

Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Aplicaciones con Tecnología a Internet

**Seminario**

**Aplicación web configurable para la automatización de procesos de evaluación y selección de la Empresa**

**Seminario**

presentado ante la ilustre

Universidad Central de Venezuela

**por los bachilleres**

**Kevin D. Miranda F.**, C.I.: 26.150.260

**Luis E. Campos M.**, C.I.: 18.033.084

**Tutor**

**Prof. José Sánchez**

Caracas, Abril de 2017

**Notas para Kevin**

* **Te recomiendo que veas por youtube como crear el template de estilos para APA, así no nos toca configurar manualmente las líneas, etc, ya que estos estilos pueden cargarse desde una plantilla.**
* **hay maneras de generar automáticamente los índices, ergo, igual, averigua cómo se hace porque es útil saberlo, (son cosas principalmente de Word y no de google doc, pero podemos importar este documento desde aquí y colocarle todos los formatos en un word 2010 o algo asi, seria cuestion de tener el documento casi listo)**
* **si encuentras información en internet que nos pueda servir (las referencias que nos dio el profesor por ejemplo, guardalas en marcadores y copialas en las referencias de este documento) te recomiendo que lo hagas cada vez que copies un segmento de lo que sea desde otro sitio ya que después es complicado recordar las referencias**
* **cada vez que copies algo desde otro sitio colocar esta marca <Ref (numero)> asi podemos poner el hipervínculo al glosario / referencia que le corresponde, eso le da dinamismo al documento**
* **links, cosas que consigas, copialas en referencias**
* **si escribiste algo nuevo y quieres que lo revise, hazle una nota o subraya con un color cualquiera que se note para saber que tengo que prestarle atención a la redacción o algo.**
* **si te sientes inspirado, escribe, después nos encargamos (me encargo) de que tenga el formato appa**
* **anota cosas aqui tambien si vez algo importante que tener en cuenta**
* **cosas anotadas en azul, por lo general fueron observaciones hechas por el profesor**

**Índice General**

**Índice General (osea, esta página) 1**

**Índice de figuras 3**

**Índice de cuadros 4**

**I Introducción 1**

1 Introducción 2

**II Marco Teórico 4**

**2 Definición5**

**3 Metodología de gestión de proyectos5**

3.1. Metodología PMI (Project Management Institute)5

3.1.1. Gestión del alcance del proyecto5

3.1.2. Gestión del tiempo5

3.1.3. Gestión del costo del proyecto5

3.1.4. Gestión de la calidad5

3.1.2. Gestión de la comunicación5

**4 Contexto del problema8**

4.1. Caso de Estudio6

**III Propuesta de Trabajo Especial de Grado 8**

**8 Análisis y propuesta**9

8.1. Situación Actual9

8.2. Propuesta de TEG10

8.3. Justificación11

8.4. Objetivo general11

8.5. Objetivos específicos11

8.6. Tecnologías propuestas11

**Bibliografía 12**

**Índices de figuras**

(Por ahora estas están puestas de ejemplo, no hemos generado ningún gráfico o figura propio)

2.1. Prácticas XP6

3.1. Vista previa de la primera versión de Phoenix12

3.2. Vista previa del manejador de descargas de la primera versión12

3.3. Ventana de personalización de barras de herramientas13

4.1. Inclusión de código JavaScript en páginas Web28

**Índices de cuadros**

(ejemplos igual

)5.1. Manejadores de eventos en XUL45

6.1. Comparación de arquitecturas de servicios web48

6.2. Servicios MIME soportados por REST52

El presente documento sigue la siguiente estructura:

* **Parte I – Introducción:** Sección en la que se expone el problema, la falta de eficiencia de los métodos convencionales de reclutamiento y sus procesos. En resumen, la propuesta para solucionar dichos problemas.
* **Parte II – Marco Conceptual:** se exponen los aspectos teóricos relacionados con los procesos, metodologías, patrones y tecnologías a utilizar para la implementación de la aplicación web, con la finalidad de presentar tanto las ventajas y desventajas de cada propuesta.
* **Parte III – Propuesta de TEG:** se plantea la situación actual por lo que se propone una solución, justificando la solución planteado y puntualizando el objetivo general y los objetivos específicos de dicha solución

**Parte I. El Problema**

**Introducción:**

En la actualidad, el éxito de las organizaciones, depende en gran medida de la capacidades y aptitudes de sus recursos humanos, por ello la adecuada selección de aspirantes es esencial para el crecimiento de cualquier empresa, cada año se desperdician recursos humanos, tiempo y materiales en procesos obsoletos de selección, que son ineficientes en el manejo de aspirantes capacitados en búsqueda de oportunidades laborales, por ello que empresas pierden excelentes profesionales o se ven obligados a invertir en procesos convencionales a los que les resulta imposible de manejar el volumen creciente personas.

Esta necesidad de reclutar nuevos aspirantes en las empresas viene dado por diferentes causas como expansión de la empresa, generación de nuevos puestos de trabajo, rotación de personal, un requerimiento o simple necesidad de fuerza laboral, etc. En la mayoría de los casos, estos procesos de selección se realizan de forma presencial con el aspirante, donde se informa del puesto, necesidad o actividad, se comienza con la entrega de documentos, a veces de ambas partes, para la postulación de dicho aspirante. De esta forma, la empresa debe emplear tiempo en proveer a cada aspirante de la información pertinente y el aspirante debe dirigirse a la empresa a obtener dicha información y proveer la suya, posterior a ello, de acuerdo a las necesidades de la empresa, ciertos aspirantes serán seleccionados para continuar con otros procesos para su selección, donde comúnmente se harán evaluaciones, entrevistas para conocer las expectativas del aspirante, conocer su psiquis, etc.

Por estos motivos, se propone el desarrollo de una aplicación web, con una flexibilidad de configuración que permita generar y ofrecer información sobre las diferentes vacantes, que administre la información obtenida de los aspirantes y maneje cada etapa de la selección; de tal modo que genere métricas que la empresa pueda utilizar para tomar mejores decisiones sobre la selección, y a su vez, dar información a los aspirantes, reduciendo los tiempos y costos necesarios para ambos usuarios.

**Planteamiento del Problema:**

Para cualquier empresa está sobreentendida la necesidad de contratar nuevo personal constantemente, sobre todo si se tiene un crecimiento de cualquier magnitud, cualquier tarea sin atender podría ser crucial para el desarrollo y progreso de la empresa, es por eso que se tiene la necesidad no solo de contratar, sino de captar candidatos que llenen dichos roles de la manera más eficiente posible, es simple estadística, mientras más candidatos puedas procesar, más probabilidades se tienen de conseguir alguno que llene las expectativas o que sea ideal para ciertos puestos disponibles.

Pero captar una gran cantidad de candidatos implica una gran cantidad de gasto de recursos para todos los involucrados, sea desde la empresa o desde el punto de vista de los candidatos, cualquiera que sea el gasto tiempo, dinero o material.

De esta necesidad nace nuestra propuesta que pretende mitigar parte del desperdicio de recursos y a su vez ofrecer una herramienta que beneficie a los involucrados.

**Objetivo General**

Evaluar los focos de pérdidas para los actores tanto empresariales como personas empleables, estudiar posibles soluciones desde un punto de vista web, para proponer un sistema que permita mitigar parte del consumo innecesario de recursos, y a su vez, que sea factible técnica, económica y financieramente, para los involucrados en los procedimientos de captación de personal.

**Objetivos Específicos**

1. Recopilar datos sobre los métodos comunes de captación de personal.
2. Estudiar Propuestas que permitan ayudar y automatizar el reclutamiento.
3. Crear Diagramas que ilustran los datos básicos que se tienden a intercambiar durante la búsqueda de empleo y su peso a la hora de estudiar un candidato.
4. Evaluar datos y procedimientos que aporten beneficios y valor al sistema, así como a todos los involucrados durante el reclutamiento,
5. Desarrollar soluciones o prácticas que simplifiquen el proceso de reclutamiento.
6. Realizar procedimientos flexibles y configurables para que puedan adaptarse a cualquier empresa que requiera contratación de personal, sin importar su escala y complejidad.

**Justificación de la Investigación**

El objetivo prioritario no es documentar ni informar cuáles son los gastos que afronta una empresa durante los procesos de reclutamiento, sino proponer una solución eficiente y de mayor flexibilidad que las herramientas convencionales que se encuentran en la actualidad, quizás no sea una respuesta obvia, pero la idea es demostrar que estos procesos pueden ser optimizados de tal modo que consecuentemente se haga más fácil captar personal o en su defecto aumentar la calidad de los candidatos que son seleccionados, la idea en general es diseñar un sistema que permita tanto a la persona jurídica como natural, simplificar sus procedimientos de contratación o poder llenar vacantes y necesidades emergentes de un modo que no involucra sacrificios significativos para los involucrados.

Cabe resaltar que, La contratación de personal, es un sujeto complejo, existen conjuntos de prácticas que son de uso común, pero probablemente no sean las más eficientes o están lentamente quedando obsoletos, una contratación errónea acarrea gastos significativos, pérdidas y consecuencias que no son sencillas de manejar, como emocionales, legales y morales, es por ello que cualquier aporte a tomar decisiones mejores durante el proceso de reclutamiento puede resultar en una bendición.

