

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), 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
Login de la Aplicación	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. Permite el ingreso a la aplicación mediante un usuario y una contraseña.
		Comprobación de la identidad del usuario ingresado	
		Ingreso a la aplicación	
		Comprobación de Sesiones del servidor	
		Comprobación de sesiones disponibles	
		Apertura de Sesión	
		Comprobación de acceso a contrato	
Ingreso	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	

Búsqueda de información al sistema comercial SICO	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	
Búsqueda de Actividades realizadas	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	
Monitoreo de Cuadrillas	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	
Avance de Obra	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.
Reportes	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	

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. (Coniel Cia.Ltda, 2013)

2.1.2 UBICACIÓN

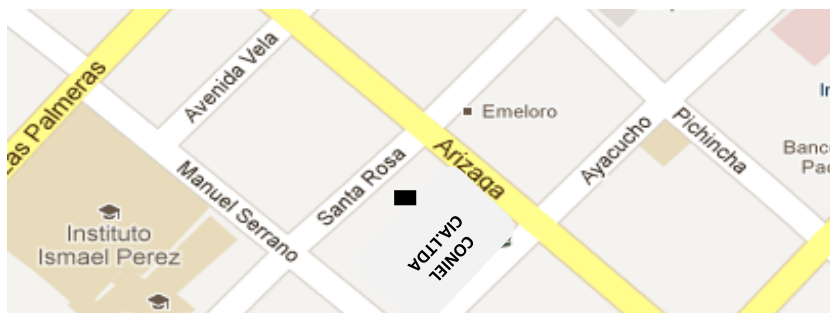


Figura 1. Croquis de ubicación de la Empresa Coniel Cía. Ltda.

Fuente: Coniel Cia.Ltda

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

2.1.3 ORGANIGRAMA ESTRUCTURAL

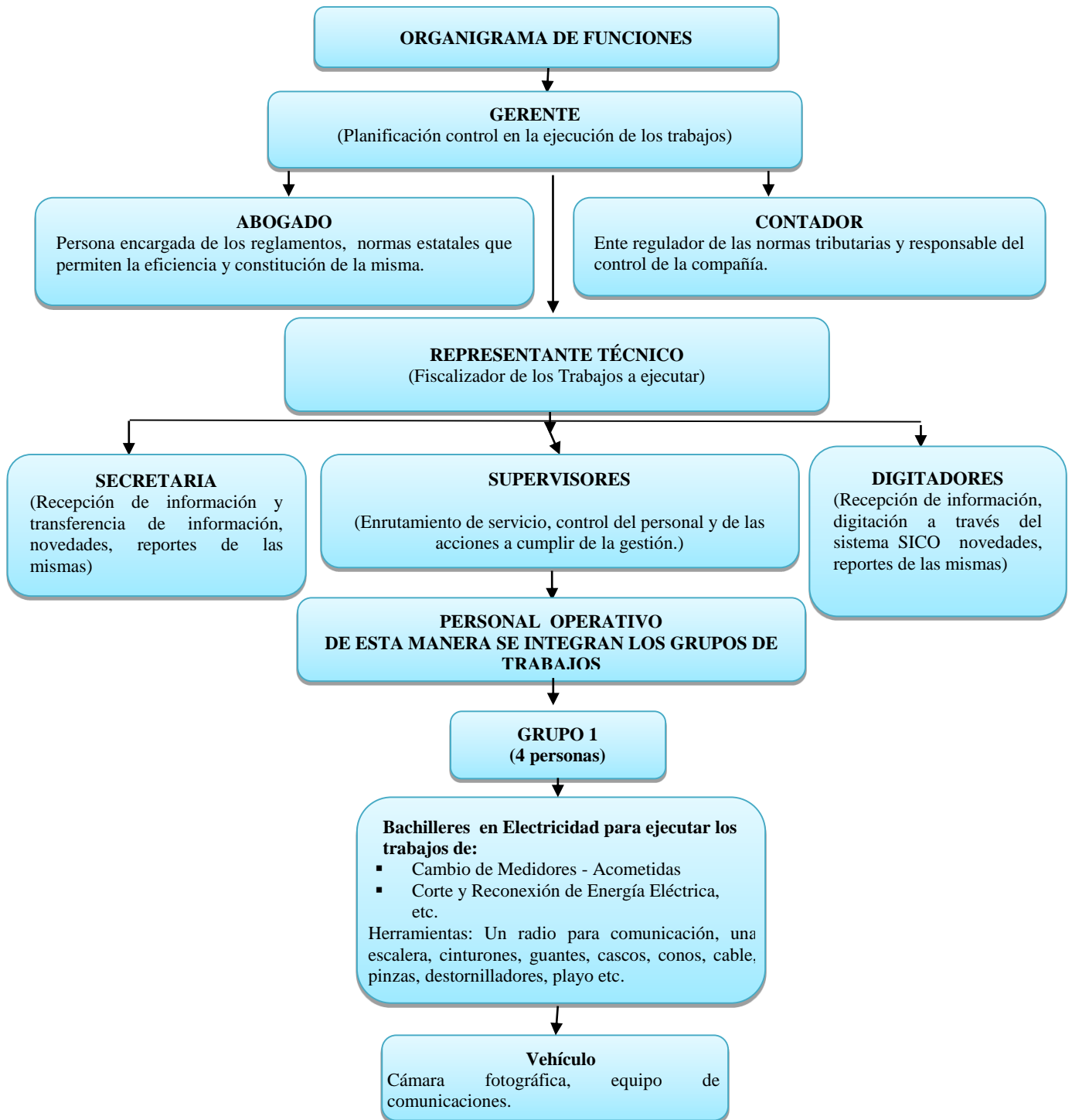


Figura 2. Estructura Organización de CONIEL Cia.Ltda.

Fuente: 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)

De acuerdo con Muñoz (2011), las aplicaciones web son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. Estas aplicaciones son desarrolladas en lenguajes que son soportados por los navegadores web ya que es sobre ellos que se ejecutaran. Además estas aplicaciones pueden ser ejecutadas independientemente del sistema operativo, ofrecen facilidad para actualizar, sin necesidad de instalar ningún software a los usuarios.

2.2.1.2 Ventajas

Según Pavón Maestras (2012) las ventajas de las aplicaciones web son:

- ✓ 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.

2.2.2 LENGUAJE DE PROGRAMACIÓN PYTHON

2.2.2.1 Concepto

González Duque (2010, pág. 7) afirma que a principio de los años 90, Guido Van Rossum creo un lenguaje de programación inspirado en el grupo de cómicos ingleses “Monty Python”. Python es un lenguaje de programación orientado a objetos, multiparadigma, multiplataforma, con estructuras de datos de alto nivel a través de un tipado dinámico. Brinda una sintaxis muy simple y limpia que ayuda a crear un código legible.

De acuerdo con Álvarez M. (2003), Python ofrece rapidez en el desarrollo de aplicaciones, es un lenguaje de programación scripting, que trabaja independiente de la plataforma, no se necesita compilar el código para que se ejecute por lo que se lo conoce como lenguaje interpretado.

2.2.2.2 Características

De acuerdo con González Duque (2010)

- ✓ Lenguaje interpretado o de script
- ✓ Tipado dinámico
- ✓ Fuertemente tipado
- ✓ Multiplataforma
- ✓ Orientado a objetos. (pág. 6)

Las características descritas hacen que Python sea un lenguaje de programación orientado a objetos que permite el desarrollo de aplicación más rápido y utilizando menos líneas de código.

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, pág. 30) , las variables ocupan un espacio en la memoria de un ordenador ya que almacenan datos que pueden ser modificados. El tipo de datos de la variable es definido por el valor que se le asigne, además se le debe dar un nombre el mismo que debe ser descriptivo y en minúsculas, si se llegase a utilizar dos palabras para el nombre se las puede separar por un guion bajo.

- Tipo de Datos

En el 2012 Bahit, define que los valores de una variable como los de una constante pueden pertenecer a un tipo de dato específico, por lo que explica los tipos más comunes:

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)

Un tipo de datos define el valor que contiene una variable o una constante, por lo que esto indica al ordenador que clase de datos va a procesar, además esto ayuda a imponer restricción sobre los valores que se pueden asignar a una variable o constante.

- 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: Bahit (2012), pág.32

Los operadores aritméticos toman valores numéricos (literales o variables) de las variables o constantes como sus operandos y devuelven un solo valor numérico.

2.2.2.4 Colecciones

González Duque (2010) explica los siguientes tipos de colecciones:

✓ **Listas**

Las listas pueden contener una colección de datos: números, cadenas, booleanos, listas, esto es conocido en otros lenguajes de programación como arrays o vectores. Para declarar una lista solo se debe ubicar entre corchetes los valores que se desea incluir en una lista. Ejemplo: l = [22, True, "una lista", [1, 2]]

✓ **Tuplas**

Una tupla es una lista inmutable, que no pueden modificarse sus valores de ningún modo después de creación. A diferencia de las listas una tupla se la declara entre (), en la que su constructor es la coma (.). Ejemplo: t = (1, 2, True, "python")

✓ **Diccionarios**

Son colecciones que relacionan una clave y un valor, por lo que también se los conoce como matrices asociativas. (págs. 24 - 26)

2.2.2.5 Control de Flujo

Fernández Montoro (2014, pág. 53), menciona que una estructura de control, es un bloque de código que permite agrupar instrucciones de manera controlada.

✓ If, else, y elif

Es una sentencia condicional, debido a que si se cumple una condición es decir si el resultado es “true” se ejecutara la sentencia que este en el if, o si el resultado es negativo “falso” se ejecutara la sentencia o sentencias que se encuentren dentro del else.

✓ For

Es un bucle repetitivo que ejecutara sentencias a cada elemento que contiene un objeto al que se le aplicara la sentencia for.

2.2.3 FRAMEWORK WEB

Es una estructura de 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)

De acuerdo con Sánchez (2006), un framework es un patrón o esquema que sirve para el desarrollo de aplicaciones, además de que ayudan a acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

2.2.3.1 Patrón MVC

Gutiérrez (s.f, pág. 2), describe que el patrón organiza las aplicaciones en tres modelos separados, es decir el primer modelo que representa los datos de la aplicación y las reglas de negocio, el segundo seria el conjunto de vistas que representa los formularios de entrada además de la salida de información, el tercero es un conjunto de controladores que procesan las peticiones de los usuarios y controlan el flujo de ejecución del sistema.

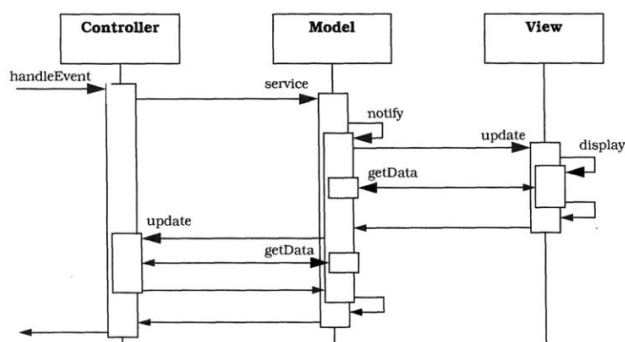


Figura 3. Modelo Vista Controlador

Fuente: Gutiérrez (s.f)

2.2.4 DJANGO

Es un framework que permite al desarrollador enfocarse en la parte creativa de las aplicaciones, evitando que se realicen operaciones repetitivas, favorece en el ahorro de tiempo, además de permitir crear y mantener aplicaciones de alta calidad con menos código. Por lo que ofrece opciones claras de cómo resolver un problema, provee un nivel de abstracción de patrones en el desarrollo, además de atajos para tareas frecuentes en la programación. (Holovaty & Kaplan-Moss, 2010, pág. 21)

Django permite que el desarrollador escriba código ágilmente. El resultado son menos líneas de código y, en consecuencia, menos probabilidades de que haya bugs. (Rodríguez Guerra, 2014)

2.2.4.1 Historia de Django

El grupo de trabajo de The World Online, responsable del desarrollo y mantenimiento de varios sitios locales, según Holovaty & Kaplan-Moss (2010, pág. 23), deseaba un entorno de desarrollo dictado por las fechas límites del periodismo. Para estos sitios se exigía que se agregaran nuevas características y que aplicaciones completas se crearan a una velocidad vertiginosa, muchas de las veces solo a horas de preaviso. Viendo esta necesidad es que Adrián y Simón desarrollaron un framework de desarrollo web que les permita ahorrar tiempo al crear aplicaciones mantenibles. Por el año 2005, después de haber desarrollado este framework hasta el punto en que hacía funcionar a la mayoría de sitios World Online, el equipo que ahora incluía a Jacob Kaplan-Moss, decidió en julio del 2005 liberar el framework como software de código abierto, al que lo llamaron Django.

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 URL conf a una función de vista que retorna un `HttpResponse` puede ser corto-circuitado o *augmented* mediante middleware.

