

**UNIVERSIDAD TÉCNICA DE MACHALA**  
**UNIDAD ACADEMICA DE INGENIERÍA CIVIL**  
**CARRERA DE INGENIERÍA DE SISTEMAS**

**TRABAJO DE TITULACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS**

**TEMA:**

**“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB  
INTEGRADA AL PROGRAMA PERSONAL COMMUNICATIONS  
ISERIES ACCESS ORIENTADA A LA GESTIÓN DEL SISTEMA  
COMERCIAL (SICO-CNEL) PARA LA AUTOMATIZACIÓN DE  
PROCESOS MANUALES DE CONIEL CIA.LTDA”**

**AUTOR:**

**CÓRDOVA DÁVILA JHONSSON XAVIER**

**TUTOR:**

**ING. FAUSTO REDROVÁN CASTILLO, MG. SC.**

**CO-TUTOR**

**ING. CARTUCHE VARGAS JOFFRE**

**MACHALA - EL ORO - ECUADOR**

**2014**

## **CERTIFICACIÓN DEL TUTOR**

Ing. FAUSTO REDROVÁN CASTILLO, Mg. Sc, Profesor de la Unidad Académica de Ingeniería Civil de la Universidad Técnica de Machala, en calidad de Tutor de la Tesis de Grado titulada: “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB INTEGRADA AL PROGRAMA PERSONAL COMMUNICATIONS ISERIES ACCESS ORIENTADA A LA GESTIÓN DEL SISTEMA COMERCIAL (SICO-CNEL) PARA LA AUTOMATIZACIÓN DE PROCESOS MANUALES DE CONIEL CIA.LTDA”, elaborado por el estudiante JHONSSON XAVIER CÓRDOVA DÁVIL, egresado de la carrera de Ingeniería de Sistemas de la Escuela de Informática, certifico que la mencionada Tesis estuvo bajo mi dirección y supervisión ajustándose a los procedimientos académicos y metodológicos establecidos por la Facultad, razón por la que autorizo su presentación para el trámite legal correspondiente.

---

Ing. Fausto Redrován Castillo, Mg. Sc.

## **CERTIFICACIÓN DEL CO-TUTOR**

Ing. JOFFRE CARTUCHE VARGAS, Profesor de la Unidad Académica de Ingeniería Civil de la Universidad Técnica de Machala, en calidad de Co-Tutor de la Tesis de Grado titulada: “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB INTEGRADA AL PROGRAMA PERSONAL COMMUNICATIONS ISERIES ACCESS ORIENTADA A LA GESTIÓN DEL SISTEMA COMERCIAL (SICO-CNEL) PARA LA AUTOMATIZACIÓN DE PROCESOS MANUALES DE CONIEL CIA.LTDA”, elaborado por el estudiante JHONSSON XAVIER CÓRDOVA DÁVILA, egresado de la carrera de Ingeniería de Sistemas de la Escuela de Informática, certifico que la mencionada Tesis estuvo bajo mi dirección y supervisión ajustándose a los procedimientos académicos y metodológicos establecidos por la Facultad, razón por la que autorizo su presentación para el trámite legal correspondiente.

---

Ing. Joffre Cartuche Vargas

## **DECLARACIÓN DE AUTORÍA**

La información vertida en este trabajo investigativo son de exclusiva autoría del Sr. Jhonsson Xavier Córdova Dávila con C.I. N° 070509557-8, quien es responsable de las ideas expuestas en el presente documento.

---

Jhonsson Xavier Córdova Dávila

## **AGRADECIMIENTO**

En primer lugar doy gracias a Dios, por haberme dado fuerza y valor para culminar esta etapa de mi vida.

Agradezco también la confianza y el apoyo brindado por parte de mis padres, que sin duda alguna en el trayecto de mi vida me han demostrado su amor, corrigiendo mis fallas y celebrando mis triunfos.

A mi mami Livia, que con su demostración de una segunda madre para mí me ha enseñado a no desfallecer ni rendirme ante nada y siempre perseverar a través de sus sabios consejos.

Al Ingeniero Fausto Redrován por toda la colaboración brindada, durante la elaboración de este proyecto.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto.

***Jhonsson Córdova***

## **DEDICATORIA**

Dedico este trabajo principalmente a Dios, el que me ha dado fortaleza para continuar cuando a punto de caer he estado.

De igual manera a mis padres, que han sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mi familia que siempre ha estado junto a mí brindándome su apoyo.

***Jhonsson Córdova***

## RESUMEN EJECUTIVO

CONIEL CIA.LTDA es una empresa dedicada a ejecutar trabajos de control de pérdidas adjudicados por CNEL EP Unidad de Negocios El Oro, por lo que maneja gran cantidad de información utilizando procesos manuales, esto hace que el procesamiento de los datos sea tedioso y lento.

Es esto lo que me ha motivado a llevar a cabo una propuesta actual desarrollada para el personal administrativo, con el fin de implementar una aplicación web que permita optimizar los recursos utilizados para la ejecución de las tareas dentro de la empresa.

Esta aplicación permitirá realizar actividades en menos tiempo, con fácil acceso a la información, a través de una interfaz amigable y de fácil uso, basada en la gestión de proyectos de la metodología ágil Uwe bajo un lenguaje de programación Python y utilizando tecnologías web de desarrollo CSS3, HTML5, JavaScript, con base de datos PostgreSQL.

La implementación de la aplicación web, ayudó a mejorar el conocimiento y la utilización de las TIC's.

***Palabras Clave:*** tedioso, interfaz amigable, aplicación web, TIC's

Revisado por: Ing. Nancy Loja Mora.

COMISION DE INVESTIGACIÓN DE LA CARRERA DE INGENIERÍA DE SISTEMAS

## **EXECUTIVE SUMMARY**

## INTRODUCCIÓN

Desde hace algunos años se ha considerado a la información como una parte fundamental en las empresas y por lo que el tratamiento de la misma debe ser de especial cuidado, además el índice de recolección y procesamiento de la información son un factor que impera en la determinación de las competencias de una empresa, ya que es muy importante para la toma de decisiones.

Hoy en día los procesos administrativos en las medianas y grandes empresas son llevados a través de mecanismos automatizados que mantienen una alta disponibilidad de información, así como un procesamiento ágil y eficiente de los datos a través de tecnologías que facilitan el manejo de la información, la optimización de los tareas dentro de una organización permite abolir las actividades repetitivas y carecientes de eficiencia a sistemas centralizados, eficiente y que brindan mejores resultados.

En nuestro medio existen varias empresas destinadas a brindar servicios de control de pérdidas de energía a nivel nacional, las mismas que manejan grandes cantidades de información y realizan una serie de pasos para concluir con su procesamiento, ya que no hacen usos de nuevas tecnologías destinadas a minimizar los tiempos de ejecución por lo que muchas veces el tratamiento de información demora 48 horas laborables, por lo que la compañía de Construcciones e Instalaciones Eléctricas CONIEL CIA.LTDA aun utilizan métodos manuales y tediosos como: informes en hojas de cálculos, ingreso de información al sistema realizando varios pasos que son repetitivos para cada abonado, el uso de interfaz poco intuitiva.

Para mejorar el desarrollo de las actividades administrativas que realizan las empresas dedicadas a los trabajos de control de pérdida de energía, es necesario implementar nuevos mecanismos para optimizar el uso de tecnologías de desarrollo web para la creación una aplicación que permita la simplificación de recursos, tiempo y pasos para la ejecución de un proceso.

Para el desarrollo e implementación de la aplicación web se utilizó el lenguaje de programación Python a través del framework Django aplicando la metodología ágil Uwe y con una base de datos POSTGRESQL para el almacenamiento de la información.



## JUSTIFICACIÓN

Actualmente los sistemas migran a la web por su facilidad de acceso, usabilidad e interacción con el usuario, con esto se logra un mejor desempeño en las aplicaciones ya que se reduce el uso de los recursos de hardware e incrementa el tiempo de vida de la misma en función de un solo equipo.

El desarrollo de aplicaciones web se realiza a través de tecnologías específicas de tal manera que se garantice mejores resultados que una aplicación de escritorio, tomando en cuenta los factores de tiempo de respuesta, trabajo en conjunto y disponibilidad.

La compañía CONIEL CIA.LTDA realiza trabajos relacionados a la ingeniería eléctrica, por lo que hace uso del sistema actual que posee una serie de desventajas que no ayudan a obtener un máximo rendimiento en el desarrollo de los procesos.

Considerando lo señalado se propone denominar este proyecto como **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB INTEGRADA AL PROGRAMA PERSONAL COMMUNICATIONS ISERIES ACCESS ORIENTADA A LA GESTIÓN DEL SISTEMA COMERCIAL (SICO-CNEL) PARA LA AUTOMATIZACIÓN DE PROCESOS MANUALES DE CONIEL CIA.LTDA”**

Con esta aplicación se busca obtener una aplicación web con opciones de tareas comunes realizadas por el personal administrativo que garantice la optimización de los procesos, la disminución del tiempo de ejecución de las tareas y el fácil acceso, haciendo uso de las nuevas tecnologías para brindar un mejor desarrollo en los procesos productivos de la compañía CONIEL CIA.LTDA.

# **1. MARCO REFERENCIAL**

## **1.1 PLANTEAMIENTO DEL PROBLEMA**

CONIEL CIA.LTDA es una empresa dedicada a ejecutar trabajos de control de pérdidas adjudicados por CNEL EP Unidad de Negocio El Oro a través del portal de compras públicas Sercop ([www.compraspublicas.gob.ec](http://www.compraspublicas.gob.ec)), la misma encargada de gestionar información relevante del contrato al Sistema Comercial SICO, de la cual se debe guardar un registro en un sistema local para su posterior facturación, dicho sistema, es manual y es controlado por una persona.

Los contratos de Pérdidas y Control de Energía se deben llevar con un manejo de información estricto, ya que estos datos se entregan periódicamente de forma digital a la CNEL EP, la información es tabulada con la ayuda de archivos de Excel. Las actividades principales que se realizan en estos contratos son: cambios de medidores, mantenimientos, servicios nuevos y consultas de información de usuarios y medidores, son realizadas a través del Sistema Comercial (SICO), que es un sistema distribuido con un servidor AS/400 que trabaja bajo una terminal IBM Personal Communications.