El realizar este estudio y recopilar datos relacionados al reclutamiento va a permitir analizar patrones o prácticas comunes que serán de utilidad para el desarrollar un sistema eficiente para el beneficio de las personas que sean involucradas, beneficios como:

1. Al reclutador, facilitar el proceso de toma de decisión acerca de si llevar adelante o no la contratación de cierta persona, gracias a la información y requerimientos recopilados.
2. Analizar las fortalezas y debilidades de las estructuras de contratación actuales, sobre las cuales el presente proyecto se basa y de este modo proponer condiciones de mejora.
3. Determinar mejoras o medidas a tomar para que los procesos de contratación que se planteen sean lo más factible posible para los involucrados.
4. Reducir el nivel de incertidumbre al que está sometida la empresa que dispone de la vacante y también mantener informados a los interesados en el puesto.
5. Justificar el uso de sistema basándonos en los recursos que ahorra para sus usuarios .
6. Determinar la plataforma tecnológica, dinámicas de trabajo, y los pasos que el sistema usará para captación de información de sus usuarios.
7. Generar información que permitirán llevar a cabo la toma de decisiones sobre contrataciones con bases sólidas, producto de un estudio previo.
8. Adquirir conocimientos que le serán de utilidad a los futuros gerentes de recursos humanos, sobretodo para evitar escenarios comunes que conlleva al desperdicio de recursos durante procesos de captación de personal obsoletos.

**Alcances y Limitaciones**

**Alcance**

El proyecto tiene como alcance crear una aplicación web que permite a usuarios en búsqueda de oportunidades laborales y a empresas con vacantes, pasar por una serie de procedimientos que los ayuden a encontrar soluciones a sus necesidades.

Los aspectos puntuales que comprende la investigación están referidos a los estudios de los procedimientos que se aplican en ciertas prácticas de búsqueda laboral y cómo evitar sus mayores defectos, de tal modo que se puedan plantear procedimientos que aprovechen mejor los recursos disponibles.

Cabe destacar que se analizaran diferentes métodos comunes de reclutamiento, es imposible abarcar todas las prácticas ya que existen métodos tan específicos de ciertas empresas, o que usan combinaciones de técnicas tan ligeramente intercambiables que no vale la pena estudiarlas.

**Limitaciones:**

**falta de instrumentos para mediciones técnicas:**

El comportamiento humano varía mucho entre individuo e individuo, tomar conclusiones del estado emocional, nivel de frustración, y similares, es cuando poco, un reto nada trivial, por lo tanto, la mayoría de los datos serán netamente basados en la interacción de las personas con la aplicación y es por ello que estos datos tienden a ser aproximados.

**Características de las muestras:**

los instrumentos y metodos de recoleccion de datos sobre la aplicación y sus características, son por lo general, encuestas, comentarios, sugerencias, cuestionarios, etc. Estos dados a su vez por personas con diferentes estratos, como por ejemplo, docentes, obreros, jefes de familia, comerciantes, etc, esa muestra de sujetos producen una calidad de información a recolectar probablemente heterogénea, y que necesitará ser interpretada.

**Sesgo del Sujeto:**

las respuestas que se obtendrán de las encuestas o entrevistas, dependen del grado de conocimiento que tengan los sujetos, de sus opiniones y al ser estas preguntas muy puntuales, la información que obtendremos de ellas será muy discreta.

**Disposición por parte de las personas en brindar información o limitaciones de tiempo en sus ocupaciones laborales para responder dichos instrumentos.**

(En este orden deberíamos desarrollar los puntos del capítulo 1)

(intentare colocar algo que limite estos puntos, cuando consiga un buen ejemplo)

**Parte II**

**Marco Teórico**

Este segmento es una introducción a los conceptos, antecedentes y las bases conceptuales que sirven como base para el desarrollo del trabajo de investigación útil para elaborar en las una ideas que van naciendo de la investigación; por tal motivo, en este capítulo se trata en detalle el estado actual de aplicaciones relacionadas con la gestión de las personas como recursos humanos, conceptos sobre características de la aplicación propuesta y conceptos relacionados.

**Antecedentes**

En Venezuela y en otras partes del mundo se han realizado diversas aplicaciones web relacionadas con la gestión de contrataciones y filtrado de personas por capacidades profesionales. Por ejemplo, en Venezuela se cuenta con páginas web como Bumeran, compuTrabajo, Beebee, etc.

Que son plataformas que ayudan a sus usuarios a encontrar empleo, tienen bastante cosas en común, pero en general se destacan por tener procedimientos cerrados e invariantes, que no ofrecen flexibilidad y que pueden resultar tediosos por la cantidad requerida de información bajo cualquier rol que se utilicen esas plataformas.

**Características positivas y negativas de los antecedentes**

Estos portales en su mayoría, por su largo tiempo de existencia ya han segmentado un gran número de usuarios,poseen diseños atractivos y funcionales, así como también, recolectado una gran cantidad de información de miles de usuarios, poseen filtros, motores de búsqueda y clasificaciones por categorías, todos ellos, con el propósito de hacerle a los usuarios las búsquedas más simples, todos esas caracteristicas son inspiración y guía de cómo ayudar a los usuarios a conseguir lo que quieren, en este caso empleo.

Aunque, cabe destacar que, poseen algunas características que causan controversia, como lo sería la rigidez de sus estructuras, no permiten ninguna personalización, teniendo como característica común ser repositorios de curriculums y formularios de datos personales, y por la parte empresarial, poco más que un anuncio de periódico, no ofrecen herramientas como: pruebas, formularios hechos a la medida para el puesto de trabajo, preguntas cruciales, etc. eso justo es lo que queremos aportar con nuestra propuesta de trabajo de grado, hacer herramientas flexibles, que ayuden a seleccionar con mayor precisión los postulantes que desean una vacante y posean los requisitos necesarios para llenarla satisfactoriamente. ref.[1] (hice un txt con las ref aparte)

esta sección se llena justo después de explicar sus antecedentes.

Colocar de que se trata el capítulo 2, y luego empiezas a engranar, lo que dices en este párrafo, con el siguiente título que vas a decir, como por ejemplo el de la metodología de gestión de proyectos.

**Problema de investigación**

Facilitar la conexión entre los participantes y las instituciones que posean vacantes en roles que los usuarios puedan aprovechar.

• Velar por el uso adecuado de los recursos ordinarios y extraordinarios de los participantes de los concursos para las vacantes propuestas por las empresas.

• Mantener una base de datos de los recursos personales de muchos individuos, para poder así contactar aquellos que puedan exceder cumpliendo con los con las vacantes.

• Velar porque las actividades que se publiquen en la aplicación sean legítimas y cumplan con un mínimo de información para que los usuarios puedan tomar buenas decisiones para postularse.

• Estudiar y evaluar las proposiciones de cualquier institución que pueda aportar una significativa mejora a los procesos de contratación de personal, y hacer las mejoras correspondientes a los recursos y herramientas del sistema.

• Divulgar el potencial investigativo y creador de la Facultad de Ciencias, así como los resultados de investigaciones ya realizadas, y los aportes que la universidad puede hacer para solucionar problemas de reclutacion de personal.

**Arquitectura Cliente-Servidor**

(esto aqui, de cliente servidor hace falta refinar un poco)  
La arquitectura cliente-servidor es una arquitectura de procesamiento cooperativo entre computadores que está conformada por dos (2) componentes principales: clientes y servidores. Los clientes se encargan de iniciar la comunicación con el servidor a través de la red, emitiendo solicitudes de recursos o peticiones de servicios, como por ejemplo consultar una información o solicitar un archivo, y el servidor se encarga de atender las solicitudes realizadas por los clientes, respondiendo con el recurso o la información solicitada (Olivares, 2012).  
El servidor suele estar ubicado en un equipo con grandes capacidades de almacenamiento y procesamiento, para poder atender de forma eficiente las peticiones realizadas por los clientes. Existen distintos tipos de servidores como los servidores web, que almacenan principalmente documentos HTML (HyperText Markup Language, Lenguaje de Marcas de Hipertexto), los servidores de base de datos, que dan servicios de almacenamiento y gestión de bases de datos a sus clientes, los servidores del correo, que almacenan, envían, reciben y realizan todas las operaciones relacionadas con el correo electrónico de sus clientes, entre otros tipos de servidores.

En la **Figura 1:**

se puede observar la representación de un cliente-servidor:

Algunas de las características principales de esta arquitectura son:

• El cliente mantiene un papel activo en la comunicación, debido a que inicia la comunicación a través de solicitudes.  
• El cliente espera y recibe respuestas del servidor.  
• Un cliente puede comunicarse con más de un servidor de manera concurrente.  
• El cliente es el componente del sistema que interactúa de manera directa con el usuario final, a través de una interfaz gráfica.  
• El servidor permanece en espera de solicitudes, por lo tanto se considera que tiene un papel pasivo en la comunicación.  
• Un servidor puede recibir y mantener la conexión con diversos clientes al mismo tiempo, sin embargo el número máximo de conexiones posibles es limitado.

**Funcionamiento de una aplicación web**  
Las aplicaciones web se basan en la arquitectura cliente-servidor, explicada anteriormente, donde el cliente es un navegador web y el servidor es un servidor web que almacena las páginas de la aplicación.

El funcionamiento de una aplicación web es sencillo, el cliente emite una petición de una página web estática o dinámica al servidor web, y éste devuelve la página solicitada. Una página web estática, es aquélla que no cambia cuando un usuario la solicita: el servidor web envía la página al navegador web solicitante sin modificarla.

