



Universidad del Azuay

Facultad de Administración de Empresas

Escuela de Ingeniería de Sistemas

Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano, utilizando el framework  
Bootstrap

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

Autor: Estéfano Barrera M.

Director: Mgst. Catalina Astudillo

Cuenca, Ecuador

2018

# Índice de Contenidos

Índice de Contenidos.....	ii
Índice de Ilustraciones .....	v
Resumen .....	ix
Abstract.....	x
Introducción .....	1
<b>CAPÍTULO 1: TECNOLOGÍAS APLICADAS .....</b>	<b>3</b>
<b>1.1. Bootstrap .....</b>	<b>3</b>
1.1.1. HTML 5.....	3
1.1.2. JavaScript .....	9
1.1.3. AJAX.....	14
1.1.4. jQuery .....	19
1.1.5. CSS3 .....	27
1.1.6. Bootstrap.....	29
<b>1.3. SQL Server 2016 Express .....</b>	<b>39</b>
1.2.1. Base de datos relacionales.....	39
1.2.2. Tipos de instrucciones SQL .....	39
1.2.3. Características SQL Server 2016 .....	40
<b>1.3. ASP.NET.....</b>	<b>41</b>
1.3.1. Componentes de ASP.NET .....	42
1.3.2. Posibilidades de desarrollo con ASP.NET.....	46
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>48</b>
<b>2.1. Fundamentos sobre Recursos Humanos .....</b>	<b>48</b>
2.1.1. Introducción .....	48
2.1.2. Componentes .....	49
2.1.3. Objetivos .....	49
<b>2.2. Fundamentos sobre Talento Humano .....</b>	<b>51</b>
2.2.1. Gestión Talento Humano.....	52

2.2.2.	Objetivos .....	52
<b>2.3.</b>	<b>Especificación de Requisitos de Software (ERS).....</b>	<b>53</b>
2.3.1.	Diagramas de Casos de Uso .....	54
2.3.2.	Descripción de Casos de Uso .....	56
<b>2.4.</b>	<b>Diseño de Datos .....</b>	<b>74</b>
2.4.1.	Diagrama de Entidad – Relación.....	74
2.4.2.	Diccionario de Datos .....	76
<b>2.5.</b>	<b>Diseño de Procesos .....</b>	<b>83</b>
2.5.1.	Diagrama de Actividades.....	83
<b>2.6.</b>	<b>Diseño Arquitectónico.....</b>	<b>89</b>
2.6.1.	Diagrama de despliegue.....	89
2.6.3.	Diseño de interfaces. ....	90
<b>CAPÍTULO 3: CODIFICACIÓN, PRUEBAS Y VALIDACIÓN .....</b>		<b>94</b>
<b>3.1.</b>	<b>Codificación del sistema.....</b>	<b>94</b>
<b>3.2.</b>	<b>Pruebas de Interfaz. ....</b>	<b>99</b>
<b>3.3.</b>	<b>Validación del sistema .....</b>	<b>101</b>
<b>CAPÍTULO 4: MANUAL DE USUARIO .....</b>		<b>105</b>
<b>4.1.</b>	<b>Ingreso al sistema .....</b>	<b>105</b>
<b>4.2.</b>	<b>Administración General .....</b>	<b>106</b>
4.2.1.	Nivel de Instrucción .....	106
4.2.2.	Procesos .....	108
4.2.3.	Habilidades y Destrezas.....	108
4.2.4.	Departamentos.....	110
4.2.5.	Configuración de Habilidades por formulario .....	111
4.2.6.	Requisitos por proceso .....	113
<b>4.3.</b>	<b>Cargos .....</b>	<b>115</b>
4.3.1.	Nuevo Cargo .....	115
4.3.2.	Edición.....	118
<b>4.4.</b>	<b>Empleados .....</b>	<b>119</b>

4.4.1.	Ingreso Empleado.....	119
4.4.2.	Edición Empleado.....	122
4.4.3.	Folder Empleado .....	122
<b>4.5.</b>	<b>Postulantes.....</b>	<b>123</b>
4.5.1.	Ingreso .....	124
4.5.2.	Visualización.....	129
4.5.3.	Confirmación de Referencias.....	130
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>133</b>
<b>BIBLIOGRAFÍA .....</b>		<b>135</b>
<b>ANEXOS.....</b>		<b>137</b>
<b>Anexo 1: HTML 5, Nuevos Elementos .....</b>		<b>137</b>
	Nuevos tipos de elementos en formularios .....	137
	Elementos estructurales. ....	138
	Elementos semánticos en línea .....	140
	Elementos de interacción. ....	141
	Elementos embebidos.....	143
<b>Anexo 2: JavaScript, métodos para acceder al DOM .....</b>		<b>145</b>
<b>Anexo 3: jQuery, Eventos, Métodos y Selectores.....</b>		<b>147</b>
	Tipos de eventos.....	147
	Selectores de Elementos .....	148
	Métodos.....	152
<b>Anexo 4: CSS3, Nuevas Funciones, Propiedades, Transiciones y Selectores. ....</b>		<b>173</b>
	Tipos de selectores.....	173
	Funciones de manipulación de colores y opacidad. ....	174
	Nuevas Propiedades. ....	175
	Nuevas Transiciones. ....	180
<b>Anexo 5: Capturas de pantalla pruebas de interfaz adaptable.....</b>		<b>181</b>
<b>Anexo 6. Desarrollo Especificación de Requisitos de Software (ERS).....</b>		<b>196</b>

## Índice de Ilustraciones

Ilustración 1 Atributo Desingmode.....	5
Ilustración 3 Agregando elemento <video>.....	6
Ilustración 5 Contenido Audio .....	8
Ilustración 7 Agregando Evento onError().....	8
Ilustración 9 Variables JavaScript.....	10
Ilustración 10 Funciones JavaScript.....	11
Ilustración 11 JavaScript En Línea .....	12
Ilustración 12 Etiquetas <script>.....	13
Ilustración 13 Archivos externos JavaScript .....	14
Ilustración 14 Diferencia método asíncrono y síncrono .....	15
Ilustración 15 Tecnologías en AJAX .....	16
Ilustración 17 Agregando librería jQuery .....	20
Ilustración 18 Agregando librería jQuery Opción 2.....	20
Ilustración 19 Método inicializa jQuery .....	21
Ilustración 20 Método attr() .....	22
Ilustración 21 Método attr() asignación.....	22
Ilustración 22 Método addClass() .....	23
Ilustración 23 Manejadores de eventos jQuery .....	24
Ilustración 24 Manipular comportamiento eventos jQuery.....	24
Ilustración 25 Comportamiento en cola jQuery .....	24
Ilustración 26 Método Load .....	25
Ilustración 27 Método getJSON .....	26
Ilustración 28 Método AJAX tipo JSON.....	26
Ilustración 29 Estructura Bootstrap .....	30
Ilustración 30 Agregando elemento DOCTYPE .....	31
Ilustración 31 Agregar elemento <meta> .....	31
Ilustración 32 Estructura Básica con Bootstrap.....	31
Ilustración 33 Elemento Container Bootstrap .....	32
Ilustración 34 Elemento Container-fluid Bootstrap.....	32
Ilustración 36 Visualización Bootstrap en Laptops .....	34
Ilustración 37 Formato Bootstrap Móviles, Tabletas y Laptops .....	34
Ilustración 38 Bootstrap Tabletas .....	35
Ilustración 39 Bootstrap Móviles.....	35
Ilustración 40 Anidamiento de columnas Bootstrap.....	35
Ilustración 41 Visualización Navegador Anidamiento Bootstrap .....	36
Ilustración 42 Atributo offset Bootstrap.....	36
Ilustración 43 Visualización atributo offset Bootstrap .....	36
Ilustración 44 Atributo Push - Pull Bootstrap .....	37
Ilustración 45 Visualización atributo push - pull Bootstrap .....	37
Ilustración 46 Arquitectura ASP.NET .....	42
Ilustración 48 Diagrama Casos de Uso .....	55
Ilustración 50 Diagrama Actividades - Mantenimiento Cargos .....	84
Ilustración 51 Diagrama Actividades Registro Postulantes .....	85
Ilustración 52 Diagrama Actividades. Folder Empleado .....	86
Ilustración 53 Diagrama Actividades. Confirmación Referencias.....	87
Ilustración 54 Diagrama Actividades. Registro Empleado.....	88

Ilustración 55 Diagrama de Despliegue.....	89
Ilustración 56 Interfaz Registro Postulantes .....	91
Ilustración 57 Interfaz Registro de Empleado.....	92
Ilustración 58 Interfaz Ingreso Cargos.....	93
Ilustración 59 Extracto Código Bootstrap Página Cargos.....	95
Ilustración 60 Comparación Bootstrap - HTML .....	96
Ilustración 61 Extracto código página Registro Cargos usando HTML .....	98
Ilustración 62 Formato Pruebas Funcionamiento .....	101
Ilustración 63 Prueba funcionalidad Ingreso sistema.....	101
Ilustración 64 Prueba funcionalidad Mantenimiento Instrucción.....	102
Ilustración 65 Prueba funcionalidad Habilidades formulario .....	102
Ilustración 66 Prueba funcionalidad Ingreso Cargos.....	103
Ilustración 67 Prueba funcionalidad Registro Postulantes .....	103
Ilustración 68 Prueba funcionalidad Confirmación Referencias .....	103
Ilustración 69 Prueba funcionalidad Registro Empleados .....	104
Ilustración 70 Prueba funcionalidad Folder Empleado.....	104
Ilustración 71 Formulario Inicio Sesión.....	105
Ilustración 72 Mantenimiento de Instrucciones .....	106
Ilustración 73 Ingreso de Instrucciones .....	107
Ilustración 74 Edición de Instrucción.....	107
Ilustración 75 Mantenimiento de Procesos.....	108
Ilustración 76 Mantenimiento Habilidades y Destrezas .....	108
Ilustración 77 Mantenimiento Habilidades Destrezas .....	109
Ilustración 78 Edición Habilidades o Destrezas .....	109
Ilustración 79 Eliminación Habilidades Destrezas .....	110
Ilustración 80 Mantenimiento Departamentos.....	110
Ilustración 81 Ingreso Departamentos.....	111
Ilustración 82 Edición Departamentos .....	111
Ilustración 83 Configuración de Habilidades por formulario.....	112
Ilustración 84 Ingreso Habilidades en Formulario.....	112
Ilustración 85 Requisitos por proceso.....	113
Ilustración 86 Ingreso requisitos por proceso .....	114
Ilustración 87 Edición Requisitos por proceso.....	114
Ilustración 88 Eliminación requisitos por proceso .....	114
Ilustración 89 Pantalla Mantenimiento Cargos.....	115
Ilustración 90 Ingreso Cargos .....	116
Ilustración 91 Secciones Ingreso Cargos.....	116
Ilustración 92 Selección Habilidades Cargo .....	117
Ilustración 93 Carga Archivo Cargo .....	117
Ilustración 94 Errores Cargos.....	118
Ilustración 95 Edición de Cargos .....	118
Ilustración 96 Listado Empleados.....	119
Ilustración 97 Ingreso Empleado .....	120
Ilustración 98 Selección Postulantes .....	120
Ilustración 99 Ingresar Requisitos Empleados .....	121
Ilustración 100 Error Ingreso Empleado .....	121
Ilustración 101 Edición de Empleados.....	122
Ilustración 102 Folder Empleado .....	123
Ilustración 103 Agregar archivo Folder.....	123
Ilustración 104 Mantenimiento de Postulantes.....	124
Ilustración 105 Pantalla Inicial.....	124
Ilustración 106 Registro Postulantes.....	125

Ilustración 107 Postulantes Datos Familiares.....	126
Ilustración 108 Postulantes Instrucción .....	126
Ilustración 109 Postulantes Ocupación Actual .....	127
Ilustración 110 Postulantes Ocupaciones Anteriores .....	127
Ilustración 111 Postulantes Referencias Personales .....	128
Ilustración 112 Postulantes Aspiración Salarial .....	128
Ilustración 113 Postulante Registro Completo .....	129
Ilustración 114 Postulante Errores .....	129
Ilustración 115 Consulta Postulantes.....	130
Ilustración 116 Pantalla Confirmación Referencias .....	131
Ilustración 117 Formulario Confirmación Referencias .....	132
Ilustración 119 Elemento <details> .....	142
Ilustración 120 Elemento <datagrid> .....	142
Ilustración 122 Método "getElementByID" .....	145
Ilustración 123 Método "getElementsByClassName" .....	145
Ilustración 124 Método "querySelector".....	145
Ilustración 125 Método "querySelectorAll" .....	146
Ilustración 127 Selector por ID elemento .....	149
Ilustración 128 Selector por clase.....	149
Ilustración 129 Selector universal.....	150
Ilustración 130 Selector múltiple .....	150
Ilustración 131 Selector Find.....	151
Ilustración 133 Método off.....	162
Ilustración 134 Método hover .....	163
Ilustración 135 Método on.....	163
Ilustración 136 Método one .....	164
Ilustración 137 Método ready .....	164
Ilustración 138 Método trigger.....	165
Ilustración 139 Método triggerHandler.....	166
Ilustración 140 Método unbind.....	166
Ilustración 142 Método \$.ajaxSetup .....	169
Ilustración 143 Método \$.get .....	170
Ilustración 144 Método \$.getJSON.....	171
Ilustración 145 Método \$.getScript.....	171
Ilustración 146 Método \$.post .....	172
Ilustración 147 Método serialize .....	172
Ilustración 148 Método serializeArray.....	173
Ilustración 149 Pseudo-elementos CSS3.....	174
Ilustración 150 Pseudo-elemento "selection" CSS3.....	174
Ilustración 151 Modelo Color RGBA CSS3.....	175
Ilustración 152 Modelo Color HSL CSS3 .....	175
Ilustración 153 Modelo Opacidad CSS3.....	175
Ilustración 154 Propiedad border-radius CSS3.....	176
Ilustración 155 Propiedad box-shadow CSS3 .....	177
Ilustración 156 Propiedad box-shadow avanzado CSS3.....	177
Ilustración 157 Propiedad background-image CSS3.....	177
Ilustración 158 Propiedad border-image CSS3.....	178
Ilustración 159 Columnas de texto CSS3.....	178
Ilustración 160 Propiedad WebFonts CSS3 .....	178
Ilustración 161 Propiedad WebFonts CSS3 Varios Navegadores .....	179
Ilustración 162 Propiedad MediaQueries CSS3.....	179
Ilustración 163 Transiciones CSS3 .....	180

Ilustración 164 Transiciones CSS3 Duración y Retraso .....	180
Ilustración 165 SE.iphone6_8 portrait 375px.....	181
Ilustración 166 SE.iphone6_8 landscape 667px.....	181
Ilustración 167 SE.android(pixel2)portrait 412px .....	182
Ilustración 168 SE.android(pixel2)landscape 684px .....	182
Ilustración 169 SE.ipad portrait 768px.....	183
Ilustración 170 SE.ipad landscape 1024px.....	183
Ilustración 171 RC.iphone6_8 portrait 375px .....	184
Ilustración 172 RC.iphone6_8 landcape 667px.....	184
Ilustración 173 RC.android(pixel2) portrait 412px.....	185
Ilustración 174 RC.android(pixel2) landscape 684px.....	185
Ilustración 175 RC.ipad portrait 768px.....	186
Ilustración 176 RC.ipad landscape 1024px.....	186
Ilustración 177 RE.iphone6_8 portrait 375px.....	187
Ilustración 178 RE.iphone6_8 landcape 667px.....	187
Ilustración 179 RE.android(pixel2) portrait 412px.....	188
Ilustración 180 RE.android(pixel2) landscape 684px .....	188
Ilustración 181 RE.ipad portrait 768px.....	189
Ilustración 182 RE.ipad landscape 1024px.....	189
Ilustración 183 RC.Ipad portrait 768px .....	190
Ilustración 184 RC.Ipad landscape 1024px .....	190
Ilustración 185 RE.Iphone6-8 portrait 375px .....	191
Ilustración 186 RE.Iphone6-8 landscape 667px.....	191
Ilustración 187 RE.android.pixel2.portrait 412px .....	192
Ilustración 188 RE.android.pixel2.landscape 684px.....	192
Ilustración 189 RE.ipad.portrait 768px.....	193
Ilustración 190 RE.ipad.landscape 1024px.....	193
Ilustración 191 SE.android.pixel2 portrait 412px.....	194
Ilustración 192 SE.android.pixel2.landscape 684px.....	194
Ilustración 193 SE.ipad.landscape 1024px.....	195

## **RESUMEN**

El presente trabajo presenta un estudio del framework Bootstrap; sus características, beneficios y elementos que permiten desarrollar aplicaciones web que se adaptan a cualquier dispositivo de escritorio o móvil en el que se ejecute, eliminando con esto la necesidad de desarrollar diferentes aplicaciones para cada dispositivo en el cual puede ser utilizado. También se aborda tecnologías que complementan el desarrollo web como HTML5, CSS3, JavaScript y jQuery. Para demostrar los beneficios de Bootstrap, se aplica en un prototipo orientado a la gestión de selección del personal que forma parte del departamento de talento humano, apoyando a la efectividad en la ejecución de sus procesos, a este prototipo se le aplica pruebas de adaptabilidad, funcionalidad y de interfaz.

## ABSTRACT

The present work presented a study of the Bootstrap framework, its characteristics, benefits and elements that allowed to develop web applications that adapt to any desktop or mobile device in which it is executed. This eliminated the need to develop different applications for each device in which it can be used. It also addressed the technologies that complement web development such as HTML5, CSS3, JavaScript and jQuery. Bootstrap was applied in a prototype oriented to the selection management of the personnel from the human talent department. This sought to demonstrate its benefits and to support the effectiveness in the execution of its processes. The adaptability, functionality and interface tests were applied to this prototype.

  
Magali Estrada  
UNIVERSIDAD DEL  
AZUAY  
Dpto. Idiomas

  
Translated by

Ing. Paul Arpi

## INTRODUCCIÓN

Actualmente existe un auge a nivel mundial en el desarrollo de software orientado a dispositivos electrónicos sean estos móviles o de escritorio. Los dispositivos electrónicos se convierten cada día más en una parte fundamental en el desarrollo de labores empresariales, por lo que el software tiene que ser compatible con cualquier dispositivo en el que sea abierto. Hoy en día existen muchas herramientas o librerías dedicadas a cumplir con este objetivo y con esto ayudar a simplificar procesos y mejorar su gestión dentro de la empresa.

Dentro de esta tecnología surge un framework llamado Bootstrap, que se caracteriza por ser una herramienta de software libre, y compatible con los sistemas operativos para dispositivos móviles que existen en el mercado como Windows Phone, IOS, Symbian, entre otros. Trabajar con el framework Bootstrap en el desarrollo de aplicaciones web elimina las preocupaciones sobre las diferentes características de los navegadores de internet y las plataformas en las cuales pueden ser ejecutadas las aplicaciones, dejando de lado características como tamaños de pantalla, resoluciones de pantalla, sistemas operativos y permitiendo enfocarse completamente en el contenido de la aplicación. Al ya no tener que preocuparse por las características de cada dispositivo el desarrollador termina con una sola versión del software, versión que se adapta automáticamente a cualquier dispositivo en el que sea ejecutado.

Por otro lado, varias empresas cuentan en su organización con el área de talento humano enfocada a la selección de personal que ingresa a la empresa, pero, por su historia han sido relevadas a trabajar con archivos físicos, manejo de grandes archivadores, pérdida de documentos, deterioro de archivos. Sin proporcionarles herramientas que puedan ayudar en su gestión.

Por lo mencionado, en el presente trabajo el objetivo principal es desarrollar un sistema web multiplataforma de gestión de selección del personal para el área de talento humano utilizando el framework Bootstrap con la finalidad de aprovechar las nuevas tecnologías de software y apoyar a la gestión y productividad del departamento de talento humano.

Esto se lleva a cabo con el cumplimiento de los siguientes objetivos específicos:

- a. Analizar el framework Bootstrap y tecnologías asociadas. Este objetivo se aborda en el primer capítulo del documento. Capítulo que analiza de manera detallada el framework Bootstrap y las tecnologías aplicadas HTML5, CSS3, jQuery y JavaScript que con sus características potencian el objetivo del framework Bootstrap.
- b. Realizar el levantamiento de información del departamento de talento humano. Este objetivo se aborda en el segundo capítulo del documento en el que se levanta la información sobre la historia, objetivos y componentes que tienen las áreas de recursos humanos y talento humano en las empresas.
- c. Diseñar y desarrollar el sistema web para el departamento de talento humano. Este objetivo se aborda en una primera instancia en el capítulo dos del documento. Capítulo en el cual se realiza el diseño del sistema a través del levantamiento de las especificaciones de software y la creación de los diagramas de casos de uso, entidad-relación, actividades e interfaz.

En el tercer capítulo se ilustra la codificación utilizada en el sistema, codificación que emplea las funcionalidades proporcionadas por el framework Bootstrap y las tecnologías asociadas. Para finalmente verificar el software desarrollado mediante las pruebas de interfaz y las pruebas de funcionalidad. Mismas que arrojarán los resultados obtenidos y los cambios realizados para alcanzar el objetivo del sistema.

## CAPÍTULO 1: TECNOLOGÍAS APLICADAS

Este capítulo es una recopilación de información sobre las herramientas necesarias para trabajar en un desarrollo adaptable de tecnologías web utilizando el framework Bootstrap. El contenido se enfoca en detallar las principales características, funcionalidades y manera de interactuar de cada una de las herramientas con el framework.

### 1.1. Bootstrap

Bootstrap es un framework desarrollado bajo librerías de software libre, que permite crear páginas web adaptables. Las páginas web adaptables brindan a los desarrolladores de software la posibilidad de presentar el contenido de la página web en diferentes dispositivos sean de escritorio o móviles con un solo código fuente. Entre las principales características del desarrollo adaptable están: minimización de tiempos de desarrollo, trabajar con una sola plataforma tecnológica, facilidad para realizar pruebas de interfaz entre otras.

A continuación, se analizará las librerías, sus elementos, características y funciones, usadas paralelamente con el framework Bootstrap, mismas que son necesarias para obtener un producto final completamente adaptable a cualquier dispositivo electrónico.

#### 1.1.1. HTML 5

HTML5 es la quinta versión del conocido lenguaje de programación web HTML desarrollado por el consorcio W3C (World Wide Web) encargado de manejar los estándares en el desarrollo web. En 1998 dicho estándar estaba por desaparecer y ser sustituido por una nueva versión llamada XHTML, hasta que un grupo de desarrolladores crearon la nueva versión de HTML. Esta nueva versión trae nuevos elementos, atributos y comportamientos que permitirán crear sitios web dinámicos, creativos, robustos e interactivos. (Matthew, 2011)

Algunos de los nuevos elementos y atributos conllevan el uso de JavaScript (JS) y Hojas de estilo (CSS3) para obtener su máximo rendimiento. Entre algunas de las

principales novedades se encuentran: elementos para incluir audio y video, creación de formularios con elementos que permiten realizar validaciones y entregar mensajes sin necesidad de enviar una solicitud hacia el servidor, método mejorado de almacenamiento de datos en el lado del cliente entre otros.

(Holzner, 2011) en su libro "HTML 5 in 10 Minutes", detalla las principales novedades de la nueva versión de HTML5.

#### 1.1.1.1. Nuevos atributos

La nueva versión de HTML tiene dos principales novedades en lo que se refiere a atributos. A continuación se detalla cuáles son:

- Contenteditable

Este nuevo atributo permite convertir el contenido de un elemento en manipulable, es decir tiene la posibilidad de escribir e inclusive ingresar elementos dentro del elemento. Este atributo no está disponible para todos los elementos, principalmente aparece en el elemento <div>.

Este atributo acepta los siguientes valores:

- true.- Convierte al objeto en editable.
- false.- No permite editar el contenido del objeto.
- inherit.- Recoge el valor del atributo del elemento padre.

- Desingmode

Este atributo convierte el contenido completo de un documento en editable, como es un atributo para un documento este no se puede incluir en elementos internos. Este atributo recibe dos posibilidades: on/off.

Ejemplo del uso del atributo desingmode.

```
2 <html>
3 <head>
4 <title>
5 Ejemplo Desingmode
6 </title>
7 <script>
8 function startEdit() {
9 // Turn on design mode in the <iframe>.
10 var editor = document.getElementById("pageEditor");
11 editor.contentWindow.document.designMode = "on";
12 }
13 function stopEdit() {
14 // Turn off design mode in the <iframe>.
15 var editor = document.getElementById("pageEditor");
16 editor.contentWindow.document.designMode = "off";
17 // Display the edited HTML (just to prove it's there).
18 var htmlDisplay = document.getElementById("editedHTML");
19 htmlDisplay.textContent = editor.contentWindow.document.body.innerHTML;
20 }
21 </script>
22 </head>
23 <body>
24 <iframe id="pageEditor" src="details.HTML"></iframe>
25 <div>
26 <button onclick="startEdit()">Start Editing</button>
27 <button onclick="stopEdit()">Stop Editing</button>
28 </div>
29 <h1>Edited HTML</h1>
30 <div id="editedHTML"></div>
31 </body>
32 </html>
```

Ilustración 1 Atributo Desingmode

### 1.1.1.2. Nuevos elementos

Los nuevos elementos presentes en esta versión de HTML, promueven un cambio en la semántica al momento de estructurar los sitios web, agregando elementos como secciones. Así también esta versión trae nuevos elementos multimedia que pueden ser embebidos en los documentos, además incorpora nuevos elementos en línea como por ejemplo el elemento <time>.

Los nuevos elementos en HTML5 son clasificados de acuerdo a su función, por lo que se encuentra los siguientes tipos: elementos estructurales, semánticos, interacción y embebidos (Harold, 2007). El detalle de los nuevos elementos incorporados se desglosa en el Anexo 1: HTML 5, Nuevos Elementos.

### 1.1.1.3. Usando los elementos Audio y Video

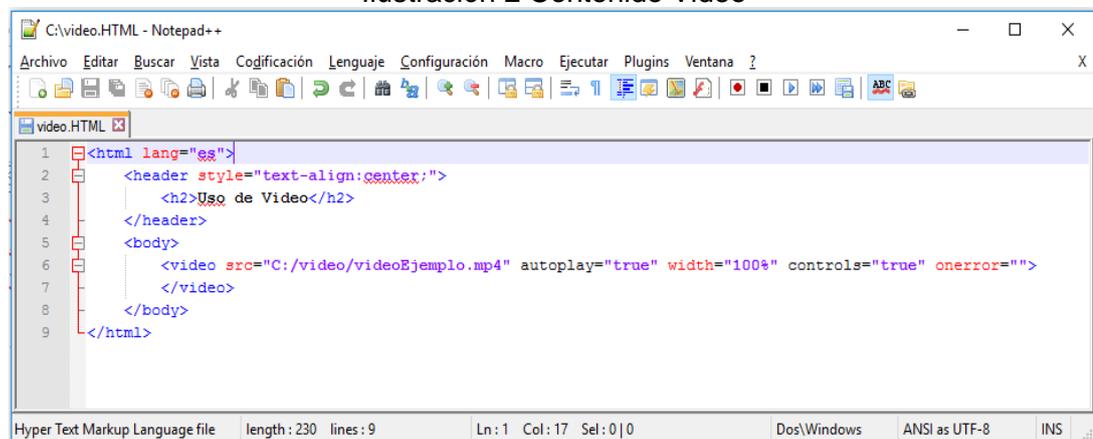
El principal inconveniente con el elemento video, es que muchos de los navegadores solo soportan videos con formato de código abierto (.ogg). Sin embargo, es sencillo cambiar los formatos conocidos de video hacia el formato que se necesite.

### 1.1.1.3.1. Añadiendo video en documento

En esta sección se presenta la creación de un documento en el que se incluye un elemento tipo video. Mediante este ejemplo se revisa las opciones disponibles para este elemento. En el anexo 1: Elementos embebidos se detalla los atributos que dispone el elemento <video>.



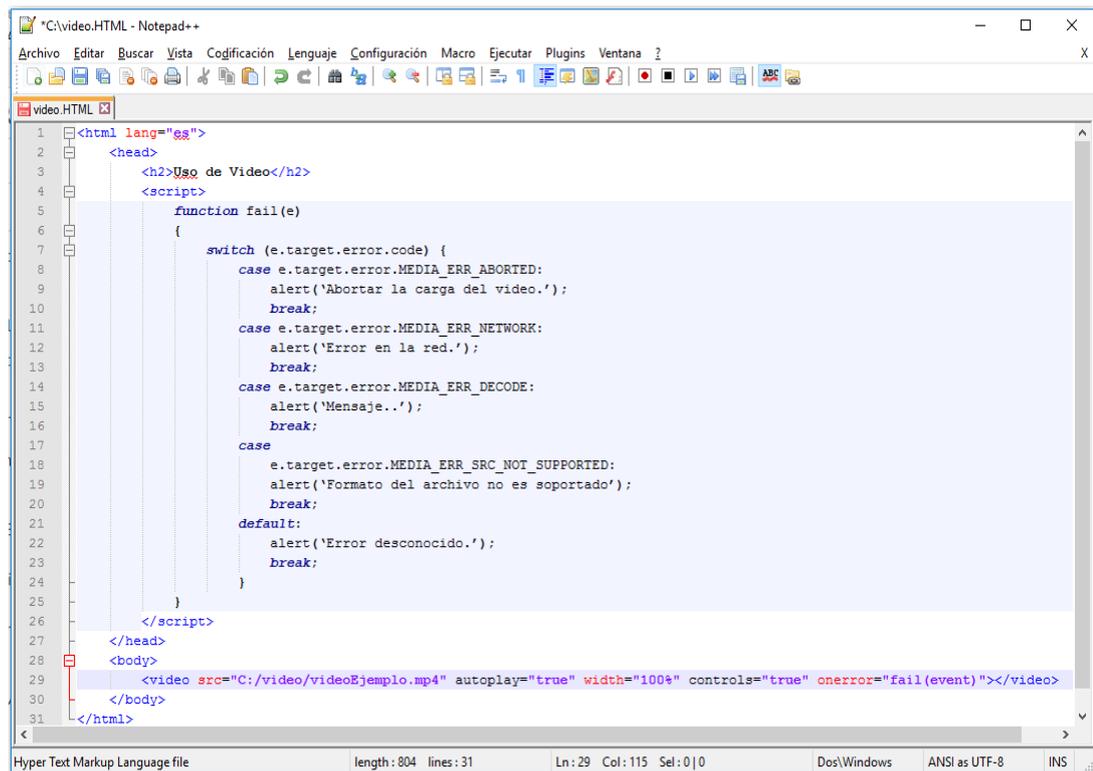
Ilustración 2 Contenido Video

A screenshot of a Notepad++ editor window titled 'C:\video.HTML - Notepad++'. The editor displays the following HTML code:

```
1 <html lang="es">
2   <header style="text-align:center;">
3     <h2>Uso de Video</h2>
4   </header>
5   <body>
6     <video src="C:/video/videoEjemplo.mp4" autoplay="true" width="100%" controls="true" onerror="">
7   </video>
8 </body>
9 </html>
```

The status bar at the bottom indicates 'Hyper Text Markup Language file', 'length: 230 lines: 9', and 'Ln: 1 Col: 17 Sel: 0|0'. The system tray shows 'Dos\Windows', 'ANSI as UTF-8', and 'INS'.

Ilustración 3 Agregando elemento <video>



```
1 <html lang="es">
2 <head>
3 <h2>Uso de Video</h2>
4 <script>
5     function fail(e)
6     {
7         switch (e.target.error.code) {
8             case e.target.error.MEDIA_ERR_ABORTED:
9                 alert('Abortar la carga del video.');
```

The image shows a Notepad++ window titled "C:\video.HTML - Notepad++". The menu bar includes "Archivo", "Editar", "Buscar", "Vista", "Codificación", "Lenguaje", "Configuración", "Macro", "Ejecutar", "Plugins", and "Ventana". The toolbar contains various icons for file operations and editing. The main text area shows an HTML document with the following code:

```
1 <html lang="es">
2 <head>
3 <h2>Uso de Video</h2>
4 <script>
5     function fail(e)
6     {
7         switch (e.target.error.code) {
8             case e.target.error.MEDIA_ERR_ABORTED:
9                 alert('Abortar la carga del video.');
```

The status bar at the bottom indicates "Hyper Text Markup Language file", "length: 804 lines: 31", "Ln: 29 Col: 115 Sel: 0 | 0", "Dos\Windows", "ANSI as UTF-8", and "INS".

Ilustración 4 Agregando Evento onError()

Si el navegador no soporta los videos en formato mp4, mpg o cualquier otro hay una opción en la cual se puede cambiar al formato soportado por la mayoría de navegadores (ogg) mediante la página web <http://media-convert.com/>.

#### 1.1.1.3.2. Añadiendo Audio en documento

El elemento audio al igual que el elemento video tiene problemas con sus formatos al momento de ser incluido, ya que muchos navegadores soportan el formato estándar (.ogg). En la sección anterior (Añadiendo video en documento) se detalla cómo cambiar el formato del audio si es necesario. En el anexo 1: Elementos embebidos se detalla los atributos que dispone el elemento <audio>.

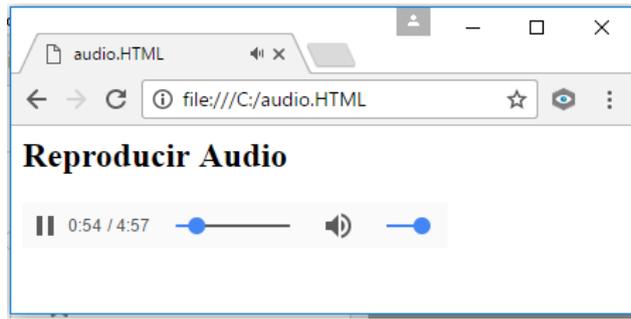


Ilustración 5 Contenido Audio

Ejemplo del uso del elemento <audio>.

```

1 <html lang="es">
2   <head>
3     <h2>Reproducir Audio</h2>
4   </head>
5   <body>
6     <audio src="C:/audio_prueba.mp3" autoplay="true" controls="true" loop="false"></audio>
7   </body>
8 </html>

```

Ilustración 6 Agregando elemento <audio>

```

1 <html lang="es">
2   <head>
3     <h2>Reproducir Audio</h2>
4     <script>
5       function fail(e)
6       {
7         switch (e.target.error.code) {
8           case e.target.error.MEDIA_ERR_ABORTED:
9             alert('Abortar la carga del audio.');
```

```

10             break;
11           case e.target.error.MEDIA_ERR_NETWORK:
12             alert('Error en la red.');
```

```

13             break;
14           case e.target.error.MEDIA_ERR_DECODE:
15             alert('Mensaje.');
```

```

16             break;
17           case
18             e.target.error.MEDIA_ERR_SRC_NOT_SUPPORTED:
19             alert('Formato del archivo no es soportado');
```

```

20             break;
21           default:
22             alert('Error desconocido.');
```

```

23             break;
24         }
25       }
26     </script>
27   </head>
28   <body>
29     <audio src="C:/audio_prueba.mp3" autoplay="true" controls="true" loop="false" onerror="fail(event)"></audio>
30   </body>
31 </html>

```

Ilustración 7 Agregando Evento onError()

#### 1.1.1.4. Nuevos controles en formularios

HTML en su nueva versión incorpora varios tipos de controles que están disponibles en el ya conocido elemento `<input>`, entre los más utilizados están el tipo `date`, `email` y `password`. En los Anexo 1: Nuevos tipos de elementos en formularios, se detalla la lista completa de tipos de elementos `<input>`.

Algo importante para tener en cuenta, es que no todos los nuevos tipos de campos son soportados por todos los navegadores disponibles. Hay que poner atención a las especificaciones detalladas por la W3C en su página oficial.

#### 1.1.2. JavaScript

JavaScript es un lenguaje script orientado a objetos, incorporado al desarrollo web, desde el año 1995. Este lenguaje permite realizar interacciones, entre objetos dentro de la página web, sin la necesidad de recargar la página. Todo esto es un gran avance en el mundo del desarrollo web, ya que elimina conexiones entre el front-end y el back-end, por lo que sin duda es un lenguaje muy utilizado por los desarrolladores (Haverbeke, 2014).

JavaScript al igual que HTML tiene algunas versiones en su historia, versiones que partieron desde la primera en el año 1995 hasta la quinta que comenzó su desarrollo en el año 2009.

(Haverbeke, 2014), en su libro “Eloquent JavaScript” aborda de manera amplia el lenguaje de programación JavaScript, esta sección es una recopilación de los aspectos básicos y principales funciones presentados en su libro.

##### 1.1.2.1. Estructura básica

Se empieza analizando la estructura básica de un documento JavaScript, conociendo expresiones, funciones, variables, palabras reservadas entre otras cosas que ayudarán a conocer de cerca el lenguaje.

```
<script>
  var calculo = 0;

  function multiplicar() {
    alert("Nueva Función");
  }
</script>
```

Ilustración 8 Estructura JavaScript

### 1.1.2.2. Variables

Las variables en JavaScript son declaradas bajo la palabra reservada “var” y pueden ser declaradas de forma global (fuera de funciones) o de forma local (dentro de una función). Los nombres de las variables pueden contener cualquier letra o número, no pueden llevar espacios en blanco pero si pueden incluir signos como el guion bajo (\_) o signo de moneda (\$).

En este lenguaje no necesariamente se necesita inicializar una variable, es decir puede o no tener un valor inicial. Entonces si la variable no tiene un valor inicial, el lenguaje entenderá su tipo (string, numérico) cuando se le asigne un valor.

```
var calculo = 0;
var mensaje;
var suma = calculo + 5;
```

Ilustración 9 Variables JavaScript

### 1.1.2.3. Palabras reservadas

En el lenguaje JavaScript existen varias palabras reservadas, manejadas en el idioma inglés. Las palabras reservadas no pueden ser utilizadas como nombres de variables o con cualquier otro objetivo.

A continuación, listamos las principales palabras reservadas.

- break
- case
- catch
- const
- continue
- default
- delete
- do
- else
- export
- extends
- false
- finally
- for
- function
- if
- import
- new

- null
- package
- private
- return
- static
- throw
- true
- try
- var
- void
- while

#### 1.1.2.4. Funciones

Las funciones son la parte fundamental de este lenguaje, ya que fueron creadas con el objetivo de clasificar en partes el código generado en un programa. Esto ayuda a estructurar de mejor manera el código, especialmente cuando su código es muy extenso. De esta manera se elimina código repetido y facilita la reutilización del código en otras funciones o secciones.

##### 1.1.2.4.1. Definición de funciones

Las funciones se declaran con la palabra reservada “function”, acompañada con un nombre que describa el contenido o el objetivo de la función. Una función puede o no recibir parámetros o variables de entrada, estas variables de ser incorporadas serán utilizadas dentro de la función.

Una función puede ser asignada directamente a una variable, es decir que el resultado de la función va ser almacenado en la variable. Adicionalmente una función puede contener una o varias funciones en su interior, que de igual manera que su objetivo principal es mejorar la estructura del código que tenga el programa.

```
function multiplicar() {
  var valor = 5, valor2 = 10;
  var total = valor * valor2;
  return total;
}

var calculos = function(){
  var resultado;
  var valor1 = 15, valor2 = 5;

  var multiplicacion = function(uno, dos) {
    resultado = uno * dos;
  }

  var suma = function(uno, dos) {
    resultado = uno + dos;
  }

  var resta = function(uno, dos) {
    resultado = uno - dos;
  }

  multiplicacion(valor1, valor2);
  return resultado;
}
```

Ilustración 10 Funciones JavaScript

### 1.1.2.5. Agregando código JavaScript en documento HTML

Existen varias maneras de agregar código JavaScript en un documento HTML, la primera es utilizando los atributos incorporados en el lenguaje HTML, atributos que manejan los eventos generados por el usuario entre los más utilizados se encuentran: onclick, onMouseOver, onMouseOut, onLoad, onFocus.



```
1 <!doctype html>
2 <html lang="es">
3   <head>
4     <title>Ejemplo uso JavaScript</title>
5   </head>
6   <body>
7     <button onclick="alert('Evento generado clic..!')">Hacer Clic</button>
8   </body>
9 </html>
```

Ilustración 11 JavaScript En Línea

En la imagen “JavaScript En Línea” se observa el uso del evento onclick(), el cual se ejecuta cuando el usuario de clic sobre el elemento en este caso el botón. Una vez presionado sobre el elemento JavaScript envía una alerta que se mostrará como una ventana pequeña en el navegador. Podemos cambiar el evento onclick() por cualquier de los eventos permitidos en el lenguaje HTML, simplemente agregando el nombre del atributo a utilizar. Esta es la manera de incorporar código JavaScript en línea.

Otra forma de agregar código JavaScript es mediante una estructura propia JavaScript, es decir agrupar todo el código en una sola sección organizada. Para esto podemos utilizar un elemento nativo <script> que viene incorporado en el lenguaje HTML. Para agregar la etiqueta <script> se recomienda introducirla dentro de la etiqueta <head>, de esta manera se tendrá organizado el código escrito (ver ilustración 12).

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4 <title>Ejemplo uso JavaScript</title>
5 <script>
6     function mostrarMensaje() {
7         alert("Mensaje alerta..!");
8     }
9     function hacer_clic() {
10        document.getElementsByTagName('button')[0].onclick = mostrarMensaje();
11    }
12    window.onload = hacer_clic();
13 </script>
14 </head>
15 <body>
16 <button>Hacer Clic</button>
17 </body>
18 </html>
```

Ilustración 12 Etiquetas <script>

#### 1.1.2.6. Accediendo a elementos HTML desde JavaScript

En el lenguaje JavaScript es posible acceder a los elementos HTML. Con este acceso se puede manipular: los valores, estados o atributos de los elementos creados en el código HTML, esto se le conoce como la manipulación del “DOM” (Document Object Model).

Para acceder a los elementos a ser manipulados, JavaScript contaba inicialmente con tres métodos, pero en la última actualización de este lenguaje se crearon 2 adicionales que potencia la calidad de JavaScript (Gauchat, 2012).

Los métodos se detallan a continuación:

- `getElementsByTagName`.- Busca los elementos HTML de acuerdo al nombre de la etiqueta.
- `getElementById`.- Busca un elemento HTML que coincida su atributo ID.
- `getElementsByClassName`.- Busca los elementos HTML de acuerdo al atributo “class” que tengan ingresado.
- `querySelector`.- Busca el primer elemento que concuerda con el filtro de búsqueda.
- `querySelectorAll`.- Busca todos los elementos que concuerdan con el filtro de búsqueda ingresado.

En el Anexo 2: JavaScript, métodos para acceder al DOM se encuentra un detalle de los métodos listados anteriormente.

#### 1.1.2.7. Agregando archivos JavaScript externos

Esta manera de enlazar código HTML con JavaScript es la más recomendable, ya que de esta forma se puede utilizar las funciones que se encuentra escritas, en cualquier documento HTML de un proyecto. Por ejemplo, se puede tener una función que multiplica dos valores y esta poder usar en diferentes documentos HTML.

Para agregar archivos externos se crear una etiqueta `<script>`, que se recomienda sea dentro de la etiqueta `<head>` y agregar el atributo "src" que especifica la dirección donde se encuentra el archivo JS y el nombre (Ver imagen 13).



```
1 <!doctype html>
2 <html lang="es">
3   <head>
4     <title>Ejemplo uso JavaScript</title>
5     <script src="validaciones.js"></script>
6   </head>
7   <body>
8     <button>Hacer Clic</button>
9   </body>
10 </html>
```

Ilustración 13 Archivos externos JavaScript

Esta práctica mejorara la velocidad de carga de la página web, además de proporcionar una estructura adecuada al documento JavaScript pensando en que esta puede crecer.

#### 1.1.3. AJAX

AJAX (Asynchronous JavaScript and XML) aparece en el año 2005, definido en ese momento de la siguiente forma "AJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes." (Garrett, 2005).

Después de algunos años AJAX es considerado como una técnica utilizada en el desarrollo web, con el propósito de transformar las aplicaciones o páginas web en

interactivas. Esto implica un trabajo mancomunado entre varias tecnologías web existentes en ese momento y el uso particular de un nuevo objeto llamado “XMLHttpRequest”, que permite una comunicación entre el cliente y el servidor de manera asíncrona.

La nueva forma de comunicación asíncrona soluciona el problema, hasta ese momento existente, de refrescar o recargar las aplicaciones cada que se necesitaba recuperar datos desde el servidor. La comunicación asíncrona permite recuperar información del servidor de manera oculta para el usuario, modificando las peticiones HTTP por peticiones JavaScript, teniendo así una interacción adecuada y mucho más rápida.

Todas las características mencionadas hacen que AJAX, sea una de las herramientas necesarias para el desarrollo de aplicaciones o páginas web. El contenido de esta sección es un resumen de las principales definiciones, propiedades y tecnologías de AJAX obtenidas de archivos cargados por (Mora), (Pérez, 2008).

Diferencia entre el método asíncrono y síncrono.

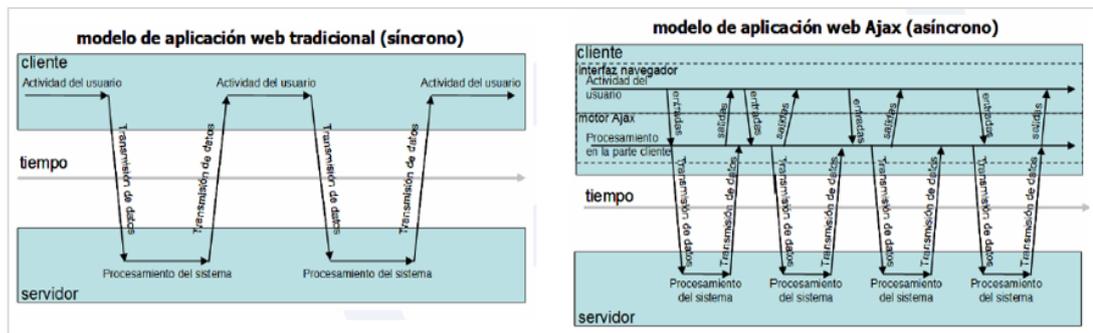


Ilustración 14 Diferencia método asíncrono y síncrono

### 1.1.3.1. Tecnologías aplicadas en AJAX

AJAX es la unión de varias tecnologías de desarrollo web, por lo que a continuación de detalla cuáles son los componentes utilizados (Mora).

Tecnologías usadas:

- XHTML y CSS.- La tecnología XHTML, una versión anterior a la nueva HTML5, con la ayuda de CSS fueron utilizadas para la presentación de la información.

- DOM.- Necesario para la manipulación dinámica de los elementos de la página, esto mediante el uso del lenguaje JavaScript.
- XML, XSLT / JSON.- XML usado para el intercambio de información entre el cliente y el servidor.
- XMLHttpRequest.- Objeto usado para la recuperación asíncrona de la información, estableciendo canales independientes en cada conexión con el servidor.
- JavaScript.- Lenguaje script utilizado para manipulación del DOM y transferencia de datos.

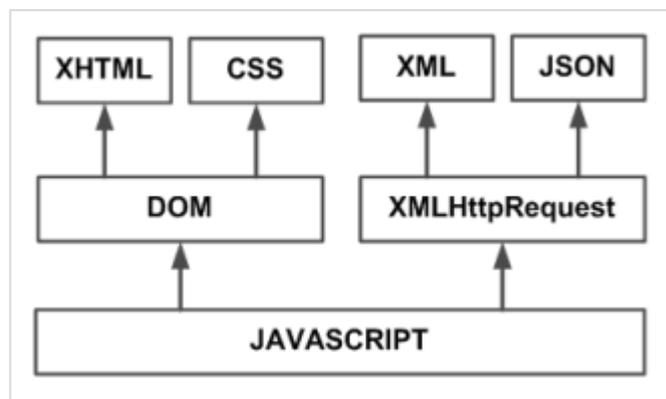


Ilustración 15 Tecnologías en AJAX

#### 1.1.3.2. XMLHttpRequest

Este objeto fue desarrollado en el año 2000, surge como respuesta a un problema que tenían en su momento con el desarrollo del famoso Exchange 2000, un servidor de correos. Este objeto posterior a su uso dentro de Exchange se implementó en el famoso buscador Internet Explorer y desde ese momento muchas tecnologías empezaron a utilizar este objeto (Pérez, 2008).

##### 1.1.3.2.1. Métodos Incluidos

El objeto XMLHttpRequest tiene algunos métodos que son usados para establecer conexiones con el servidor (Mora), en la siguiente tabla se detalla los métodos disponibles:

Método	Descripción
abort()	Detiene la petición que se ejecuta en el momento.
getAllResponseHeaders()	Devuelve una cadena de texto con todas las cabeceras de la respuesta del servidor
getResponseHeader(cabecera)	Devuelve una cadena de texto con el contenido de la cabecera solicitada.
open("método", "URL")	Establece los parámetros de la petición que se realiza al servidor. Los parámetros necesarios son el método HTTP empleado y la URL destino (puede indicarse de forma absoluta o relativa).
send("contenido")	Realiza la petición HTTP al servidor.
setRequestHeader("Cabecera", "Valor")	Permite establecer cabeceras personalizadas en la petición HTTP. Se debe invocar el método open() antes que setRequestHeader().

Tabla 1 Métodos XMLHttpRequest

Los métodos open, principalmente recibe dos parámetros (nombre del método, dirección URL del método), pero también se le puede incluir tres parámetros adicionales:

- asíncrono.- Parámetro booleano que especifica el tipo de solicitud que se realiza. El valor por defecto es verdadero (true), este valor indica que la solicitud será de asíncrona.
- usuario y contraseña.- Parámetros de tipo "string" que especifican las credenciales de autenticación necesarias para acceder al método invocado.