El sistema SICO posee una interfaz que no facilita la exploración entre las opciones que ofrece, es de complejidad muy alta a la hora de realizar una tarea, ya que las mismas se componen de varios pasos que deben ser ejecutados sin falta y en secuencia determinada, por lo que el usuario final debe someterse a capacitaciones extensas y que la mayoría de veces no brindan los resultados que se espera. Por lo que algunos procesos que se realizan dentro de las actividades administrativas son redundantes y no permiten la utilización adecuada de los recursos.

### **1.1.1 PROBLEMA CENTRAL**

Los procesos llevados actualmente en Coniel son ineficientes de tal manera que el tratamiento de la información conlleva un coste de producción muy alto debido a la no reutilización de la información proporcionada por el sistema SICO.

### **1.1.2 PROBLEMAS PARTICULARES**

- ✓ El Sistema que maneja CONIEL para el tratamiento de la información está basado en procesos ineficientes que conlleva a actividades repetitivas por lo que es necesario la optimización de los mismos.
- ✓ El manejo del sistema actual es tedioso ya que la interfaz de usuario que posee es difícil de comprender, de aprender y de usar.
- ✓ La empresa cuenta con un sistema que no suple con los requisitos de usuarios, es decir que no le permite optimizar los procesos manuales.
- ✓ A la aplicación actual no se le ha realizado una gestión de plan de pruebas que permita corroborar que en la ejecución de un proceso se necesita de una serie de pasos redundantes.
- ✓ La integridad y confiabilidad de la información en ocasiones se ha perdido ya que se la almacena en discos locales o en hojas impresas.

## **1.2 PREGUNTAS CIENTIFICAS**

### **1.2.1 PREGUNTA CENTRAL**

¿La implementación de la aplicación web permitirá automatizar los procesos administrativos de la compañía Coniel Cia.Ltda y mejorará el tratamiento de la información de tal manera que disminuya el coste de producción de las actividades?

### **1.2.2 PREGUNTAS COMPLEMENTARIAS**

- ✓ ¿La aplicación web desarrollada permitirá automatizar el tratamiento de la información dentro de la compañía evitando la redundancia de pasos o actividades para la ejecución de un proceso?
- ✓ ¿La navegabilidad de la aplicación ofrecerá una comprensión clara y una mejor orientación en las acciones a realizar por el usuario?
- ✓ ¿Si se desarrolla una aplicación tomando en cuenta los requisitos de usuario con el fin de implementar una interfaz, mejorará el sistema manual actual?
- ✓ ¿Si se establece un plan de pruebas, esto permitirá la verificación y validación de posibles errores en la ejecución de la aplicación?
- ✓ ¿Si se implementará un sistema distribuido en un servidor, los procesos aislados se pueden unificar para mejorar la administración de la empresa?

## **1.3 OBJETIVOS**

### **1.3.1 OBJETIVO PRINCIPAL**

Implementar una aplicación web integrada al programa PERSONAL COMMUNICATIONS ISERIES ACCESS orientada a la gestión del sistema comercial (SICO-CNEL) para la automatización de procesos manuales de CONIEL CIA.LTDA utilizando la metodología UWE en conjunto con el Framework Django.

### **1.3.2 OBJETIVOS ESPECIFICOS**

- ✓ Recopilar y analizar los requisitos y necesidades que forman parte de los procesos de tratamiento y administración de información en Coniel Cia. Ltda., a través de reuniones, diálogos y encuestas realizadas al personal encargado, con el fin de determinar un punto de inicio para el desarrollo de la aplicación.
- ✓ Diseñar las interfaces de usuario en función a los requisitos previamente sintetizados, mediante herramientas de diseño para la determinación de la estructura base de la aplicación.
- ✓ Codificar la funcionalidad de las interfaces tomando como punto de partida los prototipos y estructura, con el fin de constituir una aplicación dinámica y funcional, utilizando frameworks para desarrollo de aplicaciones web.
- ✓ Realizar pruebas de funcionamiento a la aplicación, utilizando herramientas de testeo de aplicaciones web, con la finalidad de retroalimentar el código escrito para lograr una aplicación robusta y estable.

- ✓ Imprimir un servidor web interno mediante la utilización de tecnologías actuales para la puesta en producción de la aplicación.

## 1.4 ALCANCE

Para el desarrollo del proyecto de tesis se realizará una investigación en la compañía de Construcciones e Instalaciones Eléctricas Coniel Cia.Ltda., poniendo énfasis a los procesos que realiza el personal administrativo durante sus actividades diarias e identificar y analizar los problemas que tienen con el tratamiento y procesamiento de la información en el sistema actual, para de esta manera integrarlos en una aplicación web que permitirá un eficiente manejo de los datos disminuyendo los tiempos de ejecución de cada proceso.

La aplicación será diseñada de tal manera que sea intuitiva y de fácil manejo, para que el personal administrativo tenga una idea clara de las opciones que puede realizar, pudiendo así obtener mayor productividad y usabilidad de los recursos con los que cuenta la empresa.

Para la realización del trabajo de titulación se utilizará un lenguaje de programación Python, el Framework Django, base de datos PostgreSQL, además para la gestión del proyecto se usará la metodología ágil Uwe que contemplará las siguientes acciones:

MÓDULO	FUNCIÓN	SUBFUNCIÓN	DESCRIPCIÓN
<b>Login de la Aplicación</b>	Ingreso de usuario y contraseña	Validación y Verificación de los datos	Valida y Verifica los datos ingresados, la disponibilidad de la sesión, las sesiones activas en el servidor y finalmente se dará apertura a una sesión.
		Comprobación de la identidad del usuario ingresado	
		Ingreso a la aplicación	Permite el ingreso a la aplicación mediante un usuario y una contraseña.
		Comprobación de Sesiones del servidor	
		Comprobación de sesiones disponibles	
		Apertura de Sesión	
		Comprobación de acceso a contrato	
<b>Ingreso</b>	Ingreso de información de las actividades realizadas	Ingresar cambio de medidor al sistema SICO	Permite al personal administrativo ingresar la información de las actividades realizadas al sistema comercial SICO, para actualizar los cambios realizados a cada abonado.
		Ingresar cambio de materiales al sistema SICO	
		Ingresar servicios nuevo al sistema SICO	

<b>Búsqueda de información al sistema comercial SICO</b>	Búsqueda de información de abonados por criterio en el sistema comercial SICO.	Búsqueda de datos por cuenta (código único de abonado)	En este módulo permite realizar búsquedas por criterios al sistema comercial SICO, y provee información necesaria para la ejecución de otras tareas.
		Búsqueda de datos por medidor	
		Búsqueda de datos por nombre de abonado	
		Búsqueda de datos por geo código	
<b>Búsqueda de Actividades realizadas</b>	Búsqueda de las actividades realizadas	Verificación de información	Permite realizar búsquedas de las actividades realizadas por el personal, para controlar los estados de las mismas, además puede verificar información de alguna actividad específica.
		Seguimiento de estado de actividad	
<b>Cuadrillas</b>	Lista de información de cuadrilla	Ubicación de cuadrillas	Da acceso a los datos de las personas que conforman las cuadrillas, además de la ubicación en la que se encuentran laborando.
		Lista de detalle de cuadrillas	
<b>Avance de Obra</b>	Tabulación de datos	Porcentaje de avance en relación al total del contrato	Calcula el porcentaje de avance del total del contrato el mismo que ayuda a controlar que no se exceda del valor fijado y conocer la cantidad faltante para la culminación del contrato.
<b>Reportes</b>	Presentación de informes	Reporte de actividades realizadas	Genera reportes de las actividades realizadas y de los materiales utilizados, además de generar facturas por fecha o globales.
		Reporte de materiales utilizados	
		Factura por fecha	
		Factura Global	
<b>Administración del sistema</b>	Administrar módulos del sistemas	<b>Administración de usuarios:</b> <ul style="list-style-type: none"> <li>- Creación de usuarios y asignación de permisos.</li> <li>- Asignación de usuario a contrato</li> <li>- Asignación de usuario al sistema comercial SICO</li> </ul>	Permite administrar los módulos del sistema y solo el usuario Administrador puede ingresar y tener acceso a estas opciones.

		<b>Administración de Contratos</b> - Creación de contratos. - Asignación de usuario SICO a contrato.	
		Creación y Asignación de materiales de contrato	
		Creación y Asignación de rubros del contrato	

**Tabla 1.** *Detalle del alcance del proyecto*

## CÁPITULO II

### 2. MARCO TEÓRICO

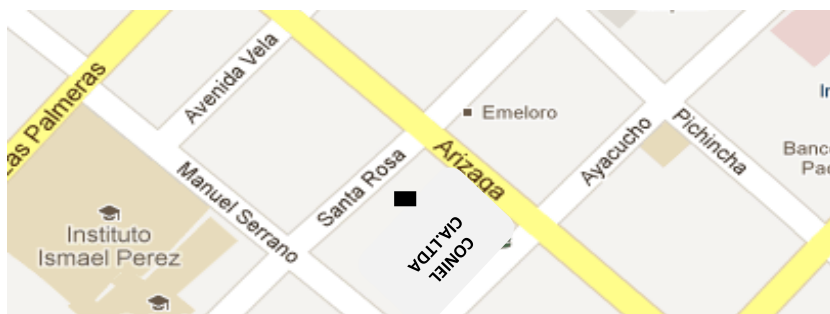
#### 2.1 ANTECEDENTES HISTÓRICOS DE LA EMPRESA

##### 2.1.1 RESEÑA HISTÓRICA DE CONIEL CIA.LTDA

La compañía de Construcciones e Instalaciones Eléctricas CONIEL CIA.LTDA, tiene sus inicios debido a la gran demanda de servicios eléctricos en nuestra Provincia. Fue constituida el 02 de julio del 2002, resultado de la asociación entre el Tnlg. Julio Loaiza y el Sr. Luis Pérez, para de esta manera dar paso al crecimiento de la empresa Privada cuyo único objetivo era ganar experiencia en el ámbito profesional y brindar servicios de calidad a la ciudadanía. Con el pasar de los años, la compañía es fuertemente constituida y cuenta con un amplio número de contratos firmados con la CNEL EP, los mismos que conforman su experiencia profesional. La empresa ha cumplido a cabalidad con las actividades de la prestación de los servicios adjudicado con la CNEL EP. La Compañía ha ido creciendo y ganando terreno en la Provincia y haciéndose acreedora a muy buenas referencias de trabajo por parte de la CNEL EP.

Actualmente es una compañía fuertemente establecida que brinda servicios a la comunidad trabajando en conjunto con la CNEL EP.