Por el contrario, una página web dinámica es resuelta y adaptada a los datos del cliente en específico, ya que tiene líneas de código no determinadas, y que son resueltas por el servidor según la información almacenada del usuario en la sesión actual; el servidor web puede transferir la página a un software especial, a una librería o a un módulo externo, generalmente llamado servidor de aplicaciones, que se encarga de leer el código de la página y generará en función de las instrucciones del código dinámico un contenido específico para el usuario.

El resultado tiende a ser una página estática (código HTML puro) que el servidor de aplicaciones devuelve al servidor web, que a su vez la envía al navegador solicitante, El servidor de aplicaciones también permite trabajar con recursos del lado del servidor, como las bases de datos. Sin embargo, el servidor de aplicaciones no puede comunicarse con la base de datos directamente, necesita de un controlador que actúa de intermediario para establecer la comunicación.

**En la Figura 2**. se puede observar el proceso para solicitar una página web estática al servidor web y en la

**Figura 2.1** el proceso para solicitar una página web dinámica al servidor de aplicaciones con datos extraídos de una base de datos.

Aunque los servidores web están principalmente diseñados para presentar contenido estático, hoy en día la mayoría de éstos tienen complementos para soportar lenguajes de scripting (guiones) como Perl, PHP, ASP ref[2,3,4], entre otros, pudiendo servir contenido dinámico sin necesidad de comunicarse con un servidor de aplicaciones. También, muchos de los servidores de aplicaciones de la actualidad son capaces de hacer lo que un servidor web es capaz de hacer, es decir, que pueden servir tanto contenido dinámico como estático.

**Ventajas**

Algunas de las ventajas de las aplicaciones web son:

• Las aplicaciones web pueden ser utilizadas por múltiples usuarios al mismo tiempo.

• No necesitan ser descargadas, instaladas y configuradas. Se acceden vía en línea y están listas para trabajar sin importar cuál es su configuración o su hardware.

• Una aplicación web se puede ejecutar en múltiples plataformas (hardware o sistema operativo), ya que sólo se necesita de un navegador web para cada plataforma.

• Todos los cambios de interfaz o de funcionalidad que se deseen hacer en la aplicación web, se realizan en el código que reside en el servidor web y no en cada uno de los clientes.

• Se evitan problemas de inconsistencia de actualizaciones, ya que no existen clientes con distintas versiones de la aplicación.

**2.4.3. Desventajas**

Algunas desventajas son:

• Los múltiples accesos realizados en forma simultánea pueden degradar el rendimiento de las aplicaciones, debido a la sobrecarga del servidor donde se encuentran alojadas.

• Como el navegador es la herramienta utilizada para ejecutar las aplicaciones web, éstas dependen de la configuración del mismo, para que puedan trabajar de forma óptima, ya sea en cuanto a las Cookies, JavaScript o archivos Flash.

• Si el servidor donde está almacenada la aplicación web y la información no tiene los niveles de seguridad necesarios para resguardar los datos, éstas pueden ser vulnerables a los ataques de agentes maliciosos.

**Patrón Modelo Vista Controlador**

El Modelo Vista Controlador (MVC), es un patrón de diseño muy utilizado para el desarrollo de aplicaciones web, el cual tiene como idea fundamental separar los componentes principales de las aplicaciones, como lo son el sistema de gestión de base dedatos, las interfaces de usuario y las tareas de manejar los eventos que se produzcan.

A continuación se explican las partes del patrón MVC:

• Modelo: es la capa donde se trabaja con los datos, por tanto contiene mecanismos para acceder a la información. Cabe mencionar que en esta capa es habitual utilizar un ORM (Object Relational Mapping, Mapeo Objeto Relacional), que permita trabajar con abstracción de bases de datos y persistencia en objetos, en vez de usar directamente sentencias SQL (Structured Query Language, Lenguaje de Sentencias Estructuradas), que suele depender del motor de base de datos con el que se esté trabajando.

Principales responsabilidades:

o Acceder a la capa de almacenamiento de datos. o Definir las reglas de negocio (lógica del negocio). Ejemplo: "Si la persona X

tiene el privilegio Y, no puede ver la información Z".

• Vista: es la interfaz de usuario que permite mostrar los datos del modelo e interactuar con ellos mediante acciones o eventos.

Principales responsabilidades:

o Recibir datos provenientes del modelo y mostrarlos al usuario. o Tener registro de su controlador asociado.

• Controlador: actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Principales responsabilidades:

o Recibir los eventos de entrada que se realizan en la vista, como un clic, pulsar un botón, entre otros. o Contener las reglas de gestión de eventos (lógica de la aplicación). Ejemplo: "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

**Flujo de control del patrón MVC**

Aunque se pueden encontrar diferentes implementaciones del MVC, el flujo de control más frecuente de este patrón es el siguiente:

**Figura 3:**

1. El usuario interactúa con la vista realizando alguna acción, por ejemplo, pulsar un botón.

2. El controlador recibe la notificación de la acción del usuario y gestiona el evento,normalmente a través de un gestor de eventos.

3. El controlador hace la llamada al modelo, donde le pide visualizar o actualizar los datos.

4. El modelo interactúa con la base de datos, ya sea en forma directa o con una capa de abstracción, y realiza la consulta o la actualización de los datos.

5. El modelo le envía al controlador la información obtenida o le notifica de los cambios realizados.

6. El controlador le manda la respuesta a la vista y ésta reacciona en función de la

respuesta que recibe (actualizar la interfaz, abrir un enlace, etc.)

**Ventajas del MVC**

El patrón MVC para el desarrollo de aplicaciones web presenta las siguientes ventajas:

• Disminuye el acoplamiento entre las partes de la aplicación, ya que las vistas se desacoplan del modelo y el modelo se desacopla de la forma en que se muestra e ingresa los datos.

• Aumenta la cohesión, ya que cada elemento del patrón está altamente especializado en su tarea.

• Facilita el desarrollo de la misma aplicación en distintos dispositivos o canales.

**Figura 4 - (Flujo de control del patrón MVC)**

• Permite una mayor claridad de diseño.

• Facilita el mantenimiento.

• Aumenta la escalabilidad.

**Tecnologías del lado del cliente**  
Las tecnologías del lado del cliente son ejecutadas e interpretadas del lado del cliente en una aplicación web, es decir, en el navegador web. En muchos casos el buen funcionamiento de estas tecnologías va a depender del tipo de navegador web y de la versión de cada uno de ellos.  
Por lo general estas tecnologías son utilizadas para mostrar información, darle formato a dicha información, solicitar datos, etc. Entre las tecnologías que se explican a continuación están: el Lenguaje de Marcas de Hipertexto, Hojas de Estilo en Cascada, JavaScript y jQuery.

**Lenguaje de Marcas de Hipertexto**  
El Lenguaje de Marcas de Hipertexto, mejor conocido como HTML (HiperText Markup Language), “es el lenguaje con el que se escriben las páginas web, haciendo uso de marcas (etiquetas) para describir la forma en la que deberían aparecer los textos y los gráficos en un navegador web” (EcuRed, 2012).

HTML fue diseñado por el físico Tim Berners Lee en 1989, se publicó en un documento formal en 1991 y se convirtió en un estándar para el desarrollo de páginas web en 1995, ya que ofrece una estructura básica para la definición del contenido que por lo general se encuentra en una página web, como imágenes, texto, entre otros.

A partir de 1996, la World Wide Web Consortium (W3C), que es la organización que define la mayoría de los estándares para el desarrollo web, se ha encargado de publicar constantes revisiones (HTML 3.2, HTML 4, HTML 4.01) de este estándar, ampliándolo y depurándolo con cada versión. HTML 5 fue la última revisión aprobada por esta organización, en el año 2014.

HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Incorpora nuevas etiquetas con mayor significado semántico, mejora la integración con las tecnologías de desarrollo web actual, permite desplegar objetos y animaciones en formato 3D (tridimensional), tiene un mejor soporte multimedia y ahorra la necesidad de tener que procesar medios de audio o video a través de complementos instalados en el navegador web.  
  
**En la Figura 5:** se muestran las principales diferencias entre las etiquetas de HTML 4.01 y HTML 5

donde se puede observar que con HTML 5 se disminuye el uso de la etiqueta <div> (utilizada para definir un bloque de contenido), y se introducen etiquetas con mayor significado semántico para que la estructura de una página web sea más fácil de entender.

**Hojas de Estilo en Cascada**  
Mientras que HTML permite definir la estructura una página web, las Hojas de Estilo en Cascada (Cascading Style Sheets - CSS) son las que ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, tabletas, celulares u otros dispositivos capaces de mostrar contenidos web (De Luca Damián, 2010).

CSS fue diseñado por Håkon Wium Lie y Bert Bos entre 1994 y 1995 y se convirtió en un estándar en 1996, con el nombre de CSS 1. Al igual que con HTML, la W3C se ha encargado de publicar constantes revisiones de este lenguaje, siendo la versión CSS 2.1 el estándar actual que ofrece gran compatibilidad con la mayoría de los navegadores web del mercado.

