

Sistema Web de Gestion de Consultorios Medicos

Universidad Nacional de Salta



Facultad de Ciencias Exactas
Seminario de Computación

Alumno: Ricardo Daniel Quiroga
Director de Tesis: Ernesto Sanchez
Proyecto Especifico
Octubre de 2014

Las ideas se piensa, se debaten y se roban.

Índice general

1. Introducción	1
1.1. Objetivos Generales	1
1.2. Resumen	1
1.3. Organización del Informe	2
2. Elección de la Tecnología	3
2.1. ¿Por qué Web y no Desktop?	3
2.2. Apache	4
2.3. mod_wsgi	4
2.3.1. WSGI	4
2.4. PosgreSQL	5
2.5. Python	5
2.5.1. La Filosofia detras de Python	6
2.5.2. Baterias Incluidas	7
2.5.3. Implementaciones	7
2.6. Django	7
2.6.1. MVC	7
2.6.2. Django y el MVT	9
2.6.3. El Modelo	9
2.6.4. La Vista	10
2.6.5. La Plantilla	11
2.6.6. La Configuración de Rutas	12
3. Etapas del desarrollo de Proyecto	15
3.1. Elección del Tema	15
3.1.1. Elección de Tecnología	15
3.1.2. Elección del Tema y Funcionalidades	15
3.2. Análisis de Requisitos y Búsqueda de Información	16
3.3. Funcionalidades Incluidas y Descartadas	16
3.3.1. Funcionalidades Incluidas	16
3.3.2. Funcionalidades no Incluidas	17
3.4. Seguimiento y desarrollo del Proyecto	17
4. Antecedentes	19
4.1. Historia Clínica	19
4.2. Problemas del Sistema Actual	22
4.2.1. Almacenamiento	22
4.2.2. Búsqueda y Localización	22
4.2.3. Deterioro	22
4.2.4. Otros Problemas	22
4.2.5. La Solución Planteada	22
4.3. Gestión de Turnos	23

5. El Proyecto	25
5.1. Motivación	25
5.2. Descripción del Proyecto	25
5.3. Arquitectura de la Aplicación	25
5.4. Modulo Usuarios	25
5.4.1. Modelos	26
5.4.2. Usuarios y Permisos	27
5.4.2.1. Usuarios no Registrados	27
5.4.2.2. Paciente	27
5.4.2.3. Medico	27
5.4.2.4. Administrativo	28
5.4.2.5. Admin	28
5.5. Modulo Gestión de Turnos	29
5.5.1. Definición del Modelo	29
5.5.2. Gestionar Datos de Usuarios	31
5.5.3. Mensajería Interna	31
5.5.4. Asignación Especialidades Medicas	31
5.5.5. Asignación de Turnos	31
5.5.6. Diseño de Modelos Para la Gestión de Turnos	31
5.5.6.1. BussinesHour(Horario De Atención)	31
5.5.6.2. DayOffAttention (Días de Atención)	32
5.5.6.3. Turn (Turno)	32
5.5.6.4. Consideraciones	32
5.6. Modulo Historia Clínica	33
5.6.1. ¿Que es una Historia Clínica?	33
5.6.2. La Historia Clínica en la Ley Argentina	33
5.6.3. Funcionalidades	34
5.6.4. Definicion de Modelos	35
5.6.5. Modelos Básicos	35
5.6.5.1. Image	35
5.6.5.2. MedicalConsultation (Consulta Médica)	36
5.6.5.3. MedicalPrescription (Receta Médica)	36
5.6.5.4. Antecedentes Perinatales	36
5.6.5.5. ToxicHabit (Hábitos Tóxicos del Paciente)	37
5.6.6. Modelos de los diferentes tipos de Examen	37
5.6.6.1. BasicExam (Examen Básico)	37
5.6.6.2. HeadExam (Examen de Cabeza)	38
5.6.6.3. NeckExam (Examen de cuello)	38
5.6.6.4. PFTSEExam (Examen de Piel Faneras y Tejido Subcutáneo)	38
5.6.6.5. OsteoArticularExam (Examen Osteo Articular)	38
5.6.6.6. RespiratorySystemExam (Examen del Sistema Respiratorio)	39
5.6.6.7. CardiovascularSystemExam (Examen del Aparato Cardiovascular) .	39

6. Instalacion y Configuracion	41
6.1. Requerimientos	41
6.1.1. Requerimientos de Hardware	41
6.1.2. Requerimientos de Software	41
6.2. Apache	42
6.2.1. Instalacion en Limpio	42
6.2.2. Instalacion mediante WAMP, LAMP, MAMP, WAPP	42
6.2.3. Configuracion	42
6.2.4. Instalacion de PosgreSQL	43
6.2.5. Creacion de la Base de Datos	43
6.3. Instalacion de Python	44
6.3.1. Probando Python	44
6.4. Instalar Django	45
6.5. Instalando el Resto de Las Dependencias	45
6.5.1. psycopg2	45
6.5.2. ReportLab	45
6.5.3. easy-thumbnails	45
6.5.4. django_extensions	46
6.5.5. django_cron	46
6.5.6. Descargar e Instalacion de mod_wsgi	46
6.5.7. Cargando el Modulo en Apache	46
6.6. Configuracion del Proyecto	46
6.7. Configuracion Inicial	48
7. Guia de Referencia	51
7.1. Organizacion de la Aplicacion	51
7.1.1. Menu Principal	52
7.1.2. Menu Secundario	52
7.1.3. Cuerpo de la Aplicacion	52
7.1.4. Informacion de Usuario	52
7.2. Panel de Usuario No Registrado	52
7.2.1. Registrarse	52
7.2.2. Iniciar Session	55
7.3. Panel de Usuario Paciente	55
7.3.1. Listado de Medicos	55
7.3.2. Mensajes	56
7.3.3. Opciones	58
7.4. Panel de Usuario Administrativo	58
7.4.1. Pacientes, Medicos, Administrativos	59
7.5. Panel de Usuario Medicos	61
8. Conclusion y Mejoras	63
8.1. Resultado	63
8.1.1. Ventajas Percibidas	63
8.1.2. Desventajas Percibidas	64
8.2. Futuras Mejoras	64

Índice de figuras

2.1. Diagrama del Patrón MVC Modelo Vista Controlador	8
2.2. Generaciones de Herramientas de Desarrollo Web	9
2.3. El patrón Modelo Vista Template de Django	9
4.1. Almacenamiento Fisico de Archivos	19
4.2. Modelo Historia Clinica Ministerio de Salud de La Nacion Pag 1	20
4.3. Modelo Historia Clinica Ministerio de Salud de La Nacion Pag 2	21
5.1. Diagrama con modelos que componen el modulo Usuarios	26
5.2. Vista del Panel Administracion provisto por Django	28
5.3. Modulo Gestión Turnos - Diagrama Modelos correspondiente a la Gestión de Turnos	30
5.4. Modulo Gestión Turnos - Modelos Adicionales	30
5.5. Historia Clínica Modelos Básicos	35
5.6. Historia Clínica Modelos de Estudios	37
5.7. Historia Clínica Modelos de Estudios Continuación	38
6.1. Ejecutando Python en la Terminal	44
7.1. Organizacion Espacial del contenido de la aplicacion	51
7.2. Formulario Registro Paciente	53
7.3. Formulario Registro Paciente	53
7.4. Bandeja de Correo con el mensaje	54
7.5. Cuerpo del mensaje con la informacion de activacion de usuario	54
7.6. Formulario de Activacion de usuario	54
7.7. Listado de Medicos	55
7.8. Formulario de Consulta Online	56
7.9. Mostrar Datos del Medico	56
7.10. Redactar un Mensaje	57
7.11. Bandeja de Entrada	57
7.12. Mostrar Mensaje	58
7.13. Mostrar Administrativo	59
7.14. Vista para busqueda de usuario, en este caso usuarios Pacientes	59
7.15. Mostrar Medico	60
7.16. Vista Listado Turnos Solicitados al Medico.	60
7.17. Mostrar Paciente	61

Capítulo 1

Introducción

El presente trabajo de tesis es para obtener el título intermedio de Computador Universitario perteneciente a la carrera Licenciatura en Análisis de Sistemas (plan 97) de la Universidad Nacional de Salta.

El tema elegido para desarrollar corresponde a un sistema de gestión para consultorios médicos, en el se intento reflejar todos los conocimientos que adquirí durante el cursado de la carrera. En cuanto a las razones para la elección del tema son entre otras, el buscar desarrollar un sistema que por sus dimensiones se proponga como un reto, ya que hasta la fecha solo tenía experiencia en cuanto al desarrollo de pequeñas aplicaciones.

El área de aplicación quedo definida, por la simple razón que tenía contacto con profesionales del área de la Salud.

En cuanto a tecnología quise implementarlo usando algo distinto del tradicional PHP y MySQL para Web, opte por probar una tecnología que no conocía Python, Django y PosgreSQL, la cuales tenía bastante buenas opiniones por parte de terceros y por suerte no decepciono sobre todo Django, que cambio la forma de pensar que tenia a la hora de encarar un proyecto web.

1.1. Objetivos Generales

El objetivo del proyecto de tesis fue el de diseñar y desarrollar un sistema centralizado para el sector de la Salud, específicamente aplicándose al área de consultorios médicos, permitir un mejor seguimiento y control de la evolución de los pacientes mediante la informatización de los diferentes exámenes y consultas que se le realizan al paciente posibilitando la unificación de su historia clínica. Además de también gestionar la asignación de turnos a los pacientes.

Lo que se pretende es brindar un sistema modular y eficiente que permita su fácil aplicación y además de brindar la posibilidad de modificación tanto para adecuación para casos específicos como extensión de sus funcionalidades.

1.2. Resumen

El **Sistema Web de Gestión de Consultorio Médicos**, desde ahora el **Sistema**, está pensado para satisfacer las necesidades de un consultorios médicos o cualquier otra actividad en donde sea necesario almacenar información demográfica de pacientes , historias clínicas, prescripciones

así también como la asignación de turnos.

Al ser un sistema centralizado, se puede acceder a él desde cualquier navegador web actual que cuente con conexión a Internet, lo que permite entre otras cosas:

Disminuir los tiempos de esperas por parte de pacientes a la hora de solicitar ser atendidos, solo necesita solicitar un turno vía web el sistema automáticamente le asignara una fecha y hora acorde a sus requerimientos. Permite a los médicos manejar más fácilmente su agenda para atención de pacientes.

Mejorar el seguimiento de los pacientes por parte de los médicos, centralizando toda su información ya que con ello el médico podrá monitorear la evolución de su paciente donde sea que se encuentre ya que solo necesitara una PC con conexión a Internet.

1.3. Organización del Informe

El informe está organizado en 8 capítulos, entre los que se incluye esta reseña que corresponde al **Capítulo I** la cual es una introducción la cual explica las necesidades que motivaron el desarrollo del mismo y un resumen general de lo que se pretende implementar con el mismo.

En el **Capítulo II** se refiere en cuanto a la tecnología y razones por las cuales se decidió utilizar.

El **Capítulo III** trata sobre las etapas del desarrollo del mismo y las diferentes metas alcanzadas.

El **Capítulo IV** explica la problemática y el funcionamiento de los actuales sistemas aplicados al área.

El **Capítulo V** habla sobre cuestiones técnicas relacionadas con el desarrollo, los módulos y funcionalidades que fueron necesarios implementar, y las soluciones que se plantearon.

El **Capítulo VI** no guía en la implementación del sistema, los requerimientos del mismos y toda la configuración necesaria para lograr una correcta instalación y funcionamiento del mismo.

En el **Capítulo VII** pongo una sencilla visita y explicación de las diferentes áreas del sistema.

Por último el **Capítulo VIII** es una conclusión, que analiza el desarrollo del mismo, perspectivas a futuro y como podría evolucionar el sistema.

Capítulo 2

Elección de la Tecnología

2.1. ¿Por qué Web y no Desktop?

Una aplicación *Desktop*¹ es aquella que requiere ser instalada en el Ordenador (PC) del Usuario, y que es ejecutada directamente por el sistema operativo, ya sea Microsoft Windows, GNU/Linux, Mac OS, etc.

Algunos Ejemplos de Estas Aplicaciones:

- Winamp
- Adobe Photoshop
- iTunes
- Microsoft Office (Word, Excel, Power Point, etc.)

Aunque suelen ser más robustas y estables que las aplicaciones web presentan varios inconvenientes tales como:

- Su acceso solo se limita a la PC donde fue instalada.
- La aplicación es dependiente del sistema operativo que utilice la PC, aunque existen programas multiplataforma no aseguran una compatibilidad completa.
- Requieren una instalación personalizada
- En caso de Actualizaciones requieren que estas se hagan de forma manual en cada PC donde se instalo la aplicación.
- Suelen tener requerimientos especiales de software y librerías para poder funcionar.

Una aplicacion web, es aquella que solo requiere ser instalada en un servidor, su ejecucion requiere únicamente disponer de un ordenador con conexión a internet y un navegador en contraparte de las desktop que requiere que se instale en cada ordenador donde se pretende usar.

Por lo cual brinda una serie de ventajas tales como:

- Portabilidad, se ejecuta desde cualquier ordenador que posea conexión a internet sin depender de software adicional, plataforma y/o sistema operativo.²

¹también llamada aplicación de escritorio

²Esta característica no es del todo cierto algunas aplicaciones requieren determinados complementos instalados en el navegador web o no soportan ciertos navegadores.

- La informacion que se maneja es multiusuario por lo que son especialmente utiles para desarrollar aplicaciones multiusuarios basadas en compartir informacion.
- Consumen muy pocos recursos, por lo que el usuario no necesita tener un ordenador con grandes prestaciones para trabajar con ellas.
- Son faciles de actualizar y mantener.
- Se pueden utilizar en miles de equipos sin limitacion y restriccion alguna.
- Su funcionalidad es independiente del sistema operativo Instalado en el ordenador del usuario.
- No hay problemas de incompatibilidad de version de software ya que los usuarios trabajan con la misma version.

En resumen el sistema por sus caracteristicas podria haberse implementado como un sistema desktop pero se ubiesen perdido las caracteristicas que deseaba para el mismo.

2.2. Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basa inicialmente en código del popular NCSA HTTPD 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y “civilizasen” el paisaje que habían creado los primeros ingenieros de internet.

Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor “parcheado”) suena igual que Apache Server.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (`httpd`) de la *Apache Software Foundation*.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

2.3. mod_wsgi

`mod_wsgi` es un módulo de Apache que provee una interfaz WSGI para correr aplicaciones web escritas en Python sobre Apache, esto es todo lo que necesitas aplicar para que tus archivos `*.py` se ejecuten por medio de un navegador Web.

2.3.1. WSGI

WSGI es el acrónico de *Web Server Gateway Interface* que es una especificación para una simple y universal interfaz entre una aplicación web (en nuestro caso una aplicación escrita en Django) y un servidor web para el lenguaje de programación Python. La especificación WSGI Es

un estandar de Python el cual se describe con detalle en la PEP 33³.

2.4. PosgreSQL

PostgreSQL es un gestor de base de datos relacional que puede correr tanto bajo sistemas operativos Windows como en distribuciones Linux como Red Hat, Suse, CentOS, etc.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

El nombre hace referencia a los orígenes del proyecto como la base de datos (post-Ingres), y los autores originales también desarrollaron la base de datos Ingres.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos.

Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos comprendía las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgres usó muchas ideas de Ingres pero no su código.

2.5. Python

Django está escrito puramente en Python, por lo que obviamente necesitaremos instalar Python que es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License,¹ que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

³<http://www.python.org/dev/peps/pep-0333/>

Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

2.5.1. La Filosofía detrás de Python

Los usuarios de Python se refieren a menudo a la Filosofía Python que es bastante análoga a la filosofía de Unix. El código que sigue los principios de Python de legibilidad y transparencia se dice que es "pythonic". Contrariamente, el código opaco u ofuscado es bautizado como "no pythonic" (*unpythonic*,^{en} inglés).

Estos principios fueron famosamente descritos por el desarrollador de Python Tim Peters en El Zen de Python, algunos de ellos son:

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Aunque lo práctico gana a la pureza.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una y preferiblemente sólo una manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.¹⁵
- Ahora es mejor que nunca.
- Aunque nunca es a menudo mejor que ya mismo.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (namespaces) son una gran idea ¡Hagamos más de esas cosas!

2.5.2. Baterias Incluidas

Python tiene una gran biblioteca estandar, usada para una diversidad de tareas. Esto viene de la filosofía *pilas incluidas (batteries included)* en referencia a los módulos de Python. Los módulos de la biblioteca estandar pueden mejorarse por módulos personalizados escritos tanto en C como en Python. Debido a la gran variedad de herramientas incluidas en la biblioteca estandar, combinada con la habilidad de usar lenguajes de bajo nivel como C y C++, los cuales son capaces de interactuar con otras bibliotecas, **Python es un lenguaje que combina su clara sintaxis con el inmenso poder de lenguajes menos elegantes.**

2.5.3. Implementaciones

En la actualidad existen diversas implementaciones de Python

- **CPython** es la implementación original, disponible para varias plataformas en el sitio oficial de Python.
- **IronPython** es la implementación para .NET
- **Stackless Python** es la variante de CPython que trata de no usar el stack de C www.stackless.com
- **Jython** es la implementación hecha en Java
- **Pippy** es la implementación realizada para Palm pippy.sourceforge.net
- **PyPy** es una implementación de Python escrita en Python y optimizada mediante JIT pypy.org

2.6. Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el paradigma conocido como **Model Template View**. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la **World Company de Lawrence, Kansas**, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt http://es.wikipedia.org/wiki/Django_Reinhardt.