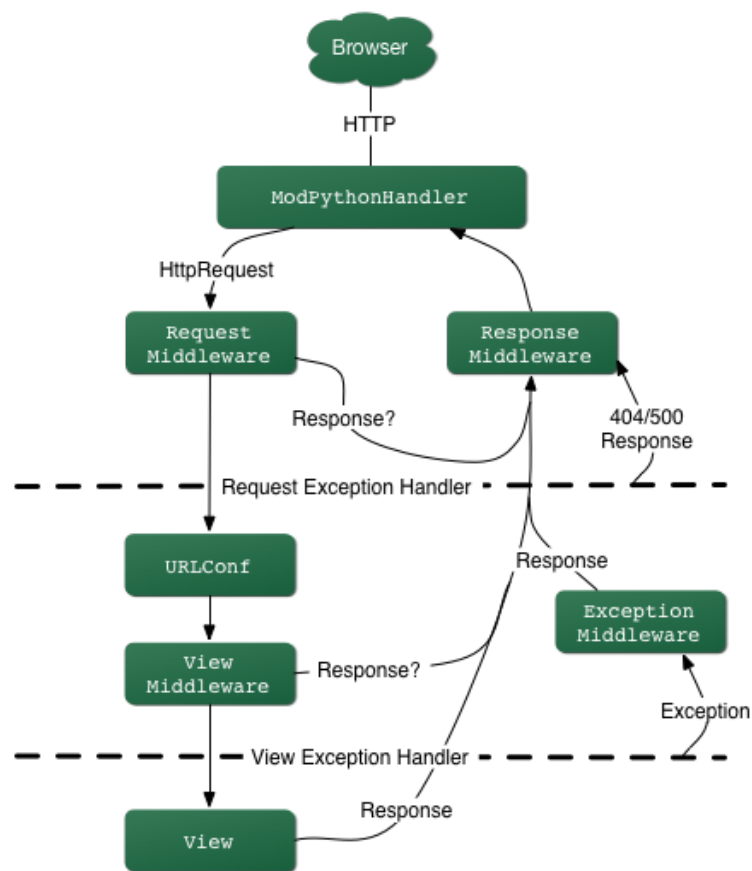


Figura 4. Flujo de una petición y respuesta a Django

Fuente: Holovaty & Kaplan-Moss (2010)

El navegador envía una petición HTTP, por lo que se construye la HttpRequest a través de un manejador específico que nos ayudaran a controlar el flujo del procesamiento de la respuesta y pasar los componentes. 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.

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)

2.2.5 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. (Martínez, PostgreSQL-ES, 2010)

De acuerdo con Denzer (2002), PostgreSQL es un sistema gestor de base de datos relacionales avanzado basado en código abierto, lo que quiere decir que el código fuente de este programa está disponible a cualquier persona, está bajo licencia BSD. Permite la manipulación bajo las reglas del álgebra relacional, las tablas se pueden relacionar unas a otras haciendo uso de llaves, además los datos se almacenan en tablas conformadas por columnas y renglones.

2.2.5.1 Componentes de un sistema PostgreSQL

Martínez (2010) explica lo componentes:

- ✓ **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos.
- ✓ **Demonio postmaster:** Escucha por un puerto/socket por conexiones entrantes de clientes.
- ✓ **Ficheros de configuración:** Los utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- ✓ **Procesos hijos postgres:** Se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a los 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.

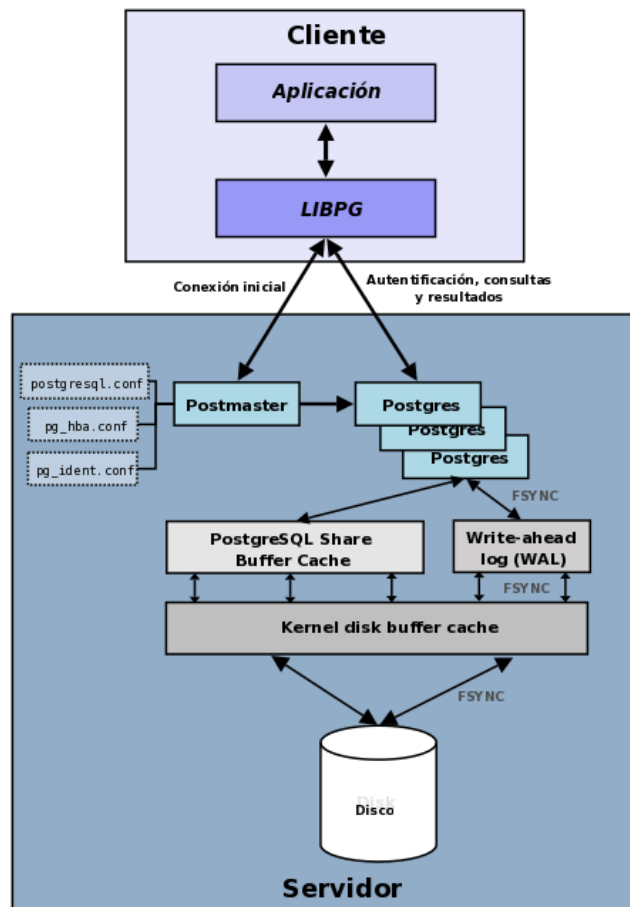


Figura 5. Componentes de un Sistema PostgreSQL

Fuente: Martínez (2010)

2.2.5.2 Arquitectura de PostgreSQL

Según Gibert Ginestà y Pérez Mora (2012), 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). Un proceso servidor *postgres* puede atender exclusivamente a un solo cliente; es decir, hacen falta tantos procesos servidor *postgres* como clientes haya. Cuando los clientes realicen una petición de conexión el encargado de ejecutar un servidor para cada uno es el proceso **postmaster**. El proceso postmaster se ejecuta en un sitio (*host*), pero solo se puede ejecutar un proceso postmaster y varios procesos postgres. (pág. 65)

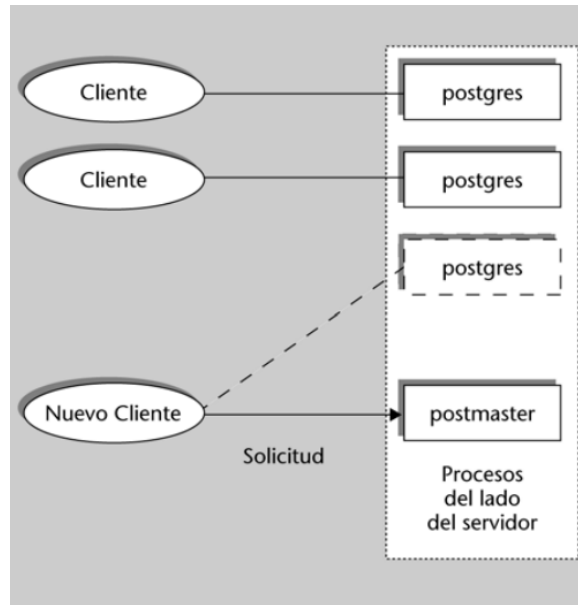


Figura 6. Arquitectura de PostgreSQL

Fuente: Gibert Ginestà & Pérez Mora (2012, pág. 65)

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)

De acuerdo con Pavan (2013), Html5 es un lenguaje markup usado para estructurar y presentar el contenido para la web. Es usado para definir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes de una aplicación para la web. Este lenguaje se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>), también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

2.2.6.1 Componentes Básicos

Gauchat (2013) afirma que html5 provee una estructura con los elementos necesarios para ubicar el contenido ya sea este estático o dinámico. Es la combinación de tres tecnologías Html para la estructura del documento, CSS se encarga de presentar esa estructura y su contenido en la pantalla y JavaScript se encarga de hacer el resto, son altamente dependientes y actúan como una sola unidad organizada bajo las especificaciones de html5. Por lo que Html5 provee tres características: estructura, estilo y funcionalidad. (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>**
Esta etiqueta permite definir el tipo de documento que estamos creando, para que de esta forma el navegador se active para que pueda interpretar las incorporaciones de HTML5 siempre que sea necesario.
- ✓ **<html>**
La estructura tipo árbol de este lenguaje tiene su raíz en el elemento <html>.
- ✓ **<head>**
El elemento <head> va primero porque define la sección cabecera del documento, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre.
- ✓ **<body>**
El cuerpo representa la parte visible de todo documento y esta sección es especificada entre etiquetas <body>.
- ✓ **<meta>**
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.
- ✓ **<link>**
Este elemento es usado para incorporar estilos, códigos JavaScript, imágenes o iconos desde archivos externos. (págs. 19-22)

2.2.6.3 Estructura del cuerpo del documento HTML5

La estructura del cuerpo es decir las que se encuentran dentro de la etiqueta <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. (Gauchat, 2013, pág. 24)

Gauchat (2013) describe los siguientes elementos:

- ✓ **<header>**
Se encargan de indicar a los navegadores que lo que contienen dentro de <header> es la cabecera de la página.
- ✓ **<nav>**
Nav está diseñado para que ahí coloques la botonera de navegación principal.
- ✓ **<section>**
Define un área de contenido única dentro del sitio.

✓ **<aside>**

Cualquier contenido que no esté relacionado con el objetivo primario de la página va en un aside o barra lateral de información.

✓ **<footer>**

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

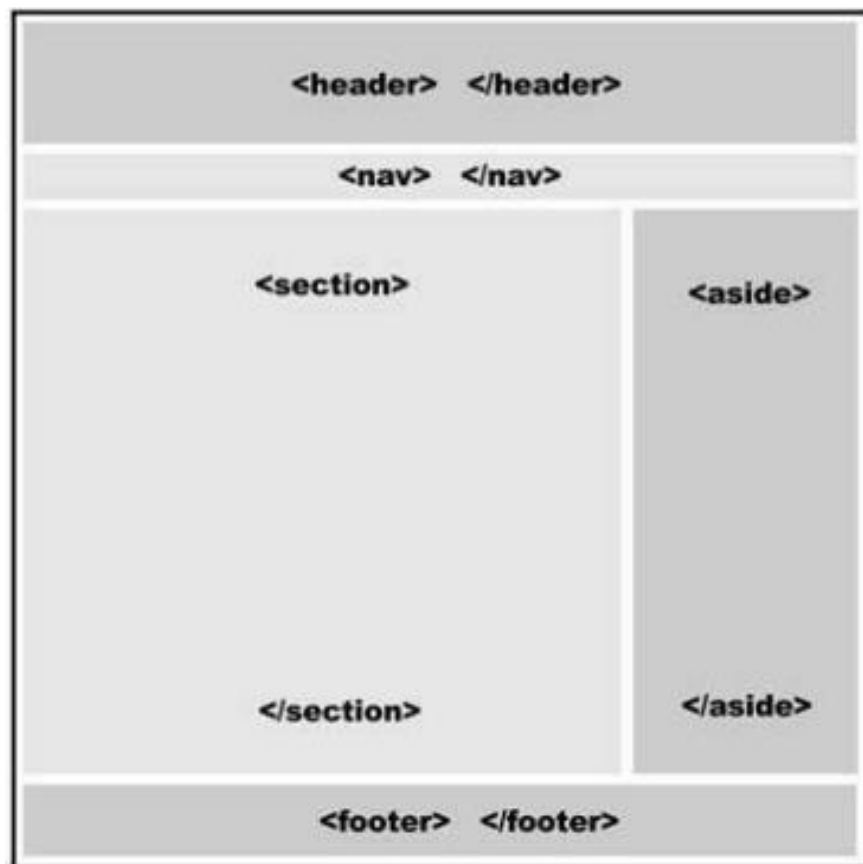


Figura 7. Representación visual de un diseño utilizando elementos HTML5.

Fuente: Gauchat (2013, pág. 24)

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.

✓ **<hgroup>**

Permite colocar un h1, h2 y h3 dentro del header sin afectar el contenido, permitiendo usar otro h1 en el sitio.

✓ **<figure> y <figcaption>**

Fue creada para ayudarnos a ser aún más específicos a la hora de declarar el contenido del documento. (págs. 33-35)

2.2.7 INTRODUCCIÓN A CSS3

De acuerdo con Gauchat (2013, pág. 42), es una tecnología que permite crear estilos para páginas web, es la mejor forma de separar el contenido de su presentación y sirve para crear páginas web complejas. Es un lenguaje de hojas de estilos que fueron creados para definir el aspecto o la presentación de los documentos.

Según Wium Lie & Bos (2005), las hojas de estilo en cascada (CSS) es un lenguaje de estilo de hojas usado para describir la presentación de las páginas web, que permite la separación del contenido del documento de la presentación del documento (disposición, colores, fuentes, etcétera).

2.2.7.1 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. (págs. 45-46).

2.2.7.2 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.

✓ **box-shadow:**

Esta propiedad crea sombras para la caja formada por el elemento.

✓ **text-shadow**

Esta propiedad es similar a box-shadow pero específica para textos, es decir crea sombra de texto.

✓ **@font-face**

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

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

Se aplicada a las propiedades background o background-image para generar un gradiente lineal.