##### 2.1.2 UBICACIÓN

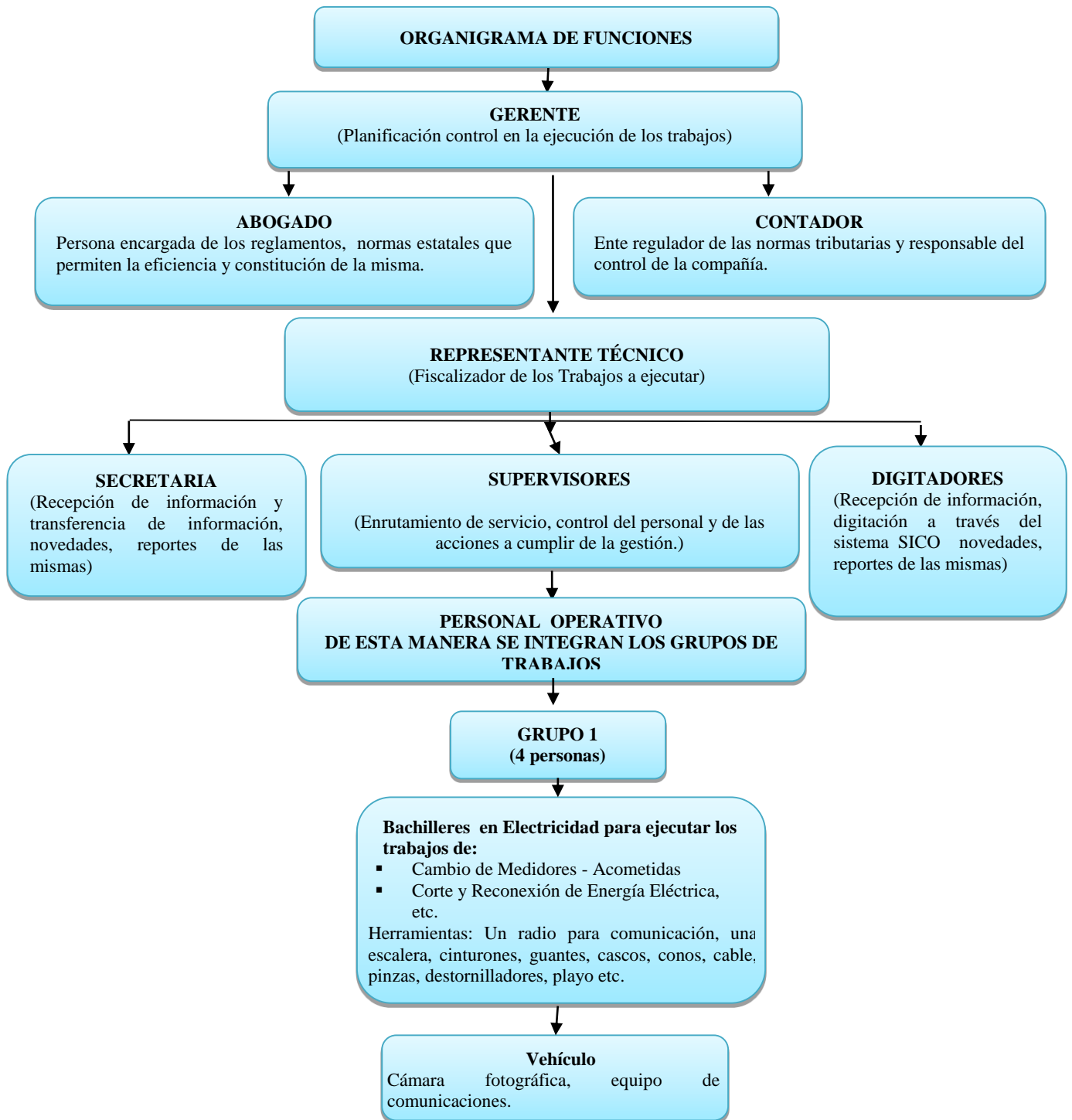


*Figura 1.* Croquis de ubicación de la Empresa Coniel Cía. Ltda.

*Fuente:* <https://www.google.com.ec/maps>

**Dirección:** Machala, Santa Rosa e/ Arizaga y Gral. Manuel Serrano.  
**Teléfono:** 2938-581

### 2.1.3 ORGANIGRAMA ESTRUCTURAL



**Figura 2.** Estructura Organización de CONIEL Cia.Ltda.

**Fuente:** Imagen proporcionada por CONIEL CIA.LTDA



## **2.2 ANTECEDENTES CONCEPTUALES**

### **2.2.1 APLICACIONES WEB**

#### **2.2.1.1 Concepto**

Es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (servidor web) y el protocolo mediante el que se comunica (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. Los entornos en los que se ejecutan las aplicaciones web pueden ser: intranet y extranet. (Luján, 2012, pág. 47)

#### **2.2.1.2 Ventajas**

- ✓ No requiere instalar software
- ✓ Bajo coste en actualizar los equipos con una nueva versión
- ✓ Acceso a la última y mejor versión
- ✓ Información centralizada
- ✓ Seguridad
- ✓ Movilidad
- ✓ Reducción de costes. (Pavón Maestras, 2012)

### **2.2.2 LENGUAJE DE PROGRAMACIÓN PYTHON**

#### **2.2.2.1 Concepto**

González Duque (2010) afirma que:

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Es un lenguaje interpretado, interactivo y orientado a objetos que ofrece una gran cantidad de estructuras de datos de alto nivel por medio de un tipado dinámico y fuerte, además de estas características es multiparadigma y multiplataforma. (pág. 7)

Python es un lenguaje de programación orientado a objetos que ofrece sintaxis limpias que favorecen un código legible.

#### **2.2.2.2 Características**

- ✓ “ Lenguaje interpretado o de script
- ✓ Tipado dinámico
- ✓ Fuertemente tipado
- ✓ Multiplataforma

- ✓ Orientado a objetos. ” (González Duque, 2010, pág. 6)

### 2.2.2.3 Elementos

Como en la mayoría de los lenguajes de programación de alto nivel, en Python se compone de una serie de elementos que alimentan su estructura. Entre ellos, podremos encontrar los siguientes:

- **Variables:**

Para Bahit (2012):

Una variable es un espacio para almacenar datos modificables, en la memoria de un ordenador. En Python, una variable se define con la sintaxis:

```
nombre_de_la_variable = valor_de_la_variable
```

Cada variable, tiene un nombre y un valor, el cual define a la vez, el tipo de datos de la variable. Existe un tipo de *variable*, denominada constante, la cual se utiliza para definir valores fijos, que no requieran ser modificados. Utilizar nombres descriptivos y en minúsculas. Para nombres compuestos, separar las palabras por guiones bajos. Antes y después del signo =, debe haber uno (y solo un) espacio en blanco. (pág. 30)

- **Tipo de Datos**

Según Bahit (2012):

Una variable (o constante) puede contener valores de diversos tipos. Entre ellos:

- a) Cadena de texto (string):**

```
mi_cadena = "Hola Mundo!"  
mi_cadena_multilinea = """  
Esta es una cadena  
de varias líneas  
"""
```

- b) Número entero:**

```
edad = 35
```

- c) Número entero octal:**

```
edad = 043
```

- d) Número entero hexadecimal:**

```
edad = 0x23
```

- e) Número real:**

```
precio = 7435.28
```

- f) Booleano (verdadero / Falso):**

```
verdadero = True
```

```
falso = False (pág. 31)
```

## - Operadores Aritméticos

Entre los operadores aritméticos que Python utiliza, podemos encontrar los siguientes: (Bahit, 2012, pág. 32)

Símbolo	Significado	Ejemplo	Resultado
+	Suma	a = 10 + 5	a es 15
-	Resta	a = 12 - 7	a es 5
-	Negación	a = -5	a es -5
*	Multiplicación	a = 7 * 5	a es 35
**	Exponente	a = 2 ** 3	a es 8
/	División	a = 12.5 / 2	a es 6.25
//	División entera	a = 12.5 / 2	a es 6.0
%	Módulo	a = 27 % 4	a es 3

*Tabla 2.* Operadores Aritméticos en Python.

*Fuente:* Libro Python para principiantes

### 2.2.2.4 Colecciones

González Duque (2010) especifica los siguientes tipos:

#### ✓ Listas

La lista es un tipo de colección ordenada. Sería equivalente a lo que en otros lenguajes se conoce por arrays, o vectores. Las listas pueden contener cualquier tipo de dato: números, cadenas, booleanos, y también listas. Crear una lista es tan sencillo como indicar entre corchetes, y separados por comas, los valores que queremos incluir en la lista: `l = [22, True, "una lista", [1, 2]]`

#### ✓ Tuplas

Es un tipo de colección ordenada, para lo que se utilizan paréntesis en lugar de corchetes. `t = (1, 2, True, "python")`, el constructor de la tupla es la coma, no el paréntesis. Además hay que tener en cuenta que es necesario añadir una coma para tuplas de un solo elemento, para diferenciarlo de un elemento entre paréntesis.

#### ✓ Diccionarios

Los diccionarios, también llamados matrices asociativas, deben su nombre a que son colecciones que relacionan una clave y un valor. (págs. 24 - 26)

### 2.2.2.5 Control de Flujo

Fernández Montoro (2014) menciona que existen las siguientes sentencias de control de flujo:

✓ **If, else, y elif**

La sentencia *if/else* funciona evaluando la condición indicada, si el resultado es True se ejecutara la siguiente sentencia o sentencias, en caso negativo se ejecutara las sentencias que aparecen a continuación del *else*. Python utiliza la indentación para establecer sentencias que pertenecen al mismo bloque. Además, en el carácter dos puntos (:) indica el comienzo del bloque.

✓ **For**

Para iterar existen dos sentencias que ayudan a crear bucles, for y while. La primera de ellas aplica a una serie de sentencias sobre cada uno de los elementos que contiene el objeto sobre el que se aplicará la sentencia for. (pág. 53)

### 2.2.3 FRAMEWORK WEB

“Una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación, se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.” (Gutiérrez, s.f, pág. 1)

Un framework web es diseñado para ayudar en el desarrollo web, permiten acelerar los procesos de desarrollo, reutilizar código ya escrito, además de promover buenas prácticas de desarrollo.

