

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
Revision 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
2.2.1. ¿Por qué use Apache?	4
2.3. mod_wsgi	5
2.3.1. WSGI	5
2.4. PosgreSQL	5
2.4.1. ¿Por qué use PostgreSQL?	5
2.5. Python	6
2.5.1. La Filosofía detrás de Python	6
2.5.2. Baterías Incluidas	7
2.5.3. Implementaciones	7
2.5.4. ¿Por qué elegí Python?	8
2.6. Django	8
2.6.1. MVC	8
2.6.2. Django y el MVT	9
2.6.3. El Modelo	10
2.6.4. La Vista	11
2.6.5. La Plantilla	12
2.6.6. La Configuración de Rutas	13
2.6.7. Porque Django y no otro Framework	13
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	23
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	28
5.4.2.2. Paciente	28
5.4.2.3. Medico	28
5.4.2.4. Administrativo	28
5.4.2.5. Admin	29
5.5. Modulo Gestión de Turnos	29
5.5.1. Definición del Modelo	30
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	32
5.5.6.1. BussinesHour(Horario De Atención)	32
5.5.6.2. DayOffAttention (Días de Atención)	32
5.5.6.3. Turn (Turno)	32
5.5.6.4. Consideraciones	33
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	34
5.6.3. Funcionalidades	35
5.6.4. Definición de Modelos	35
5.6.5. Modelos Básicos	36
5.6.5.1. Image	36
5.6.5.2. MedicalConsultation (Consulta Médica)	36
5.6.5.3. MedicalPrescription (Receta Médica)	37
5.6.5.4. Antecedentes Perinatales	37
5.6.5.5. ToxicHabit (Hábitos Tóxicos del Paciente)	37
5.6.6. Modelos de los diferentes tipos de Examen	38
5.6.6.1. BasicExam (Examen Básico)	38
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)	39
5.6.6.6. RespiratorySystemExam (Examen del Sistema Respiratorio)	39
5.6.6.7. CardiovascularSystemExam (Examen del Aparato Cardiovascular) .	40

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. Instalación en Limpio	42
6.2.2. Instalación mediante WAMP, LAMP, MAMP, WAPP	42
6.2.3. Configuración	43
6.2.4. Instalación de PosgreSQL	43
6.2.5. Creación de la Base de Datos	44
6.3. Instalación de Python	44
6.3.1. Probando Python	45
6.4. Instalar Django	45
6.5. Instalando el Resto de Las Dependencias	45
6.5.1. psycopg2	46
6.5.2. ReportLab	46
6.5.3. easy-thumbnails	46
6.5.4. django_extensions	46
6.5.5. django_cron	46
6.5.6. Descargar e Instalación de mod_wsgi	47
6.5.7. Cargando el Modulo en Apache	47
6.6. Configuracion del Proyecto	47
6.7. Configuración Inicial	49
7. Guia de Referencia	51
7.1. Organización de la Aplicación	51
7.1.1. Menú Principal	52
7.1.2. Menú Secundario	52
7.1.3. Cuerpo de la Aplicación	52
7.1.4. Información de Usuario	52
7.2. Panel de Usuario No Registrado	52
7.2.1. Registrarse	52
7.2.2. Iniciar Sesión	55
7.3. Panel de Usuario Paciente	55
7.3.1. Listado de Médicos	55
7.3.2. Mensajes	56
7.3.3. Opciones	58
7.4. Panel de Usuario Administrativo	58
7.4.1. Pacientes, Médicos, Administrativos	59
7.5. Panel de Usuario Médicos	61
7.5.1. Cronograma	61
7.5.2. Pacientes	62
7.5.3. Turnos	66
7.5.3.1. Registrar nuevo Turno	66
8. Conclusion y Mejoras	69
8.1. Resultado	69
8.1.1. Ventajas Percibidas	69
8.1.2. Desventajas Percibidas	70
8.2. Futuras Mejoras	70
Glosario	71

Índice de figuras

2.1. Diagrama del Patrón MVC Modelo Vista Controlador	9
2.2. Generaciones de Herramientas de Desarrollo Web	10
2.3. El patrón Modelo Vista Template de Django	10
4.1. Almacenamiento Físico de Archivos	19
4.2. Modelo Historia Clínica Ministerio de Salud de La Nación Pag 1	20
4.3. Modelo Historia Clínica Ministerio de Salud de La Nación Pag 2	21
5.1. Diagrama con modelos que componen el modulo Usuarios	27
5.2. Vista del Panel Administración provisto por Django	29
5.3. Modulo Gestión Turnos - Diagrama Modelos correspondiente a la Gestión de Turnos	30
5.4. Modulo Gestión Turnos - Modelos Adicionales	31
5.5. Historia Clínica Modelos Básicos	36
5.6. Historia Clínica Modelos de Estudios	38
5.7. Historia Clínica Modelos de Estudios Continuación	39
6.1. Ejecutando Python en la Terminal	45
7.1. Organización Espacial del contenido de la aplicación	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 información de activación de usuario	54
7.6. Formulario de Activación de usuario	54
7.7. Listado de Médicos	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 búsqueda 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
7.18. Vista del Cronograma	62
7.19. Mostrar Paciente	62
7.20. Consulta Médica	63
7.21. Imprimir Receta o Prescripción Medica	63
7.22. Examen Único	64
7.23. Listado de Exámenes Aparato Cardiovascular	64
7.24. Vista Muestra Datos Examen Cardiovascular	65

7.25. Nuevo Examen Físico	65
7.26. Vista para Selección de día de calendario para Asignar turno	66
7.27. Vista de Confirmación de los datos del Turno	67
7.28. Vista muestra la informacion del Turno	67
7.29. Ejemplo de Comprobante de turno	68

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 la 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 aplicación web, es aquella que solo requiere ser instalada en un servidor, su ejecución 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 información que se maneja es multiusuario por lo que son especialmente útiles para desarrollar aplicaciones multiusuario basadas en compartir información.
- Consumen muy pocos recursos, por lo que el usuario no necesita tener un ordenador con grandes prestaciones para trabajar con ellas.
- Son fáciles de actualizar y mantener.
- Se pueden utilizar en miles de equipos sin limitación y restricción alguna.
- Su funcionalidad es independiente del sistema operativo Instalado en el ordenador del usuario.
- No hay problemas de incompatibilidad de versión de software ya que los usuarios trabajan con la misma versión.