En el año 2005 se comenzó a definir el sucesor de la versión CSS 2.1, al cual se lo conoce como CSS 3, pero actualmente sigue en desarrollo y sólo se han publicado algunos borradores. Esta versión a pesar de que aún no es compatible con todos los navegadores, ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual, como funciones avanzadas de movimiento y transformación, opciones de bordes redondeados, sombras en textos y cajas, entre otras opciones, sin la necesidad de usar un editor gráfico (De Luca Damián, 2010).

La sintaxis del lenguaje CSS consta de elementos llamados reglas, las cuales se dividen en selectores y declaraciones, donde el selector es el identificador al que se le quiere dar la declaración de un conjunto de propiedades con sus valores, como se puede ver en la

**Figura 6.**

**Figura 7.**

muestra un ejemplo de código CSS incrustado en HTML, para cambiar el tipo, el tamaño y el color de la fuente de un título y un párrafo.

**JavaScript**  
JavaScript (JS) es un lenguaje de scripting, interpretado, diseñado para la manipulación de eventos y objetos contenidos en una página web, de forma local y sin necesidad de transmisión alguna por la red.

JavaScript permite crear pequeños programas llamados script, que pueden acompañar un documento HTML o estar contenidos en su interior. Dicho programa se ejecuta en el navegador web cuando se carga el documento, o cuando se produce algún evento, como por ejemplo la activación de un enlace por parte del usuario (España, 2003).

Gracias a los scripts se pueden realizar tareas como:

• Validación de datos introducidos en un formulario.  
• Mensajes de alerta.  
• Algunos cálculos simples.  
• Control de eventos que se producen en la página: un elemento toma el foco, un elemento es activado con el ratón, etc.

La ventaja de poder de realizar tareas de este tipo en el navegador web, es que por ejemplo si el usuario no rellena correctamente un formulario, no necesita esperar mucho tiempo hasta que el servidor vuelva a mostrar el formulario indicando los errores existentes.

Para insertar código JavaScript en un documento HTML se utiliza la etiqueta <script> de la siguiente manera:

***<script language="JavaScript">  
 <!--<instrucciones> -->***

***</script>***  
Actualmente existe una gran cantidad de bibliotecas basadas en JavaScript, siendo jQuery una de las más utilizadas, ya que ésta posee una gran cantidad de funciones para simplificar la manera de interactuar con los documentos HTML, manipular documentos a través del árbol DOM (Document Object Model, Modelo de Objetos del Documento), gestionar eventos, animar y agregar interacción a las páginas web con la técnica AJAX (Asynchronous JavaScript And XM, JavaScript Asíncrono y XML).

La técnica AJAX tiene como objetivo crear páginas web dinámicas, mediante la realización de solicitudes asíncronas hacia el servidor, es decir sin la participación directa del usuario, las cuales permiten actualizar ciertas secciones de la página sin necesidad de recargarla por completo. En ambientes donde no se implementa esta técnica, cualquier actualización no será visible hasta tanto se recargue por completo el contenido de la página (W3Schools, 2015).

**Bootstrap**

Bootstrap, es un marco de trabajo (framework) que permite crear interfaces web con las tecnologías de CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una computadora de escritorio o portátil, una tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como diseño adaptativo (responsive design) (Solis, 2014).

**Tecnologías del lado del servidor**

Las tecnologías del lado del servidor permiten construir la lógica de negocio de la aplicación web. Esto implica el procesamiento de las solicitudes que se realizan desde el navegador web así como también la comunicación con el sistema encargado del manejo de datos.

Entre las tecnologías del lado del servidor que se explican a continuación están: el lenguaje de programación Ruby, el marco de trabajo Ruby on Rails, el sistema manejador de base de datos PostgreSQL y el Protocolo Ligero/Simplificado de Acceso a Directorios.

**Ruby**

Ruby es un lenguaje interpretado orientado a objetos, de propósito general y multiplataforma. Una de las principales filosofía del lenguaje es enfocarse en la simplicidad y la productividad, por lo que su sintaxis es muy sencilla de aprender, pero a su vez es muy poderosa y versátil para crear código escalable, sencillo y de manera eficiente (Graterol et al., 2014).

Ruby se utiliza para resolver problemas de cualquier índole en el área de la computación, aunque comenzó a hacerse más conocido por su uso para desarrollo web gracias al framework Ruby on Rails, mas no es el único uso que le podemos dar a este lenguaje (Graterol et al., 2014).

Algunas de las características generales del lenguaje son las siguientes (Ruby Lang Org, 2015):

• Licencia de código abierto: el código fuente de Ruby es libre de ser descargado e inspeccionado por sus usuarios.

• Multiparadigma: soporta más de un paradigma de programación, desde la funcional, hasta la imperativa, debido a los esfuerzos de su creador por crear un lenguaje que fuese un híbrido de todos sus lenguajes favoritos.

• Todo es visto como un objeto: no existen los tipos de datos elementales, sino que todas las operaciones sobre datos, son en realidad invocaciones a métodos pertenecientes a una clase.

• Altamente flexible: permite redefinir su estructura como el programador lo desee. Incluso las clases propias pueden ser redefinidas a conveniencia para modificar el comportamiento de las operaciones.

• Extensible: se le pueden agregar un gran número de funcionalidades nuevas, a través de su sistema de Gemas, que son bibliotecas desarrolladas por la comunidad y distribuidas libremente a través del repositorio oficial conocido como RubyGems.

• Múltiples hilos: el lenguaje implementa en su ejecución el paradigma de múltiples hilos, sin importar la plataforma sobre la cual se encuentre trabajando.

**Ventajas**

Algunas de las ventajas de Ruby son:

• Código abierto.

• Curva de aprendizaje muy corta.

• Sintaxis concisa y legible.

• Buenas prácticas y lineamientos de calidad reforzados por el lenguaje.

• Aumenta la productividad.

• Soporte extensivo a herramientas de todo tipo incluido por defecto en la biblioteca estándar.

• Dispone de una cantidad enorme de paquetes de terceros, disponibles a través del índice de paquetes de Ruby (llamados Gemas).

• Reduce dramáticamente el tiempo de desarrollo gracias a la disponibilidad de tantos paquetes.

**Desventajas**

Algunas de las desventajas de Ruby son:

• Comunidad pequeña en comparación con otros lenguajes.

• Demanda baja en el mercado laboral.

• Ruby necesita del manejo de tareas asíncronas no nativas para lograr un mejor rendimiento y escalado de aplicaciones donde haya mucha concurrencia.

• El uso de memoria de Ruby no está del todo optimizado.

**Ruby on Rails**

Ruby, al igual que otros lenguajes de programación, cuenta con varios frameworks que facilitan la creación de aplicaciones web. No obstante, el framework más utilizado por los desarrolladores, gracias a sus bondades y facilidad de uso, es Ruby on Rails.

Ruby on Rails, también conocido como Rails o RoR, “es un entorno de desarrollo web para Ruby de código abierto que está optimizado para la satisfacción de los programadores y para la productividad sostenible. Permite escribir un buen código evitando la repetición y favoreciendo la convención antes que la configuración” (Rails Guides, 2015).

Se basa principalmente en dos (2) filosofías de trabajo (Rails Guides, 2015):

• Don’t Repeat Yourself (DRY): en español No Te Repitas, sugiere que escribir el mismo código una y otra vez en cualquier componente de software resulta redundante y es una pérdida de tiempo que se puede evitar, por lo que se busca reutilizar componentes ya existentes.

• Convención sobre configuración: se basa en establecer estándares o convenciones reconocidas a nivel mundial como configuración por defecto para las tecnologías de trabajo, reduciendo así la cantidad de modificaciones que los desarrolladores deben hacer sobre las mismas para hacer funcional la aplicación.

Además, Rails está orientado al paradigma Modelo Vista Controlador (MVC), por lo que utiliza distintas tecnologías para dar soporte a cada una de estas partes. A continuación, se explica cada parte por separado para conocer las tecnologías asociadas a cada una.

• Modelo: consiste en las clases persistentes que representan a las tablas de la base de datos. En Rails la gema encargada de realizar automáticamente la correspondencia entre los modelos definidos por los desarrolladores y las tablas almacenadas en la base de datos es ActiveRecord. Una gran bondad de ActiveRecord es que permite realizar migraciones sin mucho esfuerzo, ya que los desarrolladores no tienen que preocuparse por crear manualmente la base de datos y sus tablas utilizando el lenguaje propio del manejador asociado, sino que pueden realizar creación, modificación y eliminación de tablas en la base de datos a través de comandos propios de Rails y efectuando las modificaciones pertinentes únicamente sobre los modelos (Rails Guides, 2015).

• Vista: es la responsable de generar la interfaz de usuario, normalmente basada en los datos del modelo. Existen varias formas de gestionar las vistas, el método que aplica Rails es usar Ruby embebido (archivos.html.erb), que es código HTML mezclado con código Ruby. La gema encargada de controlar las vistas en Rails es el ActiveView, la cual es responsable del despliegue de visualización de contenido en pantalla (Rails Guides, 2015).

• Controlador: es un intermediario entre los modelos y las vistas. Se encarga de recibir las peticiones que se generan de lado del cliente, y procesarlas de acuerdo a la lógica programada por los desarrolladores para generar las vistas asociadas a los resultados. En Rails, este módulo se conoce como ActionPack, y contiene la clase ApplicationController, encargada principal de todos los controladores del sistema (Rails Guides, 2015).