#### 2.2.3.1 Patrón MVC y Model 2

Gutiérrez (s.f.) describe que:

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (pág. 2)

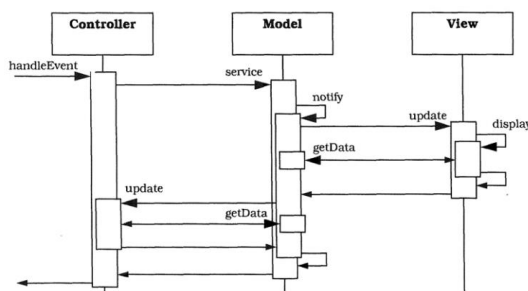
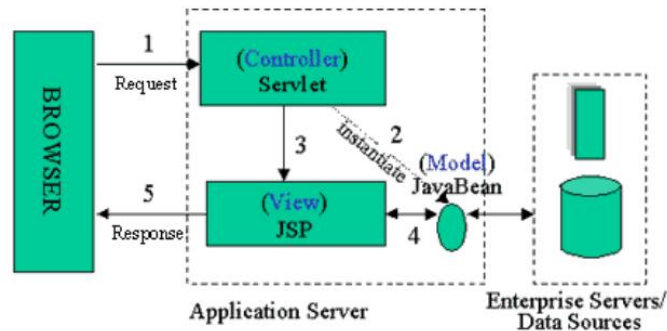


Figura 3. Modelo Vista Controlador

Fuente: [http://www.lsi.us.es/javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/javierj/investigacion_ficheros/Framework.pdf)

Una aplicación de este patrón en entornos Java para programación Web es lo que se conoce con el nombre de arquitectura model 2. Esta arquitectura consiste, a grandes rasgos, en la utilización de servlets para procesar las peticiones (controladores) y páginas JSP para mostrar la interfaz de usuario (vistas), implementando la parte del modelo mediante JavaBeans o POJOs. (págs. 2-3)



**Figura 4.** Patrón Model 2

**Fuente:** [http://www.lsi.us.es/javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/javierj/investigacion_ficheros/Framework.pdf)

## 2.2.4 DJANGO

Es un framework de desarrollo Web que ahorra tiempo, puedes crear y mantener aplicaciones Web de alta calidad con menos código. Permite enfocarse en la parte creativa, la parte divertida de las aplicaciones Web, al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. (Holovaty & Kaplan-Moss, 2010, pág. 21)

Django es un framework web escrito en python, permite escribir aplicaciones en menor tiempo y con pocas líneas de código.

### 2.2.4.1 Historia de Django

Holovaty & Kaplan-Moss (2010) narran que:

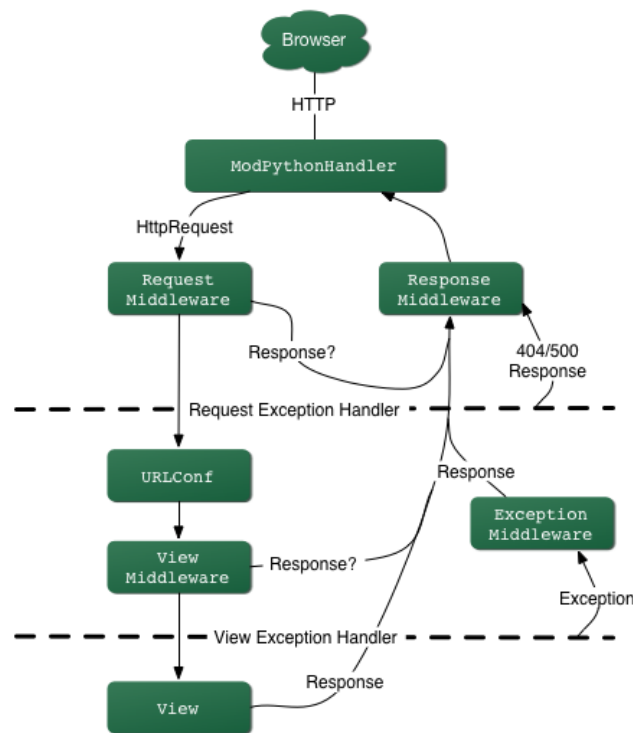
Django nació naturalmente de aplicaciones de la vida real escritas por un equipo de desarrolladores Web en Lawrence, Kansas. Nació en el otoño boreal de 2003, cuando los programadores Web del diario *Lawrence Journal-World*, Adrian Holovaty y Simon Willison, comenzaron a usar Python para crear sus aplicaciones. El equipo de The World Online, responsable de la producción y mantenimiento de varios sitios locales de noticias, prosperaba en un entorno de desarrollo dictado por las fechas límite del periodismo. Para los sitios incluidos LJWorld.com, Lawrence.com y KUsports.com los periodistas (y los directivos) exigían que se agregaran nuevas características y que aplicaciones enteras se crearan a una velocidad vertiginosa, a menudo con sólo días u horas de preaviso. Es así que Adrián y Simón desarrollaron por necesidad un framework de desarrollo Web que les ahorrara tiempo era la única forma en que podían crear aplicaciones mantenibles en tan poco tiempo.

En el verano boreal de 2005, luego de haber desarrollado este framework hasta el punto en que estaba haciendo funcionar la mayoría de los sitios World Online, el equipo de World Online, que ahora incluía a Jacob Kaplan-Moss, decidió liberar el framework como software de código abierto. Lo liberaron en julio de 2005 y lo llamaron Django, por el guitarrista de jazz Django Reinhardt. (pág. 23)

#### 2.2.4.2 ¿Cómo procesa una petición Django?

Holovaty & Kaplan-Moss (2010) explican que:

Además del mapeo directo de URLs con funciones vista, Django provee un poco más de flexibilidad en el procesamiento de peticiones. El flujo típico resolución de URLconf a una función de vista que retorna un `HttpResponse` puede ser corto-circuitado o *augmented* mediante middleware.



**Figura 5.** Flujo de una petición y respuesta a Django

**Fuente:** [http://librosweb.es/libro/django\\_1\\_0](http://librosweb.es/libro/django_1_0)

Cuando llega una petición HTTP desde el navegador, un *manejador* específico a cada servidor construye la `HttpRequest`, para pasarla a los componentes y maneja el flujo del procesamiento de la respuesta. El manejador luego llama a cualquier middleware de Petición o Vista disponible. Estos tipos de middleware son útiles para *augmenting* los objetos `HttpRequest` así como también para proveer manejo especial a determinados tipos de peticiones. En el caso de que alguno de los mismos retornara un `HttpResponse` la vista no es invocada.

Finalmente, el *middleware de respuesta* es bueno para el procesamiento posterior a un `HttpResponse` justo antes de que se envíe al navegador o haciendo una limpieza de recursos específicos a una petición. (Django 1.0, 2010, pág. 36)

#### **2.2.4.3 Sistema de plantillas de Django**

“Una plantilla de Django es una cadena de texto que pretende separar la presentación de un documento de sus datos. Una plantilla define rellenos y diversos bits de lógica básica (esto es, etiquetas de plantillas) que regulan cómo debe ser mostrado el documento. Normalmente, las plantillas son usadas para producir HTML, pero las plantillas de Django son igualmente capaces de generar cualquier formato basado en texto.” (Holovaty & Kaplan-Moss, 2010, pág. 46)

#### **2.2.5 POSTGRES**

Según Gibert Ginestà & Pérez Mora (2012), los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS,s) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: clases, herencia, tipos y funciones.

##### **2.2.5.1 Breve historia de Postgres**

Gibert Ginestà & Pérez Mora (2012) afirman que:

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y brevemente llamado Postgres95) está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

##### **2.2.5.2 PostgreSQL**

“Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema.” (PostgreSQL-ES, 2010)

Es un sistema de bases de datos relación orientada a objetos, de código libre, basado en el proyecto postgres y utiliza el lenguaje SQL.

### 2.2.5.3 Componentes de un sistema PostgreSQL

PostgreSQL-ES (2010) describe los siguientes componentes:

**Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir via TCP/IP ó sockets locales.

**Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autentificar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes

**Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg\_hba.conf y pg\_ident.conf

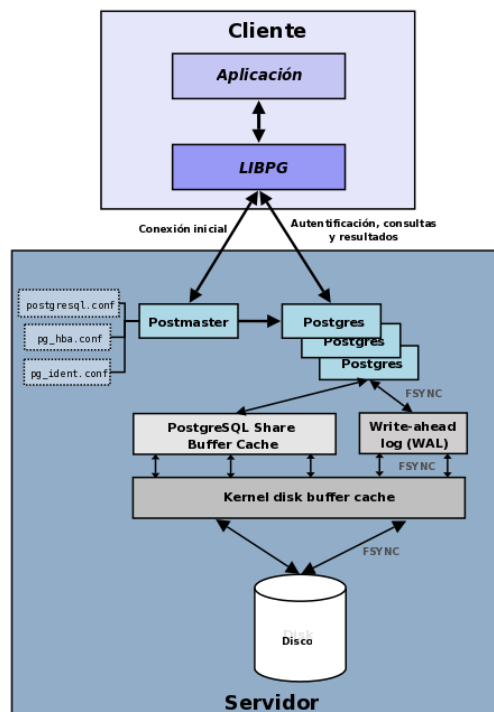
**Procesos hijos postgres:** Procesos hijos que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.

**PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.

**Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).

**Kernel disk buffer cache:** Caché de disco del sistema operativo

**Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.



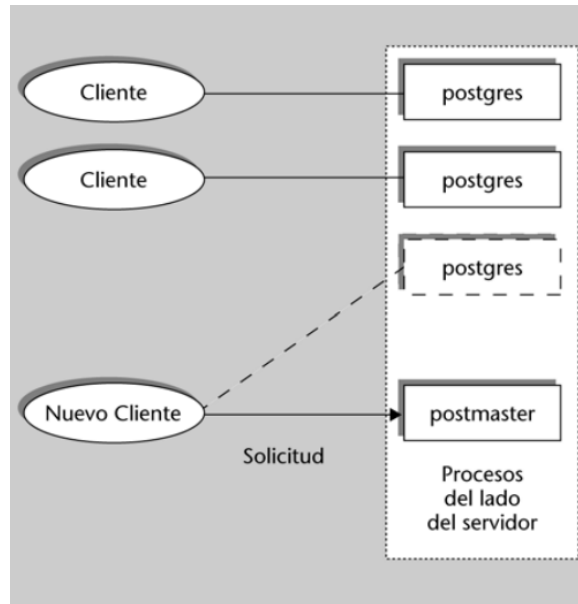
*Figura 6.* Componentes de un Sistema PostgreSQL

*Fuente:* [http://www.postgresql.org/es/sobre\\_postgresql](http://www.postgresql.org/es/sobre_postgresql)



#### 2.2.5.4 Arquitectura de PostgreSQL

“PostgreSQL está basado en una arquitectura cliente-servidor. El programa servidor se llama **Postgres** y entre los muchos programas cliente tenemos, por ejemplo, **pgaccess** (un cliente gráfico) y **psql** (un cliente en modo texto).” (Gibert Ginestà & Pérez Mora, 2012, pág. 65)



*Figura 7. Arquitectura de PostgreSQL*

*Fuente:* <http://ocw.uoc.edu/>

#### 2.2.6 HTML5

“Es un conjunto de etiquetas, es el lenguaje en el que es creada la web. También es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript.” (Vega & Vab Der Henst, 2011)

HTML 5, se trata de un conjunto de etiquetas que sirven para dar formato a diferentes tipos de contenidos, sobre todo texto.