En junio del 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

2.6.1. MVC

Antes de Explicar cómo funciona Django empezare por una breve explicación de el patrón (MVC) Modelo Vista Controlador el cual es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de

diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

De manera genérica, los componentes de MVC se podrían definir como sigue:

El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' actuando como Middleware⁴.

La Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

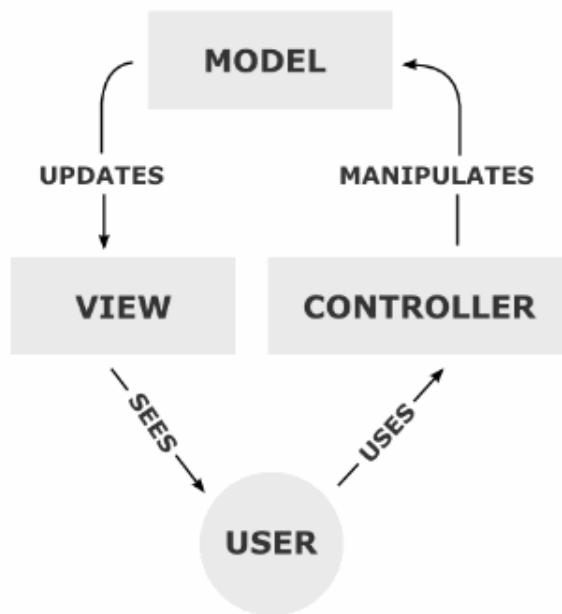


Figura 2.1: Diagrama del Patrón MVC Modelo Vista Controlador

⁴Middleware es el software que proporciona un enlace entre aplicaciones de software independientes. Middleware a veces se llama a la vía que conecta dos aplicaciones y pasa los datos entre ellas. Los Middleware permiten que los datos contenidos en una base de datos puedan ser accedidos a través de otra. Ahorra el tiempo a los programadores.

2.6.2. Django y el MVT

Si hiciéramos una clasificación de Herramientas de desarrollo web, podríamos clasificar a Django como parte de la tercera generación:

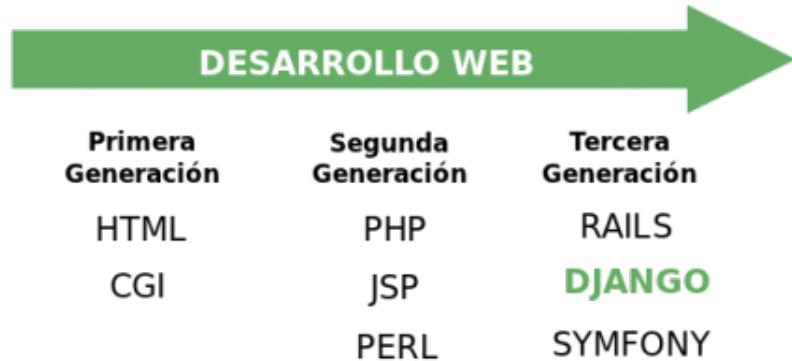


Figura 2.2: Generaciones de Herramientas de Desarrollo Web

Sin embargo mas allá de las clasificaciones que podrían existir, está el entender cómo funciona realmente, al entenderlo se puede llegar a dominarlo.

Dijimos que era un framework MTV (una modificación de MVC, nada que ver con un canal de música), esto se debe a que los desarrolladores no tuvieron la intención de seguir algún patrón de desarrollo, sino hacer el framework lo más funcional posible.

- **El Modelo** en Django sigue siendo el modelo
- **La Vista** en Django se llama Plantilla (Template)
- **El controlador** en Django se llama Vista

Una imagen nos hará entender mejor esta relación:

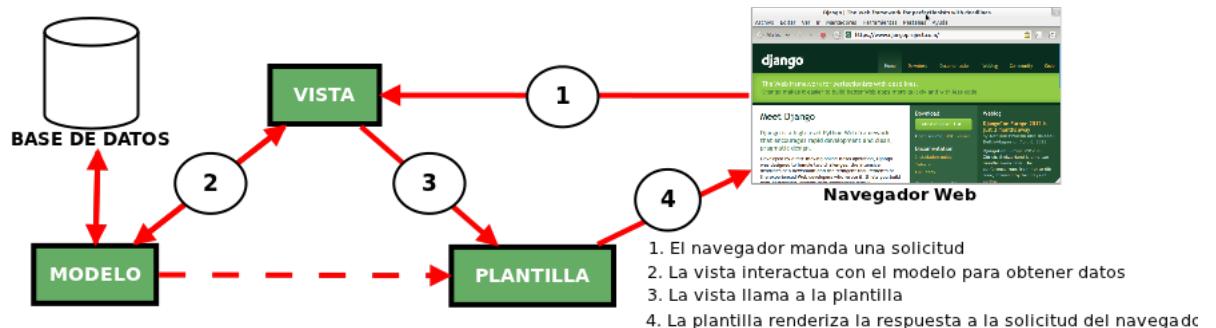


Figura 2.3: El patrón Modelo Vista Template de Django

2.6.3. El Modelo

El modelo define los datos almacenados, se encuentra en forma de clases de Python, las clases definidas son traducidas por Django y este genera las Tablas necesarias para el funcionamiento del modelo dentro de la base de datos, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.

Aquí un extracto del código mostrando cómo se implementa uno de los tantos modelos con los que trabaja el Sistema

```
1 class Message(models.Model):
2     """
3         Clase Para Manejar mensajes entre usuarios
4     """
5     from_user = models.ForeignKey(User, related_name='from_user')
6     to_user = models.ForeignKey(User, related_name='to_user')
7     date = models.DateTimeField("Fecha y Hora", auto_now_add=True)
8     issue = models.CharField("Asunto", max_length=125, default='')
9     content = models.TextField("Cuerpo del Mensaje")
10    read = models.BooleanField("Leido", default=False)
11
12
13    class Meta:
14        db_table = "Messages"
15        verbose_name = "InboxMessage"
16        verbose_name_plural = "InboxMessages"
```

5
6

2.6.4. La Vista

la vista se presenta en forma de funciones en python, su propósito es determinar qué datos serán visualizados, entre otras cosas más que iremos viendo conforme avanzamos con el curso. el orm de django permite escribir código python en lugar de sql para hacer las consultas que necesita la vista.

La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.

Aquí muestro una vista sencilla que realiza una consulta base de datos que listara todos los usuarios que sean médicos.

```
1 def patient_show_medics_list(request):
2     """
3         Muestra el listado de Medicos
4     """
5     mi_template = get_template('Patients/GestionTurnos/medics-list.html')
6     dict = generate_base_keys(request)
7
8     if is_patient(request.user):
9         dict['medics'] = UserInformation.objects.filter( \\"
```

⁵Como algunos lo notaran la variable `from_user` del modelo internamente es una relación 1:M dentro de la base de datos.

⁶La Clase interna Meta define atributos especiales como `db_name` que hace referencia a como se llamara la tabla dentro de la Bases de Datos.

```

10                     user__groups__name='Medico')
11         html_cont = mi_template.render(Context(dict))
12         return HttpResponseRedirect(html_cont)
13
14     else:
15         #si hay un usuario logueado intentanto acceder sera enviado a una
16         # pagina de error
17         path = request.META['PATH_INFO']
18         return HttpResponseRedirect("/restricted-access%s" %path)

```

Aunque es un ejemplo sencillo podemos apreciar el potencial de Django, como vemos no vemos ningún código SQL, pues bien dicho código SQL se ejecuta internamente nos aleja del problema de las restricciones de la Base de Datos ya sea que usemos PosgreSQL (como en este sistema), MySQL, SQLServer o SQLite nosotros solo escribiremos código Python, El framework se encargara de traducir esa instrucción al motor de bases de datos correspondiente que estemos usando.

```

dict['medics'] = UserInformation.objects.filter( \\
                                                 user__groups__name='Medico')

```

Traducido a SQL terminaríamos con algo tan horrible como esto:

```

SELECT * FROM UserInformation as Info
INNER JOIN User ON Info.username = User.username
INNER JOIN GroupsByUsers ON User.username = GroupsByUsers.username
...

```

2.6.5. La Plantilla

La plantilla es básicamente una página HTML con algunas etiquetas extras propias de Django, en si no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, etc).

La plantilla recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas permiten que sea flexible para los diseñadores del frontend, pueden Extenderse a partir de otras plantillas incluso tiene estructuras de datos como if, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python.

Esto permite que la lógica del sistema siga permaneciendo en la vista. Aquí la vista para Iniciar sesión:

```

{% extends 'base.html' %}

{% block style %}
<link type="text/css" rel="stylesheet" media="all"
      href="/media/css/fancy-forms.css" />
<link type="text/css" rel="stylesheet" media="all"
      href="/media/css/gradient-buttons.css" />
<link type="text/css" rel="stylesheet" media="all"
      href="/media/css/messages.css" />
{% endblock %}

```

```

{%
    block contenido %}
<br /><br /><br />
{%
    if not_login %}
    <div class="fancy-form-white" style="width: 350px;
        margin: 0 auto;">
        <h3 class="title">Iniciar Session</h3><br />
        <form action"." method="POST">
        <table style="margin: 0 auto; width: 330px;" >
            <tr>
                <td><label for="username">Usuario:</label></td>
                <td><input type="text" name="username" value="">
                    tabindex="1" id="username"></td>
                <td rowspan="2">
                    <input type="submit" value="Login" tabindex="3"
                        class="grad-button-blue" style="height: 50px;">
                </td>
            </tr>
            <tr>
                <td><label for="password">Contraseña:</label></td>
                <td><input type="password" name="password" value="">
                    tabindex="2" id="password"></td>
            </tr>
        </table>
        </form>
        <br />
    </div>

{%
    if login_error %}
        <br />
        <br />
        <div class="alert">Alerta: Error Usuario y/o Contraseña Incorrectos</div>
{%
    endif %}

{%
    else %}
        <div class="alert">Alerta: Usted ya ha iniciado session con el
            usuario <strong>{{ username }}</strong></div>
        <br />
        <a href="/logout">Cerrar Session</a>
{%
    endif %}
{%
    endblock %}

```

2.6.6. La Configuración de Rutas

Django posee un mapeo de URLs que permite controlar el despliegue de las vistas, esta configuración es conocida como URLConf. El trabajo del URLConf es leer la URL que el usuario solicitó, encontrar la vista apropiada para la solicitud y pasar cualquier variable que la vista necesite para completar su trabajo. El URLConf está construido con expresiones regulares en Python y sigue la filosofía de Python: Explicito es mejor que implícito. Este URLConf permite que las rutas que maneje Django sean agradables y entendibles para el usuario.

Fragmento del archivo urls.py del Proyecto

```
(r'^$', base_views.index),
```

```
(r'^index/$', base_views.index),
(r'^login/$', base_views.login),
(r'^logout/$', base_views.logout),
(r'^change-password/$', base_views.change_password),
(r'^restricted-access/$', base_views.restricted_access),
(r'^restricted-access/(.+)/$', base_views.restricted_access),
```


Capítulo 3

Etapas del desarrollo de Proyecto

3.1. Elección del Tema

Al momento de elegir el tema del seminario se me presento un numero de problemas ya que primero no sabía que área se iba a aplicar, quería realizar algo distinto, y algo que presente un reto a nivel de programación, ahora que lo pienso el haber querido esto último me termino saliendo caro por el tiempo que termine ocupando para terminar el proyecto.

3.1.1. Elección de Tecnología

Al momento de elegir no tenía casi experiencia desarrollando aplicaciones web por lo que al menos ya estaba decidido que iba a ser web la aplicación, digamos fue mas una cuestión de moda, en esos días había empezado el boom de la web 2.0 por lo que fue mas por una moda.

3.1.2. Elección del Tema y Funcionalidades

Sé que suena repetitivo la elección del tema y las funcionalidades que se iban a desarrollar fue una de las parte más difícil del proyecto, a la hora de decidir que quería hacer, aunque bien el tema elegido salió de la por una interconsulta con un posible cliente que al final no llegó a mas que intercambiar un par de emails, pero fundo las bases y me parecio era una buena idea, consulte un poco los seminarios anteriores aunque había temas relacionados el desarrollo no era muy extensos ni aplicado exactamente al área que pretendía.

Consultando algunos seminarios de tesis de otros alumnos aplicadas al tema ¹ detecte, que había desarrolladas aplicaciones dedicadas a la facturación, atención, y otras áreas. Con ello el ámbito que abarcaría el sistema ² trataba de juntar dos partes el seguimiento o evolución del paciente lo cual se registra mediante el documento de **Historia Clínica**, lo que era bueno porque al menos no iba a reinventar la rueda. Llegado el momentos de presentar el tema a mi director de tesis me dijo que era poco, supongo que fue porque no supe explicarlo bien, en fin había que agregar algo mas y en esos día tuve que asistir a interconsulta medica, cuando entre en el consultorio note que la secretaria estaba con una hoja de cálculos en MS Excel registrando y comprobando los turnos para el día, automáticamente me di cuenta de una gran falencia, sin importar lo automatizado que estuviese el consultorio las tareas más simples como la de asignar turnos se seguía realizando de una manera muy poco práctica, la secretaria tenía una hoja de cálculo pre formateada con los horarios de cada día, la cual debía duplicar para registrar un nuevo día, a partir de ahí surgió la idea de implementar una funcionalidad que agilice dicho problema y por qué no brindar una plataforma

¹Hago referencia a la administración de consultorio médico

²Hago referencia al sistema que se desarrollo como parte de este proyecto de tesis

virtual para que los pacientes pudiesen solicitar turnos, dejando de depender tanto de los teléfono, con ello el tema y las funcionalidades principales quedaron definidas.

3.2. Análisis de Requisitos y Búsqueda de Información

Luego de elegido el tema llegó el momento de averiguar cómo funcionaba todo, aunque esto sea un software académico quería que después de desarrollado al menos fuera útil aunque sea como sistema de referencia y no solo dejarlo morir apilado como un seminario mas, como suele suceder.

Sobre la parte de gestión de turno el problema era algo sencillo, ya que todo el mundo alguna vez tuvo que sacar turno para ya sea ser atendido o hacer un determinado trámite, la dificultad radicaba mas en que no tenía idea como modelizar un problema y los datos que se deberían manejar.

En cuanto a historia clínica, hubo que visitar algunos centros médicos y especialistas y solicitar información acerca de como ellos manejaban las historia clínicas de sus paciente, que tipo de información era imprescindible registrar en ella y cual podría ser secundaria, fue algo tedioso digamos ya que la información sobre los pacientes es información sensible y no es de libre acceso.

Además sumando lo que había recabado y gracias a un amigo que me paso un par de modelos que consiguió y sumado un poco de búsqueda de información referente en la web, logre generar un panorama más o menos concreto de lo que iba a hacer, la verdad me hubiese gustado tener algún especialista afín por cada tema que consultaba sobre los diferentes estudios pero supongo eso era pedir demasiado.

3.3. Funcionalidades Incluidas y Descartadas

El nombre del tema o las funcionalidades principales elegidas no dice mucho en si sobre que se iba a desarrollar sobre todo en la parte de historia clínica ya que esta área abarcaba un gran número de posibles estudios que podrían ser incluidos en este documento los cuales algunos son muy específicos del área de estudio, por ejemplo un odontograma solo seria de interés para un medico odontólogo y en si mucha relevancia no tendría al momento de tratar otras enfermedades, por lo que había que definir que estudios se deberían incluir como mínimo y cuales se descartarían ³.

3.3.1. Funcionalidades Incluidas

Luego de revisar los modelos en papel⁴ en su mayoría de diferentes organismos médicos que consulte se determino que se incluirían los siguientes estudios:

- Hábitos Tóxicos.
- Antecedentes Perinatales ⁵
- Grupo Familiar ⁶.
- Almacenamiento de Imágenes Relacionadas con Diferentes Estudios.
- Examen de Cabeza.

³Al menos en esta versión, téngase en cuenta que esto es un software desarrollado con fines académicos aunque es factible el desarrollo de nuevos módulos, su implementación y aplicación en casos reales.

⁴Hago referencia a las historias clínicas que por lo general son almacenadas en papel normalmente en archivos pre formateados e impresos.

⁵Hace Referencia a antecedentes del nacimiento

⁶Esto no es un Examen en si sino mas bien que sirve para consulta en caso del que el paciente posea enfermedades Hereditarias.

- Examen de Cuello.
- Examen Signos Vitales.
- Examen de Piel Faneras y Tejido Subcutáneo.
- Examen Osteo Articular.
- Examen Sistema Respiratorio.
- Examen del aparato Cardiovascular.

Adicionalmente el sistema también registran las Interconsultas Medicas y los Medicamentos que fueron recetados al paciente.

3.3.2. Funcionalidades no Incluidas

Si tuviese que nombrar que estudios no se implementaron no terminaría mas aquí hablando del modulo de Historia Clínica, en esta sesión mas que nada contemplería funcionalidades que en un principio estaban previstas ⁷ pero a medida que el desarrollo del sistema avanzó perdieron importancia por su aplicación o por cuestiones de falta de tiempo, aquí enumeró algunas de las funcionalidades que se descartaron:

- Gestión de Turnos - Definir Especialidad del Médico para atención.
- Gestión de Turnos - Especificar Habitación del Consultorio para el Turno.
- Historia Clínica - Subir Archivos.
- Historia Clínica - Análisis de Laboratorios.
- Historia Clínica - Conectar con el Sistema Vademécum.

