

Para utilizar este archivo, es necesario, tan solo realizar un “include” del mismo.

### **Lista de archivos dependientes.**

- activar.php
- activar\_todos.php
- administrador.php
- bolsa2.php
- buscar.php
- consulta\_pendientes.php
- consulta\_sugerencias.php
- desactivar.php
- desactivar\_todos.php
- eliminación\_usuarios.php
- guardar\_usuarios.php
- home.php
- ingreso.php
- ingreso\_usuarios.php
- ingresoalsistema.php
- inicio.php
- listado\_final.php
- modificacion\_usuarios.php
- mostrar\_hojavida.php
- quitar\_validacion.php
- redireccionausuario.php
- usuarios\_pendientes.php
- verificar\_estudiantes.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

**4.2.8 Archivo “conexion\_mysql.php”.**

**Descripción.** Contiene código PHP que sirve para realizar la conexión con la base de datos mysql.

**Modo de invocación.**

Para utilizar este archivo, es necesario, tan solo hacer un “include” del mismo.

**Lista de archivos dependientes.**

- activar.php
- activar\_todos.php
- bolsa2.php
- buscar.php
- consulta\_pendientes.php
- consulta\_sugerencias.php
- desactivar.php
- desactivar\_todos.php
- eliminación\_usuarios.php
- guardar\_usuarios.php
- ingreso.php
- listado\_final.php
- modificacion\_usuarios.php
- mostrar\_hojavida.php
- quitar\_validacion.php
- usuarios\_pendientes.php
- redireccionausuario.php

- verblob.php
- verificar\_estudiantes.php

#### **Descripción de funciones que incorpora.**

Las funciones de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que contienen la clave de acceso a la base de datos MySQL.

#### **Descripción de variables relevantes.**

Las variables de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que dicho archivo contienen la clave de acceso a la base de datos MySQL.

#### **4.2.9 Archivo “consulta\_pendientes.php”.**

**Descripción.** Contiene el código PHP necesario para visualizar un combo que indica dos opciones “Pendientes por validar” y “Pendientes por activar”, y para que luego de escoger una de ellas mostrar en pantalla un listado con las hojas de vida correspondientes a su búsqueda pertinente.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- Este archivo no tiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- `mostrar_listado($sql, $titulo, $error)`: Función PHP que muestra un listado de todas las hojas de vida pendientes por validar o activar según la opción que el usuario haya seleccionado en un “combobox”. Esta función no devuelve ningún valor.  
\$sql: Este elemento guarda la consulta “sql” que se ejecutará

\$titulo: Este elemento guarda una cadena que define el título que se visualizará antes de mostrar el listado resultado

\$error: Este parámetro deberá llevar la cadena que se mostrará en caso de que la consulta “sql” no devuelva ningún resultado.

#### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario"]" debe llegar con el valor "adm" caso contrario se mostrará en la página web el mensaje de error correspondiente.

La variable de sesión "\$\_SESSION["nombre\_usuario"]" debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

#### **4.2.10 Archivo “consulta\_sugerencias.php”.**

**Descripción.** Contiene el código PHP necesario para visualizar las sugerencias y comentarios realizados por los estudiantes el momento de ingresar su hoja de vida. Para visualizar esta consulta se puede escoger de entre dos opciones, mostrar todas las sugerencias realizadas, o mostrarlas según un rango de fechas.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- Este archivo no tiene archivos dependientes.

#### **Descripción de funciones.**

- desactivar\_casillas(): Función JavaScript que permite activar o desactivar un conjunto de campos que pertenece a la opción de “mostrar según rango de fechas”.
- eserror(lugar, tipo, datoerroneo, i): Función JavaScript que guarda en el vector “errores”, los errores encontrados el momento de validar el rango de

fechas antes de realizar la búsqueda correspondiente. Esta función devuelve el número de error encontrado hasta el momento.

lugar: Indica el lugar en donde se ha encontrado el error.

tipo: Indica el tipo de error que se ha producido.

datoerroneo: Visualiza cual es el dato introducido incorrectamente.

I: Devuelve el número de error correspondiente.

- `valida()`: Función JavaScript que valida el rango de fechas introducido, y devuelve “verdadero” cuando no existen errores, y “falso” cuando se haya encontrado uno o más de ellos.
- `comparar_fechas(diain, mesin, añoin, diafi, mesfi, añofi)`: Función llamada desde la función `valida()` que compara dos rangos de fechas. Esta función devuelve “verdadero” cuando existe un error en la comparación de las fechas, caso contrario devuelve “falso”.

dia<sub>in</sub>: Día de la fecha de inicio.

mes<sub>in</sub>: Mes de la fecha de inicio.

año<sub>in</sub>: Año de la fecha de inicio.

dia<sub>fi</sub>: Día de la fecha fin.

mes<sub>fi</sub>: Mes de la fecha fin.

año<sub>fi</sub>: Año de la fecha fin.

- `valida_fecha(d, m, a)`: Función JavaScript que comprueba la validez de una fecha. Devuelve “falso” si no ha encontrado error.

d: Día de la fecha que se va a validar.

m: Mes de la fecha que se va a validar.

a: Año de la fecha que se va a validar.

- `cargar_dia($dia)`: Esta función PHP permite que se cargue un combo con todos los días del mes, en caso de ser necesario se visualizará el día indicado. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el número que corresponde al día que se visualizará. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”. Esta función no devuelve ningún valor.

\$dia: Este parámetro debe contener el día que se desea visualizar por defecto en el combo, si no se requiere que se visualice ningún dato se debe enviar una cadena en blanco.

- `convertir_desde($dia, $mes, $ano)`: Esta función PHP permite restar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.

\$dia: Indica el día de la fecha que se va a convertir.

\$mes: Indica el mes de la fecha que se va a convertir.

\$ano: Indica el año de la fecha que se va a convertir.

- `convertir_hasta($dia, $mes, $ano)`: Esta función PHP permite sumar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.

\$dia: Indica el día de la fecha que se va a convertir.

\$mes: Indica el mes de la fecha que se va a convertir.

\$ano: Indica el año de la fecha que se va a convertir.

- `mostrar_listado($sql, $error)`. Esta función PHP permite visualizar el listado correspondiente basándose en una consulta "sql" realizada previamente, dependiendo del tipo de búsqueda que se va a realizar.

\$sql: Parámetro que indica la consulta "sql" que se va a realizar.

\$error: Parámetro que indica el mensaje que se visualizará en caso de error, o mensaje que se desee dar al usuario.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario"]" debe llegar con el valor "adm", caso contrario se mostrará en la página web el mensaje de error correspondiente. La variable de sesión "\$\_SESSION["nombre\_usuario"]" debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

#### **4.2.11 Archivo “datos\_usuario.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para dibujar en pantalla una tabla con los datos necesarios que deben ser visualizados dependiendo el tipo de acción que se vaya a realizar, que puede ser “ingreso de usuarios”, “modificación de usuarios” o “eliminación de usuarios”.

#### **Modo de invocación.**

Para utilizar este archivo es necesario tan solo realizar el “include” del mismo.

#### **Lista de archivos dependientes.**

- eliminación\_usuarios.php
- ingreso\_usuarios.php
- modificacion\_usuarios.php
- guardar\_usuarios.php

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

#### **Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### **4.2.12 Archivo “desactivar.php”.**

**Descripción.** Contiene el código PHP necesario para desactivar o quitar la validación de una o más hojas de vida, es decir escribir “0” en el campo “estado” o escribir “no” en el campo “validación” de la tabla “hojajvida”, dependiendo del valor de la variable de sesión “\$\_SESSION[“quitar”]”.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario es decir algo similar al siguiente ejemplo:

```
<form name="formulario" action="desactivar.php" method="post">
```



#### **Lista de archivos dependientes.**

- Este archivo no contiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

#### **Descripción de variables relevantes.**

Para poder utilizar este archivo los valores de las variables de sesión "\$\_SESSION["accion"]", "\$\_SESSION["tipo\_usuario"]" y "\$\_SESSION["quitar]" deben llegar con los valores "guardar", "adm" y "desactivar" o "quitar" respectivamente; caso contrario, se mostrará en la página web un mensaje de error.

Si la variable "\$\_SESSION["quitar]" es igual a "quitar" entonces escribirá "no" en el campo "validacion" de la tabla "hojavidas", caso contrario si es igual a "desactivar" escribirá "0" en el campo "estado" de la misma tabla.

La variable de sesión "\$\_SESSION["sql]" debe llegar con el sql correspondiente a la búsqueda efectuada en desactivar\_todos.php, además la variable de sesión "\$\_SESSION["nombre\_usuario]", debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

#### **4.2.13 Archivo "desactivar\_todos.php".**

**Descripción.** Contiene el código PHP necesario para realizar una búsqueda y visualizar los resultados en forma de una lista de las hojas de vida activadas. Dicha búsqueda se puede realizar por el código de usuario y cédula del mismo, o por rango de fechas de activación. Luego de visualizado este resultado, se pueden escoger uno o varios registros para ser desactivados luego de presionar el botón "Desactivar hojas de vida". Si no existe ningún registro, se visualiza el mensaje de error correspondiente.

### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

### **Lista de archivos dependientes.**

- desactivar.php

### **Descripción de funciones que incorpora.**

- desactivar\_casillas(): Función JavaScript que permite activar o desactivar un conjunto de campos que pertenece a la opción de “mostrar según rango de fechas”.
- eserror(lugar, tipo, datoerroneo, i): Función JavaScript que guarda en el vector “errores” los errores encontrados el momento de validar el rango de fechas antes de realizar la búsqueda correspondiente. Esta función devuelve el número de error encontrado hasta el momento.

lugar: Indica el lugar en donde se ha encontrado el error.

tipo: Indica el tipo de error que se ha producido.

datoerroneo: Visualiza cual es el dato introducido incorrectamente.

i: Devuelve el número de error correspondiente.

- valida(): Función JavaScript que valida el rango de fechas introducido, y devuelve “verdadero” cuando no existen errores, y “falso” cuando se hayan encontrado uno o más de ellos.
- comparar\_fechas(diain, mesin, anoin, diafi, mesfi, anofi): Función JavaScript llamada desde la función valida() que compara dos rangos de fechas. Esta función devuelve verdadero cuando existe un error en la comparación de las fechas, caso contrario devuelve falso.

diain: Día de la fecha de inicio.

mesin: Mes de la fecha de inicio.

anoin: Año de la fecha de inicio.

diafi: Día de la fecha fin.

mesfi: Mes de la fecha fin.

anofi: Año de la fecha fin.

- `activar_desactivar(elemento)`: Esta función JavaScript se emplea para seleccionar o quitar la selección de todos los registros listados en una página web por medio de la selección de sus correspondientes checkbox. Esta función no devuelve ningún valor.  
elemento: Este parámetro debe contener el nombre del formulario en el que se esté trabajando es decir “document.nombreformulario”.
- `valida_fecha(d, m, a)`: Función JavaScript que comprueba la validez de una fecha.  
d: Día de la fecha que se va a validar.  
m: Mes de la fecha que se va a validar.  
a: Año de la fecha que se va a validar.
- `cargar_dia($dia)`: Esta función PHP permite que se cargue un combo con todos los días del mes, en caso de ser necesario se visualizará el día indicado. Esta función no devuelve ningún valor.  
Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el número que corresponde al día que se visualizará. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.  
\$dia: Este parámetro debe contener el día que se desea visualizar por defecto en el combo, si no se requiere que se visualice ningún dato se debe enviar una cadena en blanco.
- `convertir_desde($dia, $mes, $ano)`: Esta función PHP permite restar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.  
d: Día de la fecha que se va a validar.  
m: Mes de la fecha que se va a validar.  
a: Año de la fecha que se va a validar
- `convertir_hasta($dia, $mes, $ano)`: Esta función PHP permite sumar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.

\$día: Indica el día de la fecha que se va a convertir.

\$mes: Indica el mes de la fecha que se va a convertir.

\$ano: Indica el año de la fecha que se va a convertir.

- `mostrar_listado($sql, $error)`. Esta función PHP permite visualizar el listado correspondiente, dependiendo de una consulta “sql” realizada previamente, dependiendo del tipo de búsqueda que se va a realizar.

\$sql: Parámetro que indica la consulta “sql” que se va a realizar.

\$error: Parámetro que indica el mensaje que se visualizará en caso de error, o mensaje que se desee dar al usuario.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión `$_SESSION["tipo_usuario"]` debe llegar con el valor "adm"; caso contrario se mostrará en la página web el mensaje de error correspondiente. Las variables de sesión `$_SESSION["quitar"]` y `$_SESSION["accion"]` deben contener las cadenas “desactivar” y “guardar” respectivamente para que pueda trabajar correctamente al llamar al archivo “desactivar.php”, además la variable `$_SESSION["nombre_usuario"]`, debe venir con su valor correspondiente, para que pueda ser visualizado en el menú.

#### **4.2.14 Archivo “eliminacion\_usuario.php”.**

**Descripción.** Contiene el código PHP necesario para permitir al administrador ingresar el código del usuario que desee eliminar y luego junto con el archivo “datos\_usuario.php” visualizar los datos de dicho usuario, para que el administrador pueda estar seguro que es el usuario que va a eliminar.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

**Lista de archivos dependientes.**

- guardar\_usuarios.php
- datos\_usuario.php

**Descripción de funciones que incorpora.**

- cargar\_buscar(). Función PHP que permite cargar un formulario cada vez que la página sea recargada, para poder buscar el usuario que desee eliminar.

**Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario"]" debe llegar con el valor "adm"; caso contrario se mostrará en la página web el mensaje de error correspondiente. A la variable de sesión \$\_SESSION["tipo"] se le debe asignar el valor "eliminacion" para que pueda trabajar correctamente cuando se invoque el archivo "guardar\_usuarios.php", y "\$\_SESSION["nombre\_usuario"]" debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

La variable \$\_SESSION["buscar\_usuario"] con valor "true" llamará a la función "cargar\_buscar()", caso contrario llamará a la misma función y a continuación hará el "include" del archivo "datos\_usuario.php" con los datos resultantes de la búsqueda, y con las opciones de modificación desactivadas.