##### 2.2.6.1 Componentes Básicos

Gauchat (2013) afirma que:

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que es extremadamente significativo. La estructura provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. (pág. 18)

### 2.2.6.2 Estructura global de un documento HTML5

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas. Los nuevos elementos semánticos incorporados en HTML5 son:

Según Gauchat (2013) los nuevos elementos HTML5 son los siguientes:

✓ **<!DOCTYPE>**

En primer lugar necesitamos indicar el tipo de documento que estamos creando. Esto en HTML5 es extremadamente sencillo: `<!DOCTYPE html>`. Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan. De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario.

✓ **<html>**

Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento `<html>`.

✓ **<head>**

El código HTML insertado entre las etiquetas `<html>` tiene que ser dividido entre dos secciones principales. Al igual que en versiones previas de HTML, la primera sección es la cabecera y la segunda el cuerpo. El elemento `<head>` va primero, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre.

✓ **<body>**

La siguiente gran sección que es parte principal de la organización de un documento HTML es el cuerpo. El cuerpo representa la parte visible de todo documento y es especificado entre etiquetas `<body>`. Estas etiquetas tampoco han cambiado en relación con versiones previas de HTML.

✓ **<meta>**

Algunos cambios e innovaciones fueron incorporados dentro de la cabecera, y uno de ellos es la etiqueta que define el juego de caracteres a utilizar para mostrar el documento.

✓ **<title>**

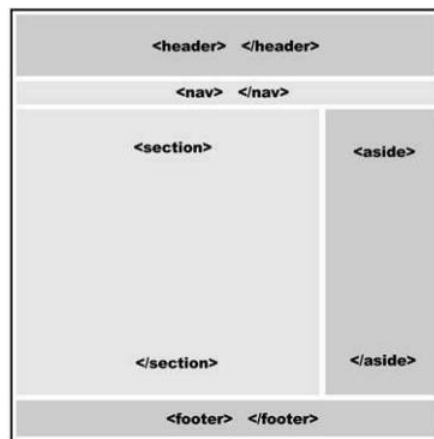
La etiqueta `<title>`, como siempre, simplemente especifica el título del documento: El texto entre las etiquetas `<title>` es el título del documento que estamos creando. Normalmente este texto es mostrado en la barra superior de la ventana del navegador.

✓ **<link>**

Elemento que va dentro de la cabecera del documento es `<link>`. Este elemento es usado para incorporar estilos, códigos JavaScript, imágenes o iconos desde archivos externos. Uno de los usos más comunes para `<link>` es la incorporación de archivos con estilos CSS. (págs. 19-22)

### 2.2.6.3 Estructura del cuerpo del documento HTML5

“La estructura del cuerpo (el código entre las etiquetas `<body>`) generará la parte visible del documento. HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo. Las secciones más importantes son diferenciadas y la estructura principal ya no depende más de los elementos `<div>` o `<table>`.” (Gauchat, 2013, pág. 24)



*Figura 8.* Representación visual de un diseño utilizando elementos HTML5.

*Fuente:* El gran libro de HTML5, CSS3 y JavaScript

Gauchat (2013) describe los siguientes elementos:

✓ **<header>**

Hacer cosas como `<div id="header">` es un poco estúpido cuando el 99% de los proyectos web tienen una cabecera. `<header>` está diseñada para reemplazar la necesidad de crear divs sin significado semántico.

✓ **<nav>**

Igual que `<header>`, `<nav>` está diseñado para que ahí coloques la botonera de navegación principal. Puedes colocar cualquier etiqueta dentro, aunque lo recomendado es usar listas `<ul>`.

✓ **<section>**

Define un área de contenido única dentro del sitio. En un blog, sería la zona donde están todos los posts. En un video de youtube, habría un section para el video, uno para los datos del video, otro para la zona de comentarios.

✓ **<aside>**

Cualquier contenido que no esté relacionado con el objetivo primario de la página va en un aside. En un blog, obviamente el aside es la barra lateral de información. En el home de un periódico, puede ser el área de indicadores económicos.

✓ **<footer>**

Este es obvio. Es el pie de página y todo lo que lo compone. (págs. 27-30)

#### **2.2.6.4 Elementos dentro del cuerpo del documento HTML5**

“Los elementos HTML5 ayudan a identificar cada sección del diseño, el contenido, etc. Por lo que se necesita poder diferenciar estos elementos y establecer una relación entre ellos dentro de la estructura.” (Gauchat, 2013, pág. 32)

Gauchat (2013) define los siguientes elementos:

✓ **<article>**

Define zonas únicas de contenido independiente. En el home de un blog, cada post sería un article. En un post del blog, el post y cada uno de sus comentarios sería un <article>.

✓ **<hgroup>**

Muchos headers necesitan múltiples títulos, como un blog que tiene un título y un tagline explicando el blog. <hgroup> permite colocar un h1, h2 y h3 dentro del header sin afectar el SEO, permitiendo usar otro h1 en el sitio. En el HTML actual, sólo puedes usar h1 una vez por sitio o el h1 pierde prioridad de SEO.

✓ **<figure> y <figcaption>**

La etiqueta <figure> fue creada para ayudarnos a ser aún más específicos a la hora de declarar el contenido del documento. Antes de que este elemento sea introducido, no podíamos identificar el contenido que era parte de la información pero a la vez independiente, como ilustraciones, fotos, videos, etc. (págs. 33-35)

## **2.2.7 INTRODUCCIÓN A CSS3**

Gauchat (2013) define que:

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Se utiliza el lenguaje CSS para definir el aspecto de cada elemento de una página: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (pág. 42)

CSS es un lenguaje que sirve para definir la presentación de un documento estructurado escrito ya sea en html o xml.

### **2.2.7.1 Breve Historia**

Vega & Vab Der Henst (2011) consideran que:

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores. El organismo W3C (<http://www.w3.org/>) (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve

propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (*Cascading HTML Style Sheets*) y la SSP (*Stream-based Style Sheet Proposal*).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (*Cascading Style Sheets*).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1". A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS. El 12 de Mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como "CSS nivel 2". La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2 que aún se está elaborando (la última actualización es del 19 de julio de 2007). La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño.

### 2.2.7.2 Conceptos básicos sobre estilos

Aplicar estilos a los elementos HTML cambia la forma en que estos son presentados en pantalla. Los navegadores proveen estilos por defecto que en la mayoría de los casos no son suficientes para satisfacer las necesidades de los diseñadores.

Gauchat (2013) define los siguientes estilos:

- ✓ **Estilos en línea**

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo style.

- ✓ **Estilos embebidos**

Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes. El elemento <style> permite a los desarrolladores agrupar estilos CSS dentro del documento. (págs. 45-46).

### 2.2.7.3 Nuevas propiedades CSS3

CSS3 provee nuevas propiedades para crear efectos visuales y dinámicos que son parte esencial de la web:

Según Gauchat (2012) las nuevas propiedades de css3 son las siguiente:

- ✓ **border-radius:**

Esta propiedad genera esquinas redondeadas para la caja formada por el elemento. Posee dos parámetros diferentes que dan forma a la esquina. El primer parámetro determina la curvatura horizontal y el segundo la vertical, otorgando la posibilidad de crear elipses.

- ✓ **box-shadow:**

Esta propiedad crea sombras para la caja formada por el elemento. Puede tomar cinco parámetros: el color, el desplazamiento horizontal, el desplazamiento vertical, el valor de difuminación, y la palabra clave inset para generar una sombra interna.

✓ **text-shadow**

Esta propiedad es similar a box-shadow pero específica para textos. Toma cuatro parámetros: el color, el desplazamiento horizontal, el desplazamiento vertical, y el valor de difuminación.

✓ **@font-face**

Esta regla nos permite cargar y usar cualquier fuente que necesitemos.

✓ **linear-gradient(posición inicio, color inicial, color final)**

Esta función puede ser aplicada a las propiedades background o background-image para generar un gradiente lineal. Los atributos indican el punto inicial y los colores usados para crear el gradiente.

✓ **radial-gradient(posición inicio, forma, color inicial, color final)**

Esta función puede ser aplicada a las propiedades background o background-image para generar un gradiente radial. La posición de inicio es el origen y puede ser declarado en píxeles, porcentaje o como una combinación de las palabras clave center, top, bottom, left y right. Existen dos valores para la forma: circle y ellipse, y puntos de terminación pueden ser declarados para cada color indicando la posición donde la transición comienza.

✓ **rgba()**

Esta función es una mejora de rgb(). Toma cuatro valores: el color rojo (0-255), el color verde (0-255), el color azul (0-255), y la opacidad (un valor entre 0 y 1).

✓ **hsla()**

Esta función es una mejora de hsl(). Puede tomar cuatro valores: el tono (un valor entre 0 y 360), la saturación (un porcentaje), la luminosidad (un porcentaje), y la opacidad (un valor entre 0 y 1).

✓ **outline**

Esta propiedad fue mejorada con la incorporación de otra propiedad llamada outline-offset. Ambas propiedades combinadas generan un segundo borde alejado del borde original del elemento.

✓ **border-image**

Esta propiedad crea un borde con una imagen personalizada. Necesita que el borde sea declarado previamente con las propiedades border o border-width, y toma al menos tres parámetros: la URL de la imagen, el tamaño de las piezas que serán tomadas de la imagen para construir el borde, y una palabra clave que especifica cómo esas piezas serán ubicadas alrededor del elemento.