No son todas las funcionalidades que se descartaron pero considero que son las más importantes o relevantes.

3.4. Seguimiento y desarrollo del Proyecto

El seguimiento y desarrollo del proyecto digamos no fue continuo fue más algo bastante accidentado si podría decirse, y bueno hasta la fecha solo había programado pequeñas aplicaciones, enfrentaba el problema de que me topaba con una metodología diferente de lo que manejaba, a la hora de encarar lo relacionado con la programación.

Al menos a la hora de mantener el código agradezco haber conocido en unas charlas unas semanas antes la herramienta de control de versiones GIT ⁸ que digamos a simple vista te permite gestionar las diferentes versiones de tu código incluso volver a versiones anteriores si fuese necesario. Este proyecto de seminario incluido tanto el informe del mismo como su código fuente se encuentran hospedados en un repositorio online de git en la siguiente dirección web url <http://www.github.com/l2radamanthys/SGCM.git> allí si desean pueden descargar la última versión del mismo ya que esta alojado en un repositorio público.

En cuanto a programación como feedback puedo decir lo siguiente:

⁷ las mismas no estaban especificadas por escrito en el anteproyecto sino más bien son características que tenía pensadas

⁸ www.git-scm.com

El tamaño del sistema me supero enormemente, eso se nota más cuando miro hacia atrás y considero el tiempo que me consumió estar programando, considero que hubiese sido mejor encarar el sistema en un grupo de 2 o más personas.

Más de lo mismo, querer hacerlo todo uno desde 0 tampoco sirve, menos si no planificas y quieres ponerte a programar desde el primer día, a razón de eso termine dándome cuenta a mitad de proyecto que tenía que reconstruí todo, porque nunca me senté a pensar como debía estructurar el sistema desarrollado, se desarrollo sobre la marcha y ahí están las consecuencias.

Capítulo 4

Antecedentes

4.1. Historia Clinica

Actualmente en la mayoria de los centros de salud la informacion que compone la Historias Clinicas de cada paciente es almacenan mediante documentos fisicos, en su mayoria totalmente elaborados a mano, en otros casos usando plantillas con campos preformateados (Como se puede Observar en las Figuras 3.1 y 3.2), lo cual genera varios problemas.

En cuanto al software existente aplicado a la manipulacion de historia clinica, este suele ser muy incompleto y abocado a la especialidad del medico, es por ello que si esta informatizado cada area suele manejar sistemas que terminan siendo incompatibles entre si.



Figura 4.1: Almacenamiento Fisico de Archivos

Centro de atención

DNI

- Nombre
- Apellido
- Calle
- Barrio
- Partido
- Localidad
- Nro.
- Teléfono
- Fecha de nacimiento
- Nacionalidad
- Lugar de Origen

Estado civil

- Soltero Separado
- Unión estable Divorciado
- Casado

Trabajo

- Si No
- Dependiente Cuenta propia
- Ocasional Tipo de ocupación

Obra social

- Si No
- ¿Cuál?
- Nº

Educación

- Analfabeto
- Primaria incomp.
- Primaria comp.
- Secundario incomp.
- Secundario comp.
- Terciario/univer.

Condiciones de vida

- Rancho o casilla
- Vivienda móvil
- Pieza de inquilinato
- Lugar no apto para vivienda
- 3 o más pers. por cuarto
- Hijo 6 a 12 años sin escolaridad
- Baja capacidad subsistencia
- Letrina
- Cocina / calefacción
- Gas natural
- Agua corriente
- Agua de pozo
- Baño interno o descarga
- Baño público
- Baño afuera
- Cloacas
- Pozo ciego
- Gas envasado
- Kerosén
- Electricidad
- Leña o carbón

Situación de vulnerabilidad

- Maltrato familiar
- Alcoholismo flia.
- Adicciones flia.
- Madre sola
- Situación de calle
- N.B.I.

Consumo de sustancias

Tabaquismo

- ¿Fuma? Desde (edad) cant.
- Nunca Fumó fumador pasivo
- Dejó de fumar fecha
- ¿Cuántos minutos pasan desde que se levanta hasta que prende un cigarrillo?
- ¿Piensa dejar de fumar?
- En los próximos 6 meses
- En el próximo mes

Alcohol

- ¿Bebe?
- ¿Lo criticó alguien porque tomaba mucho?
- ¿Sintió ganas de disminuir la bebida?
- ¿Se sintió alguna vez culpable por tomar mucho?
- ¿Toma algo a la mañana para sentirse mejor?

Drogas

- ¿Consumió alguna vez? Veces por sem. ¿Cuál?
- Diuresis
- ¿Se inyectó alguna vez? Catarsis
- Sueño

Datos personales

Antecedentes personales

- HTA
- DIABETES
- Enfermedad coronaria / IAM
- ACV
- EPOC ASMA
- Alergia
- Enfermedad reumática
- Enf. Oncológico
- TBC
- VIH
- Chagas
- ITS
- Psicopatológicos
- Neurológicos
- Otros

Menarca IRS

- G P A C
- Ritmo menstrual FUM
- Fecha último PAP
- MAC Si No
- Dispareunia Si No
- Prob. Sexuales Si No

Eyaculación precoz Disfunción erétil

- Si No
- Si No

Antecedentes familiares

- HTA
- Cardiopatías
- DIABETES
- ACV
- Ca. de colon
- Ca. de pulmón
- Ca de mama
- Otra enf. Oncol.
- Consumo drogas
- Abuso alcohol
- Depresión
- Otras

madre padre herm.

Internaciones / operaciones / accidentes Si No

Indicar motivo y fecha

¿ALERGIA a medicamentos? Si No ¿Cuál?

Medicamentos (principio activo) Presentación Dosificación

1	5
2	6
3	7
4	8

Observaciones

Familograma

Matrimonio Pareja no casada

Separación Divorcio

Hombre Mujer Hijos

Gemelos Muerte

Relación estrecha Relación conflictiva

Observaciones

Médico de cabecera

Fecha de apertura

Figura 4.2: Modelo Historia Clinica Ministerio de Salud de La Nacion Pag 1

*Recomendaciones orientadas sujetas a modificaciones y actualización según guías de práctica de la autoridad sanitaria



Figura 4.3: Modelo Historia Clinica Ministerio de Salud de La Nacion Pag 2

4.2. Problemas del Sistema Actual

4.2.1. Almacenamiento

A medida que crece el numero pacientes y la informacion que va anexando a cada Archivo se va necesitando mas espacio fisico para almacenar dicha informacion. Por ello con el paso del tiempo las instalaciones dedicadas a tal fin suelen verse colapsadas por los grandes volumenes de informacion que deben manejar.

4.2.2. Busqueda y Localizacion

La Busqueda de expedientes se puede agilizar un poco utilizando una buena organizacion, el problema es que la mayoria de los edificios para tal fin, suelen estar saturados por grandes volumenes de archivos fisicos, Por lo que encontrar un archivo requerido suele ser una tarea costosa y lenta.

4.2.3. Deterioro

En lo que hace a la conservación propiamente dicha, el combate de los problemas habituales que se derivan de las condiciones climáticas, de la humedad, de las plagas, del deterioro natural del papel, especialmente por su fabricación con celulosa desde hace dos siglos, es una constante que, pese a sus avances, no ha encontrado soluciones definitivas. Por ende, una preocupación común a los archivos y bibliotecas, es encontrar remedios prácticos y asequibles para asegurar la preservación de sus acervos. De lo anterior se desprende la necesidad, en el nivel nacional, de procurar el establecimiento de políticas y normas sobre conservación en las instituciones públicas y privadas dedicadas a la protección del patrimonio.

4.2.4. Otros Problemas

Otros problemas que acarrean el uso de archivos fisicos son:

- Pérdida, alteración o daño de documentos importantes
- Gasto excesivo en fotocopias
- Altos costos de personal para administrar, suplir, mantener y recuperar el archivo fisico
- Costos asociados al transporte de documentos ya sea interna o externamente
- Costos asosiados al espacio fisico requerido para su almacenamiento
- Falta de condiciones adecuadas para el almacenamiento de documentos como: ventilación, humedad, temperatura
- Falta de respaldo adecuado en caso de catastrofe como incendio, inundación o terremoto

4.2.5. La Solucion Planteada

Por ello este era un escenario perfecto donde Es Necesario informatizar el Actual Sistema, lo cual solucionaria los 2 principales problemas del mismo que son el Excesivo espacio de almacenamiento y el lento trabajo de busqueda ademas de:

- Reducir costos de Personal Administrativo, ya que las busquedas y registro las hara el Sistema.

- Brindar la informacion de manera rapida en situaciones criticas que requieren un rapido accionar por parte del Medico.
- Disponibilidad En todo momento y cualquier lugar para consulta por parte de los Medicos ya que solo requerira disponer de un usuario y un ordenador con coneccion a Internet para poder consultar.

Es cierto que los sistemas informaticos sufren problemas de Almacenamiento, Busqueda (en el caso de grandes volumenes de informacion) y deterioro por el paso del tiempo Pero en este caso el primero se soluciona Agregando mas espacio de disco, cosa que hoy en dia es algo relativamente barato a razon de 1 peso = 1 Gb.¹

El problema de las busqueda no afecta mucho con la velocidad de los equipos actuales se puede consultar bases de datos con millones de registro en unas pocas milesimas de segundos dicho tiempo resulta impersestible para la persona en la mayoria de las veses, en todo caso dependera de la implementacion y el motor de bases de datos mas que de las prestaciones del hardware.

En cuanto al deterioro, puede que con el tiempo los equipos de hardware tales como discos duros fallen en algun momento, pero esto es salvable siempre y cuando se realicen buenas practicas tales como implementar un sistemas de backup, tambien replicacion de datos en caso de que se necesite alta disponibilidad de la informacion que se almacena.

4.3. Gestión de Turnos

En lo que respecta a asignacion de turnos, el sistema actual en la mayoria de los casos no ha tenido un mejor panorama en cuanto a implatancion de un software, aunque ya en esta area existen algunas aplicaciones que intentan solucionar el problema de manera mas o menos eficientes.

¹Gb hace referencia a Gigabyte que es una medida utilizada en informatica la cual normalmente hace referencia a tamaños de almacenamiento.

Capítulo 5

El Proyecto

5.1. Motivación

En la actualidad existen pocos sistemas Aplicados en el ámbito de la gestión en el area de medicina y los existentes suelen ser solo para áreas específicas

5.2. Descripción del Proyecto

Lo que se pretendió con este proyecto era poder desarrollar un sistema que unifique las áreas de gestión y asignación de turnos y el manejo de historia clínica en un único sistema.

Cabe aclarar que el mismo se desarrollo con el propósito de poder ser utilizado principalmente en poli consultorios médicos ya que no plantea cuestiones tales como internaciones, traslado de pacientes, etc. como para poder ser de correcta utilidad en clínicas y hospitalares.

5.3. Arquitectura de la Aplicación

Implementado en Python utilizando en Framework Django, utilizando el motor de bases de datos PostgreSQL, funciona con una interfaz web por lo que se accede al mismo mediante un Navegador Web, Internamente maneja 2 módulos principales que son el “Modulo de Gestión de Turnos” el “Modulo de manejo de Historia Clínica”, al ser un sistema web implementa un tercer modulo de manera implícita que control de acceso mediante la definición de Grupos Usuarios y sus correspondientes permisos.

5.4. Modulo Usuarios

La gestión de usuarios es un proceso bastante común en casi todos los sistemas, muchos desarrolladores terminan programando funcionalidades de autenticación una y otra vez a lo largo de los años y casi siempre funcionando de la misma manera. Django se pensó para simplificar la vida no para complicarla, por eso al ser una tarea bastante común en casi todas las aplicaciones, viene incluido un completo sistema de autenticación que gestiona:

- Usuarios
- Grupos
- Permisos
- Sesiones de Usuarios y Cookies

Aunque en cuanto a lo que se refiere manejo de sesiones es un completo sistema solo maneja un pequeño conjunto de datos por lo que hubo que extender mediante la adición de un Modelo adicional para complementar la información de los usuarios.

5.4.1. Modelos

Aquí un diagrama con todos los modelos que componen el modulo Usuarios el cual está integrado en parte con el modulo Gestión de Turnos. Los Diagramas utilizados no corresponden a ninguna notación en particular solo sirven para mostrar la interacción en las diferentes clases, por lo que los modelos quizás se aproximen mas a una notación de Diagramas de clases en UML¹ que la típica definición del DER² aunque internamente se trabaje con un motor SQL (podrías usar un motor de bases de datos NoSQL) y no con clases el uso de un determinado motor de base de datos será transparente para el usuario no se necesitará modificar más que la configuración de la aplicación el resto seguirá como si nunca lo hubieses notado.³

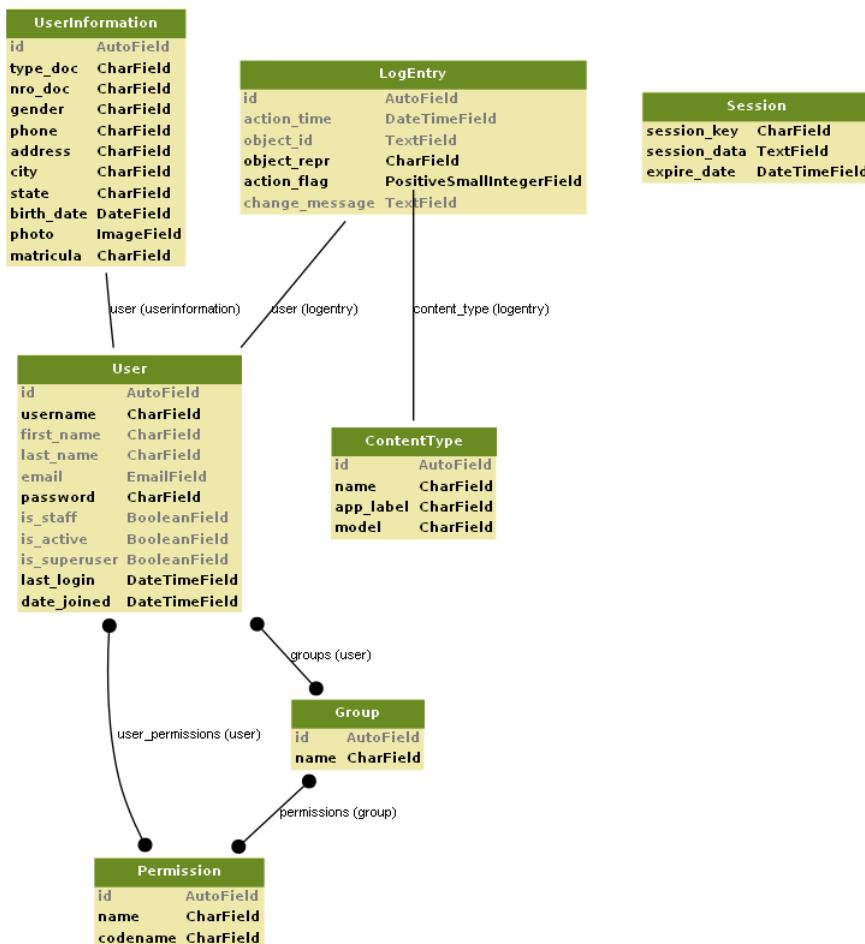


Figura 5.1: Diagrama con modelos que componen el modulo Usuarios

¹UML o Proceso Unificado de Desarrollo de Software es una metodología de desarrollo que proporciona diferentes herramientas una de ellas es el diagrama de clases.

²El DER o Diagrama Entidad Relacional el cual se usa para mostrar la interacción y cardinalidad entre las tablas de una base de datos.

³A lo sumo elegiremos entre uno y otro motor de bases de datos por una cuestión de performance más que nada.

El único modelo que fue necesario agregar es **UserInformation** que es para extender la información que se registra en el modelo **User** el resto vienen con Django. En Resumen aunque se podría haber desarrollado Un Modulo desde cero que gestione las sesiones de usuarios hubiese generado trabajo extra sin sentido.

5.4.2. Usuarios y Permisos

El sistema contempla 4 tipos de usuarios los cuales son:

- Usuarios no registrados
- Pacientes
- Médicos
- Administrativos

5.4.2.1. Usuarios no Registrados

Los **Usuarios no registrados** que vendría a ser cuando el usuario ingresa a la aplicación y no hay ninguna sesión iniciada, pueden acceder al sistema para consultar información básica y horarios de atención de los especialistas que forman parte de la institución, además de tener la opción de registrarse como paciente.

5.4.2.2. Paciente

El rol **Paciente** corresponde a los usuarios comunes, un usuario paciente puede ser creado por cualquiera de los otros roles, en caso que sea un usuario no registrado quien da el alta como paciente, el mismo deberá confirmar el registro mediante un código de verificación que el sistema le enviará al correo antes de poder comenzar a usar su cuenta, en los otros casos (el usuario Paciente es registrado por un medico o un Administrativo) no se requerida dicha confirmación.