**4.2.15 Archivo "fin.htm".**

**Descripción.** Contiene código HTML que servirá para pintar el pie de página de todas las páginas del sitio web de la bolsa de trabajo. Estos datos son dirección, teléfono, e-mail, y otros.

**Modo de invocación.**

Para utilizar este archivo, es necesario, tan solo hacer un "include" del mismo.

**Lista de archivos dependientes.**

- activar.php

- activar\_todos.php
- administrador.php
- bolsa2.php
- buscar.php
- consulta\_pendientes.php
- consulta\_sugerencias.php
- desactivar.php
- desactivar\_todos.php
- eliminación\_usuarios.php
- guardar\_usuarios.php
- home.php
- ingreso.php
- ingreso\_usuarios.php
- ingresoalsistema.php
- inicio.php
- listado\_final.php
- modificacion\_usuarios.php
- mostrar\_hojavida.php
- quitar\_validacion.php
- redireccionausuario.php
- usuarios\_pendientes.php
- verificar\_estudiantes.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### **4.2.16 Archivo “funciones.php”.**

**Descripción.** Contiene una serie de funciones escritas en código PHP cuya ejecución será necesaria para el correcto funcionamiento de la Bolsa de Trabajo.

#### **Modo de invocación.**

Para utilizar este archivo es necesario, tan solo hacer un “include” del mismo.

#### **Lista de archivos dependientes.**

- activar.php
- activar\_todos.php
- administrador.php
- bolsa2.php
- buscar.php
- consulta\_pendientes.php
- consulta\_sugerencias.php
- desactivar.php
- desactivar\_todos.php
- eliminación\_usuarios.php
- guardar\_usuarios.php
- ingreso.php
- ingreso\_usuarios.php
- listado\_final.php
- modificacion\_usuarios.php
- mostrar\_hojavida.php
- quitar\_validacion.php
- redireccionausuario.php
- usuarios\_pendientes.php
- verificar\_estudiantes.php

#### **Descripción de funciones.**

Las funciones de este archivo son en su totalidad funciones PHP, y antes de utilizar cualquiera de ellas es necesario primero realizar un “include” de “funciones.php”.

- `cargar_mes($seleccionado, $nombre_mes)`: Esta función permite que se cargue un combo con todos los meses del año y de ser necesario se seleccionará por default uno de los meses. Esta función no devuelve ningún valor.

Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el número que corresponde al mes que se visualizará. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: En este primer parámetro se indicará el mes por defecto de la lista, para visualizarlo en la página resultado. Si no se desea ningún mes en especial, se debe enviar una cadena vacía.

\$nombre\_mes: El elemento que se envíe como parámetro debe ser el nombre del vector, que contenga todos los meses del año. En este caso el elemento que se envíe debe ser el array `$nombre_mes`, que ya está definido en el archivo `matrices.php`.

- `cargar_ano($seleccionado)`: Esta función permite cargar en un combo una lista de años que va desde 1960 hasta el año actual de ejecución de la página. Si es necesario, se visualizará por defecto uno de los años de la lista. Esta función no devuelve ningún valor.

Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el mismo del año que se visualizará. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: Este parámetro indica el año por defecto de la lista que será visualizado. Si no se desea que se visualice ningún año de la lista, se deberá enviar una cadena en blanco.

- `cargar_de_matriz($seleccionado, $matriz)`: Esta función es llamada siempre que se desee cargar en un combo datos desde una matriz de dos dimensiones. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será aquel que se encuentre en la primera dimensión



(\$matriz[0][x]) y se visualizará el contenido de la segunda dimensión (\$matriz[1][x]).

Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: En este parámetro se indicará el elemento por defecto de la lista, para visualizarlo en la página resultado. Si no se desea ningún elemento en especial, se debe enviar una cadena vacía.

\$nombre\_mes: El elemento que se envíe como parámetro debe ser el nombre de la matriz, que contenga los datos que serán cargados.

Esta función devuelve verdadero cuando el valor enviado dentro del parámetro “\$seleccionado”, coincide con uno de los valores de la matriz “\$matriz”; caso contrario devuelve falso.

- cargar\_de\_vector(\$seleccionado, \$vector): Esta función es llamada, siempre que se desee cargar en un combo desde un vector. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el valor que se visualice. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: En este parámetro se indicará el elemento por defecto de la lista. Si no se desea ningún elemento en especial, se debe enviar una cadena vacía.

\$vector: El elemento que se envíe como parámetro debe ser el nombre del vector, que contenga los datos que serán cargados.

Esta función devuelve verdadero cuando el valor enviado dentro del parámetro “\$seleccionado”, coincide con uno de los valores del vector “\$vector”; caso contrario devuelve falso.

- cargar\_porcentajes(\$seleccionado): Esta función carga un combo con valores del 1 hasta el 100 con saltos de 5, y permite dejar uno de esos valores listos para la visualización. Esta función no devuelve ningún valor. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el mismo que se cargue para la visualización. Por lo tanto, antes de

llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: En este parámetro se indicará el valor por defecto de la lista, para visualizarlo en la página resultado. Si no se desea ningún elemento en especial, se debe enviar una cadena vacía.

- `cargar_duracion($seleccionado)`: Esta función carga un combo con valores del 1 hasta el 100, y permite dejar uno de esos valores listos para la visualización. Esta función no devuelve ningún valor. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el mismo número que se cargue para la visualización.

Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”.

\$seleccionado: En este parámetro se indicará el valor por defecto de la lista, para visualizarlo en la página resultado. Si no se desea ningún elemento en especial, se debe enviar una cadena vacía.

- `dibujar_check($vname, $vnombre, $valor)`: Esta función permite dibujar un “input” tipo “checkbox” en una página. Esta función no devuelve ningún valor.

\$vname: Este parámetro indicará el nombre y el id del check

\$vnombre: Es el texto que se visualizará al lado del check

\$valor: Se debe pasar como parámetro “si” o “no”. Cuando se pasa por parámetro “si”, el checkbox quedará marcado.

- `dibujar_radio($vname, $valor)`: Esta función permite dibujar un “input” tipo “radio” en una página, con los valores predeterminados “si” y “no”. Esta función no devuelve ningún valor.

\$vname: Este parámetro indicará el nombre del radiobutton.

\$valor: Se debe pasar como parámetro “si” o “no”. Cuando se pasa por parámetro “si”, el radiobutton marcará la opción “Si”, si no marcará “No”.

- `inserta_tabuladores($numero)`: Esta función insertará el código HTML necesario para insertar un número determinado de tabuladores. Se utiliza

junto con la función `cierra_tabuladores($numero)`. Esta función no devuelve ningún valor.

\$numero: Este parámetro indica el número de tabuladores que se deseen insertar.

- `cierra_tabuladores($numero)`: Esta función inserta el código HTML necesario para cerrar un número determinado de tabuladores. Se utiliza junto con la función `inserta_tabuladores($numero)`. Esta función no devuelve ningún valor.

\$numero: Este parámetro indica el número de tabuladores que se deseen cerrar.

- `inserta_cabecera($titulo, $tamano, $alineacion)`: Esta función se encarga de pintar un título en forma de tabla en cada página desde donde se le llame. Esta función no devuelve ningún valor.

\$titulo: Se pasa como parámetro la cadena que describe la página desde donde se le hace la llamada.

\$tamano: Indica el tamaño en porcentaje que tendrá la tabla.

\$alineacion: Indica la alineación que tendrá el texto a ser visualizado dentro de la tabla, ésta se escribe en inglés (left, center, right).

- `poner_menu($home, $otro, $dirotro, $usuario, $ancho)`: Esta función se encarga de dibujar un menú en forma de tabla que se pintará en forma horizontal en cada página desde donde se le llame. El menú consta de dos partes, la superior que tiene el nombre del usuario activo, y la inferior en donde se pintan los vínculos del menú. Esta función no devuelve ningún valor.

\$home: Se pasa como parámetro una cadena con la dirección web de la página principal. El vínculo irá siempre descrito como “Home”

\$otro. Es una cadena que describe un vínculo cualquiera al que se desee hacer referencia con el parámetro “\$dirotro.”

\$dirotro: Es una cadena con la dirección web de la página que se desee acceder, según la descripción que se haya definido en el parámetro “\$otro”.

\$usuario: Se debe pasar como parámetro la variable que contenga el nombre del usuario que está utilizando el programa, para que sea visualizado en la parte superior del menú.

\$ancho: Es un valor numérico que indica el porcentaje con respecto a la página web que tendrá la tabla que se dibujará en forma de menú.

- `escribir_titulo($titulo, $alineacion, $tamano)`: Esta función se encarga de pintar un título que podría servir para algún subconjunto de opciones o datos en forma de tabla, de color azul. Esta función no devuelve ningún valor.

\$titulo: Cadena que visualizará el título.

\$alineacion: Indica la alineación que tendrá el texto a ser visualizado dentro de la tabla, ésta se escribe en inglés (left, center, right).

\$tamano: Indica el tamaño en porcentaje que tendrá la tabla que visualizará el título.

- `mostrar_error($error)`: Esta función visualizará en forma de tabla cualquier tipo de mensaje con fondo amarillo y fuente de color negra. Esta función no devuelve ningún valor.

\$error: Cadena que describe el mensaje que se desea visualizar.

#### **Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### **4.2.17 Archivo “guardar\_usuario.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para modificar la tabla “usuarios” de la base de datos MySQL “hojavida”, según el archivo desde donde provenga el llamado (ingreso\_usuario.php, modificacion\_usuarios.php, eliminacion\_usuario.php).

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir, algo similar al siguiente ejemplo:

```
<form name="formulario" action="guardar_usuarios.php" method="post">
```

#### **Lista de archivos dependientes.**

- Este archivo no tiene archivos dependientes.

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

Para poder utilizar este archivo, los valores de las variables de sesión "\$\_SESSION["tipo"]" y "\$\_SESSION["tipo\_usuario"]" deben llegar con los valores "modificación", "ingreso", "eliminación" y "adm" respectivamente; caso contrario se mostrará en la página web un mensaje de error, además la variable "\$\_SESSION["tipo\_usuario"]" debe contener su valor respectivo para que pueda ser visualizado en el menú.

**4.2.18 Archivo “home.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para permitir al estudiante ingresar su código y número de cédula para poder ingresar a la página “bolsa2.php”, e ingresar o modificar sus datos.

**Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

**Lista de archivos dependientes.**

- bolsa2.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

Se deben especificar los valores correspondientes de “ingreso” y “est” en las variables de sesión "\$\_SESSION["tipo"]" y "\$\_SESSION["tipo\_usuario"]" respectivamente para que pueda acceder y trabajar en el archivo “bolsa2.php”.

#### 4.2.19 Archivo “ingreso.php”.

**Descripción.** Contiene el código PHP necesario para preparar los datos que se van a guardar en la tabla “hojavida” de la base de datos MySQL “bolsatrabajo”, dependiendo de factores como el tipo de transacción, la acción a realizar y el usuario activo. Al decir preparar datos nos referimos a convertir en mayúsculas los datos introducidos, quitar espacios en blanco innecesarios, transformar valores, entre otros. Además, si los cambios son realizados por el estudiante, después de grabar los mismos, se mostrará la hoja de vida correspondiente.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir, algo similar al siguiente ejemplo:

```
<form name="formulario" action="ingresar.php" method="post">
```

#### **Lista de archivos dependientes.**

- Este archivo no tiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- porcentaje\_estado(\$estado, \$porcentaje): Función que devuelve el porcentaje de estudio, dependiendo de un estado de estudio de alguna carrera de pregrado o postgrado es decir, si el estado de estudios es “egresado”, “graduado” o “estudiante” tendrá como resultado “100” caso contrario devolverá el valor designado previamente.

\$estado: Parámetro que indica el estado de estudios de pregrado o postgrado recibido del formulario anterior.

\$porcentaje: Parámetro que indica el porcentaje de estudios de pregrado o postgrado recibido del formulario anterior.

#### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario "]" debe llegar con el valor “adm”, “usr” o “est”, además del valor “guardar” en "\$\_SESSION["accion"]"; caso contrario se mostrará en la página web el mensaje de error correspondiente.

Si el llamado a este archivo lo hace un estudiante que ingresa por primera vez sus datos en la bolsa de trabajo, la variable `$_SESSION["path"]` debe contener la dirección física en donde se encuentra alojada su fotografía en el servidor.

En caso de que la llamada a este archivo lo realice otro tipo de usuario, la variable de sesión `$_SESSION["tipo"]` debe tener el valor “activación” o “validacion”. Y Las variables de sesión `$_SESSION["nombre_usuario"]` y `$_SESSION["codigo_usuario"]` deben venir con sus valores correspondientes para que puedan ser visualizados en el menú y los mensajes.

Si la variable `$_SESSION["banprimeravez"]` tiene el valor de “falso” se realizará un “update” del registro correspondiente, caso contrario se realizará un “insert”. Si el usuario activo es un estudiante, se debe asignar el valor “hojavidia” a la variable `$_SESSION["desde"]` para poder visualizar al final la hoja de vida del mismo.

#### **4.2.20 Archivo “ingreso\_usuario.php”.**

**Descripción.** Contiene el código PHP necesario para permitir al administrador ingresar un nuevo usuario validador, detallando su código de usuario, apellidos, nombres, estado, teléfonos, y clave de acceso.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- guardar\_usuarios.php
- datos\_usuario.php

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

Para poder utilizar este archivo el valor de la variable de sesión "\$\_SESSION["tipo\_usuario "]" debe llegar con el valor "adm"; caso contrario se mostrará en la página web el mensaje de error correspondiente. A la variable de sesión \$\_SESSION["tipo"] se le debe asignar el valor "ingreso" para que pueda trabajar correctamente cuando se invoque el archivo "guardar\_usuarios.php", y "\$\_SESSION["nombre\_usuario"]" debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

**4.2.21 Archivo "ingresoalsistema.php".**

**Descripción.** Este archivo contiene el código PHP necesario para permitir a los validadores o administradores ingresar al sistema. El momento de realizar el "submit" de esta página, se llama el archivo "redireccionausuario.php".

**Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

**Lista de archivos dependientes.**

- redireccionausuario.php
- administrador.php
- verificar\_estudiantes.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

Si al recargar esta página, la variable de sesión \$\_SESSION["tipo\_usuario"] existe y tiene un valor, se destruirá la sesión y se la volverá a crear automáticamente.



#### **4.2.22 Archivo “inicio.php”.**

**Descripción.** Este archivo contiene el código PHP y HTML necesario para permitir a los usuarios escoger entre “ingresar una hoja de vida” y “consultar hojas de vida”.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

#### **Lista de archivos dependientes.**

- Este archivo no tiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

#### **Descripción de variables relevantes.**

- Este archivo no tiene variables relevantes.

#### **4.2.23 Archivo “listado\_final.php”.**

**Descripción.** Archivo con el código PHP necesario para listar los resultados obtenidos según la búsqueda realizada en el archivo “buscar.php”. Los registros se presentan en forma de vínculos que permitirán al usuario final dar clic sobre ellos para visualizar la hoja de vida del alumno correspondiente.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir algo similar al siguiente ejemplo:

```
<form name="formulario" action="listado_final.php" method="post">
```

### **Lista de archivos dependientes.**

- mostrar\_hojavida.php

### **Descripción de funciones que incorpora.**

- mostrar\_listado(\$sql, \$error). Esta función PHP permite visualizar el listado correspondiente basándose en una consulta “sql” realizada previamente, dependiendo del tipo de búsqueda que se va a realizar.

\$sql: Parámetro que indica la consulta “sql” que se va a realizar.

\$error: Parámetro que indica el mensaje que se visualizará en caso de error.

- cargar\_niveles\_herramienta(\$nombre\_campo, \$nivel\_herramienta). Esta función PHP devuelve una cadena en donde se incluirá cierto código que será posteriormente agregado a una consulta “sql” en proceso, según el nivel de conocimiento de una herramienta cualquiera.

\$nombre\_campo: Parámetro que define el nombre del campo que se evaluará.

\$nivel\_herramienta: Parámetro que define el nivel de la herramienta que se evaluará.

### **Descripción de variables relevantes.**

Se le debe asignar “listado\_final” a la variable de sesión \$\_SESSION[“desde”], para que se pueda visualizar correctamente la hoja de vida cuando se de clic sobre uno de los vínculos mostrados.

#### **4.2.24 Archivo “listaerrores.htm”.**

**Descripción.** Este archivo contiene el código HTML y JavaScript necesario para visualizar en una nueva ventana un listado con los errores de ingreso de las páginas, tales como campos en blanco, datos alfabéticos en campos numéricos, datos incompletos, entre otros.

#### **Modo de invocación.**

Para poder utilizar este archivo es necesario primero hacer el llamado al archivo o función de validación realizado en JavaScript, el cual deberá contener una variable

llamada “contadorerrores” que contenga el número de errores encontrado en el formulario, y un vector denominado “errores”, que contendrá separado por “@” tres datos (error, descripción del error y lugar del error). Y luego llamar a este archivo desde el botón que realice el submit del formulario que se validará, pidiendo que se ejecute la llamada a una nueva ventana. Como en el siguiente ejemplo:

```
<input name="Guardar" type="button" value="Guardar" onClick="if
(validar(document.usuarios)== true) {document.usuarios.submit();} else
{window.open('listaerrores.htm', 'Error' , 'width=800, height=600,
scrollbars=YES')}" /> ';
```

#### **Lista de archivos dependientes.**

- Este archivo no contiene archivos dependientes.

#### **Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

#### **Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### **4.2.25 Archivo “matrices.php”.**

**Descripción:** Contiene en código PHP un conjunto de vectores y matrices necesarios para el funcionamiento del sitio web.

#### **Modo de invocación.**

Para utilizar este archivo es necesario tan solo hacer un “include” del mismo.

#### **Lista de archivos dependientes.**

- bolsa2.php
- buscar.php
- consulta\_sugerencias.php
- desactivar\_todos.php
- mostrar\_hojavida.php

- quitar\_validacion.php

**Descripción de funciones que incorpora.**

- Este archivo no incorpora funciones.

**Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

**4.2.26 Archivo “mensaje.htm”.**

**Descripción.** Este archivo contiene el código HTML necesario para dibujar un mensaje en pantalla. Este mensaje es aquel que recuerda al usuario entregar la documentación que respalde la información ingresada en la hoja de vida en un lapso máximo de 72 horas.

**Modo de invocación.**

Para utilizar este archivo es necesario tan solo hacer un “include” del mismo.

**Lista de archivos dependientes.**

- bolsa2.php
- ingreso.php

**Descripción de funciones que incorpora.**

- Este archivo no incorpora funciones.

**Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

**4.2.27 Archivo “modificacion\_usuario.php”.**

**Descripción.** Contiene el código PHP necesario para permitir al administrador ingresar el código del usuario que desea modificar, y luego de presionar el botón

“Ver” visualizar los datos del mismo, con acceso a modificación de nombres, apellidos, teléfonos, estado y contraseña, los datos serán modificados totalmente luego de presionar el botón “Guardar” en donde se accede a la página “guardar\_usuarios.php”. En caso de existir algún error de validación se abrirá un “pop up” con la descripción de la información incorrecta.

### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación de este archivo.

### **Lista de archivos dependientes.**

- datos\_usuario.php
- guardar\_usuarios.php

### **Descripción de funciones que incorpora.**

- cargar\_buscar(). Función PHP que permite cargar un formulario con un cuadro de texto y un botón “submit” cada vez que la página es recargada, para poder buscar el usuario que desee modificar.

**Descripción de variables relevantes.** Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario"]" debe llegar con el valor "adm"; caso contrario, se mostrará en la página web el mensaje de error correspondiente. A la variable de sesión \$\_SESSION["tipo"] se le debe asignar el valor “modificacion” para que pueda trabajar correctamente cuando se invoque el archivo “guardar\_usuarios.php”, y "\$\_SESSION["nombre\_usuario"]" debe venir con su valor correspondiente para que pueda ser visualizado en el menú.

La variable \$\_SESSION["buscar\_usuario"] con valor “true” llamará a la función “cargar\_buscar()”, caso contrario llamará a la misma función y a continuación hará el “include” del archivo “datos\_usuario.php” con los datos resultantes de la búsqueda y con las opciones de modificación activas.

#### 4.2.28 Archivo “mostar\_hojavida.php”.

**Descripción.** Este archivo contiene el código PHP necesario para visualizar la hoja de vida de un estudiante cuantas veces sea necesario, una vez que la información del mismo haya sido ingresada. Si el llamado a esta página es realizado por un administrador, la hoja de vida será visualizada así ésta no se encuentre validada ni activa; al contrario si un usuario final de la red, invoca a este archivo, solo podrá visualizar las hojas de vida de usuarios con estado activo.

#### **Modo de invocación.**

La forma correcta de invocar a este archivo cuando el usuario que realiza llamada es un administrador o un usuario cualquiera de la web es mediante un vínculo que llame a esta página, pasando como parámetro el código del estudiante del que desea visualizar la hoja de vida dentro de una variable llamada “código\_estudiante”. Como lo indica el siguiente ejemplo:

```
<a href="mostrar_hojavida.php?codigo_estudiante=$codigo">
```

Cuando la llamada es realizada por el estudiante luego de modificar o ingresar los datos en su hoja de vida. Se debe realizar tan solo un “include” del archivo sin pasar ningún parámetro de referencia.

#### **Lista de archivos dependientes.**

- activar\_todos.php
- consulta\_pendientes.php
- desactivar\_todos.php
- ingreso.php
- listado\_final.php

#### **Descripción de funciones que incorpora.**

- mayusculas(\$palabra, \$matriz, \$mayusc): Esta función verifica si una cadena forma parte de una matriz determinada, y si es necesario convierte todos sus caracteres a mayúsculas, en los dos casos devuelve el valor coincidente en la matriz, con su respectiva transformación a mayúsculas y respetando las tildes.  
\$palabra: Cadena que va a ser verificada.  
\$matriz: Nombre de la matriz, en donde se buscará determinada cadena.

\$mayusc: Se debe enviar “si” en este parámetro para que transforme a mayúsculas el valor enviado en la variable “\$palabra”, caso contrario enviar “no”.

- `cargar_experiencia($sempre1, $lactem1, $lcargo1, $lmesin1, $lanoin1, $lmesfi1, $lanofi1, $ljefdi1, $ltel11, $ltel21, $lactca1, $nombre_mes)`: Esta función visualiza de forma ordenada en la hoja de vida la experiencia laboral del estudiante. Esta función debe encerrarse entre etiquetas `<table>` `</table>`, ya que dentro de su código dibuja `<td>` y `<tr>`.

\$sempre1: Nombre de la empresa.

\$lactem1: Actividad laboral de la empresa.

\$lcargo1: Cargo que desempeñó el estudiante en la empresa.

\$lmesin1: Mes de inicio de labor en esa empresa

\$lanoin1: Año de inicio de labor en esa empresa

\$lmesfi1: Mes de finalización de labores en esa empresa

\$lanofi1: Año de finalización de labores en esa empresa

\$ljefdi1: Nombre del jefe directo

\$ltel11: Número telefónico de la empresa

\$ltel21: Número telefónico de la empresa

\$lactca1: Actividades desempeñadas en dicha empresa.

\$nombre\_mes: Matriz que contiene los nombres de los meses.

- `cargar_disponibilidad ($area)`: Función que permite visualizar de forma ordenada en la hoja de vida la disponibilidad de horarios de trabajo del estudiante. Esta función debe encerrarse entre etiquetas `<table>` `</table>`, ya que dentro de su código dibuja `<td>` y `<tr>`.

### **Descripción de variables relevantes.**

El único requisito antes de llamar a este archivo es la definición de la variable “\$\_SESSION[“desde”]” a la cual se le debe asignar el lugar desde donde proviene el llamado, lo más importante es que si el llamado viene de la página “ingreso.php”, esta variable debe tener el valor “hojavida” para las validaciones correspondientes, si proviene de otra página el valor puede ser cualquiera, que serviría más que nada de referencia.

#### **4.2.29 Archivo “oracle\_conexion.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para abrir una conexión con la base de datos de Oracle de la Universidad del Azuay.

#### **Modo de invocación.**

Para utilizar este archivo es necesario tan solo realizar un “include” del mismo.

#### **Lista de archivos dependientes.**

- bolsa2.php

#### **Descripción de funciones que incorpora.**

Las funciones de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que contienen la clave de acceso a la base de datos Oracle.

#### **Descripción de variables relevantes.**

Las variables de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que dicho archivo contienen la clave de acceso a la base de datos Oracle.

#### **4.2.30 Archivo “oracle\_consulta.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para realizar una consulta de la base de datos Oracle de la Universidad del Azuay.

#### **Modo de invocación.**

Para utilizar este archivo es necesario primero guardar la sentencia “sql” en una variable llamada “\$consulta\_p”, luego hacer un “include” del mismo, y por último guardar en una variable el resultado de la función “oci\_fetch\_array(\$sql, OCI\_RETURN\_NULLS)”, en dicha variable es donde se guardarán el conjunto de registros respuesta. Todos estos pasos se deben realizar cada vez que se desee realizar una consulta Oracle.



**Lista de archivos dependientes.**

- bolsa2.php

**Descripción de funciones que incorpora.**

Las funciones de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que contienen la clave de acceso a la base de datos Oracle.

**Descripción de variables relevantes.**

Las variables de este archivo deben ser manipuladas tan solo por el administrador de la base de datos, ya que dicho archivo contienen la clave de acceso a la base de datos Oracle.

**4.2.31 Archivo “oracle\_cierra.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para cerrar una conexión existente con la base de datos de Oracle de la Universidad del Azuay.

**Modo de invocación.** Para utilizar este archivo es necesario tan solo realizar un “include” del mismo.

**Lista de archivos dependientes.**

- bolsa 2.php

**Descripción de funciones que incorpora.**

Las funciones de este archivo deben ser manipuladas tan solo por el administrador de la base de datos.

**Descripción de variables relevantes.**

Las variables de este archivo deben ser manipuladas tan solo por el administrador de la base de datos.

#### **4.2.32 Archivo “quitar\_validacion.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para visualizar un formulario de búsqueda en forma de tabla, que se divide en dos partes: “búsqueda por alumno (código y cédula)” y “búsqueda por rango de fechas (fecha inicial y fecha final)”. Luego de pulsar el botón “ver” el usuario podrá visualizar una lista de todas las hojas de vida que se encuentran validadas. Junto a cada registro se dibujarán “checkboxes” para que el administrador seleccione una o más hojas de vida que desee quitar la validación. La transacción será completada una vez que el administrador haya presionado el botón “quitar validación hojas de vida” el cual dirige a la página “desactivar.php”.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación a este archivo.

#### **Lista de archivos dependientes.**

- No existen archivos dependientes

#### **Descripción de funciones que incorpora.**

- `desactivar_casillas()`: Función JavaScript que permite activar o desactivar un conjunto de campos que pertenece a la opción de “mostrar según rango de fechas”.
- `eserror(lugar, tipo, datoerroneo, i)`: Función JavaScript que guarda en el vector “errores” los errores encontrados el momento de validar el rango de fechas antes de realizar la búsqueda correspondiente. Esta función devuelve el número de error encontrado hasta el momento.  
lugar: Indica el lugar en donde se ha encontrado el error.  
tipo: Indica el tipo de error que se ha producido.  
datoerroneo: Visualiza cual es el dato introducido incorrectamente.  
i: Devuelve el número de error correspondiente.
- `valida()`: Función JavaScript que valida el rango de fechas introducido, y devuelve “verdadero” cuando no existen errores, y “falso” cuando se hayan encontrado uno o más de ellos.