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

Se aplicada a las propiedades background o background-image para generar un gradiente radial.

✓ **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**

Se combina con la propiedad outline-offset para generar un segundo borde alejado del borde original del elemento.

✓ **border-image**

Crea un borde con una imagen personalizada. Necesita que el borde sea declarado previamente con las propiedades border o border-width.

✓ **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. (págs. 62-66)

2.2.8 JAVASCRIPT

Es un lenguaje interpretado usado para múltiples propósitos. Según Gauchat (2013, pág. 84) 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.

Según Cárdenas&Gracia (2009), JavaScript es un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas JavaScript van incrustados en los documentos HMTL, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos.

2.2.8.1 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 bucle:

- ✓ **Bucles Utilizando For**

Un bucle utilizando for se compone de cuatro estados, el estado expresión inicial es ejecutado una sola vez permite preparar o declarar variables, el estado de condición es ejecutado antes de cada repetición y retorna un valor el cual determina si el bucle debe seguir ejecutando o no, si el estado condicional retorna un valor falso el bucle se detiene. El estado incremento de la expresión es ejecutado al final de cada repetición y otorga la oportunidad de cambiar el estado de importantes variables.

- ✓ **Bucles Utilizando While**

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

- ✓ **Bucles Utilizando Do-while**

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

- ✓ **Break y Continue**

Esta declaración sirve cuando se desea parar un bucle sin necesidad de esperar que la condición no siga evaluando un valor verdadero. (págs. 13-15)

2.2.8.2 Incorporando Javascript

Existen técnicas para incorporar código Javascript dentro de HTML, las cuales Gauchat (2013, págs. 85-87), las define como:

- ✓ **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. El elemento `<script>` y su contenido pueden ser posicionados en cualquier lugar del documento, dentro de otros elementos o entre ellos.

2.2.9 AJAX

2.2.9.1 Introducción

Murphey (2013) considera que Ajax (Asynchronous JavaScript and XML) es un método, 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. Esta respuesta es asíncrona, por lo que el resto de código de la aplicación continuara ejecutándose, lo que hace que sea imperativo que se ejecute una función de devolución para manejar respuestas. (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** es utilizado para realizar solicitudes donde se va a obtener datos de servidor. Las solicitudes ejecutadas pueden ser almacenadas en la cache del navegador.

El método **POST** es utilizado cuando se ejecutan operaciones en las que se va a enviar datos a guardar en el servidor. Este tipo de método no se guarda en la cache del navegador. (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)

2.2.10 JQUERY

2.2.10.1 Concepto JQuery

De acuerdo con Álvarez (2012), 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.

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. Incluye funciones que serán útiles en cualquier script, posee una serie de clases y métodos útiles para hacer tareas reiterativas y además ayudan a hacer un código limpio, corto y reutilizable.

Core tiene las funciones clasificadas en diversos apartados:

- ✓ **\$() (La función JQuery):**

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.

2.2.11 ¿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, 2011)

De acuerdo con Pavón Maestras (2012), es una framework de twitter para desarrollo de aplicaciones web, sencillo y ligero, basado en los últimos estándares de desarrollo de Web HTML5, CSS3 y JavaScript/JQuery para el desarrollo de proyectos responsive, mobile first en la web.

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 un 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, 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 son una serie de servidores de automatización que permiten a los controladores, 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 8. 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 9. Unificación de las fases de la Metodología UWE

Fuente: Ludwing (2013)

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.

CAPITULO III

3. DESARROLLO DE LA PROPUESTA

3.1 CAPTURA ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS

3.1.1 MODELO DE REQUERIMIENTOS

3.1.1.1 Lista de actores

Se definen los actores que son las personas u otros sistemas que interactúan directamente con el sistema.

No.	Actor	Descripción
1	Usuario	Monitorea el personal en el sitio de trabajo y las actividades realizadas, ingresa actividades, controla el avance de obra y genera reportes de materiales y actividades.
2	Sistema Sico Cnel.	Proporciona datos de clientes de Cnel. EP, y permite el registro de las actividades de Coniel a la base de datos de Cnel. EP.

Tabla 3. Lista de Actores

Elaborado por: Jhonsson Córdova

3.1.1.2 Diagramas de caso de uso

Los diagramas de casos de usos serán representados en paquetes en el que cada uno corresponderá a un módulo de la aplicación web.

a) Diagrama de Caso de Uso del Módulo de Login de Usuario

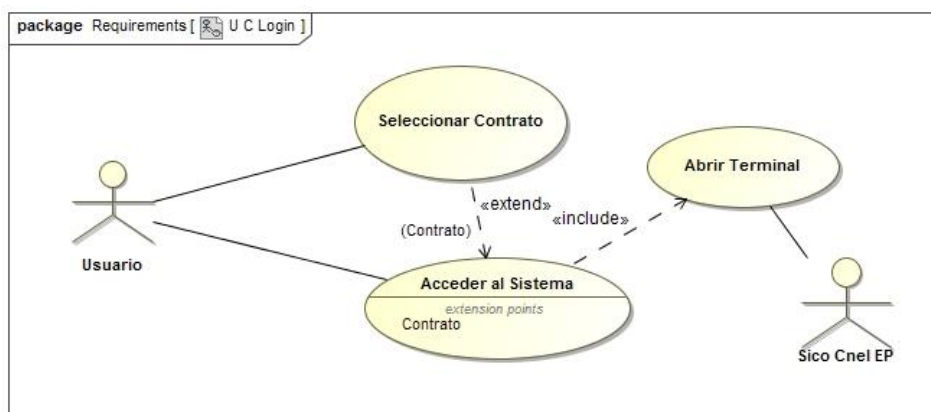


Figura 10. Caso de Uso del Módulo de Login de Usuario

Elaborado por: Jhonsson Córdova

b) Diagrama de Caso de Uso del Módulo de Ingreso de Actividades

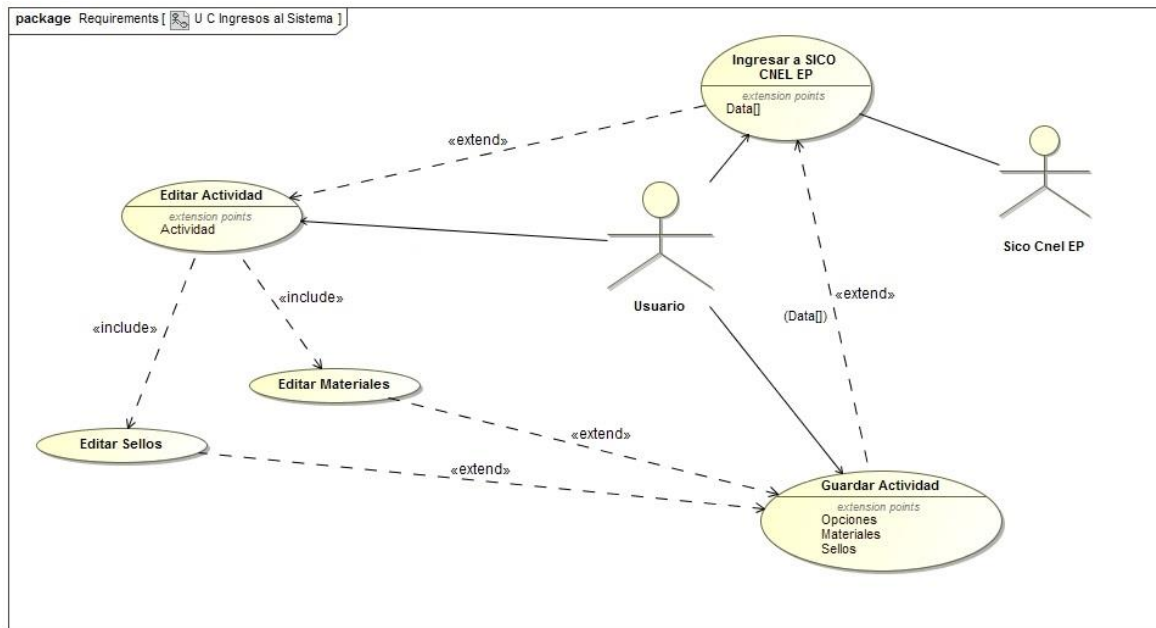


Figura 11. Caso de Uso del Módulo de Ingreso de Actividades

Elaborado por: Jhonsson Córdova

c) Diagrama de Caso de Uso del Módulo de Búsqueda de Información en el Sistema SICO