En cuanto a los privilegios del usuario Paciente, este además de poder consultar la información de los especialistas puede solicitar un turno para ser atendido a un especialista en particular, también realizarle una interconsulta (mediante el sistema interno de mensajería) y modificar sus datos básicos, en resumen sus posibles funciones son:

5.4.2.3. Medico

Los Usuarios **Médicos** los cuales son asignados por los Administrativos a los Especialistas, en cuanto a privilegios y funcionalidades dentro del sistema, los mismos pueden:

- Registrar Pacientes
- Modificar datos de Pacientes
- Enviar Mensajes a cualquier Usuario
- Registrar Turnos
- Cancelar Turnos
- Administrar sus Horario de Atención
- Cancelar días de atención

Son los únicos usuarios que tienen acceso al Modulo *Historia Clínica*, en cuanto a privilegio sobre este modulo diremos que tiene la posibilidad de Crear, Modificar, Borrar (salvo casos específicos, que por su naturaleza no se permite dicha modificación.) un conjunto de Estudios, para más detalle se recomienda consultar el apartado sobre tal modulo.

5.4.2.4. Administrativo

En cuanto a los usuarios **Administrativo** poseen los mismos permisos que un usuario **Medico** exceptuando que no poseen acceso a las funcionalidades del Modulo Historia Clínica, como privilegio especial pueden administrar las cuentas de usuario de todos los roles incluidos en el sistema, incluido los Medico y otros Administrativos.

5.4.2.5. Admin

Existe un rol adicional que Django crea y gestiona por aparte, el mismo queda delegado para los administradores del sistemas ya que mediante el se puede acceder y modificar cualquier parte de la base de datos, por lo que podríamos decir que es un **Súper Usuario**, o usuario **Root** como para hacer analogía con los usuarios en entornos Unix, el mismo no forma parte del sistema desarrollado sino como funcionalidad adicional Django provee un panel de administración, para dicho tipo de usuario, al cual se puede acceder desde `/admin/` por ejemplo si estuviésemos ejecutando en un servidor local la ruta completa seria <http://127.0.0.1/admin/>⁴.

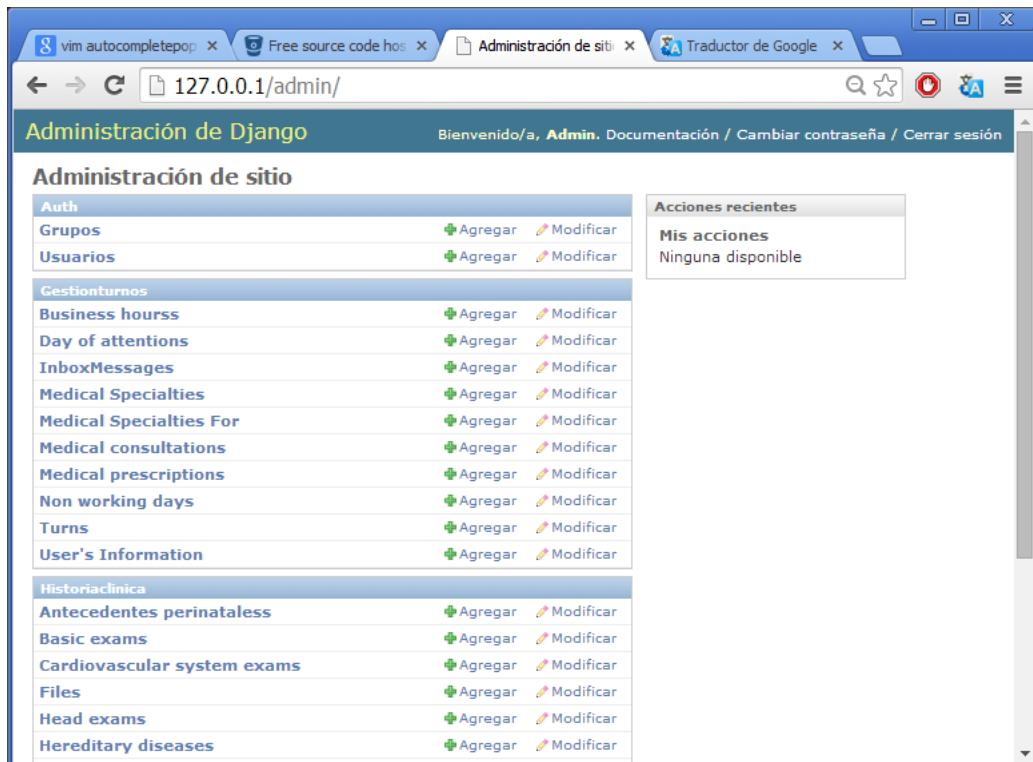


Figura 5.2: Vista del Panel Administracion provisto por Django

Cabe aclarar que el Súper Usuario del sistema de autenticación de Django dentro del sistema en sí mismo no posee ningún privilegio adicional, es mas para compatibilizar el usuario con la funcionalidad del sistema, cuando se inicializan por primera vez, se crea un usuario de este tipo llamado **admin**⁵ al cual se le asignan privilegio de *Administrativo*.

⁴Se puede consultar mas acerca de Django Admin en <https://docs.djangoproject.com/en/dev/ref/contrib/admin/>

⁵Se debe crear con ese nombre de usuario para evitar problemas en la inicialización, después se puede eliminar si se desea el usuario.

5.5. Modulo Gestión de Turnos

Dejando de lado el modulo Usuarios que nos provee Django el sistema desarrollado se divide esencialmente en 2 partes o módulos, aquí explicare como se diseño e implemento el Modulo Gestión de Turnos, que a mi consideración fue el que mayor reto aporto a la hora de pensar un solución para poder implementarlo.

El modulo se encarga de implementar las siguientes funciones

- Gestionar Datos de Usuarios
- Mensajería Interna
- Asignación de Especialidades Medicas
- Asignación de Turnos

5.5.1. Definición del Modelo

Aquí se muestra el diagrama de modelos que componen el modulo **Gestión de Turnos**⁶, por la cantidad de modelos se mostrara en 2 diagramas, igualmente téngase en cuenta que corresponden a un único modelo, lo que haremos será separar en los modelos específicos utilizados para gestión de turnos y el resto de los modelos definidos que complementan la funcionalidad del modulo.

5.5.2. Gestionar Datos de Usuarios

Esta funcionalidad describe, todo lo referido en cuanto al alta, baja y modificación de los datos de todos los usuarios. Implementa la vista tanto para modificación de datos personales, vistas de administrador para gestionar datos de otros usuarios.

5.5.3. Mensajería Interna

Permite la comunicación interna entre los usuarios, su principal utilidad es permitir que los Pacientes puedan realizar pequeñas interconsultas a los médicos atraves de la plataforma, sin requerir una consulta médica.

5.5.4. Asignación Especialidades Medicas

Dentro el modulo permite asignarles especialidades medicas correspondientes a los profesionales, aunque no realiza una distinción específica a la hora de asignar turnos, simplemente considera que el médico en dicho horario puede atender cualquier consulta relacionada a sus especializaciones.⁷

5.5.5. Asignación de Turnos

La *Asignación de Turnos* a los pacientes es la principal funcionalidad del modulo, que entre otras funcionalidades permite:

- Definir días de atención
- Definir días feriados o de Vacaciones
- Asignar turnos
- Controlar la asistencia de los pacientes.

⁶Vuelven a aparecer los modelos **User** y **UserInformation** por que casi todos los otros modelos dependen de alguna forma de ellos

⁷Téngase en cuenta que es raro ver un medico con varias especializaciones medicas y que el sistema fue pensado para ser utilizado en un consultorio médico, donde no se suele contar con equipamiento de alta complejidad.

Por cuestiones de tamaño del diagrama y por la cantidad modelos utilizados en el modulo se complicaba poder mostrarlos todo en una misma página por lo que para mejor visualización e interpretación separe el mismo en 2 partes:

El primer diagrama unifica todo lo referente a la asignación de turnos, que es lo principal del modulo:

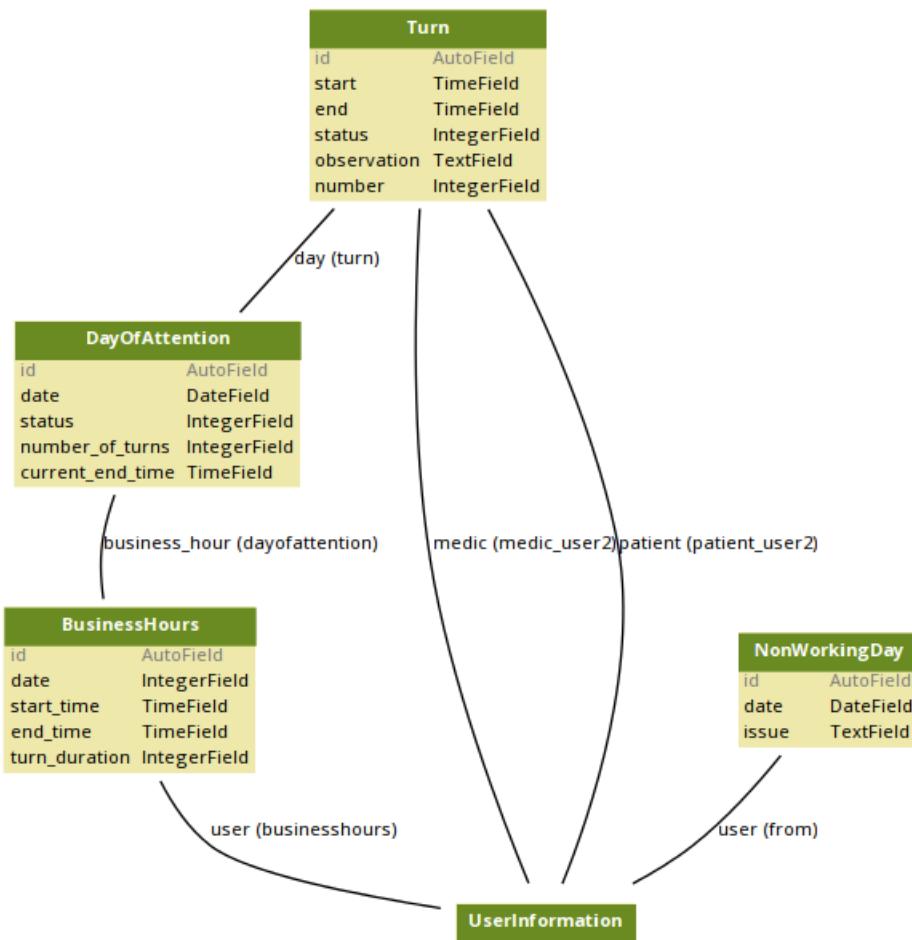


Figura 5.3: Módulo Gestión Turnos - Diagrama Modelos correspondiente a la Gestión de Turnos

En el segundo diagrama se muestra los modelos necesarios para las funcionalidades adicionales.

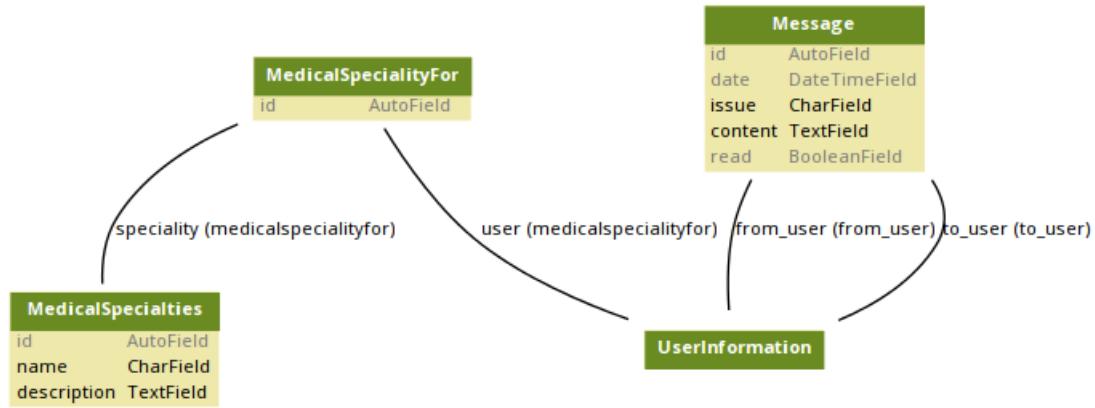


Figura 5.4: Modulo Gestión Turnos - Modelos Adicionales

5.5.6. Diseño de Modelos Para la Gestión de Turnos

Aquí se muestran las algunas consideraciones que se tuvieron en cuenta a la hora de diseñar los modelos para implementar la funcionalidad de asignación de turnos propiamente dicha la cual define el nombre del modulo.

5.5.6.1. BussinesHour(Horario De Atención)

Este modelo se utiliza para definir el horario de atención de cada médico, en el se especifican parámetros como:

- **user**: referencia al médico al cual pertenece
- **date**: define el día de atención ⁸
- **start_time, end_time**: marcan el horario de inicio y fin del día de atención del médico.
- **turn_duration** duración estimada del turno en minutos.

En base a esto se puede calcular un dato adicional que es la cantidad de turnos que se pueden asignar en tal día, y se hace de la siguiente manera:

```
numero\_turnos = (hora\_fin - hora\_inicio) // turn\_duration
```

Esto devolverá un valor entero por truncamiento, que será el numero máximo de turnos que se puedan asignar.

5.5.6.2. DayOffAttention (Días de Atención)

Este modelo maneja la disposición horaria de una fecha en particular, se basa en los datos que se definieron en el modelo anterior **BussinesHour**, por lo que solo se pueden generar en las fechas que correspondan con los días de la semana asignados, cuenta con los siguientes parámetros:

- **bussines_hour**: referencia al horario de atención definido por el médico.
- **date**: qué fecha cae ese día, esto se refiere al día del año específico.

⁸Este hace referencia a los días de la semana ósea Lunes, Martes,..., etc. por el momento solo se puede definir un único día de atención por día por medico.

- **status:** información del estado del día, es de tipo booleano y especifica el estado del día de atención siendo el valor TRUE para explicar que se pueden asignar más turnos y FALSE que no se pueden asignar más.⁹
- **number_of_turns:** numero de turnos que van siendo asignados
- **current_end_time:** horario actual de finalización

5.5.6.3. Turn (Turno)

Este modelo se define para registrar la información correspondiente a los turnos asignados es dependiente del modelo anterior **DayOffAttention** donde se especifican el resto de los datos como la fecha.

En cuanto a sus atributos, mucho no hay que explicar y son:

- **day:** hace referencia a un día de atención (DayOffAttention)
- **medic:** referencia a los datos del médicos
- **patient:** referencia al paciente.
- **start, end:** hacen referencia a las horas de inicio y fin correspondientemente.
- **status:** estado del turno es de tipo enumerado define varios posibles estados entre los que están (pendiente, concretado, cancelado medico, cancelado paciente).
- **observation:** campo de texto, para registrar cualquier observación pertinente.
- **number:** se refiere al número de orden para atención.

5.5.6.4. Consideraciones

En cuanto a funcionamiento de la asignación de turnos se tienen en cuenta las siguientes consideraciones:

Un turno no puede ser cancelado después de su hora de inicio por el paciente, siendo así posible de ser cancelado por el médico.

Al ser cancelado por un paciente se debe cambiar correspondiente estado dentro de día de atención.

Si un medico cancela un turno el mismo no se modificara el estado en la tabla día de atención, como el caso de los paciente que cancelen un turno, para el sistema hará de cuenta que los mismo ocurrieron, aunque si un turno es cancelado por el médico el mismo no será reprogramado.

En todos los casos los usuarios deberían poder recibir el correspondiente notificación de que se cancelo dicho turno invitándolos a reprogramar el mismo.

Los pacientes no pueden solicitar turnos durante el horario de atención del mismo, si el servidor comprueba que existen turnos sin actualizar estado como pendientes, si el médico o administrador no actualizo los mismos y ya paso la hora del mismo tiene que enviar notificaciones correspondientes.

⁹ Que un día de atención este marcado como no disponible (status=FALSE) puede significar que el cupo este lleno o que ese día el médico no pueda asistir o sea feriado por ejemplo, para determinar de qué se trata si dice no disponible y no hay ningún turno asignado (number_of_turns=0) significara que ese día el médico no atiende

5.6. Modulo Historia Clínica

Este modulo del sistema tiene como tarea manejar y recolectar toda información referente a las historia clínica de los pacientes.

5.6.1. ¿Que es una Historia Clínica?

Antes de entrar en todo lo referente sobre el desarrollo del correspondiente modulo tomo un momento para explicar concretamente a que no referimos cuando hablamos de la misma por lo que aquí tenemos la siguiente definición:

La historia clínica es un documento médico-legal que surge del contacto entre el profesional de la salud (médico, podólogo, psicólogo, asistente social, enfermero, kinesiólogo, odontólogo, etc.) y el paciente donde se recoge la información necesaria para la correcta atención de los pacientes. La historia clínica es un documento válido desde el punto de vista clínico y legal, que recoge información de tipo asistencial, preventivo y social.

La Historia Clínica se origina con el primer episodio de enfermedad o control de salud en el que se atiende al paciente, ya sea en el hospital o en el centro de atención primaria, o en un consultorio médico. La historia clínica está incluida dentro del campo de la semiología clínica ¹⁰.

5.6.2. La Historia Clínica en la Ley Argentina

La documentación médica comprendida en lo que comúnmente se denomina “historia clínica” la cual no se encontraba regida por leyes específicas en la Argentina hasta el 19 de noviembre del 2009 donde se promulga la Ley 26.529 [26].