```
open(string metodo, string URL [,boolean asincrono, string usuario, string password]);
```

#### 1.1.3.2.2. Propiedades XMLHttpRequest

La siguiente tabla muestra las propiedades definidas para el objeto XMLHttpRequest (Mora).

Propiedad	Descripción
readyState	Valor numérico de tipo entero, que almacena el estado de la petición (Vea tabla valores permitidos).
responseText	Almacena el contenido de la respuesta del servidor en forma de cadena de texto.
responseXML	Almacena el contenido de la respuesta del servidor en formato XML. El objeto devuelto se puede manipular como un objeto DOM.
status	Contiene el código del estado HTTP devuelto por el servidor (200 para una respuesta correcta, 404 para "No Encontrado", 500 para un error del servidor, etc.).
statusText	Contiene el código del estado HTTP devuelto por el servidor en forma de cadena de texto. Ejemplo: "OK", "Not Found", "Internal Server Error", etc.

Tabla 2 Propiedades XMLHttpRequest

Valores permitidos en la propiedad readyState:

Valor	Descripción
0	No inicializado (objeto creado, pero no se ha invocado el método open).
1	Cargando (objeto creado, pero no se ha invocado el método send).
2	Cargado (se ha invocado el método send, pero el servidor aún no ha respondido).
3	Interactivo (se han recibido algunos datos, aunque no se puede emplear la propiedad responseText).
4	Completo (se han recibido todos los datos de la respuesta del servidor).

Tabla 3 Valores Propiedad readyState

### 1.1.3.3. Ejemplo uso AJAX desde JavaScript

En el siguiente ejemplo se muestra el uso del objeto AJAX, mismo que se programa desde el código JavaScript.

```

$.ajax({
  url: '@Url.Action("consultaIdEmpleadoLogXIdPersona","Funciones")?idPersona=' + $('#idUsuaioLog').val(),
  async: true,
  success: function (info) {
    idEmpleSuc = info;
  }
});

```

Ilustración 16 Método AJAX desde JavaScript

En el ejemplo, la palabra “info” es el dato devuelto por el método invocado, mismo que para el ejemplo se almacena en una variable.

#### 1.1.4. jQuery

jQuery es una biblioteca gratuita, o también llamado framework, basado en el lenguaje JavaScript. Este framework fue desarrollado inicialmente por John Resig, pero posterior a su primera versión se fueron incorporando nuevos desarrolladores hasta que actualmente son todo un grupo de colaboradores lo que están atrás de este framework y que constantemente están cargando nuevas actualizaciones (Point, Tutorials Point, s.f.).

El objetivo de un framework, como jQuery, es simplificar el trabajo. En este caso facilita la manera de interactuar con los elementos creados en los documentos HTML, también facilita la manera de interactuar con el árbol DOM, manejar eventos, desarrollar animaciones y principalmente manipular interacciones con la técnica AJAX. Es por todo lo detallado que el eslogan de este framework es “Write less, do more”, es decir se puede hacer mucho más con menos código.

En esta sección se abordará de manera resumida el framework, proporcionando así una sólida base sobre el uso de jQuery. Además de brindar la capacidad para seguir con una investigación profunda sobre este tema.

El contenido de esta sección aborda los aspectos principales y fundamentos necesarios para empezar a trabajar con la librería jQuery, publicados en el sitio web (Point, Tutorials Point, s.f.).

##### 1.1.4.1. Agregando jQuery

Existen dos formas de agregar la librería a una aplicación o página web:

- Descargar librería jQuery.- se puede descargar la librería desde la página oficial de jQuery (<https://jquery.com/download/>) y posterior agregar al documento HTML usando la etiqueta <script>.

```
<script type="text/javascript" src="/jquery/jquery-2.1.3.min.js"></script>
```

Ilustración 17 Agregando librería jQuery

Se recomienda descargar la versión mimetizada, esta versión descargará un solo archivo JavaScript, que permitirá tener una mayor rapidez de carga de la página web gracias a su tamaño reducido.

- Librería online.- Esta opción permite trabajar directamente con un documento que está en línea, cargado bajo la red de contenido de desarrollo (CDN). Este repositorio almacena las principales librerías que se utilizan a nivel mundial, para el desarrollo de software en general.

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
```

Ilustración 18 Agregando librería jQuery Opción 2

Adicional a las librerías básicas de jQuery, este framework también incorpora otros módulos como la versión UI. Este módulo proporciona una serie de temas desarrollados por los colaboradores de jQuery. Estos temas brindan estilos diferentes para los elementos, animaciones u objetos que dispone la librería. Para descargar dichos temas se puede navegar directamente a la página oficial (<http://jqueryui.com/download>).

#### 1.1.4.2. Inicializar página

Para interactuar de forma segura con los elementos agregados en el documento web, es necesario enterarse cuando ya termina de cargar por completo la página. jQuery tiene incorporado un bloque “\$(document).ready()”, el cual se ejecuta cuando y solo cuando la página se haya terminado de cargar, asegurando así el no tener errores al momento de manipular el contenido de la página.

```
<script type="text/javascript">
  $(document).ready(function () {
    alert("Documento cargado completamente..");
  });

  $(function () { //Forma abreviada
    alert("Documento cargado completamente..");
  });
</script>
```

Ilustración 19 Método inicializa jQuery

En la imagen anterior se puede observar cómo utilizar el bloque “ready”, en el cual se puede agregar código JavaScript. Se observa también que el código JavaScript está dentro de la etiqueta <script>, siempre que se necesites agregar código usando jQuery debe hacerlo de esa forma.

#### 1.1.4.3. Selección de Elementos

La librería jQuery está desarrollada con el poder de soportar gran parte de los selectores que maneja CSS3, además de algunos más que no están estandarizados. Esta propiedad es la más significativa de esta librería, y por la que lleva su nombre, nombre que significa “permitir seleccionar algunos elementos y realizar acciones con ellos” (jQuery, s.f.).

jQuery maneja principalmente tres tipos de selectores o también llamados consultas:

- Consultas CSS
- Consultas XPath
- Consultas Transversales

En el Anexo 3: Selectores de Elementos se detalla cada uno de los selectores que dispone jQuery.

#### 1.1.4.4. Atributos y Propiedades jQuery

jQuery permite la manipulación de atributos y propiedades de los elementos del DOM. Aunque con el uso de JavaScript se puede realizar casi la misma manipulación de los atributos, jQuery proporciona una forma más útil y sencilla.

Entre los atributos que más se manipulan están:

- className
- tagName
- id
- href
- title
- rel
- src

#### 1.1.4.4.1. Obtener el valor de los atributos

jQuery utiliza el método “attr()” para obtener el valor de los atributos que puede tener un elemento cualquiera del DOM.

```
$(document).ready(function () {  
    var valor = $('#cajal').attr('title');  
});
```

Ilustración 20 Método attr()

En la imagen anterior se observa cómo obtener el valor de un atributo, en este caso el atributo title, y se la almacena en una variable que posteriormente se la puede utilizar.

#### 1.1.4.4.2. Establecer valores a los atributos

Para otorgar un valor a un atributo de un elemento del documento HTML, jQuery utiliza el mismo método “attr()”, usado para obtener el valor del atributo. Pero para otorgar el valor debemos incluir dos parámetros: el nombre del atributo y el valor a establecer.

```
$(document).ready(function () {  
    $('#cajal').attr('value', '520');  
});
```

Ilustración 21 Método attr() asignación

En el ejemplo se estableciendo un valor de 520 a una caja de texto, para eso se utiliza el atributo “value”.

#### 1.1.4.4.3. Manipulando atributo de clases

Con jQuery también existe la posibilidad de manipular los estilos o clases de los elementos del DOM. Para agregar una clase a un elemento se utiliza el método “addClass()”. Este evento brinda la posibilidad de agregar varias clases al mismo tiempo, separando cada una con un espacio.

```
$(document).ready(function () {  
    $('#cajal').addClass('estilo');  
});
```

Ilustración 22 Método addClass()

#### 1.1.4.5. Transversalidad del DOM

jQuery proporciona métodos adicionales, a los ya vistos anteriormente, para mejorar la selección de elementos de un documento HTML. Estos métodos son usados principalmente para buscar elementos mas no son utilizados para manipular los objetos del DOM.

En el Anexo 3: Métodos se encuentra un listado de los métodos más usados para la manipulación de objetos.

#### 1.1.4.6. Manipulación del DOM

jQuery viene incorporado con múltiples métodos que permiten manipular de manera eficiente y rápida el DOM. Reduciendo con esto una gran cantidad de código, que antes con otros lenguajes era necesario. Algunos de los métodos detallamos pueden ser los mismos que se utiliza para manipular atributos de los elementos del DOM.

En el Anexo 3: Métodos se incorpora una tabla que lista todos los métodos disponibles en jQuery. Métodos que permiten manipular el árbol DOM, mostrando también un ejemplo que ayude a entender cada método.

#### 1.1.4.7. Manejo de Eventos

jQuery incorpora una lista de eventos y funciones que son usados para la administración de los eventos que pueden generarse en la interacción con el documento o aplicación. Es decir, con jQuery tenemos la posibilidad de cambiar la acción que puede realizar un evento. Por ejemplo, si tenemos un evento mouse clic podemos cambiar para que en vez de señalar un elemento este elimine el elemento. Así podremos tener un control absoluto de las acciones que pueda realizar nuestro documento o aplicación.

En la siguiente imagen vemos la sintaxis por defecto para declarar un manejador de eventos:

```
$('#Selector').nombre_Evento(function() {  
    //Función que administra un evento  
    //this es el elemento que dispara el evento  
});
```

Ilustración 23 Manejadores de eventos jQuery

jQuery proporciona la posibilidad de cambiar el comportamiento del evento después de haber sido disparado. Esto es posible agregando un parámetro llamado “event”, como se muestra en la imagen siguiente, para posterior poder manipular el comportamiento.

```
$('#Selector').nombre_Evento(function(evento) {  
    evento.preventDefault();  
});
```

Ilustración 24 Manipular comportamiento eventos jQuery

Otra funcionalidad de jQuery es la posibilidad de agregar múltiples funciones sobre un mismo evento que aplica a un elemento. Esta funcionalidad se la conoce como comportamiento en cola.

```
$('#Selector')  
    .nombre_Evento(function() {  
    })  
    .nombre_Evento(function() {  
    });
```

Ilustración 25 Comportamiento en cola jQuery

En el Anexo 3: Tipos de eventos. se encuentra un detalle de los eventos más utilizados en el desarrollo web.

#### 1.1.4.8. Uso AJAX con jQuery

Como se ha visto en la sección propia de AJAX, este es un estándar para la obtención de datos desde el servidor. jQuery es una buena herramienta que ayuda a interactuar con el método AJAX facilitando su uso.

##### 1.1.4.8.1. Carga de Datos Simple

jQuery provee un método simple para la carga de datos, sea de manera estática o de manera dinámica.

Sintaxis:

**[ selector ].load( URL, [datos], [función])**

Parámetros:

- **URL.**- Dirección donde se encuentra el recurso que accede el request.
- **data.**- Parámetro opcional que puede ser un objeto plano o una cadena, que será enviada al servidor mediante la llamada request.
- **función.**- Parámetro opcional, el cual almacena una función que será invocada cuando regresa de la llamada request.

Ejemplo de su uso.

```
$("#div").click(function() {  
    $('#boton').load('/jquery/result.html', function() {  
        alert("Llamada terminada.");  
    });  
});
```

Ilustración 26 Método Load

#### 1.1.4.8.2. Obteniendo datos tipo JSON

Este método es utilizado cuando las llamadas al servidor devuelven un objeto tipo JSON. Este objeto puede contener varios valores que únicamente los podemos acceder mediante llamadas tipo JSON.

El método que dispone jQuery, para obtener valores tipo JSON, es `getJSON()`, cuya sintaxis es la siguiente:

**[ selector ].getJSON( URL, [ datos ], [ función ])**

Parámetros:

- URL.- Dirección donde se encuentra el recurso al cual se va a invocar mediante el método GET.
- datos.- Parámetro opcional que puede ser un objeto plano o una cadena, que será enviada al servidor mediante la llamada request.
- función.- Parámetro opcional, el cual almacena una función que será invocada cuando regresa de la llamada request.

Ejemplo de su uso:

```
$("#div").click(function(){
    $.getJSON('/jquery/result.json', function(jd) {
        $('#stage').html('<p> Name: ' + jd.nombre + '</p>');
        $('#stage').append('<p>Age : ' + jd.edad + '</p>');
        $('#stage').append('<p> Sex: ' + jd.genero + '</p>');
    });
});
```

Ilustración 27 Método getJSON

Ejemplo usando AJAX:

```
$.ajax({
    dataType: "json",
    url: url,
    data: data,
    success: success
});
```

Ilustración 28 Método AJAX tipo JSON

#### 1.1.4.8.3. Método AJAX en jQuery

jQuery trae consigo una serie de métodos que permiten hacer llamadas asíncronas hacia el servidor. A continuación se lista los métodos principales, el detalle de los métodos se encuentra en el Anexo 3: Métodos.

- \$.ajax
- \$.ajaxSetup
- \$.get
- \$.getJSON
- \$.getScript
- \$.post
- serialize
- serializeArray

#### 1.1.5. CSS3

CSS es un lenguaje de estilos que manipula la presentación o interfaz gráfica de los documentos web, principalmente HTML. Este lenguaje nace de la necesidad de mejorar la estructura en el desarrollo web clasificando el contenido de la presentación de un documento.

El objetivo principal del lenguaje CSS, es buscar la forma de visualizar de diferente manera el contenido, de una aplicación o sitio web, en los diferentes dispositivos en lo que son abiertos. Por ejemplo, una aplicación será visualizada de diferente manera en una computadora de escritorio como en una tableta o un teléfono celular (Puig, 2015).

CSS como los demás lenguajes ha evolucionado en el transcurso de su historia, desde el año 1996 que surgió su primera edición que principalmente se enfocó en unificar la sintaxis de una hoja de estilos. Para el año de 1998 surge la segunda edición conocida como CSS2, la cual tuvo gran aceptación y acogida por todos los navegadores hasta ese entonces existentes.

En el año 2005 comienza el desarrollo de la nueva versión conocida como CSS3 la cual fue integrada a los navegadores de acuerdo al desarrollo de cada uno de ellos,

es decir cada navegador busco agregar nuevas especificaciones de CSS3 de acuerdo a su necesidad. Uno de los trascendentales problemas que tiene esta versión hasta la actualidad, es que no todas las características incorporadas en la última versión de CSS están disponibles en todos los navegadores, por lo que hay que tener cuidado de esto al momento de realizar el diseño.

La novedad de la tercera versión frente a sus antecesoras, es que esta incorpora una clasificación de sus características. Esta clasificación se realiza en temas como por ejemplo: colores y gamas, fuentes, lenguajes verticales, modelo de caja, modelo de línea, page-media, selectores, unidades de medida, etc. La clasificación por temas brinda, a más de un tema de organización, la posibilidad de un crecimiento organizado, es decir cada tema podrá evolucionar en paralelo sin causar trabas o problemas con el resto de temas.

Como muchos otros lenguajes, W3C (World Wide Web Consortium) maneja las especificaciones de CSS, por lo que si es necesario ahondar en algún tema se puede recurrir a la página web.

#### 1.1.5.1. Beneficios CSS3

CSS3 se transformó en un estándar utilizado para la maquetación de aplicaciones o páginas web, a continuación se detalla cuáles son los principales beneficios por lo que este lenguaje llegó a tener gran aceptación según (Puig, 2015, pág. 10).

- Incremento del rendimiento.- El uso de menos etiquetas HTML indican una menor cantidad de código para descargarse del servidor y menos código a la hora de interpretar y dibujar el navegador. Esta reducción implica un menor consumo del ancho de banda y un mejor rendimiento del ordenador.
- Reducción de tiempo.- Con el uso del lenguaje CSS y especialmente con la versión CSS3, se modifica la arquitectura en el desarrollo, teniendo ahora una mejor estructura de las aplicaciones y por ende un mejor tiempo al momento del desarrollo y en cualquier modificación.

#### 1.1.5.2. Incorporaciones CSS3

La última versión de CSS3 trae consigo una serie de novedades en cuanto a nuevos selectores, el uso de colores, nuevos atributos que dan transparencia a los elementos y una serie de nuevas propiedades. Entre las propiedades que más han gustado están la de los bordes redondeados y la utilización de media queries.

El detalle de todas las nuevas incorporaciones se detalla en el Anexo 4: CSS3, Nuevas Funciones, Propiedades, Transiciones y Selectores. Ahí se muestra su uso con algunos ejemplos.

#### 1.1.6. Bootstrap

Bootstrap es un framework desarrollado por los creadores de la muy conocida red social Twitter y publicado en el año 2011. Este framework está desarrollado bajo los últimos estándares de desarrollo web como: HTML5, CSS3 y jQuery.

El objetivo de Bootstrap es brindar una herramienta que permita, a los desarrolladores de software, diseñar páginas o aplicaciones web que sean visibles por los usuarios en cualquier dispositivo electrónico como computadoras de escritorio, tabletas, computadoras portátiles y dispositivos móviles. Esta funcionalidad es conocida como diseño adaptable o responsive que pretende que con un solo diseño web se tenga una visualización adecuada en cualquier dispositivo.

La tecnología en la que se base Bootstrap, para hacer un diseño adaptable, es crear un avanzado sistema de rejillas, mismas que están divididas en columnas para un posicionamiento de los elementos. Esto permite especificar la posición de un elemento según el dispositivo que abra la página (Sánchez, 2015).

Importante también es recalcar que es un framework sencillo y ligero, basta con agregar un archivo CSS y otro JavaScript para poder comenzar a utilizar Bootstrap. Además que Bootstrap es compatible con todos los navegadores habituales (Google Chrome, Firefox, Opera, Safari, Internet Explorer), eliminando la preocupación de las funcionalidades compatibles.

El contenido de esta sección es una breve recopilación sobre los principales funciones y la forma de incorporar Bootstrap al diseño de sitios web tomada del libro (Sánchez, 2015)

### 1.1.6.1. Estructura Bootstrap

En el siguiente gráfico se observa los archivos que contiene el paquete de Bootstrap, mismo que puede ser descargado desde su página oficial.

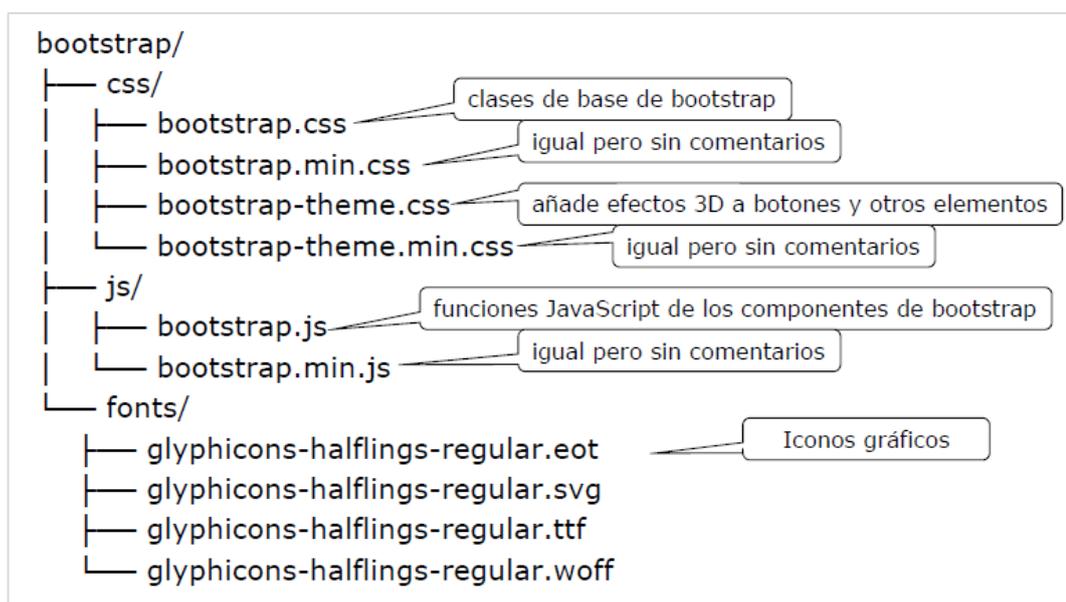


Ilustración 29 Estructura Bootstrap

El contenido de Bootstrap se clasifica en tres elementos principales: CSS, JavaScript y fonts. Las dos primeras carpetas contienen archivos que hacen referencia a su nombre y en la última carpeta se almacena archivos que guardan los gráficos de los íconos usados en este framework.

Se recomienda incluir solo los archivos mimetizados para los contenidos CSS y JavaScript, con esto tendrá una carga más ligera del framework.

### 1.1.6.2. Agregando Bootstrap

Para empezar a utilizar el framework Bootstrap en aplicaciones o páginas web, primero es necesario incluir algunas opciones de HTML5, mismas que son necesarias para el correcto funcionamiento de Bootstrap.

Añadir el elemento "doctype":

```
<!DOCTYPE html>
<html lang="es">
...
</html>
```

Ilustración 30 Agregando elemento DOCTYPE

Añadir la etiqueta <meta>, esta línea permite que la página se muestre correctamente en dispositivos móviles, además de permitir especificar que al contenido se le pueda realizar un acercamiento (zoom).

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```

Ilustración 31 Agregar elemento <meta>

Ya con agregar las especificaciones anteriores se puede incorporar el script y la hoja de estilos de Bootstrap. En la siguiente imagen se puede observar una estructura básica de una página web con Bootstrap.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Plantilla básica de Bootstrap</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/jquery-2.1.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </head>
  <body>
    <h1>¡Hola Mundo!</h1>
  </body>
</html>
```

Ilustración 32 Estructura Básica con Bootstrap

Es importante mencionar que para comenzar con el uso del framework Bootstrap es necesario agregar previamente la librería jQuery.

### 1.1.6.3. Sistema de Rejilla

Bootstrap está basado en un sistema de rejillas, la cual se basa en la creación o disposición del contenido de nuestra web dentro de rejillas flexibles, mismas que se escalarán al tamaño y posición adecuada de forma automática dependiendo del tamaño de la pantalla en la que se renderice.

#### 1.1.6.3.1. Elemento Contenedor

Para poder usar el sistema de rejillas de Bootstrap, es necesario agregar un elemento contenedor (“container” o “container-fluid”) que provee el framework. Estos elementos son considerados como elementos raíz para el uso de rejillas por lo que son imprescindibles.

Al usar el elemento “.container”, el contenido de la página se mostrará centrado y con un ancho específico.

```
<div class="container">  
...  
</div>
```

Ilustración 33 Elemento Container Bootstrap

Por otro lado si es necesario que el contenido de la página ocupe todo el ancho de la pantalla entonces puede usar el elemento “.container-fluid”.

```
<div class="container-fluid">  
...  
</div>
```

Ilustración 34 Elemento Container-fluid Bootstrap

#### 1.1.6.3.2. Funcionamiento rejilla

El sistema de rejilla está dispuesto para manipular el contenido de la página web y su adaptación a los diferentes tamaños de pantalla de forma automática. Este sistema divide la pantalla en 12 columnas que estarán incorporadas dentro de filas (.row). El número de columnas que se ocupa en la pantalla puede ser personalizado, por lo que se puede decidir si la pantalla ocupará n número de columnas.

Consideraciones al momento de utilizar la rejilla:

- Las columnas siempre irán agrupadas dentro de filas (.row).
- Cada fila debe tener un máximo de 12 columnas.
- Si la cantidad de columnas excede de 12, la cantidad sobrante se coloca en la siguiente fila.
- El tamaño de las columnas se especificará con clases css que Bootstrap define para cada tamaño de pantalla, por ejemplo “.col-md-XX”, donde “XX” es el tamaño de la columna, que podrá tomar valores entre 1 y 12.

Bootstrap maneja 4 prefijos, que especifica el tipo de dispositivo, al cual aplicará el diseño adaptable. Esto no quiere decir que siempre se tenga que especificar el prefijo para cada dispositivo, ya que si el prefijo para el dispositivo no está descrito, Bootstrap asumirá el prefijo del dispositivo de tamaño inferior. Es decir las pantallas más grandes heredan los prefijos de las inferiores. En la siguiente tabla se detalla cuáles son los prefijos.

<b>Pantalla</b>	<b>Prefijo</b>	<b>Ancho</b>
Tamaño extra pequeño Teléfonos (<768px).	.col-xs-	Automático
Tamaño pequeño Tabletas (>= 768px).	.col-sm-	750px
Tamaño medio Escritorio (>= 992px).	.col-md-	970px
Tamaño grande Escritorio (>= 1200px).	.col-lg-	1170px

Tabla 4 Prefijos Bootstrap

1.1.6.3.3. Ejemplos de su uso:

En el siguiente ejemplo HTML se crea tres filas con diferentes cantidades de columnas. Para este caso se aplica un estilo para dispositivos con pantallas medianas (PC, Laptop), es decir en dispositivos más pequeños se acoplara de acuerdo a la pantalla, sin garantizar un buen diseño.

```

<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>

```

Ilustración 35 Formato Bootstrap para Laptops

En la siguiente imagen se ilustra el resultado al abrir nuestra página en el navegador.

.col-md-8	.col-md-4	
.col-md-4	.col-md-4	.col-md-4
.col-md-6	.col-md-6	

Ilustración 36 Visualización Bootstrap en Laptops

Como se menciona al principio, este diseño se aplica solo para pantallas medianas, por lo que si se abre la página en un dispositivo móvil, este se visualizará un elemento en cada fila. Ahora según los prefijos que ya se detallaron anteriormente, nos es fácil intuir que se tiene que hacer para poder adaptar nuestro diseño a las tabletas y teléfonos móviles. Lo que se necesita es usar las clases “col-xs-” y “col-sm-”, si se quiere adaptar a pantallas mucho más grandes simplemente se agrega la clase “col-lg-”, pero en esta sección no se abordará dicho formato.

```

<div class="container">
  <div class="row">
    <div class="col-xs-9 col-sm-8 col-md-8 b">.col-sm-8</div>
    <div class="col-xs-3 col-sm-4 col-md-4">.col-md-4</div>
  </div>
  <div class="row">
    <div class="col-xs-4 col-sm-6 col-md-4 b">.col-md-4</div>
    <div class="col-xs-4 col-sm-6 col-md-4 b">.col-md-4</div>
    <div class="col-xs-4 col-sm-12 col-md-4">.col-md-4</div>
  </div>
  <div class="row">
    <div class="col-xs-6 col-sm-6 col-md-6 b">.col-md-6</div>
    <div class="col-xs-6 col-sm-6 col-md-6">.col-md-6</div>
  </div>
</div>

```

Ilustración 37 Formato Bootstrap Móviles, Tablet y Laptops

En la imagen anterior se ilustra el uso de los prefijos para tres tipos de dispositivos, en cada dispositivo se puede personalizar el diseño. Por lo que no siempre se tendrá pantallas iguales de las páginas, en cada dispositivo que se abra.

A continuación, se ilustra el resultado de la página en los dos dispositivos faltantes:

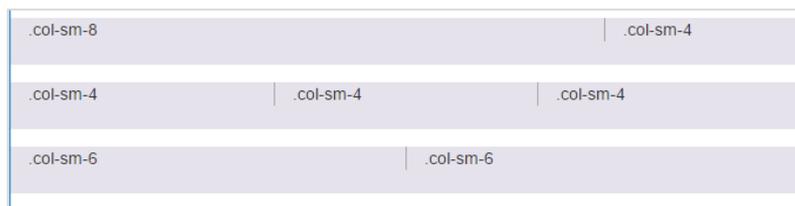


Ilustración 38 Bootstrap Tabletas

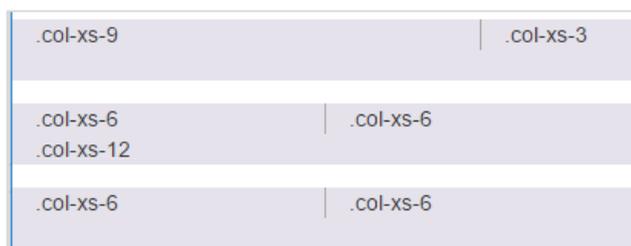


Ilustración 39 Bootstrap Móviles

#### 1.1.6.3.3.1. Anidamiento de columnas

Esta característica permite agregar columnas dentro de otras creando así nuevas filas, con distribuciones distintas si así lo necesitan. En el siguiente ejemplo se ilustra como una columna aloja una nueva fila, misma que consta de dos columnas.

```

<div class="container">
  <div class="row">
    <div class="col-sm-9">
      Nivel 1: .col-sm-9
      <div class="row">
        <div class="col-xs-8 col-sm-6">
          Nivel 2: .col-xs-8 .col-sm-6
        </div>
        <div class="col-xs-4 col-sm-6">
          Nivel 2: .col-xs-4 .col-sm-6
        </div>
      </div>
    </div>
  </div>
</div>

```

Ilustración 40 Anidamiento de columnas Bootstrap



Ilustración 41 Visualización Navegador Anidamiento Bootstrap

### 1.1.6.3.3.2. Márgenes entre columnas

Esta característica es una de las más utilizadas y funcionales que tiene Bootstrap. Permite establecer espacios entre columnas, dejando lugares vacíos en la rejilla. La característica se agrega mediante el siguiente atributo “col-prefijo-offset-\*”.

Este atributo se puede incluir para cada prefijo que tiene Bootstrap, por lo que si se quiere se podría agregar 4 atributos “offset”. El valor posterior es el número de columnas que se dejarán en blanco en la rejilla.

En el siguiente ejemplo se indica el uso del atributo “offset”.

```

<div class="container">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>
  </div>
  <div class="row">
    <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
    <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
  </div>
  <div class="row">
    <div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>
  </div>
</div>

```

Ilustración 42 Atributo offset Bootstrap

Vista en el navegador:

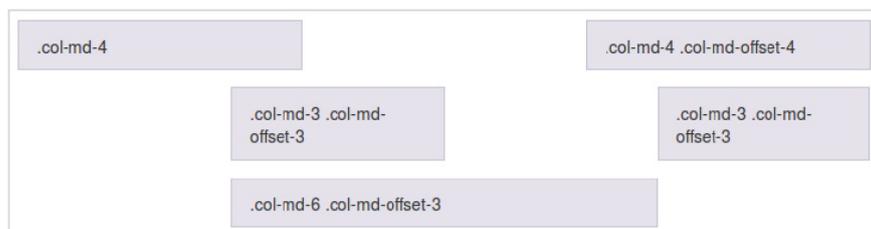


Ilustración 43 Visualización atributo offset Bootstrap

### 1.1.6.3.3. Ordenación de Columnas

Esta propiedad o atributo permite modificar el orden de las columnas de la rejilla Bootstrap. Para agregar esta funcionalidad se usa el atributo “.col-prefijo-push-\*” para empujar “n” espacios hacia la derecha o el atributo “.col-prefijo-pull-\*” para empujar “n” espacios hacia la izquierda.

En el siguiente ejemplo se ilustra el uso de estos atributos:

```
<div class="container">
  <div class="row">
    <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>
    <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>
  </div>
</div>
```

Ilustración 44 Atributo Push - Pull Bootstrap

Vista en el navegador:



Ilustración 45 Visualización atributo push - pull Bootstrap

Es importante recalcar que estos atributos tienen inconvenientes si existe un cambio de fila, por lo que la visualización no será la adecuada.

### 1.1.6.4. Utilidades responsive

Bootstrap viene incorporado con atributos que le permiten mostrar u ocultar contenido de acuerdo al dispositivo en el que sea abierto. Estos atributos se usarán seguidos por cualquiera de los prefijos que ya se mencionaron anteriormente.

En la siguiente tabla se detalla los atributos que se utilizan y en que dispositivo afecta:

	<b>Teléfonos (&lt;768px)</b>	<b>Tabletas (&gt;=768px)</b>	<b>Escritorio (&gt;=992px)</b>	<b>Escritorio (&gt;=1200px)</b>
.visible-xs-*	Visible	Oculto	Oculto	Oculto
.visible-sm-*	Oculto	Visible	Oculto	Oculto
.visible-md-*	Oculto	Oculto	Visible	Oculto
.visible-lg-*	Oculto	Oculto	Oculto	Visible
.hidden-xs	Oculto	Visible	Visible	Visible
.hidden-sm	Visible	Oculto	Visible	Visible
.hidden-md	Visible	Visible	Oculto	Visible
.hidden-lg	Visible	Visible	Visible	Oculto

Tabla 5 Atributos Adaptables Bootstrap

Algo importante en el uso del atributo “visible” es la posibilidad de incluir, al final, el tipo de visualización que queremos. Los tipos admitidos son los mismos con los que trabaja CSS (block, inline, inline-block).

### 1.3. SQL Server 2016 Express

Microsoft SQL Server es un sistema de administración y análisis de base de datos relacionales que provee organización de la infraestructura, aplicaciones y datos a los clientes en una sola plataforma. La nueva versión 2016 se basa en las funciones críticas ofrecidas en su versión anterior, proporcionando un rendimiento, una disponibilidad y una facilidad de uso para las aplicaciones más importantes. Esta nueva versión trae consigo nuevas capacidades en memoria en la base de datos, utilizada principalmente para transacciones en línea (OLTP) y el almacenamiento de datos (Varga, Cherry, & Joseph, 2016).

#### 1.2.1. Base de datos relacionales.

SQL Server usa el estándar SQL (Lenguaje de consulta estructurada) como la base de su tecnología. Dicho lenguaje apoya la creación y mantenimiento de la base de datos relacional y la gestión de los datos dentro de ella. Las bases de datos relacionales se basan en tablas, que son un conjunto de filas y columnas que representan una entidad única, creadas con el uso de la técnica de la normalización.

Las entidades de la base de datos se unen con la creación de relaciones, mismas que pueden ser de diferentes tipos, de acuerdo a la asociación de las entidades.

Tipos de relaciones:

- Una a una.
- Una a varias.
- Varias a varias.

#### 1.2.2. Tipos de instrucciones SQL

SQL separa las instrucciones de acuerdo a la función que realiza cada una. Según lo mencionado SQL separa en tres tipos de instrucciones:

- Lenguaje de definición de datos (DDL).- Las instrucciones de tipo DDL permiten crear, modificar o borrar objetos (tablas, vistas, esquemas) en una base de datos. Las principales palabras utilizadas en este tipo de instrucción son: CREATE,

DROP y ALTER, mismas que se utilizan, según mismo orden detallado, para crear, eliminar y modificar objetos.

- Lenguaje de control de datos (DCL).- Las instrucción de tipo DCL permite controlar quién o qué tiene acceso a los objetos de la base de datos. Permitiendo agregar o restringir el acceso con las instrucciones GRANT o REVOKE.
- Lenguaje de manipulación de datos (DML).- Las instrucciones de tipo DML permiten agregar, modificar, borrar o recuperar datos almacenados en los objetos de la base de datos. Las instrucciones usadas principalmente en este tipo son SELECT, INSERT, UPDATE y DELETE.

### 1.2.3. Características SQL Server 2016

La nueva versión 2016 de SQL Server incorpora las siguientes características:

- Procesamiento de transacciones en línea (OLTP).
- Ampliación de memoria a SSD.
- ColumnStore actualizable en memoria.
- Características AlwaysOn.
- Respaldo cifrado.
- Administración de recursos E/S.
- Respaldo Smart.
- SQL XI (Integración con XStore).

### 1.3. ASP.NET

ASP por sus siglas en inglés (Active Server Pages) fue introducido en el año de 1996, exactamente en el mes de diciembre, por la compañía Microsoft. El principal objetivo de esta tecnología era poder crear páginas web dinámicas, ya que para ese entonces se disponía simplemente de páginas HTML estáticas. Microsoft en la versión oficial 3.0 de ASP definió a esta tecnología como “Páginas abiertas de servicio activo, de compilación libre el cual permite combinar HTML, scripts, y reusar componentes ActiveX para crear”. ASP es soportado por todos los servidores web de Microsoft, por lo que levantar páginas web con esta tecnología no tiene problemas (Sae-Chin, 2008).

ASP.NET es un entorno de desarrollo Web, que surge de la evolución de ASP. Esta fue introducida en el año 2002 con su nueva tecnología .NET Framework. Este entorno fue desarrollado bajo el soporte de la tecnología Common Language Runtime (CLR), misma que nos permite codificar en cualquiera de los lenguajes que los soportan, por ejemplo: Microsoft Visual Basic, C#, JScript, J# (Point, Learn ASP.NET web application framework, 2014).

El entorno ASP.NET contempla la posibilidad de desarrollar formularios web, aplicaciones MVC y páginas web ASP.NET, cada una de estas realizadas en frameworks diferentes. El desarrollo ASP.NET permite el uso de varias tecnologías como HTML, CSS y JavaScript.

El siguiente contenido es una breve recopilación sobre la tecnología ASP.NET obtenida de (Point, Learn ASP.NET web application framework, 2014).

La siguiente imagen ilustra la arquitectura del entorno ASP, desde la solicitud de llamada HTTP hasta la conexión con la base de datos.

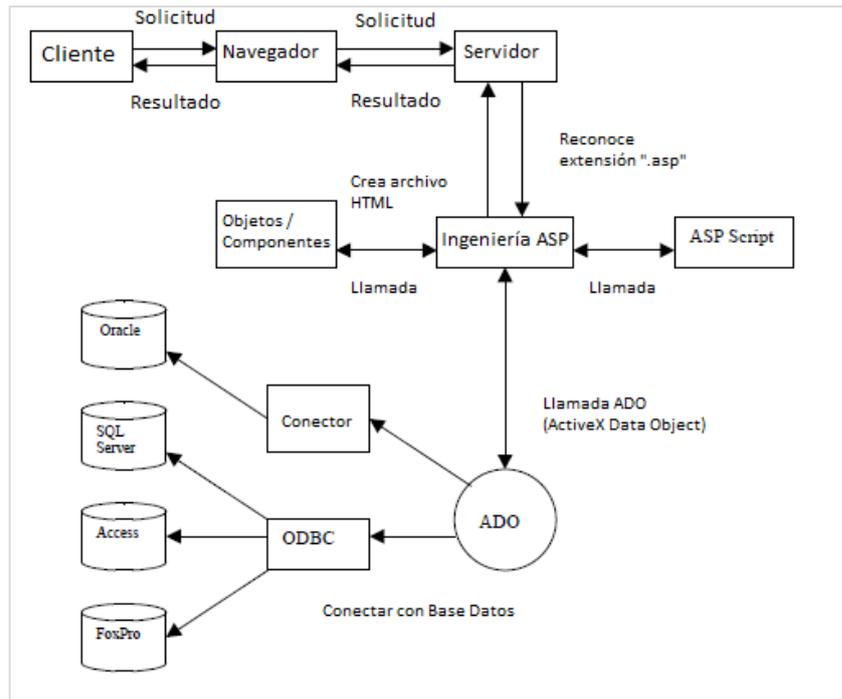


Ilustración 46 Arquitectura ASP.NET

### 1.3.1. Componentes de ASP.NET

- Marco de trabajo de página y controles.
- Compilador de ASP.NET.
- Infraestructura de seguridad.
- Funciones de administración de estado.
- Configuración de la aplicación.
- Supervisión de estado y características de rendimiento.
- Capacidad de depuración.
- Marco de trabajo de servicios Web XML.
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones.
- Entorno de diseñador extensible.

#### 1.3.1.1. Marco de trabajo de página y controles

El marco en ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor web, el cual permite generar y representar de forma dinámica páginas web ASP.NET. El desarrollo en ASP.NET está basado en un entorno orientado a objetos,

por lo que en las páginas web se puede trabajar con elementos HTML que usen eventos, propiedades y métodos. El marco de trabajo de páginas ASP.NET quita los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página.

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Además proporciona funciones, a través de temas o máscaras, para controlar la apariencia y funcionamiento del sitio web. Otro de los aspectos principales del marco de trabajo es la posibilidad de definir páginas principales y acompañarlas con páginas de contenido individual, teniendo así un diseño estándar para la aplicación creada (Microsoft, s.f.).

#### 1.3.1.2.        Compilador de ASP.NET

ASP.NET incluye un compilador que recoge todo el código ASP.NET, sus componentes, páginas y controles permitiendo optimizar el rendimiento y tener un enlace en tiempo de compilación.

#### 1.3.1.3.        Infraestructura de seguridad

La seguridad en ASP.NET es robusta gracias a que maneja una infraestructura avanzada para el acceso y autenticación de usuarios ayudada también por las características de seguridad propias de la tecnología .NET. Para la autenticación de usuarios tenemos dos opciones:

- Autenticación de Windows.
- Autenticación de la base de datos.

ASP.NET se ejecuta con una identidad particular de Windows de modo que puede asegurar su aplicación utilizando las capacidades de Windows como, por ejemplo, las listas de control de acceso (ACL) de NTFS, permisos de la base de datos, etc.

#### 1.3.1.4. Funciones de administración de estado

ASP.NET brinda funciones de estado intrínsecas y distribuidas. Las funciones intrínsecas permiten almacenar información entre las solicitudes de página. Entre la información que se puede guardar esta información de la aplicación, información de la sesión, información de la página, información del usuario y finalmente información definida por el desarrollador. Toda esta información almacenada puede ser independiente de los controles de la página.

Las funciones distribuidas a más de permitir almacenar información entre las solicitudes de página, como las funciones intrínsecas, permite almacenar información en múltiples instancias de la misma aplicación en un equipo o en varios.

#### 1.3.1.5. Configuración de la aplicación

Las aplicaciones ASP.NET utilizan un sistema de configuración que le permite definir valores de configuración para su servidor Web, para un sitio Web o para aplicaciones individuales. Puede crear valores de configuración cuando se implementan las aplicaciones ASP.NET y puede agregar o revisar los valores de configuración en cualquier momento con un impacto mínimo en aplicaciones y servidores Web de operaciones. Los valores de configuración de ASP.NET se almacenan en archivos basados en la tecnología XML. Dado que estos archivos XML son archivos de texto ASCII, es fácil realizar cambios de configuración a sus aplicaciones Web. Puede extender el esquema de configuración para satisfacer sus requisitos.

#### 1.3.1.6. Supervisión de estado y características de rendimiento

ASP.NET incluye características que le permiten supervisar el estado y el rendimiento de su aplicación ASP.NET. La supervisión del estado de ASP.NET permite proporcionar información sobre eventos claves que proporcionan información sobre el estado de una aplicación y sobre las condiciones de error. Estos eventos muestran una combinación de diagnósticos y características de supervisión, a la vez que proporcionan un elevado grado de flexibilidad en lo que respecta a lo que se registra y cómo se registra.

ASP.NET admite dos grupos de contadores de rendimiento a los que pueden obtener acceso las aplicaciones:

- ✓ El grupo de contadores de rendimiento del sistema ASP.NET
- ✓ El grupo de contadores de rendimiento de la aplicación ASP.NET

#### 1.3.1.7. Capacidad de depuración

ASP.NET aprovecha la infraestructura de depuración en tiempo de ejecución para permitir la depuración entre lenguajes y equipos. Se pueden depurar tanto objetos administrados como no administrados, así como todos los lenguajes compatibles con el Common Language Runtime (CLR) y los lenguajes de script. Además, el marco de trabajo de páginas ASP.NET proporciona un modo de seguimiento que permite insertar mensajes de instrumentalización en las páginas Web ASP.NET.

#### 1.3.1.8. Marco de trabajo de servicios Web XML

ASP.NET es compatible con los servicios Web XML. Un servicio Web XML es un componente que incluye funcionalidad de empresa que permite a las aplicaciones intercambiar información entre firewalls utilizando estándares como los servicios de mensajería HTTP y XML. Los servicios Web XML no están relacionados con ninguna tecnología de componentes ni con ninguna convención de llamada a objetos en concreto. Como resultado, pueden obtener acceso a los servicios Web XML los programas escritos en cualquier lenguaje, que usen cualquier modelo de componentes y se ejecuten en cualquier sistema operativo.

#### 1.3.1.9. Entorno de host extensible y administración del ciclo de vida de las aplicaciones

ASP.NET incluye un entorno de host extensible que controla el ciclo de vida de una aplicación desde el momento en que un usuario cualquiera tiene acceso a un recurso (como una página) en la aplicación hasta el momento en que se cierra la aplicación. Aunque ASP.NET se basa en un servidor Web (IIS) como un host de la aplicación, ASP.NET proporciona gran parte de la propia funcionalidad de host. La arquitectura

de ASP.NET permite responder a los eventos de aplicación y crear controladores y módulos HTTP personalizados.

#### 1.3.1.10. Entorno de diseñador extensible

ASP.NET incluye la compatibilidad mejorada para crear diseñadores de controles de servidor Web para utilizarlos con una herramienta de diseño visual como Visual Studio. Los diseñadores permiten crear una interfaz de usuario en tiempo de diseño para un control; de este modo, los desarrolladores pueden configurar las propiedades y el contenido del control en una herramienta de diseño visual.

#### 1.3.2. Posibilidades de desarrollo con ASP.NET

ASP.NET ofrece tres frameworks diferentes para crear aplicaciones web: Formularios Web ASP.NET, MVC ASP.NET y Páginas Web ASP.NET. Todos estos frameworks tienen soporte, actualizaciones y versiones posteriores que hacen que sean opciones robustas y confiables para su objetivo que es crear aplicaciones dinámicas.

##### 1.3.2.1. Formularios Web ASP.NET

Este framework es orientado hacia los desarrolladores que prefieren una programación orientada a controles, muy parecida a los formularios de Windows. Los formularios web son la mejor opción para un rápido desarrollo de aplicaciones (RAP), sin inmiscuirse en el uso avanzado de HTML y JavaScript.

Características:

- Modelo de Eventos.
- Generador HTML.
- Controles para acceso y visualización de datos.
- Preservación de estados en las solicitudes HTTP.

Los formularios web es vista de que son orientados a controles necesitan menos código al momento de crear una aplicación, por lo que puede ser ideal para un

pequeño grupo de desarrolladores o diseñadores que crean aplicaciones rápidamente.

#### 1.3.2.2. Modelo Vista Controlador (MVC) ASP.NET

Esta opción brinda a los desarrolladores la posibilidad de mantener un patrón o arquitectura al momento de crear aplicaciones web, permitiendo separar los componentes del proyecto. Con esto el desarrollador podrán separar los datos, los métodos para acceder a los datos y la interfaz de la aplicación que se crea.

Componentes:

- Modelo (M)
- Vista (V)
- Controlador (C)

Trabajar con este modelo nos brinda mayor facilidad de manipulación del código al trabajar en equipo. Ya que separa equipos que se encarguen de la parte que maneja el giro del negocio y otro equipo que esté a cargo del código que manipula la interfaz del usuario.

#### 1.3.2.3. Páginas Web ASP.NET

Este framework es usado para crear páginas web simples ayudadas principalmente con el uso de sentencias o sintaxis tipo razor. Este modelo es el más rápido para combinar el uso de HTML y código de acceso al servidor, usado principalmente para crear videos o links para acceso a redes sociales, por lo que este framework es orientado hacia estudiantes o desarrolladores que empiezan a trabajar en el desarrollo web.

Uno de los principales aportes en este framework es el uso de helpers que se pueden agregar en páginas web, para facilitar una tarea en pocas líneas de código, que normalmente se crearía de manera tediosa usando varias líneas de código.

## **CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA**

En este capítulo se aborda los conceptos fundamentales sobre recursos humanos y talento humano, proporcionando una idea clara sobre los componentes, objetivos y la organización de cada una de estas dentro de una empresa. En este capítulo también se levantan los requisitos de software usando el formato sugerido en la norma IEEE 830 acompañado de los diagramas de casos de uso, entidad relación, base de datos, actividades, despliegue y diseño de interfaz.

### **2.1. Fundamentos sobre Recursos Humanos**

#### 2.1.1. Introducción

El departamento conocido como recursos humanos aparece inicialmente en empresas grandes, dedicándose a trabajos administrativos como selección de personal, realización de contratos, pago de nóminas, seguros sociales, entre otros. Este nuevo departamento en muchas empresas, principalmente pequeñas, está integrado en el departamento administrativo, pero en base al crecimiento de la empresa nace también la necesidad de tener un departamento que se dedique a la gestión y a la administración del personal de la empresa.

El departamento de recursos humanos se organiza de acuerdo a dos factores: el tamaño y la actividad de la empresa, por lo que el departamento puede tener varios niveles jerárquicos dependiendo de la cantidad de empleados en la empresa. Los objetivos iniciales del departamento de recursos humanos era el organizar, dirigir, coordinar y estudiar las actividades de los trabajadores de la empresa, pero en el transcurso de los años se fueron agregando funciones de control como: la formación, evaluación y desarrollo del personal de la empresa (Montes, 2006).

Gracias a las funciones que realiza el departamento de recursos humanos muchas empresas la consideran como de interés capital en el éxito o en el fracaso de la empresa.

### 2.1.2. Componentes

La composición del departamento de recursos humanos podrá ser diferente en cada empresa, ya que esto depende del número de trabajadores que disponga cada una de ellas y su actividad. Principalmente el departamento debe ser presidido por un directorio, mismo que estará en el segundo nivel jerárquico en el organigrama e igual línea que otras direcciones de la empresa.

La organización dentro del departamento de recursos humanos se la realiza de acuerdo a sus funciones, por lo que su organización puede ser de la siguiente forma:



Ilustración 47 Organigrama Departamento Recursos Humanos

### 2.1.3. Objetivos

(Chiavenato, 2002) considera que el objetivo principal del departamento de talento humano es el seleccionar y desarrollar personal con habilidades, motivación y satisfacción para conseguir los objetivos de la organización.

Asimismo Chiavenato considera que los principales objetivos son los siguientes:

- Lograr eficiencia de los trabajadores.
- Conseguir condiciones favorables de trabajo.
- Satisfacción del personal.
- Generar procesos de mejora continua.
- Contribuir al desarrollo equilibrado entre la organización y las personas que lo componen.

En base a los objetivos por los que se estableció el departamento de recursos humanos las funciones se enmarcaron en tres tipos:

1. organización del personal,
2. gestión del personal y
3. evaluación y control del desempeño.

**1. Organización del personal.-** En este grupo encontramos las siguientes funciones:

<b>Funciones</b>	<b>Tareas</b>
Planificación de las necesidades del personal	<ul style="list-style-type: none"> <li>• Catálogo de puestos de trabajo.</li> <li>• Perfil de cada puesto de trabajo.</li> </ul>
Selección del personal	<ul style="list-style-type: none"> <li>• Incorporación de postulantes.</li> <li>• Selección del perfil idóneo.</li> </ul>
Motivación y formación	<ul style="list-style-type: none"> <li>• Acogida de nuevos trabajadores.</li> <li>• Sistemas de retribución.</li> <li>• Plan de promoción.</li> </ul>
Valoración de los puestos	<ul style="list-style-type: none"> <li>• Jerarquización de los puestos de trabajo.</li> <li>• Valoración de la retribución por categorías.</li> <li>• Estudio de los requisitos de cada puesto.</li> </ul>

Tabla 6 Funciones organizacionales Recursos Humanos

**2. Gestión del personal.-** En este grupo encontramos las siguientes funciones:

<b>Funciones</b>	<b>Tareas</b>
Operaciones administrativas	<ul style="list-style-type: none"> <li>• Elaboración de contratos.</li> <li>• Gestión de incidencias.</li> <li>• Elaboración de nóminas y seguro social.</li> <li>• Tramitación de despidos.</li> </ul>
Relaciones laborales	<ul style="list-style-type: none"> <li>• Prevención de riesgos laborales.</li> <li>• Convenio colectivo.</li> <li>• Resolución de conflictos.</li> </ul>
Servicios sociales	<ul style="list-style-type: none"> <li>• Operaciones de gestión de beneficios sociales.</li> </ul>

Tabla 7 Funciones de Gestión Recursos Humanos

**3. Evaluación y Control del desempeño.-** En este grupo encontramos las siguientes funciones:

<b>Funciones</b>	<b>Tareas</b>
Control de incidencias	<ul style="list-style-type: none"> <li>• Control de asistencia.</li> <li>• Control de ausentismo.</li> <li>• Control de incidencias.</li> </ul>

Elaboración de resultados	• Control de productividad.

Tabla 8 Funciones de evaluación y control Recursos Humanos

Las funciones del departamento de recursos humanos pueden estar sujetas al tamaño de cada empresa, por lo que en empresas grandes el departamento de recursos humanos puede realizar completamente todas las funciones asignadas a ella, gracias a una gran y compleja estructura del departamento. Pero en otras empresas, consideradas pequeñas, muchas de las funciones del departamento de recursos humanos pueden ser ejecutadas o encomendadas a otras empresas como consultorías, asesorías, gestorías, etc.

Este trabajo se enfoca en otorgar una herramienta que automatice el trabajo que se realiza en las funciones de tipo organizacional de las empresas.

## 2.2. Fundamentos sobre Talento Humano

El término talento humano aparece de después de un largo tiempo de haber sido considerado como recursos humanos. El departamento de recursos humanos al igual que otros departamentos de las empresas, están sujetos a cambios y evoluciones que responden a dinámicas internas o externas provocadas por modelos políticos, económicos, sociales, producción y globalización de los mercados.

El cambio global de los mercados llevó a modificar las políticas de la gestión del personal, considerando al recurso humano ya no como un instrumento si no como un capital de la empresa. Capital que brinda movimiento y acción a la empresa a través de sus habilidades y características (Humano, 2013).

Pilar Jericó en su libro Gestión del Talento Humano, define al talento como aquella gente cuyas capacidades están comprometidas a hacer cosas que mejoren los resultados de la organización (Jericó, 2018). De igual manera el autor define los siguientes tipos de talento: Directivo, comercial, técnico, operativo, innovador y emprendedor. Pero la aportación de cada uno de estos talentos dependerá del rol en el cual se desempeñe, ya que un profesional puede alcanzar buenos resultados en su área adecuada, pero no tan buenos en áreas ajenas a la suya.

### 2.2.1. Gestión Talento Humano

La gestión o inclusive la incorporación del área de talento humano son muy susceptibles a la mentalidad de la organización. Todo dependerá de la estructura, organización, procesos, tecnología e inclusive de la cultura de las empresas (Chiavenato, 2002).

La gestión del talento humano se entiende como una serie de decisiones acerca de la relación de los empleados que influyen en la eficiencia de estos y de las organizaciones. Realizando una planeación, organización, dirección y control de las actividades del recurso humano.

Según Chiavenato, la gestión del talento humano se basa en tres aspectos básicos:

- Son seres humanos: Cada individuo tiene su propia personalidad, conocimientos, habilidades y actitudes.
- Activadores inteligentes de los recursos organizacionales: Grupo de personas impulsadoras que le dan el toque dinámico a la organización.
- Socios de la organización: Empleados considerados como socios de la empresa, provocando con esto esfuerzo, dedicación, compromiso, lealtad entre otros beneficios para la empresa.

### 2.2.2. Objetivos

La gestión del área de talento humano se encarga de que exista una colaboración eficiente entre los empleados de la empresa sin importar el nivel jerárquico que tengan cada uno de ellos.

Chiavenato considera que el objetivo general de la gestión del talento humano es una correcta integración de la estrategia, la estructura, los sistemas de trabajo y las personas, con la finalidad de lograr de las personas el despliegue de todas sus habilidades y capacidades y lograr la eficiencia y la competitividad organizacional.

Otro autor define algunos objetivos específicos que se despliegan del objetivo general que propone Chiavenato. Los objetivos específicos son (Cuevas, 2011):

- Ayudar a la organización a alcanzar sus objetivos y realizar su misión.
- Proporcionar competitividad a la organización.

- Suministrar a la organización empleados bien entrenados y motivados.
- Permitir la autorrealización y la satisfacción de los empleados en el trabajo.
- Desarrollar y mantener la calidad de vida en el trabajo.
- Administrar el cambio.
- Establecer políticas éticas y desarrollar comportamientos socialmente responsables.

Procesos de Gestión del Talento Humano: Basados en los objetivos del área y la moderna gestión Chiavenato define los principales procesos que debe realizar el departamento:

- Análisis y descripción de cargos.
- Diseño de cargos.
- Reclutamiento y selección de personal.
- Contratación de candidatos seleccionados.
- Orientación e integración (inducción) de nuevos funcionarios.
- Administración de cargos y salarios
- Incentivos salariales y beneficios sociales
- Evaluación del desempeño de los empleados
- Comunicación con los empleados
- Capacitación y desarrollo del personal
- Desarrollo organizacional
- Higiene, seguridad y calidad de vida en el trabajo
- Relaciones con los empleados y relaciones sindicales

En base a la importancia que en estos últimos años tiene el área de talento humano, este trabajo se enfoca en proporcionar una herramienta que automatice y ayude en la gestión de esta área, principalmente en lo que se refiere a diseño de cargos, reclutamiento y selección de personal.

### **2.3. Especificación de Requisitos de Software (ERS)**

Para la especificación de requisitos del sistema se utiliza el formato sugerido en la norma IEEE 830. En el documento de especificación de requisitos se aclara la funcionalidad, requerimientos, objetivos a alcanzar entre otras especificaciones. El

detalle de la especificación se encuentra en el **Anexo 6. Desarrollo Especificación de Requisitos de Software (ERS)**

### 2.3.1. Diagramas de Casos de Uso

Los casos de uso es una técnica para identificar los actores y el comportamiento de cada uno de ellos con el sistema.

Los casos de uso se acompañan con una descripción que especifica la funcionalidad que se incorporará al sistema. Un Caso de Uso puede 'incluir' la funcionalidad de otro Caso de Uso o puede 'extender' otro Caso de Uso con su propio comportamiento (Geoffrey Sparks).

En la ilustración 48 se visualiza el diagrama de casos de uso para el sistema de “Gestión de Selección de Personal”. Este diagrama muestra el usuario y los diferentes tipos de opciones o comportamientos que puede tener frente al sistema.

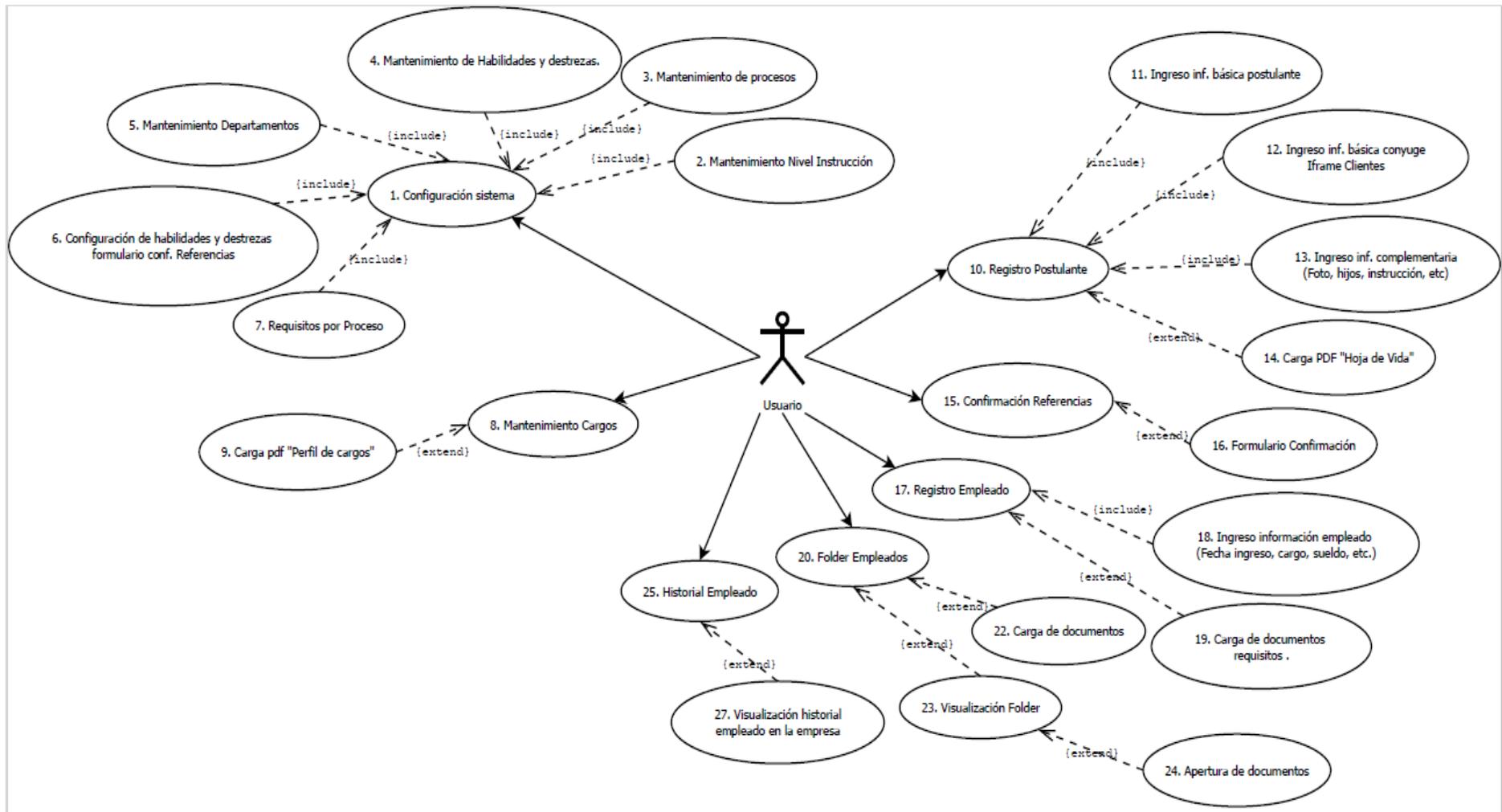


Ilustración 48 Diagrama Casos de Uso

### 2.3.2. Descripción de Casos de Uso

La descripción de los casos de uso muestra un resumen sobre el objetivo, acciones, requisitos, entre otros, de cada caso de uso. Existen diferentes formatos usados para la descripción de los casos de uso, en este documento se utiliza el formato proporcionado por IBM en su página oficial (IBM).

## CASOS DE USO

<b>Caso de Uso</b>	Mantenimiento Nivel Instrucción	<b>Número:</b>	02
<b>Código</b>	PRYTLCU02	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se creará un proceso CRUD en donde se definen los niveles de educación que pueden ser utilizados en el ingreso de los postulantes a los diferentes cargos de la empresa.		
<b>Precondición</b>	Ninguna		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Presentar una pantalla en forma de grilla donde se muestra toda la información almacenada, además en la parte inferior se presentan iconos referenciales para Agregar, Editar, Eliminar y Búsqueda.
<b>Pos condición</b>			
<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Presenta un formulario en donde se ingresa la descripción del nivel de instrucción.
3	Ingresar todos los datos	4	Validar que el tipo y la longitud de caracteres es correcto.
5	Pulsa en guardar	6	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se almacena y se recarga la grilla donde se presenta la información almacenada.
<b>Pos condición</b>			
<b>Flujo Alternativo: Editar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Editar	2	Muestra un formulario con la información a editar

3	Ingresar los datos a modificar	4	El sistema valida que el tipo y la longitud sea la correcta
5	Pulsar en modificar	6	En caso de existir algún error no se almacenará hasta que se corrijan los errores, caso contrario se almacena la información editada.
<b>Pre condición</b>		Seleccionar fila a editar	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsa en el botón eliminar	2	Se presenta una pantalla en donde se muestra un mensaje, el cual pide autorización para eliminar la fila seleccionada.
3	Pulsar en eliminar	4	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>		Seleccionar fila a eliminar	
<b>Observaciones</b>			

<b>Caso de Uso</b>	Mantenimiento Procesos	<b>Número:</b>	03
<b>Código</b>	PRYTLCU03	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se crea un proceso CRUD en donde se definen los procesos de la empresa que serán manejados dentro del sistema.		
<b>Precondición</b>	Ninguna		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Muestra una pantalla en forma de grilla donde se presentara toda la información almacenada, además en la parte inferior se presenta iconos referenciales para Agregar, Editar, Eliminar y Búsqueda.
<b>Pos condición</b>			
<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Presenta un formulario en donde se ingresa la descripción del proceso a crear.
3	Ingresar todos los datos	4	Validar que el tipo y la longitud de caracteres es correcto.
5	Pulsa en guardar	6	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se almacena y se recarga la grilla donde se presenta la información almacenada.
<b>Pos condición</b>			
<b>Flujo Alternativo: Editar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Editar	2	Muestra un formulario con la información a editar
3	Ingresar los datos a modificar	4	El sistema valida que el tipo y la longitud sea la correcta

5	Pulsar en modificar	6	En caso de existir algún error no se almacenará hasta que se corrijan los errores, caso contrario se almacena la información editada.
<b>Pre condición</b>		Seleccionar fila a editar	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsa en el botón eliminar	2	Se muestra una pantalla en donde se presenta un mensaje, el cual pide autorización para eliminar la fila seleccionada.
3	Pulsar en eliminar	4	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>		Seleccionar fila a eliminar	
<b>Observaciones</b>			

<b>Caso de Uso</b>	Mantenimiento de Habilidades y destrezas	<b>Número:</b>	04
<b>Código</b>	PRYTLCU04	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se crea un proceso CRUD en donde se define una base de conocimientos de habilidades y destrezas, mismas que serán utilizadas en otros casos de uso del sistema.		
<b>Precondición</b>	Ninguna		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Muestra una pantalla en forma de grilla donde se presenta toda la información almacenada, además en la parte inferior se muestra iconos referenciales para Agregar, Editar, Eliminar y Búsqueda.
<b>Pos condición</b>			
<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Presenta un formulario en donde se ingresa el tipo (Funcionales / Organizacionales) y la descripción de la habilidad o destreza.
3	Ingresar todos los datos	4	Validar que el tipo y la longitud de caracteres es correcto.
5	Pulsa en guardar	6	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se almacena y se recarga la grilla donde se presenta la información almacenada.
<b>Pos condición</b>			
<b>Flujo Alternativo: Editar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Editar	2	Presenta un formulario con la información a editar

3	Ingresar los datos a modificar	4	El sistema valida que el tipo y la longitud sea la correcta
5	Pulsar en modificar	6	En caso de existir algún error no se almacenara hasta que se corrijan los errores, caso contrario se almacena la información editada.
<b>Pre condición</b>		Seleccionar fila a editar	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsa en el botón eliminar	2	Se presenta una pantalla en donde se muestra un mensaje, el cual pide autorización para eliminar la fila seleccionada.
3	Pulsar en eliminar	4	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>			
<b>Observaciones</b>			

<b>Caso de Uso</b>	Mantenimiento de Departamentos	<b>Número:</b>	05
<b>Código</b>	PRYTLCU05	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se crea un proceso CRUD en donde se define los departamentos que tiene la empresa, según organigrama organizacional.		
<b>Precondición</b>	Ninguna		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Presentar una pantalla en forma de grilla donde se muestra toda la información almacenada, además en la parte inferior se presentan iconos referenciales para Agregar, Editar, Eliminar y Búsqueda.
<b>Pos condición</b>			

<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Muestra un formulario en donde se ingresa el nombre y la descripción del departamento a ingresar.
3	Ingresar todos los datos	4	Valida que el tipo y la longitud de caracteres es correcto.
5	Pulsa en guardar	6	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se guarda y se recarga la grilla donde se presenta la información almacenada.
<b>Pre condición</b>			
<b>Flujo Alternativo: Editar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Editar	2	Presenta un formulario con la información a editar
3	Ingresar los datos a modificar	4	El sistema valida que el tipo y la longitud sea la correcta
5	Pulsar en modificar	6	En caso de existir algún error no se almacenara hasta que se corrijan los errores, caso contrario se almacena la información editada.
<b>Pre condición</b>		Seleccionar fila a editar	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsa en el botón eliminar	2	Se presenta una pantalla en donde se muestra un mensaje, el cual pide autorización para eliminar la fila seleccionada.
3	Pulsar en eliminar	4	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>			
<b>Observaciones</b>			

<b>Caso de Uso</b>	Configuración de habilidades / destrezas por formulario	<b>Número:</b>	06
<b>Código</b>	PRYTLCU06	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se crea un proceso que permite la configuración (Agregar, eliminar, buscar) de habilidades y destrezas que serán consideradas para su evaluación en el formulario seleccionado.		
<b>Precondición</b>	Ingresar base de conocimientos de habilidades y destrezas		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Presenta una pantalla en forma de grilla donde se muestra toda la información almacenada, además de un listado, que podrá ser seleccionado, de formularios precargados. También en la parte inferior se presentan iconos referenciales para Agregar, Eliminar y Búsqueda.
<b>Pos condición</b>			
<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Presenta un formulario en donde se debe seleccionar la habilidad o destreza que desea ingresar para la evaluación en el formulario seleccionado.
3	Pulsa en guardar	4	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se almacena y se recarga la grilla donde se presenta la información almacenada.
<b>Pre condición</b>	Seleccionar formulario de entre listado precargado.		
<b>Flujo Alternativo: Editar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Editar	2	Presenta un formulario con la información a editar

3	Ingresar los datos a modificar	4	El sistema valida que el tipo y la longitud sea la correcta
5	Pulsar en modificar	6	En caso de existir algún error no se almacenara hasta que se corrijan los error, caso contrario se almacena la información editada.
<b>Pre condición</b>		Seleccionar fila a modificar	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsa en el botón eliminar	2	Se presenta una pantalla en donde se presenta un mensaje, el cual pide autorización para eliminar la fila seleccionada.
3	Pulsar en eliminar	4	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>		Seleccionar fila a eliminar	
<b>Observaciones</b>			

<b>Caso de Uso</b>	Requisitos por proceso	<b>Número:</b>	07
<b>Código</b>	PRYTLCU07	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Administrador		
<b>Descripción</b>	Se crea un proceso CRUD que permite la configuración de requisitos necesarios para completar cierto proceso dentro del sistema.		
<b>Precondición</b>	Ingresar base de conocimientos de procesos de la empresa.		
<b>Flujo Principal personal para autorizar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Muestra una pantalla en forma de grilla donde se presentara toda la información almacenada, además de un listado, que podrá ser seleccionado, de procesos disponibles en el sistema. También en la

			parte inferior se presentan iconos referenciales para Agregar, Editar, Eliminar y Búsqueda.
<b>Pos condición</b>			
<b>Flujo Alternativo: Agregar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Pulsar en Agregar	2	Presenta un formulario en donde se debe ingresar una descripción del requisito necesario para el proceso seleccionado.
3	Ingresar todos los datos	4	El sistema valida que el tipo y la longitud sea la correcta
5	Pulsa en guardar	6	En caso de existir algún error no se almacenará hasta que se corrija los errores, caso contrario se almacena y se recarga la grilla donde se presenta la información almacenada.
<b>Pre condición</b>		Seleccionar proceso de entre listado.	
<b>Flujo Alternativo: Eliminar</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
3	Pulsa en el botón eliminar	4	Se muestra una pantalla en donde se presenta un mensaje, el cual pide autorización para eliminar la fila seleccionada.
N	Pulsar en eliminar	N	El sistema eliminara de forma lógica la fila seleccionada.
<b>Pre condición</b>		Seleccionar fila a eliminar	
<b>Observaciones</b>			

<b>Caso de Uso</b>	Mantenimiento Cargos	<b>Número:</b>	08
<b>Código</b>	PRYTLCU08	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Usuario		
<b>Descripción</b>	Se realiza el ingreso y edición de los perfiles de los cargos que tiene la empresa. Estos perfiles serán los comparados con los perfiles de los postulantes a los diferentes cargos de la empresa.		
<b>Precondición</b>	Ingresar cargos, habilidades y destrezas disponibles en la empresa.		
<b>Flujo Principal Ingresar al mantenimiento</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder al mantenimiento	2	Muestra una pantalla en donde se debe ingresar los siguientes datos: nombre del cargo, seleccionar el departamento al cual pertenece el cargo, nombre del jefe inmediato, misión del cargo, responsabilidades, formación académica, conocimientos específicos, capacitación adicional, nivel experiencia, habilidades y destrezas requeridas para el cargo.
3	Ingresar todos los datos requeridos		
4	Presiona botón para ingresar archivo perfil de cargo en formato PDF	5	Despliega ventana para la selección de archivos.
6	Selecciona archivo	7	Guarda archivo en memoria y visualiza el nombre del archivo ingresado por el usuario.
8	Clic en botón "Grabar Cargo"	9	Sistema valida la información sea la correcta o presenta el error en donde se produzca.
		10	Visualiza confirmación al grabar cargo
		11	Deshabilita botón "Grabar Cargo"
		12	
<b>Pos condición</b>			
<b>Flujo Alternativo: Modificación de Cargos</b>			

Acción del Usuario		Respuesta del Sistema	
1	Acceder al mantenimiento	2	Presenta una pantalla en donde se debe ingresar los siguientes datos: nombre del cargo.
3	Ingresa los datos	4	El sistema verifica si el nombre del cargo ingresado ya existe, si es ese el caso el sistema muestra los datos guardados en el cargo para que puedan ser modificados si es necesario.
5	Modifica los datos necesarios		
6	Usuario da clic en botón "Modificar Cargo"	7	Sistema valida la información sea la correcta o presenta el error en donde se produzca.
		8	Visualiza confirmación al grabar cargo
		9	Deshabilita botón "Modificar Cargo"
<b>Pos condición</b>			
<b>Observaciones</b>			

<b>Caso de Uso</b>	Registro Postulante	<b>Número:</b>	10
<b>Código</b>	PRYTLCU09	<b>Versión</b>	1
<b>Importancia</b>	Alto	<b>Estado</b>	Pendiente
<b>Actor</b>	Usuario		
<b>Descripción</b>	Se registra los postulantes para los diferentes cargos que dispone la empresa.		
<b>Precondición</b>	Ingresar cargos y niveles de instrucción.		
<b>Flujo Principal Ingresar al mantenimiento</b>			
Acción del Usuario		Respuesta del Sistema	
1	Acceder al mantenimiento	2	Muestra una pantalla en donde se debe seleccionar inicialmente el cargo al cual aplica, posterior ingresar los siguientes datos: cédula identidad, nombres, apellidos, dirección domiciliaria, teléfono, cédula identidad cónyuge (*), nombres

			<p>cónyuge (*), fecha nacimiento cónyuge (*), número de hijos (*), nivel de instrucción, ocupación actual (*), ocupaciones anteriores (*), referencias personales, aspiración salarial, disponibilidad.</p> <p>En caso de ingresar número de hijos, debe llenar la información de cada hijo (Nombres, documento identidad, país nacimiento, fecha nacimiento, discapacidad) en la grilla presentada.</p> <p>Los campos marcados con el carácter asterisco (*) son campos que pueden ir en blanco.</p>
3	Ingresar todos los datos		
4	Presiona botón para ingresar archivo hoja de vida en formato PDF	5	Despliega ventana para la selección de archivos.
6	Selecciona archivo	7	Guarda archivo en memoria y visualiza el nombre del archivo ingresado por el usuario.
8	Clic en botón "Guardar Formulario"	9	Sistema verifica si los datos del postulante aún no han sido registrados en el sistema, en el caso de que ya exista envía mensaje y no permite guardar datos.
		10	Sistema valida la información sea la correcta o presenta el error en donde se produzca.
		11	Visualiza confirmación al grabar postulante.
		12	Deshabilita botón "Guardar Formulario"
<b>Pos condición</b>			
<b>Observaciones</b>			

<b>Caso de Uso</b>	Confirmación Referencias	<b>Número:</b>	15
<b>Código</b>	PRYTLCU10	<b>Versión</b>	1
<b>Importancia</b>	Media	<b>Estado</b>	Pendiente
<b>Actor</b>	Usuario		
<b>Descripción</b>	Se valida las referencias ingresadas por los postulantes a los diferentes cargos de la empresa. La validación o confirmación de la información se la realiza mediante un formulario pre configurado en el sistema. Toda la información que el usuario llene en el formulario será proporcionada por la persona ingresada como referencia del postulante.		
<b>Precondición</b>			
<b>Flujo Principal Ingresar al mantenimiento</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder a la página	2	Muestra una pantalla con un campo para la búsqueda del postulante y una grilla en la cual se mostrará todas las referencias que el postulante ingresó en su formulario.
3	Ingresar cédula del postulante	4	Verifica si el dato ingresado coincide con algún registro de postulantes almacenados en el sistema.
		5	Despliega lista de postulantes que coincida con los valores ingresados.
6	Selecciona postulante de entre lista desplegada	7	Verifica postulante seleccionado, y muestra información sobre sus referencias laborales ingresadas en el formulario de registro de postulantes.
8	Clic en botón "Guardar Formulario"	9	Muestra botón "VERIFICAR REFERENCIA" en la parte inferior de la grilla
10	Presiona botón "VALIDAR REFERENCIA"	11	Sistema muestra formulario en el que el usuario debe marcar con un check sobre el valor considerado para cada habilidad o destreza que se presenta en el formulario.

12	Selecciona valores y observaciones		
13	Pulsa el botón "Guardar información"	14	Sistema valida la información sea la correcta o presenta el error en donde se produzca.
		15	Cierra formulario "VALIDACIÓN REFERENCIAS".
		16	Guarda formulario en formato PDF en el folder del empleado.
<b>Pos condición</b>		Actualiza grilla con las referencias del postulante.	
<b>Observaciones</b>			

<b>Caso de Uso</b>	Registro de Empleados	<b>Número:</b>	17
<b>Código</b>	PRYTLCU11	<b>Versión</b>	1
<b>Importancia</b>	Alta	<b>Estado</b>	Pendiente
<b>Actor</b>	Usuario		
<b>Descripción</b>	Se registran los empleados que trabajan en las diferentes áreas de la empresa.		
<b>Precondición</b>	Ingresar personal como postulantes. Ingresar cargos de la empresa.		
<b>Flujo Principal Ingresar al mantenimiento</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder a la página	2	Muestra una pantalla en la que el usuario debe ingresar el postulante que fue seleccionado para ingresar como empleado a la empresa. Adicional se muestran los siguientes campos: sueldo, fecha de ingreso, sucursal, cargo y requisitos para la admisión.
3	Ingresar cédula del postulante	4	Verifica si el dato ingresado coincide con algún registro de postulantes almacenados en el sistema.

		5	Llena información de apellidos, nombres, dirección y teléfono en los campos del formulario.
6	Completa información del formulario		
7	Pulsa el botón "Registrar Empleado"	8	Sistema valida la información sea la correcta o presenta el error en donde se produzca.
		9	Sistema muestra formulario en el que el usuario debe marcar con un check sobre el valor considerado para cada habilidad o destreza que se presenta en el formulario.
		10	Muestra mensaje de confirmación y deshabilita botón "Registrar Empleado".
<b>Pos condición</b>			
<b>Observaciones</b>			

<b>Caso de Uso</b>	Folder Empleados	<b>Número:</b>	20
<b>Código</b>	PRYTLCU12	<b>Versión</b>	1
<b>Importancia</b>	Alta	<b>Estado</b>	Pendiente
<b>Actor</b>	Usuario		
<b>Descripción</b>	Se ingresa los documentos en las carpetas electrónicas de los empleados de la empresa.		
<b>Precondición</b>	Ingresar empleados.		
<b>Flujo Principal Ingresar a la página</b>			
<b>Acción del Usuario</b>		<b>Respuesta del Sistema</b>	
1	Acceder a la página	2	Muestra una pantalla con una grilla que presenta un listado de empleados registrados en la empresa. Además se muestra un botón para abrir el folder del empleado y otro para agregar archivos al folder del empleado.
3	Selecciona empleado del cual quiere ver el folder.		
4	Presiona botón "Abrir Folder Empleado"	5	Abre una ventana auxiliar con una lista de todos los archivos guardados en el folder del empleado.
6	Selecciona un documento de entre la lista mostrada en el folder	7	Visualiza el documento en una ventana emergente
8	Presiona botón "Agregar Archivo"	9	Verifica si fue seleccionado un empleado de la grilla, si es así muestra una ventana emergente con un botón para explorar archivos de la computadora. Si no fue seleccionado algún empleado el sistema muestra un mensaje con el error.
10	Presiona botón "Seleccionar archivo"	11	Despliega una ventana emergente para la selección de un archivo que esté

			almacenado en el computador del usuario.
12	Selecciona un archivo desde el navegador de archivos	13	Muestra mensaje con la confirmación de la carga del archivo.
<b>Pos condición</b>	Cierra ventana emergente.		
<b>Observaciones</b>			

## 2.4. Diseño de Datos

### 2.4.1. Diagrama de Entidad – Relación

El Modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación (Storti, 2007).

Se observa en la ilustración 48 el diagrama entidad relación del sistema de “Gestión de Personal”. Mismo que contiene todas las entidades, con sus atributos, que se crean en la base de datos del sistema.



#### **2.4.2. Diccionario de Datos**

El diccionario de datos es una representación del esquema de la base de datos. Este esquema contiene metadatos, es decir, datos acerca de los datos. Un diccionario de datos contempla la descripción de las tablas, relaciones, atributos, propiedades de los atributos, restricciones de integridad, entre otros (Silberschatz, F. Korth, & Sudarshan, 2002).

En la siguiente tabla se detalla el diccionario de datos del sistema “Gestión de Personal”, mismo que contiene un listado de todas las tablas, atributos y propiedades creadas en la base de datos.

Nombre Tabla	Descripción	#	Nombre Columna	Descripcion	Tipo	Longitud	Permite Nulo	Llave Primaria	Llave Foranea	Autoincrementable	Calculado	
Calificaciones_Formulario	Guarda las calificaciones de las distintas habilidades o destrezas incluidas en un formulario. Dicha calificación será otorgada por la persona ingresada como referencia del postulante.	1	ID_DET_FORMULARIO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_frm_ref_personal_postulante	Identificador de la tabla Formulario_Postulante. Especifica el formulario realizado a la referencia personal del postulante, al cual pertenece las calificaciones creadas.	int	4	NO	NO	SI	NO	NO	NO
		3	id_habilidad_formulario	Identificador de la tabla Habilidad_Formulario. Especifica que habilidad o destreza considerada en el formulario será calificada.	int	4	NO	NO	SI	NO	NO	NO
		4	frm_valor	Valor que indica la calificación que otorga la referencia personal ingresada por el postulante.	nvarchar	10	NO	NO	NO	NO	NO	NO
		5	cal_frm_usuario	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		6	cal_frm_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	SI	NO	NO	NO	NO	NO
Capacitacion_Adicional_Cargo	Guarda las distintas capacitaciones que debería tener un postulante al cargo.	1	ID_CAP_ADICIONAL	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_cargo	Identificador de la tabla Cargo	int	4	NO	NO	SI	NO	NO	NO
		3	cap_adi_descripcion	Descripción de la capacitación adicional que debe tener un postulante para el cargo.	nvarchar	500	NO	NO	NO	NO	NO	NO
		4	cap_adi_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	cap_adi_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		6	cap_adi_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		7	cap_adi_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		8	cap_adi_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
CARGO	Guarda los cargos que dispone la empresa en su organigrama. Dichos cargos estarán disponibles para las postulaciones.	1	ID_CARGO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_empresa	Identificador de la tabla Empresa. Indica la empresa en la que se crea el cargo.	int	4	SI	NO	SI	NO	NO	NO
		3	id_departamento	Identificador de la tabla DEPARTAMENTO. Indica en que departamento se creó el cargo.	int	4	NO	NO	SI	NO	NO	NO
		4	car_nombre	Nombre del cargo.	nvarchar	300	NO	NO	NO	NO	NO	NO
		5	car_nombre_jefe_inmediato	Nombre del jefe inmediato del cargo.	nvarchar	500	SI	NO	NO	NO	NO	NO
		6	car_mision_Cargo	Detalle de la misión del cargo.	nvarchar	2000	SI	NO	NO	NO	NO	NO
		7	car_nombre_archivo_perfil_cargo	Nombre del archivo que contiene el detalle del perfil del cargo. Archivo unicamente con formato PDF.	nvarchar	200	SI	NO	NO	NO	NO	NO
		8	car_path_archivo_perfil_Cargo	Dirección de la ubicación del archivo cargado como perfil del cargo.	nvarchar	-1	SI	NO	NO	NO	NO	NO
		9	car_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		10	car_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		11	car_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		12	car_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		13	car_eliminado	Descripción si el registro esta eliminado o no	nchar	1	NO	NO	NO	NO	NO	NO
Cargo_Habilidades	Especifica las habilidades o destrezas que debería cumplir un postulante para un cargo específico.	1	ID_CARGO_HABILIDAD	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_cargo	Identificador de la tabla Cargo. Especifica a que cargo se le agregan las habilidades o destrezas.	int	4	NO	NO	SI	NO	NO	NO
		3	id_habilidad	Identificador de la tabla Habilidades_Destrezas.	int	4	NO	NO	SI	NO	NO	NO
		4	car_hab_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	car_hab_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		6	car_hab_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		7	car_hab_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		8	car_hab_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Conocimientos_Especificos_Cargo	Especifica los conocimientos específicos que debería tener un postulante para un cargo.	1	ID_CON_ESPECIFICOS	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_cargo	Identificador de la tabla Cargo. Especifica el cargo al cual se le agregan los conocimientos específicos.	int	4	NO	NO	SI	NO	NO	NO
		3	con_esp_descripcion	Descripción del conocimiento específico necesario para cubrir el cargo.	nvarchar	500	NO	NO	NO	NO	NO	NO
		4	con_esp_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	con_esp_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		6	con_esp_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO

		7	con_esp_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		8	con_esp_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO
DEPARTAMENTO	Lista los departamentos que dispone la empresa.	1	ID_DEPARTAMENTO	Identificador de la tabla DEPARTAMENTO.	int	4	NO	SI	NO	SI	NO
		2	id_empresa	Identificador de la tabla EMPRESA. Especifica la empresa que dispone el departamento creado.	int	4	NO	NO	SI	NO	NO
		3	dep_nombre	Nombre del departamento o área disponible en la empresa.	nvarchar	500	NO	NO	NO	NO	NO
		4	dep_descripcion	Descripción que identifique al departamento creado.	nvarchar	500	SI	NO	NO	NO	NO
		5	dep_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO
		6	dep_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO
		7	dep_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		8	dep_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO
		9	dep_eliminado	Descripción si el campo esta eliminado o no	char	1	NO	NO	NO	NO	NO
EMP_POSTULANTE	Guarda toda la información sobre los postulantes a los diferentes cargos de la empresa.	1	ID_EMP_POSTULANTE	Identificador de la tabla EMP_POSTULANTE.	int	4	NO	SI	NO	SI	NO
		2	id_empresa	Identificador de la tabla Empresa. Especifica la empresa en la que se registra el postulante.	int	4	NO	NO	SI	NO	NO
		3	id_cargo	Identificador de la tabla Cargo. Especifica el cargo al cual se postula.	int	4	NO	NO	SI	NO	NO
		4	id_tipo_documento	Identificador de la tabla Tipo_Documento. Especifica el tipo de documento de identidad del postulante	int	4	NO	NO	SI	NO	NO
		5	pos_nombres	Nombres completos del postulante.	nvarchar	500	SI	NO	NO	NO	NO
		6	pos_apellidos	Apellidos completos del postulante	nvarchar	250	SI	NO	NO	NO	NO
		7	pos_documento_identidad	Documento de identidad del postulante.	nvarchar	50	SI	NO	NO	NO	NO
		8	pos_emp_direccion	Dirección del domicilio o trabajo del postulante	nvarchar	500	SI	NO	NO	NO	NO
		9	pos_emp_telefono	Número de teléfono del postulante.	nvarchar	150	SI	NO	NO	NO	NO
		10	pos_emp_nombre_conyuge	Nombres completos del conyuge del postulante.	nvarchar	500	SI	NO	NO	NO	NO
		11	pos_emp_cedula_conyuge	Número de cédula del conyuge del postulante.	nvarchar	15	SI	NO	NO	NO	NO
		12	pos_emp_fecha_nac_conyuge	Fecha de nacimiento del conyuge del postulante.	datetime	8	SI	NO	NO	NO	NO
		13	pos_emp_numero_hijos	Número de hijos que tiene el postulante.	int	4	NO	NO	NO	NO	NO
		14	pos_emp_aspiracion_salarial	Aspiración salarial del postulante.	decimal	5	SI	NO	NO	NO	NO
		15	pos_emp_disponibilidad_inmediata	Especifica si el postulante tiene disponibilidad inmediata en caso de ser seleccionado para el cargo.	bit	1	SI	NO	NO	NO	NO
		16	pos_emp_turnos_rotativos	Especifica si el postulante está dispuesto a tener horarios rotativos.	bit	1	SI	NO	NO	NO	NO
		17	pos_emp_det_trabajo_competente	Detalle del trabajo y funciones para las que el postulante se cree competente.	nvarchar	1000	SI	NO	NO	NO	NO
		18	pos_emp_ruta_hoja_vida	Ruta de la ubicación del archivo que contiene el curriculum del postulante.	nvarchar	1000	SI	NO	NO	NO	NO
		19	pos_emp_nnombre_hoja_vida	Nombre del documento cargado con la hoja de vida del postulante.	nvarchar	200	SI	NO	NO	NO	NO
		20	pos_emp_usuario_creacion	Usuario con el que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO
		21	pos_emp_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO
		22	pos_emp_usuario_modificacion	Usuario con el que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO
		23	pos_emp_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		24	pos_emp_eliminado	Descripción si el campo esta eliminado o no	char	1	SI	NO	NO	NO	NO

EMPLEADO_TALENTO_HUMANO	Guarda la información sobre todos los empleados que están registrados en la empresa.	1	ID_EMPLEADO_TAL_HUMANO	Identificador de la tabla EMPLEADO_TALENTO_HUMANO	int	4	NO	SI	NO	SI	NO
		2	id_empresa	Identificador de la tabla Empresa. Especifica la empresa en la que se registra el empleado.	int	4	NO	NO	SI	NO	NO
		3	id_sucursal	Identificador de la tabla Sucursal. Especifica la sucursal en la que trabaja el empleado.	int	4	NO	NO	SI	NO	NO
		4	id_postulante	Identificar de la tabla EMP_POSTULANTE.	int	4	SI	NO	SI	NO	NO
		5	id_cargo	Identificador de la tabla Cargo.	int	4	NO	NO	SI	NO	NO
		6	emp_tal_hum_fecha_ingreso	Fecha en la que empieza a trabajar el empleado en la sucursal seleccionada.	datetime	8	NO	NO	NO	NO	NO
		7	emp_tal_hum_fecha_salida	Fecha en la que termina el contrato del empleado con la empresa.	datetime	8	SI	NO	NO	NO	NO
		8	emp_tal_hum_motivo_salida	Detalle del motivo de la salida del empleado de la empresa.	nvarchar	250	SI	NO	NO	NO	NO
		9	emp_tal_hum_sueldo	Sueldo que percibe el empleado.	decimal	5	SI	NO	NO	NO	NO
		10	rrhh_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO
		11	rrhh_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO
		12	rrhh_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO
		13	rrhh_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		14	rrhh_eliminado	Descripción si el registro esta eliminado o no	nchar	1	NO	NO	NO	NO	NO
Empresa	Guarda la información sobre las empresas que manejan el sistema.	1	ID_EMPRESA	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO
		2	emp_nombre	Nombre de la empresa	nvarchar	500	NO	NO	NO	NO	NO
		3	emp_eliminado	Campo que indica si el registro se encuentra eliminado.	char	1	NO	NO	NO	NO	NO
Folder_Empleado	Guarda los archivos digitales de los empleados de la empresa. Algunos de estos archivos pueden ser requisitos para su ingreso a la empresa.	1	ID_FOLDER_EMPLEADO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO
		2	id_empleado	Identificador de la tabla EMPLEADO_TALENTO_HUMANO. Especifica el empleado al cual pertenece el folder creado.	int	4	NO	NO	SI	NO	NO
		3	id_requisito	Identificador de la tabla Requisitos. Especifica si el archivo cargado en el folder del empleado es uno de los requisitos necesarios para el ingreso a la empresa.	int	4	SI	NO	SI	NO	NO
		4	folder_direccion_archivo	Dirección de la ubicación del archivo guardado en el folder del empleado.	nvarchar	1000	NO	NO	NO	NO	NO
		5	folder_nombre_archivo	Nombre del archivo guardado en el folder del empleado.	nvarchar	200	NO	NO	NO	NO	NO
		6	folder_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO
Formacion_Academica_Cargo	Detalla la formación académica que debería cumplir un postulante para ocupar un cargo específico.	1	ID_FORM_ACADEMICA_CARGO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO
		2	id_cargo	Identificador de la tabla Cargo. Especifica el cargo al cual se le agrega la formación académica necesaria para desempeñar el cargo.	int	4	NO	NO	SI	NO	NO
		3	form_aca_descripcion	Descripción de la formación académica.	nvarchar	500	SI	NO	NO	NO	NO
		4	form_aca_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO
		5	form_aca_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO
		6	form_aca_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO
		7	form_aca_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		8	form_aca_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO
Formulario	Detalla los formularios que estarán disponibles en el sistema. Estos formularios deberán ser evaluados a los postulantes.	1	ID_FORMULARIO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO
		2	frm_descripcion	Descripción del formulario creado.	nvarchar	500	NO	NO	NO	NO	NO
		3	frm_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO
		4	frm_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO
		5	frm_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO
		6	frm_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO
		7	frm_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO

Formulario_Postulante	Guarda la información de los formularios realizados a los diferentes postulantes a los cargos de la empresa.	1	ID_FRM_REF_POSTULANTE	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_formulario	Identificador de la tabla Formulario. Especifica el formulario que se llenará con la ayuda de la referencia personal del postulante.	int	4	NO	NO	SI	NO	NO	NO
		3	id_post_hist_laboral	Identificador de la tabla Pos_Referencias_Personales. Especifica la referencia personal a la cual se le consulta la información para llenar el formulario.	int	4	NO	NO	SI	NO	NO	NO
		4	frm_ref_observaciones	Observaciones que pudieron ser proporcionadas por la persona indicada como referencia del postulante.	nvarchar	500	SI	NO	NO	NO	NO	NO
		5	frm_ref_path_archivo	Dirección de la ubicación del archivo generado al completar el formulario.	nvarchar	1000	SI	NO	NO	NO	NO	NO
		6	frm_ref_nombre_archivo	Nombre del archivo generado al completar el formulario.	nvarchar	500	SI	NO	NO	NO	NO	NO
		7	frm_ref_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		8	frm_ref_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		9	frm_ref_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Habilidad_Formulario	Detalla las habilidades o destrezas que serán evaluadas en los formularios creados.	1	ID_FORMULARIO_HAB	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_formulario	Identificador de la tabla Formulario. Especifica el formulario al cual se le agregan las habilidades o destrezas.	int	4	NO	NO	SI	NO	NO	NO
		3	id_hab_destreza	Identificador de la tabla Habilidades_Destrezas. Especifica las habilidades o destrezas que constarán en el formulario para ser evaluadas.	int	4	NO	NO	SI	NO	NO	NO
		4	frm_hab_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	frm_hab_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		6	frm_hab_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		7	frm_hab_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		8	frm_hab_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Habilidades_Destrezas	Base de conocimientos que almacena información sobre habilidades o destrezas que pueden cumplir los postulantes.	1	ID_ID_HABILIDAD_DES	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_empresa	Identificador de la tabla Empresa. Especifica la empresa en la cual se crean las habilidades o destrezas.	int	4	NO	NO	SI	NO	NO	NO
		3	hab_tipo	Especifica el tipo de habilidad o destrezas creada. Estas pueden ser de tipo organizacionales o funcionales.	nvarchar	50	NO	NO	NO	NO	NO	NO
		4	hab_descripcion	Descripción de la habilidad o destreza creada.	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	hab_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		6	hab_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		7	hab_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		8	hab_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		9	hab_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Historial_Empleado	Guarda el historial de todos los movimientos que tenga el empleado dentro de la empresa.	1	ID_HISTORIAL_EMPLEADO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_empleado	Identificador de la tabla Empleado_Talento_Humano.	int	4	NO	NO	SI	NO	NO	NO
		3	id_sucursal	Identificador de la tabla Sucursal. Especifica la sucursal en la que laboró el empleado.	int	4	SI	NO	SI	NO	NO	NO
		4	id_cargo	Identificador de la tabla. Especifica el cargo que ocupó el empleado.	int	4	SI	NO	SI	NO	NO	NO
		5	historial_descripcion	Descripción del movimiento del empleado en la sucursal de la empresa.	nvarchar	500	NO	NO	NO	NO	NO	NO
		6	historial_sueldo	Detalle del sueldo que percibía el empleado.	decimal	5	SI	NO	NO	NO	NO	NO
		7	historial_fecha_ingreso	Fecha de ingreso del empleado en la sucursal de la empresa.	datetime	8	SI	NO	NO	NO	NO	NO
		8	historial_fecha_movimiento	Fecha en la que se registra el historial del empleado.	datetime	8	NO	NO	NO	NO	NO	NO
		9	historial_usuario	Usuario que registra el movimiento del empleado.	nvarchar	500	NO	NO	NO	NO	NO	NO

Historial_Laboral_Postulante	Guarda la información sobre la experiencia laboral que tienen los postulantes.	1	ID_POST_HIST_LABORAL	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_postulante	Identificador de la tabla postulante. Especifica a que postulante pertenece el historial guardado.	int	4	NO	NO	SI	NO	NO	NO
		3	hist_nombre_empresa	Nombre de la empresa en la que trabajó el postulante.	nvarchar	250	NO	NO	NO	NO	NO	NO
		4	hist_fecha_ingreso	Fecha en la cual ingresó el postulante a la empresa.	datetime	8	NO	NO	NO	NO	NO	NO
		5	hist_fecha_salida	Fecha de salida de la empresa.	datetime	8	SI	NO	NO	NO	NO	NO
		6	hist_cargo	Último cargo que ocupó el postulante en la empresa.	nvarchar	250	NO	NO	NO	NO	NO	NO
		7	hist_sueldo	Último sueldo percibido por el postulante en la empresa.	decimal	5	NO	NO	NO	NO	NO	NO
		8	hist_jefe_inmediato	Nombres completos del jefe inmediato que tenía el postulante en la empresa.	nvarchar	500	NO	NO	NO	NO	NO	NO
		9	hist_motivo_salida	Detalle del motivo de salida del postulante de la empresa.	nvarchar	500	SI	NO	NO	NO	NO	NO
		10	hist_descripcion_tareas	Breve descripción de las tareas que realizaba el postulante en la empresa.	nvarchar	500	SI	NO	NO	NO	NO	NO
		11	hist_eliminado	Descripción si el registro esta eliminado o no	char	1	SI	NO	NO	NO	NO	NO
Instruccion_Postulante	Guarda la información sobre la instrucción académica que tiene el postulante.	1	ID_INST_POSTULANTE	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_instruccion	Identificador de la tabla Nivel_Instrucción.	int	4	NO	NO	SI	NO	NO	NO
		3	id_postulante	Identificador de la tabla EMP_POSTULANTE.	int	4	NO	NO	SI	NO	NO	NO
		4	pos_inst_establecimiento	Nombre del establecimiento educativo en el cual realizó la instrucción.	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	pos_inst_ultimo_curso_aprobado	Último curso aprobado en el establecimiento educativo.	nvarchar	150	NO	NO	NO	NO	NO	NO
		6	pos_inst_titulo	Título obtenido al terminar la instrucción.	nvarchar	250	NO	NO	NO	NO	NO	NO
		7	pos_inst_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		8	pos_inst_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		9	pos_inst_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		10	pos_inst_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		11	pos_inst_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Nivel_Experiencia_Cargo	Especifica el nivel de experiencia que debe cumplir un postulante para un cargo específico.	1	ID_NIV_EXPERIENCIA	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_cargo	Identificador de la tabla Cargo. Especifica la experiencia requerida para aplicar al cargo.	int	4	NO	NO	SI	NO	NO	NO
		3	niv_exp_descripcion	Descripción de la experiencia necesaria para aplicar al cargo.	nvarchar	250	NO	NO	NO	NO	NO	NO
		4	niv_exp_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		5	niv_exp_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		6	niv_exp_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		7	niv_exp_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		8	niv_exp_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Nivel_Instruccion	Base de conocimientos que almacena los diferentes niveles de instrucción académica que puede alcanzar una persona.	1	ID_NIV_INSTRUCCION	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_empresa	Identificador de la tabla Empresa. Especifica la empresa a la cual pertenece la instrucción creada.	int	4	NO	NO	SI	NO	NO	NO
		3	niv_inst_descripcion	Descripción de la instrucción.	nvarchar	250	NO	NO	NO	NO	NO	NO
		4	niv_inst_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Pos_Cargas_Familiares	Almacena la información sobre las cargas familiares que tienen los postulantes a los diferentes cargos de la empresa.	1	ID_CARGA_FAMILIAR	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_postulante	Identificador de la tabla EMP_POSTULANTE.	int	4	NO	NO	SI	NO	NO	NO
		3	car_fam_nombres	Nombres de la carga familiar del postulante.	nvarchar	500	NO	NO	NO	NO	NO	NO
		4	car_fam_documento_identidad	Número de documento de identidad de la carga familiar del postulante.	nvarchar	50	NO	NO	NO	NO	NO	NO
		5	car_fam_pais_nacimiento	País de nacimiento de la carga familiar del postulante.	nvarchar	150	NO	NO	NO	NO	NO	NO
		6	car_fam_fecha_nacimiento	Fecha de nacimiento de la carga familiar del postulante.	datetime	8	NO	NO	NO	NO	NO	NO
		7	car_fam_discapacidad	Especifica si la carga familiar tiene o no una discapacidad.	bit	1	NO	NO	NO	NO	NO	NO
		8	car_fam_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	NO
		9	car_fam_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	NO
		10	car_fam_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	NO
		11	car_fam_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	NO
		12	car_fam_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO

Pos_Referencias_Personales	Almacena la información sobre las referencias personales que incluye un postulante cuando aplica a un cargo de la empresa.	1	ID_REF_PERSONAL	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_postulante	Identificador de la tabla EMP_POSTULANTE. Especifica a que postulante se asocia la referencia personal creada.	int	4	NO	NO	SI	NO	NO	NO
		3	ref_pers_nombres	Nombres completos de la persona creada como referencia del postulante.	nvarchar	500	NO	NO	NO	NO	NO	NO
		4	ref_pers_telefono	Número de teléfono de la referencia personal del postulante.	nvarchar	50	NO	NO	NO	NO	NO	NO
		5	ref_pers_actividades	Actividad a la que se dedica la referencia personal del postulante.	nvarchar	500	NO	NO	NO	NO	NO	NO
		6	ref_pers_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	NO
Procesos	Almacena los diferentes procesos que son ejecutados en la empresa.	1	ID_PROCESO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_empresa	Identificador de la tabla Empresa. Especifica a que empresa pertenece el proceso creado.	int	4	NO	NO	SI	NO	NO	
		3	pro_descripcion	Descripción del proceso que se incorpora en la empresa.	nvarchar	500	NO	NO	NO	NO	NO	
		4	pro_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	
Requisitos	Almacena los diferentes requisitos necesarios para cumplir con un proceso manejado en la empresa.	1	ID_REQUISITO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_proceso	Identificador de la tabla Procesos. Especifica los requisitos necesario para completar un proceso.	int	4	NO	NO	SI	NO	NO	
		3	req_descripcion	Descripción del requisito necesario para el proceso.	nvarchar	500	NO	NO	NO	NO	NO	
		4	req_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	
		5	req_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	
		6	req_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	
		7	req_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	
		8	req_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	
Responsabilidades_Cargo	Especifica las responsabilidades laborales que tendrán un empleado al cumplir con un cargo específico.	1	ID_RESPONSABILIDADES_CARGO	Identificador de la tabla.	int	4	NO	SI	NO	SI	NO	
		2	id_cargo	Identificador de la tabla Cargo. Especifica las responsabilidades que tendrá que cumplir la persona en el cargo.	int	4	NO	NO	SI	NO	NO	
		3	resp_descripcion	Descripción de la responsabilidad que tiene el personal en el cargo.	nvarchar	500	NO	NO	NO	NO	NO	
		4	resp_usuario_creacion	Usuario que estaba logueado al momento de crear un nuevo registro	nvarchar	500	NO	NO	NO	NO	NO	
		5	resp_fecha_creacion	Fecha del sistema al momento de agregar un nuevo registro	datetime	8	NO	NO	NO	NO	NO	
		6	resp_usuario_modificacion	Usuario que estaba logueado al momento de modificar un registro	nvarchar	500	SI	NO	NO	NO	NO	
		7	resp_fecha_modificacion	Fecha del sistema al momento de modificar un registro	datetime	8	SI	NO	NO	NO	NO	
		8	resp_eliminado	Descripción si el registro esta eliminado o no	char	1	NO	NO	NO	NO	NO	
Sucursal	Lista las sucursales que tienen las empresas.	1	ID_SUCURSAL	Identificador de la tabla Sucursal.	int	4	NO	SI	NO	SI	NO	
		2	id_empresa	Identificador de la empresa a la cual pertenece la sucursal.	int	4	NO	NO	SI	NO	NO	
		3	suc_nombre	Nombre de la sucursal.	nvarchar	500	NO	NO	NO	NO	NO	
		4	suc_eliminado	Especifica si el registro esta eliminado (S/N).	char	1	NO	NO	NO	NO	NO	

## **2.5. Diseño de Procesos**

### 2.5.1. Diagrama de Actividades

El modelo de actividades es usado para proporcionar una descripción gráfica de la funcionalidad del sistema. El diagrama detalla el comportamiento de los procedimientos, transacciones o casos de uso que disponga un sistema (Drake).

Se muestran a continuación los diferentes diagramas de actividades correspondientes al sistema "Gestión de Personal". Dichas actividades son las que tiene opción los usuarios registrados en el sistema.

En la siguiente ilustración se visualiza el diagrama de actividad para el proceso de mantenimiento de cargos.

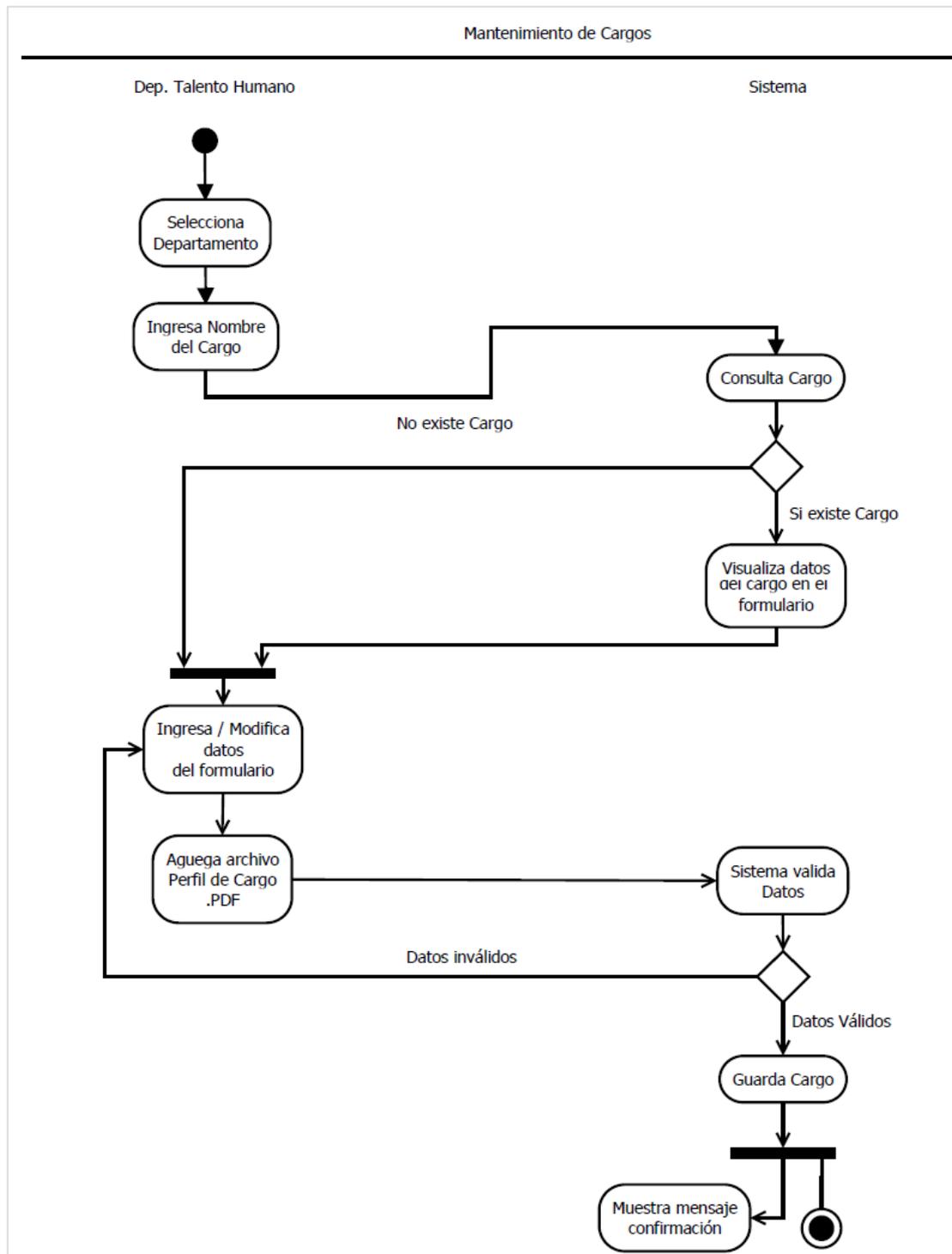


Ilustración 50 Diagrama Actividades - Mantenimiento Cargos

En la siguiente ilustración se muestra la secuencia que tiene que seguir el usuario para realizar el proceso de registro de postulantes desde el sistema "Selección de Personal".

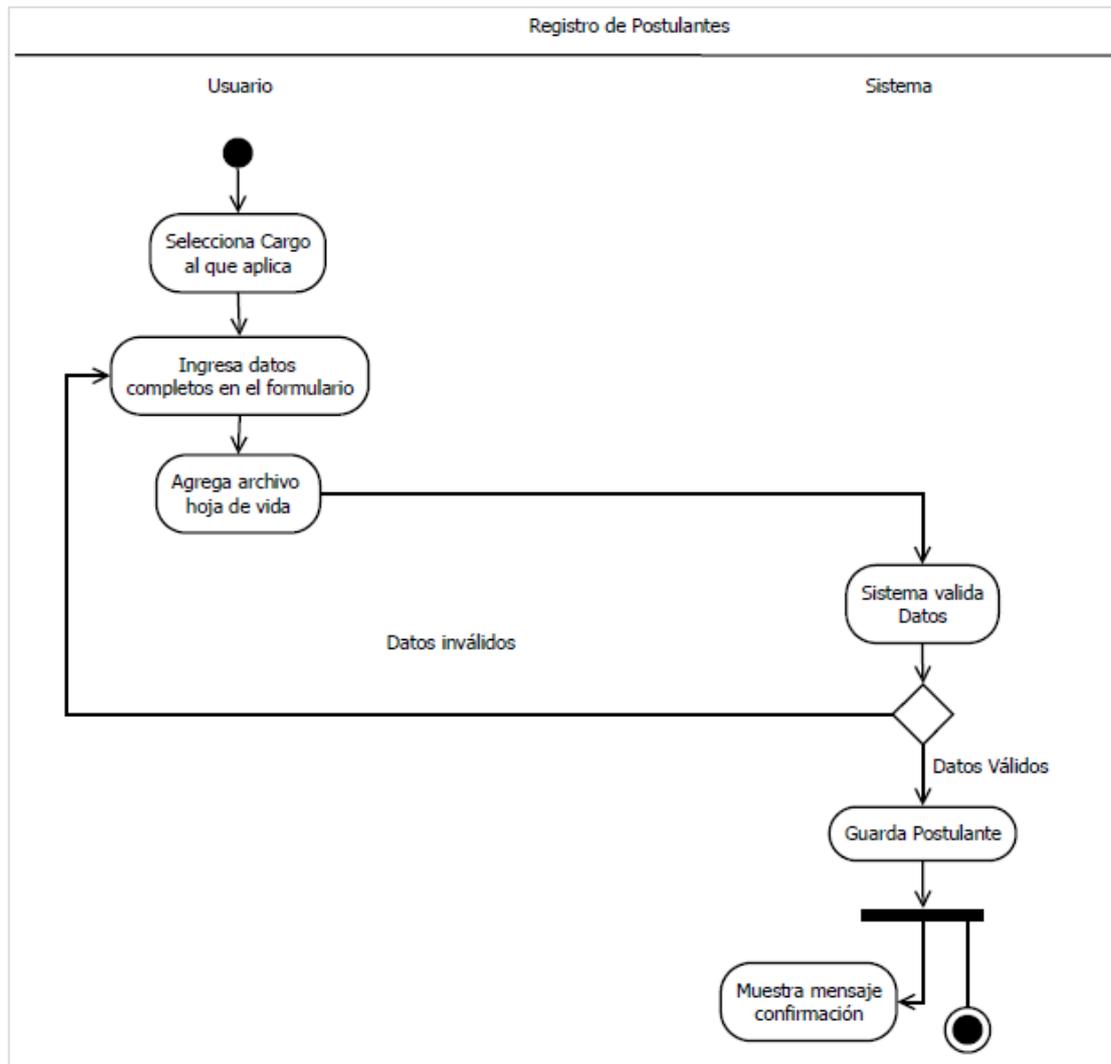


Ilustración 51 Diagrama Actividades Registro Postulantes

En la siguiente ilustración se detalla la secuencia que tiene que seguir el usuario para realizar el proceso de carga de archivos o documentos en la carpeta digital del empleado.

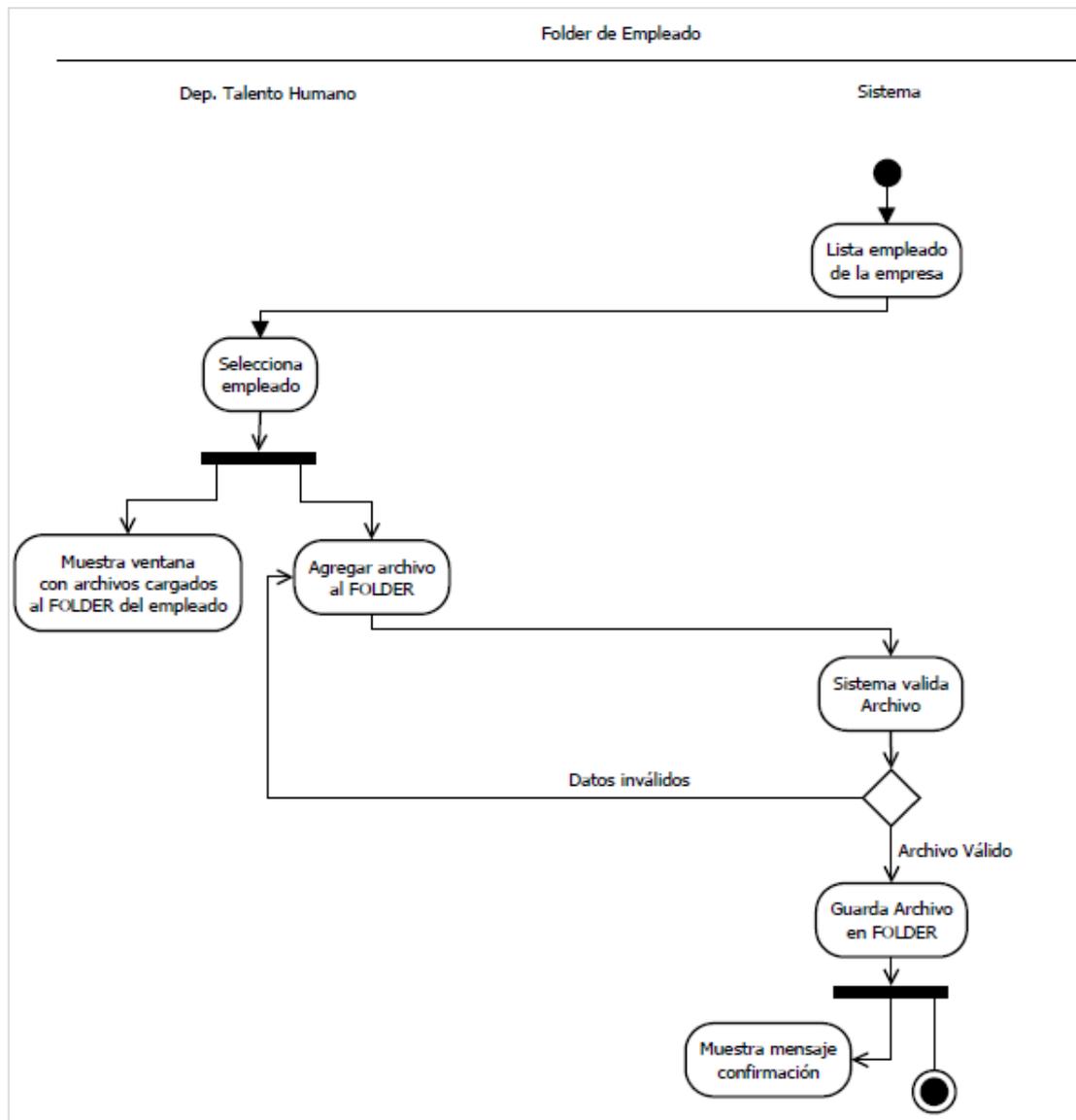


Ilustración 52 Diagrama Actividades. Folder Empleado

En la siguiente imagen se ilustra la secuencia que tiene que seguir el usuario para realizar el proceso de confirmación de referencias de los postulantes ingresados en el sistema.

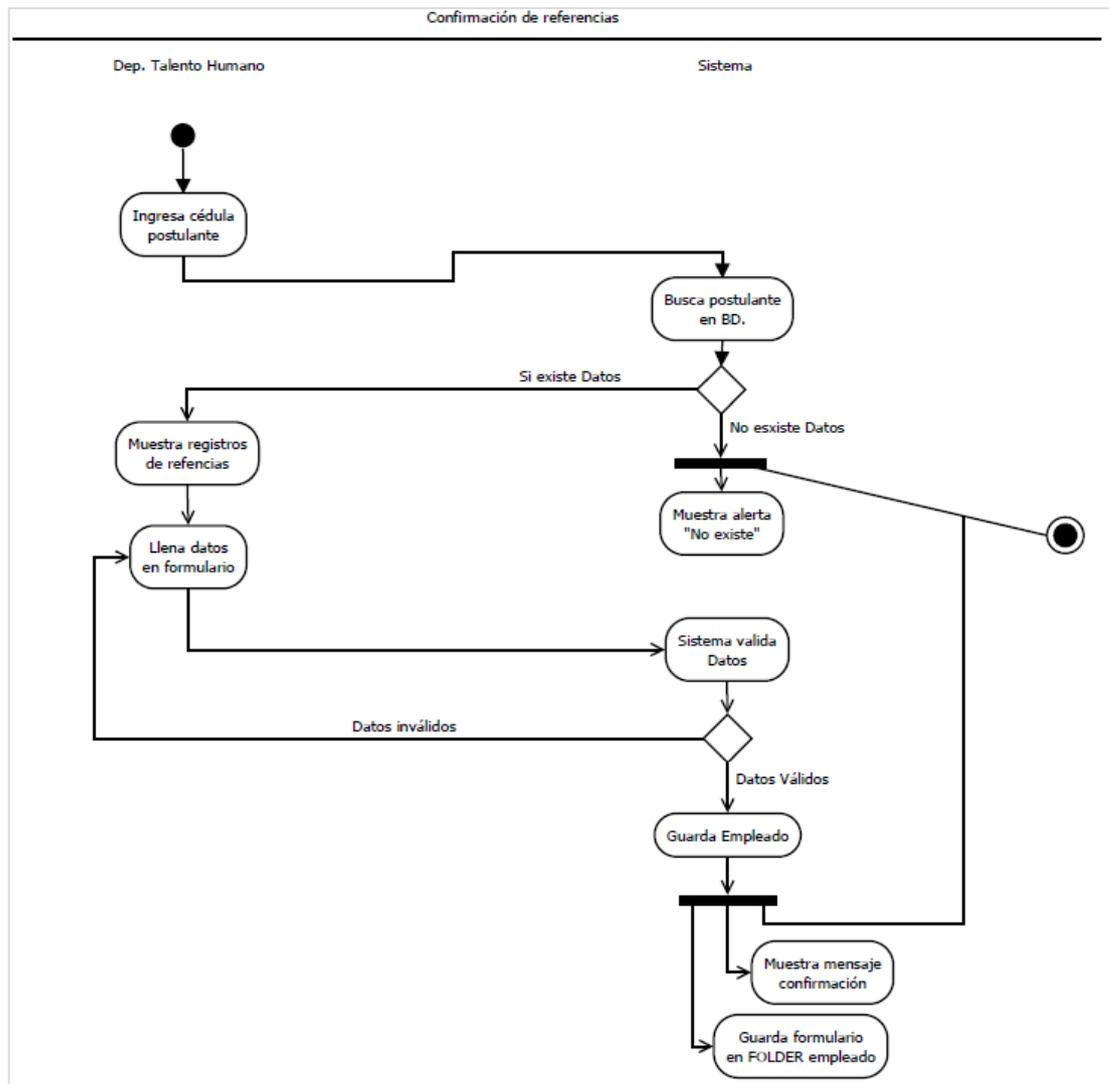


Ilustración 53 Diagrama Actividades. Confirmación Referencias

En la siguiente ilustración se detalla la secuencia de pasos que tiene que seguir el usuario para completar el proceso de registro de empleados en el sistema “Selección de Personal”.

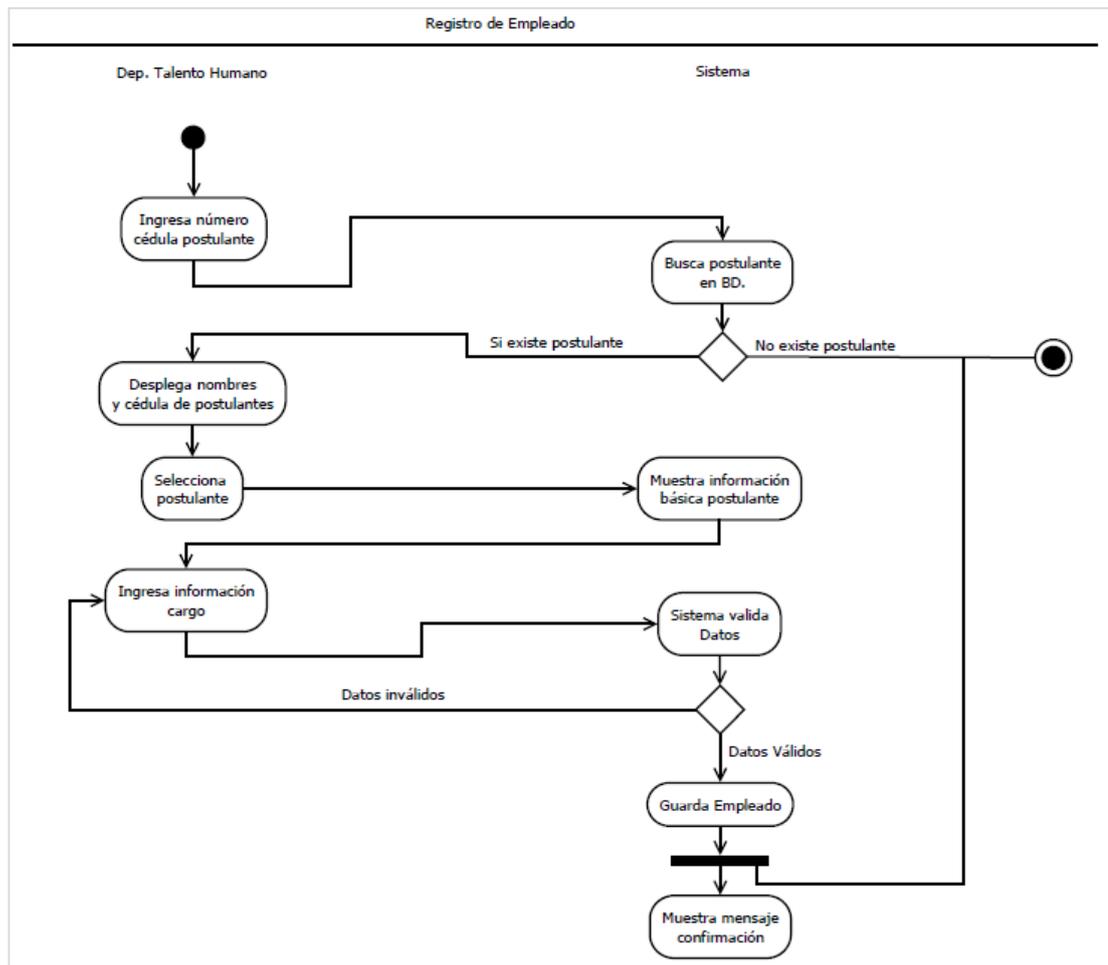


Ilustración 54 Diagrama Actividades. Registro Empleado

## 2.6. Diseño Arquitectónico

### 2.6.1. Diagrama de despliegue

Diagrama que se utiliza para presentar una perspectiva de la arquitectura del sistema en tiempo de ejecución. Muestra la configuración de los elementos de hardware, llamados nodos, y muestra cómo los elementos y artefactos del software se conectan en esos nodos.

En la siguiente ilustración se muestra la arquitectura del sistema “Gestión de Selección”. El sistema contempla el uso del servidor web “Internet Information Server (IIS)” para alojar el sitio web, conectado con un servidor de base de datos SQL Server versión 2016 para almacenar la base de datos y la utilización de navegadores de internet compatibles con la versión de HTML 5 para la utilización del sitio web.

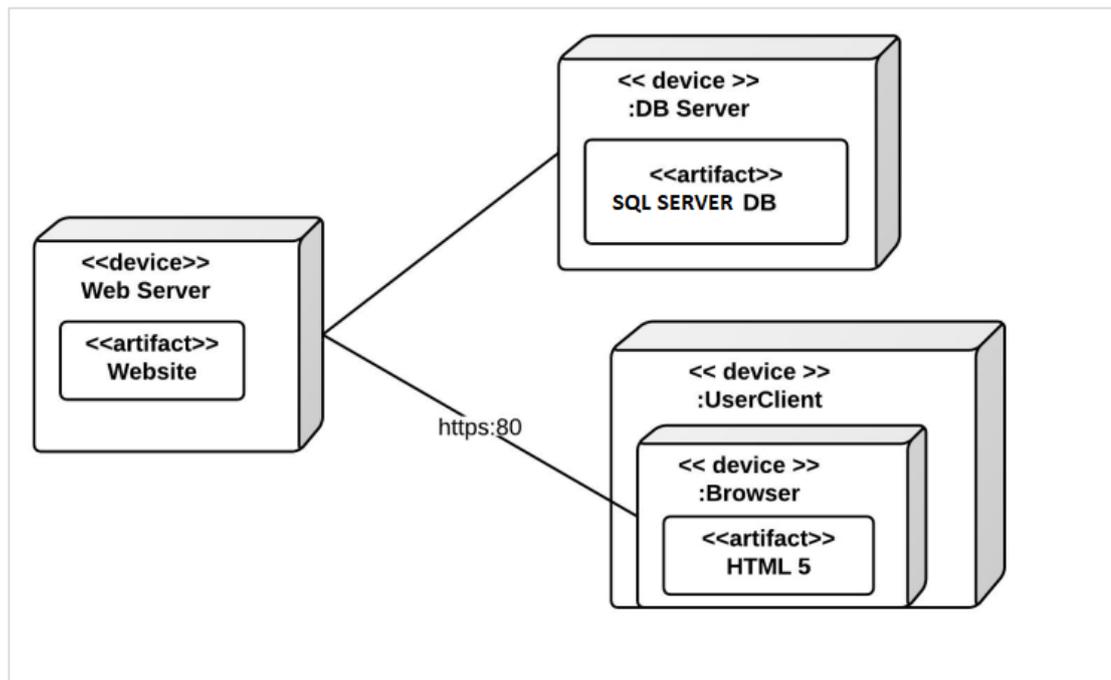


Ilustración 55 Diagrama de Despliegue

### 2.6.3. Diseño de interfaces.

El diseño de interfaces es el diseño de las aplicaciones de software en los cuales se define la estructura, aspectos específicos de la interfaz, así como la navegación que dispone la aplicación (James, 2011).

Se presenta a continuación los diseños de interfaz que se visualizan los usuarios del sistema de "Gestión de Personal". Dichas ilustraciones muestran los elementos y opciones que dispone el usuario para realizar un proceso del sistema.

La ilustración 56 muestra el diseño de pantalla que permite al usuario ingresar una solicitud de trabajo. Está compuesta por 8 secciones: datos personales, datos familiares, instrucción o educación recibida, ocupación actual, ocupaciones o trabajos anteriores, referencias personales y objetivos e inspiraciones.

### Solicitud de Trabajo

Para el cargo de: Administrativo

---

#### 1. DATOS PERSONALES

Apellidos:

Nombres:

Cédula Ciudadanía N°:

Dirección Domiciliaria:

Teléfono:  Celular:

---

#### 2. DATOS FAMILIARES

Nombre del conyuge:  Cédula Ciudadanía N° (Conyuge):

Fecha de nacimiento conyuge:

Número de hijos:

Nombres:  Documento identidad:  País Nacimiento:  Fecha Nac.

---

#### 3. INSTRUCCION

Nivel Instrucción:  Establecimiento:  Último curso aprobado:  Título obtenido:

---

#### 4. OCUPACION ACTUAL

Establecimiento:  Nombre jefe inmediato:  Cargo:

Fecha ingreso:     Sueldo:

Descripción de tareas que desempeña:

---

#### 5. OCUPACIONES ANTERIORES

Establecimiento	Fecha ingreso	Fecha Salida	Cargo	Sueldo	Jefe inmediato	Motivo Salida

---

#### 6. REFERENCIAS PERSONALES

Nombres Completos:  Teléfono:  Actividades a las que se dedica:

---

#### 7. OBJETIVOS Y ASPIRACIONES

Diga por qué medio se informó de nuestro llamamiento?

Indique para qué tipo de trabajo se siente más preparado y competente?

Aspiración Salarial:  Disponibilidad inmediata:  Turnos rotativos:

[Guardar Postulante](#)

Ilustración 56 Interfaz Registro Postulantes

En la siguiente ilustración se observa la pantalla que tiene disponible el usuario para el registro de empleados en el sistema de “Gestión de Personal”. En la pantalla el usuario debe seleccionar el postulante al cual se le ingresará como empleado, posterior ingresar información sobre el cargo, salario y fecha de ingreso para finalmente cargar los archivos de los documentos requeridos para el proceso.

**REGISTRO DE EMPLEADO**

**1. DATOS PERSONALES**

Apellidos:

Nombres:

Cédula:

Dirección:

Teléfono:

**2. INFORMACIÓN DEL CARGO**

Sucursal:

Cargo a desempeñar:

Sueldo:

Fecha Ingreso:

**3. Requisitos de Admisión**

HOJA VIDA ACTUALIZADA		COPIAS A COLOR DE CÉDULA DE CUIDADANÍA	
COPIAS CERTIFICADO DE VOTACIÓN		CERTIFICADO DE SALUD ACTUALIZADO	

Registrar Empleado

Ilustración 57 Interfaz Registro de Empleado

En la siguiente ilustración se observa la pantalla que tiene el usuario para el ingreso de cargos en el sistema “Gestión de Personal”. Para crear nuevos cargos se debe ingresar el nombre, departamento al cual pertenece, nombre del jefe inmediato, misión, responsabilidades, formación académica, conocimientos específicos, capacitación adicional, nivel de experiencia y las habilidades requeridas.

**Cargos**

Nombre del Cargo:

Departamento:

Jefe Inmediato:

**1. Misión y Responsabilidades**

Misión:

Responsabilidades:  +

**2. Formación Académica**

+

**3. Conocimientos Específicos**

+

**4. Capacitación Adicional**

+

**5. Nivel Experiencia**

+

**6. Habilidades y Destrezas requeridas**

+

 **Grabar Cargo**

Ilustración 58 Interfaz Ingreso Cargos

## **CAPÍTULO 3: CODIFICACIÓN, PRUEBAS Y VALIDACIÓN**

En este capítulo se aborda la programación usada para el desarrollo adaptable del sistema web, así como las pruebas funcionales y de interfaz realizadas al sistema. También se incorpora los resultados obtenidos de las pruebas realizadas y las correcciones realizadas para su correcto funcionamiento o visualización del sistema.

### **3.1. Codificación del sistema**

A continuación, se ilustra el código usado en las principales páginas del sistema. Se demuestra la aplicación de las características de Bootstrap frente a la programación estándar usada en el lenguaje HTML para cumplir con las funcionalidades que incorpora el nuevo framework Bootstrap.