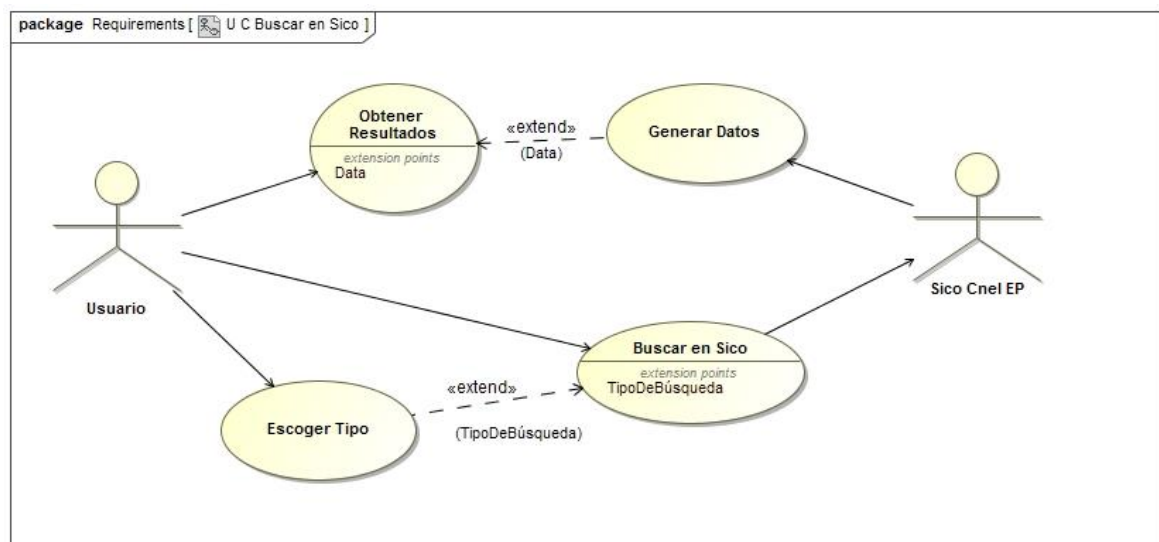


Figura 12. Caso de Uso del Módulo de Búsqueda de Información en SICO

Elaborado por: Jhonsson Córdova

d) Diagrama de Caso de Uso del Módulo de Listas de Actividades Realizadas

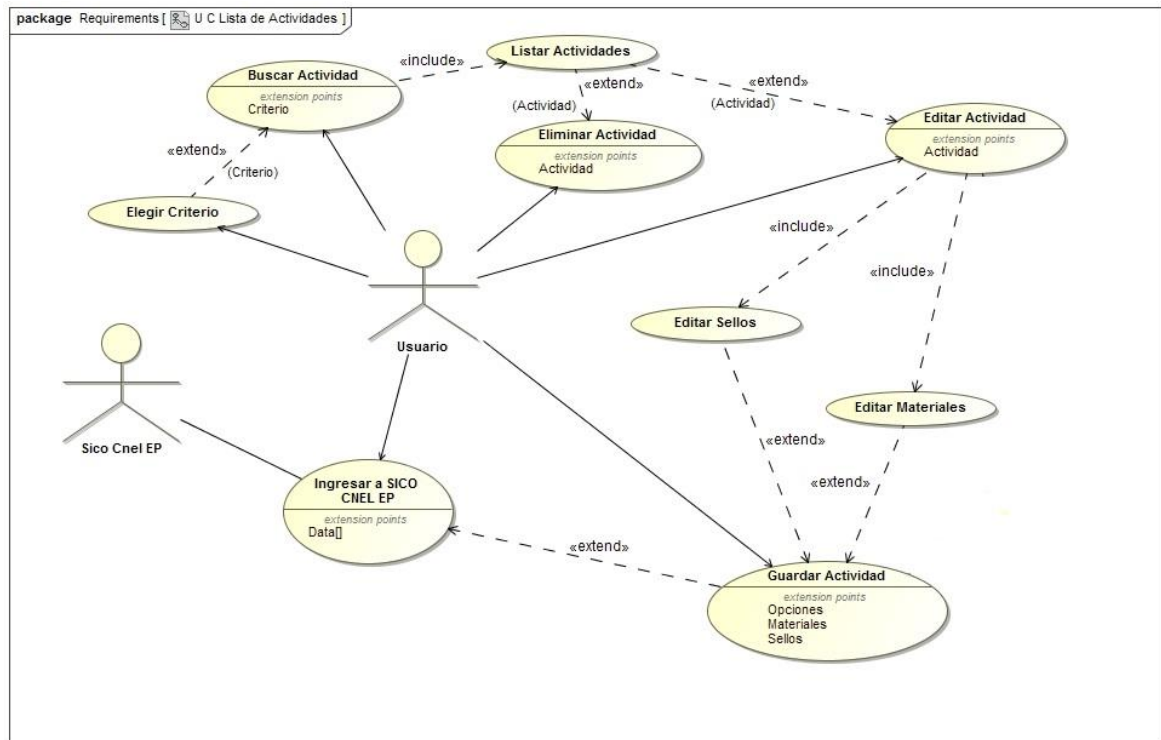


Figura 13. Caso de Uso del Módulo de Lista de Actividades Realizadas

Elaborado por: Jhonsson Córdova

e) Diagrama de Caso de Uso del Módulo de Fotos

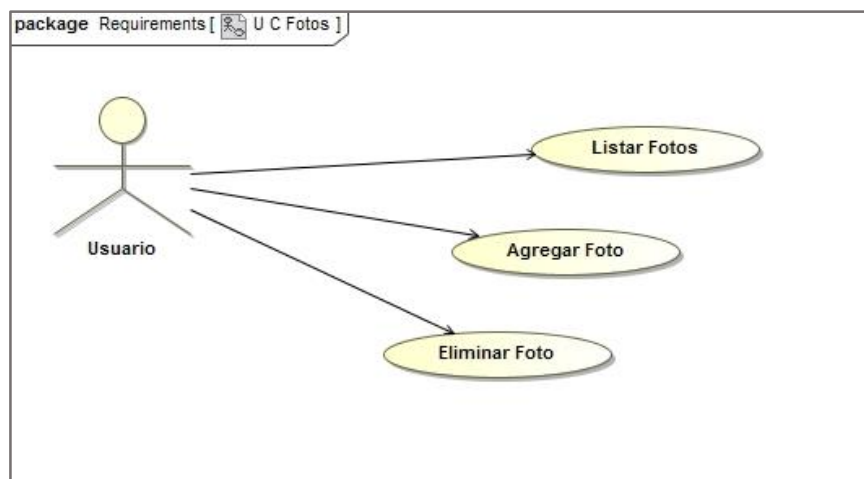


Figura 14. Caso de Uso del Módulo de Fotos

Elaborado por: Jhonsson Córdova

f) Diagrama de Caso de Uso del Módulo de Avance de Obra

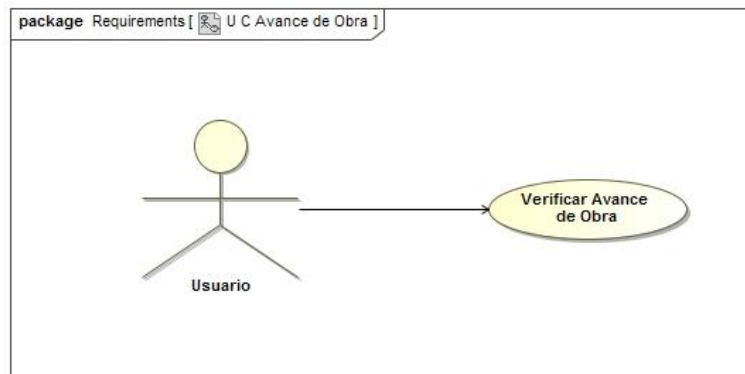


Figura 15. Caso de Uso del Módulo de Avance de Obra

Elaborado por: Jhonsson Córdova

g) Diagrama de Caso de Uso del Módulo de Monitoreo de Cuadrillas

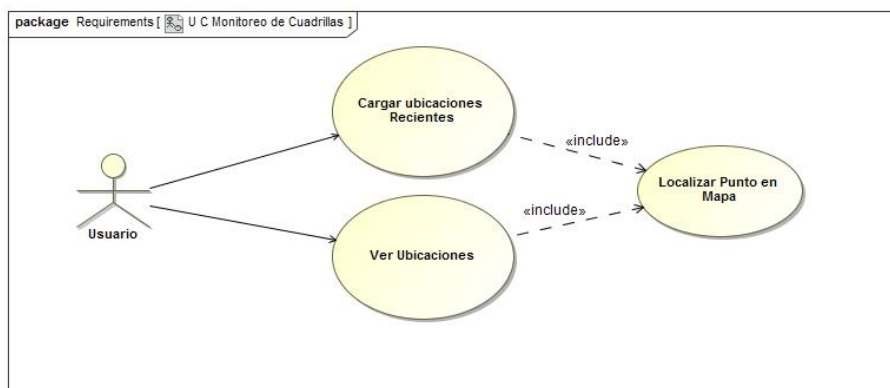


Figura 16. Caso de Uso del Módulo de Monitoreo de Cuadrillas

Elaborado por: Jhonsson Córdova

h) Diagrama de Caso de Uso del Módulo de Reportes

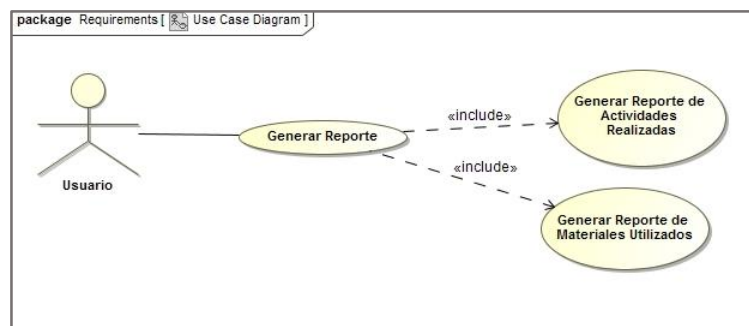


Figura 17. Caso de Uso del Módulo de Reportes

Elaborado por: Jhonsson Córdova

i) Diagrama de Caso de Uso del Módulo de Perfil de Usuario

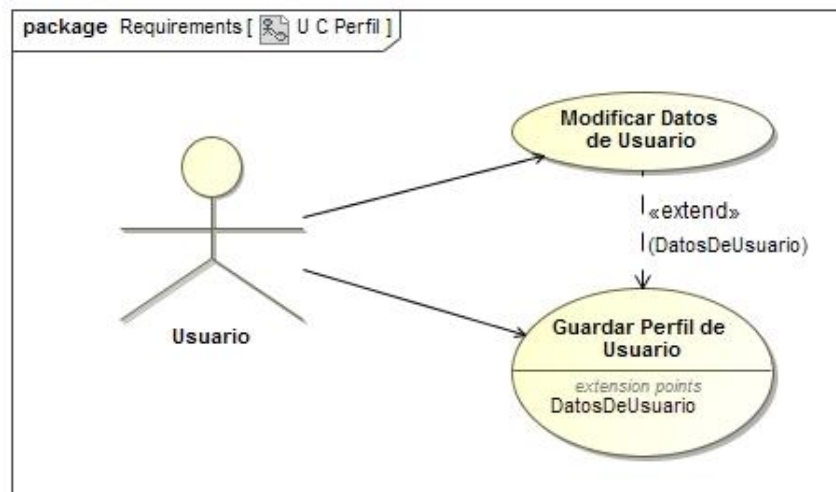


Figura 18. Caso de Uso del Módulo de Perfil de Usuario

Elaborado por: Jhonsson Córdova

3.1.1.3 Descripción de casos de uso

Para una mejor comprensión de los diagramas de casos de uso definidos anteriormente, se realizará una descripción de cada uno mediante tablas que contengan información que permitan un mejor entendimiento de los mismos.

❖ Descripción del Caso de Uso del módulo de Login de usuarios

Caso de Uso:	Módulo de Login de Usuario
Descripción:	Permite el inicio de sesión e ingreso a las opciones de la aplicación
Actores:	Usuario Sistema Sico Cnel. EP
Pre-condiciones:	El usuario debe tener asignado un usuario y contraseña
Actividades	
Escenario principal:	<ol style="list-style-type: none"> 1. El Usuario accede a la página mediante la URL. 2. El Usuario ingresa en la pantalla de Login, su usuario y contraseña 3. La aplicación verifica y valida los datos ingresados 4. La aplicación permite el acceso a la página principal y a las opciones del usuario.

Escenarios Alternos:	<p>a) El usuario ingresa nombre de usuario o contraseña no registrado o incorrectos en la aplicación</p> <ol style="list-style-type: none"> 1. El sistema emite un mensaje de error de usuario o contraseña no son correctos, Inténtelo nuevamente. <p>b) El usuario ingresa un nombre de usuario y contraseña correctos pero una selecciona un número de contrato inválido</p> <ol style="list-style-type: none"> 1. El sistema envía un mensaje de error de usuario especificado no cuenta con permisos necesarios para acceder al contrato <p>c) El usuario ingresa un nombre de usuario y contraseña correctos pero que ya está en uso</p> <ol style="list-style-type: none"> 1. El sistema envía un mensaje de error del usuario especificado ya está en uso
-----------------------------	---

Tabla 4. Descripción CU del Módulo de Login de Usuario

Elaborado por: Jhonsson Córdova

❖ Descripción del Caso de Uso del módulo de Ingreso de Actividades

Caso de Uso:	Módulo de Ingreso de Actividades
Descripción:	Permite al usuario el ingreso de actividades al sistema SICO, Además de ingresar, actualizar y guardar las actividades al sistema local.
Actores:	Usuario Sistema Sico Cnel. EP
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	<p>Actualizar actividad</p> <ol style="list-style-type: none"> 1. El Usuario busca o selecciona la actividad que desea actualizar 2. El sistema muestra los detalles de la actividad seleccionada 3. El usuario guarda en el sistema los cambios realizados 4. El sistema redirecciona a la página del detalle de la actividad seleccionada para que el usuario pueda visualizar los cambios efectuados.
Escenarios Alternos:	<p>a) Ingresar actividad a sistema SICO</p> <ol style="list-style-type: none"> 1. El usuario puede ingresar actividades al sistema SICO, lo puede realizar seleccionando el tipo de solicitud, y además ingresando los datos necesarios para la ejecución de este proceso. 2. El usuario guarda en el sistema los campos ingresados.

	<p>3. El sistema muestra un mensaje donde especifica que el ingreso fue realizado completamente, además de que va mostrando en tiempo de ejecución los estados por los que la solicitud va cambiando.</p> <p>b) Ingreso de actividad a sistema local</p> <p>1. El usuario puede ingresar actividades al sistema local, lo puede realizar seleccionando el tipo de solicitud, y además ingresando los datos necesarios para la ejecución de este proceso.</p> <p>2. El usuario guarda en el sistema los campos ingresados.</p> <p>3. El sistema muestra un mensaje donde especifica que el ingreso fue realizado completamente, además redirecciona a la página y bloquea la edición del formulario.</p> <p>c) Eliminar actividad del sistema local</p> <p>1. El usuario selecciona la actividad que desea eliminar</p> <p>2. El usuario elimina del sistema la actividad seleccionada</p> <p>3. El usuario confirma la eliminación de la actividad</p> <p>4. El sistema retorna a la lista de actividades en la que se podrá visualizar que la actividad ya no se encuentra en el sistema.</p>
--	--

Tabla 5. Descripción del CU del Módulo de Ingreso de Actividades

Elaborado por: Jhonson Córdova

❖ Descripción del Caso de Uso del módulo de Búsqueda de Información en el sistema SICO

Caso de Uso:	Módulo de Búsqueda de Información en el sistema SICO
Descripción:	Permite al usuario realizar consultas por criterios en el sistema SICO Cnel. EP.
Actores:	Usuario Sistema Sico Cnel. EP
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	<p>Búsqueda de Información</p> <ol style="list-style-type: none"> 1. El Usuario selecciona el criterio por el que desea realizar la búsqueda 2. El usuario ingresa el dato por el que quiere realizar la consulta 3. El usuario presiona el botón Buscar 4. El sistema SICO Cnel. EP recibe el dato y realiza el proceso de búsqueda 5. El sistema retorna el resultado de la búsqueda 6. El usuario visualiza el resultado de la búsqueda realizada

Escenarios Alternos:	<p>a) El usuario seleccionó el criterio cuenta e ingresó un dato incorrecto a buscar</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error “Error cuenta ingresada no válida” <p>b) El usuario seleccionó el criterio nombre e ingresó un dato incorrecto a buscar</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error “Error, nombre de cliente no válido” <p>c) El usuario seleccionó el criterio medidor e ingresó un dato incorrecto a buscar</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error “Error, número de medidor ingresado no valido” <p>d) El usuario seleccionó el criterio geocódigo e ingresó un dato incorrecto a buscar</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error “Error, geocódigo incorrecto”
-----------------------------	--

Tabla 6. Descripción del CU del Módulo de Búsqueda de Información en el sistema SICO