En el capítulo primero de la ley se enumeran los derechos de los pacientes, en el artículo 2, inciso “a”. Renueva el derecho a la intimidad y la confidencialidad, donde se hace hincapié sobre la responsabilidad de preservar la intimidad y confidencialidad de toda la documentación médica concerniente a los pacientes, particularmente el inciso “d” del mismo artículo:

“El paciente tiene derecho a que toda persona que participe en la elaboración o manipulación de la documentación clínica, o bien tenga acceso al contenido de la misma, guarde la debida reserva, salvo expresa disposición en contrario emanada de autoridad judicial competente o autorización del propio paciente”.

Garantiza además el respeto por la autonomía del paciente y el derecho a recibir la información necesaria para su salud, incluyendo el derecho a negarse a ser informado.

El capítulo III reza sobre el Consentimiento Informado, el cual está basado en el principio de autonomía, es decir, el derecho del paciente a ser reconocido como persona libre y dueña de tomar sus decisiones. Para ello el paciente debe estar en condiciones de comunicar su decisión y éste ha sido informado adecuadamente de sus opciones, es decir, no pueden ser decisiones hechas como resultado de delirio o alucinaciones. La decisión del paciente es consistente con sus valores y metas y se mantiene estable en el tiempo si no han habido modificaciones hechas por el mismo sujeto. Los familiares de un paciente no están en el derecho de requerir al médico del paciente que no se le comunique ciertos detalles o información al mismo.

Ahora bien vallamos a lo que nos interesa:

¹⁰La Semiología Clínica es el cuerpo del conocimiento que se ocupa de la identificación de las diversas manifestaciones patológicas [?]

La ley define a la Historia Clínica como el documento “obligatorio, cronológico,¹¹ foliado y completo en el que consta toda actuación realizada al paciente por profesionales y auxiliares de la salud.” Define que la historia clínica es propiedad del paciente, siendo este el titular de la misma. Siempre que un paciente solicite la historia clínica, la institución competente debe entregarle una copia autenticada en 48 horas. Si no es entregada en ese plazo, el paciente está autorizado a interponer un recurso de Habeas Data, juzgado de por medio.

Entre los datos que han de consignarse en forma obligatoria esta la fecha de inicio y confección de la historia clínica, datos identificatorios del paciente y su núcleo familiar, datos del profesional interviniente y su especialidad, registros claros y precisos de los actos realizados por profesionales y auxiliares intervinientes, antecedentes genéticos, fisiológicos y patológicos si los hubiere, y todo acto médico realizado o indicado.

Incluye en la historia clínica a todos los documentos que hagan referencia a información de salud del paciente, añadiendo los consentimientos informados, hojas de indicaciones, hojas de enfermería, estudios complementarios, incluyendo las “prácticas realizadas, rechazadas o abandonadas.”¹² Esto último es interesante: si el paciente abandona o rechaza un tratamiento propuesto, es responsabilidad del médico consignarlo, que a fin de cuentas es el beneficiario de que aquello quede asentado desde el punto de vista médico-legal.

Autoriza a reclamar una copia de la historia clínica al paciente y su representante legal, al cónyuge o conviviente de hecho (sin importar el sexo), y a los herederos forzosos. Lo que no queda claro del art. 19 inciso b es si los cónyuges y convivientes requieren o no la autorización del paciente.

Se añade esta ley al capítulo 11 del Código de ética de la Asociación Médica Argentina, del año 2001. En ella se explaya en forma más extensa y detallada sobre la confección. Particular interés debiéramos prestarle al art. 168:

“La historia clínica ha de ser un instrumento objetivo y comprensible por terceros, y no solo por quienes escriben en ella..^a su vez, el art. 171 especifica que ”debe ser legible, no debe tener tachaduras, no se debe escribir sobre lo ya escrito, no debe ser borrada, no se debe dejar espacios en blanco y ante una equivocación debe escribirse ERROR y aclarar lo que sea necesario. No se debe añadir nada entre renglones. ^{12”}

5.6.3. Funcionalidades

Las funcionalidades que se implementan en correspondiente módulos son solo básicas y comprenden la documentación prácticas más comunes dentro del área de la medicina, esto no implica que solo valla a servir para eso únicamente, por su estructura el modulo contempla la posibilidad de agregar nuevos componentes para estudios específicos que sean requeridos y que no hayan sido contemplados en el actual sistema.

El modulo se encarga básicamente de registrar los diferentes estudios que se le practican a un paciente, adicionalmente registra información correspondiente a las interconsultas ¹³ y las observaciones del médico, así como los medicamentos que fueron recetados por el especialista.

¹¹En este sentido la ley está un poco atrasada ya que no contemplaba en ese momento la informatización de la historia clínica.

¹²Esto último es solo para la versión en papel

¹³Consultas medicas realizadas por el paciente

5.6.4. Definicion de Modelos

Los modelos que componen el Modulo corresponden a los diferentes tipos de exámenes de practica mas común¹⁴ en lo que Corresponden a la historia clínica.

Aquí también por la cantidad de modelos se hace difícil poder colocarlos todos en un único diagrama dentro de la pagina por lo que también se separara en varias partes sin romper las relaciones de los mismo, ósea aunque se realice una separación de los mismos corresponden a un único modulo por lo que tendrían que verse como un todo y no como partes separadas:

5.6.5. Modelos Básicos

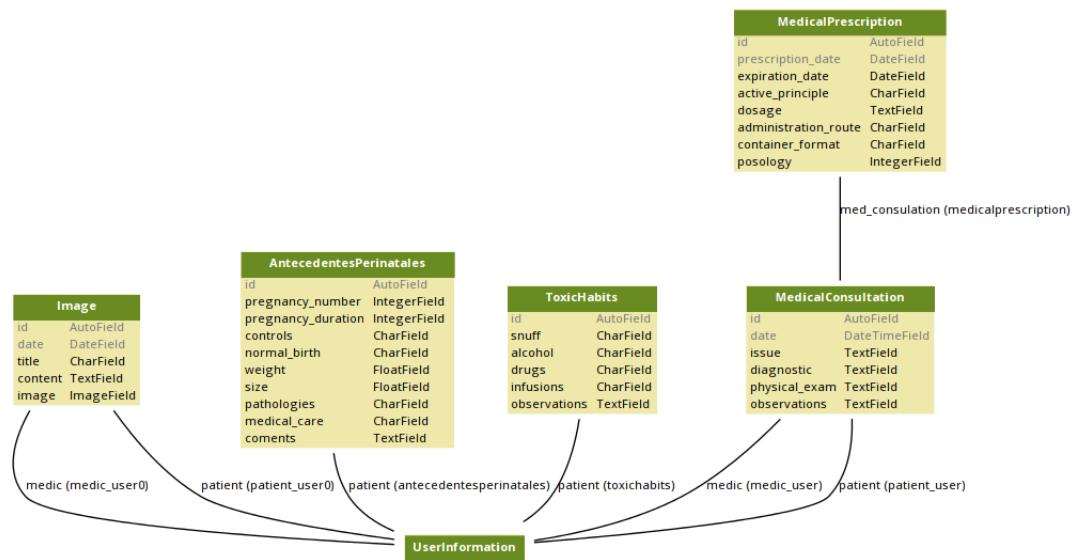


Figura 5.5: Historia Clínica Modelos Básicos

5.6.5.1. Image

Modelo encargado de almacenar imágenes de importancia sobre estudios realizados al paciente.

- **title** titulo de la imagen
- **content** Información adicional sobre la imagen.
- **date** Fecha que se tomo la imagen
- **image** referencia al archivo

5.6.5.2. MedicalConsultation (Consulta Médica)

Este Modelo almacena la información que el médico registra durante la consulta médica desde el motivo que origino la consulta, observaciones del médico, y demás datos de relevancia:

- **date** Fecha de la consulta

¹⁴Esto no es que este definido en algún lado que sean solo estos, sino que más bien son los exámenes que durante el análisis de diferentes modelos se presentaban mas comúnmente.

- **issue** Motivo de la consulta
- **diagnostico** Diagnóstico del médico.
- **physical_exam** Observaciones del Examen Físico ¹⁵
- **observations** Anotaciones Adicionales.

5.6.5.3. MedicalPrescription (Receta Médica)

Almacena información relevante sobre el medicamento prescripto en la receta médica, como se observa en el modelo las recetas médicas dependen directamente de una consulta médica, se asume que no se puede recetar un medicamento sin existir una consulta médica previa, aunque a partir de una consulta médica, se pueden generar varias recetas. Los datos que almacena el modelo se corresponden según las normas actuales sobre información que debe contener una receta médica y son:

- **prescription_date** Fecha que fue prescripto el medicamento
- **expiration_data** Vencimiento de la receta, pasada esta fecha la misma debe considerarse inválida.
- **active_principle** Principio Activo o nombre genérico de la monodroga que se receta.
- **dosage** Tamaño de la dosis
- **administration_route** Modo en que se debe tomar la droga
- **container_format** Formato del Envase y cantidad.
- **posology** Cada cuánto debe administrarse y porque periodo

5.6.5.4. Antecedentes Perinatales

Este tipo de modelo registra un único informe y se corresponde a los datos de nacimiento del paciente si es que existen, registra la siguiente información:

- **pregnancy_number** Número de embarazo de la madre.
- **pregnancy_duration** Duración del embarazo en semanas.
- **controls** Si asistió a controles médicos durante el embarazo.
- **normal_birth** Nació de parto normal o por cesárea.
- **weight** Peso al nacer.
- **size** Tamaño al nacer.
- **pathologies** Presentó complicaciones al nacer.
- **medical_care** Requirió atención médica especial.
- **observations** Otras anotaciones.

¹⁵Corresponde a un examen menor o visual simple que el médico realiza al paciente

5.6.5.5. ToxicHabit (Hábitos Tóxicos del Paciente)

Se refiere a si el paciente consume algún tipo de droga ósea refleja las adicciones del mismo sin importar si las drogas son legales o ilegales, para ayudar a determinar posibles causas y complicaciones en su evolución.

- **snuff** Tabaco
- **alcohol** Alcohol
- **drugs** Drogas
- **infusions** Infusiones
- **observations** Anotaciones Adicionales.

5.6.6. Modelos de los diferentes tipos de Examen

En este caso no se entrara en detalle sobre los datos e información que se registra estos modelos por cuestiones de que la mismas escapan del conocimiento común de las personas, si desea saber más acerca de ello en la bibliografía están las documentación en base que datos o modelos se modelaron los exámenes.

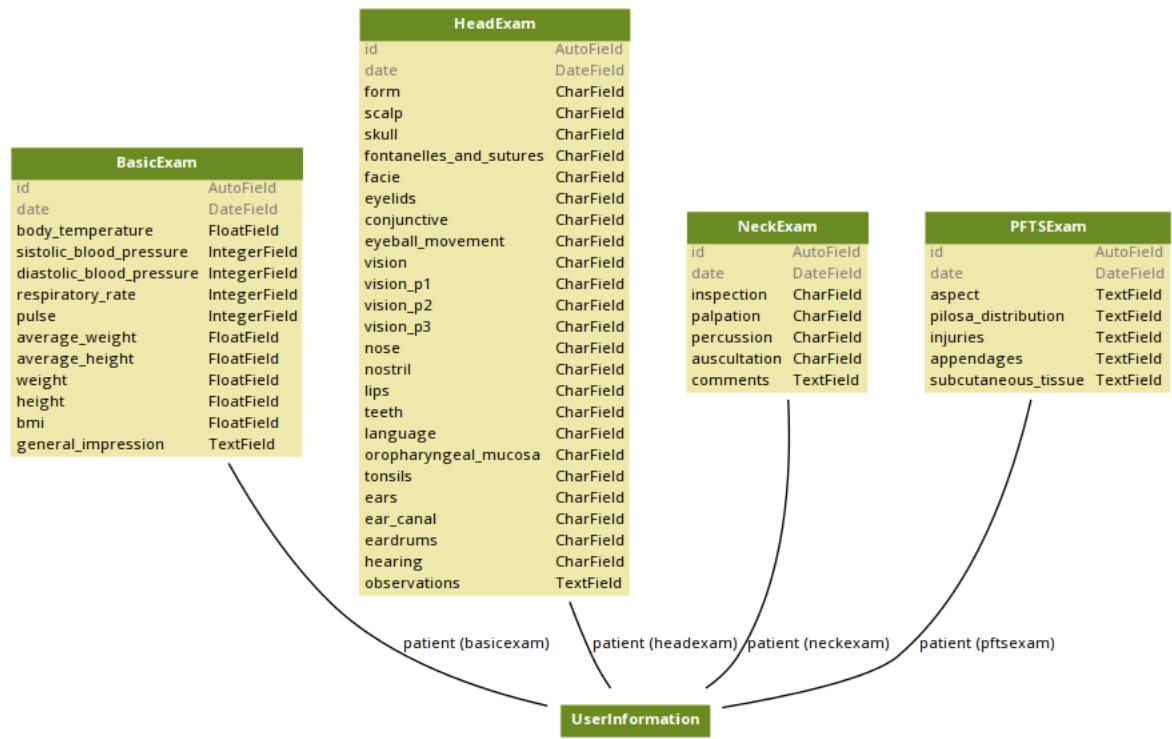


Figura 5.6: Historia Clínica Modelos de Estudios

5.6.6.1. BasicExam (Examen Básico)

El Modelo **BasicExam** refleja la información correspondiente a un examen físico general sin entrar en análisis de examen de especiales de las diferentes áreas del cuerpo.

5.6.6.2. HeadExam (Examen de Cabeza)

Un examen más especializado de la cabeza, en el que se analiza conjuntamente boca, ojos, nariz, forma del cráneo y oídos.

5.6.6.3. NeckExam (Examen de cuello)

Está orientada a buscar cambios en la forma del cuello (adenopatías, bocio, lipomas, quistes o tumores), en este caso es muy importante la ubicación del aumento de volumen cervical que oriente en el diagnóstico.

5.6.6.4. PFTSEExam (Examen de Piel Faneras y Tejido Subcutáneo)

Este modelo está orientado a registrar información correspondiente a estudio y descripción completa de: piel, faneras cutáneas, mucosas, tejido celular subcutáneo y músculos.

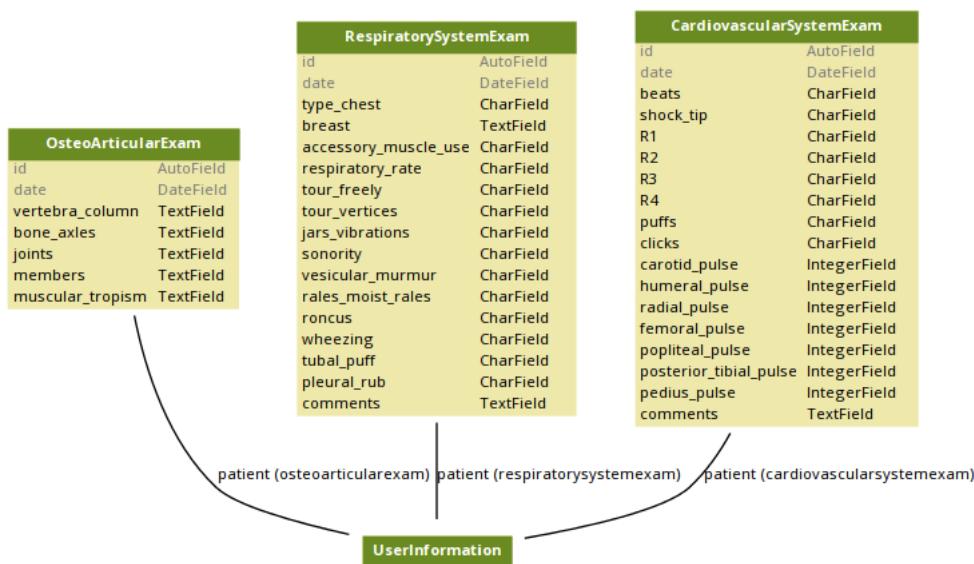


Figura 5.7: Historia Clínica Modelos de Estudios Continuación

5.6.6.5. OsteoArticularExam (Examen Osteo Articular)

El Examen Osteo Articular contempla la evaluación de la fuerza muscular esquelética medición de los miembros y movilidad Articular, en el se analizan la:

- Simetría Estructural y Alineación
- Facilidad y Amplitud de Movimientos
- Masa y tono Muscular
- Fuerza Muscular
- Apariencia de la piel sobre las articulaciones.
- Dolores, crepitaciones y deformidades

5.6.6.6. RespiratorySystemExam (Examen del Sistema Respiratorio)

Consiste en un examen en el cual se analizan cada una de los componentes del sistema respiratorio, se analizan:

- Tórax
- Sonoridad
- Capacidad Pulmonar
- Boca, fosas nasales, faringe, laringe y tráquea
- Vías Aéreas
- Frecuencia Respiratoria
- Músculos Accesorios y Vesiculares

5.6.6.7. CardiovascularSystemExam (Examen del Aparato Cardiovascular)

Este análisis trata de almacenar lo referente al estudio del sistema cardiovascular el cual está constituido por el corazón y los vasos sanguíneos (arterias, capilares y venas) el análisis permite recoger información sobre:

- Numero Latidos Por Minutos
- Choques de Punta
- R1, R2, R3 y R4
- Murmullo Vesicular y Chasquidos
- Pulso Carotideo
- Pulso Humeral
- Pulso Radial
- Pulso Femoral
- Pulso Poplítico
- Pulso Tibial Posterior
- Pulso Pedio
- Observaciones Generales

Capítulo 6

Instalacion y Configuracion

En este Capitulo Aparte de guiarle para realizar una exitosa implementacion Local del Servidor de Produccion se hara referencia a cada una de las Herramientas y librerias Utilizadas.