```

9 <form name="frmCargos" method="post">
10 <div class="contenedor_Pagina" id="contenedorCombos">
11 <div class="row contenedor_cargos bordes">
12 <div class="row espacio_cargo">
13 <div class="col-xs-12 col-sm-9 col-md-6 col-md-offset-3 col-sm-offset-3">
14 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
15 <label for="nombreCargo">Nombre de Cargo:</label>
16 </div>
17 <div class="col-xs-12 col-sm-8 col-md-8">
18 <input type="text" id="nombreCargo" name="nombreCargo" value="@ViewBag.nombreCargo" class="form-control bordes_campos" placeholder="Nombre de Cargo:" style="width: 100%" />
19 <input type="hidden" id="idCargo" value="@ViewBag.idCargo" />
20 </div>
21 </div>
22 </div>
23 <div class="row espacio_cargo">
24 <div class="col-xs-12 col-sm-9 col-md-6 col-md-offset-3 col-sm-offset-3">
25 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
26 <label for="Departamentos">Departamento:</label>
27 </div>
28 <div class="col-xs-12 col-sm-8 col-md-8">
29 @Html.DropDownList("Departamentos", (IEnumerable<SelectListItem>)ViewBag.Departamentos, "Seleccione Departamento", new { id = "Departamentos", name = "Departamentos", @class
30 </div>
31 </div>
32 </div>
33 <div class="row espacio_cargo">
34 <div class="col-xs-12 col-sm-9 col-md-6 col-md-offset-3 col-sm-offset-3">
35 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
36 <label for="nombreJefe">Jefe inmediato:</label>
37 </div>
38 <div class="col-xs-12 col-sm-8 col-md-8">
39 <input type="text" id="nombreJefe" name="nombreJefe" placeholder="Jefe inmediato" class="form-control bordes_campos" style="width: 100%" />
40 </div>
41 </div>
42 </div>
43 </div>
44 <section>...</section>
287 <div class="row">...</div>
295 </div>
296 </form>

```

Ilustración 59 Extracto Código Bootstrap Página Cargos

En la imagen anterior se muestra un extracto del código HTML potenciado con las propiedades y características de Bootstrap, utilizado en la pantalla registro de cargos. Específicamente, en la línea 12 utiliza la clase “row” misma que contiene tres columnas para visualizaciones en equipos de escritorio o laptops.

```

9 <form name="frmCargos" method="post">
10 <div class="contenedor_Pagina" id="contenedorCargos">
11 <div class="row contenedor_cargos bordes">
12 <div class="row espacio_cargo">
13 <div class="col-xs-12 col-sm-9 col-md-6 col-md-offset-3 col-sm-offset-3">
14 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
15 <label for="nombreCargo">Nombre de Cargo:</label>
16 </div>
17 <div class="col-xs-12 col-sm-8 col-md-6">
18 <input type="text" id="nombreCargo" name="nombreCargo" value="@viewBag.nombreCargo" placeholder="Nombre de Cargo:" style="width: 100%" />
19 </div>
20 </div>
21 </div>
22 <div class="row espacio_cargo">
23 <div class="col-xs-12 col-sm-8 col-md-6">
24 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
25 <label for="nombreCargo">Nombre de Cargo:</label>
26 </div>
27 <div class="col-xs-12 col-sm-8 col-md-6">
28 <input type="text" id="nombreCargo" name="nombreCargo" value="@viewBag.nombreCargo" placeholder="Nombre de Cargo:" style="width: 100%" />
29 </div>
30 </div>
31 </div>
32 </div>
33 <div class="row espacio_cargo">
34 <div class="col-xs-12 col-sm-9 col-md-6 col-md-offset-3 col-sm-offset-3">
35 <div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo">
36 <label for="nombreJefe">Jefe inmediato:</label>
37 </div>
38 <div class="col-xs-12 col-sm-8 col-md-6">
39 <input type="text" id="nombreJefe" name="nombreJefe" placeholder="Jefe inmediato" class="form-control bordes_campos" style="width: 100%" />
40 </div>
41 </div>
42 </div>
43 </div>
44 </section>...</section>
287 <div class="row">...</div>
295 </div>
296 </form>

```

Bootstrap	HTML Estandar
<pre> &lt;div class="row espacio_cargo"&gt;   &lt;div class="col-xs-12 col-sm-8 col-md-6 col-md-offset-3 col-sm-offset-3"&gt;     &lt;div class="col-xs-8 col-sm-4 col-md-4 direccion_etiquetas label_izquierdo"&gt;       &lt;label for="nombreCargo"&gt;Nombre de Cargo:&lt;/label&gt;     &lt;/div&gt;     &lt;div class="col-xs-12 col-sm-8 col-md-6"&gt;       &lt;input type="text" id="nombreCargo" name="nombreCargo" value="@viewBag.nombreCargo" placeholder="Nombre de Cargo:" style="width: 100%" /&gt;     &lt;/div&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	<pre> &lt;table&gt;   &lt;tr&gt;     &lt;td class="label_izquierdo"&gt;       &lt;label for="nombreCargo"&gt;Nombre de Cargo:&lt;/label&gt;     &lt;/td&gt;     &lt;td&gt;       &lt;input type="text" id="nombreCargo" name="nombreCargo" value="@viewBag.nombreCargo" placeholder="Nombre de Cargo:" style="width: 100%" /&gt;     &lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt; </pre>

Ilustración 60 Comparación Bootstrap - HTML

En la ilustración 60 se compara el código necesario para generar un diseño visualizado en una fila con dos columnas. En el código Bootstrap se utiliza una clase “row” que especifica la creación de una nueva fila y el uso de las clases “col-xs-12 col-sm-6 col-md-4” para definir la cantidad de columnas que utiliza cada elemento o grupo de elementos en la fila creada. Por otro lado, en el código HTML estándar, se utiliza el elemento tipo “<table>” que especifica la creación de una tabla, misma que define la creación de una fila con el elemento “<tr>”, y la utilización del elemento “<td>” para definir la creación de columnas.