Elaborado por: Jhonsson Córdova

❖ **Descripción del Caso de Uso del módulo de Listas de Actividades Realizadas**

Caso de Uso:	Módulo de Listas de Actividades Realizadas
Descripción:	Permite al usuario realizar consultas por criterios las actividades realizadas en el sistema local, además de actualizar, eliminar y guardar las actividades.
Actores:	Usuario Sistema Sico Cnel. EP
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	<p>Actualizar Información de actividad</p> <ol style="list-style-type: none"> 1. El Usuario busca o selecciona la actividad que desea actualizar 2. El sistema muestra los detalles de la actividad seleccionada 3. El usuario guarda en el sistema los cambios realizados 4. El sistema redirecciona a la página del detalle de la actividad seleccionada para que el usuario pueda visualizar los cambios efectuados.
Escenarios Alternos:	<p>a) Búsqueda de información en el sistema local</p> <ol style="list-style-type: none"> 1. El Usuario selecciona el criterio por el que desea realizar la búsqueda

	<ol style="list-style-type: none"> 2. El usuario ingresa el dato por el que quiere realizar la consulta 3. El usuario presiona el botón Buscar 4. El sistema realiza el proceso de búsqueda 5. El sistema retorna el resultado de la búsqueda 6. El usuario visualiza el resultado de la búsqueda realizada <p>b) Eliminar actividad del sistema local</p> <ol style="list-style-type: none"> 1. El usuario selecciona la actividad que desea eliminar 2. El usuario elimina del sistema la actividad seleccionada 3. El usuario confirma la eliminación de la actividad 4. El sistema retorna a la lista de actividades en la que se podrá visualizar que la actividad ya no se encuentra en el sistema.
--	---

Tabla 7. Descripción del CU del Módulo de Listas de Actividades Realizadas

Elaborado por: Jhonsson Córdova

❖ Descripción del Caso de Uso del módulo de Fotos

Caso de Uso:	Módulo de Fotos
Descripción:	Permite al usuario Agregar, eliminar y listar fotos
Actores:	Usuario
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	<p>Lista de Fotos por actividad</p> <ol style="list-style-type: none"> 1. El Usuario selecciona la actividad de la que desea listar las fotos 2. El sistema muestra las fotos de la actividad en galería 3. El usuario visualiza las fotos, puede dar clic para visualizar en tamaño real.
Escenarios Alternos:	<p>a) Agregar foto</p> <ol style="list-style-type: none"> 1. El Usuario selecciona la actividad a la que desea agregarle fotos 2. El usuario presiona el botón seleccionar imagen que le permite seleccionar desde su ordenador. 3. El usuario selecciona la imagen 4. El usuario presiona el botón subir foto 5. El sistema retorna a la galería en donde se puede visualizar que la imagen ha sido agregada a la actividad

	b) Eliminar foto <ol style="list-style-type: none"> 1. El usuario selecciona la actividad a la que desea eliminarle fotos 2. El sistema muestra las imágenes que contiene la actividad seleccionada 3. El usuario da clic en la 'X' que está en la parte inferior de la foto que desea eliminar 4. El sistema muestra un mensaje de confirmación 5. El usuario confirma que desea eliminar 6. El sistema retorna a la galería de la actividad en donde se podrá visualizar que la foto ha sido eliminada
--	---

Tabla 8. Descripción del CU del Módulo de Fotos

Elaborado por: Jhonsson Córdova

❖ **Descripción del Caso de Uso del módulo de Avance de Obra**

Caso de Uso:	Módulo de Avance de Obra
Descripción:	Permite al usuario verificar el avance de obra
Actores:	Usuario
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	Avance de obra <ol style="list-style-type: none"> 1. El sistema mostrará un cuadro estadístico de porcentaje de avance de obra

Tabla 9. Descripción del CU del Módulo de Avance de Obra

Elaborado por: Jhonsson Córdova

❖ **Descripción del Caso de Uso del módulo de Monitoreo de Cuadrillas**

Caso de Uso:	Módulo de Monitoreo de Cuadrillas
Descripción:	Permite al usuario monitorear la ubicación de las cuadrillas
Actores:	Usuario
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	Mapa con ubicación general en la ciudad de Machala <ol style="list-style-type: none"> 1. El usuario visualizará la ubicación de las cuadrillas de forma dinámica.

Tabla 10. Descripción del CU del Módulo de Monitoreo de Cuadrillas

Elaborado por: Jhonsson Córdova

❖ **Descripción del Caso de Uso del módulo de Reportes**

Caso de Uso:	Módulo de Reportes
Descripción:	Permite al usuario generar reportes de actividades realizadas y de materiales utilizados
Actores:	Usuario
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	Generar reporte de actividades realizadas <ol style="list-style-type: none"> 1. El sistema muestra las actividades realizadas hasta la fecha 2. El usuario escoge la opción de generar reporte 3. El sistema descarga un archivo de Excel con el reporte generado
Escenario Alternativo:	a) Generar reporte de materiales utilizados <ol style="list-style-type: none"> 1. El sistema lista los materiales que están asignados al contrato. 2. El sistema lista los materiales que se han utilizado 3. El sistema lista los materiales que aún no son utilizados

	<ol style="list-style-type: none"> 4. El usuario escoge la opción de generar reporte 5. El sistema descarga un archivo de Excel con las listas descritas anteriormente.
--	---

Tabla 11. Descripción del CU del Módulo de Reportes

Elaborado por: Jhonsson Córdova

❖ Descripción del Caso de Uso del módulo de Perfil de Usuario

Caso de Uso:	Módulo de Perfil de Usuario
Descripción:	Permite al usuario modificar datos de perfil de usuario
Actores:	Usuario
Pre-condiciones:	El usuario debe haber iniciado sesión en el sistema
Actividades	
Escenario principal:	<p>Actualizar Perfil</p> <ol style="list-style-type: none"> 1. El usuario podrá actualizar su nombre de usuario y su clave de acceso al sistema 2. El sistema emitirá un mensaje de confirmación 3. El usuario guardará los cambios realizados 4. El sistema retornará la información del usuario con las actualizaciones correspondientes.

Tabla 12. Descripción del CU del Módulo de Perfil de Usuario

Elaborado por: Jhonsson Córdova

3.2 INGENIERÍA DE LA SOLUCIÓN WEB

3.2.1 MODELOS DE ANÁLISIS DE LA APLICACIÓN

3.2.1.1 Modelo de Contenido

El modelo de contenido es representado mediante el Diagrama de clases de análisis, ya que permite visualizar las entidades de la aplicación web.

- Diagrama de clases de la aplicación

3.2.1.2 Modelo de Navegación

Permite representar la navegación entre las páginas de la aplicación web. Dichas asociaciones se etiquetan, pueden tener asociados atributos y pueden ser unidireccionales o bidimensionales. Este modelo se representa a través de diagramas de navegabilidad.

- Diagrama de Navegación de la Aplicación Web

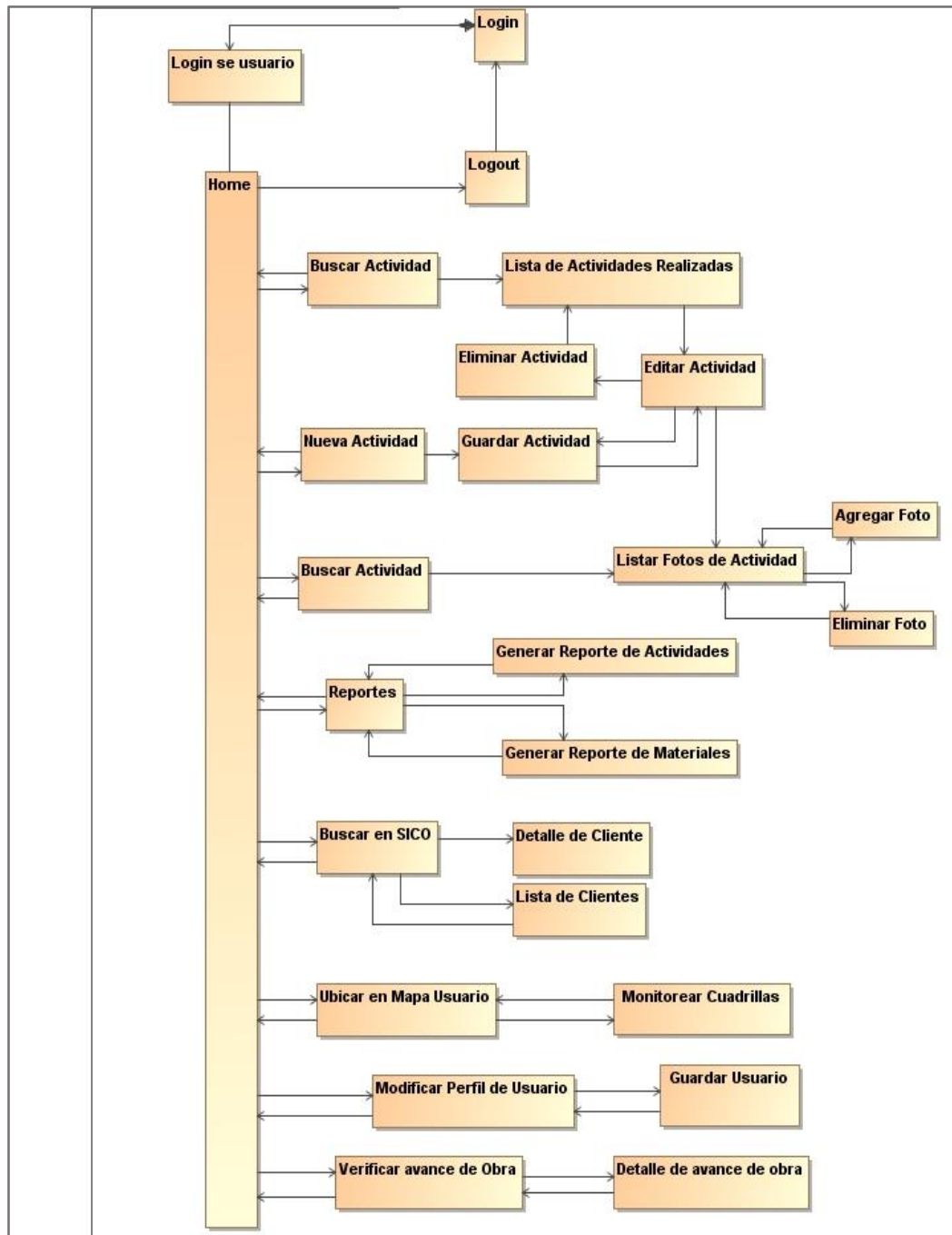


Figura 19. Diagrama de Navegabilidad de la Aplicación

Elaborado por: Jhonsson Córdova

3.2.1.3 Modelo de Presentación

Los modelos de Presentación definen la forma en que se presentará la información al usuario, es un aspecto importante en una aplicación web, los esquemas de las interfaces de usuario permiten comprender y especificar las interacciones entre los actores y la aplicación. Los esquemas de las interfaces del usuario o de presentación, no sólo ayudan posteriormente a construir una adecuada interfaz gráfica en la aplicación, sino que ayudan a comprender mejor lo casos de uso y los requisitos.