**Sistemas Manejadores de Bases de Datos**

Según Silberschatz A. et al. (2002) un Sistema Manejador de Base de Datos (SMBD) “consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos” y su objetivo principal “es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente”.

En la actualidad existe una gran variedad de sistemas manejadores de bases de datos, muchos de ellos son propietarios o comerciales como Oracle, Microsoft SQL Server, IBM Informix, Sybase, etc. y otros no comerciales o de código abierto como MySQL o PostgreSQL, siendo éste último el que se explica a continuación.

**PostgreSQL**

PostgreSQL es un potente Sistema Manejador de Base de Datos objeto-relacional, de código abierto, que cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección (The PostgreSQL Global Development Group, 1996-2015).

Trabaja muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema, por lo que se considera una herramienta que favorece a los usuarios con sistemas empresariales de gran tamaño.

PostgreSQL se ha enfocado tradicionalmente en la fiabilidad, integridad de datos y características integradas enfocadas al desarrollador. Tiene un planificador de consultas extremadamente sofisticado, que es capaz de unir cantidades relativamente grandes de tablas eficientemente (2ndQuadrant Ltd, 2001-2015).

Se distribuye bajo la Licencia PostgreSQL, que es una licencia similar a la de la Distribución de Software de Berkeley (BSD) y a la del Instituto de Tecnología de Massachusetts (MIT), que permite a los usuarios hacer cualquier cosa que quieran con el código, incluyendo la reventa de los binarios sin el código (2ndQuadrant Ltd, 2001-2015).

Algunas de las ventajas de PostgreSQL son:

• Es código abierto.

• La velocidad de respuesta se mantiene al aumentar el tamaño de la base de datos, cosa que no sucede con otros programas que se suelen poner lentos.

• Proporciona estabilidad y confiabilidad.

• Tiene una gran capacidad de almacenamiento.

• Soporta gran número de peticiones simultáneas a la base de datos de forma correcta.

• Puede operar sobre distintas plataformas, incluyendo Linux, Windows, Unix, Solaris y MacOS X.

• Provee un buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.

• Soporta los tipos de datos, cláusulas, funciones y comandos de tipo estándar SQL92/SQL99 y extendidos propios de PostgreSQL.

Algunas de las desventajas son:

• En comparación con otros sistemas manejadores de base de datos, como por ejemplo MySQL, es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección.

• Cuenta con muchos foros oficiales de ayuda, pero no con una documentación de ayuda obligatoria.

• La sintaxis de algunos comandos o sentencias no es tan intuitiva.

**Directorio y servicio de directorio**

**Directorio**

Un directorio es un repositorio único para la información relativa a los usuarios y recursos de una organización, el cual está organizado de manera lógica y jerárquica. Por ejemplo, en un directorio se puede almacenar información acerca de las cuentas de usuario de las personas de una organización (nombres, contraseñas, números de teléfono, etc.) (EcuRed, 2012).

Generalmente se describe un directorio como una base de datos, pero es una base de datos especializada cuyas características la apartan de una base de datos relacional de propósito general (Calzada, 2001).

Algunas de las características que diferencian un directorio de una base de datos relacional son (Calzada, 2001):

• Los directorios están optimizados para accesos en lectura, frente a las bases de datos convencionales, que se encuentran optimizadas para lectura y escritura.

• Los directorios están optimizados para almacenar información relativamente estática, por lo que no son recomendables para almacenar datos que cambian con frecuencia.

• Los directorios no soportan transacciones. Las transacciones son operaciones de base de datos que permiten controlar la ejecución de una operación compleja, de modo que dicha operación se completa totalmente o no se ejecuta en absoluto. Las bases de datos convencionales implementan esta funcionalidad, a costa de hacer su implementación más compleja. Pero el tipo de información que se almacena generalmente en el directorio no requiere una consistencia estricta y se considera aceptable que el número de teléfono de una persona no esté actualizado de forma temporal.

• El diseño del directorio es bastante diferente del diseño de una base de datos relacional. En las bases de datos se tiende a diseñar un modelo de datos para asuntos de negocios y los requisitos de los procesos, el cliente, el servicio y el administrador. Con los directorios, lo que se hace es colocar la información en un repositorio común para muchos usos y usuarios. Su diseño y esquema de la información deben ser desarrollados conforme a lo que está representando, a objetos en la vida real. En la mayoría de los casos, estos objetos representan los usuarios, agendas, listas, preferencias, derechos, productos y servicios, dispositivos, perfiles, políticas, números de teléfono, rutas, etc. Además, se debe considerar también los aspectos operacionales de diseño, en vista del funcionamiento y de escala.

**Servicio de directorio**

Un servicio de directorio es una aplicación o un conjunto de aplicaciones que proporciona métodos para almacenar los datos de un directorio y a su vez ponerlos a disposición de los administradores y los usuarios que se encuentren en la misma red (EcuRed, 2012).

Un servicio de directorio está altamente optimizado para lecturas y proporciona alternativas avanzadas de búsqueda en los diferentes atributos que se puedan asociar a los objetos de un directorio.

**Protocolo Ligero/Simplificado de Acceso a Directorios**

Las aplicaciones que deseen acceder a un directorio, deben utilizar el Protocolo Ligero/Simplificado de Acceso a Directorios, mejor conocido como LDAP (Lightweight Directory Access Protocol), el cual es un protocolo de tipo cliente-servidor que permite establecer la comunicación entre la aplicación y el servicio de directorio para que ésta pueda acceder o modificar la información del repositorio.

LDAP le brinda al usuario la posibilidad de:

* - Conectarse al directorio.
* - Desconectarse del directorio.
* - Buscar información.
* - Comparar información.
* - Insertar entradas.
* - Cambiar entradas.
* - Eliminar entradas.

**CAPÍTULO 3 MARCO METODOLÓGICO**

En todo desarrollo de software, aplicar cierto grado de disciplina siempre ayuda a conseguir un producto final de mayor calidad. Las metodologías ágiles son las más adecuadas para proyectos donde es necesaria la flexibilidad y la adaptación a los cambios, como es el caso de las aplicaciones web, que suelen tener requerimientos variables y usualmente se desea que estén disponibles en poco tiempo.

Algunas metodologías ágiles de uso común en la actualidad son: Programación Extrema (XP), Proceso Unificado Ágil (AUP), Proceso Unificado Abierto (OpenUP), Scrum, AgilUs, entre otras.

AgilUs es una metodología ágil que se adapta bien a proyectos individuales o de equipos pequeños, se centra en el usuario y sus necesidades, e incorpora técnicas de evaluación de usabilidad rápidas, económicas y que no requieren de plataformas tecnológicas complejas. Por todas estas razones, esta metodología resultó ser la adecuada para el desarrollo de este Trabajo Especial de Grado.

En este capítulo se explica en profundidad la metodología AgilUs, analizando sus principales características y describiendo las etapas que contempla.

**Metodología AgilUs**

La metodología AgilUs de Alecia Acosta nace como línea de investigación del Centro Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación de la Universidad Central de Venezuela. Se basa en el concepto de usabilidad, en la necesidad de desarrollar software usable. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin de que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo (Acosta, 2011).

AgilUs es un método de desarrollo iterativo e incremental que pone el mayor peso del desarrollo en la consecución de la usabilidad. Se centra en que la construcción de las interfaces de usuario se realice desde el comienzo del desarrollo del sistema y no al final.

Esta metodología reduce la cantidad de actividades y artefactos que se generan, propicia la participación del usuario y se adapta a los cambios que puedan surgir en el desarrollo. Está orientada al desarrollo de sistemas con alto grado de interactividad con los usuarios (front-end) y menos complejidad en la lógica de la aplicación (back-end).

AgilUs provee un conjunto de “buenas prácticas” para el desarrollo de software desde una perspectiva ágil, donde requiere involucrar al usuario de manera activa durante el proceso de desarrollo. Algunas de estas “buenas prácticas” son:

• Diseño centrado en el usuario (DCU): es un enfoque de diseño y desarrollo que se centra en los deseos, limitaciones y necesidades de los usuarios finales de un software.

• Diseño basado en prototipos: el desarrollo de software es guiado por la construcción de prototipos de alta fidelidad y la evaluación de los mismos por los usuarios y por especialistas en usabilidad.

• Desarrollo ágil, incremental e iterativo: el desarrollo es lo más simple posible, provee resultados sin necesidad de esclarecer todo los requerimientos al inicio del desarrollo y permite regresar a etapas anteriores una vez recibida la retroalimentación de las evaluaciones realizadas.

• Usabilidad como atributo de la calidad: se siguen las recomendaciones del estándar ISO/IEC 9126‐1 para la producción de software usable, ya que la usabilidad es considerada un atributo de la calidad interna y externa del software.

• Interacción continua con el usuario: la presencia constante y participativa del usuario es fundamental. La usabilidad del sistema sólo puede ser determinada por el usuario.

Acosta (2011) explica que AgilUs centra el desarrollo de software en los siguientes principios:

• Integra la Interacción Humano Computador (IHC) y la Ingeniería de Software (IS): IS e IHC son complementarias, no son disciplinas excluyentes. Un diseño centrado en el usuario impacta positivamente en la calidad del software (ISO 9126-1).

• La usabilidad debe considerarse desde el principio del desarrollo: la usabilidad aumenta la calidad del software, entonces es conveniente incluirla desde el principio en el desarrollo como uno de los requerimientos fundamentales para impactar positivamente en la calidad del producto final.