✓ **transform**

Esta propiedad modifica la forma de un elemento. Utiliza cuatro funciones básicas: scale (escalar), rotate (rotar), skew (inclinarse), y translate (trasladar o mover).

✓ **transition**

Esta propiedad puede ser aplicada para crear una transición entre dos estados de un elemento. Recibe hasta cuatro parámetros: la propiedad afectada, el tiempo que le tomará a la transición desde el comienzo hasta el final, una palabra clave para especificar cómo la transición será realizada (ease, linear, ease-in, ease-out, ease-in-out) y un valor de retardo que determina el tiempo que la transición tardará en comenzar. (págs. 62-66)

## 2.2.8 JAVASCRIPT

Gauchat (2013) afirma que:

JavaScript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web. (pág. 84)

JavaScript es un lenguaje de programación que se usa para crear páginas web dinámicas, incorporando efectos, animaciones, etc.

### 2.2.8.1 Operadores

Murphey (2013) considera los siguientes operadores:

**a) Operadores Básicos**

Los operadores básicos permiten manipular valores.

✓ **Concatenación**

```
var foo = 'hola';  
var bar = 'mundo';  
console.log(foo + ' ' + bar); // la consola de depuración muestra 'hola mundo'
```

✓ **Multipliación y división**

```
2 * 3;  
2 / 3;
```

✓ **Incrementación y decrementación**

```
var i = 1;  
var j = ++i; // incrementación previa: j es igual a 2; i es igual a 2  
var k = i++; // incrementación posterior: k es igual a 2; i es igual a 3
```

## b) Operaciones con Números y Cadenas de Caracteres

En JavaScript, las operaciones con números y cadenas de caracteres (en inglés strings) pueden ocasionar resultados no esperados.

### ✓ Suma vs. Concatenación

```
var foo = 1;
var bar = '2';

console.log(foo + bar); // error: La consola de depuración muestra 12. Forzar a
una cadena de caracteres actuar como un número

var foo = 1;
var bar = '2';

// el constructor 'Number' obliga a la cadena comportarse como un número
console.log(foo + Number(bar)); // la consola de depuración muestra 3
```

El constructor Number, cuando es llamado como una función (como se muestra en el ejemplo) obliga a su argumento a comportarse como un número.

## c) Operadores Lógicos

Los operadores lógicos permiten evaluar una serie de operandos utilizando operaciones AND y OR.

### ✓ Operadores lógicos AND y OR

```
var foo = 1;
var bar = 0;
var baz = 2;

foo || bar; // devuelve 1, el cual es verdadero (true)
bar || foo; // devuelve 1, el cual es verdadero (true)
foo && bar; // devuelve 0, el cual es falso (false)
foo && baz; // devuelve 2, el cual es verdadero (true)
baz && foo; // devuelve 1, el cual es verdadero (true)
```

El operador || (OR lógico) devuelve el valor del primer operando, si éste es verdadero; caso contrario devuelve el segundo operando. Si ambos operandos son falsos devuelve falso (false). El operador && (AND lógico) devuelve el valor del primer operando si éste es falso; caso contrario devuelve el segundo operando. Cuando ambos valores son verdaderos devuelve verdadero (true), sino devuelve falso.

## d) Operadores de Comparación

Los operadores de comparación permiten comprobar si determinados valores son equivalentes o idénticos.



```

var foo = 1;
var bar = 0;
var baz = '1';
var bim = 2;
foo == bar; // devuelve falso (false)
foo != bar; // devuelve verdadero (true)
foo == baz; // devuelve verdadero (true); tenga cuidado
foo === baz; // devuelve falso (false)
foo !== baz; // devuelve verdadero (true)
foo === parseInt(baz); // devuelve verdadero (true)
foo > bim; // devuelve falso (false)
bim > baz; // devuelve verdadero (true)
foo <= baz; // devuelve verdadero (true) (págs. 8-10)

```

#### 2.2.8.2 Declaración Switch

“La declaración Switch evalúa el valor de una variable o expresión, y ejecuta diferentes bloques de código dependiendo de ese valor.” (Murphey, 2013, pág. 12)

#### 2.2.8.3 BUCLES

Los bucles (en inglés loops) permiten ejecutar un bloque de código un determinado número de veces.

Murphey (2013), considera los siguientes métodos de ejecutar un bucles:

##### ✓ Bucles Utilizando For

Un bucle utilizando for se compone de cuatro estados y posee la siguiente estructura:

```

for ([expresiónInicial]; [condición]; [incrementoDeLaExpresión])
[cuerpo]

```

El estado expresión Inicial es ejecutado una sola vez, antes que el bucle comience. Éste otorga la oportunidad de preparar o declarar variables. El estado condición es ejecutado antes de cada repetición, y retorna un valor que decide si el bucle debe continuar ejecutándose o no. Si el estado condicional evalúa un valor falso el bucle se detiene. El estado *incrementoDeLaExpresión* es ejecutado al final de cada repetición y otorga la oportunidad de cambiar el estado de importantes variables. Por lo general, este estado implica la incrementación o decrementación de un contador.

##### ✓ Bucles Utilizando While

Un bucle utilizando while es similar a una declaración condicional if, excepto que el cuerpo va a continuar ejecutándose hasta que la condición a evaluar sea falsa.

```

while ([condición]) [cuerpo]

```

Bucle utilizando while con la combinación de la condición y la incrementación

```
var i = -1;
while (++i < 100) {
  // Este bloque de código se ejecutará 100 veces
  console.log('Actualmente en ' + i);
}
```

Se comienza en -1 y luego se utiliza la incrementación previa (++i).

#### ✓ **Bucles Utilizando Do-while**

Este bucle es exactamente igual que el bucle utilizando while excepto que el cuerpo es ejecutado al menos una vez antes que la condición sea evaluada.

do [cuerpo] while ([condición])

```
Un bucle utilizando do-while
do {
  // Incluso cuando la condición sea falsa
  // el cuerpo del bucle se ejecutará al menos una vez.
  alert('Hola');
} while (false);
```

Este tipo de bucles son bastantes atípicos ya que en pocas ocasiones se necesita un bucle que se ejecute al menos una vez. De cualquier forma debe estar al tanto de ellos.

#### ✓ **Break y Continue**

Usualmente, el fin de la ejecución de un bucle resultará cuando la condición no siga evaluando un valor verdadero, sin embargo también es posible parar un bucle utilizando la declaración break dentro del cuerpo.

Detener un bucle con break

```
for (var i = 0; i < 10; i++) {
  if (something) {
    break;
  }
} (págs. 13-15)
```

### **2.2.8.4 Incorporando Javascript**

Existen técnicas para incorporar código Javascript dentro de HTML. Sin embargo, al igual que en CSS, solo la inclusión de archivos externos es la recomendada a usar en HTML5.

Para Gauchat (2013) se puede incorporar JavaScript haciendo uso de las siguientes técnicas:

### ✓ En línea

Esta es una técnica simple para insertar Javascript en nuestro documento que se aprovecha de atributos disponibles en elementos HTML. Estos atributos son manejados usando el manejador de eventos onclick, un código es ejecutado cada vez que el usuario hace clic con el ratón sobre el texto. Lo que el manejador onclick está diciendo es algo como: “cuando alguien haga clic sobre este elemento ejecute este código” y el código en este caso es una función predefinida en Javascript.

### ✓ Embebido

Para trabajar con códigos extensos y funciones personalizadas debemos agrupar los códigos en un mismo lugar entre etiquetas `<script>`. El elemento `<script>` actúa exactamente igual al elemento `<style>` usado para incorporar estilos CSS. Nos ayuda a organizar el código en un solo lugar, afectando a los elementos HTML por medio de referencias. Del mismo modo que con el elemento `<style>`, en HTML5 no debemos usar ningún atributo para especificar lenguaje. Ya no es necesario incluir el atributo `type` en la etiqueta `<script>`. El elemento `<script>` y su contenido pueden ser posicionados en cualquier lugar del documento, dentro de otros elementos o entre ellos.

Actualmente existen tres métodos disponibles para referenciar elementos HTML desde Javascript:

- **getElementsByTagName** referencia un elemento por su nombre o palabra clave.
- **getElementById** referencia un elemento por el valor de su atributo `id`.
- **getElementsByClassName** es una nueva incorporación que nos permite referenciar un elemento por el valor de su atributo `class`. (págs. 85-87)

## 2.2.9 AJAX

### 2.2.9.1 Introducción

Murphey (2013) considera que:

El método XMLHttpRequest (XHR) permite a los navegadores comunicarse con el servidor sin la necesidad de recargar la página. Este método, también conocido como Ajax (Asynchronous JavaScript and XML), permite la creación de aplicaciones ricas en interactividad. Las peticiones Ajax son ejecutadas por el código JavaScript, el cual envía una petición a una URL y cuando recibe una respuesta, una función de devolución puede ser ejecutada la cual recibe como argumento la respuesta del servidor y realiza algo con ella. Debido a que la respuesta es asíncrona, el resto del código de la aplicación continua ejecutándose, por lo cual, es imperativo que una función de devolución sea ejecutada para manejar la respuesta. A través de varios métodos, jQuery provee soporte para Ajax, permitiendo abstraer las diferencias que pueden existir entre navegadores. A pesar que la definición de Ajax posee la palabra “XML”, la mayoría de las aplicaciones no utilizan dicho formato para el transporte de datos, sino que en su lugar se utiliza HTML plano o información en formato JSON (JavaScript Object Notation). En general, Ajax no trabaja a través de dominios diferentes. Sin embargo, existen excepciones, como los servicios que proveen información en formato JSONP (JSON with Padding), los cuales permiten una funcionalidad limitada a través de diferentes dominios. (pág. 55)

### 2.2.9.2 Conceptos claves de Ajax

La utilización correcta de los métodos Ajax requiere primero la comprensión de algunos conceptos clave.

Murphey (2013) explica lo siguiente:

#### a) GET vs. POST

El método **GET** debe ser utilizado para operaciones no destructivas es decir, operaciones en donde se está “obteniendo” datos del servidor, pero no modificando. Por ejemplo, una consulta a un servicio de búsqueda podría ser una petición GET. Por otro lado, las solicitudes GET pueden ser almacenadas en la cache del navegador, pudiendo conducir a un comportamiento impredecible si no se lo espera. Generalmente, la información enviada al servidor, es enviada en una cadena de datos.