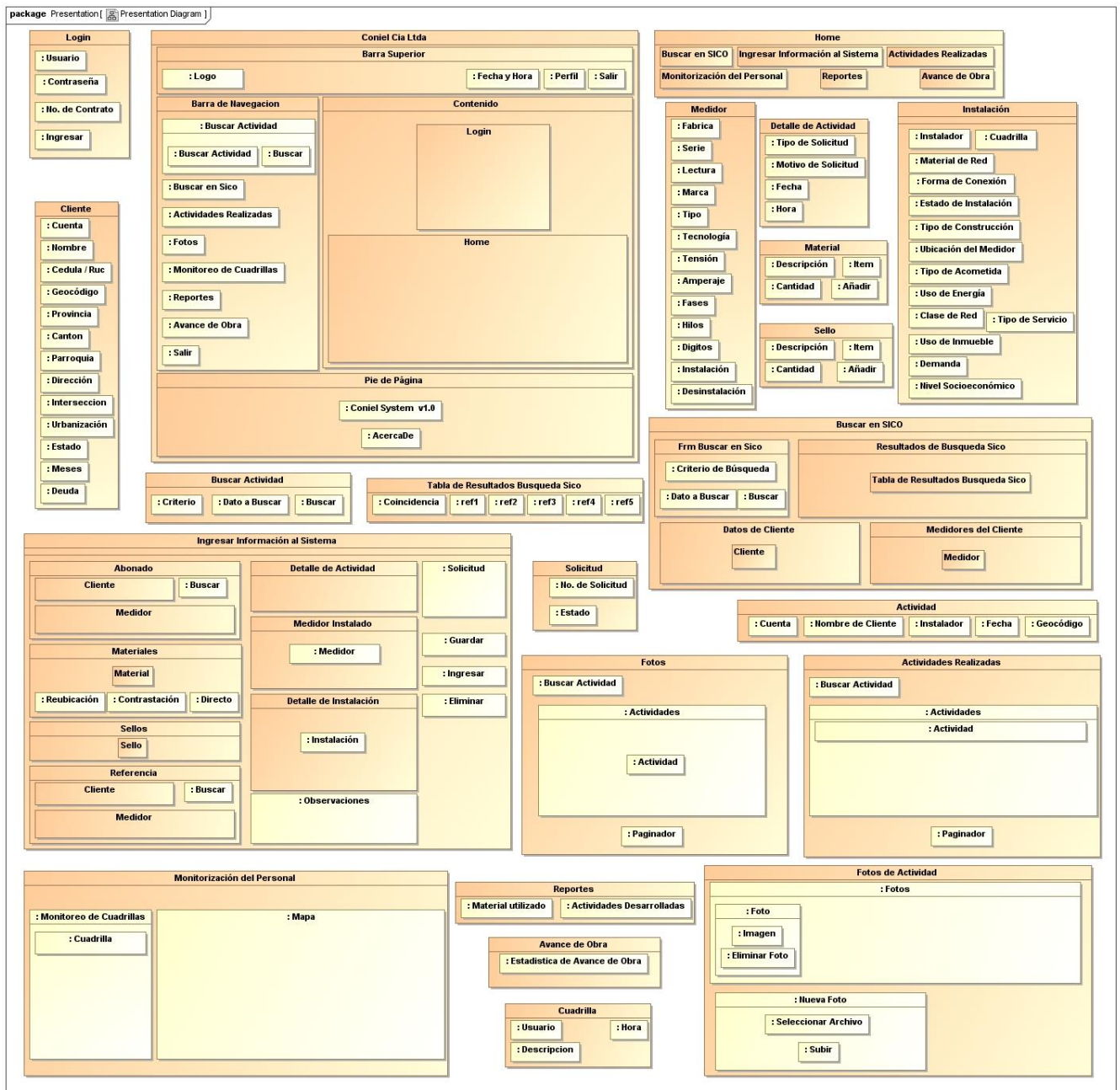


Figura 20. Diagrama de Presentación de la Aplicación

Elaborado por: Jhonsson Córdova

3.2.2 MODELOS DE DISEÑOS

3.2.2.1 Arquitectura de la Aplicación

Todo sistema esta compuestos por varias partes, por lo que la arquitectura de la aplicación define las relaciones que hay entre ellas.

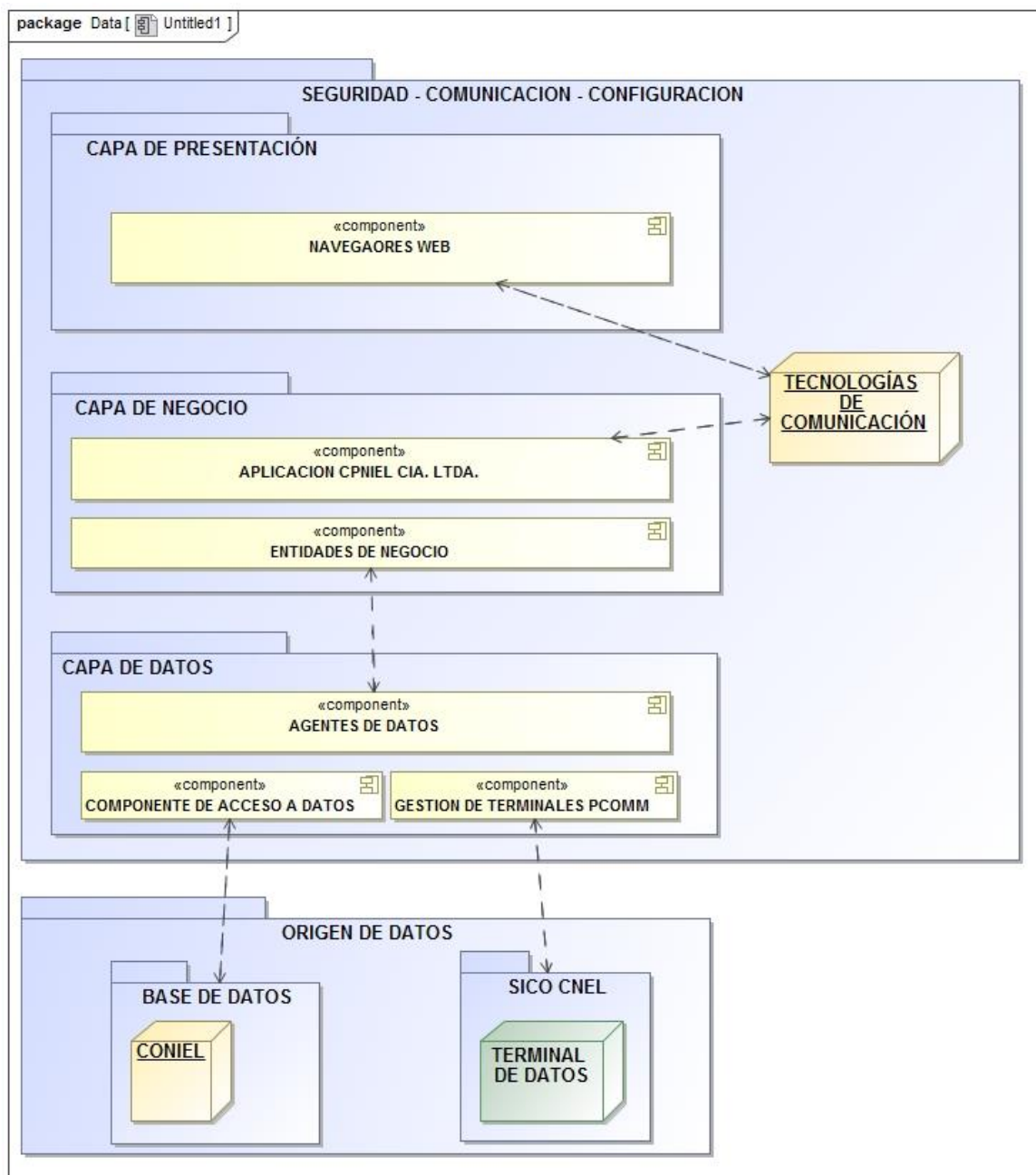


Figura 21. Arquitectura de la Aplicación

Elaborado por: Jhonsson Córdova

3.2.2.2 Modelo de Datos

Este módulo permite definir el modelo de datos de la aplicación web, se implementará en el modelo físico lo que se encuentre almacenado en PostgreSQL, ya que es nuestro DBMS seleccionada para el almacenamiento.

Este modelo describe los elementos de la realidad que se encuentran involucrados en el desarrollo de esta aplicación, y la manera en la que están relacionados entre sí, esto permite identificar la estructura de información para obtener mejor resultados de la comunicación y de la precisión en la aplicación en el intercambio de datos.

3.3 IMPLEMENTACIÓN DE LA APLICACIÓN WEB

Durante esta fase se definirá la manera de implementar los modelos del diseño, es decir se analizarán y revisarán los componentes de la aplicación web, además de que se revisarán a detalle la implementación final de un caso de uso del sistema.

3.3.1 ESTRUCTURA DE IMPLEMENTACIÓN DEL SISTEMA

Define la estructura de los directorios y subdirectorios que se utilizaron para el desarrollo del sistema. La estructura utilizada para el desarrollo de la aplicación con el framework Django en su versión 1.6 bajo el Ide de desarrollo Pycharm se muestra a continuación:

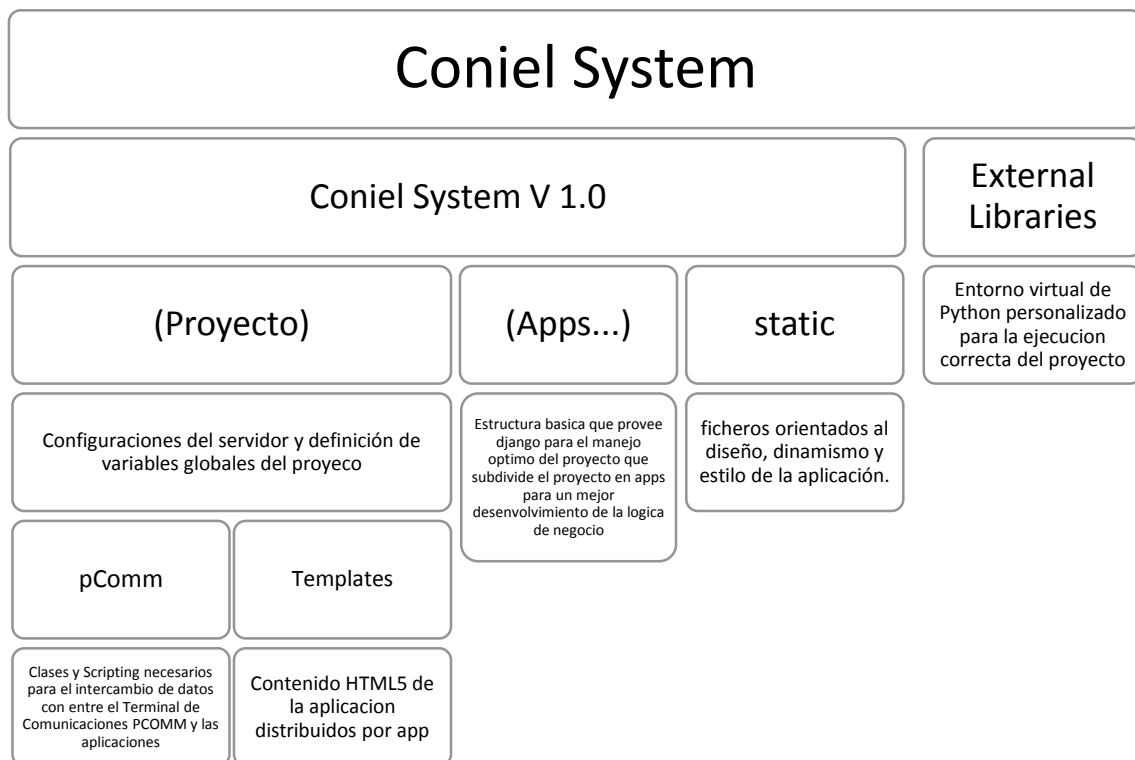


Figura 22. Estructura de Implementación del Sistema

Elaborado por: Jhonsson Córdova

En la **figura 22**, se muestra la organización del proyecto utilizando el esquema que nos provee el framework para el desarrollo adaptándolo para la integración al sistema Personal Communications Terminal de IBM, además nos muestra que cada parte del proyecto tiene su directorio en específico para promover un código limpio y estable.

3.3.2 EJEMPLO DE IMPLEMENTACIÓN DEL CASO DE USO DE LOGIN DE USUARIO

A continuación se muestra la implementación del caso de Uso de Login especificado en la primera fase de la metodología.

- Se crea el proyecto de django mediante el id de desarrollo Pycharm.

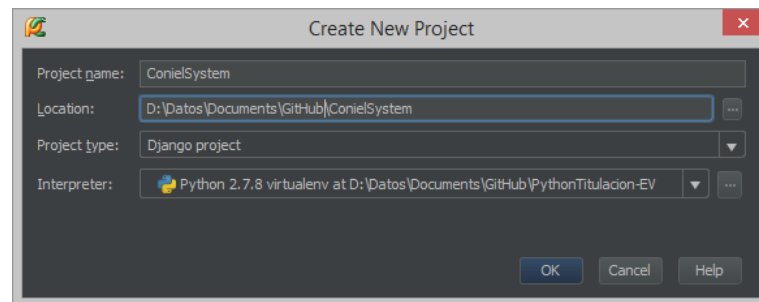


Figura 23. Ejemplo de creación de nuevo proyecto con Pycharm

Elaborado por: Jhonsson Córdova

- Se modifica las configuraciones del proyecto contenidas en el fichero settings.py para la ejecución correcta del servidor en base los puntos predefinidos que seguirá la aplicación.

```
...
# Database
# https://docs.djangoproject.com/en/1.6/ref/settings/#databases
DATABASES = {

    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'Coniel',
        'USER': 'postgres',
        'PASSWORD': '*****',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    },
}
# Internationalization
# https://docs.djangoproject.com/en/1.6/topics/i18n/
LANGUAGE_CODE = 'es-ec'
TIME_ZONE = 'America/Guayaquil'
...
```

- Se crea la app de la misma manera, ésta se encarga de gestionar los usuarios y los accesos a la aplicación.

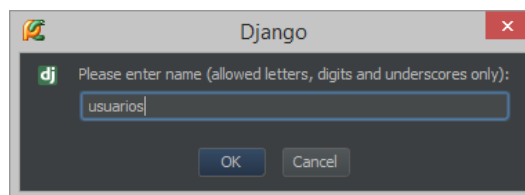


Figura 24. Ejemplo de crear nueva aplicación Django con Pycharm

Elaborado por: Jhonsson Córdova

- Se registra la app recientemente creada en el proyecto.

Settings.py

```
...
# Application definition
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'suit',
    'django.contrib.admin',
    'django_ajax',
    'bootstrap3',
    'serviciosWeb',
    'busquedas',
    'usuarios',
    'ingresos',
    'inventario',
    'django_extensions',
)
...
```

- Tomando como base el proyecto de django se modifican los ficheros que se nos proporcionan para el desarrollo, se comienza a aplicar la lógica mediante modelos, los mismos que son la definición de las clases obtenidas en las fases previas de la presente metodología.