• La usabilidad determina la utilidad: un software se considera útil en la medida que pueda ser usado a fin de producir resultados, en forma eficiente, intuitiva y satisfactoria para los usuarios.

• El usuario determina la usabilidad: la usabilidad no es una propiedad abstracta. Un software sólo será considerado usable en un contexto específico y por un tipo de usuario específico. El objetivo es lograr que todos los usuarios del software encuentren usables las tareas que pueden realizar.

**Ciclo de vida**

“El ciclo de vida de AgilUs hace énfasis en la importancia del usuario y sus evaluaciones. Está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega” (Acosta, 2011).

AgilUs está formado por cuatro (4) etapas diferentes: Requisitos, Análisis, Entrega y Prototipaje, y en cada una de ellas se incluyen actividades y artefactos poco costosos para la construcción de la usabilidad, como se muestra en la

**Figura 8**

**Etapas**

A continuación se describe las cuatro (4) etapas de AgilUs y las técnicas de evaluación de usabilidad utilizadas (Acosta, 2011):

• Requisitos: se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del software,ya que un mal análisis de requerimientos traería como consecuencia un software que no cumple con las necesidades del usuario.

Las técnicas de evaluación de usabilidad (indagación) que se utilizan en esta etapa son:

- Tormenta de ideas: técnica de grupo para la generación de ideas nuevas y útiles, que permitan mediante reglas sencillas aumentar las probabilidades de innovación y originalidad.

- Encuesta: conjunto de preguntas normalizadas dirigidas a una muestra representativa de la población para obtener información estadística acerca de sus opiniones o hechos específicos.

- Cuestionario: lista de preguntas que se distribuye entre los usuarios para obtener información de sus opiniones. Este instrumento se puede utilizar en cualquier etapa del proceso de desarrollo, dependiendo de las preguntas que estén formuladas.

- Entrevista: dialogo entre un entrevistador y un entrevistado, donde el entrevistador hace una serie de preguntas al entrevistado con el fin de conocer sus opiniones e ideas.

- Evaluación de sistemas existentes: consiste en la revisión de versiones anteriores del mismo sistema, así como sistemas de la competencia o afines, con el objetivo de identificar ventajas, desventajas y problemas de usabilidad.

- Perfil de usuario: describe a los usuarios previstos del sistema, detallando sus

características más relevantes.

- Requerimientos funcionales y no Funcionales: recopilación de condiciones o

capacidades que debe cumplir el sistema.

• Análisis: se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.

Las técnicas de evaluación de usabilidad que se utilizan en esta etapa son:

- Prototipo en papel: bosquejo del sistema que se desea realizar. Resulta muy

útil cuando el cliente no tiene claros algunos puntos.

- Guía de estilo: documento que recoge normativas y patrones básicos relacionados con el aspecto de la interfaz para su aplicación en el desarrollo de nuevas pantallas dentro de su entorno.

- Modelo de casos de uso: modelo que permite especificar las funcionalidades y el comportamiento del sistema mediante su interacción con usuarios u otros sistemas.

- Modelo de objetos del dominio: modelo que describe y prioriza cada uno de los

objetos identificados en el sistema.

- Objetos de interfaz: modelo que permite comprender la parte visual de la

aplicación.

- Patrones de interacción: describen aspectos concernientes a la interfaz de usuario; están orientados a presentar soluciones apropiadas a problemas recurrentes que se les presentan a los usuarios cuando utilizan las aplicaciones interactivas.

• Prototipaje: se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final y se realizan evaluaciones de usabilidad apropiadas a esta etapa: evaluaciones heurísticas y listas de comprobación.

Las técnicas de evaluación de usabilidad que se utilizan en esta etapa son:

- Prototipo rápido: técnica basada en ordenador que pretende reducir el ciclo iterativo de desarrollo. Los prototipos iterativos desarrollados podrán ser rápidamente reemplazados o modificados según los informes procedentes de otras evaluaciones a medida que se evoluciona en el desarrollo de las tareas a realizar. Existen muchas herramientas para la generación de prototipos rápidos, siendo habituales una secuencia de imágenes en Microsoft PowerPoint o Visual Basic.

- Evaluación heurística: se basa en una inspección donde especialistas en usabilidad juzgan si cada elemento de la interfaz sigue los principios de usabilidad establecidos.

- Lista de comprobación: consta de la verificación del cumplimiento de una lista de principios que el equipo de desarrollo acuerda para el diseño de la interfaz de usuario.

- Pensamiento en voz alta: técnica donde se les pide a los usuarios que expresen en voz alta sus pensamientos, sentimientos y opiniones mientras que interaccionan con el sistema o un prototipo del mismo.

- Prototipo ejecutable: técnica donde el prototipo utilizado para las pruebas se corresponde con la interfaz real en la mayor medida posible. Es utilizado para maquetar la interfaz y en él son aceptadas entradas desde ratón o teclado, tal como lo haría la interfaz real, a su vez que responde a esos eventos de idéntica forma (mostrando ventanas, mensajes de respuesta, entre otros).

• Entrega: se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación.

Las técnicas de evaluación de usabilidad que se utilizan en esta etapa son:

- Pruebas de aceptación: pruebas que aportan datos tanto cuantitativos como cualitativos sobre usuarios reales que llevan a cabo tareas reales con el sistema. Los profesionales de la usabilidad pueden evaluar algunos aspectos de la accesibilidad utilizando protocolos normalizados de pruebas de usabilidad.

- Protocolo de preguntas: consiste en la formulación de preguntas directas al usuario acerca del sistema para detectar qué partes de la interfaz resultan obvias y que otras resultan oscuras.

**Errores comunes**

Acosta (2011) señala que al implementar AgilUs se pueden cometer algunos errores comunes que surgen a causa de la disminución deliberada o accidental de la participación del usuario en el proceso de desarrollo o de una mala comprensión del rol que debe ocupar a la hora de tomar decisiones con respecto al diseño del sistema y sus interfaces.

Uno de los errores más frecuentes es que el equipo de desarrollo determine la usabilidad del sistema sin incluir al usuario, intentando predecir o aproximar las reacciones, deseos, capacidades y carencias de éste. Otro error común es pensar que el cliente y el usuario son la misma persona. A pesar de que el cliente es el propietario del sistema, éste no debería tomar decisiones, por ejemplo en términos de usabilidad, ya que los usuarios son los que deberían decidir, indirectamente a través de sus evaluaciones y comentarios, qué se debe hacer y por qué.

**Tabla resumen**

**Resumen de la Metodología AgilUs**

**Definición**

Metodología ágil fuertemente orientada a la producción de software usable, por ello el usuario es el actor principal durante el proceso de desarrollo.

**Etapas**

- Requisitos. - Análisis. - Prototipaje. - Entrega.

**Roles** - No tiene roles.

Tiempos de entrega Al finalizar una iteración se entrega un prototipo para ser evaluado.

**Artefactos**

Etapa de Requisitos: - Tormenta de ideas. - Encuestas, entrevistas, cuestionarios. - Evaluación de sistemas existentes. - Perfiles de usuario. - Requerimientos funcionales y no funcionales.

Etapa de Análisis: - Prototipo en papel. - Guía de estilo. - Casos de uso. - Modelo de objetos del dominio. - Objetos de interfaz. - Patrones de interacción.

Etapa de Prototipaje: - Evaluación heurística. - Lista de comprobación. - Pensamiento en voz alta. - Prototipo ejecutable.

Etapa de Entrega: - Protocolo de preguntas. - Pruebas de aceptación. - Aplicación final.

**Ventajas**

- Constante interacción con el usuario durante todo el proceso de

desarrollo. - Incremento de la productividad, la satisfacción y la reducción de costos de entrenamiento y soporte del software, ya que las técnicas de evaluación de usabilidad propuestas en el método no son costosas ni requieren de infraestructura tecnológica compleja.

**Desventajas**

- No contempla en su ciclo de vida una fase de mantenimiento, lo cual dificulta su evolución y/o corrección una vez finalizado el desarrollo del producto. - Puede ocurrir una disminución de la participación del usuario en el proceso de desarrollo o puede haber una mala comprensión de su rol.

**CAPÍTULO 4 MARCO APLICATIVO**

En este capítulo se describen aspectos y características relevantes de la aplicación y se detallan los pasos que se realizaron para el desarrollo de este Trabajo Especial de Grado, siguiendo de manera parcial los lineamientos de la metodología AgilUs.

por ahora pienso que deberíamos poner aqui, arquitectura CLiente-servidor, API restfull, ruby on rails, mysql, que es un Mailer, definir que es un servicio de reclutamiento, que tipo de datos se piden en entrevistas de trabajo, que información válida a un profesional, como certificar una persona como valida, que tipo de pruebas te hace el sistema, como sera la configuracion, roles de usuarios (empresas/vacante - reclutados/interesados), definir un glosario de terminos, de donde sacaremos la información, como filtraremos a las personas, que cosas generarán información para el sistema, que tipo de feedback daremos, cosas asi… averiguar métodos de reclutamiento comunes y enumerar sus pros y contras para colocar una buena justificación del proyecto,

claro, esto es un marco teórico por ahora, pero mientras mas completo, menos trabajo tendremos que completar para que sea válido para el trabajo de grado.

si encuentras Ideas que aporten para este segmento escribelo, metodologías herramientas, gráficos, etc, todo sirve, tecnología a utilizar, cronograma del proyecto, lo que sea, cosas para complementar asi como el ejeplo dejado por el profesor, de las metodologias de gestion de proyectos (crei que deberia ser metodologias de ‘desarrollo de proyectos….’)