En resumen el sistema por sus características podría haberse implementado como un sistema desktop pero se hubiesen perdido las características 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.

2.2.1. ¿Por qué usar Apache?

Se eligió apache como servidor HTTP antes otras opciones como Nginx o ISS por cuestiones de costos beneficios (Nginx funciona mejor para servir archivos estáticos, en cambio si se necesita correr aplicaciones en Servidor Apache es mejor opción), posibilidad de portabilidad, viene con una función de soporte para PHP, Python, Perl y otros lenguajes. Por ejemplo, el mod_python y mod_php Apache Módulos PHP proceso y código Perl dentro del proceso de Apache. mod_python es más eficiente que el uso de CGI o FastCGI, ya que no tiene que cargar el intérprete de Python para cada solicitud, por ser el servidor más usado está altamente documentado y es fácil conseguir soporte.

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.

Existen varias interfaces WSGI para apache, se eligió mod_wsgi sobre el resto (como por ejemplo mod_python) por recomendaciones de los desarrolladores de Django.

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 estándar 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.

2.4.1. ¿Por qué usar PosgreSQL?

Porque elegí PosgreSQL como motor de bases de datos, primero que nada costos, PosgreSQL es software libre y se distribuye libremente por lo que el costo de adquirir el motor de bases de datos fue 0 \$, otra característica es por su performance ante otros Motores como MySQL que tras varias transacciones comerciales/empresariales la han llevado de un lado a otro, finalmente ahora es propiedad de Oracle (Una conocida Empresa que se dedica a comercializar motores de

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

bases de datos privativos) estos hechos han llevado a MySQL a vivir un deterioro global puesto que este sistema gestor de base de datos actualmente ofrece un rendimiento excesivamente bajo en comparación incluso con motores pequeños como SQLite.

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.

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 similar 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 practico 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 mas 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

2.5.4. ¿Por que elegí Python?

Antes de conocer Python programaba en C++, Java y PHP, ahora ¿porque no desarrolle usando algunos de esos lenguajes? Como dice el Actor Eugenio Derbez en parodia el Longe Moco, ¡fue horrible, fue horrible..! Son lenguajes muy conocidos pero es demasiado engorroso lograr desarrollar aplicaciones que sean escalables y de rápido desarrollo y sobre todo mantenibles con el tiempo, tienen una pésima curva de aprendizaje (excepto PHP pero por contraparte su notación es horrible y dificulta la lectura y peor que se mescla en muchos casos con el HTML), por eso usar python fue un gran salto y me pude dedicar a programar no a resolver problemas de almacenamientos de datos por ejemplo.

Podria haber usado un 4GL, si, pero como que estando todavía en esta época sigo considerando como si les faltara algo para ser completos..

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.

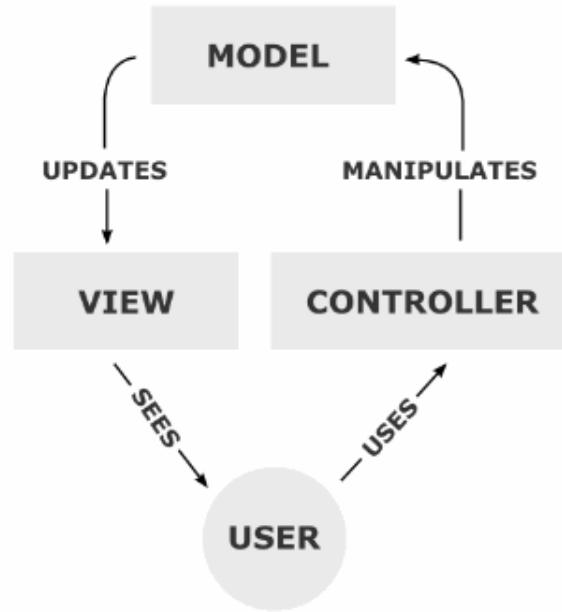


Figura 2.1: Diagrama del Patrón MVC Modelo Vista Controlador

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:

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:

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.