Usuarios/models.py

```
class usuarioSico(models.Model):
    nombre = models.CharField(max_length=10, verbose_name='Nombre de Usuario en Sico')
    clave = models.CharField(max_length=10, verbose_name='Contraseña en Sico')
    contrato = models.ForeignKey('inventario.contrato')

    def __unicode__(self):
        return '%s - %s' % (self.nombre, self.contrato.num)

    class Meta:
        verbose_name='Usuario en Sico'
        verbose_name_plural='Usuarios en Sico'

#MODIFICANDO EL MODELO POR DEFECTO USER
User.add_to_class('sesion_sico', models.CharField(max_length=2, null=True, blank=True))
User.add_to_class('usuario_sico', models.ManyToManyField(usuarioSico, blank=True, null=True))
```

Inventario/models.py

```
class contrato(models.Model):
    num = models.CharField(max_length=10, primary_key=True, verbose_name='Número de Contrato')
    descripcion = models.CharField(max_length=150, verbose_name='Descripción')
    zonas = models.CharField(max_length=150, verbose_name='Zona(s)')
    codigoInstalador = models.PositiveSmallIntegerField(verbose_name='Código de Instalador')
    inicioVigencia = models.DateField(verbose_name='Vigente Desde')
    finalVigencia = models.DateField(verbose_name='Vigente Hasta')

    def __unicode__(self):
        return self.num
```


- A continuación se registra las urls que son identificadas como válidas para esta aplicación editando el fichero urls.py.

```
admin.autodiscover()

urlpatterns = patterns("",

    #Administración
    url(r'^admin/', include(admin.site.urls)),

    #Usuarios
    url(r'^$', 'usuarios.views.main', name='main'),
    url(r'^login$', 'usuarios.views.ingreso', name='login'),
    url(r'^logout$', 'usuarios.views.salir', name="logout"),
    url(r'^home$', 'usuarios.views.home', name='home'),
```

- A continuación se modifica las vistas especificadas para cada url de esta manera se gestiona las peticiones de los usuarios.

```
def main(request):
    #return render_to_response('main.html', {}, context_instance=RequestContext(request))
    return ingreso(request)

@login_required()
def salir(request):
    cerrarSico(request.user.sesion_sico)
    request.user.sesion_sico = ""
    request.user.save()
    logout(request)
    return HttpResponseRedirect('/login')

def cerrarSico(sesionAct):
    c = manejadorDeConexion()
    c.closeProgram(sesionAct)

def ingreso(request):
    error = None
    try:
        if request.user.sesion_sico:
            return home(request)
        else:
            salir(request)
    except:
        pass
    if request.POST:
        form = Login(request.POST)
        if form.is_valid():
            username = form.cleaned_data["usuario"]
            password = form.cleaned_data["clave"]
            contrato = form.cleaned_data["sico"]
            user = authenticate(username=username, password=password)
            if user is not None:
                if user.is_active:
                    if user.sesion_sico:
                        error = "El Usuario especificado ya esta en uso."
                    try:
                        logout(request)
                    except:
                        pass
```

```

else:
    try:
        u = usuarioSico.objects.get(user=user, contrato=contrato)
    except:
        u = False
    if isinstance(u, usuarioSico):
        if integracion(u.nombre, u.clave, user):
            login(request, user)
            request.session['contrato'] = contrato
            return HttpResponseRedirect('/home')
        else:
            error = 'El Sistema Comercial(Sico Cnel) no esta disponible por el momento...'
            user.sesion_sico=""
            user.save()
    else:
        error = 'El Usuario Especificado no cuenta con permisos necesarios para acceder al contarto'
elif username and password:
    error = "Su Usuario o Contraseña no son correctos, Intentelo nuevamente."
form = Login()
return render_to_response('usuarios/login.html', {'f': form, 'errors': error},
    context_instance=RequestContext(request))

@login_required()
def home(request):
    print 'Sesion : %s' % request.user.sesion_sico
    if request.user.sesion_sico:
        return render_to_response('usuarios/home.html', {}, context_instance=RequestContext(request))
        return HttpResponseRedirect('/login')

def integracion(u, c, user):
    conn = None
    while True:
        conn = manejadorDeConexion()
        user.sesion_sico = str(conn.getAvailableConnection())
        if not User.objects.filter(sesion_sico=user.sesion_sico):
            user.save()
            break

    return conn.openSession(connectionName=user.sesion_sico, usuario=u, contrasenia=c)

```

- Adicionalmente a través de la siguiente clase se gestionan las peticiones al terminal de PCOMM, y de esta manera logramos que nuestro servidor abra una terminal de datos por cada usuario logeado en el sistema.

```

import pythoncom

__author__ = 'Jhonsson'
from time import sleep
import win32com
import win32com.client

class manejadorDeConexion:
    def __init__(self):
        pythoncom.CoInitialize()
        self.PCommConnMgr = win32com.client.Dispatch('PCOMM.autECLConnMgr')
        self.connList = self.PCommConnMgr.autECLConnList
        self.activeSession = None
        self.activeConnection = ""
        self.estado = False

```

```

def setActiveSession(self, connectionName):
    self.activeSession = win32com.client.Dispatch("PCOMM.autECLSession")
    if connectionName:
        self.activeConnection = connectionName
        self.activeSession.SetConnectionByName(self.activeConnection)

    # return value from screen at position (row, col)

def getText(self, row, col, length=None, connectionName=None):
    if connectionName is not None:
        self.setActiveSession(connectionName)

    if length is None:
        self.activeSession.autECLPS.autECLFieldList.Refresh()
        field = self.activeSession.autECLPS.autECLFieldList.FindFieldByRowCol(row, col)
        length = field.Length
    result = self.activeSession.autECLPS.GetText(row, col, length)
    return result

def sendKeys(self, count, key, row=None, col=None, connectionName=None, wait=True):
    n = 0
    if connectionName is not None:
        self.setActiveSession(connectionName)

    while n < count:
        if row is None or col is None:
            self.activeSession.autECLPS.SendKeys("%s" % key)
        else:
            self.activeSession.autECLPS.SendKeys("%s" % key, row, col)
        if wait:
            if not self.activeSession.autECLOIA.WaitForAppAvailable(5000): return False
            if not self.activeSession.autECLOIA.WaitForInputReady(5000): return False
        n += 1
    return True

def getAvailableConnection(self):
    self.connList.Refresh()
    self.activeConnection = chr(self.connList.Count + 65)
    self.openProgram()
    self.connList.Refresh()
    return self.activeConnection

def openSession(self, connectionName=None, usuario="none", contrasenia="none"):

    self.estado = False

    if connectionName is not None:
        self.setActiveSession(connectionName)
    else:
        self.setActiveSession(self.getAvailableConnection())

    segundos = 10

    while not self.activeSession.Ready and segundos >= 0:
        print "la session no esta lista aun ..."
        #return False
        sleep(1)
        segundos -= 1

    if segundos >= 0:

```

```

        if self.activeSession.autECLOIA.WaitForAppAvailable(5000):
            if self.activeSession.autECLOIA.WaitForInputReady(5000):
                print self.getText(22, 50, length=8)
                print self.getText(21, 50, length=12)

        if self.activeSession.autECLPS.WaitForString('USUARIO', 21, 50, 10000):
            if self.sendKeys(1, usuario, row=21, col=63):
                if self.sendKeys(1, contrasenia, row=22, col=63):
                    if self.sendKeys(6, '[enter]'):
                        self.estado = True
                        return self.estado

        self.closeProgram(self.activeConnection, directo=True)

        return self.estado

def openProgram(self):
    print "Abriendo el Programa, Sesion: {0}".format(self.activeConnection)
    self.PCommConnMgr.StartConnection(
        "PROFILE=.\\ControlSystem\\pComm\\sico\\CNEL.WS
        WINSTATE=MIN".format(self.activeConnection))
    CONNNNAME={0}

def closeProgram(self, connection, directo=False):
    if not connection:
        connection = self.activeConnection
    try:
        print "Cerrando el Programa, Sesion: {0}".format(connection)
        if not directo:
            if not self.activeSession:
                self.setActiveSession(connection)
            self.sendKeys(10, '[pf12]', connectionName=connection, wait=False)
            self.PCommConnMgr.StopConnection(connection, "saveprofile=no")
        except:
            print "No se ha podido cerrar la sesion: {0}".format(connection)

```

- Se escriben los ficheros html que son interpretados por los navegadores de la siguiente manera dentro del directorio de templates del proyecto

templates/base.html

```

{% load staticfiles %}
{% load bootstrap3 %}

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es-es">
<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title id="title">
        {% block title %}
        P&#225;gina Sin T&#237;tulo
        {% endblock %}
    </title>

    <!-- CSS de Bootstrap -->
    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet" media="screen">
    <link href="{% static 'css/bootstrap-theme.min.css' %}" rel="stylesheet" media="screen">

```

```

<!-- CSS de JQuery-UI -->
<link href="{% static 'css/jquery-ui.min.css' %}" rel="stylesheet" media="screen">
<link href="{% static 'css/jquery-ui.structure.min.css' %}" rel="stylesheet" media="screen">
<link href="{% static 'css/jquery-ui.theme.min.css' %}" rel="stylesheet" media="screen">

<!-- CSS Personal -->
<link href="{% static 'css/jhonsson.css' %}" rel="stylesheet" media="screen">

<!-- Favicon-->
<link id="favicon" rel="icon" type="image/x-icon" sizes="16x16" href="{% static 'favicon/favicon.ico' %}">
<link id="favicon2x" rel="icon" type="image/png" sizes="32x32" href="{% static 'favicon/favicon.png' %}">
<link rel="shortcut icon" type="image/png" href="{% static 'favicon/favicon.png' %}">

<!-- JS Librería jQuery -->
<script src="{% static 'js/jquery-1.11.1.js' %}"></script>
<script src="{% static 'js/jquery-ui.min.js' %}"></script>

<!-- JS de Bootstrap -->
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<script src="{% static 'js/bootstrap-hover-dropdown.min.js' %}"></script>
<script src="{% static 'js/jhonsson.js' %}"></script>

<!-- JS Librería jQueryDajax respuestas asincronas -->
<script src="{% static 'js/jquery.ajax.js' %}"></script>
<script src="{% static 'js/jquery.ajax-plugin.js' %}"></script>
<script src="{% static 'js/jquery.dajax.core.js' %}"></script>

<!-- librerías opcionales que activan el soporte de HTML5 para IE8 -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->

{% block jquery %}
<script>
    var dx=$('#document');
    dx.ready(navegacion);

    function navegacion()
    {
        $('#{{ nav }}').addClass('active');
    }
</script>
{% endblock %}

</head>

<body onLoad="window.setTimeout('hora()',1000);">
<div id="wrap">
    <nav class="navbar navbar-default navbar-fixed-top" role="navigation">
        <div style="padding-right: 20px">
            <div class="navbar-header">
                <a href="{% url 'home' %}">
                    
                </a>
            </div>
            <div class="navbar-collapse collapse">
                {% block navTop %}
                {% include "navTop.html" %}
                {% endblock %}
            </div><!--/.nav-collapse -->
        </div>
    </nav>
    <div class="container-fluid">

```

```

<div class="row" >
  <div class="col-sm-3 col-md-2 sidebar">
    {% block navLeft %}
      {% include "navLeft.html" %}
    {% endblock %}
  </div>
  <div class="col-sm-9 col-md-10 main">
    {% block content %}
      <h1>Contenido</h1>
    {% endblock %}
  </div>
</div>
</div>
<div id="push"></div>
</div>
<footer id="footer" class="footer">
  <div class="container text-center">
    <p>Coniel System V 1.0 - 2014.<br>
    <a href="#Acercade" class="text-info">Acerca de</a></p>
    {% block footer %}
    {% endblock %}
  </div>
</footer>
{% block jqueryfooter %}

{% endblock %}
</body>
</html>

```

Templates/usuarios/Login.html

```

{% extends "base.html" %}
{% load bootstrap3 %}
{% load static %}

{% block title %}
  Acceso al Sistema
{% endblock %}

{% block content %}
  <div class="row">
    <div class="col-sm-11 col-md-9">
      <form role="form" class="form-signin" method="post" id="login-form" action="{% url 'login' %}">

        {% csrf_token %}
        <div class="panel panel-primary">
          <div class="panel-heading">
            <h8>
              <span class="glyphicon glyphicon-log-in icon-white"></span>
              Inicio de Sesi&oacute;n
            </h8>
          </div>
          <div class="panel-body">
            <div class="form-row control-group">
              {{ f.usuario.label }}:
              <div class="controls">
                {% bootstrap_field f.usuario layout='inline' %}
              </div>
            </div>
            <div class="form-row control-group">
              {{ f.clave.label }}:
              <div class="controls">
                {% bootstrap_field f.clave layout='inline' %}
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
    <div class="form-row control-group">
        {{ f.sico.label }}:
        <div class="controls">
            {% bootstrap_field f.sico layout='inline' %}
        </div>
    </div>
    <div class="form-row submit-row">
        <input type="submit" class="btn btn-info" value="Ingresar"/>
    </div>
    {% if errors %}
        <br>
        <div class="alert alert-danger alert-dismissible">
            <button type="button" class="close" data-dismiss="alert">&times;</button>
            <label class="control-label small" for="inputEmail">
                {{ errors }}
            </label>
        </div>
    {% endif %}
</div>
</div>
<br>
<input type="hidden" name="next" value="{{ next }}" />
</form>
</div>
<div class="col-sm-1 col-md-3"></div>

</div>
{% endblock %}