```

<html>
  <div class="contenedor_cargos">
    <div class="centrar">
      <table>
        <tr>
          <td class="label_izquierdo">
            <label for="nombreCargo">Nombre de Cargo:</label>
          </td>
          <td>
            <input type="text" id="nombreCargo" name="nombreCargo" value="@ViewBag.nombreCargo" class="form-control bordes_campos" placeholder="Nombre de Cargo:" style="width: 100%" />
            <input type="hidden" id="idCargo" value="@ViewBag.idCargo" />
          </td>
        </tr>
        <tr>
          <td class="label_izquierdo">
            <label for="Departamentos">Departamento:</label>
          </td>
          <td>
            @Html.DropDownList("Departamentos", (IEnumerable<SelectListItem>)ViewBag.Departamentos, "Seleccione Departamento", new { id = "Departamentos", name = "Departamentos", @class
          </td>
        </tr>
        <tr>
          <td class="label_izquierdo">
            <label for="nombreJefe">Jefe inmediato:</label>
          </td>
          <td>
            <input type="text" id="nombreJefe" name="nombreJefe" placeholder="Jefe inmediato" class="form-control bordes_campos" style="width: 100%" />
          </td>
        </tr>
      </table>
    </div>
    <div class="centrar">
    <div class="centrar">
    <div class="centrar">
    <div class="centrar">
    <div class="centrar">
    <div class="centrar">
  </div>
</html>

```

Ilustración 61 Extracto código página Registro Cargos usando HTML

### **3.2. Pruebas de Interfaz.**

Esta sección se enfoca en realizar un test o prueba sobre la interfaz adaptable del sistema en diferentes dispositivos móviles. Las pruebas se dirigirán hacia las principales pantallas del sistema, dichas pantallas son: Registro de Postulante, Registro de Cargos y Registro de Empleados.

Para realizar las pruebas de interfaz adaptables se utiliza el siguiente enlace <http://www.responsinator.com/>. Dicho enlace muestra una amplia gama de dispositivos móviles en los cual se realiza las pruebas. En esta ocasión se analiza el diseño en los siguientes dispositivos móviles: iphone, ipad y equipos con sistema android.

En la siguiente tabla se resume las pruebas realizadas, los objetos analizados, los dispositivos utilizados y los resultados obtenidos.

Las capturas de pantalla con el resultado de cada una de las pruebas de adaptabilidad se muestran en el Anexo 5: Capturas de pantalla pruebas de interfaz adaptable. En el anexo también se muestran las capturas de pantalla de las páginas que en las pruebas iniciales devolvieron algún error u observación y que posterior fueron corregidas.

		Dispositivo						
		Iphone 6-8		Android		Ipad		
		Pantalla (px)		Pantalla (px)		Pantalla (px)		
Página	Evaluación Interfaz	375	667	412	684	768	1024	Correcciones
Registro_Cargos	Campos de Texto	OK	OK	OK	OK	X	X	Modificar tamaño de etiquetas (1024px)
	Lista de Valores	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
	Secciones del formulario	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
	Grillas	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
	Menú	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
		Ilustración 171 RC.iphone6_8 portrait 375px		Ilustración 173 RC.android(pixel2) portrait 412px		Ilustración 175 RC.ipad portrait 768px		
		Ilustración 172 RC.iphone6_8 landscape 667px		Ilustración 174 RC.android(pixel2) landscape 684px		Ilustración 176 RC.ipad landscape 1024px		
Registro_Empleados	Campos de Texto	X	X	X	X	X	X	Verificar posición íconos, y modificar estructura campos.
	Lista de Valores	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
	Secciones del formulario	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
	Grillas	X	X	X	X	X	X	No contiene
	Menú	OK	OK	OK	OK	X	OK	Cambiar tamaño máximo para cambiar de dispositivo (768px).
		Ilustración 177 RE.iphone6_8 portrait 375px		Ilustración 179 RE.android(pixel2) portrait 412px		Ilustración 181 RE.ipad portrait 768px		
		Ilustración 178 RE.iphone6_8 landscape 667px		Ilustración 180 RE.android(pixel2) landscape 684px		Ilustración 182 RE.ipad landscape 1024px		
Registro_Postulantes	Campos de Texto	OK	OK	X	X	OK	X	Modificar estructura campos cabecera y tamaño etiquetas.
	Lista de Valores	X	OK	OK	OK	OK	OK	Campo muy grande (375px)
	Secciones del formulario	OK	OK	OK	OK	OK	OK	
	Grillas	X	OK	OK	OK	OK	OK	Tamaño grilla muy grande (375px)
	Menú	X	X	X	X	X	X	No contiene
		Ilustración 165 SE.iphone6_8 portrait 375px		Ilustración 167 SE.android(pixel2)portrait 412px		Ilustración 169 SE.ipad portrait 768px		
		Ilustración 166 SE.iphone6_8 landscape 667px		Ilustración 168 SE.android(pixel2)landscap e 684px		Ilustración 170 SE.ipad landscape 1024px		

Tabla 9 Pruebas de Interfaz y Adaptación

### 3.3. Validación del sistema

Las pruebas de funcionamiento del sistema se realizaron enfocadas en los casos de uso detallados en secciones anteriores del documento. Para el desarrollo de las pruebas de funcionamiento se utilizó el formato que se muestra en la ilustración 61.

Nombre proyecto:						
ID Test:				Fecha ejecución:		
Nivel prioridad:						
Nombre Módulo:						
Título Test:						
Descripción:						
Precondiciones:						
Dependencias:						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
Post-condiciones:						

Ilustración 62 Formato Pruebas Funcionamiento

En las siguientes ilustraciones se visualizan los casos de uso que fueron validados y cuáles fueron los resultados alcanzados.

Nombre proyecto: Gestión de selección de personal						
ID Test:	C.U. PRYLTCU01			Fecha ejecución:	16/1/2018	
Nivel prioridad:	Alto					
Nombre Módulo:	Administración					
Título Test:	Prueba ingreso al sistema					
Descripción:	Prueba sobre el ingreso de los usuarios en la página del sistema.					
Precondiciones: Usuario tenga un usuario y contraseña válido.						
Dependencias: Ninguna						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia la página de inicio.	ebarrera	Usuario debería estar habilitado.	Credenciales ingresadas correctamente. Ingresar al sistema	Ok	
2	Ingresar un usuario válido.	Ebarrera1%				
3	Insertar una contraseña válida.					
4	Clic en el botón "Ingresar".					
Post-condiciones:						

Ilustración 63 Prueba funcionalidad Ingreso sistema

Nombre proyecto:	Gestión de selección de personal					
ID Test:	C.U. PRYTLCU02		Fecha ejecución:	16/1/2018		
Nivel prioridad:	Alto					
Nombre Módulo:	Administración					
Título Test:	Prueba mantenimiento nivel instrucción					
Descripción:	Prueba sobre la inserción de los niveles de instrucción en el sistema.					
Precondiciones:	Usuario este logueado correctamente en el sistema.					
Dependencias:	Ninguna					
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página "Configuración"	Primaria	Registro debe guardarse	Registro guardado exitosamente y visualizado en la grilla.	OK	
2	Abrir sección "Nivel Instrucción"		Mostrar registro en la grilla			
3	Pulsar botón nuevo.					
4	Ingresar descripción de nivel instrucción.					
5	Clic botón "Guardar"					
6	Seleccionar fila a editar.	PRIMARIA	Datos actualizados debe guardarse y mostrar el cambio en la grilla.	Registro actualizado y visualizado en la grilla.	OK	
7	Precionar botón editar					
8	Editar descripción de nivel instrucción					
9	Clic botón "Guardar"					
10	Seleccionar fila a eliminar	PRIMARIA	Dato eliminado y debe desaparecer de la grilla	Registro eliminado exitosamente ya no se muestra en grilla.	OK	
11	Pulsar botón eliminar					
12	confirmar eliminación					
Post-condiciones:						

Ilustración 64 Prueba funcionalidad Mantenimiento Instrucción

ID Test:	C.U. PRYTLCU06		Fecha ejecución:	16/1/2018		
Nivel prioridad:	Alto					
Nombre Módulo:	Administración					
Título Test:	Prueba configuración de habilidades y destrezas por formulario					
Descripción:	Prueba sobre la selección de habilidades y destrezas necesarias para completar un formulario.					
Precondiciones:	Usuario este logueado correctamente en el sistema.					
Dependencias:	Haber ingresado información en la base de conocimientos de habilidades y destrezas. Tener ingresado los formularios en B.D.					
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página "Configuración"	Confirmación referencias	Registro sea guardado y mostrado en la grilla.	Registro guardado exitosamente y visualizado en la grilla.	OK	
2	Abrir sección "Configuración habilidades por formulario"	Toma de decisiones				
3	Seleccionar el formulario					
4	Pulsar botón nuevo					
5	Seleccionar habilidad					
6	Clic botón "Guardar"					
7	Seleccionar registro a eliminar	Toma de decisiones	Registro sea eliminado de B.D. y no se muestre en la grilla.	Registro eliminado exitosamente y no se encuentra en la grilla.		
8	Pulsar botón eliminar					
9	Confirmar eliminación de registro					
Post-condiciones:						

Ilustración 65 Prueba funcionalidad Habilidades formulario

Nombre proyecto: Gestión de selección de personal						
ID Test:	C.U. PRYTLCU08	Fecha ejecución:			16/1/2018	
Nivel prioridad:	Alto					
Nombre Módulo:	Mantenimiento de Cargos					
Título Test:	Prueba ingreso de cargo.					
Descripción:	Prueba sobre la funcionalidad del sistema al ingresar un nuevo cargo.					
Precondiciones: Usuario este logueado correctamente en el sistema.						
Dependencias: Haber ingresado información de departamentos y haber cargado base de conocimientos sobre habilidades.						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página	Asistente Diseño	Se guarde el registro en	Registro guardado	OK	
2	Ingresar Datos cargo	Departamento Diseño	B.D. y que se muestre	exitosamente y		
3	Seleccionar habilidades y destrezas requeridas para el cargo	Jefe Juan Perez	el registro en la grilla.	reflejado en la grilla.		
4	Agregar archivo perfil del					
5	Clic botón Grabar Cargo					
Post-condiciones:						

Ilustración 66 Prueba funcionalidad Ingreso Cargos

Nombre proyecto: Gestión de selección de personal						
ID Test:	C.U. PRYTLCU09	Fecha ejecución:			16/1/2018	
Nivel prioridad:	Alto					
Nombre Módulo:	Postulantes					
Título Test:	Prueba Registro Postulante.					
Descripción:	Prueba sobre la funcionalidad del sistema al registrar un nuevo postulante.					
Precondiciones: Haber ingresado información sobre cargos y niveles de instrucción.						
Dependencias: Ingreso de información sobre tipos de documento de identidad.						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia la página de registro	Asistente Diseño	Registro se guarde en	El registro se guarda	OK	
2	Seleccionar el cargo al aplicar	Cédula: 0102085800	B.D.	exitosamente		
3	Ingresar número de documento identidad	Juan Carlos Maldonado				
4	Ingresar información postulante	Aspiración 900				
5	Ingresar aspiración salarial, disponibilidad de tiempo y dispuesto	Turnos rotativos NO Disponibilidad				
6	Clic botón "Guardar formulario"	inmediata NO				
Post-condiciones:						

Ilustración 67 Prueba funcionalidad Registro Postulantes

Nivel prioridad:	Media					
Nombre Módulo:	Postulantes					
Título Test:	Prueba Confirmación de Referencias.					
Descripción:	Prueba de la página de confirmación de referencias de los postulantes ingresados.					
Precondiciones: Haber ingresado información sobre postulantes y la información del historial laboral.						
Dependencias:						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página "Listado Postulantes"	Juan Carlos Maldonado	Formulario sea guardado en formato PDF y se visualice en la grilla.	Formulario guardado exitosamente, se muestra icono del archivo en la grilla.	OK	
2	Seleccionar postulante	Memo desing				
3	Pulsar botón "Confirmación Referencias"					
4	Seleccionar una referencia laboral					
5	Clic botón "Validar Referencia"					
6	Calificar habilidades / referencias del formulario					
7	Clic botón "Guardar Formulario"					
Post-condiciones:						

Ilustración 68 Prueba funcionalidad Confirmación Referencias

Nombre proyecto: Gestión de selección de personal						
ID Test:	C.U. PRYTLCU11			Fecha ejecución:	16/1/2018	
Nivel prioridad:	Alta					
Nombre Módulo:	Empleados					
Título Test:	Prueba Registro de Empleados					
Descripción:	Prueba de la página de registro de empleados.					
Precondiciones: Haber ingresado información sobre postulantes, sucursales y cargos disponibles.						
Dependencias:						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página Registro Empleado	Juan Carlos Maldonado	El postulante sea	Empleado registrado	OK	
2	Seleccionar postulante	Sueldo 500	registrado como	exitosamente y		
3	Ingresar información del cargo a ocupar	Ingreso 01/02/2018	empleado, y que el	visualizado en la lista		
4	Adjuntar archivos para los requisitos	Sucursal España	empleado se visualice	de empleados.		
5	Clic botón "Registro Empleado"	Cargo Asistente diseño				
Post-condiciones:						
Completar la carga de archivos para cumplir con los requisitos necesarios para ingreso de empleado.						

### Ilustración 69 Prueba funcionalidad Registro Empleados

Nombre proyecto: Gestión de selección de personal						
ID Test:	C.U. PRYTLCU12			Fecha ejecución:	16/1/2018	
Nivel prioridad:	Alta					
Nombre Módulo:	Empleados					
Título Test:	Prueba Folder Empleados					
Descripción:	Prueba de la página de carga de archivos de empleados.					
Precondiciones: Haber registrado empleados en el sistema.						
Dependencias:						
Paso	Acción	Datos	Resultado esperado	Resultado Test	Estado (Ok/Error)	Notas
1	Navegar hacia página Listado Empleados	Juan Carlos Maldonado	Visualizar correctamente	Archivo visualizado	OK	
2	Seleccionar empleado	Archivo Hoja_Vida.PDF	archivo cargado en el	correctamente.		
3	Clic botón "Folder Empleado"		folder del empleado			
4	Clic nombre del archivo a revisar.					
Post-condiciones:						

### Ilustración 70 Prueba funcionalidad Folder Empleado

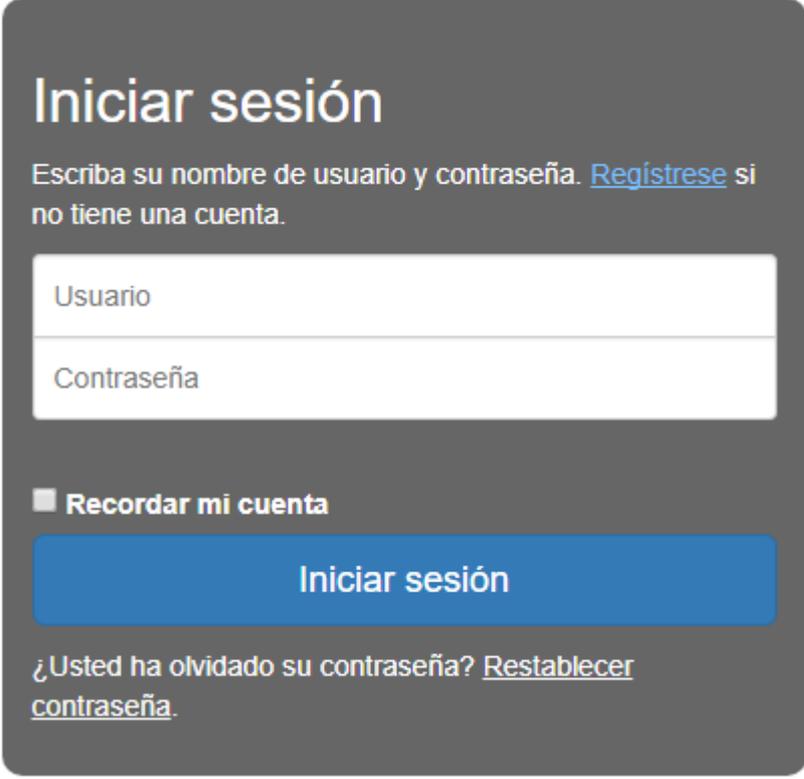
## CAPÍTULO 4: MANUAL DE USUARIO

En este capítulo se incorpora un manual de usuario sobre las funcionalidades del sistema, explicando de manera detallada y respaldada con imágenes cada proceso que se realiza en el sistema.

### 4.1. Ingreso al sistema

Para acceder al sistema se ingresa en el navegador la siguiente dirección web [www.thumano.improled.com.ec](http://www.thumano.improled.com.ec) la cual muestra su página inicial y en caso de ya tener una cuenta registrada ingrese su usuario y contraseña.

En caso de no tener cuenta puede registrarse desde la opción “Regístrese” en la cual debe ingresar su usuario y una contraseña.



El formulario de inicio de sesión tiene un fondo gris oscuro con texto y botones en blanco y azul. El título "Iniciar sesión" está en blanco. El texto de instrucciones y los enlaces "Regístrese si no tiene una cuenta" y "Restablecer contraseña" están en blanco. Los campos de entrada "Usuario" y "Contraseña" tienen un fondo blanco. El botón "Iniciar sesión" es azul con texto blanco. El checkbox "Recordar mi cuenta" tiene un cuadrado gris.

**Iniciar sesión**

Escriba su nombre de usuario y contraseña. [Regístrese](#) si no tiene una cuenta.

Usuario

Contraseña

Recordar mi cuenta

**Iniciar sesión**

¿Usted ha olvidado su contraseña? [Restablecer contraseña](#).

Ilustración 71 Formulario Inicio Sesión

## 4.2. Administración General

Se configura las opciones que estarán disponibles en el sistema para todos los usuarios. La administración del sistema es un tema fundamental y por eso es necesario realizarlo antes de que otros usuarios utilicen el sistema.

En una sola pantalla el usuario tiene disponible los mantenimientos para administrar toda la información de configuración. Es necesario que el usuario navegue hasta el menú “Administración” y seleccione la opción “Configuración”.

### 4.2.1. Nivel de Instrucción

Puede ingresar los diferentes niveles de instrucción o educación que tendrá disponible un usuario al momento de registrar su postulación en uno de los cargos creados en el sistema.

En la siguiente ilustración se visualiza la pantalla de mantenimiento que el usuario encuentra en algunas secciones de este manual incluida esta.

Niveles de Instrucción	
	DESCRIPCIÓN
1	PRIMARIA
2	SECUNDARIA
3	SUPERIOR
4	MAGISTER
5	DOCTORADO

Registros ingresados

Barra de Navegación

Ilustración 72 Mantenimiento de Instrucciones

En la grilla de mantenimiento se encuentra una barra de navegación en la cual tiene disponible las opciones de ingreso, edición, eliminación y búsqueda de datos. De igual manera en la barra de navegación se encuentra disponible la paginación en la cual puede navegar hacia las diferentes páginas de contenido que puede tener la grilla.

#### 4.2.1.1. Ingreso

Posterior a seleccionar el botón de ingresar (+) se despliega una ventana con los campos necesarios para el ingreso (Ver ilustración 73). El usuario tiene que ingresar los datos requeridos y presionar en el botón guardar. Posterior se cierra la ventana de ingreso y se actualiza la información mostrada.

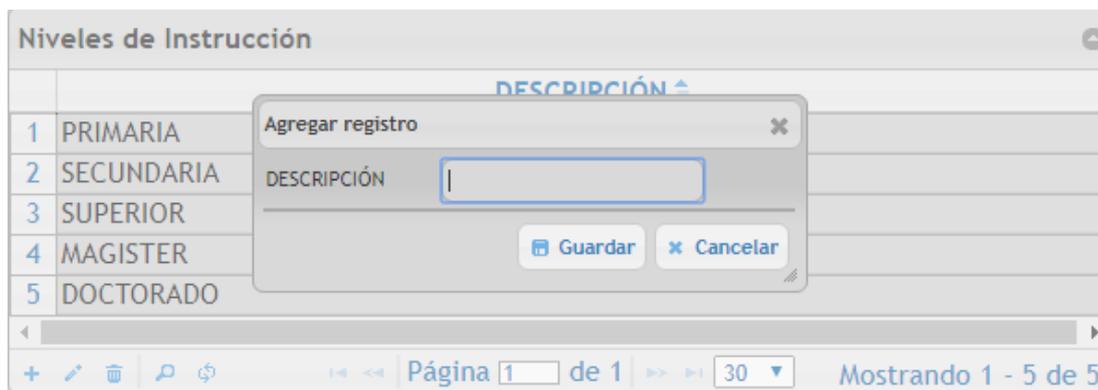


Ilustración 73 Ingreso de Instrucciones

#### 4.2.1.2. Edición

Para editar o eliminar un registro se debe seleccionar el registro a modificar y posterior presionar en los botones de edición que se encuentran en la barra de navegación.

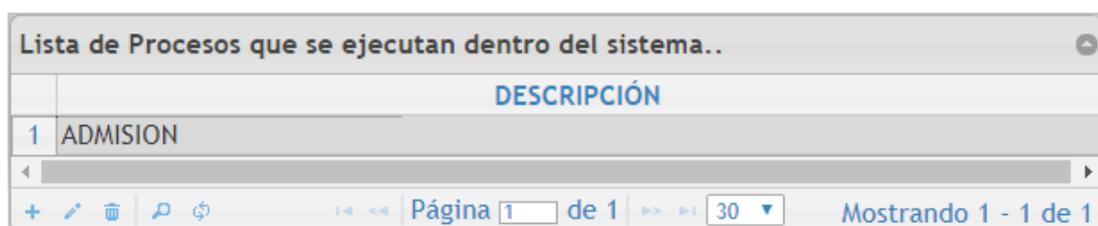


Ilustración 74 Edición de Instrucción

#### 4.2.2. Procesos

Se puede ingresar los diferentes procesos que van a ser manejados en el sistema. En este manual el único proceso que se considera es la admisión.

En la siguiente ilustración se visualiza la pantalla de mantenimiento para la administración de procesos.



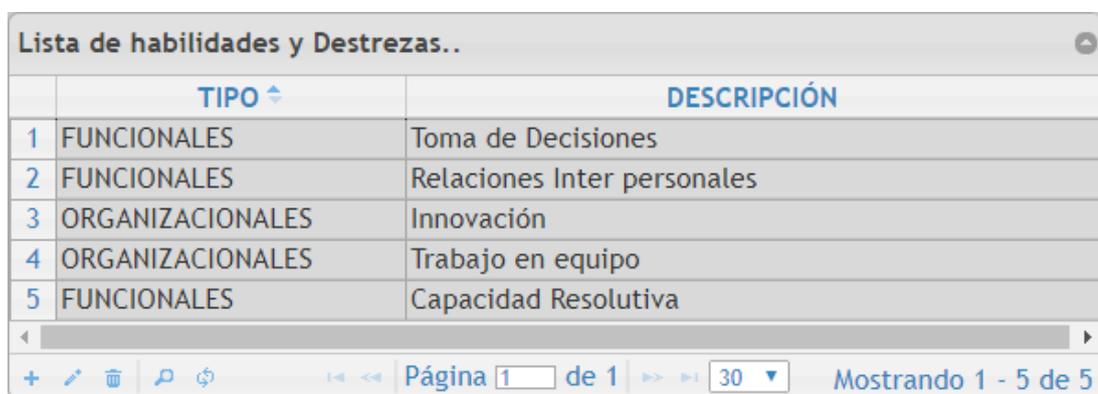
	DESCRIPCIÓN
1	ADMISION

Ilustración 75 Mantenimiento de Procesos.

Al igual que en la sección de “Niveles de Instrucción” se dispone de los diferentes botones de edición para el ingreso, modificación y eliminación del contenido. No se va a indagar a profundidad este tema ya que en este manual sólo se maneja un proceso que es la “Admisión”.

#### 4.2.3. Habilidades y Destrezas.

Se puede realizar un mantenimiento de la base de conocimientos de las habilidades y destrezas que van a ser utilizadas en otras secciones del sistema. Las habilidades o destrezas que ingrese el usuario se clasifican en dos tipos: funcionales y organizacionales.



	TIPO	DESCRIPCIÓN
1	FUNCIONALES	Toma de Decisiones
2	FUNCIONALES	Relaciones Inter personales
3	ORGANIZACIONALES	Innovación
4	ORGANIZACIONALES	Trabajo en equipo
5	FUNCIONALES	Capacidad Resolutiva

Ilustración 76 Mantenimiento Habilidades y Destrezas

#### 4.2.3.1. Ingreso

Para ingresar una nueva habilidad o destreza se debe navegar hacia el botón de ingresar (+) que se encuentra en la barra de navegación, posterior se abre una ventana con el formulario en el cual debe ingresar el tipo (Funcional/Organizacional) y la descripción de la habilidad o destreza y presionar en el botón “Aceptar”.

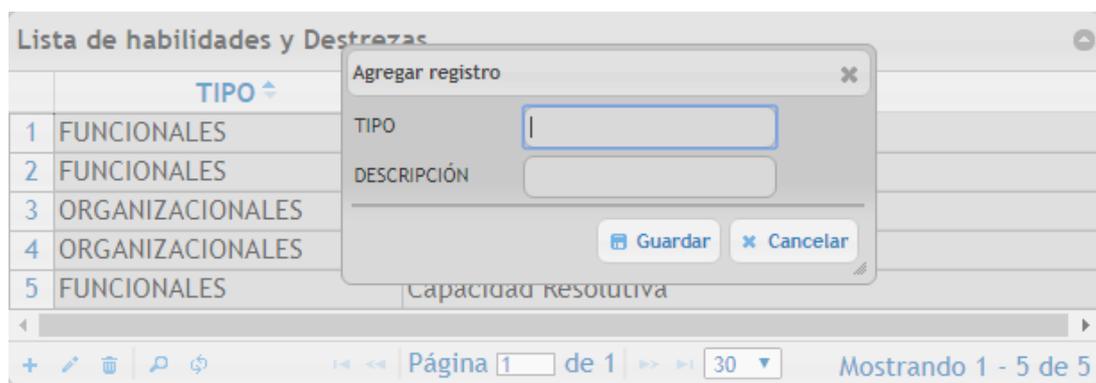


Ilustración 77 Mantenimiento Habilidades Destrezas

#### 4.2.3.2. Edición

Para modificar o eliminar un registro se debe seleccionar la fila o registro que va a actualizar y posterior presionar en el botón editar o eliminar según la necesidad. En el caso de modificación del registro el sistema muestra el formulario con los datos cargados para su posterior modificación (Ilustración 78). En el caso de eliminación de registros el sistema muestra una alerta para la confirmación de la acción a realizar y el usuario debe confirmar o cancelar la acción.

Posterior a la confirmación o cancelación de las acciones de edición el sistema cierra la ventana con el formulario y vuelve a cargar la información que contiene la grilla.



Ilustración 78 Edición Habilidades o Destrezas

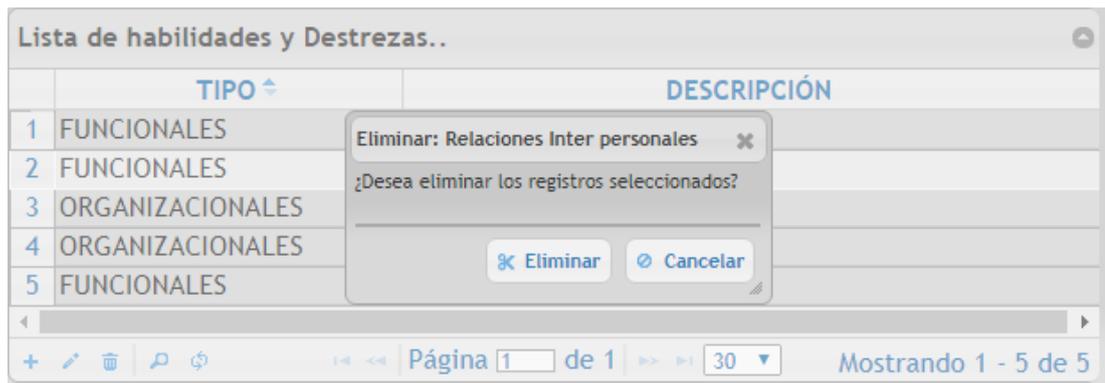


Ilustración 79 Eliminación Habilidades Destrezas

#### 4.2.4. Departamentos

Se puede ingresar los departamentos que componen la empresa, estos departamentos son los que están especificados en el organigrama de la empresa.

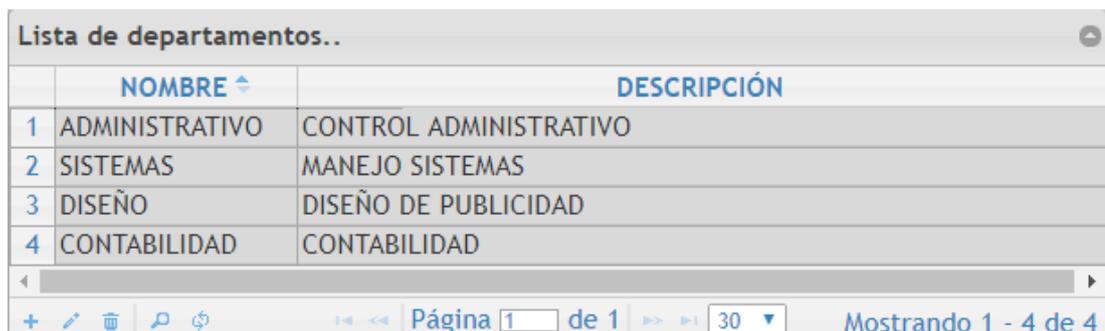


Ilustración 80 Mantenimiento Departamentos

##### 4.2.4.1. Ingreso

La opción de ingreso se realiza desde la barra de navegación de la grilla. Para ingresar un nuevo registro el sistema muestra una ventana con el formulario de departamentos que consta del nombre del departamento y una descripción del mismo. Posterior se presiona en el botón guardar y el sistema valida la información ingresada y refresca los registros ingresados.

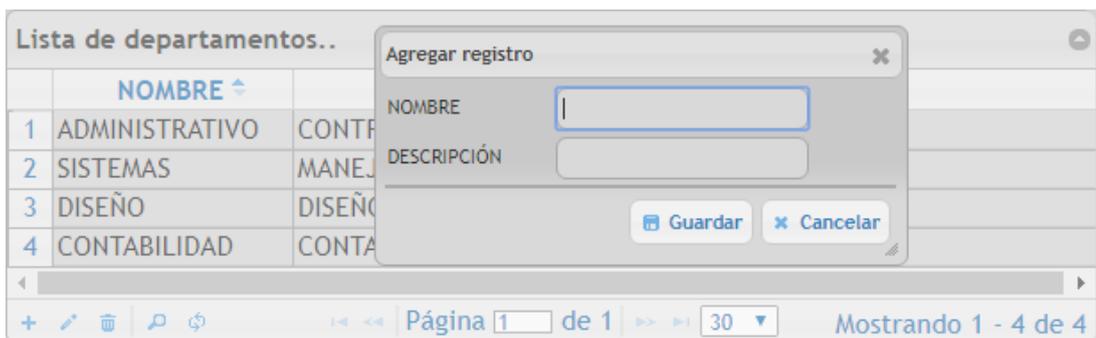


Ilustración 81 Ingreso Departamentos

#### 4.2.4.2. Edición

Para modificar o eliminar un registro se debe seleccionar la fila o registro que va a actualizar y posterior presionar en el botón editar o eliminar según la necesidad. En el caso de modificación del registro el sistema muestra el formulario con los datos cargados para su posterior modificación (Ilustración 82). En el caso de eliminación de registros el sistema muestra una alerta para la confirmación de la acción a realizar y el usuario debe confirmar o cancelar la acción.

Posterior a la confirmación o cancelación de las acciones de edición el sistema cierra la ventana con el formulario y vuelve a cargar la información que contiene la grilla.

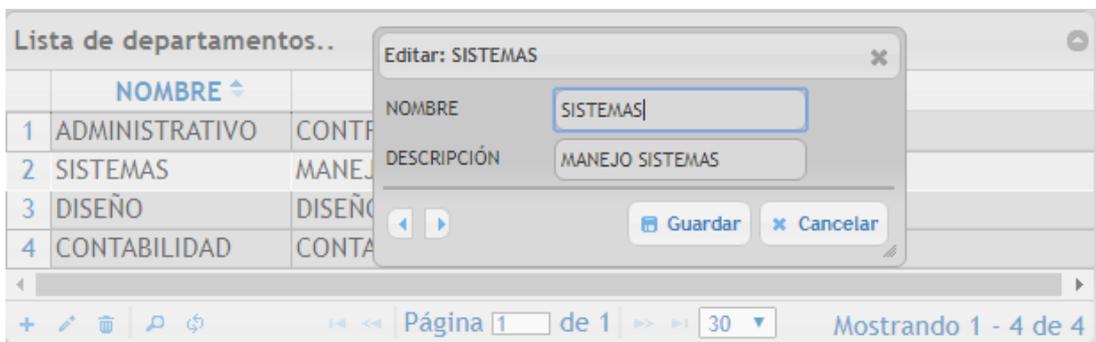


Ilustración 82 Edición Departamentos

#### 4.2.5. Configuración de Habilidades por formulario

Pueden configurar las habilidades o destrezas que van a ser evaluadas en un formulario específico. En este manual se considera únicamente el formulario “Confirmación de Referencias”.

En la siguiente ilustración se observa la pantalla que se muestra en la sección de configuración, misma que está compuesta por una lista de formularios y una grilla con opciones de ingreso y edición de registros.

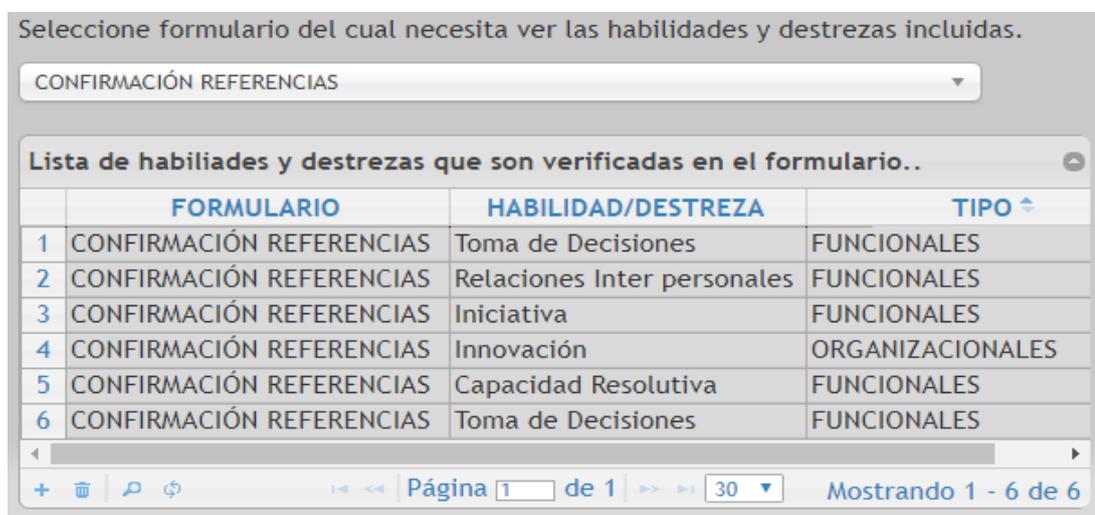


Ilustración 83 Configuración de Habilidades por formulario

#### 4.2.5.1. Ingreso

Para agregar una nueva habilidad o destreza a ser considerada en un formulario específico se debe seleccionar inicialmente el formulario al cual se le agregan las nuevas habilidades para posterior navegar hacia el botón de ingreso de la grilla. Luego de presionar en el botón de agregar el sistema muestra un formulario en el cual el usuario debe seleccionar la habilidad o destreza que desea ingresar.

Posterior al ingreso el sistema cierra la ventana de ingreso y vuelve a recargar la información.

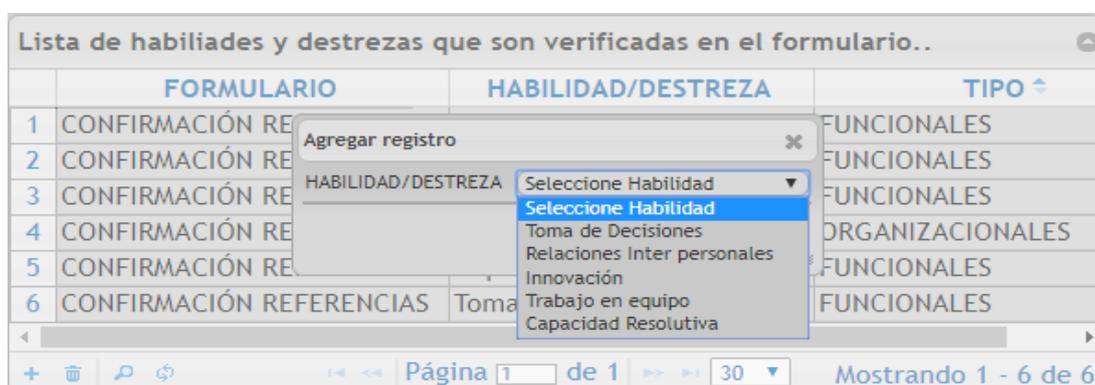


Ilustración 84 Ingreso Habilidades en Formulario

#### 4.2.5.2. Edición

Esta sección tiene solamente la opción de eliminación, para esto debe seleccionar el registro a eliminar y seleccionar el botón, posterior el sistema muestra un mensaje de confirmación de la acción a realizar.

#### 4.2.6. Requisitos por proceso

Se puede configurar los requisitos (documentos) necesarios para completar un proceso, que para este manual es "Admisión". En la siguiente ilustración se muestra la pantalla necesaria para realizar este proceso.



Ilustración 85 Requisitos por proceso

#### 4.2.6.1. Ingreso

Inicialmente se debe seleccionar un proceso de entre el listado que se encuentra disponible en la lista de procesos, posterior el sistema muestra los requisitos cargados en el proceso seleccionado.

Se debe navegar hacia el botón de ingreso (+) disponible en la barra de navegación y posterior se abre un formulario en el cual se debe ingresar la descripción del requisito y la cantidad solicitada (Ver ilustración 86).

Luego de ingresar los datos en el formulario y presionar en el botón guardar el sistema valida la información ingresada y de ser correcto cierra la ventana y vuelve a cargar la información en la grilla.

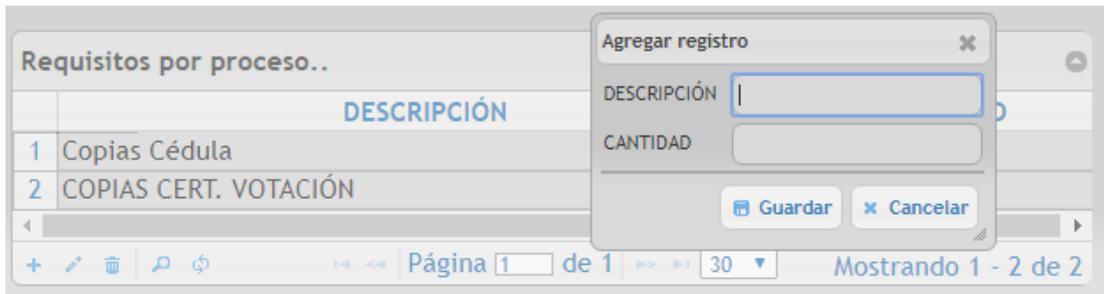


Ilustración 86 Ingreso requisitos por proceso

#### 4.2.6.2. Edición

Para modificar o eliminar un registro debe seleccionar la fila o registro que va a actualizar y posterior presionar en el botón editar o eliminar según la necesidad. En el caso de modificación del registro el sistema muestra el formulario con los datos cargados para su posterior modificación (Ilustración 16). En el caso de eliminación de registros el sistema muestra una alerta para la confirmación de la acción a realizar y el usuario debe confirmar o cancelar la acción.

Posterior a la confirmación o cancelación de las acciones de edición el sistema cierra la ventana con el formulario y vuelve a cargar la información que contiene la grilla.



Ilustración 87 Edición Requisitos por proceso

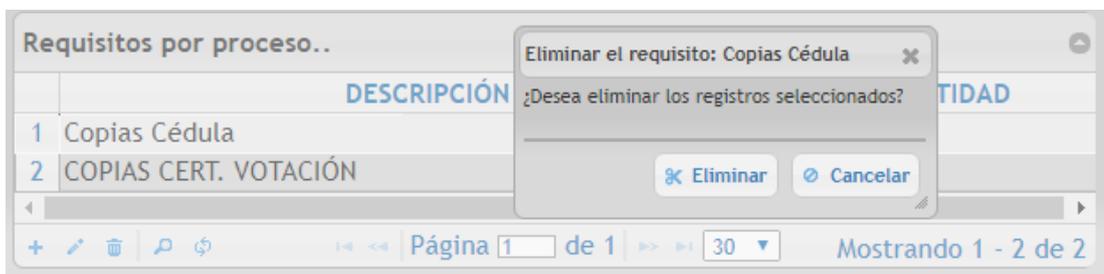


Ilustración 88 Eliminación requisitos por proceso

### 4.3. Cargos

Esta opción del sistema está disponible en el menú “Cargos” seleccionando la opción “Mantenimiento Cargos”. Dicha opción permite administrar los cargos que están disponibles en el sistema para ser usados al registrar postulantes y empleados.

Al ingresar a la opción de “Mantenimiento Cargos” se encuentra inicialmente una lista con los cargos ingresados y las opciones para ingresar nuevos y editar alguno de los ya ingresados.



	Nombre	Departamento	Jefe Inmediato	Mision
12	ASISTENTE DISEÑO	DISEÑO	Juan Perez	Lograr asentar las marcas de la empresa en el mercado nacional.
19	Gerente General	ADMINISTRATIVO	Juan Perez	Manejar la empresa con mira hacia la excelencia.

Ilustración 89 Pantalla Mantenimiento Cargos

#### 4.3.1. Nuevo Cargo

Se puede crear nuevos cargos navegando hacia el botón de ingresar (+) que se encuentra en la parte superior del listado de cargos. Luego de presionar el botón el sistema navega hacia una nueva página misma que contiene un formulario completo con información necesaria para el ingreso.

Ilustración 90 Ingreso Cargos

En el formulario de ingreso se debe principalmente ingresar los datos básicos del sistema como el nombre del cargo, seleccionar el departamento al cual pertenece el cargo, nombre del jefe inmediato del cargo.

Posterior se encuentra una serie de secciones con información complementaria requerida. En la primera sección se debe ingresar la misión y por lo menos una responsabilidad del cargo, posteriormente seleccionar el botón continuar.

Para crear y eliminar nuevas responsabilidades se tiene los botones de agregar (+) y eliminar (🗑️).

En las secciones formación académica, conocimientos específicos, capacitación adicional y nivel de experiencia se tiene que ingresar al menos una descripción para cada sección. Adicional cada sección tiene los botones de agregar y eliminar para crear nuevas descripciones (Ver ilustración 91).

Ilustración 91 Secciones Ingreso Cargos

Posterior en la sección 6 “Habilidades y Destrezas” se debe seleccionar las habilidades o destrezas requeridas para cumplir con el cargo ingresado. El listado de habilidades que se muestra es obtenido desde la página de configuración que previamente tuvo que ser ingresado.

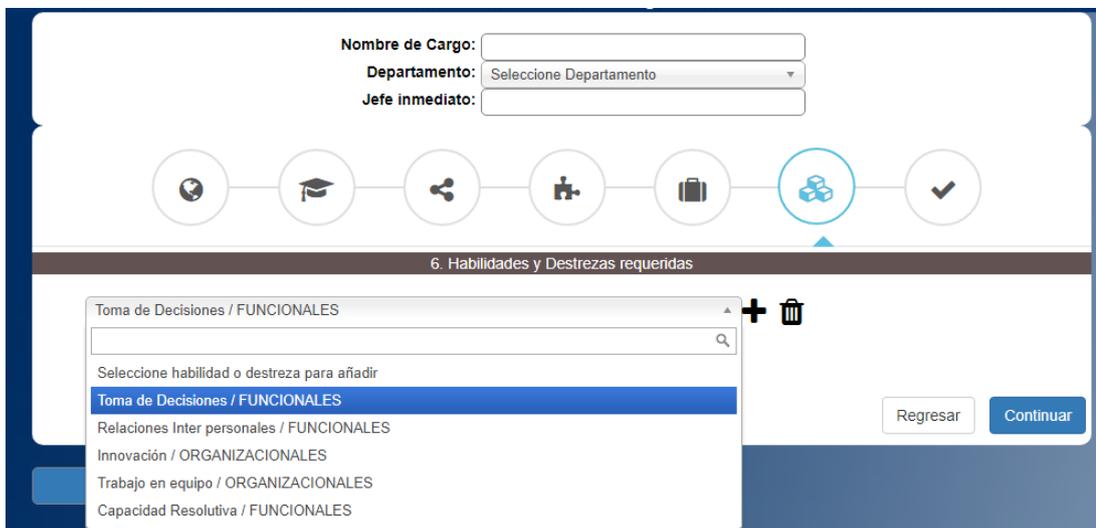


Ilustración 92 Selección Habilidades Cargo

Finalmente en la última sección se tiene la posibilidad de agregar un archivo que contiene la especificación del cargo en formato PDF. Para finalmente presionar en el botón grabar cargo.

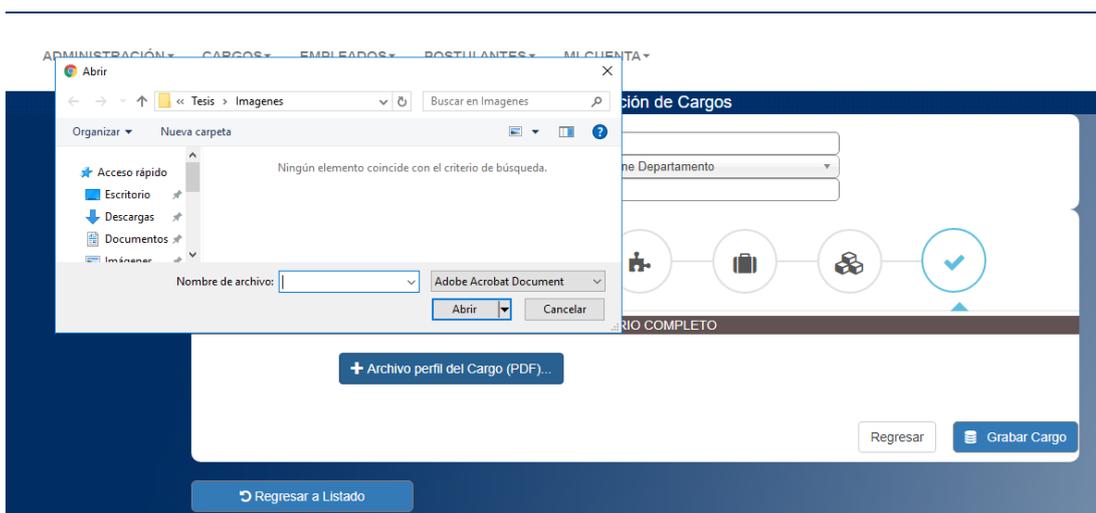


Ilustración 93 Carga Archivo Cargo

Luego de pulsar sobre el botón “Grabar Cargo” el sistema valida que todos los campos requeridos del formulario estén llenos caso contrario muestra una alerta y pinta de color rojo los campos faltantes (Ilustración 23).

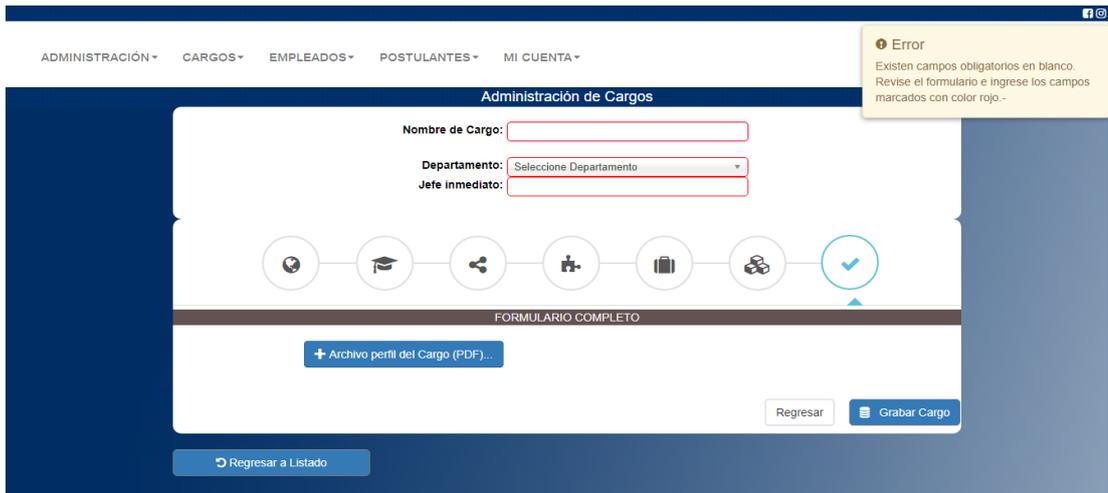


Ilustración 94 Errores Cargos

#### 4.3.2. Edición

Una vez dentro de la página de mantenimiento de cargos se tiene la opción de edición (✎) misma que le llevará hacia la página “Administración Cargos” y la cual se carga con la información del cargo que va a editar.

Luego de que se cargue la información se podrá editar la información que este en el formulario y posterior puede presionar en el botón “Guardar Cargo”. El sistema verifica la información ingresada y en el caso de que todo esté correcto regresa a la página principal del mantenimiento de cargos.



Ilustración 95 Edición de Cargos

## 4.4. Empleados

Esta opción del sistema está disponible en el menú “Empleado” seleccionando la opción “Mantenimiento”. Dicha opción permite administrar los empleados que están registrados en la empresa.

Al ingresar a la opción de “Mantenimiento” se encuentra inicialmente una lista con los empleados ingresados y las opciones para ingresar nuevos y editar alguno de los ya ingresados, agregar archivos al folder del empleado.

En el listado que se presenta inicialmente se observa una columna al final con la palabra “En proceso” que indica que aún no se han cargado todos los archivos de los requisitos para el proceso de contratación o admisión.



Empresa	Sucursal	Empleado	Cargo	
IMPORTADORA TOMBAMBA	ESPAÑA	Juan Carlos	ASISTENTE DISEÑO	En proceso
IMPORTADORA TOMBAMBA	ESPAÑA	Ana Carolina	Gerente General	En proceso

Ilustración 96 Listado Empleados

### 4.4.1. Ingreso Empleado

Se puede navegar a la opción de ingreso pulsando sobre el botón registrar (+) el cual lleva hacia la página de registro de empleados. La pantalla de registro presenta un formulario con la información necesaria para el ingresar un nuevo empleado.

Ilustración 97 Ingreso Empleado

En la ilustración anterior se observa una parte inicial en la que es necesario ingresar o seleccionar el postulante que va a ser ingresado como empleado. No es posible agregar un empleado sin antes haber creado como postulante.

Una vez seleccionado el postulante se carga la información básica en el formulario del empleado permitiendo modificar dicha información si es necesario (Ilustración 98).

	Nro. Documento	Nombres
25	0102085800	Maldonado Juan Carlos
26	0101440527	Piedra Calderon Elena Maria
1028	0102080650	Torres Montalvo Ana Carolina

Ilustración 98 Selección Postulantes

Finalmente para registrar un nuevo empleado es necesario ingresar el sueldo que va a percibir, la fecha de ingreso a la empresa, la sucursal en la que trabaja y el cargo

asignado. En otra sección del formulario se listan los requisitos (documentos) necesarios para completar el proceso, ahí se puede cargar los archivos digitales que correspondan a cada requisito (ilustración 99). No es obligatorio ingresar ese instante todos los archivos puesto que posterior puede agregar los requisitos.

Finalmente se debe presionar sobre el botón registrar empleado, mismo que verifica la información y envía una alerta en caso de error (ilustración 100) o guarda el empleado y regresa a la pantalla inicial del mantenimiento de empleados.

The screenshot shows the 'Registro de Empleados' interface. At the top, it says '1. DATOS PERSONALES'. Below this is a search bar for 'Nro. Documento:'. A modal window is open in the center with the title 'Cargar archivo para requisito seleccionado..' and a button '+ Seleccione Archivo...'. Below the modal, there is a section titled '3. Requisitos de Admision' with a list of requirements: 'HOJA VIDA ACTUALIZADA', 'COPIAS A COLOR LEGIBLES DE CÉDULA DE CIUDADANÍA', 'COPIAS CERTIFICADO DE VOTACIÓN (ACTUALIZADA)', and 'CERTIFICADO DE SALUD ACTUALIZADO (RESULTADOS DE EXAMENES DE SANGRE, ORINA Y HECES)'. At the bottom right, there are 'Regresar' and 'Continuar' buttons.

Ilustración 99 Ingresar Requisitos Empleados

The screenshot shows the 'Registro de Empleados' interface with an error message. The error message box at the top right says 'Error' and 'Existen campos obligatorios en blanco. Revise el formulario e ingrese los campos marcados con color rojo.'. Below the error message, the form shows '1. DATOS PERSONALES' with fields for 'Nro. Documento:', 'Apellidos:', 'Nombres:', 'Dirección:', and 'Teléfono:'. Below this is a progress indicator with three steps. The second section is '2. Información del Cargo' with fields for 'Sueldo:', 'Fecha Ingreso:', 'Sucursal:', and 'Cargo:'. The 'Sueldo:' and 'Fecha Ingreso:' fields are highlighted in red, indicating they are required and empty. A 'Continuar' button is at the bottom right.

Ilustración 100 Error Ingreso Empleado

#### 4.4.2. Edición Empleado

Una vez se encuentre sobre la página de mantenimiento de empleados tiene la opción de edición (  ) misma que le llevará hacia la página “Registro Empleados” y la cual se carga con la información del empleado que va a editar.

Una vez se cargue la información se puede editar los datos que este en el formulario y posterior puede presionar en el botón “Actualizar Empleado”. El sistema verifica la información ingresada y en el caso de que todo esté correcto regresa a la página principal del mantenimiento de empleados.



The screenshot displays a web form titled "Registro de Empleados" with two main sections: "1. DATOS PERSONALES" and "2. Información del Cargo".

**1. DATOS PERSONALES**

- Nro. Documento: 0102085800
- Apellidos: Maldonado
- Nombres: Juan Carlos
- Dirección: Panamericana Norte
- Teléfono: 2875882

**2. Información del Cargo**

- Sueldo: 1000 \$
- Fecha Ingreso: 09/10/2017
- Sucursal: ESPAÑA
- Cargo: ASISTENTE DISEÑO

Buttons: "Continuar" (bottom right), "Regresar a Listado" (bottom left).

Ilustración 101 Edición de Empleados

#### 4.4.3. Folder Empleado

Se puede listar los archivos cargados en el folder del empleado navegando hacia el mantenimiento de empleados y luego de seleccionar uno de los empleados ingresados, presionar el botón folder empleado (  ). Posterior se abre una ventana emergente con el listado de archivos digitales cargados al folder del sistema.

Se puede abrir cualquier de los archivos ingresados presionando sobre el nombre del archivo.



Ilustración 102 Folder Empleado

Para agregar un nuevo archivo al folder del empleado se debe seleccionar el empleado al cual pertenece el archivo a ingresar y posterior seleccionar el botón agregar archivo que se encuentra en la página mantenimiento de empleados. Luego de presionar sobre el botón el sistema muestra una ventana en la cual se puede o no seleccionar un requerimiento de entre la lista disponible, es decir que el archivo a cargar cubre un requerimiento necesario para la admisión o simplemente es un archivo adicional que desea cargar al folder del empleado.



Ilustración 103 Agregar archivo Folder

#### 4.5. Postulantes

Esta opción del sistema está disponible en el menú “Postulantes” seleccionando la opción “Mantenimiento”. Dicha opción permite administrar los postulantes que están registrados en la empresa.

Al ingresar a la opción de “Mantenimiento” se encuentra inicialmente una lista con los postulantes ingresados y las opciones para ingresar nuevos y consultar alguno de los ya ingresados. Adicionalmente se tiene a disposición el botón para acceder a la página “Confirmación de Referencias”.

Listado de Postulantes					
Doc. Identidad	Nombre	Apellido	Dirección	Teléfono	Cargo Aplica
0102085800	Juan Carlos	Maldonado	Panamericana Norte	2875882	ASISTENTE DISEÑO
0101440527	Elena Maria	Piedra Calderon	Calle de los Principes	2804669	Gerente General
0102080850	Ana Carolina	Torres Montalvo	Solano y Remigio Romero	2815474	ASISTENTE DISEÑO

Ilustración 104 Mantenimiento de Postulantes

#### 4.5.1. Ingreso

Se tienen dos opciones para el registro o ingreso de postulantes. Para los usuarios tipo visitantes (no están registrados) tienen habilitado la página “Registro de Postulantes” desde la pantalla inicial del sistema presionando sobre el link “Trabaje con Nosotros” (Ilustración 105).

The screenshot shows a web page titled "Trabaje con Nosotros" with a dark blue header. The main content area features a dark gray login box with the following elements:

- Iniciar sesión** (Login)
- Text: "Escriba su nombre de usuario y contraseña. [Regístrese](#) si no tiene una cuenta."
- Input field for "Usuario" (Username)
- Input field for "Contraseña" (Password)
- Checkbox labeled "Recordar mi cuenta" (Remember my account)
- Blue button labeled "Iniciar sesión" (Login)
- Text: "¿Usted ha olvidado su contraseña? [Restablecer contraseña](#)." (Forgot your password? Reset password.)

Ilustración 105 Pantalla Inicial

La otra forma que se tiene para ingresar postulantes es desde el menú “Postulantes” en la opción “Mantenimiento” y presionando sobre el botón registrar postulante (+).

Esta opción está habilitada solo para los usuarios registrados en el sistema (Ilustración 106).

Una vez ingresado en la página de registro de postulantes el sistema muestra un formulario para el ingreso de información básica del postulante así como información del cónyuge, educación e historial laboral.

The screenshot displays a web form titled "Solicitud de Empleo" with two main sections: "1. DATOS PERSONALES" and "2. DATOS FAMILIARES".

**1. DATOS PERSONALES**

- Aplica: Seleccione Cargo (dropdown menu)
- Tipo Documento: Seleccione Tipo\_Documento (dropdown menu)
- Nro. Documento: [input field]
- Apellidos: [input field]
- Nombres: [input field]
- Dirección: [input field]
- Teléfono: [input field]

A progress bar below the first section contains seven icons: a group of people (selected), a graduation cap, a briefcase, a document, a person, a dollar sign, and a checkmark.

**2. DATOS FAMILIARES**

- Nombres (Cónyuge): [input field]
- Cédula (Cónyuge): [input field]
- Fecha Nac (Cónyuge): [calendar icon]
- Nro. Hijos: [input field]

Below the family data section is a table titled "Lista de cargas familiares..".

	NOMBRES	DOCUMENTO IDENTIDAD	PAÍS NACIMIENTO	FECHA NACIMIENTO	DISCAPACIDAD (SI/NO)	
No hay resultados						

At the bottom right of the form is a blue "Continuar" button.

Ilustración 106 Registro Postulantes

En la sección principal del formulario se debe seleccionar el cargo al cual está aplicando posterior seleccionar el tipo de documento de identidad e ingresar el número de documento. El sistema valida el número de documento en caso de ser incorrecto envía una alerta y borra el número de documento ingresado. Posterior al ingreso del número de identificación se debe ingresar los datos básicos como sus apellidos, nombres, dirección y teléfono.

En la segunda sección del formulario (Datos Familiares) se debe ingresar la información de su cónyuge y de sus hijos en caso tener. Antes de agregar la información de los hijos se debe ingresar el número de hijos que tiene para luego completar la información en la grilla (Ilustración 107).

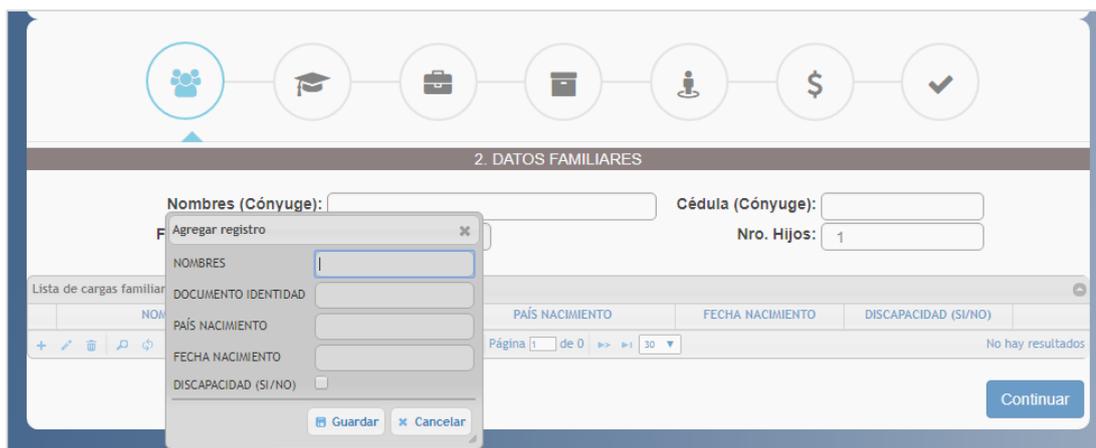


Ilustración 107 Postulantes Datos Familiares

En la tercera sección (Instrucción) se debe ingresar la información sobre la educación obtenida, por ejemplo, nivel escolar, bachillerato, superior, etc. Todos los campos de este formulario son obligatorios por lo que no podrá completar el ingreso si no ingresa toda la información requerida.



Ilustración 108 Postulantes Instrucción

En la cuarta sección (Ocupación Actual) se puede ingresar, sin ser obligatorio, la información sobre su trabajo actual. Información como la empresa, cargo, salario y una breve descripción de las tareas.

Se tiene un botón que le permite saltar esta sección puesto que no so campos obligatorios.

**4. OCUPACION ACTUAL**

Establecimiento:  Cargo desempeña:

Jefe Inmediato:  Salario:

Fecha ingreso:

Descripción de tareas que desempeña..

Regresar Saltar Continuar

Ilustración 109 Postulantes Ocupación Actual

En la quinta sección (Ocupaciones Anteriores) se tiene la posibilidad de ingresar información sobre su historial laboral, empresas en donde ha trabajado y los cargos que ha desempeñado. Esta sección al igual que la anterior puede ser omitida ya que la información requerida no es obligatoria.

**5. OCUPACIONES ANTERIORES**

Lista de ocupaciones anteriores..

ESTABLECIMIENTO	FECHA INGRESO (y)	FECHA SALIDA	CARGO	SUELDO	JEFE INMEDIATO	MOTIVO SALIDA
No hay resultados						

Regresar Saltar Continuar

Ilustración 110 Postulantes Ocupaciones Anteriores

En la sexta sección (Referencias Personal) se debe ingresar al menos un registro en la grilla. Para el ingreso de información puede utilizar los botones disponibles en la barra de navegación de la grilla. Todos los campos: nombres, teléfono y actividades son campos obligatorios al momento de crear un nuevo registro.



Ilustración 111 Postulantes Referencias Personales

La séptima sección (Aspiración Salarial) es información relevante para la postulación, ya que es preciso conocer la aspiración salarial, la disponibilidad y si está dispuesto a trabajar en horarios rotativos. Esta información es obligatoria y no puede completar el registro si no está ingresada. Para cambiar de opción de si a no o viceversa en los campos disponibilidad y turnos rotativos se tiene que presionar sobre el botón y con eso se cambiará de opción.



Ilustración 112 Postulantes Aspiración Salarial

En la última sección se puede cargar de manera no obligatoria el archivo digital con su hoja de vida y posterior presionar el botón guardar formulario para completar con el registro. En caso de que la información ingresada esté completa y correcta el sistema guarda el postulante y regresa a la página inicial, caso contrario el sistema muestra una alerta con los errores y marcados con rojo los campos que son obligatorios (Ilustración 114).



Ilustración 113 Postulante Registro Completo



Ilustración 114 Postulante Errores

#### 4.5.2. Visualización

La opción de visualización de información de postulantes está disponible sólo para usuarios registrados del sistema. Se puede acceder navegando hasta la pantalla inicial del mantenimiento y presionando sobre el botón consultar (🔍). Posterior el sistema abre la página registro de postulantes con la información cargada en el sistema.

**Solicitud de Empleo**  
1. DATOS PERSONALES

**Aplica:**   
**Tipo Documento:**   
**Nro. Documento:**   
**Apellidos:**   
**Nombres:**   
**Dirección:**   
**Teléfono:**

2. DATOS FAMILIARES

**Nombres (Cónyuge):**   
**Fecha Nac (Cónyuge):**

**Cédula (Cónyuge):**   
**Nro. Hijos:**

Lista de cargos familiares...

NOMBRES	DOCUMENTO IDENTIDAD	PAÍS NACIMIENTO	FECHA NACIMIENTO	DISCAPACIDAD (SI/NO)
No hay resultados.				

[Continuar](#)

[Regresar a Listado](#)

Ilustración 115 Consulta Postulantes

#### 4.5.3. Confirmación de Referencias

El formulario confirmación de referencias permite validar la información sobre el historial laboral ingresado por cada postulante. Este formulario contiene un listado de habilidades o destrezas que son evaluadas por cada responsable de la empresa ingresada en el historial laboral.

La opción confirmación de referencias está disponible desde la pantalla inicial del mantenimiento de postulantes presionando sobre el botón confirmación de referencias ( ). Se debe seleccionar un postulante de entre la lista mostrada antes de presionar en el botón confirmación de referencias.

Luego de presionar el botón el sistema navega hacia la página confirmación de referencias y muestra el historial laboral ingresado por el postulante seleccionado.

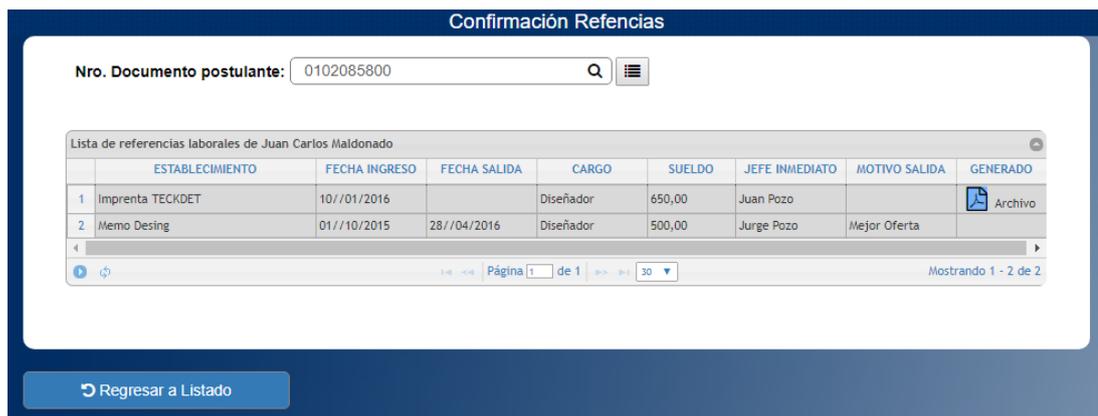


Ilustración 116 Pantalla Confirmación Referencias

En la ilustración anterior se puede apreciar que el postulante ha ingresado dos trabajos que ha desempeñado. Una vez realice la confirmación de la referencia, en la última columna de la grilla se observa un ícono de un archivo que aclara que ya ha sido generado el formulario de confirmación de referencias.

En las filas que tengan vacía la columna “Generado” quiere decir que todavía no ha sido generado el formulario de confirmación de referencias. Para generar el formulario se debe seleccionar una referencia laboral y luego presionar en el botón “Validar Referencia” ( ) que se encuentra en la barra de navegación de la grilla.

Luego de presionar en el botón el sistema muestra el formulario de confirmación de referencias cargado con información sobre el postulante, la referencia laboral que se está validando y el listado de habilidades o destrezas que se van a evaluar (ilustración 117).

Finalmente debe calificar las respuestas de la persona entrevistada sobre las habilidades del postulante teniendo tres opciones 1,2 y 3 siendo 3 la calificación más alta del desempeño. Luego de calificar las habilidades se debe presionar en el botón “Guardar Formulario” el sistema valida la información ingresada y en el caso de que todo esté correcto muestra una alerta de confirmación y cierra el formulario.

## Formulario Confirmación Referencias

**Nombre:** Juan Carlos Maldonado  
**Empresa:** Imprenta TECKDET  
**Contacto:** Juan Pozo  
**Cargo Desempeñado:** Diseñador  
**Tiempo Laborado:** 24 días 3 meses 1 años

Marque con un check considerando:

- 1 Malo
- 2 Bueno
- 3 Excelente

	1	2	3
Toma de Decisiones	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relaciones Inter personales	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Iniciativa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Innovación	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Capacidad Resolutiva	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Toma de Decisiones	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

 Guardar Formulario

Cerrar

Ilustración 117 Formulario Confirmación Referencias

## CONCLUSIONES Y RECOMENDACIONES

Después de realizar el estudio sobre el framework adaptable Bootstrap y el uso de las tecnologías HTML5, Javascript, CSS3 y AJAX se ha llegado a las siguientes conclusiones y recomendaciones.

### Conclusiones:

- La utilización del framework Bootstrap en el desarrollo de aplicaciones web reduce considerablemente el tiempo asignado para el desarrollo de la aplicación. Esto gracias a que los desarrolladores trabajan en un solo código fuente de la aplicación y eliminan la creación de las diferentes versiones para ciertos dispositivos.
- Las aplicaciones web desarrolladas con el framework Bootstrap se adaptan satisfactoriamente en cualquier dispositivo móvil o de escritorio en el que sea ejecutado. Adaptabilidad que permite a los usuarios trabajar en la aplicación sin que esta pierda sus funcionalidades.
- El uso del sistema de gestión de selección de personal mejora la productividad y el tiempo de respuesta de los usuarios al completar el proceso de contratación en la empresa. Esto gracias a que el sistema registra de manera adecuada los datos de los postulantes y los cargos a los que aplica cada uno permitiendo analizar la información de los postulantes. Además de mejorar de manera sustancial la manipulación de los documentos de cada empleado y teniendo un control eficaz sobre estos.

### Recomendaciones:

- Se recomienda la utilización del framework Bootstrap en el desarrollo de software web ya que es una herramienta libre, sin limitaciones, con una comunidad que constantemente publica ejemplos y mejoras y que actualmente está siendo utilizado por las principales compañías desarrolladores de software a nivel mundial como Twitter.

- Se recomienda complementar el desarrollo web con el uso de los lenguajes HTML5, framework jQuery y el lenguaje CSS3 ya que sus nuevos elementos, características y funcionalidades permiten manipular los elementos del DOM y sus atributos consiguiendo un software completamente adaptable.

## BIBLIOGRAFÍA

- Chiavenato, I. (2002). *Gestión del Talento Humano*. Mc. Graw Hill.
- Cuevas, J. C. (2011). *Objetivos y actividades de la gestión del talento humano*. Obtenido de <http://psicologiayempresa.com/objetivos-y-actividades-de-la-gestion-del-talento-humano>.
- Drake, J. (s.f.). <https://www.ctr.unican.es>. Recuperado el 17 de 10 de 2017
- Garrett, J. J. (18 de 02 de 2005). *AdaptivePath*. Obtenido de <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona, España: Marcombo.
- Geoffrey Sparks, S. S. (s.f.). <http://www.sparxsystems.com.ar>. Recuperado el 15 de 10 de 2017
- Harold, E. R. (2007). *New elements in HTML 5*. IBM Corporation.
- Haverbeke, M. (2014). *Eloquent JavaScript A Modern Introduction to Programming*.
- Holzner, S. (2011). *HTML5 in 10 Minutes*. Pearson Education.
- Humano, G. d. (2013). *Gerencie.com*. Obtenido de <http://www.gerencie.com/gestion-del-talento-humano.html>
- IBM. (s.f.). *ibm*. Recuperado el 17 de 10 de 2017
- James, J. (2011). *The Elements of user experience*. Berkeley: Pearson Education.
- Jericó, P. (2018). *Gestión del Talento Humano*.
- jQuery. (s.f.). *API JQuery*. Obtenido de <http://api.jquery.com/category/selectors/>
- Matthew, M. (2011). *HTML 5 The missing manual*. O" Reilly Media.
- Microsoft. (s.f.). *Información general sobre ASP.NET*.
- Montes, M. J. (2006). *Selección de Personal*. Ideaspropias.
- Mora, S. L. (s.f.). *Ajax Technology in Web*. Recuperado el 17 de 10 de 2017
- Pérez, J. E. (2008). *Introducción a AJAX*.
- Point, T. (2014). *Learn ASP.NET web application framework*.
- Point, T. (s.f.). *Tutorials Point*. Obtenido de [www.tutorialspoint.com](http://www.tutorialspoint.com)
- Puig, J. C. (2015). *CSS3 y Javascript Avanzado*. Catalunya: Universidad Oberta de Catalunya.
- Sae-Chin, J. (2008). *Interactive Multimedia Design & Production*. Austin: University of Texas.
- Sánchez, A. J. (2015). *Bootstrap 3 Introducción al diseño responsive*.
- Silberschatz, A., F. Korth, H., & Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS*. Madrid: McGraw-Hill Inc.

Storti, G. (2007). *www.belgrano.esc.edu.ar*. Recuperado el 17 de 10 de 2017

Varga, S., Cherry, D., & Joseph, D. (2016). *Introducing Microsoft SQL Server 2016*. Washington.

## ANEXOS

### Anexo 1: HTML 5, Nuevos Elementos

En el libro (Matthew, 2011), clasifica los nuevos elementos de HTML 5, según su funcionalidad, a continuación, se detalla cada uno de estos y como incorporarlos en páginas web.

Nuevos tipos de elementos en formularios

La siguiente lista fue tomada de (Holzner, 2011) capítulo 4.

TIPO	DESCRIPCION
button	Campo tipo botón.
checkbox	Campo tipo check.
color	Campo tipo color, que muestra el color seleccionado.
date	Campo de fecha (año, mes, día).
datetime	Campo de fecha y hora
datetime-local	Campo de fecha y hora
email	Campo de correo electrónico
file	Campo para agregar archivos
hidden	Campo oculto
image	Campo de tipo imagen
month	Campo tipo fecha, permite manipular el mes.
number	Campo de tipo numérico
password	Campo de tipo contraseña
radio	Campo tipo radio
range	Campo tipo rango, selecciona un valor entre un rango.
reset	Campo tipo reinicio. Regresan valores originales.
search	Campo tipo búsqueda.
submit	Campo tipo confirmación.
tel	Campo tipo texto.
text	Campo tipo texto.
url	Campo tipo dirección.
week	Campo tipo fecha, considerada por semanas del año.

## Elementos estructurales.

Estos elementos reemplazan la mala estructura llevada hasta ahora al momento de crear sitios web, en los cuales las diferentes secciones son manejadas solamente con el elemento <div> complicando con esto la comprensión de la estructura del documento. En la imagen “Estructura HTML5” se puede observar la diferencia entre la estructura manejada con HTML y la nueva estructura proporcionada por HTML5.

- <article></article>.- Este elemento representa un componente de una página que consiste en una composición autónoma en un documento web, con la intención de que pueda ser reutilizado y repetido. Podría utilizarse en los artículos de los foros, una revista o el artículo de periódico, una entrada de un blog, un comentario escrito por un usuario, un widget interactivo o gadget, o cualquier otro artículo independiente de contenido.

Cuando los elementos de <article> son anidados, los elementos de <article> interiores representan los artículos que en principio son relacionados con el contenido del artículo externo. Por ejemplo, un artículo de un blog que permite comentarios de usuario, dichos comentarios se podrían representar con <article>.

- <aside></aside>.- Representa una sección de la página que abarca un contenido tangencialmente relacionado con el contenido que lo rodea, por lo que se le puede considerar un contenido independiente. Este elemento se utiliza para efectos tipográficos, barras laterales, elementos publicitarios, para grupos de elementos de la navegación, u otro contenido que se considere separado del contenido principal de la página.
- <figure></figure>.- Este elemento es una alternativa a la hora de incorporar imágenes dentro de documentos web. Este elemento va incorporado con el elemento <figcaption> que establece el título de la imagen agregada.
- <footer></footer>.- El elemento representa el pie de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página, como el autor, el copyright o el año.

- `<header></header>`.- Este elemento representa un grupo de artículos introductorios o de navegación.
- `<hgroup></hgroup>`.- En la versión anterior de HTML, tenemos una gran variedad de elementos (h1, h2, h3, etc) al momento de crear títulos en páginas web. El número que acompaña a la letra “h” establece el tamaño de la letra del título de la página. Ahora en la nueva versión se encuentra el elemento `<hgroup>` que especifica que el contenido que está dentro del elemento será considerado como un subtítulo de la página.
- `<nav></nav>`.- Este elemento representa una sección de una página que es un link a otras páginas o a partes dentro de la página: una sección con links de navegación.

No todos los grupos de enlaces en una página tienen que estar en un elemento `<nav>`, sólo las secciones que consisten en bloques principales de la navegación son apropiadas para ser utilizadas con el elemento `<nav>`. Puede utilizarse particularmente en el pie de página para tener un menú con un listado de enlaces a varias páginas de un sitio, como el Copyright; home page, política de uso y privacidad. No obstante, el elemento `<footer>` es plenamente suficiente sin necesidad de tener un elemento `<nav>`.

- `<section></section>`.- Elemento que permite crear secciones dentro de un documento, por ejemplo, si una página refleja las noticias del mundo una sección podría ser “Sección de deportes”. Este elemento puede contener otros elementos del mismo tipo, creando así subsecciones, si se mantiene el ejemplo anterior una subsección sería “Campeonato Ecuatoriano”.

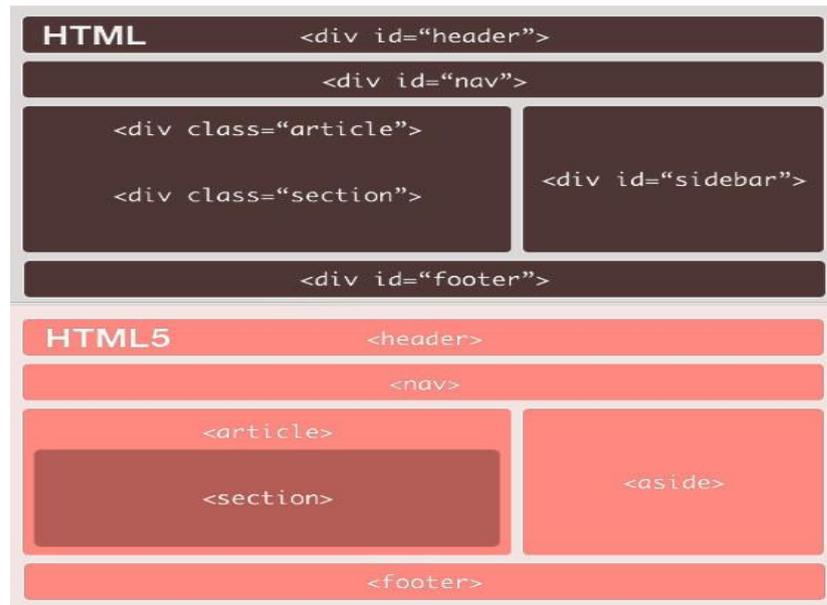


Ilustración 118 Estructura HTML5

### Elementos semánticos en línea

- `<mark></mark>`.- Este elemento resalta un texto dentro de cualquier parte de un documento.

Las `<mark>`Islas Galapagos`</mark>` son un gran tesoro protegido del mundo.

- `<meter></meter>`.- Este elemento representa un valor numérico en un rango específico. Dicho elemento muestra de forma gráfica en el documento web el valor establecido.

#### Atributos:

- value: Valor que plasmará el elemento.
- min: Valor mínimo que se podrá incluir en el elemento.
- low: Valor más bajo dentro de una muestra.
- high: Valor más alto dentro de una muestra.
- max: Valor máximo que puede tener el elemento.
- optimum: Valor óptimo.

Todos los atributos detallados reciben un valor numérico que indican un rango relevante.

`<p>Tu puntaje es <meter value="75,5" min="0" max="100" low="65" high="96" optimum="100">B+</meter>.</p>`

- `<progress></progress>`.- Elemento que es bastante usado en el desarrollo de formularios, ahora traído al desarrollo de documentos web o aplicaciones web. Este elemento tiene 2 atributos primordiales: `value` que guarda el valor actual del elemento y `max` que guarda el valor final que puede alcanzar el elemento dentro de un proceso.