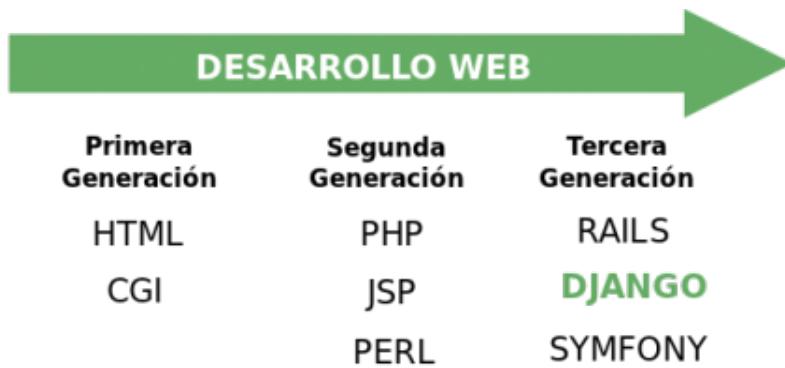


Figura 2.2: Generaciones de Herramientas de Desarrollo Web

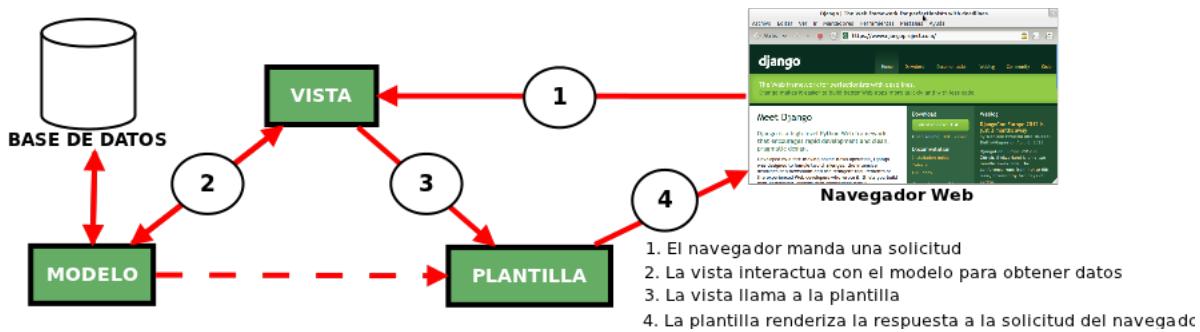


Figura 2.3: El patrón Modelo Vista Template de Django

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

```

```

16     verbose_name = "InboxMessage"
17     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( \
10                                         user__groups__name='Medico')
11
12     html_cont = mi_template.render(Context(dict))
13     return HttpResponseRedirect('/restricted-access')
14
15 else:
16     #si hay un usuario logueado intentanto acceder sera enviado a una
17     # pagina de error
18     path = request.META['PATH_INFO']
19     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')
```

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.

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 sesión con el usuario <strong>{{ username }}</strong></div>
        <br />
        <a href="/logout">Cerrar Sesión</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),

```

2.6.7. Porque Django y no otro Framework

Django no es el único framework disponible para Python existen varias alternativas grandes como TurboGears, Web2Py, Grops, Pylons, Zope2, Bottle, Flask, se puede consultar una extensa lista de alternativas disponibles en <https://wiki.python.org/moin/WebFrameworks>.

Django termino siendo el elegido por razones de que es fácil de instalar extensiones, muy buena documentación, gran aceptación como principal framework web en Python, en constante evolución, posee una comunidad de desarrolladores activa (y en español), curva de aprendizaje relativamente corta, posee un sistema de plantillas que permite usar herencia, separa el diseño de la Interfaz del resto de la programación, completa interfaz administrativa que se genera automáticamente (ósea tienes la mitad del trabajo hecho) entre otros por nombrar las características por las que me decidí usarlo, vengo de frameworks en otros lenguajes y al utilizarlo por primera vez simplemente me pareció mágicos.

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 Clínica

Actualmente en la mayoría de los centros de salud la información que compone la Historias Clínicas de cada paciente es almacenan mediante documentos físicos, en su mayoría totalmente elaborados a mano, en otros casos usando plantillas con campos preformatados (como se puede observar en las figuras 3.1 y 3.2), lo cual genera varios problemas.

En cuanto los diferentes centros de salud donde se pudo relevar al software existente y su aplicación a la manipulación de historia clínica, este suele ser muy incompleto y abocado a la especialidad del médico, es por ello que si esta informatizado cada área suele manejar sistemas que terminan siendo incompatibles entre sí.



Figura 4.1: Almacenamiento Físico 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)

1	Presentación	Dosisificación	Medicamentos (principio activo)	Presentación	Dosisificación
2					
3					
4					

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 Clínica Ministerio de Salud de La Nación Pag 1

	T.A.	F.C.	Peso	Talla	IMC.	Perímetro abdominal
• Examen Físico	Piel Ojos: ¿usa lentes? Agudeza visual Oídos Dentadura Pulmones Corazón Abdomen Genitales externos Mamas	Sin lesiones significativas no si último control OD OI Otoscopía normal en buen estado buena entrada sin ruidos agregados ritmo regular, sin soplos blando, indoloro, sin visceromegalias normales sin nódulos	Hallazgos anormales			
• Lista de problemas	Problema Código Fecha inicio Fecha resolución	Problema Código Fecha inicio Fecha resolución	Problema Código Fecha inicio Fecha resolución	Problema Código Fecha inicio Fecha resolución	Problema Código Fecha inicio Fecha resolución	Problema Código Fecha inicio Fecha resolución
• Cuidados preventivos	Intervención Consejo alimentación saludable Consejo actividad física Consejo salud sexual y reproductiva Consejo anti tabáquico Consejo prevención de accidentes Vacuna doble adultos Vacuna hepatitis b Vacuna sarampión - rubeóla Vacuna antigripal Vacuna antineumocócica Otra Examen odontológico Agudeza visual Tensión arterial Índice de masa corporal (peso/talla) Perímetro abdominal Colesterol total plasmático Glucemia Riesgo cardiovascular global Sangre oculta en materia fecal Mamografía Papanicolaou	Periodicidad / Población Variable, según riesgo Variable, según riesgo Variable, según riesgo Variable, según riesgo Variable, según riesgo 1 dosis cada 10 años 3 dosis (0-1-6 meses) en grupos de riesgo 1 dosis en personas No inmunizadas 1 dosis anual en grupos de riesgo según calendario 1 dosis en grupos de riesgo. Evaluar revacunación Variable según riesgo regional o campaña Anual, población general Anual, población general Variable, según el riesgo Anual población general para su cálculo: Anual población general. Medición con centímetro a la altura del ombligo Variable, según el riesgo Anual. Mayores de 45 años. Menores de 45 años IMC >29.9 y FRCV Variable según riesgo. Población general mayor de 39 años sin enfermedad cardiovascular previa Anual. Mayores de 50 años Anual mujeres mayores de 50 años a 69 años Mujeres a partir de los 25 años, 2 controles Consecutivos anuales normales, luego controles cada 3 años	Mes/año			
*Recomendaciones orientadas sujetas a modificaciones y actualización según guías de práctica de la autoridad sanitaria						

Figura 4.3: Modelo Historia Clínica Ministerio de Salud de La Nación Pag 2

4.2. Problemas del Sistema Actual

Nos referimos al *Sistema Actual* en cuanto a cómo funcionan las organizaciones en nuestro caso consultorios médicos y solo hablamos del software que haya implantado sino al funcionamiento del mismo viendo la organización desde un enfoque sistémico¹. Paso a describir algunos de los problemas a los que se enfrentan las organizaciones cuando almacenan las historias clínicas en formato papel:

4.2.1. Almacenamiento

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

4.2.2. Búsqueda y Localización

La Búsqueda de expedientes se puede agilizar un poco utilizando una buena organización, el problema es que la mayoría de los edificios para tal fin, suelen estar saturados por grandes volúmenes de archivos físicos, 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 físicos son:

- Perdida, alteración o dato de documentos importantes
- Gasto excesivo en fotocopias
- Altos costos de personal para administrar, suplir, mantener y recuperar el archivo físico
- Costos asociados al transporte de documentos ya sea interna o externamente
- Costos asociados al espacio físico requerido para su almacenamiento
- Falta de condiciones adecuadas para el almacenamiento de documentos como: ventilaciÃ³n, humedad, temperatura
- Falta de respaldo adecuado en caso de catástrofe como incendio, inundaciÃ³n o terremoto

¹Bajo este enfoque el mundo se organiza en torno a sistemas que funcionan y a través de los cuales se ordena la sociedad. El elemento central del enfoque sistémico es la estructura, que permanece estable a pesar de los cambios sociales que se produzcan. Se puede consultar más acerca del Enfoque Sistémico o la TGS en [27].