**Metodología de gestión de proyectos**

Marco teórico de la metodología de gestión de proyectos PMI

**Consideraciones éticas y legales**

En cuanto a la información documental consultada, se tuvo especial cuidado y respeto de los derechos de autor en diversos tópicos de interés emitidos por terceras personas, por lo que se hace referencia al autor original de las mencionadas.

Cabe mencionar que aunado a lo expuesto anteriormente, se hace uso de las normas APA para la escritura de las referencias bibliográficas y del instructivo proporcionado por la UCAB para elaborar trabajos especiales de grado correspondiente a estudios de tercer nivel, y se indica el aporte de diversos autores que contribuyen a la realización de la presente investigación.

**Capítulo 2**

**Metodología de gestión de proyectos**

Marco teórico de la metodología de gestión de proyectos PMI

Colocar de que se trata el capítulo 2, y luego empiezas a engranar, lo que dices en este párrafo, con el siguiente título que vas a decir, como por ejemplo el de la metodología de gestión de proyectos.

**Capítulo 3**

**Contexto del problema**

Objeto de estudio: Sistemas de recursos humanos.

El proceso de reclutamiento y selección de empresas consta de una serie de etapas que se aplica a cada candidato que desee aplicar para el cargo ofrecido.

Inicialmente, la empresa debe agendar y manejar el tiempo de tal forma que puedan atender a la mayor cantidad de candidatos en un día determinado, sin embargo, muchas empresas no poseen una forma de trabajo eficiente que permita evaluar a una gran cantidad de candidatos en el menor tiempo posible debido a que, generalmente, dichas evaluaciones se realiza por una sola persona que dedica parte de su tiempo a aplicar el proceso de selección, evaluación y contratación de a un candidato a la vez lo que causa que el proceso sea lento lo que puede ocasionar que los candidatos decidan retirarse del proceso.

Las empresas realizan impresiones de cuestionarios y encuestas que son aplicados a cada candidato generando pérdidas económicas si el candidato decide abandonar el proceso de selección, o si no es seleccionado para continuar, y si el candidato es seleccionado, se encuentre con el escenario de que dicho candidato ya aceptó una propuesta laboral de otra empresa debido a la lentitud del proceso y falta de feedback a lo largo del proceso. La lentitud del proceso y falta de feedback deja al candidato en incertidumbre en determinar la posibilidad que tiene en ser seleccionado y poder incursionar en el vacante lo que causa que la empresa pierda candidatos potenciales durante el proceso.

Las empresas, para reclutar al candidato ideal para la vacante, realizan diversos filtros que son aplicados a cada candidato a evaluar, lo que ocasiona que los candidatos se sientan incómodos ya que deben solicitar permisos laborales para asistir presencialmente a cada una de las etapas en el proceso de reclutamiento y selección, permisos que, son recuperados en tiempo de oficina o generan pérdidas monetarias en transporte, impresión de documentos como currículos, certificados, títulos, entre otros; esto ocasiona que los candidatos se retiren del proceso de selección por considerarlo lento, costoso y molesto.

Finalmente, en algunas empresas, el candidato seleccionado es invitado a proponer la propuesta económica, esto es, conocer del candidato la adquisición salarial que considera ganar, o la misma empresa realiza la propuesta económica además de darle a conocer los beneficios laborales que la empresa provee a sus empleados y el ambiente de trabajo. En este punto la empresa debe decidir si las consideraciones salariales son accesibles para la empresa y proceder con el reclutamiento del candidato, o en el caso de que la aspiraciones del candidato son mayores a los que la empresa puede ofrecer el candidato es retirado del proceso de selección.

En cualquiera de los casos donde el candidato fue contratado o no, la empresa ha perdido tiempo y dinero realizando dicho proceso, donde según el resultado, puede generar experiencias desagradables que motiva a futuros aspirantes a rechazar participación en el proceso de selección de otras vacantes ofrecidas por la empresa y generen referencias negativas hacia otros potenciales candidatos.

Mejorar la redacción.

Aparte, necesitan decir cosas que tengan más importancia.

Les recomiendo revisar el documento que les envíe, donde les coloco la estructura del seminario y la tesis. Hay una parte que dice Contexto del problema. Ahí hay cosas que ya pueden utilizar para el contexto del problema, y que serían un copiar y pegar, y además de las ideas que ustedes agreguen.

**Parte III**

**Propuesta de Trabajo Especial de**

**Grado**

Aquí pueden hacer algo muy similiar al lo que está en el documento de seminario que les pase en PDF.

Era algo como esto:

**Texto**

**Texto**

**Dibujo de la arquitectura de la aplicación**

**Capítulo 8**

En el presente capítulo se menciona <luego decir de que van hablar de forma general> y de forma que lo engranen con el siguiente título (Análisis y propuesta)

**Análisis y Propuesta**

Ejemplo de texto introductorio para luego abordar la situación actual: Como parte del análisis del presente trabajo especial de grado, se proporciona una descripción de diversos escenarios que muchas empresas, viven al momento de seleccionar y contratar personal, descrito a continuación:

**8.1. Situación Actual**

Hoy en día, el éxito de las organizaciones y empresas depende en gran medida de una buena selección de aspirantes que desean aportar sus conocimientos, experiencias y aptitudes para mejorar el funcionamiento y desarrollo de las tareas de la empresa.

La necesidad de reclutar nuevos aspirantes en las empresas viene dado por diferentes causas como requerir personal que contribuya a realizar, de forma eficaz y eficiente, una serie de actividades que contribuyan al logro de las diversas áreas de una organización y así alcanzar la misión y visión de dicha organización, o debido a factores externos tales como expansión de la empresa, lo que genera nuevos puestos de trabajos, pérdida de persona o despido, entre otros.

El proceso de reclutamiento de personal comienza a raíz de un requerimiento o solicitud que posee la empresa en desarrollar una serie de actividades, en un área determinada, donde la empresa carece de personal calificado para ejecutar dicha labor, donde posteriormente se procede a determinar:

1. El nombre de la vacante.
2. Las actividades y responsabilidades que deben desempeñarse en la vacante.
3. Requerimientos técnicos mínimos necesarios.
4. Requerimientos sociales e interpersonales necesarios.
5. Horario de trabajo disponible para la vacante.
6. Salarios
7. Número de vacantes.

Una vez que la empresa determina dichos datos sobre la vacante se procede a realizar la publicación de la vacante ya sea en el sitio web de la empresa o en sitios web donde potenciales candidatos pueden encontrar información de la vacante, donde posteriormente, se inicia un proceso de solicitud de una serie de candidatos para ocupar una vacante.

Dicho proceso, se ejecuta de forma presencial donde el candidato consulta información del puesto y entrega los documentos para su postulación. De esta forma, la empresa debe emplear tiempo en atender a cada candidato que desee postularse, y el aspirante debe dirigirse a la empresa a entregar los documentos para su postulación, posterior a ello, de acuerdo a las necesidades de la empresa, ciertos aspirantes serán seleccionados para continuar con el proceso de reclutamiento, donde se harán evaluaciones, entrevistas para conocer las expectativas del aspirante en la empresa, etc.

Todos estos procesos consumen recursos (tiempo y costos) para ambos lados del proceso; la empresa suele realizar impresiones de cuestionarios o exámenes lo que genera pérdidas económicas si el aspirante decide retirarse del proceso o no es seleccionado para continuar, requiere de mayor personal para poder proveer un feedback a cada aspirante sobre el estado en el proceso de selección en un tiempo suficiente que no genere incertidumbre en el aspirante sobre su selección, y por otro lado, cada aspirante se siente incómodo con las diversas etapas que debe realizar durante el proceso de selección ya que deben solicitar permisos laborales para poder asistir a dichas etapas, todo esto causa una pérdida de dinero en transportes, impresión de los documentos necesarios por la empresa como currículos, certificaciones, títulos, entre otros lo que origina que un candidato abandone el proceso por ser muy lento y costoso.

poco a poco ire quitando la redacción informal

**8.2. Propuesta de TEG**

De acuerdo a lo expuesto anteriormente, surge la iniciativa y oportunidad de crear una aplicación web que permita a la empresa ofrecer información sobre el puesto vacante y al aspirante poder postularse al proceso de selección y tener un seguimiento del mismo.

Se pretende con este trabajo especial de grado crear un sistema conformado por una aplicación web (donde se visualice la información, configurable por la empresa y el aspirante puede subir su información) para apoyar el proceso de reclutamiento y selección de personal de cualquier empresa.

Por una parte, cada empresa podrá configurar el portal de forma que provea a los aspirantes información detallada del puesto vacante, explicar las etapas a efectuarse durante el proceso de selección y proveer feedback a cada aspirante en cada etapa del proceso.

Por otra parte, el candidato podrá ingresar al portal y obtener información detallada de las vacantes disponibles por la empresa y, en caso de que le interese una vacante, postularse desde dicho portal ingresando la información necesario como currículo, documentos, certificaciones. Además el candidato interesado puede determinar su estado en el proceso de selección.