```
<progress value="1534602" max="4603807">30%</progress>
```

Elementos de interacción.

Mejoran la interacción de los usuarios con la página o aplicación web. Proporciona elementos como menús, grillas entre otros que pueden modificar su contenido de acuerdo a la interacción del usuario.

- `<details></details>`.- Es un elemento que funciona por medio de un evento clic, mostrando mayor información sobre algún tema que no tuvo la información completa a primera vista. Por ejemplo se puede utilizar para mostrar información personal de un cliente, que a primera vista muestra solo su nombre.

Atributos:

- ✓ `accesskey`
- ✓ `class`
- ✓ `contenteditable`
- ✓ `contextmenu`
- ✓ `dir`
- ✓ `draggable`
- ✓ `hidden`
- ✓ `id`
- ✓ `lang`
- ✓ `open`
- ✓ `spellcheck`
- ✓ `style`
- ✓ `tabindex`
- ✓ `title`

Ejemplo de su uso.

```

1 <html lang="es">
2 <head>
3 <h2>Reproducir Audio</h2>
4 </head>
5 <body>
6 <section>
7 <details>
8 <summary>Equipo Rojo</summary>
9 <dl>
10 <dt>Nombre:</dt> <dd>Dep. Cuenca</dd>
11 <dt>Colores:</dt> <dd>Rojo / Negro</dd>
12 <dt>Estadio:</dt> <dd>Alejandro Serrano</dd>
13 </dl>
14 </details>
15 </section>
16 </body>
17 </html>

```

Hyper Text Markup Language file length: 346 lines: 17 Ln: 3 Col: 28 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS

Ilustración 119 Elemento <details>

- <datagrid></datagrid>.- Este elemento es parecido a las grillas en aplicaciones de escritorio. Su contenido puede estar cargado por diferentes elementos como tablas o listas, que son actualizados de acuerdo a la interacción del usuario o scripts que se ejecuten.

Ejemplo de su uso.

```

1 <html lang="es">
2 <head>
3 <h2>Ejemplo Grilla</h2>
4 </head>
5 <body>
6 <datagrid>
7 <table>
8 <tr>
9 <td>Jones</td>
10 <td>Allison</td>
11 <td>A</td>
12 <td>B+</td>
13 <td>A</td>
14 </tr>
15 <tr>
16 <td>Smith</td>
17 <td>Johnny</td>
18 <td>A</td>
19 <td>C+</td>
20 <td>A</td>
21 </tr>
22 <tr>
23 <td></td>
24 <td></td>
25 <td></td>
26 <td></td>
27 <td></td>
28 </tr>
29 </table>
30 </datagrid>
31 </body>
32 </html>

```

Hy length: 618 lines: 39 Ln: 2 Col: 11 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS

Ilustración 120 Elemento <datagrid>

- <menu></menu> y <command></command>.- El elemento <menu>, como su nombre lo indica crea elementos desplegables tipo menú. Este elemento puede

contener elementos <command> que pueden o no estar dentro del elemento <menu>.

El elemento <command> puede convertirse en diferentes tipos de elementos de acuerdo a uno de sus atributos “type” que indica la manera de presentarse en el documento. Las diferentes presentaciones son: button, check box o radio button (Holzner, 2011).

Atributos del elemento <command>:

✓	accesskey	✓	label
✓	checked	✓	radiogroup
✓	dir	✓	spellcheck
✓	disabled	✓	title
✓	draggable	✓	type
✓	icon		

#### Elementos embebidos

Estos elementos son los que probablemente más expectativas levantaron, ya que en la versión anterior de HTML no se podía incluir videos dentro de las páginas web o aplicaciones. Si se necesita incluir videos necesariamente se lo hacía mediante gifs, flash o enlaces a las aplicaciones contenedoras del video.

- <video></video>.- Este elemento reproduce videos dentro de páginas o aplicaciones web. Contiene una serie de atributos que incluye controles dentro del video a reproducir.

Atributos:

- ✓ autoplay.- Atributo de tipo booleano (true/false) que indica si el video incluido empezara a reproducirse automáticamente.
- ✓ controls.- Atributo de tipo booleano (true/false) que establece si en el video se mostraran o no los controles de pausa, volumen, etc.
- ✓ height.- Atributo de tipo numérico que especifica el alto del elemento video.
- ✓ loop.- Atributo de tipo booleano (true/false) que especifica si el video se mantendrá ejecutándose continuamente, es decir una y otra vez.

- ✓ poster.- Atributo que establece una dirección (URL) de una imagen que se puede mostrar si es que el navegador que se encuentra cargando la página no soporta la carga del video.
  - ✓ preload.- Atributo que especifica si el video estará cargado completamente aun que el usuario no vaya a usarlo.
  - ✓ src.- Establece la dirección (URL) donde se encuentra almacenado el video a reproducir.
  - ✓ width.- Atributo de tipo numérico que especifica el ancho que tendrá el elemento video dentro de la página.
  - ✓ onerror.- Atributo que especifica un evento que se ejecutará cuando la carga del video produjo un error.
- <audio></audio>.- Este elemento reproduce audio en documentos o aplicaciones web. Lo importante es recalcar que al usar este elemento no es necesario incluir extensiones adicionales o plugins en los navegadores.

Atributos:

- ✓ autoplay. Atributo de tipo booleano (true/false) que indica si el audio incluido empezara a reproducirse automáticamente.
  - ✓ controls. Atributo de tipo booleano (true/false) que establece si en el audio se mostraran o no los controles de pausa, reproducir y volumen.
  - ✓ loop.- Atributo de tipo booleano (true/false) que especifica si el audio se mantendrá ejecutándose continuamente, es decir una y otra vez.
  - ✓ preload.- Atributo que especifica si el audio estará cargado completamente aun que el usuario no vaya a usarlo.
  - ✓ src.- Establece la dirección (URL) donde se encuentra almacenado el audio a reproducir.
  - ✓ onerror.- Atributo que especifica un evento que se ejecutará cuando la carga del audio produjo un error.
- <canvas></canvas>.- El elemento <canvas> es conocido como un lienzo de dibujo, el cual da la posibilidad al usuario de incluir imágenes, texto, poder dibujar líneas, cuadrados, rectángulos, inclusive llegar a hacer un videojuego dentro de esta herramienta.

## Anexo 2: JavaScript, métodos para acceder al DOM

A continuación se detalla todos los selectores que pueden ser utilizados desde JavaScript. Ahora se puede seleccionar elementos HTML aplicando toda clase de selectores CSS por medio de los nuevos métodos `querySelector()` y `querySelectorAll()`. El siguiente listado es obtenido de (Gauchat, 2012).

- `getElementsByTagName`.- Este método busca los elementos HTML de acuerdo al nombre de la etiqueta. A este método se puede especificar el elemento exacto que se quiere manipular, es decir JavaScript puede encontrar cinco elementos que coinciden con la búsqueda, pero se puede establecer que la manipulación afecte solamente al primer elemento encontrado, para esto se usa los corchetes y un número adentro. El número indicara la posición del elemento encontrado (Gauchat, 2012).

```
document.getElementsByTagName('button')[0].onclick = mostrarMensaje();
```

Ilustración 121 Método "getElementsByTagName"

- `getElementById`.- Este método busca un elemento HTML que coincida su atributo ID con el valor buscado.

```
document.getElementById('btnGuardar').onclick = mostrarMensaje();
```

Ilustración 122 Método "getElementById"

- `getElementsByClassName`.- Este método busca los elementos HTML de acuerdo al atributo "class" que tengan ingresado.

```
document.getElementsByClassName('botones').onclick = mostrarMensaje();
```

Ilustración 123 Método "getElementsByClassName"

- `querySelector`.- Este método busca el primer elemento que concuerda con el filtro de búsqueda ingresado y es ese el que será manipulado.

```
document.querySelector("#principal button").onclick = mostrar_mensaje();
```

Ilustración 124 Método "querySelector"

En el ejemplo anterior “principal” es el ID del elemento en el cual se va a buscar, mientras que “button” especifica el tipo de elemento que va a ser manipulado. Es decir JavaScript busca el primer elemento tipo botón que se encuentre dentro del elemento con ID igual a “principal”.

- querySelectorAll.- Este método busca todos los elementos que concuerdan con el filtro de búsqueda ingresado. Y todos esos serán manipulados ya sea para cambios de atributos o para agregar eventos.

```
function ejemplo(){  
    var lista = document.querySelectorAll("#principal button");  
    lista[0].onclick = mostrar_mensaje();  
}
```

Ilustración 125 Método "querySelectorAll"

En el ejemplo anterior se busca todos los elementos tipo botón que estén dentro del elemento “principal”, esto lo almacena dentro de una variable. Ahora bien, en la siguiente línea se asigna el evento “onclick” solo al primer elemento encontrado, pero se pudiera asignar el mismo evento a todos los elementos encontrados. Esto se logra recorriendo, mediante un bucle, todos los elementos encontrados.

### Anexo 3: jQuery, Eventos, Métodos y Selectores.

El contenido de este anexo es un resumen de las principales propiedades, funcionalidad y diferentes características de jQuery encontradas en (Point, Tutorials Point, s.f.).

Tipos de eventos.

<b>Evento</b>	<b>Descripción</b>
blur	Ocurre cuando el elemento pierde el foco.
change	Ocurre cuando el valor del elemento cambia.
click	Ocurre cuando se presiona un clic del ratón (mouse).
dblclick	Ocurre cuando se presiona un doble clic del ratón (mouse).
error	Ocurre cuando se genera un error en la carga.
focus	Ocurre cuando el elemento tiene el foco.
keydown	Ocurre cuando se presiona una tecla.
keypress	Ocurre cuando se presiona y suelta una tecla.
keyup	Ocurre cuando se suelta una tecla.
load	Ocurre cuando el documento está cargado por completo.
mousedown	Ocurre cuando el botón del ratón es presionado.
mouseenter	Ocurre cuando el cursor del ratón entra en una región del elemento.
mouseleave	Ocurre cuando el cursor del ratón sale de una región del elemento.
mousemove	Ocurre cuando el cursor del ratón se mueve.
mouseout	Ocurre cuando el cursor del ratón se mueve fuera de un elemento.
mouseover	Ocurre cuando el cursor del ratón está sobre un elemento.
mouseup	Ocurre cuando el botón del ratón es soltado.
resize	Ocurre cuando el tamaño de la ventana cambia.
scroll	Ocurre cuando en contenido de la ventana puede desplazarse hacia arriba y abajo.
select	Ocurre cuando un texto es seleccionado.
submit	Ocurre cuando el contenido de un formulario ha sido enviado.
unload	Ocurre cuando un documento no está cargado.

Tabla 11 Eventos jQuery

## Selectores de Elementos

### Función \$()

Todos los selectores que se utilizan en jQuery empiezan con el signo de dólar “\$”, seguido de paréntesis. Esta función es solo un sinónimo de la función jQuery() por lo que se podría usar cualquiera de las dos.

A continuación se detalla los selectores CSS más utilizados:

Selector	Descripción
Nombre	Selecciona todos los elementos por su el nombre de la etiqueta.
#ID	Selecciona un elemento, el cual coincida con el ID especificado.
.Clase	Selecciona todos los elementos que tengan una clase especificada.
Universal(*)	Selecciona todos los elementos disponibles en el DOM.
Múltiples Elementos E,F,G	Selecciona los elementos de todos los selectores especificados E, F o G.

Tabla 12 Selectores jQuery

Los selectores transversales fueron definidos para potenciar los selectores de CSS ya existentes. jQuery cuenta principalmente con dos selectores de este tipo: find y filter. El detalle de estos selectores se verá posteriormente.

## Selectores CSS

- **Selector por Nombre de etiqueta**

Selecciona el o los elementos que sean de un tipo de etiqueta especificado.

```
$(document).ready(function () {  
    $('div').css("background-color", "blue");  
});
```

Ilustración 126 Selector por Nombre etiqueta

En la imagen anterior se busca todos los elementos <div>, a los cuales se les otorgará un color azul para su fondo.

- **Selector por ID**

Selecciona un elemento HTML el cual tenga el atributo ID igual al especificado. Este selector siempre tiene que comenzar con el signo numeral y posterior el ID a buscar.

```
$(document).ready(function () {  
    $('#botonBuscar').css("background-color", "blue");  
    $('#div#seccion1').css("background-color", "yellow");  
});
```

Ilustración 127 Selector por ID elemento

En el ejemplo anterior se busca un elemento el cual tenga asignado su atributo ID igual a "botónBuscar". En la tercera línea del ejemplo anterior, se agrega primero el tipo de etiqueta y posterior el ID del elemento a buscar, esto aumenta la rapidez de la consulta ya que se concentrara solo en un tipo de etiqueta y no en todos los elementos del DOM.

- **Selector por Clase**

Selecciona uno o varios elementos que tengan asignado una clase específica. Este selector siempre debe empezar con el signo de punto (.) y posterior el nombre de la clase a buscar.

```
$(document).ready(function () {  
    $('.caja').css("background-color", "blue");  
    $('#div.caja').css("background-color", "blue");  
});
```

Ilustración 128 Selector por clase

En el ejemplo anterior se observa que en este selector se puede agregar el tipo de etiqueta que corresponde a el elemento a buscar, igual como en el selector por ID.

- **Selector Universal**

Selecciona todos los elementos disponibles en el documento HTML. Este selector siempre maneja un solo parámetro que es el signo de asterisco (\*).

```
$(document).ready(function () {  
    $('*').css("background-color", "blue");  
});
```

Ilustración 129 Selector universal

- **Selector Múltiples Elementos**

Este selector permite mezclar en la consulta varios tipos de selectores, aplicando los atributos a todos los elementos que cumplan cualquiera de los selectores especificados. En este selector múltiple cada selector debe estar separado por una coma.

```
$(document).ready(function () {  
    $('div, input').css("background-color", "blue");  
    $('p strong, .caja').css("background-color", "blue");  
    $('p strong, #div').css("background-color", "blue");  
});
```

Ilustración 130 Selector múltiple

En el ejemplo anterior en la fila tres, por ejemplo, se asigna un color de fondo azul a todos los elementos tipo <strong> que estén dentro de una etiqueta <p>, y también a todos los elementos que tengan asignado la clase “caja”. Este ejemplo muestra como mezclar varios selectores para poder asignar un mismo atributo.

## Selectores Transversales

- **Find**

Como su nombre lo indica, esta consulta busca todos los elementos de forma descendente que sean de un tipo específico. Una vez que encuentra elementos aplica los cambios o manipulaciones a todos ellos.