4.2.5. La Solución Planteada

Por ello este era un escenario perfecto donde es necesario informatizar el actual sistema, lo cual solucionaría los 2 principales problemas del mismo que son el Excesivo espacio de almacenamiento y el lento trabajo de búsqueda además de:

- Reducir costos de personal administrativo, ya que las búsquedas y registro las hará el sistema.
- Brindar la información de manera rápida en situaciones críticas que requieren un rápido accionar por parte del médico.
- Disponibilidad en todo momento y cualquier lugar para consulta por parte de los médicos ya que solo requerirá disponer de un usuario y un ordenador con conexión a Internet para poder consultar.

Es cierto que los sistemas informáticos sufren problemas de almacenamiento, búsqueda (en el caso de grandes volúmenes de información) y deterioro por el paso del tiempo pero en este caso el primero se soluciona agregando más espacio de disco, cosa que hoy en día es algo relativamente barato a razón de 1 peso = 1 Gb.².

El problema de las búsquedas no afecta mucho con la velocidad de los equipos actuales se puede consultar bases de datos con millones de registro en unas pocas milésimas de segundos dicho tiempo resulta imperceptible para la persona en la mayoría de las veces, en todo caso dependerá de la implementación y el motor de bases de datos más 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 algún momento, pero esto es salvable siempre y cuando se realicen buenas prácticas tales como implementar un sistemas de backup, también replicación de datos en caso de que se necesite alta disponibilidad de la información que se almacena.

4.3. Gestión de Turnos

En lo que respecta a asignación de turnos, el sistema actual en la mayoría de los casos no ha tenido un mejor panorama en cuanto a implantación de un software, aunque ya en esta área existen algunas aplicaciones que intentan solucionar el problema de manera más o menos eficientes.

²Gb hace referencia a Gigabyte que es una medida utilizada en informática 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 área 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 hospitales.

5.3. Arquitectura de la Aplicación

Implementado en Python utilizando en Framework Django, utilizando el motor de bases de datos PosgreSQL, 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 hubiese notado.³

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.

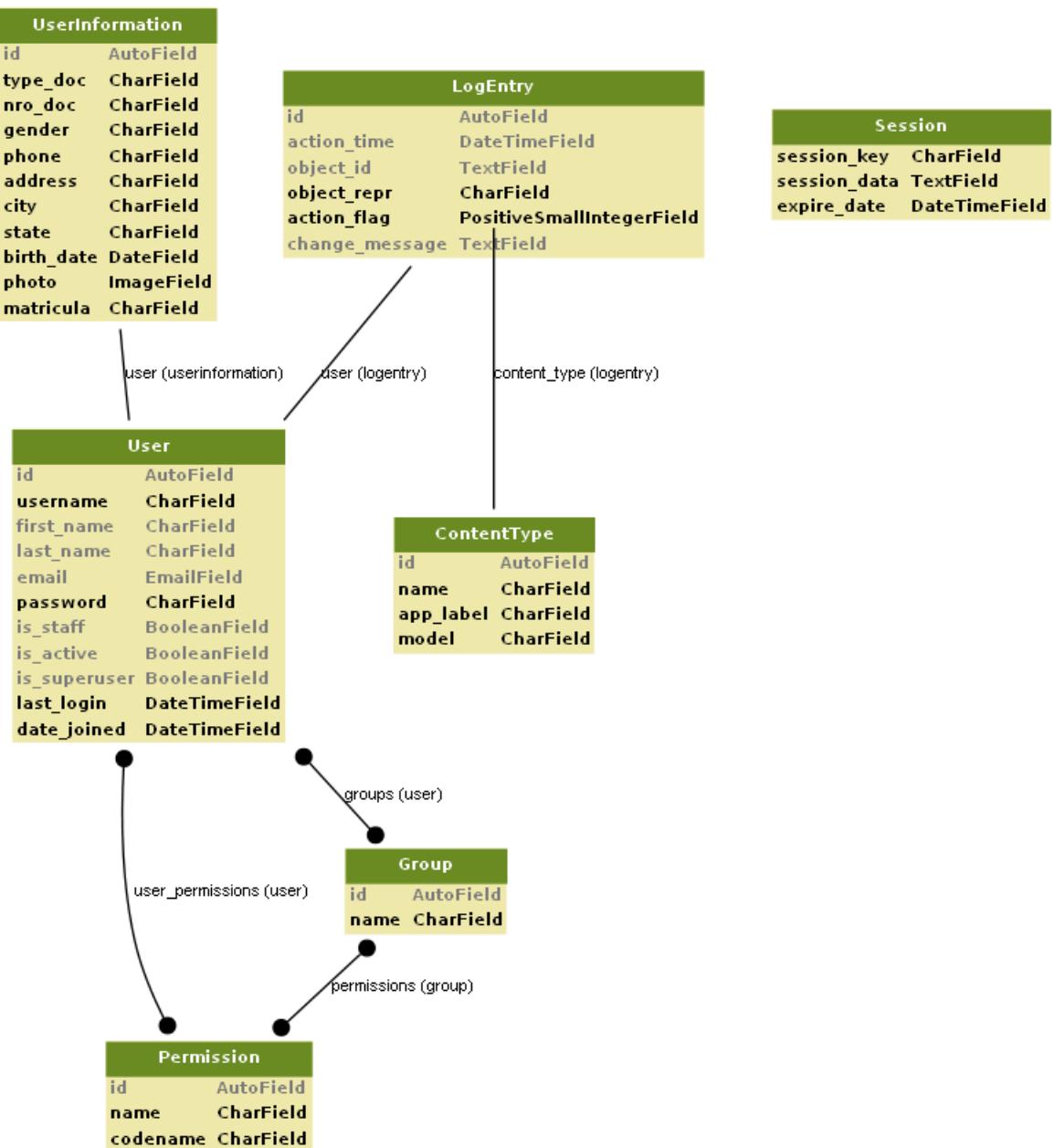


Figura 5.1: Diagrama con modelos que componen el modulo Usuarios

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 enviara al correo antes de poder comenzar a usar su cuenta, en los otros casos (el usuario Paciente es registrado por un médico 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**(esta última 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 sería <http://127.0.0.1/admin/>⁴.