El método **POST** debe ser utilizado para operaciones destructivas, es decir, operaciones en donde se está incorporando información al servidor. Por ejemplo, cuando un usuario guarda un artículo en un blog, esta acción debería utilizar POST. Por otro lado, este tipo de método no se guarda en la cache del navegador. Además, una cadena de datos puede ser parte de la URL, pero la información tiende a ser enviada de forma separada. (págs. 55-56)

#### b) Tipos de Datos

Generalmente, JQuery necesita algunas instrucciones sobre el tipo de información que se espera recibir cuando se realiza una petición Ajax. En algunos casos, el tipo de dato es especificado por el nombre del método, pero en otros casos se lo debe detallar como parte de la configuración del método:

- ✓ **Text:** Para el transporte de cadenas de caracteres simples.
- ✓ **Html:** Para el transporte de bloques de código HTML que serán ubicados en la página.
- ✓ **Script:** Para añadir un nuevo script con código JavaScript a la página.
- ✓ **Json:** Para transportar información en formato JSON, el cual puede incluir cadenas de caracteres, vectores y objetos.

Es recomendable utilizar los mecanismos que posea el lenguaje del lado de servidor para la generación de información en formato JSON.

- ✓ **Jsonp:** Para transportar información JSON de un dominio a otro.
- ✓ **Xml:** Para transportar información en formato XML.

A pesar de los diferentes tipos de datos de que se puede utilizar, es recomendable utilizar el formato JSON, ya que es muy flexible, permitiendo por ejemplo, enviar al mismo tiempo información plana y HTML. (pág. 57)

Ajax es una técnica de desarrollo web interactivas, estas aplicaciones se ejecutan de lado del cliente por medio del navegador.

### 2.2.9.3 Métodos Ajax de JQuery

Murphey (2013) explica lo siguiente:

El método \$.ajax es configurado a través de un objeto, el cual contiene todas las instrucciones que necesita jQuery para completar la petición. Dicho método es particularmente útil debido a que ofrece la posibilidad de especificar acciones en caso que la petición haya fallado o no. Además, al estar configurado a través de un objeto, es posible definir sus propiedades de forma separada, haciendo que sea más fácil la reutilización del código.

- **Opciones del método \$.ajax**

El método \$.ajax posee muchas opciones de configuración, y es justamente esta característica la que hace que sea un método muy útil. A continuación se muestran las más comunes:

- ✓ **async:** Establece si la petición será asíncrona o no. De forma predeterminada el valor es true. Debe tener en cuenta que si la opción se establece en false, la petición bloqueará la ejecución de otros códigos hasta que dicha petición haya finalizado.
- ✓ **cache:** Establece si la petición será guardada en la cache del navegador. De forma predeterminada es true para todos los dataType excepto para "script" y "jsonp".
- ✓ **complete:** Establece una función de devolución de llamada que se ejecuta cuando la petición está completa, aunque haya fallado o no.
- ✓ **context:** Establece el alcance en que la/las funciones de devolución de llamada se ejecutaran (por ejemplo, define el significado de this dentro de las funciones). De manera predeterminada this hace referencia al objeto originalmente pasado al método \$.ajax.
- ✓ **data:** Establece la información que se enviará al servidor. Esta puede ser tanto un objeto como una cadena de datos.
- ✓ **dataType:** Establece el tipo de información que se espera recibir como respuesta del servidor. Si no se especifica ningún valor, de forma predeterminada, jQuery revisa el tipo de MIME que posee la respuesta
- ✓ **error:** Establece una función de devolución de llamada a ejecutar si resulta algún error en la petición.
- ✓ **jsonp:** Establece el nombre de la función de devolución de llamada a enviar cuando se realiza una petición JSONP. De forma predeterminada el nombre es "callback success Establece una función a ejecutar si la petición ha sido satisfactoria.
- ✓ **timeout:** Establece un tiempo en milisegundos para considerar a una petición como fallada.
- ✓ **traditional:** Si su valor es true, se utiliza el estilo de serialización de datos utilizado antes de JQuery 1.4.

- ✓ **type:** De forma predeterminada su valor es “GET”. Otros tipos de peticiones también pueden ser utilizadas, sin embargo pueden no estar soportados por todos los navegadores.
- ✓ **url:** Establece la URL en donde se realiza la petición. (págs. 58-59)

## 2.2.10 JQUERY

### 2.2.10.1 Concepto JQuery

“Es un framework JavaScript, que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales.” (Álvarez, 2012)

Jquery es un conjunto de librerías, es de código libre, ofrece varias funcionalidades basadas a javascript, permitiendo acceder a objetos DOM.

### 2.2.10.2 Core de jQuery

Álvarez (2012) explica que:

El core de JQuery es la base sobre la que se trabaja para hacer cualquier cosa con jQuery. Contiene una serie de clases y métodos útiles para hacer tareas reiterativas, que vamos a necesitar en las aplicaciones. Integra desde funciones que serán útiles en cualquier script, por sencillo que sea, hasta funciones menos recurridas pero que nos facilitarán la vida a la hora de hacer código limpio, corto y reutilizable.

Core tiene las funciones clasificadas en diversos apartados:

- ✓ **\$() (La función JQuery):**

Es la función principal de JQuery, que además tiene diversas utilidades según los parámetros que se le envíen. Su utilidad principal es obtener elementos de la página.

- ✓ **Accesorios de objetos:**

Es una gama de funciones de diversa y variada utilidad, que sirven de utilidad para hacer cosas con objetos, tales como iterar con cada uno de sus elementos, saber su tamaño, longitud, el selector o contexto con el que se obtuvo, obtener todos sus elementos DOM que contenga, etc.

- ✓ **Trabajo con datos:**

Unas funciones útiles para trabajar con datos y asociarlos a elementos, una forma de guardar información adicional a elementos de la página. También tiene diversas funciones para trabajar con colas y administrar la salida y entrada de sus elementos.

- ✓ **Plugins:**

Funciones que permiten extender los elementos JQuery para incorporar nuevos métodos, algo que se utiliza habitualmente a la hora de crear plugins para añadir funcionalidades a JQuery.

✓ **Interoperabilidad:**

Funciones que permiten que JQuery no tenga problemas de compatibilidad con otras librerías JavaScript que también suelen utilizar la función dólar \$().

### 2.2.10.3 Función jQuery

Álvarez (2012) explica los parámetros que puede recibir una función JQuery:

**a) Pasando un HTML**

Una posibilidad de trabajar con la función jQuery es enviarle un string con un HTML. Esto crea esos elementos en la página y les coloca los contenidos que se indique en el string.”

**b) Pasando elementos**

Otro posible valor que se le puede enviar a la función JQuery es un elemento o una jerarquía de elementos del DOM, para extenderlos con las funcionalidades que aporta el framework para los elementos.

**c) Pasando una función**

Se envía como parámetro una función por ejemplo la función callback que se invoca automáticamente cuando el DOM está listo. Allí podemos colocar cualquier código, que se ejecutará sólo cuando el DOM está listo para recibir comandos que lo modifiquen.

### 2.2.10.4 Selección de Elementos

Murphey (2013), considera que:

El concepto más básico de jQuery es el de “seleccionar algunos elementos y realizar acciones con ellos”. La biblioteca soporta gran parte de los selectores CSS3 y varios más no estandarizados. A continuación se muestran algunas técnicas comunes para la selección de elementos:

- **Selección de elementos en base a su ID**

`$('#myId');` // notar que los IDs deben ser únicos por página

- **Selección de elementos en base al nombre de clase**

`$('div.myClass');` // si se especifica el tipo de elemento,  
// se mejora el rendimiento de la selección

- **Selección de elementos por su atributo**

`$('input[name=first_name]');` // tenga cuidado, que puede ser muy lento

- **Selección de elementos en forma de selector CSS**

`$('#contents ul.people li');` (pág. 27)

### 2.2.10.5 Elección de Selectores

“La elección de selectores es un punto importante cuando se desea mejorar el rendimiento del código. JQuery ofrece muchos selectores basados en atributos, que permiten realizar selecciones basadas en el contenido de los atributos utilizando simplificaciones de expresiones regulares.” (Murphey, 2013, pág. 29)

### 2.2.10.6 Refinamiento y Filtrado de Selecciones

Murphey (2013) explica que:

A veces, puede obtener una selección que contiene más de lo que necesita; en este caso, es necesario refinar dicha selección. JQuery ofrece varios métodos para poder obtener exactamente lo que desea.

#### a) Refinamiento de selecciones

- `$(div.foo').has('p');` // el elemento div.foo contiene elementos `<p>`
- `$(h1').not('.bar');` // el elemento h1 no posee la clase 'bar'
- `$(ul li').filter('.current');` // un item de una lista desordenada
- `// que posee la clase 'current'`
- `$(ul li').first();` // el primer item de una lista desordenada
- `$(ul li').eq(5);` // el sexto item de una lista desordenada

#### b) Selección de Elementos de un Formulario

JQuery ofrece varios pseudo-selectores que ayudan a encontrar elementos dentro de los formularios, éstos son especialmente útiles ya que dependiendo de los estados de cada elemento o su tipo, puede ser difícil distinguirlos utilizando selectores CSS estándar.

- `:button` Selecciona elementos `<button>` y con el atributo `type='button'`
- `:checkbox` Selecciona elementos `<input>` con el atributo `type='checkbox'`
- `:checked` Selecciona elementos `<input>` del tipo checkbox seleccionados
- `:disabled` Selecciona elementos del formulario que están deshabilitados
- `:enabled` Selecciona elementos del formulario que están habilitados
- `:file` Selecciona elementos `<input>` con el atributo `type='file'`
- `:image` Selecciona elementos `<input>` con el atributo `type='image'`
- `:input` Selecciona elementos `<input>`, `<textarea>` y `<select>`
- `:password` Selecciona elementos `<input>` con el atributo `type='password'`
- `:radio` Selecciona elementos `<input>` con el atributo `type='radio'`
- `:reset` Selecciona elementos `<input>` con el atributo `type='reset'`
- `:selected` Selecciona elementos `<options>` que están seleccionados
- `:submit` Selecciona elementos `<input>` con el atributo `type='submit'`
- `:text` Selecciona elementos `<input>` con el atributo `type='text'` (págs. 30-31)

### 2.2.11 ¿QUÉ ES BOOTSTRAP?

“Es una colección de herramientas para crear sitios y aplicaciones web, contiene HTML y CSS basados en plantillas de diseño para el texto, formas, botones, navegación y otros componentes,



también contiene extensiones de JavaScript opcionales, es de código abierto y está disponible en GitHub.” (Refsnes Data Inc, 2011)

Es una herramienta Open Source para el desarrollo rápido de aplicaciones web, que permiten crear interfaces web con CSS y Javascript que se adaptan dependiendo del tamaño del dispositivo en el que se ejecute la aplicación, creando un diseño adaptivo o responsive design.