- `comparar_fechas(diain, mesin, anoin, diafi, mesfi, anofi)`: Función JavaScript llamada desde la función `valida()` que compara dos rangos de fechas. Esta función devuelve “verdadero” cuando existe un error en la comparación de las fechas, caso contrario devuelve “falso”.

diain: Día de la fecha de inicio.

mesin: Mes de la fecha de inicio.

anoin: Año de la fecha de inicio.

diafi: Día de la fecha fin.

mesfi: Mes de la fecha fin.

anofi: Año de la fecha fin.

- `activar_desactivar(elemento)`: Función JavaScript que permite seleccionar o quitar la selección por medio de checkbox de todos los registros de hojas de vida pendientes por activar listados en una página web. Esta función no devuelve ningún valor.

elemento: Este parámetro debe contener el nombre del formulario en el que se esté trabajando, es decir “document.nombreformulario”.

- `valida_fecha(d, m, a)`: Función JavaScript que comprueba la validez de una fecha. Devuelve “falso” si no ha encontrado error.

d: Día de la fecha que se va a validar.

m: Mes de la fecha que se va a validar.

a: Año de la fecha que se va a validar.

- `cargar_dia($dia)`: Esta función PHP permite que se cargue un combo con todos los días del mes, en caso de ser necesario se visualizará el día indicado. Se debe tomar en consideración que dicha función no dibuja el combo completo, tan solo pinta los “option” con sus valores respectivos; el “value” de cada “option” será el número que corresponde al día que se visualizará. Por lo tanto, antes de llamar a esta función se requiera escribir el código de apertura y finalización del combo, es decir “<select> </select>”. Esta función no devuelve ningún valor.

\$dia: Este parámetro debe contener el día que se desea visualizar por defecto en el combo, si no se requiere que se visualice ningún dato se debe enviar una cadena en blanco.

- `convertir_desde($dia, $mes, $ano)`: Esta función PHP permite restar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.  
\$dia: Indica el día de la fecha que se va a convertir.  
\$mes: Indica el mes de la fecha que se va a convertir.  
\$ano: Indica el año de la fecha que se va a convertir.
- `convertir_hasta($dia, $mes, $ano)`: Esta función PHP permite sumar un día a la fecha, tomando en consideración, el mes y año de la misma; para poder realizar la comparación correspondiente el momento de realizar una búsqueda entre fechas.  
\$dia: Indica el día de la fecha que se va a convertir.  
\$mes: Indica el mes de la fecha que se va a convertir.  
\$ano: Indica el año de la fecha que se va a convertir.
- `mostrar_listado($sql, $error)`. Esta función PHP permite visualizar el listado correspondiente basándose en una consulta “sql” realizada previamente, dependiendo del tipo de búsqueda que se va a realizar.  
\$sql: Parámetro que indica la consulta “sql” que se va a realizar.  
\$error: Parámetro que indica el mensaje que se visualizará en caso de error, o mensaje que se desee dar al usuario.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión `$_SESSION["tipo_usuario"]` debe llegar con el valor "adm"; caso contrario se mostrará en la página web el mensaje de error correspondiente. Las variables de sesión `$_SESSION["quitar"]` y `$_SESSION["accion"]` deben contener las cadenas “quitar” y “guardar” respectivamente para que pueda trabajar correctamente al llamar al archivo “desactivar.php”, además la variable `$_SESSION["nombre_usuario"]`, debe venir con su valor correspondiente, para que pueda ser visualizado en el menú.

#### 4.2.33 Archivo “redireccionusuario.php”.

**Descripción.** Este archivo es llamado por el archivo “ingresoalsistema.php” una vez que se haya realizado el “submit” del mismo. La función de esta página es de re-direccionar a la página principal según el usuario. Si los datos introducidos fueron del administrador del sistema, se re-direccionará a la página “administrador.php”, si los datos fueron de un usuario validador con estado activo se re-direccionará a la página “verificar\_estudiantes.php”, si no se mostrará el error pertinente. En caso de que no existan datos en la tabla “usuario” se creará automáticamente un usuario “rootadm” con contraseña “rootadm”.

#### **Modo de invocación.**

Para utilizar este archivo, es necesario invocarlo desde el método “action” de un formulario, es decir, algo similar al siguiente ejemplo:

```
<form name="formulario" action="redireccionausuario.php" method="post">
```

#### **Lista de archivos dependientes.**

- administrador.php
- verificar\_estudiantes.php

#### **Descripción de funciones que incorpora.**

- Este archivo no incorpora funciones.

#### **Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

#### 4.2.34 Archivo “scripts.js”.

**Descripción.** Contiene código JavaScript que servirá para la ejecución de funciones de varias páginas web de la bolsa de trabajo.

### **Modo de invocación.**

Para utilizar este archivo es necesario declarar el script al inicio del archivo de la siguiente manera:

```
<script language="JavaScript" src="scripts.js"></script>
```

### **Lista de archivos dependientes.**

- bolsa2.php
- buscar.php

### **Descripción de funciones que incorpora.**

- `setState()`: Esta función lista todas las provincias pertenecientes al país Ecuador, para esto llama a su vez a la función `syncSelect(selObj, value, adata, ciudad)`.
- `syncSelect(selObj, value, adata, ciudad)`: Esta función permite listar todas las ciudades pertenecientes a una cierta provincia seleccionada.  
selObj: Este parámetro indica el nombre del combo en donde se cargarán las ciudades. Este nombre tiene que ser descrito en su totalidad, es decir “document.nombreformulario.nombrecombo”.  
value: Este parámetro debe contener el nombre de la provincia, según la cual se listarán las ciudades pertinentes.  
adata: Este parámetro debe contener el nombre de la variable “mcity” que es un array que contiene un conjunto de provincias, códigos y ciudades.  
ciudad: Este parámetro debe contener el nombre de la ciudad que se desea quede seleccionada en el combo. En caso de que este dato sea indiferente, se debe enviar una cadena vacía.
- `setCountry()`: Esta función habilita en el formulario los campos provincia y ciudad en caso de que el país sea “Ecuador” es decir el código “1”; y deshabilita los mismos campos en caso contrario. Esta función no devuelve ningún valor.
- `cambiarCole()`: Esta función permite que se active una casilla de texto en caso de que el “value” seleccionado dentro de un combo sea igual a “Otro Colegio”, para que el nombre de otro colegio pueda ser ingresado. En caso contrario, se desactiva este campo.

- `cambiarArea()`: Esta función permite que se active una casilla de texto en caso de que el “value” seleccionado dentro de un combo sea igual a “Otra Area”, para que el nombre de esta área pueda ser ingresada. En caso contrario, desactiva este campo.
- `graduados(valor, combo)`: Esta función permite que se desactive un combo determinado en caso de que cierto valor sea igual a “Egresado” o “Graduado”. Caso contrario, este combo será activado.  
valor: Este parámetro debe contener cualquiera de los valores “Egresado”, “Graduado” o “Estudiante”, para que se pueda efectuar la comparación necesaria.  
combo: Este parámetro debe contener el nombre del combo que se debe desactivar en caso de que el valor de la variable “valor” sea igual a “Egresado” o “Graduado”.

**Descripción de variables relevantes.** La variable `mcity` es un array que contiene un conjunto de provincias, códigos y ciudades que permitirán junto con la función `syncSelect()` del mismo archivo, listar todas las ciudades de acuerdo a una cierta provincia seleccionada.

#### 4.2.35 Archivo “`usuarios_pendientes.php`”.

**Descripción.** Contiene el código PHP necesario para que el administrador pueda seleccionar de un combo el tipo de consulta de usuarios que desee realizar. Esta puede ser de “usuarios activos”, “usuarios inactivos” o “todos los usuarios”.

#### **Modo de invocación.**

No existen restricciones ni recomendaciones para el modo de invocación a este archivo.

#### **Lista de archivos dependientes.**

- No existen archivos dependientes.

### **Descripción de funciones que incorpora.**

- `mostrar_listado($sql, $error)`. Esta función PHP permite visualizar el listado correspondiente basándose en una consulta “sql” realizada previamente, dependiendo del tipo de búsqueda que se va a realizar.

\$sql: Parámetro que indica la consulta “sql” que se va a realizar.

\$error: Parámetro que indica el mensaje que se visualizará en caso de error.

### **Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión “\$\_SESSION[“tipo\_usuario”]” debe llegar con el valor “adm”; caso contrario, se mostrará en la página web el mensaje de error correspondiente. La variable “\$\_SESSION[“nombre\_usuario”]” debe tener su valor correspondiente para que pueda ser visualizada en el menú.

#### **4.2.36 Archivo “val.js”.**

**Descripción.** Archivo con el código JavaScript necesario para validar los campos del formulario del archivo “bolsa2.php”.

#### **Modo de invocación.**

Para utilizar este archivo es necesario declarar el script al inicio del archivo de la siguiente manera:

```
<script language="JavaScript" src="val.js"></script>
```

Y en el botón que realice el “submit” del formulario llamar a la función validadora “validar(document.formulario)”.

#### **Lista de archivos dependientes.**

- Este archivo no contiene archivos dependientes.

### **Descripción de funciones que incorpora.**

Las funciones de este archivo, son funciones JavaScript.



- `validar(formulario)`: Esta función permite la validación de datos ingresados en el formulario utilizado en el archivo “bolsa2.php”.  
formulario: Es nombre del formulario que se desea validar. Al llamar a esta función se debe enviar como parámetro “document.nombreFormulario”.
- `eserror(lugar, tipo, datoerroneo, i)`: Esta función se ejecuta cada vez que se localiza un error. Dicha función almacena en el vector “errores[i]” separados por un “@” los siguientes valores: “lugar, tipo y datoerroneo” de la siguiente manera:  
`errores[i]= lugar+"@"+tipo+"@"+datoerroneo;`  
Devuelve como resultado el número del contador de errores correspondiente.  
lugar: Describe en qué parte del documento se ha encontrado un error.  
tipo: Describe el tipo de error encontrado.  
datoerroneo: Describe cuál es el dato erróneo.  
i: Indica el número en el que se encuentra el contador de errores.
- `valida_fecha(d, m, a)`: Esta función verifica la validez de una fecha. Devuelve “verdadero” si la fecha está correcta y “falso” en caso contrario.  
d: Valor numérico que describe el día de la fecha a validar.  
m: Valor numérico que describe el mes de la fecha a validar.  
a: Valor numérico que describe el año de la fecha a validar.
- `trim(cadena)`: Esta función elimina los espacios en blanco existentes en el inicio y fin de una cadena.  
cadena: Valor alfanumérico de donde se desee eliminar los espacios.
- `comparar_fechas(diain, mesin, anoin, diafi, mesfi, anofi)`: Función que compara entre dos fechas. Si la primera fecha (fecha de inicio) es mayor a la segunda (fecha fin), la función devolverá “falso”, caso contrario “verdadero”.  
diain: Valor numérico que indica el día de la fecha de inicio.  
mesin: Valor numérico que indica el mes de la fecha de inicio.  
anoin: Valor numérico que indica el año de la fecha de inicio.  
diafi: Valor numérico que indica el día de la fecha fin.  
mesfi: Valor numérico que indica el mes de la fecha fin.  
anofi: Valor numérico que indica el año de la fecha fin.

**Descripción de variables relevantes.** Las variables “contadorerrores” y “errores” son indispensables para la visualización de los errores.

#### 4.2.37 Archivo “val\_usuarios.js”.

**Descripción.** Archivo con el código JavaScript necesario para validar los campos del formulario de los archivos “ingreso\_usuarios.php” y “modificacion\_usuarios”.

#### **Modo de invocación.**

Para utilizar este archivo es necesario declarar el script al inicio del archivo de la siguiente manera:

```
<script language="JavaScript" src="val_usuarios.js"></script>
```

Y en el botón que realice el “submit” del formulario llamar a la función validadora.

#### **Lista de archivos dependientes.**

- ingreso\_usuarios.php
- modificacion\_usuario.php

#### **Descripción de funciones que incorpora.**

Las funciones de este archivo, son funciones JavaScript.

- validar(formulario): Esta función permite la validación de datos ingresados del formulario utilizado en el archivo “ingreso\_usuarios.php” o “modificacion\_usuarios.php”.

formulario: Es nombre del formulario que se desea validar. Al llamar a esta función se debe enviar como parámetro “document.nombreFormulario”.

- eserror(lugar, tipo, datoerroneo, i): Cada vez que se localice un error en la validación, se ejecutará esta función; la cual almacena en el vector “errores[i]” separados por un “@” los siguientes valores: lugar, tipo, datoerroneo; de la siguiente manera:

```
errores[i]= lugar+"@"+tipo+"@"+datoerroneo
```

Devuelve como resultado el número del contador de errores correspondiente.

lugar: Describe en qué parte del documento se ha encontrado un error

tipo: Describe el tipo de error encontrado.

datoerroneo: Describe cuál es el dato erróneo.

i: Indica el número en el que se encuentra el contador de errores.

- trim(cadena): Esta función elimina los espacios en blanco existentes en el inicio y fin de una cadena.

cadena: Valor alfanumérico de donde se desea eliminar los espacios.

- habilitar\_contrasena(): Esta función permite que en “modificación\_usuario.php” si se selecciona la opción marcada en un check de habilitar contraseña, se habiliten los campos de “contraseña” y “confirmar contraseña”.

**Descripción de variables relevantes.** Las variables “contadorerrors” y “errores” son indispensables para la visualización de los errores.

#### 4.2.38 Archivo “verblob.php”.

**Descripción.** Este archivo contiene el código PHP necesario para visualizar la fotografía existente de un estudiante, desde la base de datos MySql.