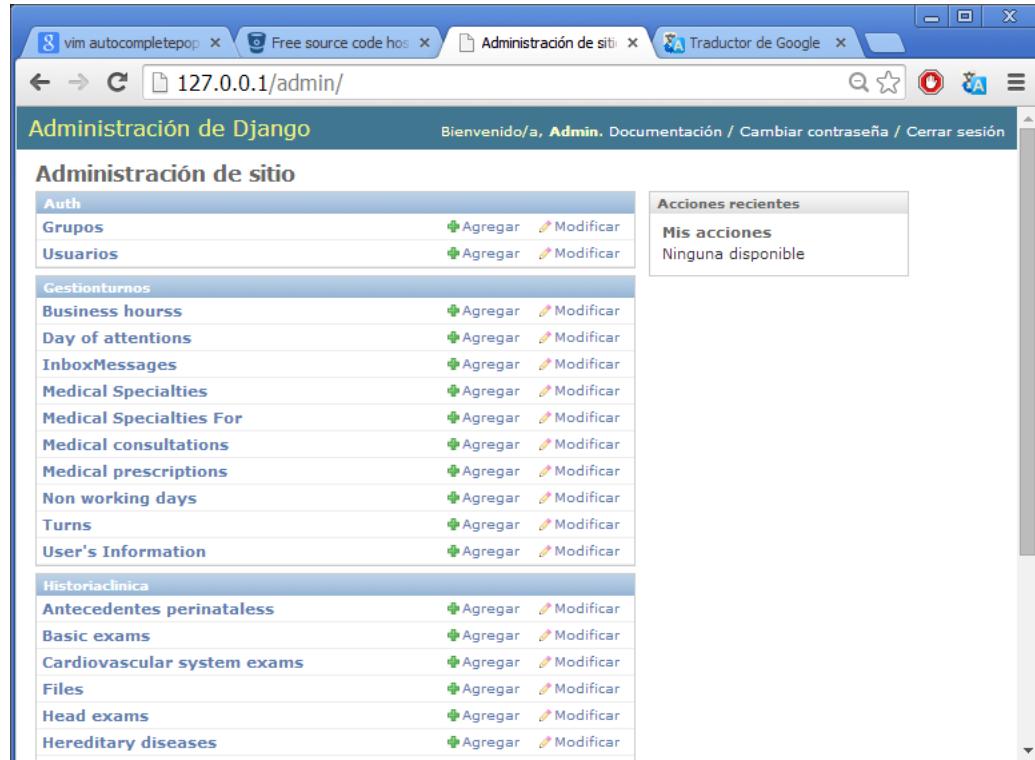


Figura 5.2: Vista del Panel Administración provisto por Django

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*.

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

⁴Se puede consultar más 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.

- 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.

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:

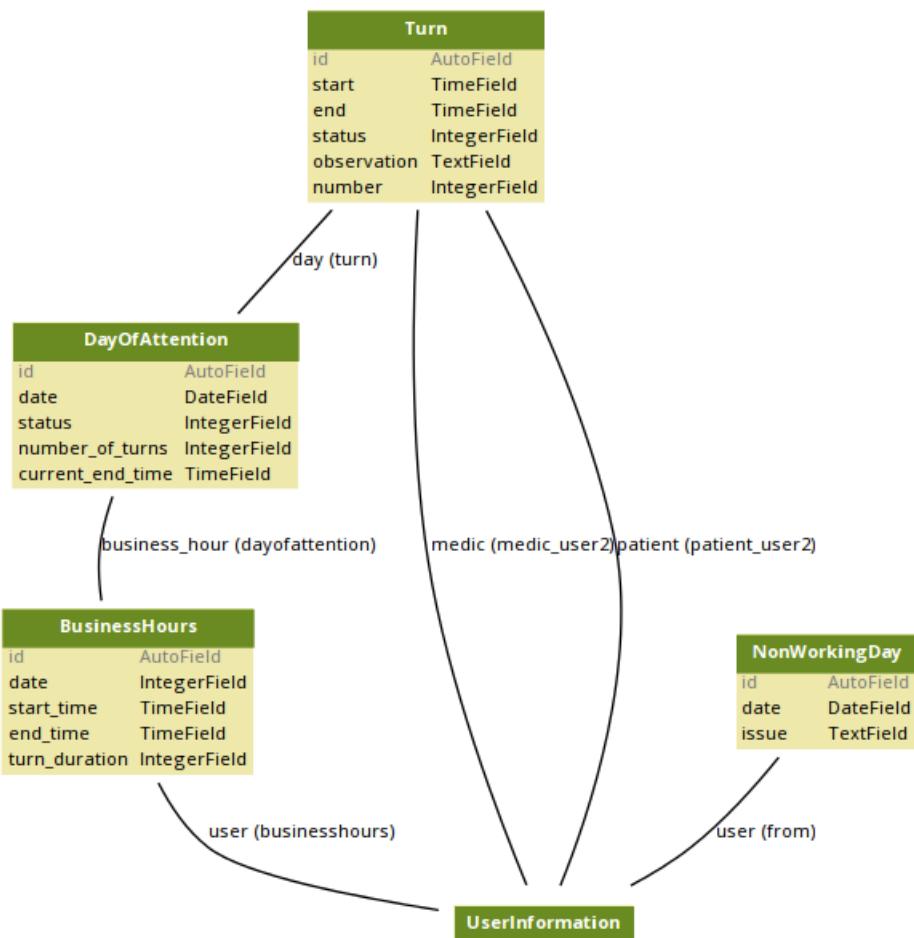


Figura 5.3: Modulo Gestión Turnos - Diagrama Modelos correspondiente a la Gestión de Turnos