```

$("#new").click(function() {
    .....
    $("#input").find("th").css('background', '#ddd');
});

```

Ilustración 131 Selector Find

- **Filter**

Esta consulta se usa para filtrar elementos de entre una lista de elementos seleccionados. En la siguiente tabla se detalla las posibles variaciones que permite el uso de este selector.

Selector	Descripción
eq( posición )	<p>Selecciona un elemento de entre una lista.</p> <pre> \$('li').eq(2).addClass("estilo"); </pre>
filter( selector )	<p>Elimina elementos de entre una lista de elementos de acuerdo al selector ingresado.</p> <pre> \$('li').filter(".centro").addClass("seleccionados"); </pre>
filter( fn )	<p>Elimina elementos que no son especificados por la función ingresada como parámetro de entre una lista de elementos.</p>
is( selector )	<p>Devuelve verdadero o falso en caso de que el selector ingresado como parámetro devuelve o no elementos.</p>
map ( callback )	<p>Modifica la posición de los elementos de un objeto a otro.</p>
not ( selector )	<p>Elimina los elementos que se encuentren según el selector ingresado como parámetro.</p>
slice ( inicio, [fin])	<p>Selecciona un conjunto de elementos según parámetros ingresados.</p>

Tabla 13 Selectores transversales tipo Filtro

## Métodos

### Métodos para manipular atributos.

Método	Descripción
attr(propiedad)	Obtiene el valor de un atributo especificado. <pre>var titulo = \$('#cajal').attr('title');</pre>
removeAttr(nombre)	Elimina un atributo de un elemento según nombre atributo. <pre>\$('#cajal').removeAttr('value');</pre>
hasClass(clase)	Devuelve verdadero o falso, si se encuentra o no, agregado la clase especificada en un elemento. <pre>\$('#cajal').hasClass('estilo');</pre>
removeClass(clase)	Elimina la clase especificada de un elemento del DOM. <pre>\$('#cajal').removeClass('estilo');</pre>
toggleClass(clase)	Añade la clase especificada si esta no está incluida en el elemento, caso contrario, es decir si el elemento si tiene asignada la clase especificada, el evento la elimina. <pre>\$('#cajal').toggleClass('estilo');</pre>
html()	Obtiene el contenido HTML del primer elemento encontrado dentro del elemento que ejecuta el evento. <pre>var contenido = \$('#cajal').html();</pre>
html(valor)	Almacena contenido HTML dentro del elemento que ejecuta el evento. <pre>\$('#div').html("&lt;strong&gt;Nueva fila&lt;/strong&gt;");</pre>
text()	Obtiene el texto del elemento que ejecuta el evento. <pre>\$('#div').text();</pre>
text(valor)	Establece el valor del atributo "text" con el texto ingresado dentro del elemento.

	<pre>\$('#div').text("Hola Mundo");</pre>
val()	<p>Obtiene el valor ingresado en un elemento.</p> <pre>\$('#caja1').val();</pre>
val(valor)	<p>Guarda el valor ingresado como atributo "value" en el elemento.</p> <pre>\$('#caja1').val("250");</pre>

Tabla 14 Métodos para manipular atributos del DOM

### Métodos Transversales

Método	Descripción y Uso
add( selector )	<p>Añade más elementos, filtrados por cualquiera de los selectores, para que puedan ser manipulados.</p> <pre>\$("#top").add(".bottom").addClass("seleccionado");</pre>
andSelf()	<p>Añade la selección anterior a la nueva selección, y la manipulación afectara a todos los elementos encontrados de las varias consultas realizadas. Es usado cuando se tiene múltiples consultas transversales.</p> <pre>\$("#div").find("li").andSelf().addClass("estilo");</pre>
children( [selector] )	<p>Obtiene uno o varios elementos que se encuentran dentro del elemento que ejecuta el método (hijos). Todos los elementos encontrados van a ser afectados por la sentencia realizada. El método permite filtrar los elementos a manipular usando cualquiera de los selectores ya conocidos. El parámetro o filtro no es obligatorio.</p> <pre>\$("#div").children(".seleccionado").addClass("azul");</pre>
closest( selector )	<p>Este método trabaja sobre el elemento seleccionado. Filtra los elementos según el selector ingresado, en caso de no encontrar coincidencias en el elemento seleccionado,</p>

	<p>recorre hacia la parte superior del documento buscando coincidencias.</p> <pre>\$(document).bind("click", function (e) {     ...     \$(e.target).closest("li").toggleClass("seleccionado"); });</pre>
contents()	<p>Este método busca todos los elementos hijos que estén dentro del elemento que ejecuta el método.</p> <pre>\$('#div').find("ul").append("Texto agregado!"); \$('#div').find("ul").append("&lt;li&gt;Texto&lt;/li&gt;");</pre>
end()	<p>Revierte la última operación de destrucción o eliminación ejecutada, modificando los elementos del documento HTML.</p> <pre>\$( "p" ).find( "span" ).end().css( "border", "2px red solid" );</pre>
find( selector )	<p>Busca elementos de forma descendente según el filtro del selector ingresado.</p> <pre>\$( "p" ).find( "span" ).addClass( "seleccionado" );</pre>
next( [selector] )	<p>Busca elementos según el filtro ingresado. La manipulación afectara solo al segundo elemento encontrado.</p> <pre>\$( "p" ).next( ".seleccionado" ).addClass( "estilo" );</pre>
nextAll( [selector] )	<p>Busca elementos del mismo tipo según el selector ingresado. Afecta a todos menos al primero de los elementos encontrados.</p> <pre>\$( "div:first" ).nextAll( ).addClass( "estilo" );</pre>
offsetParent()	<p>Devuelve la posición del primer elemento padre que contiene el elemento que ejecuta el método.</p> <pre>\$( '#cajal' ).offsetParent( );</pre>
parent( [selector] )	<p>Busca elementos que estén creados bajo el mismo nivel de estructura, es decir todos los elementos que estén dentro del mismo objeto. El parámetro es condicional, si es necesario filtrar los elementos a buscar se puede utilizar un selector.</p>

	<code>\$("#p").parent().addClass('estilo');</code>
parents( [selector] )	<p>Obtiene un conjunto de elementos padres, es decir obtiene todos los elementos que están en niveles superiores al elemento que ejecuta el método. El parámetro es condicional, si es necesario filtrar los elementos a buscar se puede utilizar un selector.</p> <pre>var parentEls = \$("#p").parents().map(function () {     return this.tagName; }).get().join(", ");</pre>
prev( [selector] )	<p>Obtiene el elemento inmediatamente anterior al elemento que ejecuta el método. Se puede utilizar el parámetro para filtrar los elementos agregando un selector.</p> <pre>\$("#p").prev("#seleccionado").addClass("estilo");</pre>
prevAll( [selector] )	<p>Obtiene todos los elementos que estén en el mismo nivel de la estructura del documento HTML.</p> <pre>\$("#div:last").prevAll().addClass("estilo");</pre>
siblings( [selector] )	<p>Obtiene todos los elementos hermanos del elemento que ejecuta el método. Es posible filtrar la búsqueda usando cualquier de los selectores.</p> <pre>\$("#cajal").siblings().css( "background-color", "red" );</pre>

Tabla 15 Métodos Transversales

#### Métodos para manipular árbol DOM.

Método	Descripción y Uso
after( contenido )	<p>Inserta el contenido agregado como parámetro después del elemento que ejecuta el método.</p> <p>El contenido enviado puede ser un valor tipo texto o también puede ser una cadena con formato HTML.</p>

	<pre> \$("div").click(function () {     \$(this).after('&lt;div class="div"&gt;&lt;/div&gt;' ); }); </pre>
append( contenido )	<p>Añade el contenido enviado como parámetro dentro del elemento que ejecuta el método.</p> <p>El contenido enviado puede ser un valor tipo texto o también puede ser una cadena con formato HTML.</p> <pre> \$("div").click(function () {     \$(this).append('&lt;div class="div" style="color: blue"&gt;Añadido&lt;/div&gt;' ); }); </pre>
appendTo( selector )	<p>Añade todo el contenido del elemento que ejecuta el método dentro de otro elemento especificado como parámetro utilizando cualquiera de los selectores disponibles.</p> <pre> \$("div").click(function () {     \$(this).appendTo("#div"); }); </pre>
before( contenido )	<p>Inserta el contenido especificado como parámetro, antes del elemento que ejecuta el método.</p> <pre> \$("div").click(function () {     \$(this).before('&lt;div class="div"&gt;&lt;/div&gt;' ); }); </pre>
clone( bool )	<p>Permite clonar elementos del DOM dentro de una ubicación específica.</p> <p>El parámetro es condicional, este indica si al clonar los elementos estos se copian con sus atributos y eventos.</p> <pre> \$("div").click(function () {     \$(this).clone(true).insertAfter(this); }); </pre>
clone()	<p>Permite clonar elementos del DOM dentro de una ubicación específica. Los elementos clonados también se copian con todos sus atributos y estilos.</p>

	<pre>\$("#div").click(function () {     \$(this).clone().insertAfter(this); });</pre>
empty()	<p>Elimina todos los elementos hijos, es decir todos los elementos que estén dentro del elemento que ejecuta el método.</p> <pre>\$("#div").click(function () {     \$(this).empty(); });</pre>
html( valor )	<p>Modifica el contenido del elemento que ejecuta el método por el contenido que se agrega como parámetro.</p> <pre>\$("#div").click(function () {     \$(this).html( "&lt;h1&gt;Clic en otro lugar&lt;/h1&gt;" ); });</pre>
html()	<p>Obtiene el contenido HTML del elemento que ejecuta el método.</p> <pre>var content = \$('#div').html();</pre>
insertAfter( selector )	<p>Inserta todo el contenido de un elemento antes de otro elemento pasado como parámetro. Para especificar el elemento del parámetro se puede utilizar cualquiera de los selectores.</p> <pre>\$("#div").click(function () {     \$("#div").insertAfter(this); });</pre>
insertBefore ( selector )	<p>Inserta todo el contenido de un elemento después de otro elemento pasado como parámetro. Para especificar el elemento del parámetro se puede utilizar cualquiera de los selectores.</p> <pre>\$("#div").click(function () {     \$("#div").insertBefore(this); });</pre>
prepend( contenido )	<p>Inserta el contenido pasado como parámetro dentro del elemento que ejecuta el método. Este método es parecido al método append().</p>

	<pre> \$("div").click(function () {     \$(this).prepend('&lt;div class="div"&gt;&lt;/div&gt;'); }); </pre>
prependTo( selector )	<p>Inserta todo el contenido del elemento que ejecuta el método, dentro del elemento especificado como parámetro. Este método es parecido al método appendTo().</p> <pre> \$("div").click(function () {     \$(this).prependTo("#div"); }); </pre>
remove ( expresión )	<p>Elimina todos los elementos que estén dentro del elemento que ejecuta el método. Método muy parecido al método empty().</p> <pre> \$("div").click(function () {     \$(this).remove(); }); </pre>
replaceAll( selector )	<p>Reemplaza todos los elementos que se agregan como parámetro por el (los) elemento que ejecuta el método.</p> <pre> \$("div").click(function () {     \$('&lt;div class="div"&gt;&lt;/div&gt;').replaceAll( this ); }); </pre>
replaceWith ( contenido )	<p>Reemplaza el contenido del elemento que ejecuta el método por el contenido ingresado como parámetro.</p> <pre> \$("div").click(function () {     \$(this).replaceWith('&lt;div class="div"&gt;&lt;/div&gt;'); }); </pre>
wrap( elemento )	<p>La función puede tomar cualquier cadena u objeto que pueda pasar a la función jQuery para especificar una estructura DOM. Esta estructura puede anidarse varios niveles de profundidad, pero debe contener sólo un elemento íntimo.</p> <pre> \$("div").click(function () {     var content = '&lt;div class="div"&gt;&lt;/div&gt;';     \$("#destination").wrap(document.createElement(content)); }); </pre>
wrap( html )	<p>La función puede tomar cualquier objeto que pueda pasar a la función jQuery para especificar una estructura DOM. Esta estructura puede anidarse varios niveles de profundidad, pero debe contener sólo un elemento íntimo.</p>

	<pre> \$("div").click(function () {     var content = '&lt;div class="div"&gt;&lt;/div&gt;';     \$("#destination").wrap(content); }); </pre>
wrapAll( elemento )	<p>Permite anidar elementos en varios niveles de profundidad. La estructura se envolverá alrededor de todos los elementos del conjunto de elementos coincidentes, como un solo grupo.</p> <pre> \$("div").click(function () {     var content = '&lt;div class="div"&gt;&lt;/div&gt;';     \$("div").wrapAll( document.createElement( content ) ); }); </pre>
wrapAll( html )	<p>Permite anidar elemento html en varios niveles de profundidad. La estructura se envolverá alrededor de todos los elementos del conjunto de elementos coincidentes, como un solo grupo.</p> <pre> \$("div").click(function () {     var content = '&lt;div class="div"&gt;&lt;/div&gt;';     \$("div").wrapAll( content ); }); </pre>
wrapInner( elem )	<p>Esta estructura anida elementos en varios niveles de profundidad, pero debe contener sólo un elemento íntimo. La estructura se envolverá alrededor del contenido de cada uno de los elementos en el conjunto de elementos coincidentes.</p> <pre> \$("div").click(function () {     \$(this).wrapInner( document.createElement( "b" ) ); }); </pre>
wrapInner( html )	<p>Esta estructura anida elementos en formato HTML en varios niveles de profundidad, pero debe contener sólo un elemento íntimo. La estructura se envolverá alrededor del contenido de cada uno de los elementos en el conjunto de elementos coincidentes.</p> <pre> \$("div").click(function () {     var content = "&lt;b&gt;&lt;/b&gt;";     \$(this).wrapInner( content ); }); </pre>

Tabla 16 Métodos manipulan DOM

## Métodos del objeto "Event".

Método	Descripción
preventDefault()	<p>Especifica al navegador que ejecute la acción por defecto del elemento que ejecuta el evento.</p> <pre data-bbox="778 546 1129 622">\$("a").click(function(event){     event.preventDefault(); });</pre>
isDefaultPrevented()	<p>Retorna el tiempo en que el método preventDefault() fue llamado.</p> <pre data-bbox="778 797 1401 965">\$("a").click(function(event){     event.preventDefault();     if ( event.isDefaultPrevented() ){         alert( "Comportamiento por defecto esta deshabilitado" );     } else {         alert( "Comportamiento por defecto esta habilitado" );     } });</pre>
stopPropagation()	<p>Impide que el evento manipule el árbol DOM, evitando que los controladores de origen sean notificados del evento.</p> <pre data-bbox="778 1189 1401 1305">\$("div").click(function(event){     alert("This is : " + \$(this).text());     // Comente la siguiente línea para ver la diferencia.     event.stopPropagation(); });</pre>
isPropagationStopped()	<p>Devuelve si el evento "stopPropagation()" se ha llamado alguna vez en el objeto "event".</p> <p>Devuelve verdadero si el evento ha sido llamado, caso contrario devuelve falso.</p> <pre data-bbox="778 1630 1401 1861">\$("div").click(function(event){     event.stopPropagation();     if ( event.isPropagationStopped() ){         alert( "Evento esta deshabilitado." );     }else{         alert( "Evento esta habilitado" );     } });</pre>
stopImmediatePropagation()	<p>Detiene el resto de los manejadores de ser ejecutados e impide que el evento manipule hasta el árbol DOM.</p>

	<pre> \$("#div").click(function(event){     alert("1 - Este es : " + \$(this).text());     // Comente la siguiente línea para ver la diferencia.     event.stopImmediatePropagation(); });  // Este evento no se ejecutará. \$("#div").click(function(event){     alert("2 - Este es : " + \$(this).text()); }); </pre>
isImmediatePropagationStopped()	<p>Devuelve si el evento “event.stopImmediatePropagation()” se llamó alguna vez en este objeto de evento.</p> <pre> \$("#div").click(function(event){     event.stopImmediatePropagation();     if ( event.isImmediatePropagationStopped() ){         alert( "El evento esta deshabilitado." );     }else{         alert( "El evento esta habilitado." );     } }); </pre>

Tabla 17 Métodos del objeto "Event"

Métodos de manipulación de eventos.

Estos métodos permiten la manipulación de los eventos disparados por los elementos del DOM. Algunos de los parámetros que tienen los métodos a continuación están entre corchetes, esto significa que no son obligatorios.

- **Método bind(tipo, [ datos ], función )**

Este método permite agregar uno o más eventos que puedan manipular acciones de los elementos. Permitiendo modificar las acciones por defecto de cada una de las acciones generadas por los eventos.

Los eventos que pueden ser agregados están detallados en el Anexo 3: Tipos de eventos.

Parámetros:

- Tipo: Uno o más tipos de eventos, separados por un espacio.
- Datos: Parámetro adicional que permite enviar datos al evento.

- Función: Función que se realizará en el evento.

Ejemplo:

```
$('#div').bind('click', function( event ){
    alert('Hola Mundo!');
});
```

Ilustración 132 Método bind

- **Método off( eventos, [ selector ], [ manejador ] )**

Este método permite detener o suspender la ejecución de un evento que está por dispararse.

Parámetros:

- eventos: Tipos de eventos, si son varios, separados con un espacio.
- selector: Filtro de elementos que se puede agregar al evento.
- manejador: Función que se realizará en el evento.

Ejemplo:

```
function un_Click() {
    $("#div").show().fadeOut("slow");
}

$("#agregar").click(function () {
    $("#boton").on("click", un_Click).text("Click!");
});

$("#quitar").click(function () {
    $("#boton").off("click", un_Click).text("No hace nada...");
});
```

Ilustración 133 Método off

- **Método hover( sobre, fuera )**

Este método simula la acción del ratón, de pasar el cursor sobre un objeto y quitar el cursor del objeto.

Parámetros:

- sobre.- Función que se ejecutará cuando el cursor del ratón pase sobre el objeto al cual se agrega el método.
- fuera.- Función que se ejecutará cuando el cursor del ratón sale del área del objeto al cual se agrega el método.

Ejemplo:

```

$( 'div' ).hover(
  function () {
    $(this).css({ "background-color": "red" });
  },
  function () {
    $(this).css({ "background-color": "blue" });
  }
);

```

Ilustración 134 Método hover

- **Método on( eventos, [ selector ], [ datos ], manejador )**

Este método agrega una función que manejará el evento. La lista de eventos disponibles está en el Anexo 3: Tipos de eventos.

Parámetros:

- eventos.- Uno o varios eventos, separados por un espacio, que se agregan al elemento.
- selector.- Parámetro opcional que establece un filtro de elementos a los cuales afectará el método.
- datos.- Datos adicionales que son pasados al evento usando "event.data".
- manejador.- Función que se agrega al evento, mismo que manejará las acciones del evento.

Ejemplo:

```

$( 'div' ).on( 'click', function( event ) {
  alert( 'Hola Mundo!' );
} );

```

Ilustración 135 Método on

- **Método one( tipo, [datos], función )**

Este método agrega una función manejadora para uno o más tipos de eventos que pueden ser ejecutados por un elemento. Lo diferente de este método es que la función se ejecutará solo una vez por cada elemento.

Parámetros:

- tipo.- Tipo de evento agregado al elemento.
- datos.- Parámetro opcional que permite enviar información adicional a la función que se ejecuta mediante el código "event.data".
- función.- Función que se agrega al evento, la cual maneja las acciones que se realizarán en el evento.

Ejemplo:

```
$('#div').one('click', function( event ){  
    alert('Hola Mundo!');  
});
```

Ilustración 136 Método one

- **Método ready( función )**

Este método agrega una función que se ejecutará solo cuando el DOM esté listo para ser manipulado. Este método es una alternativa al evento "window.onload" de JavaScript.

Parámetros:

- función.- Función que se ejecuta en el método.

Ejemplo:

```
$(document).ready(function() {  
    $("#div").text("El DOM está cargado completamente...");  
});
```

Ilustración 137 Método ready

- **Método trigger( evento, datos )**

Este método ejecuta un evento, que no ha sido disparado, pero si esté agregado en cualquiera de los elementos del DOM.

Parámetros:

- evento.- Tipo de evento a disparar.
- datos.- Datos adicionales que se pueden enviar a la función que se ejecuta en el método disparado. Usa el código "event.data".

Ejemplo:

```
$("#div1").click( function () {  
    $("#div2").trigger('click');  
});  
  
$("#div2").click( function () {  
    alert( "Evento click");  
});
```

Ilustración 138 Método trigger

- **Método triggerHandler( evento, [ datos ])**

Este método ejecuta todos los eventos que pueden ejecutarse en un elemento (para un tipo de evento específico) sin ejecutar acciones por defecto de los navegadores o eventos propios.

Parámetros:

- evento. Tipo de evento o tipo de disparador a ejecutar.
- datos. Parámetro opcional, permite pasar datos adicionales a la función que se ejecuta, usando el código "event.data".

Ejemplo del uso:

```

$("#antiguo").click(function() {
    .....
    $("input").trigger("focus");
});
$("#nuevo").click(function() {
    .....
    $("input").triggerHandler("focus");
});
$("input").focus(function() {
    .....
    $("<span>Focused!</span>").appendTo("body").fadeOut(1000);
});

```

Ilustración 139 Método triggerHandler

- **Método unbind( [ tipo ], [ función ])**

Este método remueve los eventos que están por dispararse de cualquier de los elementos del DOM.

Parámetros:

- tipo.- Tipos de eventos que pueden estar separados por un espacio.
- función.- Parámetro opcional, que especifica la función que se ejecutará en el evento.

Ejemplo de su uso:

```

function aClick() {
    .....
    $("div").show().fadeOut("slow");
}

$("#quitar").click(function () {
    .....
    $("#boton").unbind('click', aClick).text("No hace nada...");
});

```

Ilustración 140 Método unbind

Métodos AJAX.

A continuación se detalla los métodos AJAX disponibles:

- **jQuery.ajax( opciones ) / \$.ajax( opciones )**

Método usado para realizar llamadas al servidor usando el método HTTP request.

Parámetros:

- opciones.- Ajax tiene una gran lista de opciones disponibles que pueden ser enviadas junto a la llamada al servidor. Todos estos valores son opciones por lo que si no es necesario simplemente se especifica la url.

En la siguiente tabla se lista todas las opciones disponibles:

Opción	Descripción
async	Valor tipo booleano que indica el tipo de llamada asíncrona. Por defecto el valor es verdadero (true), en caso de colocar falso (false) la llamada será de tipo síncrona.
beforeSend	Función que es ejecutada después de haber realizado la llamada.
cache	Variable tipo booleano que especifica si la página solicitada podrá ser almacenada en cache o no por el navegador. El valor por defecto es verdadero (true).
complete	Función que se ejecuta solo cuando la llamada al servidor terminó.
contentType	Cadena cuyo contenido tiene formato tipo MIME para establecer la solicitud. El valor por defecto es "application/x-www-form-urlencoded".
context	Objeto que será el contexto de todas las llamadas relacionadas con AJAX. Por defecto el objeto que representa las configuraciones es (\$.ajaxSettings).
converters	Especifica el tipo de convertidor que aplica el objeto.
crossDomain	Variable de tipo booleano que especifica la posibilidad de poder re direccionar de un servidor a otro. El valor por defecto es verdadero (true).
data	Cadena de texto que es enviada al servidor con la solicitud. Es usada para enviar variables al servidor.
dataFilter	Función que se utiliza para manejar los datos de respuesta que no han sido procesadas por la solicitud XMLHttpRequest.
dataType	Cadena que define el tipo de dato esperado por el servidor ( xml, html, json, script, etc.).
error	Función que es ejecutada si la solicitud produjo un error.

global	Parámetro tipo booleano que especifica si los controladores globales estarán activos en la solicitud. Si el parámetro se establece en verdadero (true) indica que los controladores (ajaxStart, ajaxStop) podrán controlar varios eventos de Ajax.
headers	Objeto compuesto de clave y valor que es enviado en el encabezado, para enviar junto con las solicitudes que utiliza el transporte XMLHttpRequest.
ifModified	Parámetro de tipo booleano que especifica si el servidor debe comprobar si la página está modificada antes de responder a la solicitud.
isLocal	Parámetro booleano que permite que el entorno actual sea reconocido como "local".
jsonp	Anula el nombre de la función de devolución de llamada en una solicitud jsonp.
jsonpCallback	Especifica el nombre de una función que se ejecutará cuando se regrese de la solicitud JSONP.
method	Especifica el tipo de método HTTP usado en la solicitud. Valor por defecto "GET".
mimeType	Un tipo mime usado para anular el tipo mime XHR.
password	Contraseña que va a ser usado para acceder a una solicitud HTTP.
processData	Parámetro tipo booleano
scriptCharset	Solo se aplica cuando se utiliza el transporte "script". Conjunto de atributos tipo charset en el script usado en la solicitud.
statusCode	Objeto de códigos y funciones numéricas HTTP que se llamará cuando la respuesta tenga el código correspondiente.
success	Función que se ejecuta si la solicitud termina con éxito.
timeout	Tiempo, en milisegundos, que la consulta esperará la respuesta de la solicitud.
traditional	Establece como verdadero si desea utilizar el estilo tradicional de serialización.
type	Cadena que define el tipo de método HTTP que usará la solicitud (GET, POST). El valor por defecto es "GET".
url	Cadena que contiene la dirección del recurso que se solicita.
username	Nombre de usuario que puede ser usado para acceder a una solicitud HTTP.

xhr	
xhrFields	Objeto compuesto por el nombre y valor, el cual especifica el objeto XHR nativo.

Tabla 18 Opciones parámetro método \$.ajax

Ejemplo de su uso:

```
$("#boton").click(function(event){
    $.ajax( {
        url: '/jquery/result.html',
        success: function(data) {
            $('#stage').html(data);
        }
    });
});
```

Ilustración 141 Método \$.ajax

- **\$.ajaxSetup( opciones )**

Método que establece las configuraciones globales para futuras solicitudes AJAX.

Parámetros:

- opciones.- Conjunto de valores o claves que configuran la solicitud Ajax. Las opciones son las mismas que las detallas para el método “\$.ajax()”.

Ejemplo de su uso:

```
$("#driver").click(function(event){
    // Do global setting.
    $.ajaxSetup({
        url: "/jquery/result.html"
    });

    $.ajax( {
        success: function(data) {
            $('#stage').html(data);
        }
    });
});
```

Ilustración 142 Método \$.ajaxSetup

En el ejemplo anterior, se observa como en el primer método se establece una dirección url que será configurada para todos los demás métodos que ejecuten una solicitud AJAX.

- **\$.get( url, [ datos ], [ función ], [ tipo ])**

Método que es usado para obtener información desde el servidor usando solicitudes HTTP de tipo "GET".

Parámetros:

- URL. Dirección del recurso al cual se realiza la petición HTTP.
- datos. Parámetro opcional que contiene una clave y valor que puede ser enviado al servidor.
- función. Función que se ejecutará cuando la información este cargada completamente.
- tipo. Parámetro opcional que representa el tipo de dato que será retornada a la función: "xml", "html", "script", "json", "jsonp", "text".

Ejemplo del uso:

```
$.get("ajax/test.html", function( data ) {
    $('#cajal').html( data );
    alert("Datos cargados.");
});
```

Ilustración 143 Método \$.get

- **\$.getJSON( url, [ datos ], [ función ])**

Método que permite cargar información tipo JSON obtenido del servidor usando una solicitud HTTP de tipo "GET". El método devuelve un objeto de tipo XMLHttpRequest.

Parámetros:

- url.- Dirección del recurso al cual se realiza la petición HTTP.
- datos.- Parámetro opcional que contiene una clave y valor que puede ser enviado al servidor.
- función.- Función que se ejecutará cuando la información este cargada completamente.

Ejemplo de su uso:

```
$("#boton").click(function(event) {  
    $.getJSON(  
        "result.php",  
        { name: "Zara" },  
        function(data) {  
            $('#div').html(data);  
        }  
    );  
});
```

Ilustración 144 Método \$.getJSON

- **\$.getScript( url, [ función ] )**

Este método carga y ejecuta un archivo JavaScript usando una solicitud HTTP de tipo "GET". Este método retorna un objeto de tipo XMLHttpRequest.

Parámetros:

- url.- Dirección del recurso al cual se realiza la petición HTTP.
- función.- Parámetro opcional que define una función que se ejecutará cuando la información este cargada completamente.

Ejemplo de su uso:

```
function CheckJS() {  
    alert("Esto es JavaScript");  
}  
  
$("#driver").click(function(event) {  
    $.getScript('result.js', function(jd) {  
        // LLama a una función definida en la sección script  
        CheckJS();  
    });  
});
```

Ilustración 145 Método \$.getScript

- **\$.post( url, [ datos ], [ función\_regreso ], [ tipo ])**

Método que carga una página obtenido desde el servidor usando una solicitud HTTP de tipo "GET". El método retorna un objeto XMLHttpRequest.

Parámetros:

- URL.- Dirección del recurso al cual se realiza la petición HTTP.
- datos.- Parámetro opcional que contiene una clave y valor que puede ser enviado al servidor.
- función.- Función que se ejecutará cuando la información este cargada completamente.
- tipo.- Parámetro opcional que representa el tipo de dato que será retornada a la función: "xml", "html", "script", "json", "jsonp", "text".

Ejemplo del uso:

```
$("#driver").click(function(event){
    $.post(
        "result.php",
        { name: "Zara" },
        function(data) {
            $('#stage').html(data);
        }
    );
});
```

Ilustración 146 Método \$.post

- **serialize()**

Este método serializa un conjunto de elementos de texto dentro de una sola cadena de texto.

Ejemplo de su uso:

```
$("#new").click(function(){
    var str = $("form").serialize();
    $("#resultado").text( str );
});
```

Ilustración 147 Método serialize

- **serializeArray()**

Este método serializa todos los formularios y todos los elementos. Este método retorna un objeto con formato JSON.

Ejemplo de su uso:

```
$.get("ajax/test.html", function( data ) {
    $('#caja1').html( data );
    var contenido = $('#caja1').serializeArray();
});
```

Ilustración 148 Método serializeArray

#### Anexo 4: CSS3, Nuevas Funciones, Propiedades, Transiciones y Selectores.

En esta anexo se aborda los principales temas obtenidos del libro (Puig, 2015), mismos que le servirá para entender las principales funcionalidades del lenguaje de estilos CSS3.

Los selectores CSS son la herramienta más potente del lenguaje de estilos, puesto que permiten seleccionar diferentes elementos del contenido HTML en función de la etiqueta o de sus atributos sin tener que hacer uso de su clase, su ID o JavaScript.

Tipos de selectores.

- Selectores de atributos
  - ✓ [att^ = "valor"]. Selecciona elementos con un atributo que empieza con "valor".
  - ✓ [att\$ = "valor"]. Selecciona elementos con un atributo que acaba con la palabra valor.
  - ✓ [att\* = "valor"]. Selecciona elementos que contienen un atributo que tenga la palabra valor.
- Combinadores

- ✓ ~ Sibling General (Hermano). Selecciona los elementos que son “hermanos” de uno. Una etiqueta hermana es la que existe en el mismo nivel, o que tiene un padre en común y que está a continuación de la referenciada.
- Pseudo-clases
    - ✓ :nth-child(n). Selecciona elementos basándose en la posición de los hijos. Puede utilizar números, expresiones y las palabras odd, even (impar, par). Fácilmente podemos hacer el típico efecto cebra en las tablas.
    - ✓ :nth-last-child(n). Sigue la misma idea que la anterior, pero selecciona a partir del último elemento.
    - ✓ :last-child. Selecciona el último elemento de una lista.
    - ✓ :checked. Selecciona elementos que están marcados.
    - ✓ :empty. Selecciona elementos que están vacíos.
    - ✓ :not. Selecciona elementos que no cumplen la declaración especificada.

- Pseudo-elementos

Estos elementos seleccionan parte del contenido que esté dentro de una etiqueta. Una vez seleccionado el elemento podemos agregarle atributos.

```
p:first-letter { font-size: 15px }
```

Ilustración 149 Pseudo-elementos CSS3

Otra de las principales novedades es la incorporación del pseudo-elemento “:selection”. El cual puede ser utilizado solamente en campos del formulario, este elemento sirve para poder obtener y manipular el elemento seleccionado dentro del formulario. Es decir cuando en un formulario, un elemento <input> tiene el foco, inmediatamente recuperara el o los estilos agregados.

```
input:selection { background-color: #eaeaea; }
```

Ilustración 150 Pseudo-elemento "selection" CSS3

Funciones de manipulación de colores y opacidad.

- **RGBA.**- Esta función no es nueva ya que surgió en la versión CSS2. En esta versión se incorpora un nuevo modelo de color, el basado en saturación (HSL),

y la posibilidad de especificar canal alpha sobre los colores. En el siguiente grafico se ilustra un ejemplo de cómo asignar un color:

```
p { color: rgba(0,0,0,0.5) }
```

Ilustración 151 Modelo Color RGBA CSS3

En el ejemplo anterior el estilo cambiara el elemento <p>, modificando el color a uno negro con una transparencia del 50%.

- **Colores HSL.**- HSL (hue, saturation, light) es un modelo de color, que traducido seria tono, saturación y brillo. Este modelo maneja los tres parámetros manipulando la tonalidad del color dada, además de poder alterar la saturación y el color de la luz.

```
background: hsl(320,80%,30%);  
background: hsla(320,100%,10%);
```

Ilustración 152 Modelo Color HSL CSS3

- **Opacidad.**- Este modelo permite manipular la transparencia de un selector. La principal diferencia entre este modelo y el modelo RGB es que RGB aplica solo al objeto mientras que la transparencia se aplica al objeto y a todos sus hijos o elementos contenidos.

```

```

Ilustración 153 Modelo Opacidad CSS3

Nuevas Propiedades.

- **Esquinas Redondeadas.**- Esta característica es una de las más esperadas por la gente que diseña y desarrolla software, permitiendo crear cajas (divs, span, etc) con esquinas redondeadas según un radio.

Para hacer los bordes redondeados se utiliza la propiedad “border-radius”, a la cual se asigna un valor que indica el radio del borde a crear. Es importante indicar

que el atributo “border-radius” manipula a todos los bordes del elemento, es decir, el tamaño que especifiquemos será el tamaño de los cuatro bordes del elemento.

```
.elemento {  
  border-radius: 15px;  
}
```

Ilustración 154 Propiedad border-radius CSS3

En la imagen anterior, se agrega la propiedad “border-radius” al elemento que tenga su ID igual a “elemento”.

- **Bordes individuales.-** Esta nueva funcionalidad permite asignar esta propiedad a un borde en específico del elemento. Es decir solo una de las esquinas del elemento puede tener un efecto redondeado.

Para poder aplicar el efecto podemos utilizar cualquier de las siguientes propiedades:

- border-bottom-left-radius
- border-bottom-right-radius
- border-top-left-radius
- borde-top-right-radius

Cada una de las propiedades descritas indica un lado diferente del elemento, al cual podemos asignar el efecto redondeado.

- **Sombras.-** Esta propiedad se incorpora nuevamente a esta versión pese a que en la versión 2.1 fue eliminada. Para el uso de este efecto se utiliza la propiedad “box-shadow” e ingresando dos parámetros principalmente, pero pueden ser cinco parámetros para poder manipular por completo la propiedad.

Las cinco propiedades definen: desplazamiento horizontal, desplazamiento vertical, desenfocado, extensión y color de sombra. De manera opción al final podemos insertar la palabra reservada “Inset”, que especifica que la sombra se hará en la parte interna del elemento y no como normalmente en la parte externa.

```
.elemento {  
  border-radius: 15px;  
  box-shadow: 10px 5px 0 0 #888;  
}
```

Ilustración 155 Propiedad box-shadow CSS3

Como información adicional se menciona que un elemento no se restringe a tener una sola sombra, se puede crear varias sombras para un mismo elemento simplemente separando con una coma y volviendo a agregar los parámetros.

```
.elemento {  
  border-radius: 15px;  
  box-shadow: 10px 5px 0 0 #888, 10px 5px 0 0 #888;  
}
```

Ilustración 156 Propiedad box-shadow avanzado CSS3

- **Múltiples imágenes de fondo.**- Otro efecto importante, es la posibilidad de agregar varias imágenes de fondo en un solo elemento. Para esto simplemente separamos con una coma la URL de la imagen y volvemos a ingresar otra como podemos ver en la siguiente imagen.

```
.elemento {  
  background-image: url(imagen1.png), url(canvas.png);  
}
```

Ilustración 157 Propiedad background-image CSS3

De igual manera se puede especificar la posición de las imágenes de fondo, separando simplemente con una coma la posición de cada imagen.

- **Bordes con imágenes.**- Esta propiedad define una imagen que puede ser mostrada dentro de un borde del elemento. Por obvios motivos la imagen se mostrará siempre y cuando el elemento tenga un borde y además un tamaño adecuado. Para este efecto se utiliza la propiedad "border-image".

Esta propiedad puede recibir 3 parámetros, el primero la URL de la imagen, el segundo es un valor porcentaje que está relacionado con la parte que se quiere emplear y el ultimo parámetro especifica si la imagen se repetirá o se mostrara una sola vez.

```
.elemento {  
  border-image: url(imagen-borde.png) 5% repeat;  
}
```

Ilustración 158 Propiedad border-image CSS3

- **Columnas de texto.**- Otra de las novedades es la posibilidad de trabajar con columnas de texto. Realmente, con el ancho actual de las pantallas ha sido necesario hacerlo, puesto que un ancho demasiado grande en los párrafos de texto afecta a su legibilidad. Así, desde la versión 3, se puede utilizar diseños con columnas (Puig, 2015, pág. 21).

Las propiedades que se pueden utilizar son las que se observa en la siguiente imagen:

```
.elemento {  
  -moz-column-count: 3;  
  -webkit-column-count: 3;  
  -moz-column-width: 200px;  
  -webkit-column-width: 200px;  
  -moz-column-gap: 20px;  
  -webkit-column-gap: 20px;  
}
```

Ilustración 159 Columnas de texto CSS3

Al utilizar el estilo que se detalla en la imagen se genera una estructura de tres columnas siempre y cuando puedan cumplir con el tamaño de ancho de columna preferido, fijado con la propiedad “-column-width”, si se hiciera así, emplearía la parte proporcional del área disponible. Con la propiedad “-column-gap: 20px” definimos el margen que se quiere entre columnas, y también se encuentra la propiedad “column-rule”, con la que se puede definir una franja que divida las columnas.

- **WebFonts.**- Esta etiqueta permite adjuntar tipografías dentro de un documento HTML. Para agregar este efecto se utiliza la etiqueta “@font-face”, a la cual se puede agregar atributos como el tipo de letra, tamaño, entre otros.

```
@font-face {  
  font-family: Gentium;  
  src: url(gentium.otf);  
}
```

Ilustración 160 Propiedad WebFonts CSS3

Un problema con esta etiqueta, es que los navegadores tienen formatos diferentes lo que causa irregularidades en su uso. En la siguiente imagen se observa como agregar una fuente que puede ser aplicada en cualquier navegador.

```
@font-face {
  font-family: 'UbuntuRegular';
  src: url('webfont.eot');
  src: url('webfont.eot?#iefix') format('embedded-opentype'),
  url('webfont.woff') format('woff'),
  url('webfont.ttf') format('truetype'),
  url('webfont.svg#UbuntuRegular') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

Ilustración 161 Propiedad WebFonts CSS3 Varios Navegadores

- **MediaQueries.**- Desde la versión 2.1 de los CSS, existe la posibilidad de definir estilos en función del uso de la hoja: una para la pantalla (screen), otra para la impresión (print), otra para los lectores de voz (voice). A partir de la especificación 3 esto va un paso más allá, y nos permite definir estilos específicos para diferentes tamaños de pantalla.

```
@media screen and (max-width: 600px) {
  .class { background: #ccc; }
}
```

Ilustración 162 Propiedad MediaQueries CSS3

En la imagen anterior se definen estilos específicos para navegadores con un ancho de pantalla menor de 600 píxeles.

Para definir una mediaquerie existe dos posibilidades: utilizar min-width o mindevice-width. Con la primera se hace referencia al área visible en estos momentos en el dispositivo, mientras que con device-width se referencia a la resolución del dispositivo. La posibilidad de poder trabajar así, realmente, ofrece soluciones óptimas, puesto que permite emplear sólo CSS para crear una versión móvil de un sitio web.

## Nuevas Transiciones.

Las transiciones CSS son pequeños cambios en las propiedades de la hoja de estilos, desencadenados por acontecimientos generados por interacciones del usuario, como por ejemplo cuando el ratón pasa por encima de algo (hover) o cuando un campo de formulario cambia a otro campo, etc. En una transición, estos cambios en las propiedades se producen de manera progresiva durante un intervalo de tiempo (Puig, 2015, pág. 23).

```
.estilo {  
    background-color: #888;  
}  
.estilo:hover {  
    background-color: #111;  
}
```

Ilustración 163 Transiciones CSS3

En la imagen anterior, se observa que el color de fondo del elemento cambiara en el momento en que el ratón pasa por encima de este.

Las transiciones en CSS también brindan la posibilidad de definir la duración, retraso y la interpolación de tiempo, que tendrá dicha transición.

```
<style>  
    .estilo {  
        background-color: #888;  
        -webkit-transition: background 0.9s ease 1.5s;  
    }  
    .estilo:hover {  
        background-color: #624141;  
    }  
</style>
```

Ilustración 164 Transiciones CSS3 Duración y Retraso

En la imagen anterior se observa el uso del atributo “-webkit-transition”, este atributo es el que permite definir el tipo de transición que tendrá el objeto, además de la posibilidad de incluir varios parámetros adicionales. Según el orden definido los parámetros especifican: tipo de transición, duración de transición, tipo de transición y finalmente el tiempo de retraso que tendrá la transición.

Anexo 5: Capturas de pantalla pruebas de interfaz adaptable.

Las siguientes ilustraciones son el resultado de las pruebas realizadas a la página “Registro de Postulantes”.



*iPhone 6-8 portrait · width: 375px*

Ilustración 165 SE.iphone6\_8 portrait 375px



*iPhone 6-8 landscape · width: 667px*

Ilustración 166 SE.iphone6\_8 landscape 667px



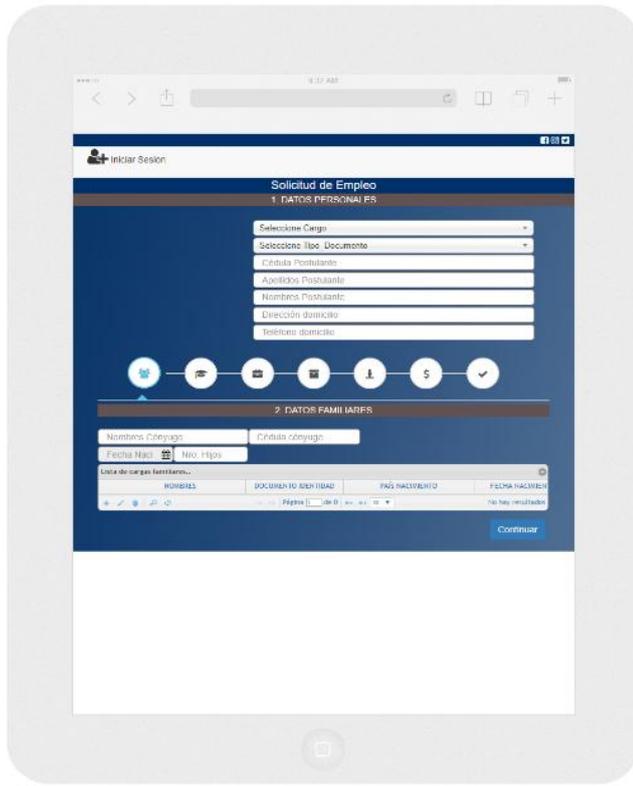
Android (Pixel 2) portrait · width: 412px

Ilustración 167 SE.android(pixel2)portrait 412px



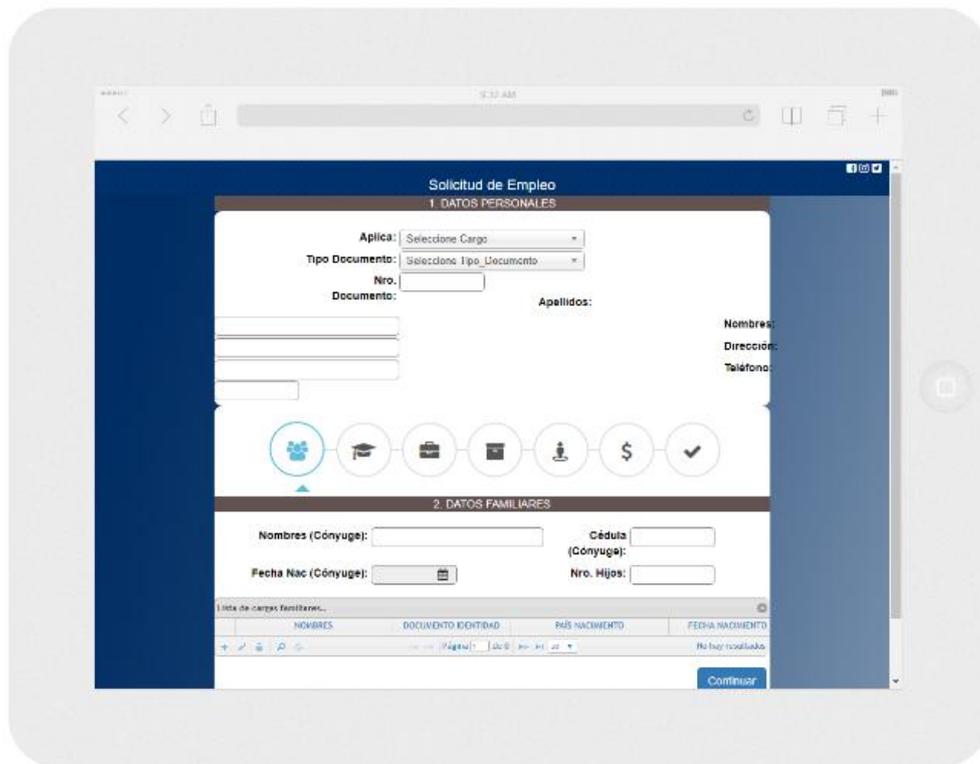
Android (Pixel 2) landscape · width: 684px

Ilustración 168 SE.android(pixel2)landscape 684px



iPad portrait - width: 768px

Ilustración 169 SE.ipad portrait 768px



iPad landscape - width: 1024px

Ilustración 170 SE.ipad landscape 1024px

Las siguientes ilustraciones son el resultado de las pruebas realizadas a la página “Registro de Cargos”.



*iPhone 6-8 portrait · width: 375px*

Ilustración 171 RC.iphone6\_8 portrait 375px



*iPhone 6-8 landscape · width: 667px*

Ilustración 172 RC.iphone6\_8 landscape 667px



Android (Pixel 2) portrait - width: 412px

Ilustración 173 RC.android(pixel2) portrait 412px



Android (Pixel 2) landscape - width: 684px

Ilustración 174 RC.android(pixel2) landscape 684px

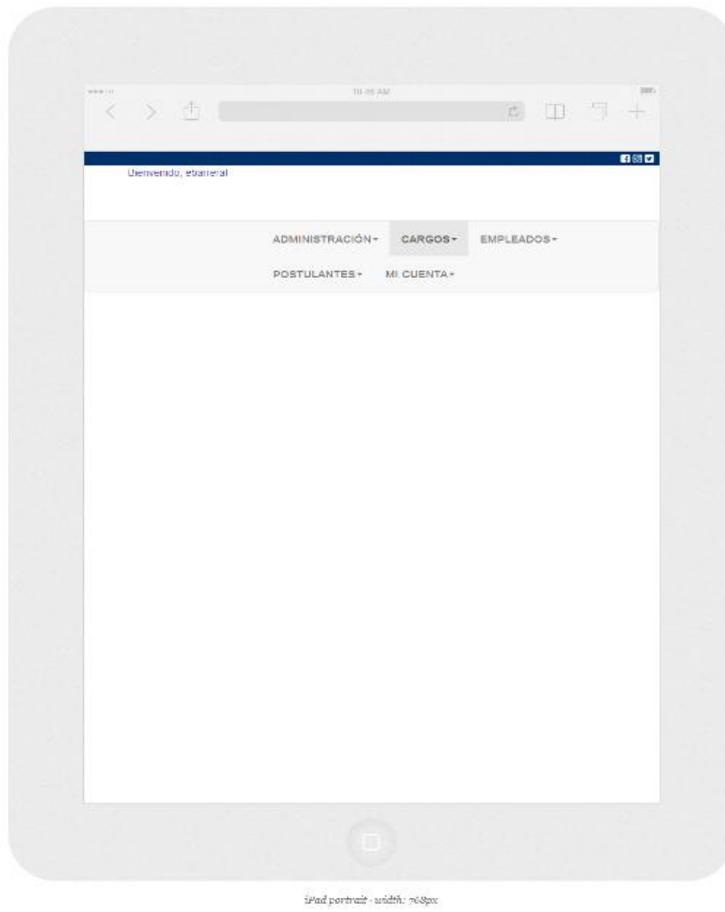


Ilustración 175 RC.ipad portrait 768px

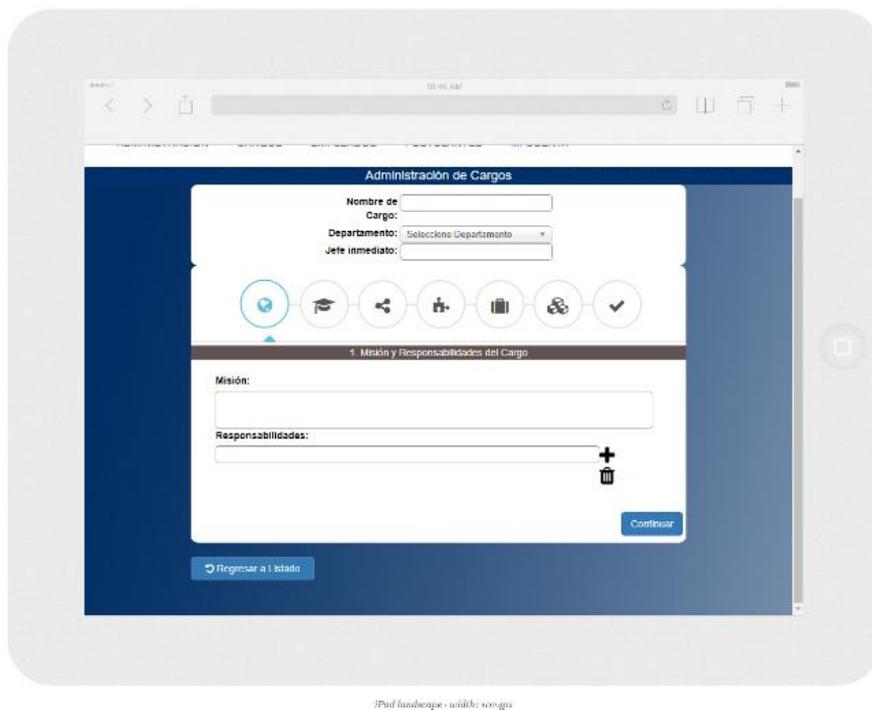


Ilustración 176 RC.ipad landscape 1024px

Las siguientes ilustraciones son el resultado de las pruebas realizadas a la página “Registro de Empleados”.



Ilustración 177 RE.iphone6\_8 portrait 375px



Ilustración 178 RE.iphone6\_8 landscape 667px



Android (Pixel 2) portrait - width: 412px

Ilustración 179 RE.android(pixel2) portrait 412px



Android (Pixel 2) landscape - width: 684px

Ilustración 180 RE.android(pixel2) landscape 684px

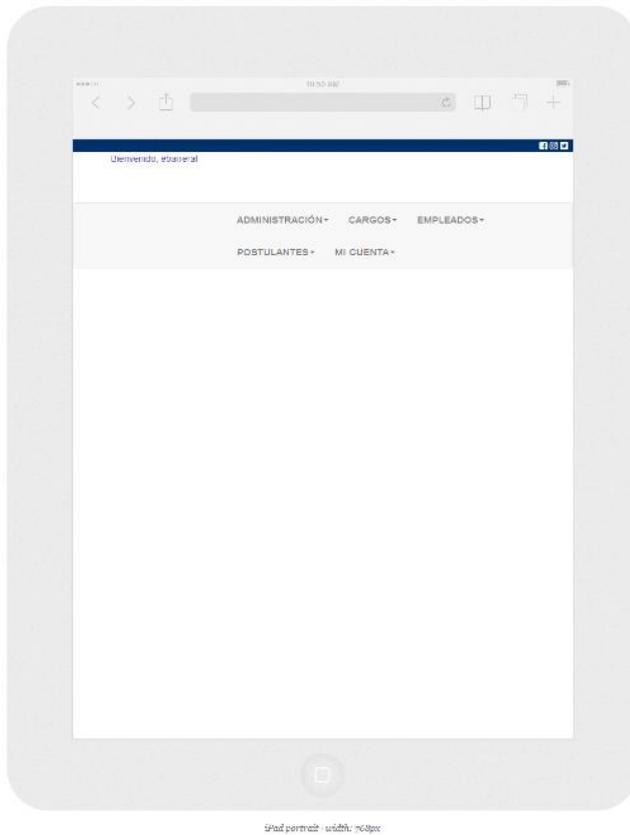


Ilustración 181 RE.ipad portrait 768px

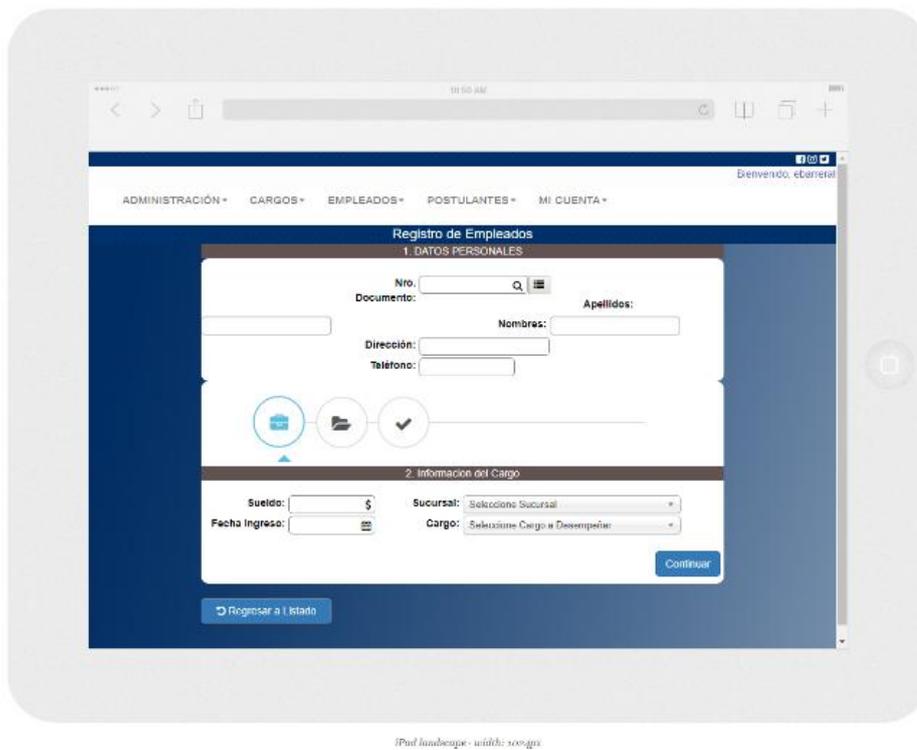


Ilustración 182 RE.ipad landscape 1024px

Las siguientes ilustraciones muestran las pruebas realizadas posteriores a la corrección de los puntos obtenidos de las pruebas anteriores.