6.1. Requerimientos

6.1.1. Requerimientos de Hardware

Cualquier equipo que cumpla con las Caracteristicas para correr Windows 7 es suficiente en terminos de requerimientos minimos de Hardware siempre y cuando el numero de usuarios esperados no sea alto, despues el resto dependera de sus necesidades.

- Procesador x86, x64 de 1 Ghz o superior.
- Memoria Ram 1 GB o Superior

6.1.2. Requerimientos de Software

- Apache 2.2
- PosgreSQL 9.2
- Python 2.7.x o Python 2.6.x
- Django 1.3.x o Superior
- PGAdmin
- psycopg2
- mod_wsgi
- ReportLab
- easy-thumbnails
- django_extensions
- django_cron

6.2. Apache

Existen 2 caminos para instalar Apache La Primera Hacer una instalacion Limpia de Apache, la 2da es cuando no se quiere trastear con tanta configuracion por lo que opta por infraestructuras tipo WAMP, LAMP, WAPP, etc.

6.2.1. Instalacion en Limpio

Solo recomiendo este tipo de instalacion desde 0 para quienes ya poseen un conocimiento avanzado en cuanto a manejo de servidores.

Descargamos de Apache.org la ultima version disponible, puedes utilizar el siguiente vinculo: <http://www.apachehaus.com/cgi-bin/download.plx>.

Crea dos carpetas en la unidad C, la primera de nombre **Apache** y la segunda **servidor**. Descomprime el archivo descargado y ejecútalo, sigue los pasos de la instalación y de los datos que te piden solo escoge el destino de la instalación, que será la carpeta que creaste en **C:\Apache**, los otros datos déjalos de la forma predeterminada para configurarlos más tarde. El programa al instalarse crea un icono en el área de notificación que te permitirá: iniciar, detener y reiniciar Apache; tienes que tener en cuenta que cualquier cambio que hagas en el archivo de configuración no tendrá efecto hasta que reinicies el servidor.

6.2.2. Instalacion mediante WAMP, LAMP, MAMP, WAPP

Existen una infinidad de Paquetes precompilados y configurados, con Apache, PHP, PosgreSQL o MySQL y mas. Dichas infraestructuras suelen nombrarse como el acronomico de las herramientas que agrupan por ejemplo:

- WAMP Windows Apache MySQL PHP
- WAPP Windows Apache PosgreSQL PHP
- LAMP Linux Apache MySQL PHP
- MAMP Mac OS Apache MySQL PHP

Algunas distribuciones mas usadas disponibles Para Windows son WAMP Server <http://www.wampserver.com/> (WAMP), XAMPP <http://sourceforge.net/projects/xampp/> (WAMP + Perl), Bitnami <http://bitnami.com/stack/wapp> (WAPP) solo nos resta elegir cualquiera de ellas e instalarlas, aparte de la ruta de instalacion nos pediran el usuario y contraseña para acceder al motor de Base de Datos.

6.2.3. Configuracion

Toda la configuración para el funcionamiento de Apache se guarda en un archivo de texto nombrado: **httpd.conf** que se encuentra en la ruta **C:\Apache\conf** si realizamos una instalacion en limpio o **C:\wamp\bin\Apache\conf** si instalamos el paquete multiple preconfigurado no es necesario realizar este paso por lo que lo podremos saltar.

Al archivo **httpd.conf** lo podemos editar en cualquier editor de texto como Notepad. Buscamos la linea que dice

```
Listem LocalHost:80
```

y la Cambiamos por:

Listem 80

Ahora buscamos la instruccion:

DocumentRoot "C:\xxxxxxxx"

y la Cambiamos por:

DocumentRoot "C:\Servidor"

Recordar que al inicio de la instalacion creamos una Carpeta llamada Servidor en la unidad C. Por ultimo solo nos queda reiniciar el servidor Apache e introducir la siguiente direccion <http://127.0.0.1> si nos aparece una pagina **It's Work!** felicidades Apache esta Funcionando.

6.2.4. Instalacion de PosgreSQL

La versión de PostgreSQL que he utilizado durante el desarrollo del sistema es la 9.2.x, quisas cuando leas esto haya salido una nueva version la cual no deberia generar inconvenientes ademas de que es posible que el proceso de instalación pueda variar.

El primer paso es descargar el instalador de PostgreSQL para Windows, lo puedes descargar desde el enlace siguiente <http://www.postgresql.org/download/windows>, nos bajara un instalador similar a **postgresql-9.2.3-rc1-windows.exe** lo ejecutamos como administrador.

Si tenemos activado el control de cuentas de usuario nos mostrará una advertencia con el texto ”¿Desea permitir que este programa realice cambios en el equipo?”, pulsaremos ”Sí”para continuar con la instalación de PostgreSQL.

Indicaremos la carpeta de instalación de PostgreSQL, donde se guardarán los ejecutables, librerías y ficheros de configuración de PostgreSQL en mi caso el directorio es **C: \PosgreSQL \9.2** , Indicaremos también la carpeta donde se guardarán los datos por defecto de PostgreSQL **C: \pgsql-data** .

Solo nos queda introducir la contraseña para el superusuario ”postgres”que será con el que iniciemos sesión para administrar la base de datos, despues podremos crear otros usuarios si es necesario. Ademas introduciremos el puerto de escucha para la conexión con el servidor PostgreSQL, por defecto el 5432.

Seleccionaremos la configuración regional y comenzara la instalacion, con esto PosgreSQL quedara instalado. Si tenemos algún cortafuegos (firewall) deberemos abrir el puerto 5432.

6.2.5. Creacion de la Base de Datos

Junto con la Instalacion de PosgreSQL se instala el PGAdmin III que es una Heramienta GUI para administrar el motor de base de Datos. Iniciamos el Programa, desplegaremos ”Server Groups”, dentro desplegaremos ”Servidores” dentro de éste pulsaremos con el botón derecho del ratón sobre ”PostgreSQL 9.0 (localhost:5432)”, en el menú emergente seleccionaremos ”Conectar”.

Introduciremos la contraseña para el superusuario postgres (la contraseña introducida en la instalación).

Pulsaremos con el botón derecho del ratón sobre ”Bases de datos”, seleccionaremos ”Nueva Base de Datos”, en la pestaña ”Propiedades”introduciremos los siguientes datos:

- Nombre: nombre de la base de datos, en nuestro caso ”BDSem”.

- Propietario: seleccionaremos el usuario creado anteriormente "posgres".
- Codificado: seleccionaremos UTF8.
- Tablespace: seleccionaremos el tablespace creado anteriormente "pg_default".
- Colación: seleccionaremos "Spanish, Argentina".
- Tipo carácter: seleccionaremos "Spanish, Argentina".

Pulsaremos OK para crear la base de datos, con esto ya tendremos nuestra base de datos aunque vacia, el resto como creacion de las Tablas correspondientes nesesarias para el proyecto lo haremos mas adelante mediante Django.

6.3. Instalacion de Python

Para este proyecto se utilizo CPython pero no la version Oficial [urlhttp://www.python.org](http://www.python.org) sino la que distribuye Active State <http://www.activestate.com> llamada **Active Python** la cual provee caracteristicas adicionales a version oficial, podremos descargar la ultima version desde <http://www.activestate.com/activepython/downloads> aunque se recomienda instalar la version 2.7.x para evitar cualquier posible problema.

6.3.1. Probando Python

Para probar que la instalacion haya sido correcta abriremos la Terminal cmd.exe y escribiremos:

```
python
```

Si todo va bien nos debera aparecer algo similar a:

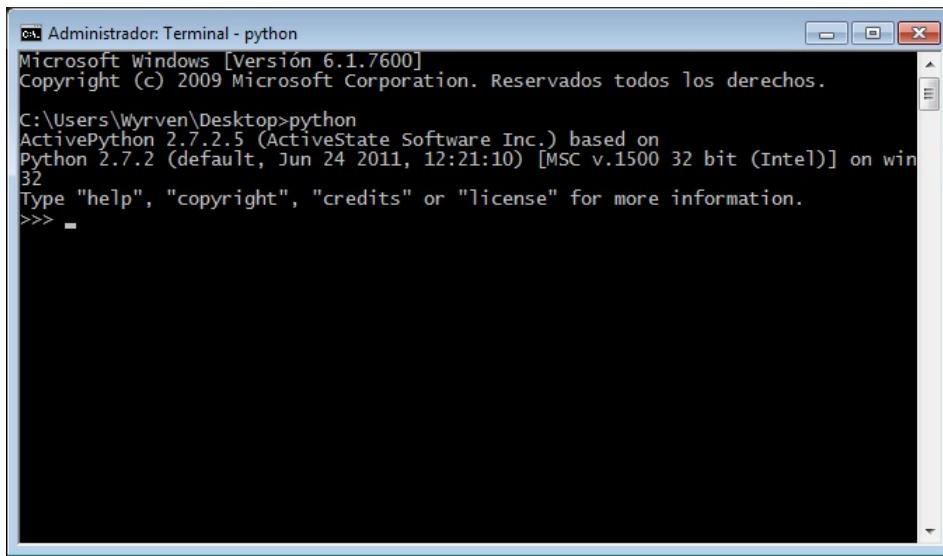


Figura 6.1: Ejecutando Python en la Terminal

En caso contrario deberias revisar que la ruta de Python este dentro de la variable PATH del sistema.

6.4. Instalar Django

Puedes bajarte Django desde el siguiente enlace <https://www.djangoproject.com/download/1.3.7/tarball/>¹ te descargara un paquete llamado Django-1.3.7.tar.gz lo descomprimes en algun directorio luego abres la Terminal y te posicionas sobre el directorio donde descomprimiste y ejecutas:

```
$ python setup.py install
```

Sino mediante el instalador de Paquetes de Python de manera mas automatica escribes en la terminal

```
pip install django==1.3.7
```

Con esto ya tendremos instalado Django.

6.5. Instalando el Resto de Las Dependencias

Ademas de Django en el Proyecto se utilizaron otras Librerias de Python las cuales algunas vienen instaladas y Otras Requieren ser instaladas de manera similar a como instalamos Django.

6.5.1. psycopg2

psycopg2 es un adaptador de base de datos PostgreSQL para el lenguaje de programación Python. psycopg2 fue escrito con el objetivo de ser muy pequeño y rápido y estable.

psycopg2 es diferente del otro adaptador de base de datos, ya que fue diseñado para aplicaciones en gran medida de subprocessos múltiples que crean y destruyen un montón de cursor y hacen que un número notable de inserciones o actualizaciones concurrentes. psycopg2 también proporcionan operaciones asincrónicas completos y apoyo a las bibliotecas de co-rutinas.

Para instalar descargue el precompilado desde <http://www.stickpeople.com/projects/python/win-psycopg/> Ejecutelo con permisos de administrador, nos pedira que seleccionemos la version de python con que se instalar.

6.5.2. ReportLab

ReportLab es la ultra-robusto motor de código abierto a prueba de tiempo para la creación de documentos PDF y gráficos vectoriales personalizado. Escrito en Python, ReportLab es rápido, flexible y una plataforma cruzada.

Proporciona un completo conjunto de herramientas de programación para la creación de documentos y gráficos complejos. Ofrecemos una serie de componentes de forma gratuita y de código abierto, además de un paquete comercial con características adicionales.

Para Instalar descargue el instalado desde <http://www.reportlab.com/software/installation/> y proceda de manera similar a como hizo con la instalacion de psycopg2.

6.5.3. easy-thumbnails

Easy_Thumbnails es Una potente aplicación thumbnailing ², pero fácil de implementacion para Django.

Para Instalar solo ejecute el siguiente comando en terminal, no Se necesita configurar nada en el proyecto el mismo esta previamente configurado.

```
pip install easy-thumbnails
```

¹la version 1.3.7 no es la ultima version disponible a la hora de crear este informe estaba por la 1.6.2 ya que Django se actualiza constantemente.

²Cuando hablamos de thumbnails nos referimos a las diferentes miniaturas que son versiones en distintos tamaños de una imagen y son usadas para ayudar a su organización y reconocimiento.

6.5.4. django_extensions

Django_Extensions es una colección de Extensiones (utilidades) Personalizadas de diferentes autores no relacionados con el Proyecto Django, para extender las capacidades del Framework.

Para Instalar solo ejecute el siguiente comando en terminal ³

```
pip install django-extensions
```

6.5.5. django_cron

Django-cron permite ejecutar código de Django de manera recurrente para el seguimiento y ejecución de las tareas. En este caso no es Necesario Instalar Nada, viene junto con el Código Fuente del Proyecto. Igualmente si tiene curiosidad puede visitar la pagina del proyecto <https://github.com/Tivix/django-cron>

6.5.6. Descargar e Instalacion de mod_wsgi

Asumiendo que ya tienes instalado Python y Apache, solo debes descargar el paquete libapache2-mod-wsgi ,la ultima version de mod_wsgi se puede descargar desde su pagina oficial <https://code.google.com/p/modwsgi/> descargaran un archivo similar a "mod_wsgi-win32-ap22py27-3.3.so" la version que descarguen de mod_wsgi depende como se ve, de la plataforma así como de la version de python que correrá en el servidor. luego por cuestiones de practicidad renombraremos el archivo de la siguiente manera:

```
mod_wsgi-win32-ap22py27-3.3.so -> mod_wsgi.so
```

Realizado dicho cambio copiamos el modulo dentro de la siguiente carpeta: APACHE_FOLDER \modules \APACHE_FOLDER vendría a ser el directorio donde tenemos la instalación de WAMP en mi caso es: C:\Apache.

6.5.7. Cargando el Modulo en Apache

Una vez que el módulo de Apache ha sido instalado en el directorio de módulos de su instalación de Apache, todavía es necesario configurar Apache para cargar el módulo en realidad.

Abrimos el archivo "httpd.conf" agregamos la siguiente linea en el mismo punto donde se cargan el resto de los modulos. ⁴

```
LoadModule wsgi_module modules/mod_wsgi.so
```

Con todo esto echo solo tenemos que reiniciar el servidor Apache, en nuestro caso clic en el icono en la barra de notificaciones luego las opciones Apache->Service->Reiniciar Servicio.

6.6. Configuración del Proyecto

Bueno Ahora solo tenemos que crear un alias en Apache ⁵ para nuestra carpeta donde colocaremos en mi caso la carpeta destino sera:

```
C:\Servidor\SGCM
```

SGCM es la carpeta contenedora del proyecto, y el alias que usaremos sera:

³Importante, no todas las funcionalidades están soportadas en Windows, pero en cuanto al proyecto no hay problemas.

⁴El archivo httpd.conf está en la siguiente ruta en el caso de mi instalación: C:\Apache\conf\httpd.conf

⁵Para mayor información de cómo crear alias en Apache consulte <http://httpd.apache.org/docs/2.2/urlmapping.html>

/sgcm/

tendremos que agregar las siguientes lineas al final del archivo httpd.conf de apache.

```
Alias /sgcm/ "C:/Servidor/SGCM/"
WSGIScriptAlias /sgcm "C:/Servidor/SGCM/handle.wsgi"

<Directory "C:/Servidor/SGCM">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride all
    Order allow,deny
    Allow from all
</Directory>
```

Hay un número de maneras en que una aplicación WSGI organizada por mod_wsgi⁶ puede montarse contra una URL específica. Estos métodos son similares a cómo se podría configurar las aplicaciones CGI tradicionales.

El principal enfoque implica declarar explícitamente en el archivo de configuración principal de Apache el punto de montaje URL y una referencia al archivo de comandos de aplicaciones WSGI. En este caso, el mapeo se fija, con cambios sólo ser capaz de ser hecho mediante la modificación de la configuración principal de Apache y reiniciar Apache.

Al utilizar mod_cgi para alojar aplicaciones CGI, esto se haría mediante la directiva ScriptAlias. Para mod_wsgi, la directiva en su lugar se llama WSGIScriptAlias.

WSGIScriptAlias /wsgi "C:/Servidor/SGCM/handle.wsgi"

Esta directiva sólo puede aparecer en los principales archivos de configuración de Apache. La directiva se puede utilizar en el ámbito del servidor, pero normalmente se coloca en el contenedor VirtualHost para un sitio en particular. No se puede utilizar en cualquiera de las directivas de contenedores ubicación, directorios o archivos, ni puede ser utilizada dentro de un archivo ".htaccess". El primer argumento de la directiva WSGIScriptAlias debe ser el punto de montaje URL para la aplicación WSGI. En este caso, la URL no debe contener una barra diagonal. La única excepción a esto es si la aplicación WSGI es para ser montado en la raíz del servidor web, en cuyo caso / sería utilizado.

El segundo argumento de la directiva WSGIScriptAlias debe ser una ruta absoluta para el archivo de comandos de aplicaciones WSGI. Es en este archivo que la muestra de código de la aplicación WSGI debe colocarse.

Tenga en cuenta que una ruta absoluta debe ser utilizado para el archivo de comandos de aplicaciones WSGI suministrado como segundo argumento. No es posible especificar una aplicación por sí sola Python nombre de módulo. Una ruta de acceso completa se utiliza para una serie de razones, la principal de las cuales por lo que todos los controles de acceso de Apache todavía pueden aplicarse para indicar que en realidad puede acceder a la aplicación WSGI.