#### **Modo de invocación.**

Para utilizar este archivo es necesario llamarlo desde una etiqueta HTML “<img/>” pasando como parámetro el código del estudiante dentro de la variable “varcodigo”, de manera similar a la siguiente:

```

```

En donde “\$cod” contiene el código del estudiante del que se va a visualizar la fotografía. Y también es necesario que dicha etiqueta se encuentre dentro de una etiqueta de formulario “<form> </form>”, con el atributo “enctype” igual a “multipart/form-data”; de manera similar a la siguiente:

```
<form name = "dp" method="post" action="ingreso.php" enctype="multipart/form-data">
```

**Lista de archivos dependientes.**

- bolsa2.php
- mostrar\_hojavida.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

- Este archivo no contiene variables relevantes.

**4.2.39 Archivo “verificar\_estudiantes.php”.**

**Descripción.** Este archivo contiene el código PHP necesario para permitir a los usuarios del sistema, validar o activar hojas de vida, según el tipo de usuario que sea. Los administradores podrán validar o activar, mientras que los usuarios regulares tendrán acceso solo a la activación. Para todos los casos se dibujará en pantalla un formulario en donde se ingresará el número de cédula del estudiante a validar o activar.

**Modo de invocación.** Si quien realiza la llamada es un usuario regular, la forma de llamar a este archivo no tiene restricciones ni recomendaciones, pero si quien realiza la llamada es un administrador, es necesario enviar como parámetro en la variable “accion” los valores “activación” o “validación” según sea el caso.

**Lista de archivos dependientes.**

- bolsa2.php

**Descripción de funciones que incorpora.**

- Este archivo no contiene funciones.

**Descripción de variables relevantes.**

Para poder utilizar este archivo, el valor de la variable de sesión "\$\_SESSION["tipo\_usuario"]" debe llegar con el valor "adm" o "usr"; caso contrario, se mostrará en la página web el mensaje de error correspondiente. La variable "\$\_SESSION["nombre\_usuario"]" debe tener su valor correspondiente para que pueda ser visualizada en el menú. La variable \$\_SESSION["tipo"], indicará de aquí en adelante el tipo de acción que se realizará, es decir "activación" o "validación", este valor debe ser asignado desde la variable "\$accion" que fue la que se envió como parámetro al hacer llamado a este archivo.

## CAPÍTULO 5

### MANUAL DEL USUARIO

#### 5.1 Introducción.

El presente manual está dirigido para dos tipos de usuario; los usuarios validadores y usuarios administradores los mismos que se definen a continuación:

**Usuario Validador:** Persona encargada de verificar que la información ingresada en la hoja de vida sea correcta comparándola con la documentación entregada.

**Usuario Administrador:** Persona responsable del manejo del sistema de la bolsa de trabajo; entre sus principales funciones está la creación, modificación y eliminación de usuarios, además de validación y activación de hojas de vida, así como las respectivas consultas de las transacciones realizadas en el sistema en general.

El objetivo que se persigue con la aplicación del presente manual es dar a conocer a los usuarios finales, las características y las formas de funcionamiento del sitio web de la bolsa de trabajo de la Universidad del Azuay.

#### 5.2 Ingreso al Sistema.

Para ingresar al sitio web de la bolsa de trabajo de la Universidad del Azuay, se debe digitar en el navegador la siguiente dirección:

<http://www.uazuay.edu.ec/servicios/bolsa/admin/home.php>



The screenshot shows a web form for system login. At the top, there is a dark blue header with the text "INGRESO AL SISTEMA" in white. Below this is a sub-header "Ingrese sus Datos" in a lighter blue bar. The main form area has a light gray background and contains two input fields: "Usuario:" with the value "usuario1" and "Clave:" with "\*\*\*\*\*". Below the fields is a button labeled "Ingresar".

Gráfico 5.2 Ingreso al sistema.

Para ingresar al sistema, el usuario debe ingresar su código de usuario y clave; los cuales deben haber sido asignados previamente por el usuario administrador.

### 5.3 Manejo del sistema por un usuario validador.

En caso de que los datos introducidos pertenezcan a un usuario validador, se presentará la siguiente pantalla:

The screenshot shows a web interface for 'Bolsa de Trabajo'. At the top, the user's name 'MONICA PATRICA DURAN CARRASCO' is displayed. Below this is a dark blue navigation bar with 'Home' and 'Cerrar Sesión' links. The main content area is titled 'Validación de Hojas de Vida'. It contains a form with a text input field labeled 'Cédula del estudiante a validar:' and a 'Validar' button below it. Red circles with numbers 1 through 5 are overlaid on the image, pointing to the user name, the 'Home' link, the 'Cerrar Sesión' link, the text input field, and the 'Validar' button, respectively.

Gráfico 5.3 Validación por un usuario validador.

En donde:

1. Nombre del usuario validador que se visualiza automáticamente según los datos ingresados por el usuario. Todas las acciones que realice un determinado usuario quedarán registradas con su nombre.
2. Vínculo que re-direcciona al usuario a la página de ingreso al sistema.
3. Vínculo que re-direcciona al usuario a la página de ingreso al sistema, y además permite al usuario desconectarse del sistema. Cabe mencionar que el usuario siempre deberá cerrar sesión una vez que haya terminado sus actividades, para que ninguna persona que haga uso del mismo navegador pueda realizar acciones malintencionadas en las cuales quedará registrado dicho usuario como único responsable.
4. Ingrese el número de cédula del estudiante cuya hoja de vida va a ser validada.
5. Presione el botón Validar para visualizar y modificar la hoja de vida del alumno que desee validar.

En caso de que la cédula ingresada sea incorrecta o no exista en la base de datos de la bolsa de trabajo se presentará en pantalla el mensaje “Error, número de cédula no encontrado”. De no existir hojas de vida pendientes por validar, se visualizará el mensaje “No existen carpetas por validar”. Si el número de cédula hace referencia a una hoja de vida que ya ha sido validada, se presentará el mensaje “Esta carpeta ya ha sido validada”, y por último si el número de cédula hace referencia a una hoja de vida que ya ha sido activada por el usuario administrador, se presentará el mensaje “Esta carpeta ya ha sido activada”.

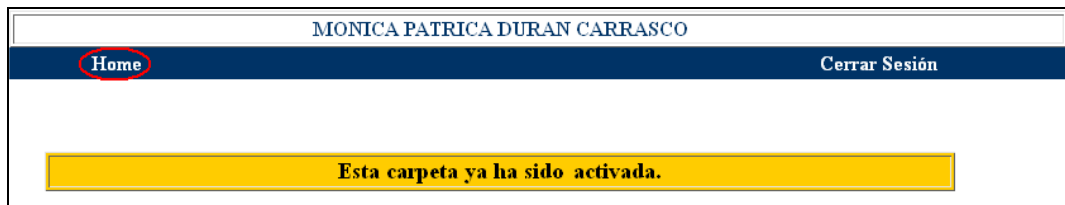


Gráfico 5.3.1 Mensaje de error en validación por usuario validador.

En estos casos de error, el vínculo “Home” re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad.

Si el número de cédula del estudiante es correcto, se visualiza la siguiente pantalla:

Gráfico 5.3.2 Inicio formulario validación de hojas de vida.



En donde:

1. Vínculo que re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad.
2. Vínculos que permiten acceder de forma inmediata a una sección de información determinada.

Horario nocturno :  Si  No

¿Dispuesto(a) a trabajar en otra ciudad?:  Si  No

Viajar:  Dentro de la región  Dentro del país  Fuera del país

¿Si la empresa lo requiere, puede disponer de su vehículo para el trabajo?:  Si  No

¿Si la empresa lo requiere, puede disponer de computador personal/agenda electrónica?:  Si  No

**Validación de Hoja de Vida:**

Validado  No Validado

Usuario: MONICA PATRICA DURAN CARRASCO

Antes de presionar el botón Grabar por favor verifique que la información ingresada esté correcta.

Guardar (5) Limpiar (4)

Gráfico 5.3.3 Final formulario validación de hojas de vida.

3. Permite cambiar el estado de una hoja de vida.
4. Permite borrar todos los cambios realizados sobre la hoja de vida, y dejar como inicialmente se visualizó.
5. Guarda los cambios realizados.

Al presionar el botón guardar, se visualizará una nueva pantalla con un listado de errores en caso de existir, en la misma se especifica la sección en donde se encuentra el error, el tipo de error, y el dato erróneo.

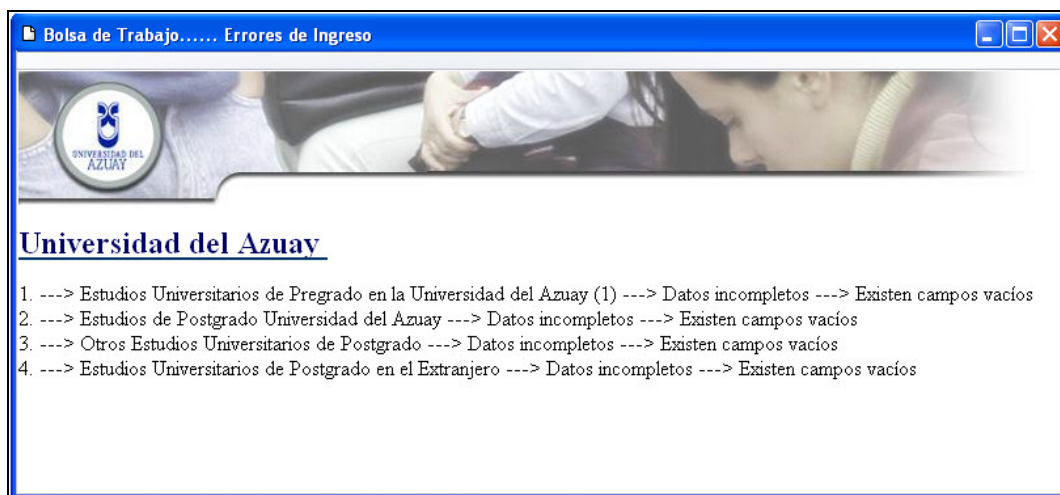


Gráfico 5.3.4 Pantalla de errores.

En caso de no existir error, se guardarán los datos modificados y se presentará la siguiente pantalla.

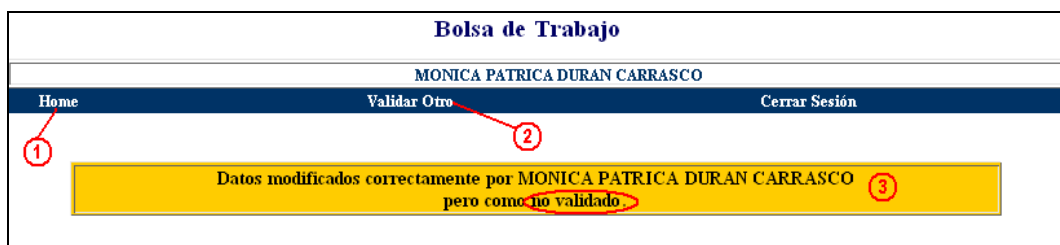


Gráfico 5.3.5 Mensaje de confirmación de datos guardados.

Siendo:

1. Vínculo que re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad, para realizar una nueva validación.
2. Vínculo que re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad, para realizar una nueva validación.
3. Mensaje que indica el nombre del usuario responsable de la validación, así como el estado con el que se guardó la hoja de vida que puede ser “validado” o “no validado”.

#### 5.4 Manejo del sistema por el usuario administrador.

En caso de que los datos introducidos pertenezcan al usuario administrador, se presentará la siguiente pantalla:

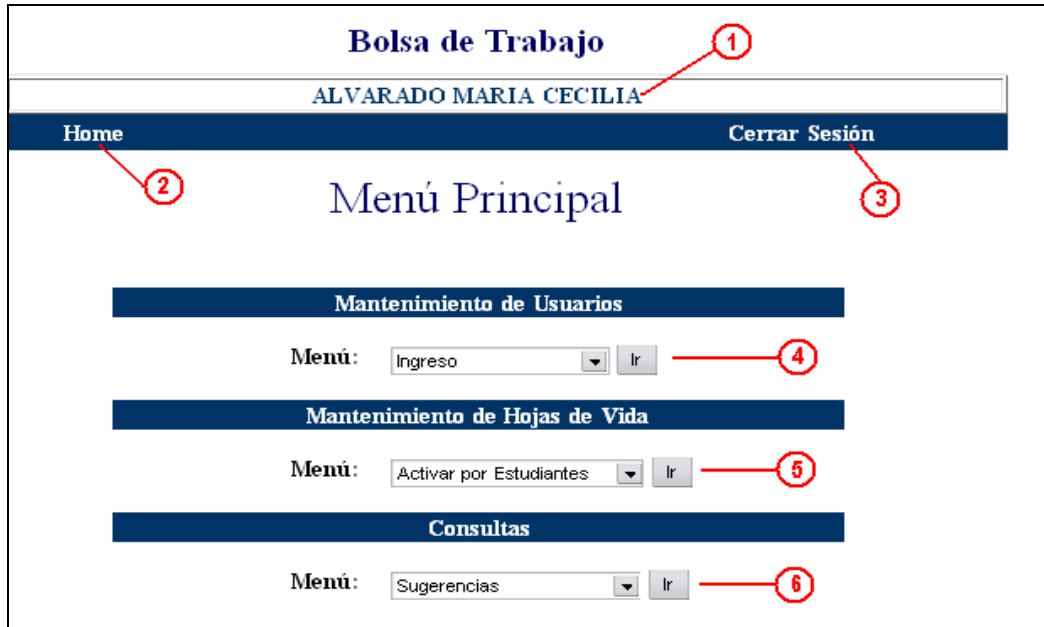


Gráfico 5.4 Página principal para del usuario administrador.

En donde:

1. Nombre del usuario administrador que se visualiza automáticamente según los datos ingresados por el usuario. Todas las acciones que realice este usuario quedarán registradas con su nombre.
2. Vínculo que re-direcciona al usuario a la página de ingreso al sistema.
3. Vínculo que re-direcciona al usuario a la página de ingreso al sistema, y además permite al usuario desconectarse del sistema. Cabe mencionar que el usuario siempre deberá cerrar sesión una vez que haya terminado sus actividades, para que ninguna persona que haga uso del mismo navegador pueda realizar acciones malintencionadas en las cuales quedará registrado dicho usuario como único responsable.
4. Menú que permite realizar mantenimiento de usuarios.
5. Menú que permite realizar mantenimiento de hojas de vida.
6. Menú que permite realizar consultas.