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

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

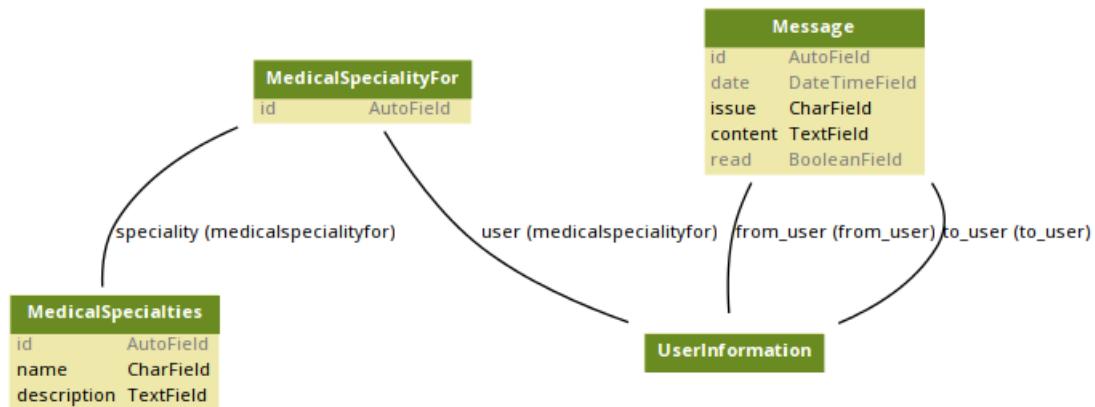


Figura 5.4: Modulo Gestión Turnos - Modelos Adicionales

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.

⁷Tengase 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.

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 él 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 número 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.
- **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

⁸ Esto 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 médico.

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

- **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 modificará 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.

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.

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

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:

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 Lo 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.

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

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 forzados. 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 explora 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 interconsultas¹³ y las observaciones del médico, así como los medicamentos que fueron recetados por el especialista.

5.6.4. Definición de Modelos

Los modelos que componen el Modulo corresponden a los diferentes tipos de exámenes de práctica 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:

¹²Esto último es solo para la versión en papel

¹³Consultas medicas realizadas por el paciente

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.

5.6.5. Modelos Básicos

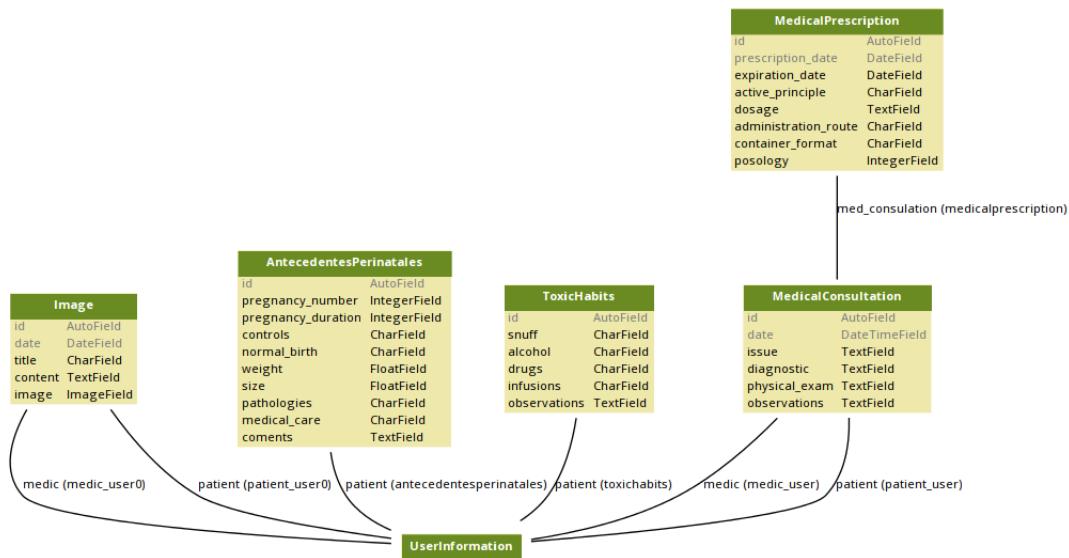


Figura 5.5: Historia Clínica Modelos Básicos

5.6.5.1. Image

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

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

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

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 medicas 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 invalida.
- **active_principle** Principio Activo o nombre genérico de la mono droga 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 cuanto 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** Numero 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** Presento complicaciones al nacer.
- **medical_care** Requirió atención medica especial.
- **observations** Otras anotaciones.

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 mimo 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.