Porque se aplicarán los controles de acceso de Apache, si la aplicación WSGI se encuentra fuera de los directorios que ya están configurados para ser accesible a Apache, habrá que decirle a Apache que los archivos dentro de ese directorio se pueden utilizar. Para ello se debe utilizar la directiva Directory.

Hasta aqui tenemos mod_wsgi y nuestro directorio listo, ahora probaremos que todo va bien para ello dentro del directorio crearemos un archivo llamado "handle.wsgi" que tendrá el siguiente contenido:

```
1 # -*- coding: utf-8 -*-
2
3 import os, sys
4 import django.core.handlers.wsgi
5
```

⁶Puede consultar <https://code.google.com/p/modwsgi/wiki/QuickInstallationGuide> si desea explorar otras opciones de configuración.

```
6 sys.path.append('C:/Servidor/SGCM')
7 sys.path.append('C:/Servidor')
8
9 os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'
10
11 application = django.core.handlers.wsgi.WSGIHandler()
```

Con esto nuestro servidor de aplicacion ya deberia funcionar aunque como veran no se cargan los archivos estaticos como imagenes y hojas de estilo por lo que necesitamos agregarlo.

Django no debería ser utilizado para servir archivos multimedia (imagen, audio, video, flash) por sí mismo; mejor deja ese trabajo al servidor web que hayas elegido. Recomendamos usar un servidor Web separado (es decir, uno que no está corriendo a la vez Django) para servir estos archivos.

Sin embargo, si no tienes opción para servir los archivos multimedia que no sea el mismo VirtualHost Apache que usa Django, aquí te mostramos como desactivar mod_python para una parte particular del sitio agregando el siguiente informacion a http.conf:

```
<Location "/media/">
    SetHandler None
</Location>
```

Cambia **Location** a la URL raiz donde se encuentran tus archivos.

Tambien puedes usar *LocationMatch* para comparar con una expresión regular. Por ejemplo, esto configura Django en la raiz del sitio pero deshabilitando Django para el subdirectorio media y cualquier URL que termine en .jpg, .gif, o .png:

```
<Location "/">
    SetHandler python-program
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE mysite.settings
</Location>

<Location "/media/">
    SetHandler None
</Location>

<LocationMatch "\.(jpg|gif|png)$">
    SetHandler None
</LocationMatch>
```

En todos estos casos, necesitarás configurar la directiva **DocumentRoot** para que Apache sepa dónde encontrar tus archivos estáticos.

Con esto Django estara funcionando correctamente, y podra cargar imagenes y demas ficheros necesarios.

6.7. Configuracion Inicial

Ahora que tenemos todo instalado y funcionando, debemos crear la configuracion inicial necesaria para que la aplicacion pueda funcionar ya que si intentamos acceder en este momento a la aplicacion nos devolvera una serie de errores, por faltar informacion, entre ellas que todavia no se crearon las tablas necesarias para soportar el modelo y los datos de los mismos.

Enpecemos creando las tablas necesarias, se supone que la base de datos ya esta creada y su nombre agregado en el archivo **settings.py** osea suponemos que hicimos todos los pasos necesarios para conectar la base de datos y demas bueno en la ruta donde localizamos el proyecto ejecutamos el siguiente comando desde terminal:

```
$ python manage.py syncdb
```

Esta instruccion le dice a Django que sincronize los modelos con la base de datos y que cree las tablas necesarias en la misma para que esto funcione, ademas no preguntara si deseamos crear una cuenta para el administrador debemos decir si, y proporcionar el nombre de la misma, el cual por requerimiento el nombre del usuario administrador debe ser **admin**, este requerimiento es necesario para luego lanzar e implementar la configuracion inicial.

Con las tablas creadas y el usuario admin creados, procedemos a crear la informacion necesaria para el funcionamiento de la aplicacion, para ello ejecutaremos el siguiente script:

```
$ python init_app.py
```

El mismo creara toda la informacion requerida por la aplicacion y con ello quedara nuestro servidor en funcionamiento, solo hara falta reiniciar Apache. Con esto concluye todo lo referente a configuracion del servidor.

Capítulo 7

Guia de Referencia

El presente capitulo no pretende ser un completo manual de usuario de la aplicacion, el sistema en si es bastante intuitivo en cuanto a su funcionamiento igual aqui se resumira un poco el funcionamiento y algunas de las diferentes sessiones de la Aplicacion.

7.1. Organizacion de la Aplicacion

La aplicacion se organiza de la siguiente manera, con las diferentes sessiones bien definidas:

- 1 Menu Principal
- 2 Menu Secundario
- 3 Cuerpo de la Aplicacion
- 4 Informacion de Usuario



Figura 7.1: Organizacion Espacial del contenido de la aplicacion

7.1.1. Menu Principal

El contenido del menu principal depende del tipo de usuario que haya iniciado session en base a ello tendra o no habilitadas diferentes funcionalidades de la aplicacion, los unico menus comunes son **Mensajes** y **Opciones**.

7.1.2. Menu Secundario

El menu secundario dependiendo de la vista donde se este, puede o no existir, y su contenido dependera del las acciones que pueden ser realizadas en ella.

7.1.3. Cuerpo de la Aplicacion

Aqui se localizara el contenido principal de la vista, ya sea un formulario para registrar alguna informacion, una lista para mostrar informacion etc.

7.1.4. Informacion de Usuario

Muestra informacion acerca de la session actual que se esta usando, tal informacion es el nombre del usuario ¹ y el tipo de usuario que puede ser (Paciente, Medico, Administrativo,Not Login en caso de no haber iniciado session)

7.2. Panel de Usuario No Registrado

Corresponde al panel que vera el usuario la primera ves que ingrese a la aplicacion las funciones que se pueden hacer son restringidas y se limitan a:

- **Inicio:** Ir a la pantalla de Inicio
- **Listado de Medicos:** Mostrar informacion basica acerca de los expecialistas con los que cuenta la institucion.
- **Registrarse:** Permite al usuario mediante una serie de pasos registrarse como paciente.
- **Iniciar Session:** Iniciar una session con un usuario registrado.

En cuanto la la vista **Inicio** es solo la pantalla principal de presentacion de la aplicacion con un logo de fondo.

La vista **Listado de Medicos** puede consultarla en la session del Panel del Paciente, ya que la unica diferencia considerable es que se agregan un par de opciones que permiten al Paciente realizar algunas acciones a diferencia del usuario no registrado que solo puede visualizar parte de la informacion.

7.2.1. Registrarse

Esta vista ofrece a los usuarios no registrados, un formulario donde deberan cargar una serie de datos para registrarse como pacientes.

¹nombre real, no el username

S.G.C.M.
Sistema de Gestión de Consultorio Medico

Home | Listado de Medicos | Iniciar Session | Registrarse

Nuevo Paciente

Nombre de Usuario (*) : ricardo
 Contrasenia (*) :
 Repita Contrasenia (*) :
 Email (*) : ricardo.quiroga.dev@gmail.co

Nombre : Ricardo D
 Apellido : Quiroga
 Fecha de Nacimiento : 20/6/1987 Calendar
 Tipo de Documento : Documento Nacional de Identidad
 Nro de Documento : 32572793
 Genero : Masculino

Direccion : 200 viivendas casa 25
 Ciudad : Salta Capital
 Provincia : Salta
 Telefono : 3874494307
 Foto : Seleccionar archivo Ningún archivo seleccionado

Registrar

Figura 7.2: Formulario Registro Paciente

Completado el registro y luego de enviado el formulario, si todos los datos son correctos nos mostrara un mensaje de que el registro fue exitoso:

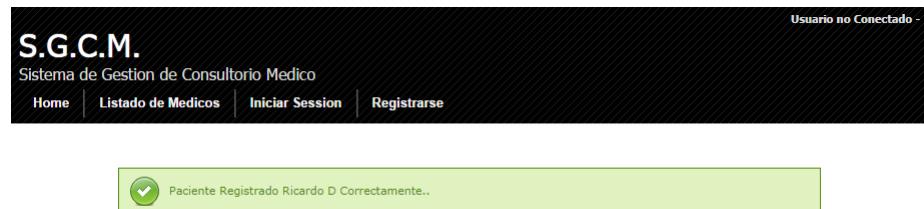


Figura 7.3: Formulario Registro Paciente

Paso siguiente deveremos revisar nuestra casilla de correo donde nos aparecera el mensaje con la direccion del formulario para activacion de usuario.²

²Si se intenta iniciar session sin haber activado el usuario nos devolvera un mensaje de error.

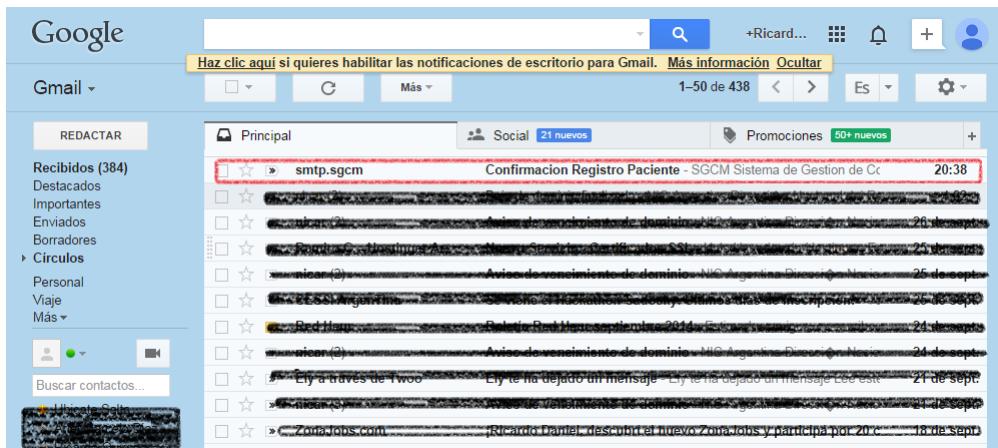


Figura 7.4: Bandeja de Correo con el mensaje

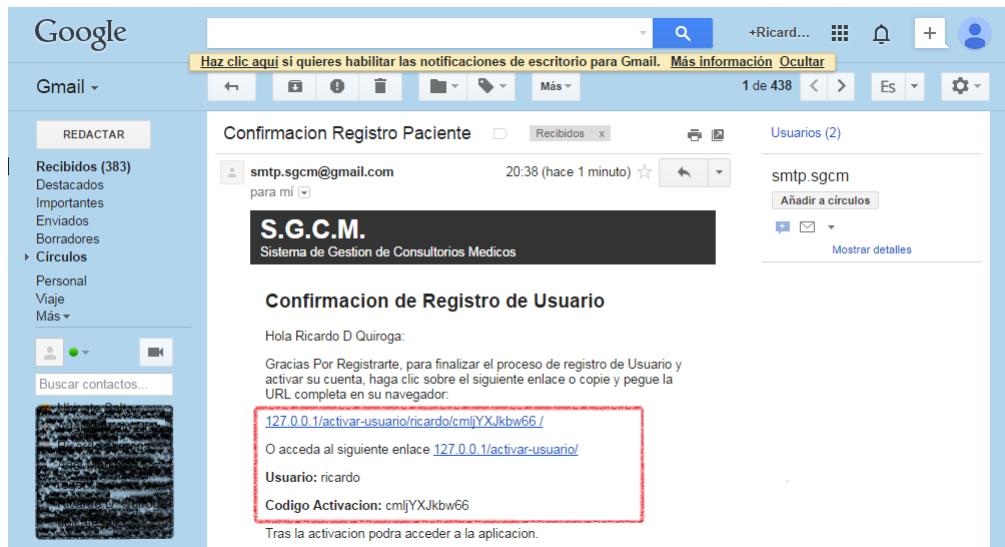


Figura 7.5: Cuerpo del mensaje con la informacion de activacion de usuario

The screenshot shows the 'Activar Usuario' (Activate User) page of the S.G.C.M. system. The page title is 'S.G.C.M.' and 'Sistema de Gestión de Consultorios Medicos'. Navigation links include 'Home', 'Listado de Medicos', 'Iniciar Sesión', and 'Registrarse'. On the right, it says 'Usuario no Conectado -'. The main form has two input fields: 'Usuario:' with value 'ricardo' and 'Codigo Activacion:' with value 'cmjYXJkbw66'. A blue 'Activar Usuario' button is at the bottom.

Figura 7.6: Formulario de Activacion de usuario

Luego de estos pasos el usuario estara registrado y activado, solo faltaria iniciar session para poder empezar a operar como paciente.

7.2.2. Iniciar Session

La vista de inicio de session no es nada de otro mundo, solo es un simple formulario donde debes introducir el usuario y contraseña validos para poder iniciar session.

7.3. Panel de Usuario Paciente

Corresponde al panel de funciones al que tendran acceso los usuarios, pacientes sigue siendo limitado pero ya se pueden hacer algunas cosas como solicitar turnos y realizar consultas medicas rapidas a un expecialista, se organiza en:

- **Inicio:** Ir a la pantalla de Inicio
- **Listado de Medicos:** Mostrar informacion acerca de los expecialistas.
- **Mensajes:** Casilla de Mensajes Internos.
- **Mis Turnos:** Informacion acerca del estado de los turnos del usuario.
- **Opciones:** Panel de Opciones

7.3.1. Listado de Medicos

Vista que permite seleccionar entre el listado de expecialistas que componen el cuerpo medico de la intitucion consultar informacion, realizar una consulta rapidas y solicitar turno.

The screenshot shows a dark-themed web interface for a medical consultation system. At the top, there's a header with the logo 'S.G.C.M.' and the subtitle 'Sistema de Gestión de Consultorio Medico'. On the right side of the header, it says 'Quiroga, Ricardo D - (Paciente)'. Below the header, there's a navigation menu with links: Home, Listado de Medicos, Mensajes, Mis Turnos, and Opciones. The main content area is titled 'Listado de Medicos' and displays a list of two entries. Each entry consists of a small placeholder profile picture and the doctor's name: 'Quiroga, Ricardo' and 'Roldan, Juan' respectively. The background of the page has a subtle diagonal striped pattern.

Figura 7.7: Listado de Medicos

S.G.C.M.
Sistema de Gestión de Consultorio Médico
Quiroga, Ricardo D - (Paciente)

Home | Listado de Médicos | Mensajes | Mis Turnos | Opciones

Información Básica
Solicitar Turno
Realizar Consulta Online

Consulta Online

Atras

Nuevo Mensaje

Medico: Juan, Roldan
Asunto:

Enviar

Figura 7.8: Formulario de Consulta Online

S.G.C.M.
Sistema de Gestión de Consultorio Médico
Quiroga, Ricardo D - (Paciente)

Home | Listado de Médicos | Mensajes | Mis Turnos | Opciones

Información Básica
Solicitar Turno
Realizar Consulta Online

Información del Médico

Atras

Datos Básicos

Nombre:	Juan Roldan
Teléfono:	No Definido
Número de Matrícula:	345678

Especialidades

[Cardiólogo](#)

Horarios Atención

Diá	Hora Inicio	Hora Fin	Duración (Minutos)

Figura 7.9: Mostrar Datos del Médico

La vista de Asignacion de turno son similares, entre si no varian mucho por lo que se explicara en la parte del panel de Medico.

7.3.2. Mensajes

Como ya se menciono esto es una session comun a todas los usuarios, se trata de un conjunto de vista donde funciona el sistema de mensajeria interna entre los usuarios de la aplicacion, su funcion esta reducida en cuanto al usuario paciente ya que este solo puede visualizar y responder los mensajes que se le envian, para enviar un mensaje a un profesional se realiza mediante la opcion

de Realizar Consulta Online, por lo que solo puede enviar mensaje a los expecialistas, las Opciones disponible son:

- **Redactar:** Escribir un Nuevo Mensaje
- **Recibidos:** Bandeja de Entrada
- **Enviado:** Bandeja de Salida

Algunas de las cuales pueden ser apreciadas en las siguientes capturas:

Este escreenshot muestra la interfaz de usuario para redactar un nuevo mensaje. En la parte superior, se muestra el logo 'S.G.C.M.' y el nombre del usuario 'Quiroga, Ricardo - (Medico)'. El menú superior incluye 'Home', 'Cronograma', 'Pacientes', 'Turnos', 'Mensajes' y 'Opciones'. La sección central titulada 'Nuevo Mensaje' tiene dos botones: 'Recibidos' (destacado) y 'Envíados'. Los campos para 'Destinatario' (conteniendo 'admin') y 'Asunto' (conteniendo 'Hola') están debajo. Un cuadro grande para el contenido del mensaje contiene la frase 'Mensaje de Prueba'. En la parte inferior derecha, hay un botón 'Enviar'.

Figura 7.10: Redactar un Mensaje

Este escreenshot muestra la lista de mensajes enviados. En la parte superior, se muestra el logo 'S.G.C.M.' y el nombre del usuario 'Quiroga, Ricardo - (Medico)'. El menú superior incluye 'Home', 'Cronograma', 'Pacientes', 'Turnos', 'Mensajes' y 'Opciones'. La sección central titulada 'Mensajes Enviados' tiene dos botones: 'Nuevo Mensaje' (destacado) y 'Recibidos'. Una tabla muestra los detalles de un mensaje: 'De:' (Ricardo, Quiroga) y 'Asunto' (Hola). A la derecha de la tabla hay un botón 'Mostrar | Eliminar'.