### 5.4.1 Mantenimiento de usuarios.

En esta sección el usuario administrador puede realizar ingresos, modificaciones, eliminaciones y consultas de usuarios.



Gráfico 5.4.1 Mantenimiento de usuarios.

1. Seleccione la acción que desee realizar y presione el botón “Ir” para visualizar la pantalla correspondiente.

### Ingreso de usuarios.

The screenshot shows a web application interface titled 'Bolsa de Trabajo' and 'ALVARADO MARIA CECILIA'. At the top, there are navigation links for 'Home' and 'Cerrar Sesión'. The main heading is 'Ingreso de Usuario'. Below this, there is a form with the following fields: 'Usuario:' (usuario1), 'Nombres:' (LISSETH MARIA), 'Apellidos:' (CURE LOPEZ), 'Estado:' (radio buttons for 'Activado' and 'Desactivado'), 'Teléfono 1:' (2888888), 'Teléfono 2:' (098888987), 'Clave:' (\*\*\*\*\*), and 'Confirme clave:' (\*\*\*\*\*). At the bottom, there are two buttons: 'Guardar' and 'Limpiar'. Red circles with numbers 1 through 8 point to various elements: 1 to 'Home', 2 to the 'Usuario' input field, 3 to the 'Nombres' input field, 4 to the 'Estado' radio buttons, 5 to the 'Teléfono 1' input field, 6 to the 'Clave' input field, 7 to the 'Limpiar' button, and 8 to the 'Guardar' button.

Gráfico 5.4.1.1 Ingreso de usuarios.

En donde:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.
2. Ingrese el código de usuario con el que el usuario validador accederá al sistema. El código debe ser mínimo de seis y máximo de ocho caracteres.
3. Ingrese los nombres y apellidos del usuario validador que está agregando. Los campos deben tener un máximo de treinta caracteres cada uno.
4. Seleccione el estado del usuario. Un usuario con estado inactivo, no podrá ingresar al sistema.
5. Ingrese los números telefónicos necesarios para contactar al usuario. Los campos deben tener un mínimo de seis y un máximo de diez caracteres.
6. Digite la clave con la que el usuario validador ingresará al sistema, y a continuación confirme la misma en el campo siguiente. La clave debe tener por lo menos seis caracteres y un máximo de diez.
7. Permite borrar la información ingresada en todos los campos.
8. Guarda los datos del nuevo usuario validador.

Al presionar el botón guardar, se visualizará una nueva pantalla con un listado de errores en caso de existir, en la misma se especifica la sección en donde se encuentra el error, el tipo de error, y el dato erróneo.

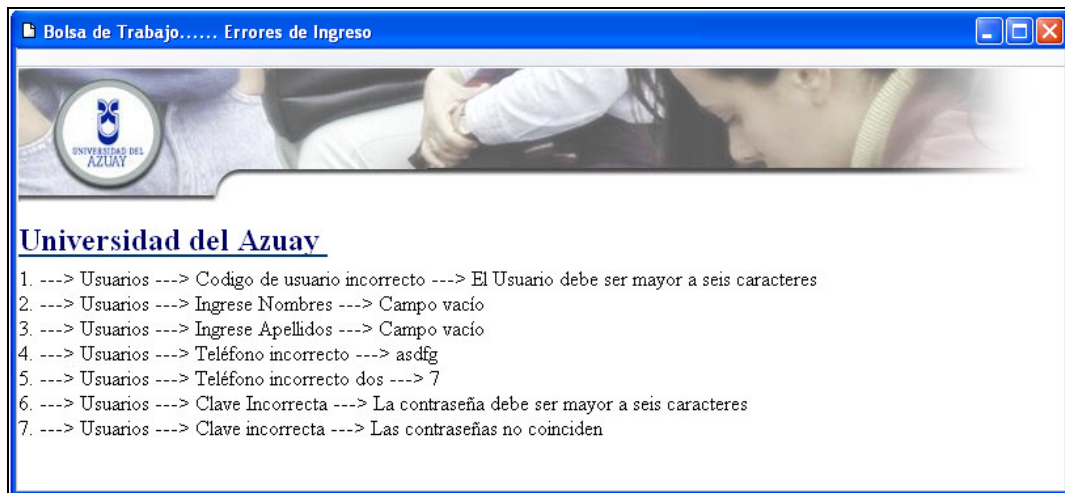


Gráfico 5.4.1.1.2 Pantalla de errores

En caso de no existir errores de validación en el ingreso de datos se presentará la siguiente pantalla:

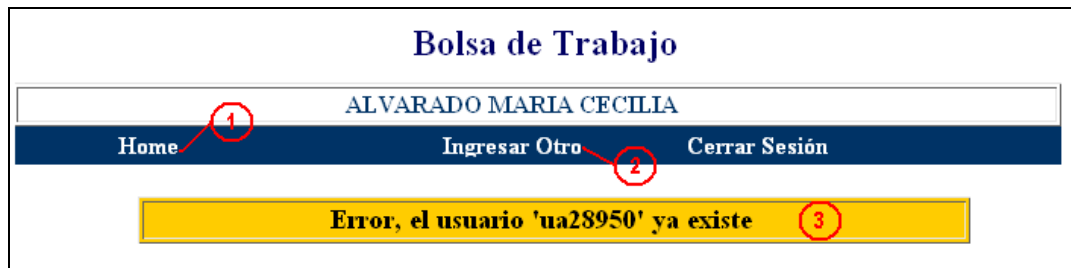


Gráfico 5.4.1.1.3 Mensajes de error en ingreso de usuarios.

Siendo:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.
2. Vínculo que re-direcciona al usuario a la página correspondiente para ingresar otro usuario validador.
3. Mensaje de visualización que indica que el usuario ha sido ingresado correctamente, o que el código del usuario que se desea ingresar ya existe, en este caso, los datos no serán guardados.

### Modificación de Usuarios.

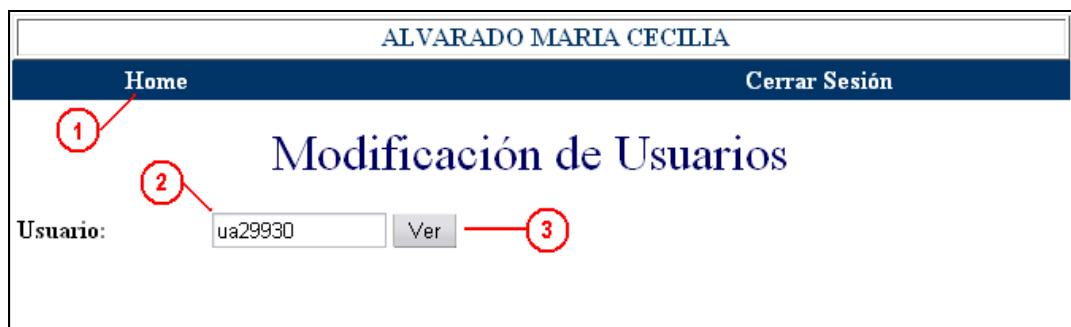


Gráfico 5.4.1.2 Modificación de usuarios.

Siendo:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.

2. Ingrese el código del usuario validador que desee modificar.
3. Presione el botón “Ver” para visualizar los datos del usuario validador antes de modificarlos.

Una vez presionado el botón “Ver” se visualizará la siguiente pantalla en donde el usuario administrador podrá realizar los cambios pertinentes.

**Bolsa de Trabajo**

ALVARADO MARIA CECILIA

Home Cerrar Sesión

### Modificación de Usuarios

Usuario:  Ver **1**

---

Usuario: ua29930 **2**

Nombres: TORRES GUZMAN

Apellidos: DIANA PAOLA

Estado:  Activado  Desactivado **3**

Teléfono 1: 2857395

Teléfono 2:

¿Modificar contraseña?  Sí  No **4**

Clave:

Confirme clave:

**6** Guardar **5** Limpiar

Gráfico 5.4.1.2.1 Presentación de la información del usuario a modificar.

En donde:

1. Ingrese un nuevo código de usuario a modificar.
2. El código de usuario no se puede modificar.
3. Seleccione el estado del usuario. Un usuario con estado inactivo, no podrá ingresar al sistema. Seleccione “Si” si desea modificar la contraseña del usuario validador, e inmediatamente se activarán los campos “Clave” y “Confirme clave” para que sean ingresados.
4. Permite borrar todos los cambios realizados sobre los datos del usuario validador, y dejar como inicialmente se visualizó.
5. Guarda los datos modificados del usuario validador.

Al presionar el botón guardar, se visualizará una nueva pantalla con un listado de errores en caso de existir, en la misma se especifica la sección en donde se encuentra el error, el tipo de error, y el dato erróneo. En caso de que no existan errores de validación de datos, se mostrará el siguiente mensaje:

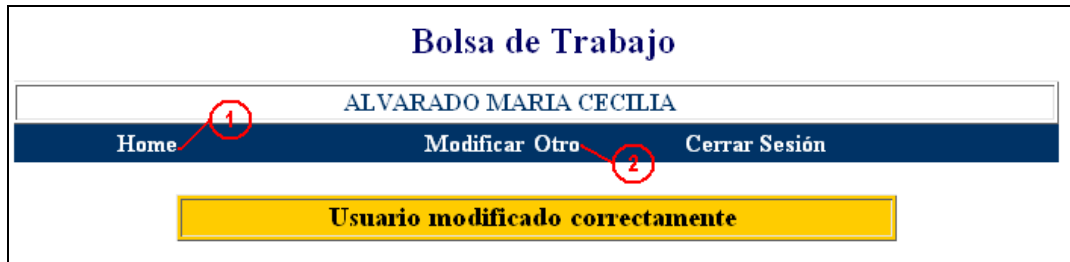


Gráfico 5.4.1.2.2 Presentación de mensaje en modificación de usuarios.

Siendo:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.
2. Vínculo que re-direcciona al usuario a la página correspondiente para ingresar el código de otro usuario validador para modificarlo.

### Eliminación de Usuarios.

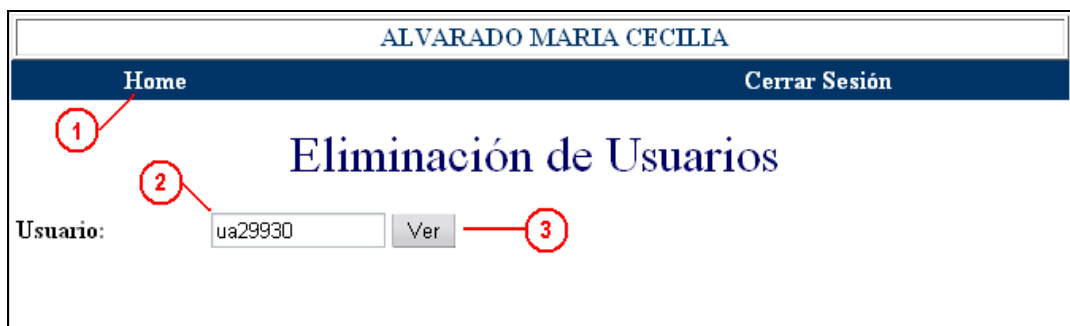


Gráfico 5.4.1.3 Eliminación de usuarios.

Siendo:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.



2. Ingrese el código del usuario validador que desee eliminar.
3. Presione el botón “Ver” para visualizar los datos del usuario validador antes de eliminarlo.

Una vez presionado el botón “Ver” se visualizará la siguiente pantalla en donde el usuario administrador puede visualizar los datos, antes de eliminar al usuario.

Gráfico 5.4.1.3.1 Presentación de información del usuario a eliminar.

En donde:

1. Ingrese el nuevo código de usuario a eliminar.
2. Si está seguro de eliminar el usuario, presione el botón “Eliminar”.

Al presionar el botón eliminar, se visualizará una nueva pantalla en donde se indicará que la operación se realizó con éxito.

Gráfico 5.4.1.3.2 Presentación de mensajes en caso de eliminación.

En donde:

1. Vínculo que re-direcciona al usuario a la página principal del administrador del sistema.
2. Vínculo que re-direcciona al usuario a la página correspondiente para ingresar el código de otro usuario validador para eliminarlo.

### Consulta de usuarios.

#	Código	Nombres	Telefono 1	Telefono 2	Tipo	Estado
1	rootadm	ALVARADO MARIA CECILIA	1234567890		adm	Activo
2	usuario1	DE TAL FULANO	1234567890	34567890	usr	Activo
3	ua29930	DIANA PAOLA TORRES GUZMAN	2857395		usr	Activo
4	ua28950	MONICA PATRICA DURAN CARRASCO	2881629		usr	Activo

Gráfico 5.4.1.4 Presentación de información en consulta de usuarios.

En donde:

1. Seleccione “Activos” para visualizar los usuarios activos, “Inactivos” para visualizar aquellos usuarios existentes pero que no tienen acceso al sistema, y “Todos” para visualizar todos los usuarios.
2. Presione el botón “Consultar” para visualizar el listado correspondiente.

### 5.4.2 Mantenimiento de hojas de vida.

En esta sección el usuario administrador puede activar, desactiva, validar y quitar validaciones de las hojas de vida.