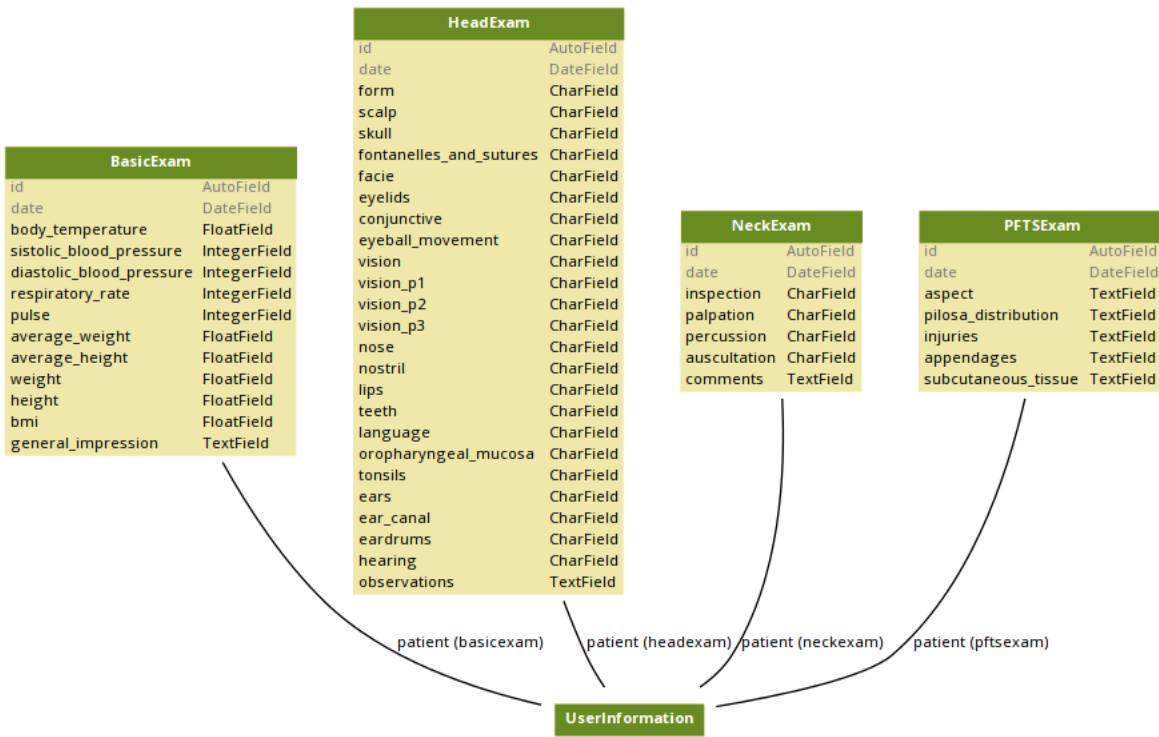


Figura 5.6: Historia Clínica Modelos de Estudios

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.

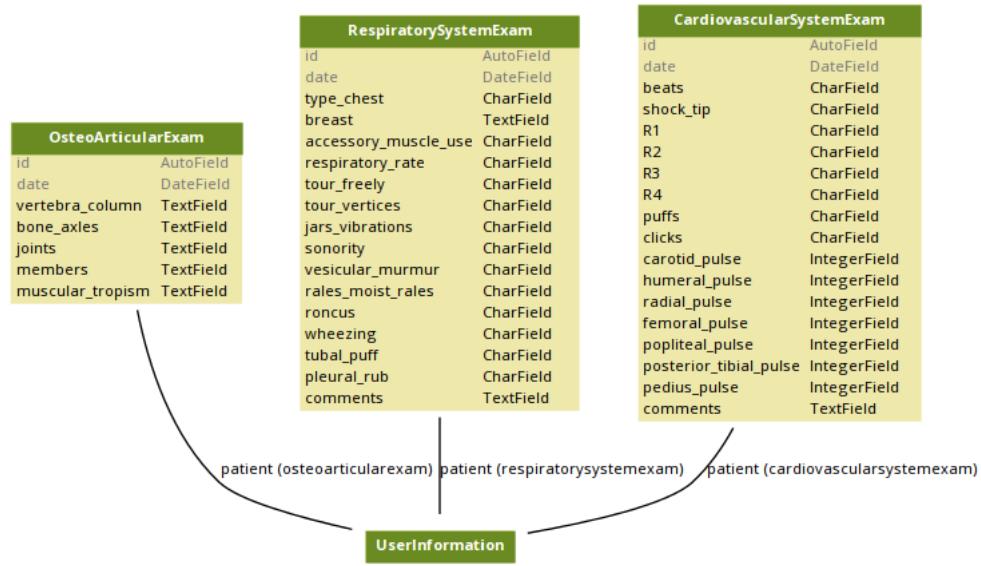


Figura 5.7: Historia Clínica Modelos de Estudios Continuación

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:

- Simetria 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 Accesoriales 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 Poplítleo
- Pulso Tibial Posterior
- Pulso Pedio
- Observaciones Generales

Capítulo 6

Instalacion y Configuracion

Este Capítulo pretende ser una guía para realizar una exitosa implementación del servidor de producción¹ la principal consideración es que la implementación será sobre el sistema operativo Windows 7, en caso de querer instalar el sistema en otro sistema operativo, existen algunas diferencias.

6.1. Requerimientos

6.1.1. Requerimientos de Hardware

Cualquier equipo que cumpla con las características para correr Windows 7 es suficiente en términos de requerimientos mínimos de Hardware siempre y cuando el número de usuarios esperados no sea alto, después el resto dependerá de sus necesidades.

- Procesador x86, x64 de 1.6 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

¹Diferenciamos servidor de producción porque durante el desarrollo las aplicaciones en Django se prueban en un servidor virtual que crea Django para que se pueda correr la aplicación fácilmente, pero esto no es suficiente a la hora de utilizar la aplicación con usuarios finales principalmente por que el servidor de desarrollo no es lo suficientemente potente además de ser lento y consumir demasiados recursos, esta más que nada pensado para que el desarrollador pruebe la aplicación sobre él para así luego pasar a una versión de producción.

- ReportLab
- easy_thumbnails
- django_extensions
- django_cron

6.2. Apache

Existen 2 caminos para instalar Apache, la primera Hacer una instalación limpia de apache, la 2da es cuando no se quiere trastear con tanta configuración por lo que opta por infraestructuras tipo WAMP, LAMP, WAPP, etc.

6.2.1. Instalación en Limpio

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

Descargamos de Apache.org la última versión disponible, se puede utilizar el siguiente vínculo: <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 ícono 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. Instalación mediante WAMP, LAMP, MAMP, WAPP

Existen una infinidad de Paquetes precompilados y configurados, con Apache, PHP, PosgreSQL o MySQL y más.

Dichas infraestructuras suelen nombrarse como el acrónico 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 de las distribuciones más 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 instalación nos pedirán el usuario y contraseña para acceder al motor de Base de Datos.

6.2.3. Configuración

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 instalación en limpio o **C:\WAMP\bin\Apache\conf** si instalamos el paquete múltiple 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 o algo más avanzado como Notepad++, Geany, etc.

Buscamos la línea que dice

```
Listem LocalHost:80
```

Y la Cambiamos por:

```
Listem 80
```

Ahora buscamos la instrucción:

```
DocumentRoot "C:\xxxxxxxx"
```

La Cambiamos por:

```
DocumentRoot "C:\Servidor"
```

Recordar que al inicio de la instalación creamos una carpeta llamada Servidor en la unidad C. Por último solo nos queda reiniciar el servidor Apache e introducir la siguiente dirección <http://127.0.0.1> si nos aparece una página **It's Work!** felicidades Apache está Funcionando. El problema más común por lo que aveses apache no inicia se debe a que el puerto puede estar siendo utilizado por otra aplicación como Skype, para ello asegúrese antes de que el puerto 80 este libre o utilice otro puerto como el 8080.