Figura 7.11: Bandeja de Entrada

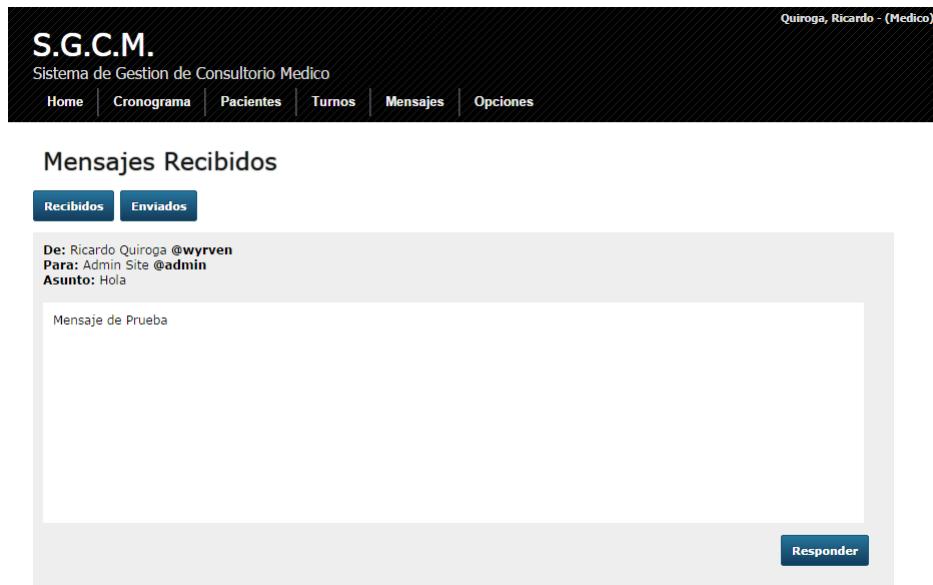


Figura 7.12: Mostrar Mensaje

7.3.3. Opciones

Otro menu comun a todo los usuarios permite la administracion de los datos e informacion del mismo, dentro de las funciones que permite este menu se encuentra:

- **Mis Datos:** Mostrar/Modificar Datos personales
- **Camivar Contraseña:** Formulario para cambio de Contraseña
- **Cerrar Session:** Cerrar Session, despedirse del sistema.

7.4. Panel de Usuario Administrativo

Corresponde al panel de funciones al que tendran acceso los usuarios Administrativos

- **Inicio:** Ir a la pantalla de Inicio
- **Pacientes:** Administrar Usuarios Pacientes
- **Medicos:** Administrar Usuarios Medicos
- **Administrativos:** Administrar Usuarios Administrativos
- **Especialidades:** Administrar Especialidades Medicas
- **Mensajes:** Casilla de Mensajes Internos.
- **Opciones:** Panel de Opciones

7.4.1. Pacientes, Medicos, Administrativos

Los tres conjuntos de vistas comparten muchas características similares por lo que se explican en conjunto y solo se mencionaran algunas de sus diferencias, cada sub menu se enlista de acuerdo a los tipos de usuarios que se desea administrar, permitiendo segun el sub menu la posibilidad de crear un tipo de usuario específico ³ Buscar un usuario, modifcar sus datos.

S.G.C.M.
Sistema de Gestión de Consultorio Medico

Site, Admin - (Administrativo)

Home | Pacientes | Medicos | Administrativos | Especialidades | Mensajes | Opciones

OPCIONES

Datos del Administrativo
Modificar Datos
Cambiar Contraseña

Datos del Administrativo

Nombre y Apellido:	Admin Site
Usuario:	admin
Email:	admin@admin.com
Documento:	D.N.I. - 123456
Fecha de Nacimiento:	20 Junio 1987
Sexo:	Masculino
Teléfono:	123456
Dirección:	No Definido
Ciudad:	Salta
Provincia:	Salta

Figura 7.13: Mostrar Administrativo

S.G.C.M.
Sistema de Gestión de Consultorio Medico

Site, Admin - (Administrativo)

Home | Pacientes | Medicos | Administrativos | Especialidades | Mensajes | Opciones

Buscar Paciente

Usuario	Apellido y Nombre
marc	Ortiz, Marcelo
karina34	Flores, Carina
andr3a	Gutierrez, Andrea Carolina
test12	Perez, Pepito
javier	Mansun, javier
javier12	Mansun, javier
drive	Diaz, Jose
minos	Diaz, Jose
testuser	Cruz, Juan Jose
testuserd	Cruz, Juan Jose
ricardo	Quiroga, Ricardo D

Figura 7.14: Vista para busqueda de usuario, en este caso usuarios Pacientes

En el caso de los usuarios medicos ademas puede consultar y modificar el estado de los turnos que les fueron solicitados, definirles expecialidades correspondiente.

³Los Usuarios registrados por el Administrador o el Medico no requieren activacion como los usuarios creados por usuarios no registrados.

The screenshot shows the S.G.C.M. (Sistema de Gestión de Consultorio Médico) interface. At the top, there is a navigation bar with links for Home, Pacientes, Medicos, Administrativos, Especialidades, Mensajes, and Opciones. On the far right of the header, it says "Site, Admin - (Administrativo)".

Datos del Medico

A detailed list of a doctor's information is displayed:

- Nombre y Apellido: Ricardo Quiroga
- Usuario: wyrven
- Email: l2radamanthys@gmail.com
- Documento: D.N.I. - 32793572
- Fecha de Nacimiento: 20 Junio 1987
- Sexo: Masculino
- Teléfono: 3875764272
- Dirección: Ricardo Raimundin 200 viviendas
- Ciudad: Salta
- Provincia: Salta
- Matrícula: 63781

Listado Expecialidades Medicas

Nombre	Opciones
Medico Clinico	Quitar

Figura 7.15: Mostrar Medico

The screenshot shows the S.G.C.M. interface again. The navigation bar and header are identical to Figure 7.15.

Turnos Solicitados

Medico: Ricardo Quiroga

Fecha	Paciente	Estado	Opciones
12/05/2014 - 08:00:00	Ortiz, Marcelo	Concretada	
12/05/2014 - 08:20:00	Flores, Carina	Vencido	
26/05/2014 - 08:00:00	Ortiz, Marcelo	Cancelado Medico	
26/05/2014 - 08:20:00	Flores, Carina	Vencido	
26/05/2014 - 08:40:00	Gutierrez, Andrea Carolina	Vencido	
26/05/2014 - 09:00:00	Perez, Pepito	Vencido	
14/07/2014 - 08:00:00	Ortiz, Marcelo	Vencido	
21/07/2014 - 08:00:00	Ortiz, Marcelo	Vencido	
21/07/2014 - 08:20:00	Gutierrez, Andrea Carolina	Vencido	
21/07/2014 - 08:40:00	Perez, Pepito	Vencido	
13/10/2014 - 08:00:00	Ortiz, Marcelo	Pendiente	
20/10/2014 - 08:00:00	Quiroga, Ricardo D	Pendiente	

Figura 7.16: Vista Listado Turnos Solicitados al Medico.

En los pacientes ademas puede asignar un turno a los mismo, cancelar un turno solicitado por el mismo, mostrar informacion e imprimir comprobante correspondiente.

The screenshot shows the S.G.C.M. application interface. At the top, there is a navigation bar with the title "S.G.C.M." and the subtitle "Sistema de Gestión de Consultorio Médico". Below the title, there are links for "Home", "Pacientes", "Medicos", "Administrativos", "Especialidades", "Mensajes", and "Opciones". On the right side of the top bar, it says "Site Admin - (Administrativo)".

On the left, a sidebar titled "OPCIONES" contains links: "Datos del Paciente", "Modificar Datos", "Nuevo Turno", and "Cambiar Contraseña".

The main content area is titled "Datos del Paciente" and displays the following patient information:

Nombre y Apellido:	Marcelo Ortiz
Usuario:	marc
Email:	marcelo@gmail.com
Documento:	D.N.I. - 31892473
Fecha de Nacimiento:	11 Abril 1984
Sexo:	Masculino
Teléfono:	123456
Dirección:	Batalla de Salta 202
Ciudad:	Salta Capital
Provincia:	Salta

Below this, there is a section titled "Turnos Solicitados" (Scheduled Appointments) which lists the following entries:

Fecha	Medico	Estado	Opciones
12/05/2014 - 08:00:00	Quiroga, Ricardo	Concretada	[Edit icon]
26/05/2014 - 08:00:00	Quiroga, Ricardo	Cancelado Medico	[Edit icon]
14/07/2014 - 08:00:00	Quiroga, Ricardo	Vencido	[Edit icon]
21/07/2014 - 08:00:00	Quiroga, Ricardo	Vencido	[Edit icon]

Figura 7.17: Mostrar Paciente

7.5. Panel de Usuario Medicos

Corresponde al panel de funciones al que tendran acceso los usuarios ...

- **Inicio:** Ir a la pantalla de Inicio
- **Cronograma:**
- **Pacientes:**
- **Turnos:**
- **Mensajes:** Casilla de Mensajes Internos.
- **Opciones:** Panel de Opciones

Capítulo 8

Conclusion y Mejoras

8.1. Resultado

El Sistemas de Gestión de Consultorios Medico proporciona soporte para Gestión de Turnos para pacientes y médicos proveyendo una nueva manera de mejorar la comunicación entre el paciente y el médico a través de Internet. Permite administrar las Historias Clínicas dejando de depender de archivos físicos y con la posibilidad de almacenar los mismos en la nube.

Considero que se alcanzaron casi todos los objetivos planteados y otros no planteados en la etapa inicial.

8.1.1. Ventajas Percibidas

Las Ventajas y Desventajas en lo que respecta al sistema fueron expuestas en el *Capítulo IV* cuando se realizó una comparación con el actual sistema, aquí se analizan las relacionadas a las herramientas que se utilizaron en su desarrollo.

- La primera ventaja que encontré fue la velocidad de desarrollo comparando con otras herramientas aunque Python no es un 4GL sino un 3GL la facilidad de entendimiento de su sintaxis hace que el código sea fácilmente entendible y legible lo que permite un mantenimiento sencillo, el código en Python se asemeja mucho a lo que hacemos cuando escribimos un algoritmo en el papel por lo que la curva de aprendizaje si ya manejas algún lenguaje es mínima, aprendí Python en 5 días.
- Django y el Modelo de desarrollo MVC (Modelo Vista Controlador) aportaron otro extra a la velocidad de desarrollo del sistema ya que solo con un par de líneas era capaz de crear vistas fácilmente adaptables, además de la característica de poder heredar plantillas por lo que en caso que quisiera realizar un cambio en el diseño de la plantilla solo requería cambiar la plantilla maestra o base sin necesidad de estar modificando una por una todas las plantillas, ni hablar si el código hubiese estado mezclado con el HTML como ocurre a veces con PHP por ejemplo.
- Otra característica interesante de Django que me ahorró sufrimiento fue la definición de Modelos, cuando trabajas con Django no hace falta conocer el motor de Base de Datos y su sintaxis, no te debes preocupar por aprender cómo realizar tal o cual consulta, dejas de pelear con los JOIN de SQL y demás, solo te dedicas a aprender a manejar el ObjectRelationalModel o ORM que forma parte de Django el cual es sencillo de aprender.
- Aunque no está relacionado en sí con el desarrollo de manera explícita agradezco haber conocido sitios como [urlhttp://www.stackoverflow.com](http://www.stackoverflow.com) que es un sitio colaborativo donde puedes

hacer preguntas y/o responderlas sobre cuestiones de programacion, instalaciones, errores, etc. Fue una gran ayuda ya que pude solucionar gracias a eso muchas de las dificultades y entender el problema de las mismas de manera rapida.

- Aprender a usar un sistema de control de versiones para el codigo fuente como GIT de mi proyecto fue de gran utilidad ya que el desarrollo de esta aplicacion no fue de manera continua sino que variada durante todo el tiempo de desarrollo.

8.1.2. Desventajas Percibidas

No todo el desarrollo fue como se esperaba, surgieron una serie de inconvenientes o limitaciones relacionadas con la herramienta.

- Hacer Deploy ¹ con la herramienta no es tan facil como cuando instalas LAMP, pierdes mucho tiempo intentando configurar el servidor, la documentacion existente sobre la misma es muy poca y normalmente incompleta.
- En su mayoria la documentacion sobre las librerias y demas herramientas se encuentra escrita en ingles, no lo consideraria en si una desventaja pero lo menciono en este apartado por mi bajo nivel en lo que respecta a lectura y comprension de texto en ingles.

8.2. Futuras Mejoras

El sistema podria evolucionar de varias maneras, al ser ul sitema diseñado mediante plantillas la principal evolucion del mismo es que se podria adaptar las interfaces a los navegadores de los dispositivos moviles inteligentes.

Otra mejora comun al sistema, seria que pueda integrarse con otros estudios como poder registrar analisis de laboratorio, odontogramas, integracion con el sistema vademecun para que sea mas sencillo elaborar una receta medica, y la posibilidad de importar y/o exportar la historia clinica a formatos conocidos como archivos PDF para permitir ser exportado a papel.

¹Implementar un Servidor de Produccion con Apache, Python, Django y PosgreSQL mod_wsgi

Bibliografía

- [1] WIKIPEDIA *Servidor HTTP Apache*
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
- [2] *Servidor Apache en Ubuntu*
<http://kuyne.blogspot.com.ar/2013/03/servidor-apache-en-ubuntu-instalacion-y.html>
- [3] *Servidor Apache en Windows*
<http://norfipc.com/internet/instalar-servidor-apache.html>
- [4] WIKIPEDIA *El Modelo Vista Controlador*
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- [5] ROGELIO LEÓN LÓPEZ, BÁRBARA GALLEGOS MACHADO Y JOSÉ DÍAZ NOVÁS *Formato recomendable para llenar la hoja de remisión médica de un paciente*
http://bvs.sld.cu/revistas/mgi/vol22_2_06/mgi10206.htm
- [6] SISTEMA CONSULTORIO WEB *Registro Consulta*
<https://www.consultorioweb.com/intranet/doctor/pacienteConsultas.aspx>
- [7] PRACTICA FINAL OBLIGATORIA: INTERNADO ROTATORIO Y PASANTIA RURAL OBLIGATORIA *Modelo Historia Clinica*
<http://www.med.unne.edu.ar/internado/his.cli.pdf>
- [8] ANY FLOWERS *Modelo Historia Clinica*
<http://www.slideshare.net/AnyFlowers/ejemplo-historia-clinica>
- [9] BIOCOP *Formato de Historia Clinica*
http://www.biocom.com/informatica_medica/historia_5_examen_fisico.html
- [10] INFOMED RED DE SALUD DE CUBA *Examen Fisico Regional*
<http://www.sld.cu/galerias/pdf/sitios/pdguanabo/cap04.pdf>
- [11] ESMAS *Problemas Auditivos Comunes*
<http://www.esmas.com/salud/enfermedades/notransmisibles/368755.html>
- [12] WIKIPEDIA *Perdida de Audicion*
http://es.wikipedia.org/wiki/P%C3%A9rdida_de_audiencia
- [13] HERNANDO VARGAS VÁSQUEZ *Rinología*
http://sisbib.unmsm.edu.pe/BibVirtualdata/libros/Medicina/cirugia/Tomo_V/archivos%20PDF/7Rinolog
- [14] WIKIPEDIA *Examen Labios*
<http://es.wikipedia.org/wiki/Labio>
- [15] CONSUMOTECA *Qué partes deben tener y datos incluir por ley las recetas médicas*
<http://www.consumoteca.com/bienestar-y-salud/medicamentos/que-partes-deben-tener-y-datos-incluir-por-ley-las-recetas-medicas/>

- [16] WIKIPEDIA *Vias de Administracion de Farmacos*
http://es.wikipedia.org/wiki/V%C3%ADas_de_administraci%C3%B3n_de_f%C3%A1rmacos
- [17] PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE - ESCUELA DE MEDICINA *Respiracion*
<http://escuela.med.puc.cl/Publ/ManualSemiologia/190Respiracion.htm>
- [18] BIOPCOM *Historia Clinica - Examen Fisico*
http://www.biocom.com/informatica_medica/historia_5_examen_fisico.html
- [19] *Examen Fisico del Sistema Ostiomioarticular*
<http://www.slideshare.net/wendy1971/examen-fisico-del-sistema-ostiomioarticular>
- [20] MODWSGI *Guia de Configuracion*
<https://code.google.com/p/modwsgi/wiki/QuickConfigurationGuide>
- [21] WSGI *Guia de Referencia WSGI - EN*
<http://wsgi.readthedocs.org/en/latest/>
- [22] APACHE *URL Mapping*
<http://httpd.apache.org/docs/2.2/urlmapping.html>
- [23] MINISTERIO DE SALUD *Formato de Historia Clinica*
http://msal.gov.ar/ENT/SRV/Materiales_Paciente/Herramientas_Utiles/Historia_Clinica/Historia_Clinica.aspx
- [24] HISTORIA CLINICA QUE ES
http://es.wikipedia.org/wiki/Historia_cl%C3%ADnica
- [25] SEMÍOLOGIA CLINICA
<http://es.wikipedia.org/wiki/Semilog%C3%ADa>
- [26] LEY 26.529 *Normativa sobre el Manejo de Historia Clinica*
http://www.msaludjujuy.gov.ar/Re2014/Archi_Varios%5Cley_26529.pdf