#### 2.2.11.1 Bootstrap Historia

“Bootstrap fue desarrollado por Mark Otto y Jacob Thornton como un marco para fomentar la coherencia entre los instrumentos internos. Antes de Bootstrap, se utilizaron varias bibliotecas para el desarrollo de la interfaz, lo que llevó a inconsistencias y una alta carga de mantenimiento. Bootstrap fue lanzado como un producto de código abierto en agosto de 2011 en GitHub. En junio 2014 Bootstrap fue el proyecto No.1 en GitHub!” (Refsnes Data Inc, 2011)

#### 2.2.11.2 ¿Por qué utilizar Bootstrap?

Refsnes Data Inc (2011) considera lo siguiente:

- ✓ **Enfoque mobile-first:** Desde Bootstrap 3, el framework consiste en estilos mobile-first a lo largo de toda la biblioteca
- ✓ **Compatibilidad con navegadores:** Manos a la Obra es compatible con todos los navegadores populares
- ✓ **Diseño web Responsive:** CSS sensible de Manos a la Obra se ajusta a ordenadores de sobremesa, tabletas y teléfonos móviles. El diseño web responsive es crear sitios web que se ajustan y alinea de acuerdo a la herramienta de comunicación que utiliza para la visualización de la misma.
- ✓ **Fácil para empezar:** Con sólo el conocimiento de HTML y CSS cualquiera puede empezar a trabajar con Bootstrap.

#### 2.2.11.3 ¿Qué incluye Bootstrap?

Refsnes Data Inc (2011) afirma que:

- ✓ **Scaffolding:** Contiene el diseño, sistema de red, sistema de red fluida, y el diseño de respuesta.
- ✓ **CSS Base:** Contiene clases de tipografía, tablas, formularios, botones, imágenes, y mucho más.
- ✓ **Componentes:** Contiene componentes reutilizables: iconos, menús desplegables, NavBars, pan rallado, alertas y más.
- ✓ **JavaScript Plugins:** Contiene más de una docena personalizados plugins JQuery. Puede incluir a todos, o uno a uno.
- ✓ **Componentes personalizables:** Personalice componentes de Bootstrap, variables menos, y plugins JQuery para crear su propia versión.

## 2.2.12 IBM PERSONAL COMMUNICATIONS

IBM Corp. (2003) afirma que:

IBM Personal Communications es un paquete de emulación de terminal y comunicación con host para Microsoft Windows que presenta emulación de terminal virtual (VT), soporte de aplicaciones SNA (arquitectura de redes de sistemas), integración y conectividad. Personal Communications es un componente clave de IBM Host Access Client Package e IBM Rational Host Integration Solution.

Personal Communications ofrece una plataforma de acceso tradicional a los datos y a las aplicaciones en distintos sistemas host.

- ✓ **Ofrece una solución para el acceso de host** a los sistemas empresariales de misión crítica.
- ✓ **Garantiza una red más segura** para la transmisión de datos importantes.
- ✓ **Proporciona soporte de desarrollo** de flexibilidad y ampliaciones añadidas de inversiones en redes.
- ✓ **Impulsa la productividad del usuario** gracias a programas de utilidades que permiten el ahorro de tiempo. (pág. 14)

### 2.2.12.1 Host Access Class Automation Objects

IBM Corp. (2003) explica que:

El anfitrión Clase Acceso Biblioteca Automatización objetos permiten al personal Comunicaciones de productos de apoyo a la tecnología de automatización basada en COM Microsoft (anteriormente conocida como la automatización OLE). Los objetos ECL automatización son una serie de servidores de automatización que permiten a los controladores de automatización, por ejemplo, Microsoft Visual Básico, para acceder a los datos y la funcionalidad de programación de comunicaciones personales. Un ejemplo de esto sería el envío de claves para las comunicaciones personales espacio de presentación. Esto se puede lograr escribiendo manualmente las claves en la Ventana de Comunicaciones Personal, pero puede también ser automatizado a través del servidor adecuado de comunicaciones personales de automatización (autECLPS en este caso). Uso de Visual Basic puede crear el objeto autECLPS y luego llamar a los SendKeys método en el que objeto con la cadena que es para ser colocado en el espacio de presentación. En otras palabras, las aplicaciones que están habilitados para el control de la automatización protocolo (controlador de automatización) pueden controlar algunos de Comunicaciones Personales operaciones (servidor de automatización).

Personal Communications ofrece varios servidores de automatización para lograr esto. Estos servidores se implementan como en el mundo real, los objetos intuitivos con los métodos y propiedades que controlan la operatividad de comunicaciones personales. (pág. 15)



**Figura 9.** Host Access Class Library Automation Objects

**Fuente:** Host Access Class Library

Se describen los métodos y las propiedades de cada objeto en detalle y es destinada a cubrir todos los posibles usuarios del objeto de automatización. Debido a que la mayor parte forma común de utilizar el objeto es a través de una aplicación de scripting tales como Visual Básicos, todos los ejemplos se muestran utilizando un formato de Visual Basic.

#### ✓ **autSystem Clase**

La clase autSystem proporciona dos funciones de utilidad que pueden ser útiles para su uso con algunos lenguajes de programación.

#### ✓ **autECLConnList Clase**

autECLConnList contiene información sobre las conexiones de todos los iniciados. Su nombre en el registro es PCOMM.autECLConnList.

El objeto autECLConnList contiene una recopilación de información acerca de las conexiones a un host. Cada elemento de la colección representa una única conexión (emulador ventana). Una conexión en esta lista puede estar en cualquier estado (por ejemplo, parado o desconectado). Todas las conexiones iniciadas aparecen en esta lista. El elemento de la lista contiene el estado de la conexión. Un objeto autECLConnList ofrece una instantánea estática de las conexiones actuales. La lista no se actualiza dinámicamente a medida que las conexiones se inician y detienen. El Método Refresh se llama automáticamente a la construcción de la autECLConnList objeto. Si utiliza el objeto autECLConnList justo después de su construcción, su lista de conexiones es actual. Sin embargo, usted debe llamar al método Refresh en el autECLConnList objeto antes de acceder a sus otros métodos si ha pasado algún tiempo desde su construcción para asegurarse de que usted tiene los datos actuales. Una vez que haya llamado Refrescar usted puede empezar a caminar a través de la colección. (pág. 16)

## 2.2.13 METODOLOGÍA UWE

### 2.2.13.1 Concepto

Koch, Knapp, Zhang, y Baumeister (2008), afirman que:

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. (pág. 157)

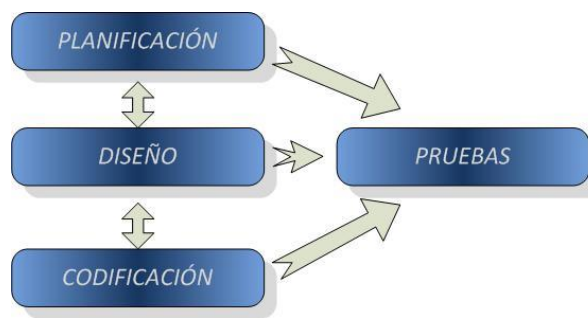
### 2.2.13.2 Fases de la Metodología Uwe.

Ludwing (2013) describe que:

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

**Las fases o etapas a utilizar son:**

- 1) Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.
- 2) Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.
- 3) Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- 4) Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- 5) La Instalación o Fase de Implementación:** es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.
- 6) El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.



**Figura 10.** Unificación de las fases de la Metodología UWE

*Fuente:* <http://uwe.pst.ifi.lmu.de/>

#### **2.2.14 ACTIVIDADES DE MODELADO DE UWE.**

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación.

##### **2.2.14.1 El modelo que propone UWE está compuesto por etapas o sub-modelos:**

Ludwing (2013) considera que:

- ✓ **Modelo Lógico-Conceptual.**

UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web.

- ✓ **Modelo de Navegación**

Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionaran.

- ✓ **Modelo de presentación**

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

## BIBLIOGRAFÍA

- Álvarez, M. (2012). <http://www.desarrolloweb.com/>. Obtenido de <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
- Bahit, E. (2012). *Python para principiantes*. Autoedición.
- Fernández Montoro, A. (2014). *Python3 al descubierto*. España: RC Libros.
- Gauchat, J. (2013). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.
- Gibert Ginestà, M., & Pérez Mora, O. (2012). <http://ocw.uoc.edu/>. Obtenido de [http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02152.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf)
- González Duque, R. (2010). *Python para todos*. España: Autoedición.
- Gutiérrez, J. (s.f). <http://www.lsi.us.es/>. Obtenido de [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)
- Holovaty, A., & Kaplan-Moss, J. (2010). *Django 1.0*. Autoedición.
- IBM. (2003). *Host Access Class Library*.
- Koch, N., Knapp, A., Zhang, G., & Baumeister, H. (2008). *Web Engineering: Modelling and Implementing Web Applications*. London: Springer.
- Ludwing, M. (2013). <http://uwe.pst.ifi.lmu.de/>. Obtenido de <http://uwe.pst.ifi.lmu.de/teachingTutorialSpanish.html>
- Luján, S. (2012). *Programación de aplicaciones web*. Club Universitario.
- Murphey, R. (2013). *Fundamentos de JQuery*. OpenLibra.
- Pavón Maestras, J. (2012). <http://www.fdi.ucm.es/>. Obtenido de <http://www.fdi.ucm.es/profesor/jpavon/web/11-Introduccion-TecnologiasWeb.pdf>
- PostgreSQL.org. (2 de 10 de 2010). <http://www.postgresql.org.es/>. Obtenido de [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

PostgreSQL-ES. (2 de Octubre de 2010). *PostgreSQL-ES*. Recuperado el 15 de Mayo de 2014, de [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

Refsnes Data Inc. (2011). <http://www.w3schools.com/>. Obtenido de [http://www.w3schools.com/bootstrap/bootstrap\\_intro.asp](http://www.w3schools.com/bootstrap/bootstrap_intro.asp)

Vega, J., & Vab Der Henst, C. (junio de 2011). <http://mlw.io/>. Obtenido de <http://mlw.io/guia-html5/>