Lo anteriormente mencionado, permite al aspirante postularse sin la necesidad de realizar gastos en transporte e impresión de documentos y realizar un seguimiento sobre el estado de su solicitud, y permite a la empresa automatizar algunas etapas del proceso de selección y proveer a todos sus aspirantes el estado de su solicitud de forma rápida y contínua.

Por otra parte, la aplicación web, se encarga de capturar la información proporcionada por cada aspirante, y almacenarla utilizando un sistema de manejador de bases de datos. Además de poder analizar y generar información para facilitar a la empresa el proceso de selección.

**8.3. Justificación**

Todas las empresas requieren de realizar los procesos de evaluación y selección para poder progresar y expandir como negocio y compañía, pero dicho proceso requiere de tiempo y recursos de parte de ambas partes del proceso.

Por lo ante expuesto, surge la necesidad de desarrollar una aplicación Web que permita a las empresas atender a una gran cantidad de candidatos de manera simultánea, y que permita brindar respuesta a los potenciales candidatos, y que a su vez, le permitan a las empresas continuar o descartar a diversos candidatos, de forma rápida y eficiente, y sin originar los inconvenientes descritos.

El uso de un portal configurable por la empresa permite:

* Crear o actualizar la información de una vacante.
* Procesar las solicitudes de los candidatos que se postulen las 24 horas del día.
* Crear o reutilizar cuestionarios o encuestas configurables, automatizando la puntuación obtenida por los candidatos que la realizan.
* Reducir los tiempos del personal de Recursos Humanos en evaluar a cada candidato.
* Enviar feedback en cada etapa del proceso al candidato de forma eficiente y constante.

El uso de un portal por parte del candidato permite:

* Postularse desde el portal.
* Ingresar la información de los documentos requeridos para ser evaluado por la empresa como currículos y certificados de forma digital.
* Reducir costos de transporte.
* Realizar las etapas las 24 horas del día.
* Realizar un seguimiento del estado del proceso de selección eficiente.

Lo anterior mencionado da flexibilidad a los candidatos en realizar el proceso de selección y, a su vez, el seguimiento de su estado durante todo el proceso desde la comodidad de su casa, sin restricciones de tiempo, lo que evita que el candidato deba solicitar permisos para retirarse de la oficina para realizar entrevistas con la incertidumbre de saber si dicha entrevista es beneficiosa o no.

Por otro lado, da flexibilidad a la empresa en realizar y configurar los cuestionarios para que estos sean evaluados de forma automática sin participación del personal de Recursos Humanos, y encuestas que pueden ser reutilizados en múltiples vacantes disponibles en la empresa. Además puede proveer a todos sus candidatos el feedback sobre su estado durante el proceso de selección de forma eficiente eliminando la incertidumbre que dichos candidatos puedan generar durante el proceso.

**Bibliografía**

(recomiendo que veas el documento correccion 1 que está en la misma carpeta de este documento, está full de referencias facilitadas por el tutor)

(acontinuacion son otras que podrian ser de utilidad)

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES  
Acosta A. (2011). AgilUs: un método ágil de desarrollo de software que incorpora la usabilidad. Centro de Ingeniería de Software y Sistemas, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela.  
Adobe Dreamweaver. (2015). Aspectos básicos de las aplicaciones Web. Recuperado en junio de 2015, de: https://helpx.adobe.com/es/dreamweaver/using/web-applications.html  
Beck K. et al. (2001). Manifiesto por el Desarrollo Ágil de Software. Recuperado en junio 2015, de: http://www.agilemanifesto.org/iso/es/  
Borges C. y Rivero A. (2006). Generador de Sitios Web de Centros de Investigación. Centro de Ingeniería de Software y Sistemas, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela. Recuperado en julio 2015, de: http://www.coordinv.ciens.ucv.ve/investigacion/genci/index.php  
Cáceres P. et al. (2001). Procesos ágiles para el desarrollo de aplicaciones web. Departamento de Ciencias Experimentales e Ingeniería, Universidad Rey Juan Carlos, Madrid, España. Recuperado en junio 2015, de: http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf  
Calzada R. (2001). Introducción al Servicio de Directorio. Universidad Carlos III de Madrid, España. Recuperado en mayo 2016, de: http://www.rediris.es/ldap/doc/ldap- intro.pdf  
Canós J., Letelier P. y Penadés M. (2003). Metodologías Ágiles en el Desarrollo de Software. Universidad Politécnica de Valencia, España. Recuperado en junio 2015, de: http://www.carlosfau.com.ar/nqi/nqifiles/XP\_Agil.pdf  
Consejo Directivo del Observatorio Nacional de Ciencia, Tecnología e Innovación (ONCTI). (2015). Reglamento del Programa de Estímulo a la Innovación e Investigación (PEII). Caracas, Venezuela. Recuperado en julio 2015, de: http://www.oncti.gob.ve/index.php?option=com\_phocadownload&view=category&download =119:reglamento-del-programa-de-estimulo-a-la-innovacion-e-investigacion- peii&id=11:documentos&Itemid=92  
Coordinación de Investigación. (2011). ¿Quiénes Somos?. Caracas, Venezuela: Facultad de Ciencias de la Universidad Central de Venezuela. Recuperado en julio 2015, de http://www.coordinv.ciens.ucv.ve/investigacion/quienes.php  
108  
Referencias bibliográficas y digitales  
De Luca D. (2010). ¿Qué es CSS3? [Mensaje en un blog]. Recuperado en julio 2015, de: http://html5.dwebapps.com/que-es-css3/  
Directorio Nacional de Investigadores e Innovadores (DINA). (2015). ¿Quiénes somos?. Lima, Perú. Recuperado en diciembre 2015, de http://dina.concytec.gob.pe/  
EcuRed. (2012). Lenguaje de Marcado de Hipertexto. Cuba. Recuperado en julio 2015, de: http://www.ecured.cu/index.php/Lenguaje\_de\_Marcado\_de\_Hipertexto  
EcuRed. (2012). Directorio Activo. Recuperado en mayo 2016, de: http://www.ecured.cu/Directorio\_Activo  
España M. (2003). Servicios Avanzados de telecomunicación. Madrid, España: Editorial Díaz de Santos S.A.  
Graterol L., Magrí S y Marín S. (2014). PHP, Python o Ruby ¿Qué son y para qué sirve cada uno? [Mensaje en un blog]. Recuperado en julio 2015, de: http://blog.escuelaweb.net/php-python-o-ruby-para-que-sirve-cada-uno/  
Laudon K. y Laudon J. (2004). Sistemas de Información Gerencial. México: Editorial Pearson Educación.  
Ministerio del Poder Popular para Ciencia, Tecnología e Innovación (MCTI). (21 de marzo de 2011). Gobierno Revolucionario fortalecerá las actividades científico tecnológicas del país. Caracas, Venezuela. Recuperado en julio 2015 de: http://www.mcti.gob.ve/actualidad/noticias/gobierno-revolucionario-fortalecera-las- actividades-cientifico-tecnologicas-del  
Nielsen J. (1995). 10 Usability Heuristics for User Interface Design. Recuperado en mayo 2016, de: http://www.useit.com/papers/heuristic/heuristic\_list.html  
Olivares C. (2012). GENCI–2 Gestor de Contenido Modular Para la Coordinación de Investigación De la Facultad de Ciencias de la UCV (tesis de pregrado). Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, Caracas, Venezuela.  
Pentaho. (2012). PENTAHO. Recuperado en mayo de 2016, de http://www.pentaho.com/  
Piattini M. (1996). Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Madrid, España: Editorial Ra-Ma.  
Rails Guides. (2015). Ruby on Rails Guides (v4.2.4). Recuperado en julio 2015, de: http://guides.rubyonrails.org/  
109  
Referencias bibliográficas y digitales  
Rivera J. (2012). Pentaho Data Integration (Kettle). Recuperado en mayo 2016, de: http://www.docfoc.com/pentaho-data-integration-56338932d4bf0  
Ruby Lang Org. (2015). Acerca de Ruby. Recuperado en julio 2015, de: https://www.ruby-lang.org/es/  
Sistema de Acreditación de los Investigadores de la Universidad de Oriente (SAI-UDO). (2011). Instrucciones. Venezuela. Recuperado en julio 2015, de: http://www.saiudo.ci- udo.com.ve/  
Sistema de Acreditación de los Investigadores de la Universidad de Oriente (SAI-UDO). (2011). PAIC-UDO. Venezuela. Recuperado en julio 2015, de: http://www.saiudo.ci- udo.com.ve/  
Silberschatz A., Korth H. y Sudarshan S. (2002). Fundamentos de Bases de Datos. (4ta ed.). Madrid, España: Mc Graw Hill.  
Solis J. (2014). ¿Qué es bootstrap y cómo funciona en el diseño web?. Recuperado en mayo 2016, de: http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y- como-funciona-en-el-diseno-web.htm  
The PostgreSQL Global Development Group. (1996-2015). PostgreSQL. Recuperado en julio 2015, de: http://www.postgresql.org/about/  
W3Schools. (2015). AJAX Introduction. Recuperado en julio 2015, de: http://www.w3schools.com/Ajax/ajax\_intro.asp  
2ndQuadrant Ltd (2001-2015). 2ndQuadrant Professional PostgreSQL. Recuperado en julio 2015, de: http://2ndquadrant.com/es/postgresql/