```

Templates/usuarios/home.html

```

{% extends 'base.html' %}
{% load staticfiles %}
{% block title %}
    {{ user.first_name }} {{ user.last_name }}
{% endblock %}
{% block content %}
    <div class="row" style="margin-top: 50px">
        <div class="col-md-4 text-center">
            <button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'buscarCuenta' %}')">
                <h4>Buscar en Sico</h4>
            </button>
        </div>
        <div class="col-md-4 text-center">
            <button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'ingresarSico' %}')">
                <h4>Ingresar Información al Sistema</h4>
            </button>
        </div>
        <div class="col-md-4 text-center">
            <button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'listadeingresos' %}')">
                <h4>Actividades Realizadas</h4>
            </button>
        </div>
    </div>
    <div class="row" style="margin-top: 50px">
        <div class="col-md-4 text-center">
            <button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'cuadrillas' %}')">
                <h4>Monitoreo de Cuadrillas</h4>
            </button>
        </div>
        <div class="col-md-4 text-center">

```

```

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'reportes' %}')">
  <h4>Reportes</h4>
</button>
</div>
<div class="col-md-4 text-center">
  <button type="button" class="btn btn-link btn-lg btn-block">
    <h4>Avances de Obra</h4>
  </button>
</div>
</div>
{% endblock %}
{% block footer %}
  <!--footer-->
{% endblock %}

```

- Se utiliza el siguiente esquema para los ficheros estáticos del proyecto de entre los cuales se destaca el uso de JQuery y Bootstrap para añadir dinamismo y diseño a las paginar html.

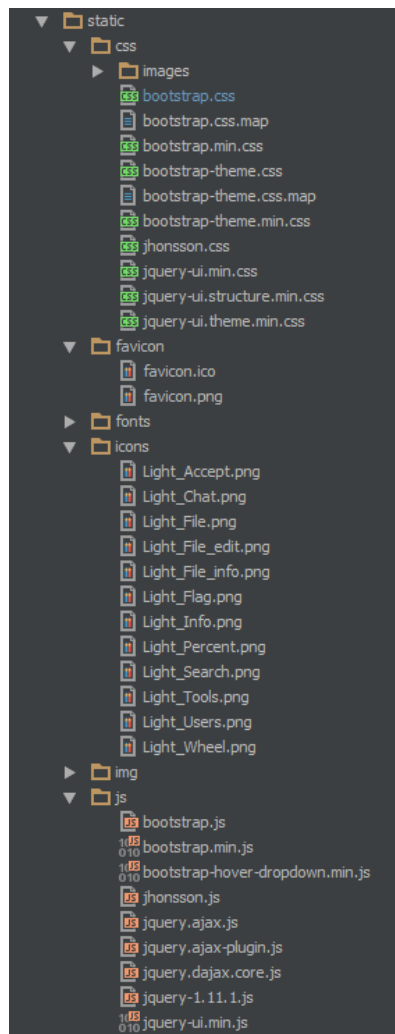


Figura 25. Detalle de la estructura de archivos estáticos del proyecto

Elaborado por: Jhonsson Córdova

- Finalmente se sincroniza con la base de datos los modelos descritos y se pone en marcha el servicio para comprobar el funcionamiento de la solución al primer caso de uso

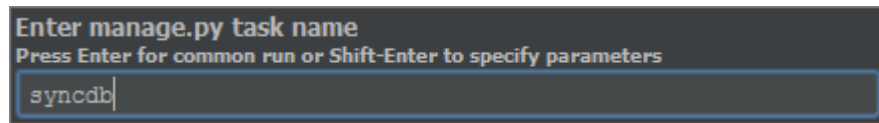


Figura 26. Sincronización de Base de Datos

Elaborado por: Jhonsson Córdova

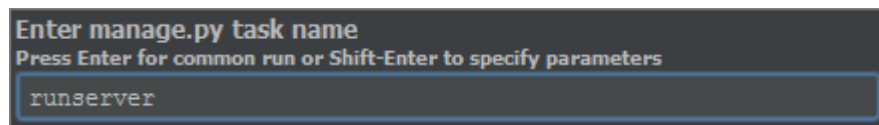


Figura 27. Levantar instancia del servidor

Elaborado por: Jhonsson Córdova

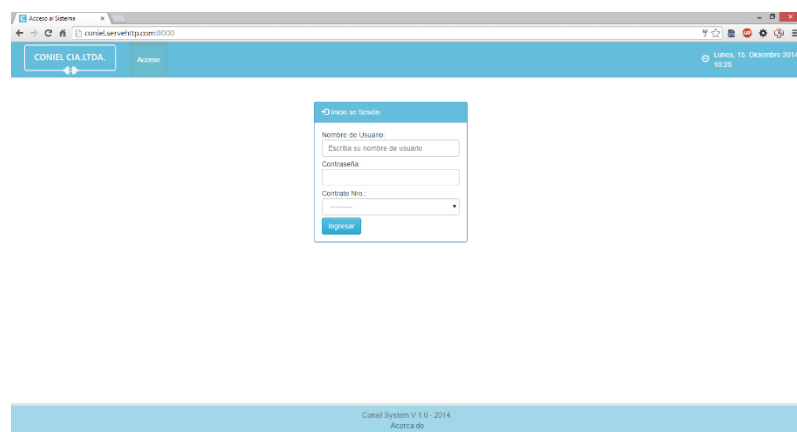


Figura 28. Implementación del Caso de Uso, Login de Usuario

Elaborado por: Jhonsson Córdova

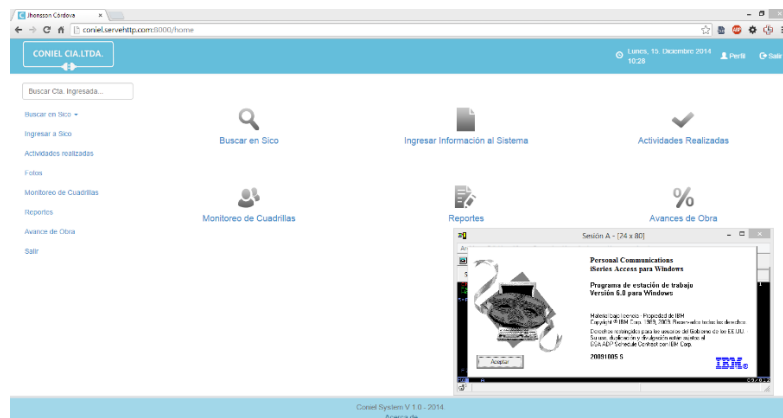


Figura 29. Inicio de sesión

Elaborado por: Jhonsson Córdova

3.4 PRUEBAS Y CONTROL DE CALIDAD DE LA SOLUCIÓN

Esta parte del módulo define la calidad de las características de la aplicación web, además de especificar las pruebas que se ejecutarán a cada solución.

3.4.1 REQUERIMIENTOS DE CALIDAD

Para la especificación de los requerimientos de calidad del sistema seguiremos el estándar ISO/IEC 9126 en cual define un Modelo para la calidad de la aplicación independiente de la tecnología que se esté usando, el cual especifica seis características de calidad las cuales detallamos a continuación:

3.4.1.1 Funcionalidad

La calidad de la funcionalidad se la evalúa analizando si el sistema cumple o no con las funciones detalladas en la fase de análisis y diseños del sistema donde se detallan los requisitos de la aplicación.

3.4.1.2 Confiabilidad

Define las condiciones específicas bajo las cuales el sistema va a mantener el rendimiento. Estas pruebas de confiabilidad se basan en probar la aplicación web para verificar y validar posibles errores que se puedan dar antes de que la solución sea implementada.

La solución propuesta en este proyecto ha sido probada en diferentes situaciones, tratando de detectar posibles errores en cualquier estado posible, para luego proceder a su depuración.

3.4.1.3 Usabilidad

Esta prueba permite identificar el esfuerzo que el usuario debe emplear para la utilización de la página web. En vista de que es uno de los aspectos más importantes, se lo revisará a profundidad haciendo énfasis en las siguientes características:

- **Tiempos de Respuestas**

La aplicación fue desarrollada para que los tiempos de respuesta en los procesos que realiza la compañía sean más rápido, y por más solicitudes que realice el usuario a través del sistema, el mismo responderá de manera inmediata.

- **Eficiencia de interacciones**

Esta característica define que el usuario podrá tener acceso al menú principal en todo momento, teniendo en cuenta esto en la aplicación web se puede visualizar el menú principal en todo momento, el mismo que estará ubicado en la parte izquierda de las páginas.

- **Colores**

La aplicación contiene armonía de colores, los que se usan son: celeste claro, blanco, gris (para las interfaces y botones), negro (para las fuentes).

- **Disposición del texto**

Las fuentes utilizadas son del tipo Sans-Serif, Monospace, Helvetica Neue, Arial, Courier New que son fuentes que no tienen terminaciones curvas por lo que son fáciles de entender. El color de la fuente es negro en la mayoría de las páginas de la aplicación.

- **Estructura de navegación**

La aplicación tiene facilidad de navegación entre las páginas, ya que la estructura del menú principal se visualiza en la parte izquierda de las páginas, de esta manera es más fácil para el usuario ya que no tendrá dificultad de recordar que páginas visitó.

3.4.1.4 Eficiencia

Esta característica se refleja en los niveles de rendimiento de la aplicación y los recursos que usa, debido a que la aplicación es nueva no se puede tener con exactitud el análisis de carga que puede darse en un ambiente ya de producción, pero tomando en cuenta el ambiente de prueba al que ha sido sometido, la aplicación tiene un alto rendimiento de los procesos para lo que ha sido desarrollada.

3.4.1.5 Portabilidad

Esta característica define la facilidad que posee la aplicación para ser trasladada de un ambiente a otro. Tomando en consideración esto, se podría decir que la aplicación es portable ya que puede ser accedida desde cualquier navegador siempre y cuando se encuentre dentro de la red de la compañía.

BIBLIOGRAFÍA

- Álvarez, M. (19 de Noviembre de 2003). *Desarrollo Web*. Obtenido de <http://www.desarrolloweb.com/articulos/1325.php>
- Álvarez, M. (2012). <http://www.desarrolloweb.com/>. Obtenido de <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
- Bahit, E. (2012). *Python para principiantes*. Autoedición.
- Cárdenas, L., & Gracia, J. (11 de Mayo de 2009). *Web Stilo*. Obtenido de <http://www.webestilo.com/javascript/js00.phtml>
- Coniel Cia.Ltda. (12 de Abril de 2013). Datos Informativos de Coniel Cia.Ltda. Machala, El Oro, Ecuador.
- Denzer, P. (23 de Octubre de 2002). *Profesores Utf's*. Obtenido de <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>
- 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
- 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
- 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: Historia, Principios Básicos y Clientes Web*. España: Club Universitario.
- Martínez, R. (2 de 10 de 2010). *Postgresql.org.es*. Obtenido de http://www.postgresql.org.es/sobre_postgresql
- Martínez, R. (2 de Octubre de 2010). *PostgreSQL-ES*. Recuperado el 15 de Mayo de 2014, de http://www.postgresql.org.es/sobre_postgresql
- Muñoz, F. (10 de Mayo de 2011). *SlideShare*. Obtenido de <http://es.slideshare.net/fjosemunoz/aplicacion-web-presentacion-impress>
- Murphey, R. (2013). *Fundamentos de JQuery*. OpenLibra.
- Pavan, B. (28 de Mayo de 2013). *Bitelia*. Obtenido de <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>
- 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
- Refsnes Data. (2011). <http://www.w3schools.com/>. Obtenido de http://www.w3schools.com/bootstrap/bootstrap_intro.asp
- Rodríguez Guerra, J. (25 de Enero de 2014). *Computer Hoy*. Obtenido de <http://computerhoy.com/noticias/internet/descubre-que-es-django-framework-web-moda-8641>
- Sánchez, J. (29 de Septiembre de 2006). *Jordisan.Net*. Obtenido de <http://jordisan.net/blog/2006/que-es-un-framework/>

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

Wium Lie , H., & Bos, B. (2005). *Cascading Style Sheets – designing for the Web* ((3rd ed.) ed.). Noruega: Addison-Wesley.