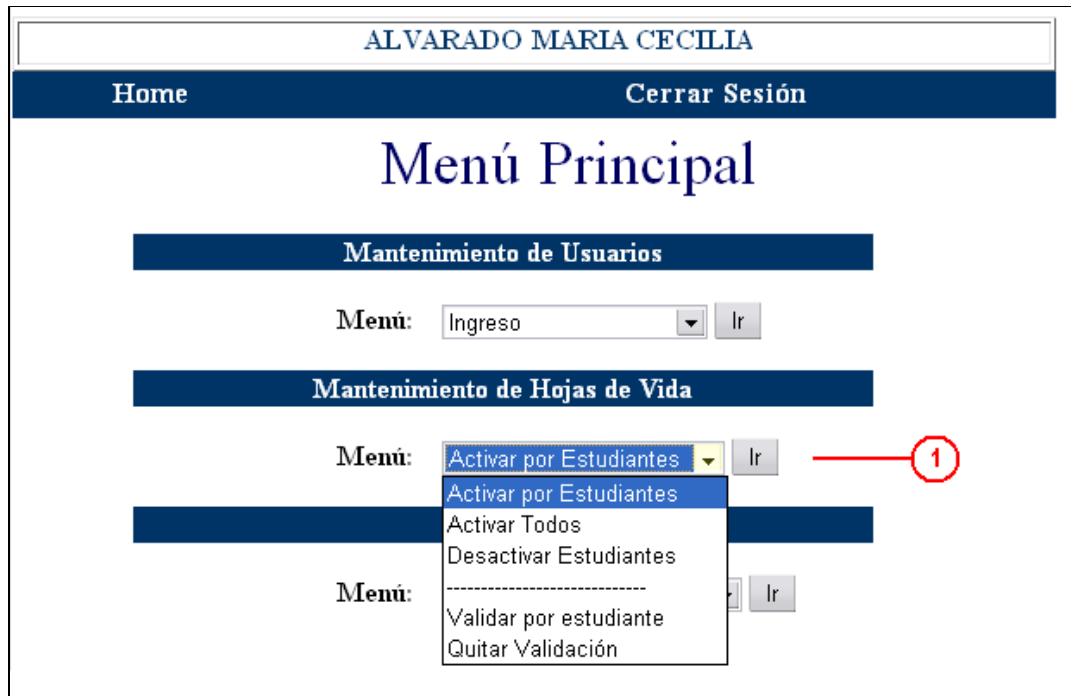


Gráfico 5.4.2 Mantenimiento de hojas de vida

1. Seleccione la acción que desee realizar y presione el botón “Ir” para visualizar la pantalla correspondiente.

### Activar por estudiantes.

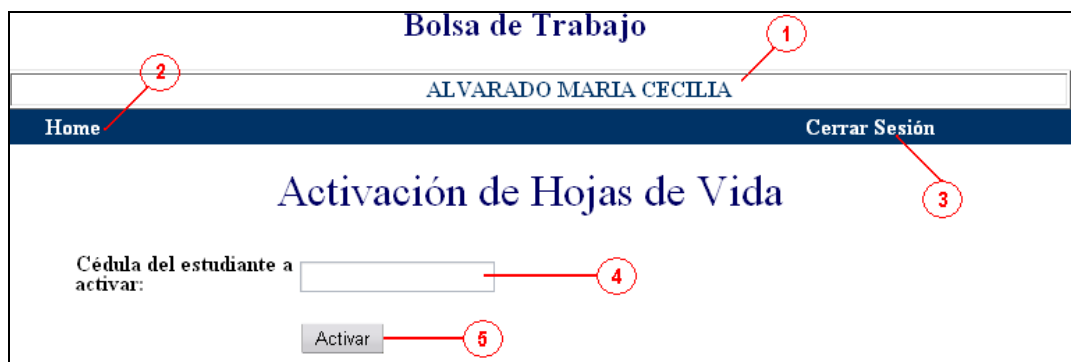


Gráfico 5.4.2.1 Activar por estudiante

En donde:

1. Nombre del usuario administrador que se visualiza automáticamente según los datos ingresados por el usuario. Todas las acciones que realice el usuario quedarán registradas con su nombre.
2. Vínculo que re-direcciona al usuario a la página principal del usuario administrador.
3. Vínculo que re-direcciona al usuario a la página de inicio, y además permite al usuario desconectarse del sistema. Cabe mencionar que el usuario siempre deberá cerrar sesión una vez que haya terminado sus actividades, para que ninguna persona que haga uso del mismo navegador pueda realizar acciones malintencionadas en las cuales quedará registrado dicho usuario como único responsable.
4. Ingrese el número de cédula del estudiante cuya hoja de vida va a ser activada, es decir visualizada en las búsquedas por internet.
5. Presione el botón Activar para visualizar y modificar la hoja de vida del alumno que desee activar.

En caso de que la cédula pertenezca a un estudiante cuya carpeta todavía no ha sido validada, se visualizará el mensaje “Esta carpeta todavía no ha sido validada”. Si la cédula de identidad no existe o está incorrecta se visualizará “Error, cédula incorrecta”. Si no existen carpetas validadas pendientes de activar se visualizará “No existen carpetas por activar”.

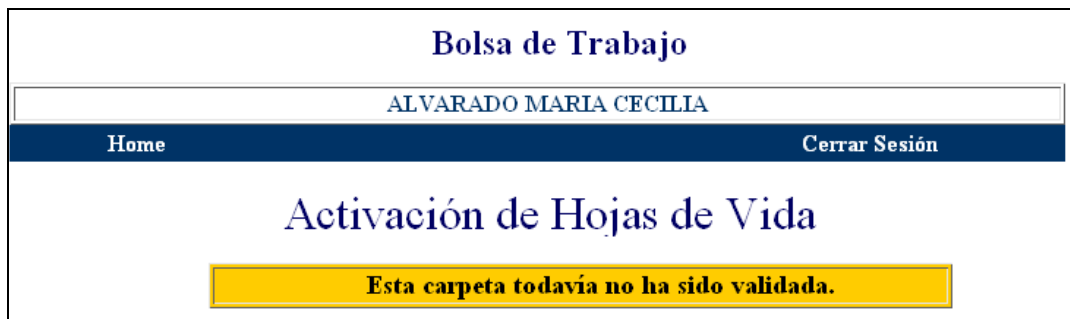


Gráfico 5.4.2.1.2 Presentación de mensaje en caso de activación de hojas de vida

Si la cédula del estudiante es correcta, se visualizará la siguiente página:

ALVARADO MARIA CECILIA

Home Cerrar Sesión

### Activación de Hojas de Vida

Validado por: ALVARADO MARIA CECILIA (Código: rootadm)

Fecha de validación: 2007-10-16

Bienvenido(a) a la Bolsa de Trabajo de la Universidad del Azuay, este es un servicio que ofrecemos a nuestros(as) estudiantes y egresados(as) para facilitar su contacto con el medio laboral.

[Datos Personales](#) [Estudios Secundarios](#) [Estudios Superiores de Pregrado](#)

[Estudios de Postgrado](#) [Viajes](#) [Formación Adicional Relevante](#)

[Conocimiento de Idiomas](#) [Conocimientos Informáticos](#) [Otras Destrezas](#)

[Experiencia Laboral](#) [Disponibilidad](#) [Otra Información que considere Importante](#)

#### Datos Personales

Fotografía: 

Elegir...

\* Género:  Masculino  Femenino

\* Primer Nombre:

\* Segundo Nombre:

Gráfico 5.4.2.1.3 Inicio formulario para la activación de hojas de vida.

En donde:

1. Nombre del usuario validador responsable de la validación de dicha hoja de vida.
2. Fecha y hora de última modificación y validación de la hoja de vida.
3. Vínculos que permiten acceder de forma inmediata a una sección de información determinada.

**Horario de trabajo:**  Tiempo completo  Medio tiempo  Por horas  Por proyecto  
**Turnos rotativos:**  Si  No  
**Horario nocturno:**  Si  No  
**¿Dispuesto(a) a trabajar en otra ciudad?:**  Si  No  
**Viajar:**  Dentro de la región  Dentro del país  Fuera del país  
**¿Si la empresa lo requiere, puede disponer de su vehículo para el trabajo?:**  Si  No  
**¿Si la empresa lo requiere, puede disponer de computador personal/agenda electrónica?:**  Si  No

**Activación de Hoja de vida:**

Estado Activo  Estado Inactivo 4

**Usuario:** ALVARADO MARIA CECILIA

Antes de presionar el botón Grabar por favor verifique que la información ingresada esté correcta.


6   5

Gráfico 5.4.2.1.4 Final del formulario para la activación de hojas de vida.

4. Permite cambiar el estado de una hoja de vida que ya ha sido validada.
5. Permite borrar todos los cambios realizados sobre la hoja de vida, y dejar como inicialmente se visualizó.
6. Guarda los cambios realizados.

Al presionar el botón guardar, se visualizará una nueva pantalla con un listado de errores en caso de existir, en la misma se especifica la sección en donde se encuentra el error, el tipo de error, y el dato erróneo.

Bolsa de Trabajo..... Errores de Ingreso



**Universidad del Azuay**

1. ---> Estudios Universitarios de Pregrado en la Universidad del Azuay (1) ---> Datos incompletos ---> Existen campos vacíos
2. ---> Estudios de Postgrado Universidad del Azuay ---> Datos incompletos ---> Existen campos vacíos
3. ---> Otros Estudios Universitarios de Postgrado ---> Datos incompletos ---> Existen campos vacíos
4. ---> Estudios Universitarios de Postgrado en el Extranjero ---> Datos incompletos ---> Existen campos vacíos

Gráfico 5.4.2.1.5 Presentación de listado de errores.

En caso de no existir error, se guardarán los datos modificados y se presentará la siguiente pantalla.

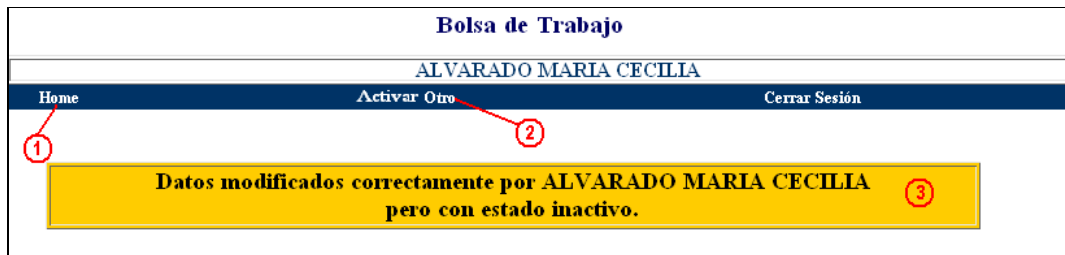


Gráfico 5.4.2.1.6 Presentación de mensaje para activación de hojas de vida

Siendo:

1. Vínculo que re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad, para realizar una nueva activación.
2. Vínculo que re-direcciona a la página que permite el ingreso de un nuevo número de cédula de identidad, para realizar una nueva activación.
3. Mensaje que indica el nombre del usuario responsable de la activación, así como el estado con el que se guardó la hoja de vida que puede ser “con estado activo” o “con estado inactivo”.

**Activar todos.**



Gráfico 5.4.2.2 Listado de hojas de vida para activar.

Siendo:

1. Indica el nombre del usuario validador.
2. Si desea visualizar la hoja de vida del estudiante, de clic sobre el nombre.
3. Presione este botón para seleccionar toda la lista de alumnos, y vuelva a presionar si desea quitar la selección de todos.
4. Presione el botón “Activar” para activar todas las hojas de vida seleccionadas.

Luego de presionar el botón “Activar”, se visualizará la siguiente pantalla.

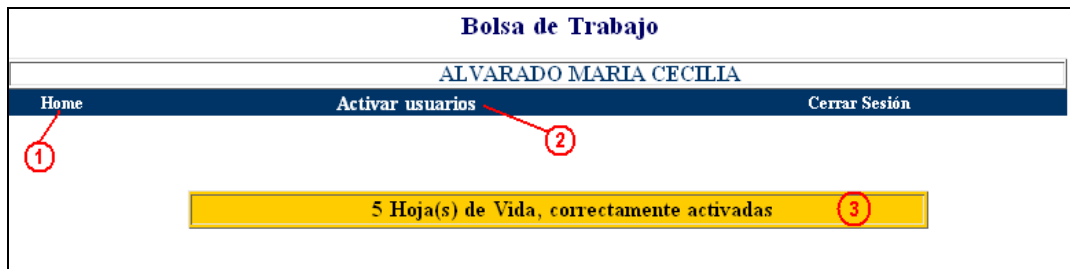


Gráfico 5.4.2.2.1 Presentación de mensaje en caso de activación.

En donde:

1. Vínculo que re-direcciona a la página principal del administrador.
  2. Vínculo que re-direcciona a la página de activar varios usuarios.
  3. Mensaje que visualiza el número de hojas que se han activado correctamente.
- Si en la página anterior no seleccionó ninguna hoja de vida para activar, se visualizará el mensaje “Error, no ha seleccionado ninguna hoja de vida para activar”.



## Desactivar todos.

**Desactivación de Hojas de Vida**

<input checked="" type="radio"/> <b>Por alumno:</b>	Código alumno: <input type="text"/>
	Cédula: <input type="text"/>
<input type="radio"/> <b>Por rango de fechas:</b>	Desde: <input type="text" value="16"/> <input type="text" value="Octubre"/> <input type="text" value="2007"/>
	Hasta: <input type="text" value="16"/> <input type="text" value="Octubre"/> <input type="text" value="2007"/>

#	Código	Nombres	Fecha última modificación	
1	30140	<a href="#">FEIJOO QUEZADA ANA MARIA</a>	2007-10-16	<input type="checkbox"/>
2	39815	<a href="#">FEIJOO QUEZADA MARIA ALEXANDRA</a>	2007-10-16	<input type="checkbox"/>
3	44666	<a href="#">FEIJOO QUEZADA MARIA VERONICA</a>	2007-10-16	<input type="checkbox"/>
4	29930	<a href="#">DURAN CARRASCO MONICA PATRICIA</a>	2007-10-16	<input type="checkbox"/>
5	28950	<a href="#">J J DIANA PAOLA TORRES BUENO</a>	2007-10-16	<input type="checkbox"/>

Gráfico 5.4.2.3 Desactivación de hojas de vida.

Siendo:

1. Si desea desactivar por alumno, debe ingresar su código de estudiante y número de cédula.
2. Si desea desactivar uno o varios estudiantes puede hacerlo introduciendo un rango de fechas.
3. Si desea visualizar la hoja de vida del estudiante, de clic sobre el nombre.
4. Presione este botón para seleccionar toda la lista de alumnos, y vuelva a presionar si desea quitar la selección de todos.
5. Presione el botón “Desactivar hojas de vida” para desactivar todas las hojas de vida seleccionadas.