Pantalla “Registro de Cargos”.

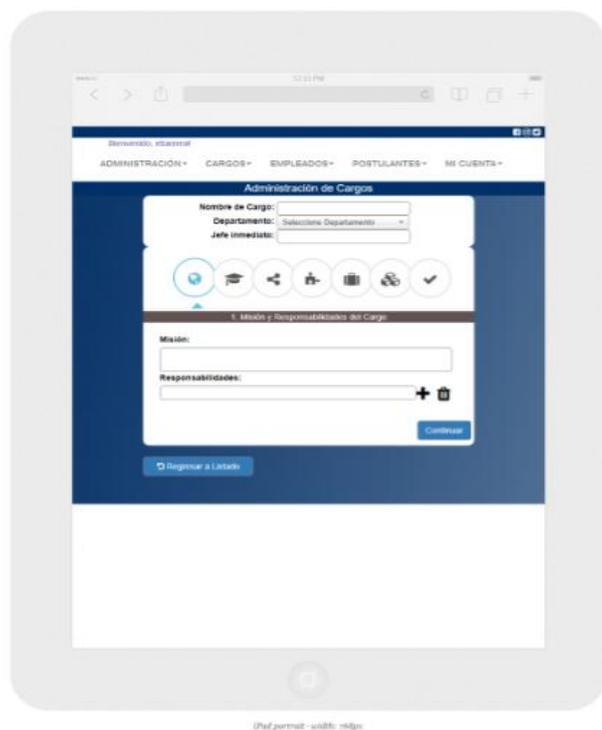


Ilustración 183 RC.Ipad portrait 768px

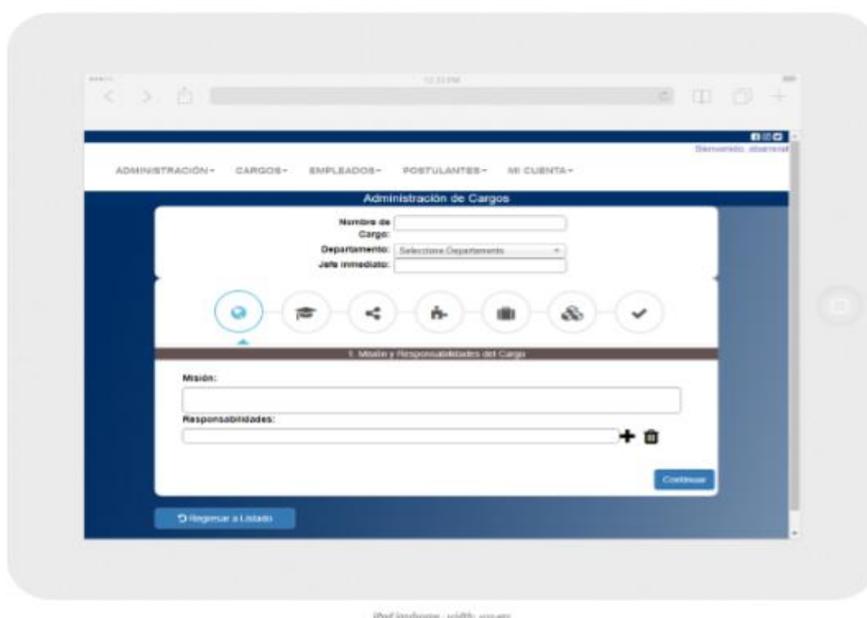


Ilustración 184 RC.Ipad landscape 1024px

Página “Registro de Empleados”.



iPhone 6-8 portrait - width: 375px

Ilustración 185 RE.Iphone6-8 portrait 375px



iPhone 6-8 landscape - width: 667px

Ilustración 186 RE.Iphone6-8 landscape 667px



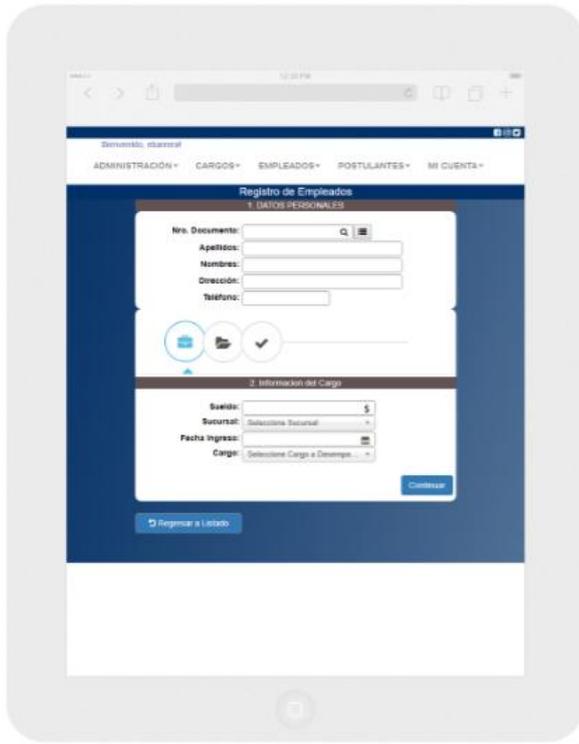
Android (Pixel 2) portrait - width: 412px

Ilustración 187 RE.android.pixel2.portrait 412px



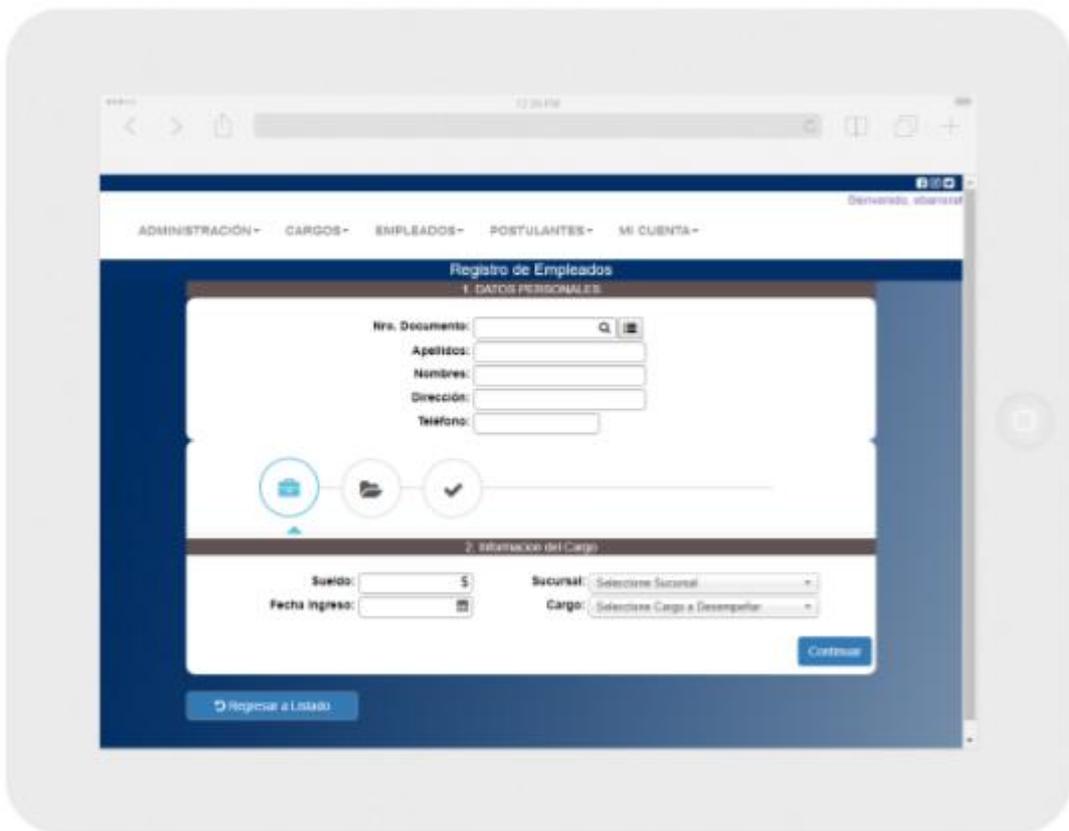
Android (Pixel 2) landscape - width: 684px

Ilustración 188 RE.android.pixel2.landscape 684px



iPad portrait - width: 768px

Ilustración 189 RE.ipad.portrait 768px



iPad landscape - width: 1024px

Ilustración 190 RE.ipad.landscape 1024px

Página “Registro Postulantes”.



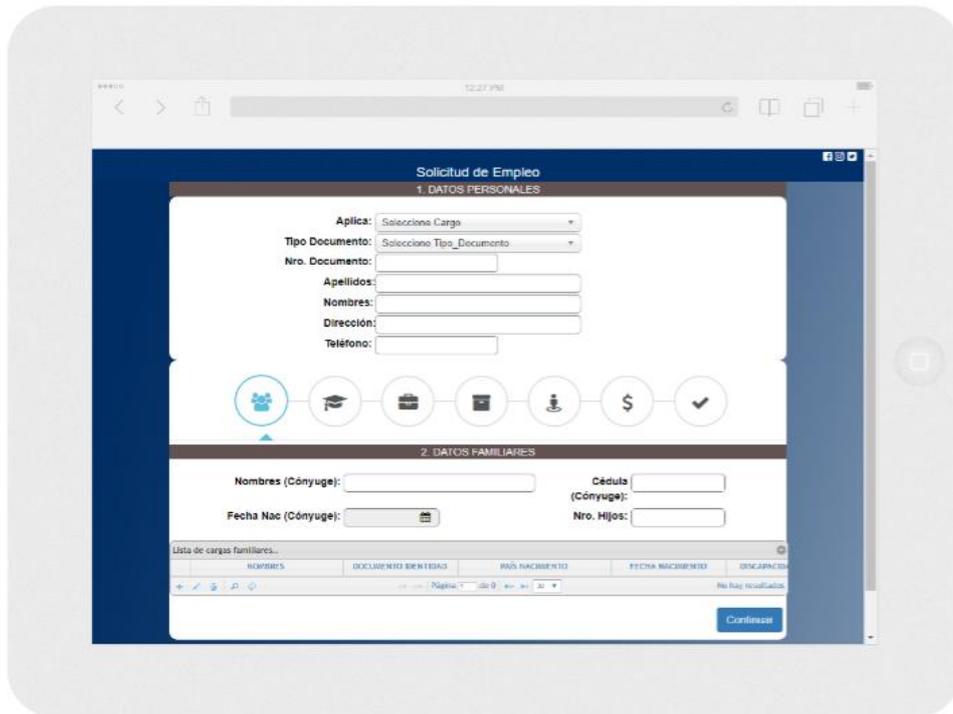
Android (Pixel 2) portrait - width: 412px

Ilustración 191 SE.android.pixel2 portrait 412px



Android (Pixel 2) landscape - width: 684px

Ilustración 192 SE.android.pixel2.landscape 684px



ipad landscape - width: 1024px

Ilustración 193 SE.ipad.landscape 1024px

## Anexo 6. Desarrollo Especificación de Requisitos de Software (ERS)

### 1. Introducción

Esta sección del documento es una especificación de los requisitos de software para el sistema gestión de selección de personal para el área de talento humano. La estructura de la especificación de requisitos está basada en las directrices propuestas en el estándar IEEE STD 830-1998

#### 1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo de un sistema de gestión web que permitirá manejar distintos procesos administrativos. Éste será utilizado por personal del área de talento humano y por usuarios en general.

#### 1.2. Ámbito del Sistema

El sistema de gestión de selección de personal tiene como objetivo ayudar a controlar el proceso de selección de los empleados de la empresa al igual que mantener una base de datos de los postulantes a los diferentes cargos de la empresa.

#### 1.3. Definiciones, Acrónimos

Usuario	Persona que usa el sistema.
ERS	Especificación de requisitos de software
RF	Requerimiento funcional.

Tabla 19 Definiciones y Acrónimos

#### 1.4. Referencias

Standard IEEE 830 – 1998

#### 1.5. Visión general del documento

Este documento está formado de tres secciones. La primera presenta una breve introducción sobre su contenido y los temas abordados. En la segunda sección se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en muchos detalles. La tercera sección del documento define detalladamente los requisitos que debe satisfacer el sistema.

## 2. Descripción General

### 2.1. Perspectiva del Producto

El sistema de gestión de selección de persona es un sistema independiente de otros sistemas que dispone la empresa, el cual trabaja bajo el entorno WEB lo que permitirá su utilización de forma rápida y eficaz.

El sistema debe cumplir con una excelente adaptación de su contenido en los diferentes dispositivos electrónicos como tabletas, celdulares, equipos de escritorio y equipos portátiles, proporcionando así una fácil utilización al usuario desde cualquier dispositivo.

### 2.2. Funciones del Producto

Las funciones del sistema se listan en la siguiente tabla.

1	Configuración del sistema	Realizar la configuración de parámetros necesario en el sistema.
2	Mantenimiento Nivel Instrucción	Realizar el ingreso, modificación y eliminación de los niveles de instrucción educativa.
3	Mantenimiento Procesos	Realizar el ingreso, modificación y eliminación de los procesos que maneja el sistema.
4	Mantenimiento Habilidades	Realizar el ingreso, modificación y eliminación de las habilidades o destrezas que son usadas en el sistema.
5	Mantenimiento Departamentos	Realizar el ingreso, modificación y eliminación de los departamentos que conforman la empresa.
6	Conf. Habilidades / Destrezas por formulario.	Realizar el ingreso y eliminación de las habilidades o destrezas que son evaluadas en los formularios.
7	Requisitos por proceso	Realizar el ingreso y eliminación de los requisitos necesarios para completar los procesos de la empresa.
8	Mantenimiento Cargos	Realizar el ingreso y modificación de los cargos que se manejan en la empresa.

9	Registro Postulantes	Realizar el registro de postulantes con toda la información personal, educativa y laboral a los diferentes cargos que maneja la empresa.
10	Confirmación Referencias	Completar un formulario de evaluación sobre las referencias laborales ingresadas por los postulantes.
11	Registro Empleado	Realizar el registro de empleados en los diferentes cargos que maneja la empresa, especificando su cargo, fecha ingreso y salario.
12	Folder Empleados	Realizar el ingreso de archivos digitales relacionados a los empleados que laboran en la empresa.
13	Historial Empleados	Realizar la consulta sobre el historial laboral en la empresa, por ejemplo cambio de sueldo, cambio de cargo, etc.

Tabla 20 Funciones del Producto

### 2.3. Características de los usuarios

• Administrador.	Usuario con conocimientos sólidos del giro de la empresa, por lo general jefe del departamento de talento humano.
• Personal Talento Humano.	Usuarios que laboran en el departamento de talento humano que se encargan de gestionar los diferentes procesos del departamento.
• Usuario sin registro.	Usuarios que no pertenecen a la empresa y que acceden al sistema desde la página web.

Tabla 21 Características Usuarios

## 2.4. Restricciones

<ul style="list-style-type: none"><li>• Conexión a internet.</li></ul>	El sistema es utilizado con una conexión a internet, si no se dispone de este servicio el sistema no está disponible.
<ul style="list-style-type: none"><li>• Tecnologías ASP.NET, HTML, JavaScript, CSS.</li></ul>	El desarrollo del sistema web se realiza obligatoriamente utilizando las tecnologías ASP.NET potenciadas con el uso de HTML, JavaScript y Hojas de estilo (CSS).
<ul style="list-style-type: none"><li>• Modelo Entity Framework.</li></ul>	La conexión con la base de datos es realizada con la tecnología ADO Entity Framework.

Tabla 22 Restricciones Sistema

## 2.5. Suposiciones y Dependencias

- Falta de disponibilidad de la empresa para analizar el departamento de talento humano.

## 3. Requisitos Específicos

### 3.1. Requisitos Funcionales

#### 3.1.1. Requisito funcional 1

**Registro de postulantes sin autenticación:** Los usuarios pueden acceder a la página web y registrarse como postulantes para uno de los cargos de la empresa sin necesidad de registrarse en el sistema o iniciar sesión.

#### 3.1.2. Requisito funcional 2

**Autenticación de usuarios:** Los usuarios pueden autenticarse o iniciar sesión para acceder a cualquier parte del sistema.

Las páginas a ser consultadas están disponibles de acuerdo a la ubicación del usuario en el sistema y a su nivel de accesibilidad o rol.

### 3.1.3. Requisito funcional 3

**Registro de usuarios:** Los usuarios pueden registrarse una sola vez en el sistema y poder acceder a las funciones de este. Para el registro de usuarios el sistema solicitará un nombre de usuario y contraseña.

### 3.1.4. Requisito funcional 4

**Consulta y Mantenimiento de Nivel de Instrucción:** El usuario con nivel de acceso adecuado puede consultar y dar mantenimiento a los niveles de instrucción que son usados en otras páginas del sistema.

### 3.1.5. Requisito funcional 5

**Consulta y Mantenimiento de procesos:** El usuario puede consultar y dar mantenimiento a los procesos administrativos que serán gestionado por el sistema.

### 3.1.6. Requisito funcional 6

**Consulta y Mantenimiento de habilidades y destrezas:** El usuario con nivel de acceso adecuado puede consultar y dar mantenimiento a la base de conocimientos de habilidades y destrezas que pudieran utilizar en otros módulos del sistema como por ejemplo al crear los cargos de la empresa.

### 3.1.7. Requisito funcional 7

**Consulta y Mantenimiento de Departamentos:** El usuario con nivel de acceso adecuado puede consultar y dar mantenimiento a los departamentos administrativos u operacionales que conforman la empresa. Los Departamentos serán utilizados por los usuarios que crean los cargos que el personal puede alcanzar.

### 3.1.8. Requisito funcional 8

**Consulta y Mantenimiento de habilidades y destrezas por formulario:** El usuario puede consultar y dar mantenimiento a las habilidades o destrezas que van a ser tomadas en cuenta al momento de calificar un formulario.

### 3.1.9. Requisito funcional 9

**Consulta y Mantenimiento de Requisitos por proceso:** El usuario puede consultar y dar mantenimiento a los requisitos necesarios para completar un proceso administrativo de la empresa procesos administrativo de la empresa. Los requisitos serán cargados de forma digital en el sistema.

### 3.1.10. Requisito funcional 10

**Mantenimiento de Cargos:** Los usuarios con nivel de accesibilidad adecuado podrán acceder a la página de mantenimiento de cargos en la cual podrán listar, ingresar y editar los cargos que dispone la empresa y a los cuales pueden postular los usuarios externos.

### 3.1.11. Requisito funcional 11

**Listado de Empleado.** El usuario con nivel de acceso adecuado podrá visualizar la información básica del empleado desde la página principal del mantenimiento de empleados.

### 3.1.12. Requisito funcional 12

**Registro de Empleados.** El usuario con rol adecuado puede realizar el registro de nuevos empleados a la empresa seleccionando de entre los postulantes registros para un cargo específico.

### 3.1.13. Requisito funcional 13

**Edición de datos de Empleado.** El usuario podrá modificar la información de los empleados, información referente al sueldo, cargo, sucursal en la que trabaja.

### 3.1.14. Requisito funcional 14

**Visualización del Folder del Empleado.** El usuario puede visualizar los archivos digitales cargados a la carpeta digital del empleado. Podrá abrir los archivos cargados y de ser necesario cambiar por un nuevo archivo en caso de ser un requisito para un proceso administrativo de la empresa.

### 3.1.15. Requisito funcional 15

**Agregar Nuevos archivos al Folder del Empleado.** El usuario puede cargar nuevos archivos digitales a la carpeta de un empleado específico. Los archivos pueden o no ser documentos requeridos por algún proceso de la empresa.

### 3.1.16. Requisito funcional 16

**Mantenimiento de postulantes:** Los usuarios con nivel de accesibilidad adecuado podrán ingresar a la página de mantenimiento de postulantes en la cual tendrán la opción de listar, ingresar y consultar la información de los postulantes registrados. El usuario también dispone de la opción de confirmación de referencias laborales misma que le permite visualizar y calificar el formulario sobre las referencias ingresadas por cada postulante.

### 3.1.17. Requisito funcional 17

**Cambio contraseña y cerrar sesión:** El usuario registrado y autenticado en el sistema tiene la opción de modificar la contraseña establecida para el ingreso al sistema. También dispone de la opción cerrar sesión misma que cerrará el sistema y elimina la información almacenada en la sesión del navegador.

## 3.2. Requisitos de Interfaz

### 3.2.1. Interfaz de usuario.

La interfaz de usuario consiste de un conjunto de páginas con botones, listas, campos de textos y grillas de mantenimiento. La interfaz se diseñó específicamente para el sistema propuesto y, será visualizada desde un navegador de internet desde cualquier dispositivo electrónico como tabletas, dispositivos móviles y equipos de escritorio.

### 3.2.2. Interfaces de hardware.

Para acceder al sistema es necesario un dispositivo electrónico con conexión estable de internet. Entre los dispositivos aceptados están tabletas, dispositivos móviles, laptops y equipos de escritorio.

### 3.2.3. Interfaces de software.

- Sistema operativo Windows 8 o superior. Soporte con últimas versiones de los navegadores de internet.
- Sistema operativo IOS 8 o superior.
- Sistema operativo Android 7 o superior.
- Explorador: Mozilla, Google Chrome, Zafari.

#### 3.2.4. Interfaces de comunicación.

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible.

#### 3.3. Requisitos de eficiencia

Garantizar que el diseño de las consultas u otro proceso no afecten el desempeño de la base de datos, provocando lapsos de tiempo considerables para cualquier transacción.

#### 3.4. Requisitos de Diseño

- Garantizar la correcta visualización de las páginas del sistema en cualquier dispositivo electrónico usando de manera correcta las clases adaptables (row, col-md, col-xs, col-lg) proporcionadas por el framework Bootstrap para el diseño adaptable.
- Garantizar que las funcionalidades del sistema no se pierdan al trabajar en cualquier dispositivo comprobando que las librerías jQuery y Bootstrap sean compatibles con los dispositivos a utilizar.
- Garantizar que la interfaz del sistema sea fácil e intuitiva para el usuario final.

#### 3.5. Atributos del Sistema

##### 3.5.1. Seguridad

- Garantizar la seguridad del sistema informático a los diferentes usuarios solicitando y verificando las credenciales otorgadas a cada uno para el ingreso al sistema.

- Garantizar la seguridad del sistema con respecto a la información y datos que se almacenen así como sus credenciales de acceso, utilizando ADO Entity Framework para la conexión con la base datos.

### 3.5.2. Mantenimiento

- Garantizar un herramienta que le ayude al usuario a navegar en el sistema y realizar los procesos adecuadamente a través de un manual de usuario que está disponible para su uso.



## ELABORACIÓN Y PRESENTACIÓN DE LA DENUNCIA/PROTOCOLO DE TRABAJO DE TITULACIÓN

### 1. Datos Generales

1.1. **Nombre del Estudiante:** Barrera Marquina Cristian Estéfano

○ Código: 38037

○ Contacto:

Teléfonos: 072875882 – 0984674126

Correo electrónico: distefano\_ba@hotmail.com

1.2. **Director Sugerido:** Mgst. Astudillo Catalina

○ Contacto:

Teléfono: 0987109923

Correo electrónico: cvastudillo@uazuay.edu.ec

1.3. **Co-director sugerido:**

1.4. **Asesor metodológico:**

1.5. **Tribunal designado:**

1.6. **Aprobación:**

1.7. **Línea de Investigación de la carrera:** Informática de computadores.

○ Código UNESCO: 1203

○ Tipo de Trabajo: Proyecto integrador.

1.8. **Área de estudio:**

1.9. **Título propuesto:** Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano, utilizando el framework Bootstrap.

### 2. Contenido

2.1. **Motivación de la investigación:** En la actualidad encontramos una evolución en las tecnologías de desarrollo de software orientadas a los dispositivos electrónicos. Tecnologías que son aplicadas en diferentes áreas de trabajo dentro de las empresas, que ayudan a simplificar procesos o mejorar su gestión. Dentro de estas tecnologías surge un framework llamado Bootstrap, que se caracteriza por ser una herramienta de software libre, y compatible con los sistemas operativos para dispositivos móviles que existen en el mercado como Windows Phone, IOS, Symbian, entre otros. Este framework facilita a los desarrolladores de software a crear sistemas web que se acoplen automáticamente a cualquier dispositivo en el cual sea abierto, sin la necesidad de desarrollar diferentes versiones del software con la única diferencia en su interfaz. Por tal motivo se ha decidido emplear este framework para el desarrollo de un sistema web destinado al área de talento humano, mismo que gestiona el

personal dentro del departamento, dando una mayor productividad en todas las tareas asignadas a los colaboradores del departamento.

- 2.2. **Problemática:** Temas que causan inquietud en el desarrollo de aplicaciones web para móviles y de escritorio, son las diferentes características propias de los navegadores y plataformas que existen; lo que obliga a desarrollar diferentes aplicaciones nativas para cada una de estas plataformas, además de las características específicas de cada dispositivo móvil o de escritorio.

Por otro lado, el área de talento humano de una empresa, generalmente se ha venido documentando a través de archivos físicos que se pueden estropear o incluso extravíar.

Por lo mencionado, se considera factible y necesario aprovechar los beneficios que otorga tanto los equipos tecnológicos de última generación y las aplicaciones de software, a fin de integrarlos y apoyar a la gestión y productividad de todos los ejecutivos que trabajan al interior del departamento de talento humano.

- 2.3. **Pregunta de investigación:** ¿Es posible desarrollar un sistema web que se adapte a cualquier dispositivo móvil o de escritorio?

- 2.4. **Resumen:** Este trabajo mejorará la técnica en el desarrollo de software, gracias a la nueva herramienta gratuita Bootstrap, que reduce el tiempo, dificultad y recursos al momento de crear software, gracias a que Bootstrap tiene diferentes elementos, con características que pueden adaptarse al dispositivo en el cual se ejecute el software. Esto elimina la necesidad de desarrollar un sistema diferente para cada sistema operativo móvil o de escritorio. Esta tecnología se verá aplicada en un sistema que formará parte del departamento de talento humano, apoyando a la efectividad en la ejecución de sus procesos, ya que por ahora es el menos relacionado con las nuevas tecnologías.

- 2.5. **Estado del Arte y marco teórico:**  
Bases Teóricas

HTML5.- (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 establece una nueva serie de nuevos elementos y atributos que reflejan el uso cotidiano de sitios web actuales; el HTML 5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante



XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). (Wempen, 2011)

jQuery.- Es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. (Foundation, 2016)

CSS3.- Es la última versión del lenguaje de las Hojas de Estilo en Cascada (Cascading Style Sheets), que trae muchas novedades esperadas con relación a su última versión 2.1, como las esquinas redondeadas, sombras, gradientes, animaciones y nuevos layouts. (Network, 2016)

Bootstrap.- Es un framework JavaScript con soporte de HTML y hojas de estilo (CSS3), que permiten a los desarrolladores realizar sus proyectos con mayor rapidez y sencillez. Este framework es diseñado para cualquiera y en donde sea, es decir, su principal característica es la de poder adaptar a cualquier dispositivo los proyectos creados en esta herramienta. (w3schools.com, 2016)

Características Bootstrap:

- Creado sobre JQuery con arquitectura de jQueryUI.
- Desarrollado para trabajar con HTML5.
- Precargado con una serie de temas que pueden ser utilizados.
- Preparado para dispositivos Smartphones & Tablets.
- Personalización de temas.

SQL Server 2014 express.- Es la versión gratuita del popular sistema de gestión de bases de datos de Microsoft. Es un sistema de administración de datos eficaz y confiable que ofrece un variado conjunto de características, protección de datos y rendimiento para clientes de aplicaciones incrustadas, aplicaciones web y almacenes de datos locales. Está disponible de forma gratuita y su redistribución con aplicaciones también es gratuita.

ASP.NET.- Es un Framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.



Talento Humano.- En la actualidad la gestión de los recursos humanos que incorporan una empresa, va más allá de la administración del personal (selección, contratación e inducción), se orienta a cambiar el paradigma de llamar a los empleados como colaboradores de la empresa, y así hacerlos parte de la organización y de los objetivos de la organización. Es por esto que la administración óptima es fundamental para que la productividad organizacional aumente día tras día. (Horizonte, 2016)

## 2.6. Hipótesis:

2.7. **Objetivo General:** Desarrollar un sistema web multiplataforma de gestión de selección del personal para el área de Talento humano, utilizando el framework Bootstrap.

## 2.8. Objetivos específicos:

- 1) Analizar el framework Bootstrap y tecnologías asociadas.
- 2) Realizar el levantamiento de información del departamento de talento humano.
- 3) Diseñar y desarrollar el sistema web para el departamento de talento humano.

2.9. **Metodología:** Para el desarrollo de este proyecto, se plantea una metodología que consta de lo siguiente:

**Tipo de Estudio.** El estudio de esta tesis se lo realizará de forma aplicada, de manera que lo principal y primordial es la adquisición de conocimientos de todo lo que se refiere a los lenguajes de programación, herramientas de desarrollo móvil y otros instrumentos útiles para la creación de aplicaciones móviles.

**Método.** Se utilizará para el desarrollo de esta tesis el método Analítico - Sintético ya que se estudiarán y analizarán varias herramientas necesarias para el desarrollo de esta aplicación web móvil.

**Medios.** Los medios para la recopilación de fuentes de información serán los siguientes:

- Internet
- Libros electrónicos
- Archivos publicados
- Análisis de ejemplos que podrán ser aplicados.

**Actividades a realizar:**

- Recolectar información sobre el framework Bootstrap.
- Levantar información del departamento de talento humano.
- Plantear la solución de software.
- Diseñar y desarrollar el software.

**2.10. Alcances y resultados esperados:**

- Documentar información sobre el framework Bootstrap.
- Documentar información del departamento de talento humano.
- Desarrollar la Base de Datos para la correcta gestión de la información.
- Desarrollar el sistema web en ASP.NET utilizando la herramienta Bootstrap.
- Validar la aplicación desarrollada en móviles de diferente índole.
- Elaborar un manual de la aplicación para proporcionar toda la información posible al usuario para la correcta utilización del mismo.

**2.11. Supuestos y riesgos:**

Falta de disponibilidad de la empresa para analizar el departamento de talento humano.

**2.12. Presupuestos:** En la siguiente tabla se detallan los valores que se van a financiar.

Rubro - Denominación	Costo (USD)	Justificación
Impresiones	150	Necesarias para presentar el documento final.
Conexión a internet	120	Necesario para la investigación.
Hosting y dominio	90	Necesario para poder publicar la página web en internet.
Imprevistos	100	Cualquier problema con algún equipo o servicio.
Transporte	100	Necesario para la movilidad del personal.
Libros	0	Serán obtenidos de bibliotecas. Necesarios para obtención de información para la investigación.
<b>Total</b>	<b>560</b>	

2.13. **Financiamiento:** Toda la financiación de los gastos del proyecto va por cuenta del participante del proyecto.

2.14. **Esquema tentativo:**

Introducción

Resumen

Objetivos

Capítulo 1: Tecnologías aplicadas

Introducción

1.1. Bootstrap 4.0

1.1.1. HTML5

1.1.2. JavaScript

1.1.3. Ajax

1.1.4. JQuery

1.1.5. CSS 3

1.1.6. Bootstrap 4.0

1.2 Base de Datos SQL Server 2014 Express

1.3 Asp.net

Capítulo 2: Análisis y Diseño del Sistema

2.1. Fundamentos sobre Recursos Humanos.

2.1.1. Introducción

2.1.2. Componentes

2.1.3. Objetivos.

2.2. Fundamentos sobre Talento Humano.

2.2.1. Componentes

2.2.2. Objetivos

2.3. Especificación de Requisitos de Software (ERS)

2.3.1. Diagrama de Casos de Uso

2.3.2. Descripción de Casos de Uso

2.4. Diseño de Datos.

2.4.1. Diagrama de Entidad - Relación

2.4.2. Diccionario de Datos

2.5. Diseño de Procesos.

2.5.1. Diagrama de Actividades.

2.6. Diseño Arquitectónico.

2.6.1. Diagrama de despliegue

2.6.2. Diseño de interfaces.

Capítulo 3: Codificación, pruebas y validación



3.1. Codificación del sistema.

3.2. Pruebas de funcionamiento

3.3. Validación del sistema.

Capítulo 4: Manual de Usuario

Conclusiones

Recomendaciones

Glosario

Bibliografía

Anexos

2.15. Cronograma:

Objetivo Específico	Actividad	Resultado esperado	Tiempo (semanas)
Analizar el framework Bootstrap y tecnologías asociadas.	1. Investigar la herramienta Bootstrap y todas las tecnologías asociadas.	<ul style="list-style-type: none"> <li>Documento que detalle las características y funciones de cada tecnología a utilizar.</li> </ul>	4
Realizar el levantamiento de información del departamento de talento humano.	1. Revisión de bibliografía existente sobre la gestión de talento humano.  2. Visita a empresa y reuniones con el personal del departamento de talento humano.	<ul style="list-style-type: none"> <li>Documento que detalle las funciones del departamento de talento humano dentro de las empresas.</li> <li>Especificación de Requisitos de Software.</li> </ul>	2
Diseñar y desarrollar el sistema web para el departamento de talento humano.	1. Diseño de Datos. 2. Codificación, pruebas y validación. 3. Documentación.	<ul style="list-style-type: none"> <li>Modelo de base de datos</li> <li>Aplicación web para el departamento de talento humano.</li> <li>Manual de usuario.</li> <li>Manual técnico.</li> </ul>	18



2.16. Referencias:

- ✓ **Kenneth E. Kendall, y Julie E. Kendall**, Análisis Y Diseño de Sistemas, Pearson Education, Mexico, 2011, Octava Edición.
- ✓ **Ian Sommerville**, Ingeniería Del Software, Pearson Education, Madrid, 2015, Novena Edición.
- ✓ **Javier Tuya, Isabel Ramos, Javier Dolado Cosin**, Técnicas cuantitativas para la gestión en la Ingeniería de Software, Netbiblo S.L, España, 2007, Primera edición.
- ✓ **Microsoft**, *Microsoft® SQL Server® 2014 Express*, Disponible en la Web:  
<http://www.microsoft.com/es-es/download/details.aspx?id=1695>, [Fecha Consulta: 07 de dic. de 2016]
- ✓ **Fundación Wikimedia, Inc**, *Especificación de requisitos de software*, Disponible en la Web:  
[http://es.wikipedia.org/wiki/Especificación\\_de\\_requisitos\\_de\\_software](http://es.wikipedia.org/wiki/Especificación_de_requisitos_de_software),  
Fecha Consulta: 12 de jun. de 2016]
- ✓ **Luis Castillo**, *Diseño de Software*, Disponible en la Web:  
<http://www.slideshare.net/lcastillo110/diseo-del-software-e-ingeniera-del-software>, [Fecha Consulta: 12 de jun. de 2016]
- ✓ **Freddy Ramirez**, *Etapas de Desarrollo de Software*, Disponible en la Web:  
<http://upsg01.foroactivo.com/t112-etapas-de-desarrollo-de-software> [Fecha Consulta: 12 de jun. de 16G]
- ✓ **Gloria Navas**, *La cultura de las aplicaciones móviles se impone en la sociedad*, Disponible en la Web: <http://www.tendencias21.net>, [Fecha Consulta: 12 de jun 2016]
- ✓ **Edelsys Hernández Meléndez**, *Metodología de la Investigación*, Disponible en la Web:  
[http://metodoinvestigacion.files.wordpress.com/2008/05/como\\_escribir\\_tesis\\_anah.pdf](http://metodoinvestigacion.files.wordpress.com/2008/05/como_escribir_tesis_anah.pdf) [Fecha Consulta: 05 de jul 2016]
- ✓ **Grupo Soluciones Horizonte**, *Importancia del talento humano como motor esencial en las organizaciones*, Disponible en la web:  
<http://www.gsh.com.co/importancia-del-talento-humano-como-motor-esencial-en-las-organizaciones/> [Fecha Consulta: 07 de dic 2016]



<http://www.gsh.com.co/importancia-del-talento-humano-como-motor-esencial-en-las-organizaciones/> [Fecha Consulta: 07 de dic 2016]

- ✓ Mozilla Foundation, CSS3, Disponible en la web:

<https://developer.mozilla.org/es/docs/Web/CSS/CSS3> [Fecha Consulta: 07 de dic 2016]

- ✓ Bootstrap, The world's most popular mobile-first and responsive front-end framework, Disponible en la web:

<http://getbootstrap.com/> [Fecha Consulta: 07 de dic 2016]

- ✓ W3schools, Bootstrap Tutorial, Disponible en la web:

<http://www.w3schools.com/bootstrap/> [Fecha Consulta: 07 de dic 2016]

- ✓ Gestiopolis, Administración de personal, Disponible en la web:

<http://www.gestiopolis.com/administracion-de-personal-definicion-objetivos-y-estructura/> [Fecha Consulta: 07 de dic 2016]

2.17. Anexos:

2.18. Firma de responsabilidad (Estudiante):

2.19. Firma de responsabilidad (Director sugerido):

2.20. Fecha de entrega:

13 de marzo de 2017

Cuenca, 15 de febrero de 2017

Señor

Ing. Oswaldo Merchán Manzano

DECANO FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

Su despacho.

De mi consideración:

Por medio de la presente me permito comunicar que he aceptado ser la directora de Tesis del estudiante: Barrera Marquina Cristian Estéfano con código 38037, egresado de la escuela de Ingeniería de Sistemas, cuyo tema es "Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano, utilizando el framework Bootstrap".

Por la atención que brinde a la presente, anticipo mis más sinceros agradecimientos.

Atentamente,



Mgst. Catalina Astudillo



UNIVERSIDAD DEL  
AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA DE LA FACULTAD DE  
CIENCIAS DE LA ADMINSTRACION DE LA UNIVERSIDAD DEL AZUAY.

**CERTIFICA:**

Que, el Señor Cristian Estefano Barrera Marquina registrada con el código 38037  
perteneciente a la escuela de Ingeniería de Sistemas, culminó su malla de estudios el día 11  
de Febrero de 2012. Finalizo su carrera el día 30 de Mayo de 2016.

Cuenca, Febrero 16 de 2017.

SECRETARIA  
A:  
FACULTAD DE  
ADMINISTRACION  
SECRETARIA

Derecho No. 113860

vcf.-

Cuenca, 16 de febrero de 2017



Señor

Ing. Oswaldo Merchán Manzano

DECANO FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

Su despacho.

Yo Cristian Estéfano Barrera Marquina, estudiante egresado de la escuela de Ingeniería de Sistemas con código 38037, solicito a usted de la manera más cordial se sirva autorizar el diseño de tesis presentado con el tema "Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano, utilizando el framework Bootstrap".

Por la atención que brinde a la presente, anticipo mis más sinceros agradecimientos.

Atentamente,



Estéfano Barrera

Cod. 38037

C.I. 0105348494

Oficio Nro. 023-2017-DIST-UDA

Cuenca, 24 de febrero de 2017

**Señor Ingeniero**  
**Oswaldo Merchán Manzano**  
**DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN**  
**Presente.-**

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 24 de febrero del 2017, recibió el proyecto de tesis titulado "Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano", presentado por Cristian Estéfano Barrera Marquina estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por la Ing. Catalina Astudillo, previo a la obtención del título de Ingeniero de Sistemas.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior a la Ing. Catalina Astudillo y como miembros del Tribunal al Ing. Luis Calderón e Ing. Juan Carlos Salgado.

Atentamente,



Ing. Marcos Orellana Cordero  
Cordinador Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay

FECHA: 01-03-2017

ESCUELA DE SISTEMAS Y TELEMÁTICA

ESTUDIANTE: CRISTIAN ESTÉFANO BARRERA MARQUINA

↳ Finaliza sus estudios 30/may/2016

↳ 9 meses egresado

Requisito Trabajo Titulación 



### CONVOCATORIA

Por disposición de la Junta Académica de la escuela de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: **“DESARROLLO DE UN SISTEMA WEB MULTIPLATAFORMA DE GESTION DE SELECCIÓN DEL PERSONAL PARA EL AREA DE TALENTO HUMANO”**, presentado por el estudiante Cristian Estéfano Barrera Marquina con código 38037, previa a la obtención del grado de Ingeniero de Sistemas, para el Lunes, 13 de marzo de 2017 a las 09h00.

Cuenca, 03 de marzo de 2017

Dra. Jenny Ríos Coello  
Secretaria de la Facultad

Ing. Catalina Astudillo Rodríguez

Ing. Luis Calderón Peralta OK.

Ing. Juan Carlos Salgado Arteaga OK.

Comunicado  
06-03-2017  
17h00.



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Cristian Estéfano Barrera Marquina

1.2 Código: 38037

1.3 Director sugerido: Ing. Catalina Astudillo Rodríguez

1.4 Codirector (opcional):

1.1 Título propuesto: ***“DESARROLLO DE UN SISTEMA WEB MULTIPLATAFORMA DE GESTION DE SELECCIÓN DEL PERSONAL PARA EL AREA DE TALENTO HUMANO”***

1.1 Revisores (tribunal): Ing. Luis Calderón Peralta e Ing. Juan Carlos Salgado Arteaga

1.5 Recomendaciones generales de la revisión:

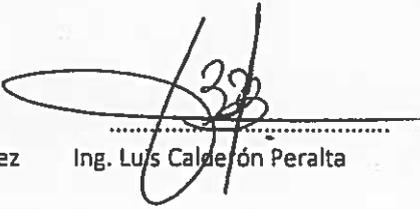
	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de Investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?				
Título Propuesto				
2. ¿Es informativo?				
3. ¿Es conciso?				
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?				
5. ¿Describe la teoría en la que se enmarca el trabajo				
6. ¿Describe los trabajos relacionados más relevantes?				
7. ¿Utiliza citas bibliográficas?				
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?				
9. ¿Tiene relevancia profesional y social?				
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?				
11. ¿Es factible de verificación?				
Objetivo general				
12. ¿Concuerda con el problema formulado?				
13. ¿Se encuentra redactado en tiempo verbal infinitivo?				

(\*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....  
.....  
.....

  
.....  
Ing. Catalina Astudillo Rodríguez

  
.....  
Ing. Luis Calderón Peralta

  
.....  
Ing. Juan Carlos Salgado Arteaga



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Cristian Estéfano Barrera Marquina

1.2 Código: 38037

1.3 Director sugerido: Ing. Catalina Astudillo Rodríguez

1.4 Codirector (opcional): \_\_\_\_\_

1.1 Tribunal: Ing. Luis Calderón Peralta e Ing. Juan Carlos Salgado Arteaga

1.2 Título propuesto: **“DESARROLLO DE UN SISTEMA WEB MULTIPLATAFORMA DE GESTIÓN DE SELECCIÓN DEL PERSONAL PARA EL ÁREA DE TALENTO HUMANO”**

1.3 Resolución:

1.3.1 Aceptado sin modificaciones

1.3.2 Aceptado con las siguientes modificaciones:

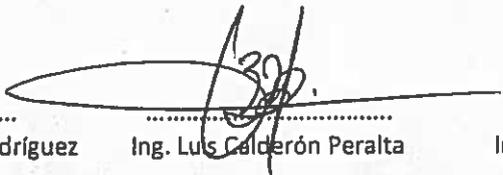
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

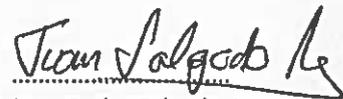
1.3.3 No aceptado  
• Justificación:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

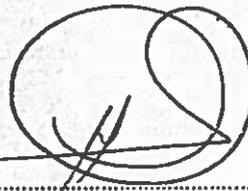
Tribunal

  
.....  
Ing. Catalina Astudillo Rodríguez

  
.....  
Ing. Luis Calderón Peralta

  
.....  
Ing. Juan Carlos Salgado Arteaga

  
.....  
Sr. Cristian Estéfano Barrera Marquina

  
.....  
Dra. Jenny Ríos Coello  
Secretaría de la Facultad

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

CERTIFICA:

Que, el Consejo de Facultad en sesión del 31 de marzo de 2017, conoció la petición del señor **CRISTIAN ESTÉFANO BARRERA MARQUINA** con código 38037 que presenta el diseño de su trabajo de titulación denominado. **DESARROLLO DE UN SISTEMA WEB MULTIPLATAFORMA DE GESTIÓN DE SELECCIÓN DE PERSONAL PARA EL ÁREA DE TALENTO HUMANO, UTILIZANDO EL FRAMEWORK BOOTSTRAP**", previa a la obtención del título de Ingeniero de Sistemas.- El Consejo de Facultad acogió el informe de la Junta Académica de Ingeniería de Sistemas y Telemática y resolvió aprobar el diseño. Designa como **Directora a la ingeniera Catalina Astudillo Rodríguez** y como miembros del tribunal examinador a los ingenieros Luis Calderón Peralta y Juan Carlos Salgado Arteaga.- En esta misma sesión el Consejo de Facultad fija como plazo para la entrega del trabajo de titulación, seis meses contados desde la fecha de su aprobación, esto es hasta el **30 DE SEPTIEMBRE DE 2017**, debiendo la Directora presentar a la Junta Académica, dos informes bimensuales del desarrollo del trabajo de titulación.

Cuenca, abril 3 de 2017



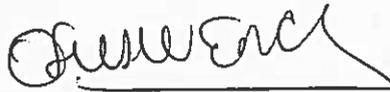
Dra. Jenny Ríos Coello  
Secretaria de la Facultad de  
Ciencias de la Administración



UNIVERSIDAD DEL AZUAY  
A.  
FACULTAD DE  
ADMINISTRACION  
SECRETARIA

Decano de la Facultad de Ciencias de la Administración, Cuenca, 20 de septiembre de 2017.- Con autorización amplia y suficiente concedida por el Consejo de Facultad en sesión del 25 de febrero de 2016, conoció la petición del estudiante **CRISTIAN ESTÉFANO BARRERA MARQUINA** con código 38037, quien solicita prórroga para la presentación del trabajo de titulación: **"DESARROLLO DE UN SISTEMA WEB MULTIPLATAFORMA DE GESTIÓN DE SELECCIÓN DE PERSONAL PARA EL ÁREA DE TALENTO HUMANO, UTILIZANDO EL FRAMEWOR BOOTSTRAP"**, previo a la obtención del título de Ingeniero de Sistemas y Telemática, cuyo plazo de presentación es hasta el 30 de septiembre de 2017, en apego al Reglamento de Régimen Académico y la normativa Institucional, *resuelve aprobar la solicitud y conceder una prórroga de seis meses, esto es hasta el 30 de marzo de 2018.*

*El estudiante Cristian Estéfano Barrera Maquina, con el período de prórroga, al haber finalizado la carrera el 30 de mayo de 2016, y haber transcurrido más de un año de finalización de estudios, conforme al Reglamento de Régimen Académico debe cumplir con la actualización de conocimientos*



**Ing. Oswaldo Merchán Manzano**  
**Decano de la Facultad de**  
**Ciencias de la Administración**

STAB IIIA  
13047