Luego de presionar el botón “Desactivar hojas de vida”, se visualizará la siguiente pantalla.

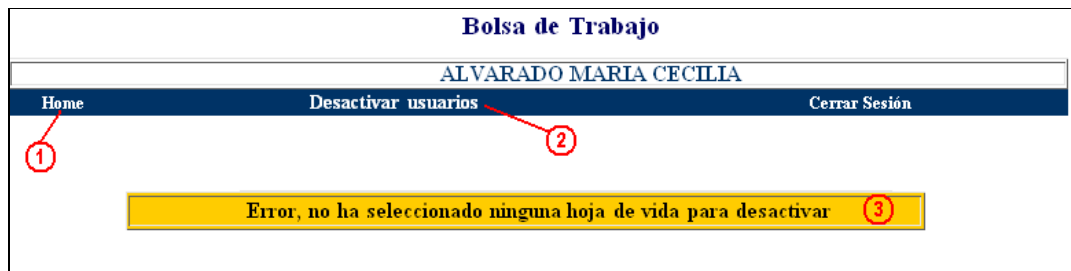


Gráfico 5.4.2.3.1 Presentación de mensaje en caso de desactivación.

En donde:

1. Vínculo que re-direcciona a la página principal del administrador.
2. Vínculo que re-direcciona a la página de desactivar usuarios.
3. Mensaje que visualiza el número de hojas que se han desactivado correctamente. Si en la página anterior no seleccionó ninguna hoja de vida para activar, se visualizará el mensaje “Error, no ha seleccionado ninguna hoja de vida para desactivar”.

### Validar por estudiante.

En caso de no existir hojas de vida pendientes por validar, se visualizará el mensaje “No existen carpetas por validar”, caso contrario se visualizará la siguiente pantalla.

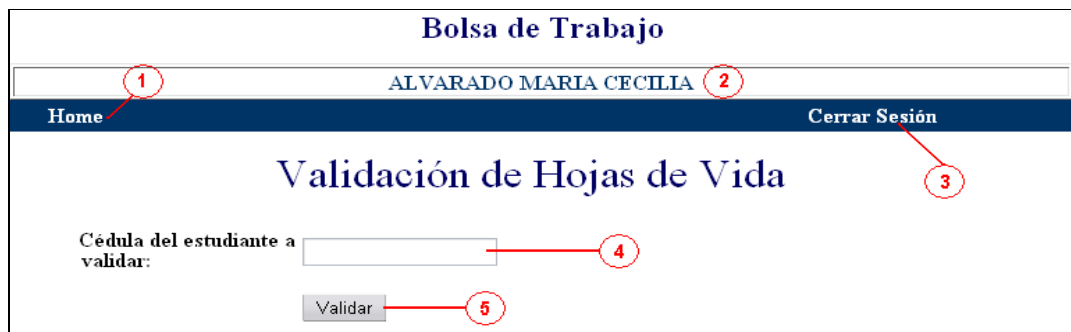


Gráfico 5.4.2.4 Validación de hojas de vida por el administrador.

En donde:

1. Vínculo que re-direcciona al usuario a la página principal del administrador.
2. Nombre del usuario administrador que se visualiza automáticamente según los datos ingresados por el usuario. Todas las acciones que realice un determinado usuario quedarán registradas con su nombre.

- Vínculo que re-direcciona al usuario a la página de inicio, y además permite al usuario desconectarse del sistema. Cabe mencionar que el usuario siempre deberá cerrar sesión una vez que haya terminado sus actividades, para que ninguna persona que haga uso del mismo navegador pueda realizar acciones malintencionadas en las cuales quedará registrado dicho usuario como único responsable.
- Ingrese el número de cédula del estudiante cuya hoja de vida va a ser validada.
- Presione el botón Validar para visualizar y modificar la hoja de vida del alumno que desee validar.

En caso de que la cédula ingresada sea incorrecta o no se encuentre en la base de datos de la bolsa de trabajo se presentará el siguiente mensaje el mensaje de error pertinente; si no se visualizará la hoja de vida del estudiante.

Gráfico 5.4.2.4. Inicio de formulario para validación de hojas de vida por el administrador.

En donde:

- Vínculo que re-direcciona a la página principal del administrador.
- Vínculos que permiten acceder de forma inmediata a una sección de información determinada.

**Horario de trabajo:**  Tiempo completo  Medio tiempo  Por horas  Por proyecto  
**Turnos rotativos:**  Si  No  
**Horario nocturno:**  Si  No  
**¿Dispuesto(a) a trabajar en otra ciudad?:**  Si  No  
**Viajar:**  Dentro de la región  Dentro del país  Fuera del país  
**¿Si la empresa lo requiere, puede disponer de su vehiculo para el trabajo?:**  Si  No  
**¿Si la empresa lo requiere, puede disponer de computador personal/agenda electrónica?:**  Si  No

**Validación de Hoja de Vida:**

Validado  No Validado

**Usuario:** ALVARADO MARIA CECILIA

*Antes de presionar el botón Grabar por favor verifique que la información ingresada esté correcta.*

Gráfico 5.4.2.4.2 Final de formulario para validación de hojas de vida por el administrador

3. Permite cambiar el estado de una hoja de vida
4. Permite borrar todos los cambios realizados sobre la hoja de vida, y dejar como inicialmente se visualizó.
5. Guarda los cambios realizados.

Al presionar el botón guardar, se visualizará una nueva pantalla en donde se listarán los errores en caso de existir, en donde se especifica la sección en donde se encuentra el error, el tipo de error, y el dato erróneo.

En caso de no existir error, se guardarán los datos modificados y se presentarán los mensajes correspondientes.

## Quitar validación.

**Quitar validación de Hojas de Vida**

<input checked="" type="radio"/> <b>Por alumno:</b>	Código alumno: <input type="text"/>
	Cédula: <input type="text"/>
<input type="radio"/> <b>Por rango de fechas:</b>	Desde: <input type="text" value="16"/> <input type="text" value="Octubre"/> <input type="text" value="2007"/>
	Hasta: <input type="text" value="16"/> <input type="text" value="Octubre"/> <input type="text" value="2007"/>

#	Código	Nombres	Fecha última modificación	
1	30140	<a href="#">FEJOO QUEZADA ANA MARIA</a>	2007-10-16	<input type="checkbox"/>
2	39815	<a href="#">FEJOO QUEZADA MARIA ALEXANDRA</a>	2007-10-16	<input type="checkbox"/>
3	44666	<a href="#">FEJOO QUEZADA MARIA VERONICA</a>	2007-10-16	<input type="checkbox"/>
4	29930	<a href="#">DURAN CARRASCO MONICA PATRICIA</a>	2007-10-16	<input type="checkbox"/>
5	28950	<a href="#">J J DIANA PAOLA TORRES BUENO</a>	2007-10-16	<input type="checkbox"/>

Gráfico 5.4.2.5 Quitar validación de hojas de vida.

Siendo:

1. Si desea quitar la validación por alumno, debe ingresar su código de estudiante y número de cédula.
2. Si desea quitar la validación de uno o varios estudiantes puede hacerlo introduciendo un rango de fechas.
3. Si desea visualizar la hoja de vida del estudiante, de clic sobre el nombre.
4. Presione este botón para seleccionar toda la lista de alumnos, y vuelva a presionar si desea quitar la selección de todos.
5. Presione el botón “Quitar validación de hojas de vida” para quitar la validación de todas las hojas de vida seleccionadas.

Luego de presionar el botón “Quitar validación de hojas de vida”, se visualizará en pantalla un mensaje indicando el número de hojas de vida correctamente modificadas, o el error correspondiente.

### 5.4.3 Consultas.

En esta sección el usuario administrador podrá realizar distintos tipos de consultas.

The screenshot shows a web interface for 'ALVARADO MARIA CECILIA'. At the top, there are links for 'Home' and 'Cerrar Sesión'. The main heading is 'Menú Principal'. Below this, there are three sections: 'Mantenimiento de Usuarios', 'Mantenimiento de Hojas de Vida', and 'Consultas'. Each section has a 'Menú:' label and a dropdown menu with an 'Ir' button. In the 'Consultas' section, the dropdown menu is open, showing options: 'Sugerencias', 'Hojas de vida pendientes', 'Sugerencias', and 'Usuarios'. A red circle with the number '1' is drawn around the 'Hojas de vida pendientes' option.

Gráfico 5.4.3 Consultas.

Seleccione la acción que desee realizar y presione el botón “Ir” para visualizar la pantalla correspondiente.

### Hojas de vida pendientes

The screenshot shows the 'Listado de alumnos de la bolsa de trabajo pendientes de validar' interface. At the top, there are links for 'Home' and 'Cerrar Sesión'. Below this, there is a dropdown menu with the text 'Pendientes por validar' and a 'Consultar' button. A red circle with the number '1' is drawn around the dropdown menu, and a red circle with the number '2' is drawn around the 'Consultar' button. Below the dropdown menu, there is a table with the following data:

#	Código	Nombres	Fecha Ingreso	Responsable
1	30140	<a href="#">FELIOO QUEZADA ANA MARIA</a>	2007-10-16	ua28950
2	39815	<a href="#">FELIOO QUEZADA MARIA ALEXANDRA</a>	2007-10-16	ua29930
3	44666	<a href="#">FELIOO QUEZADA MARIA VERONICA</a>	2007-10-16	ua29930
4	28951	<a href="#">CORDERO CAMPOVERDE VERONICA ELIZABETH</a>	2007-10-16	ua29930
5	29930	<a href="#">DURAN CARRASCO MONICA PATRICIA</a>	2007-10-16	rootadm

A red circle with the number '3' is drawn around the first row of the table.

Gráfico 5.4.3.1 Presentación de la consulta de hojas de vida.

En donde:

1. Seleccione “Pendientes por validar” para visualizar todas las hojas de vida ingresadas por los estudiantes y que no estén validadas, “Pendientes por activar” para visualizar aquellas hojas de vida que han sido validadas, pero no están activas.
2. Presione el botón “Consultar” para visualizar el listado correspondiente.
3. Si desea visualizar la hoja de vida del estudiante, de clic sobre el nombre.

### Sugerencias y comentarios.

# Sugerencia	Fecha
1 felicitaciones!!!	2007-10
2 no hay sugerencias, todo esta perfecto!	2007-10
3 chevere la paginal la UDA necesita gente como ustedes.	2007-10

Gráfico 5.4.3.2 Presentación de la consulta de sugerencias.

En donde:

1. Seleccione “Todos” para visualizar todas las sugerencias y comentarios realizados por los estudiantes.
2. Seleccione “Por rango de fechas” para realizar una búsqueda más específica.
3. Presione el botón “Ver” para visualizar el listado correspondiente.

## **CAPÍTULO 6**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **6.1 Conclusiones.**

Se ha cumplido el objetivo de crear una aplicación para el manejo de la bolsa de trabajo de la Universidad del Azuay, en base a los requerimientos proporcionados por las partes implicadas en el proyecto.

Debido a que hay varias personas involucradas en el proyecto de la bolsa de trabajo, éste fue muy difícil de coordinar, por lo cual se deberán seguir realizando cambios en el futuro.

#### **6.2 Recomendaciones.**

Es imprescindible que tanto el usuario administrador, como los usuarios validadores analicen previamente el manual del usuario presentado en este proyecto, para que tengan una idea clara del funcionamiento del sistema.

Antes de efectuar alguna modificación de las funcionalidades del sistema, se debe revisar minuciosamente el manual del programador que indica a detalle para qué se emplea cada archivo, incluyendo sus funciones, variables, entre otros.

Para que este proyecto logre el éxito deseado, creemos conveniente hacer pública la re-estructuración de la bolsa de trabajo; no solo a los estudiantes de la Universidad del Azuay, sino también a las empresas que pudieran interesarse en este servicio por los medios de comunicación comunes.



## BIBLIOGRAFÍA

<http://www.desarrolloweb.com/manuales/12>. DESARROLLOWEB. ÁLVAREZ Miguel, ÁLVAREZ Rubén y Cuenca Carlos. 30 de junio del 2007.

<http://www.desarrolloweb.com/articulos/239.php>. DESARROLLOWEB. ÁLVAREZ Miguel, ÁLVAREZ Rubén y Cuenca Carlos. 20 de julio del 2007.

[http://es.wikipedia.org/wiki/Script\\_del\\_lado\\_del\\_servidor](http://es.wikipedia.org/wiki/Script_del_lado_del_servidor). WIKIPEDIA. DROIT D'auteur. 21 de julio del 2007.

[http://php.ciberaula.com/articulo/PHPoASP/es.wikipedia.org/wiki/Script\\_del\\_lado\\_d\\_el\\_servidor](http://php.ciberaula.com/articulo/PHPoASP/es.wikipedia.org/wiki/Script_del_lado_d_el_servidor). CIBERAULA. 21 de julio del 2007.

<http://www.uterra.com/zonaPHP>. UTERRA. 21 de julio del 2007.

[http://www.wikilearning.com/lenguajes\\_de\\_lado\\_servidor-wkccp-4462-4.htm](http://www.wikilearning.com/lenguajes_de_lado_servidor-wkccp-4462-4.htm). WIKILEARNING. EMAGISTER.COM. 21 de julio del 2007.

<http://www.matpec.com.ar/desde0/desde0-5-lenguaje-servidor.htm>. MATPEC. 15 de septiembre del 2007.

<http://www.htmlquick.com/es/beyond-html.html>. HTMLQUICK. 19 de septiembre del 2007.

<http://tramullas.com>. TRAMULLAS. TRAMULLAS José. 5 de julio de 2007.

<http://www.uazuay.edu.ec/servicios/facultades/detalleteria.php>. UNIVERSIDAD DEL AZUAY. ESQUIVEL Pablo. 5 de julio del 2001.