6.2.4. Instalación de PosgreSQL

La versión de PostgreSQL que he utilizado durante el desarrollo del sistema es la 9.2.x, quizás cuando leas esto ya halla salido una nueva versión la cual no debería generar inconvenientes además de que es posible que el proceso de instalación pueda variar.

El primer paso es descargar el instalador de PostgreSQL para Windows, el cual 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 mostrara 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 instalacion de PostgreSQL, donde se guardarán los ejecutables, librerías y ficheros de configuración de PostgreSQL en mi caso el directorio es **C: \PostgreSQL \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 súper usuario ”postgres”² que será con el que iniciemos sesión para administrar la base de datos, después podremos crear otros usuarios si es necesario. Además introduciremos el puerto de escucha para la conexión con el servidor PostgreSQL, por defecto el 5432.

Seleccionaremos la configuración regional y comenzara la instalación, con esto PostgreSQL quedara instalado. Si tenemos algún cortafuego (firewall) deberemos abrir el puerto 5432.

6.2.5. Creación de la Base de Datos

Junto con la Instalación de PosgreSQL se instala el PGAdmin III que es una herramienta GUI³ 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 súper usuario 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 ”postgres”.
- 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 vacía, el resto como creación de las Tablas correspondientes necesarias para el proyecto lo haremos más adelante mediante Django.

6.3. Instalación de Python

Para este proyecto se utilizo CPython pero no la versión 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 características adicionales a versión oficial, podremos descargar la ultima versión desde <http://www.activestate.com/activepython/downloads> aunque se recomienda instalar la version 2.7.x para evitar cualquier posible problema ya que es incompatible con las versión 3.x de Python.

²Puedes cambiar la contraseña y crear un nuevo usuario para aumentar la seguridad del sistema solo recuerda que deberás cambiar la configuración de Django para que este pueda conectarse a la base de datos con un usuario diferente.

³GUI (del inglés *graphical user interface*) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz

6.3.1. Probando Python

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

```
python
```

Si todo va bien nos deberá aparecer algo similar a:

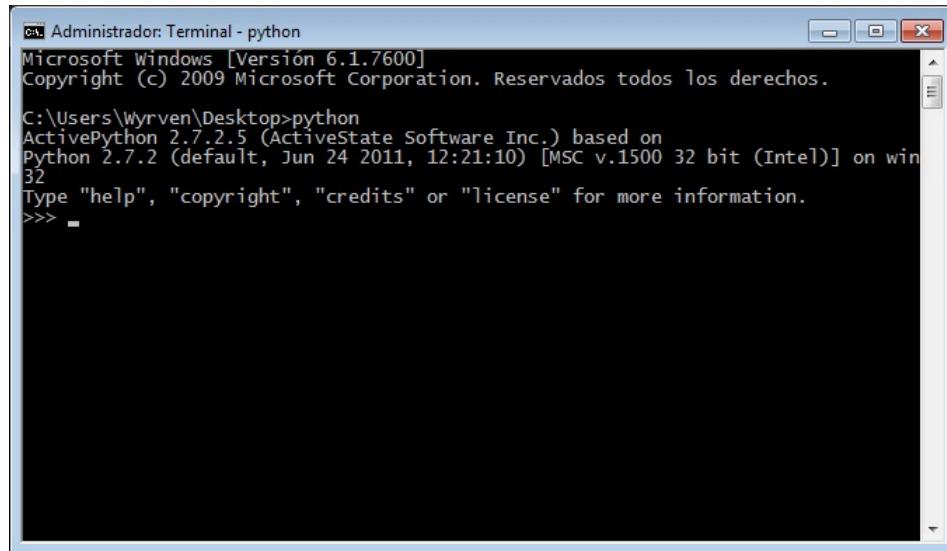


Figura 6.1: Ejecutando Python en la Terminal

En caso contrario deberías 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/>⁴ te descargara un paquete llamado Django-1.3.7.tar.gz lo descomprimes en algún 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 más automática 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 librerías de Python las cuales algunas vienen instaladas y otras requieren ser instaladas de manera similar a como instalamos Django.

⁴la versión 1.3.7 no es la última versión disponible a la hora de crear este informe estaba por la 1.6.2 ya que Django se actualiza constantemente.

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 cursoras y hacen que un número notable de inserciones o actualizaciones concurrentes. psycopg2 también proporcionan operaciones asíncronas completas y apoyo a las bibliotecas de co-rutinas.

Para instalar descargue el precompilado desde <http://www.stickpeople.com/projects/python/win-psycopg/> ejecútelo con permisos de administrador, nos pedirá que seleccionemos la versión de Python con que se instalar.⁵

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 (funciona tanto en Linux como Windows).

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 thumbnailing⁶, pero fácil de implementación 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
```

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 terminal⁷

```
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, se adjunto con el código fuente del Proyecto. Igualmente si tiene curiosidad puede visitar la página del proyecto <https://github.com/Tivix/django-cron>

⁵Se podría instalar usando el código fuente con pip pero para ello se requiere tener alguna versión del compilador Visual C++ ya que parte de la librería fue portada desde C++

⁶Cuando 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.

⁷Importante, no todas las funcionalidades están soportadas en Windows, pero en cuanto a la requeridas por el proyecto no hay problemas.

6.5.6. Descargar e Instalación de mod_wsgi

Asumiendo que ya tienes instalado Python y Apache, solo debes descargar el paquete libapache2-mod-wsgi ,la ultima versión de mod_wsgi se puede descargar desde su página oficial <https://code.google.com/p/modwsgi/> descargaran un archivo similar a "mod_wsgi-win32-ap22py27-3.3.so" la versión que descarguen de mod_wsgi depende como se ve, de la plataforma así como de la versión 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 línea en el mismo punto donde se cargan el resto de los módulos.⁸

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

Con todo esto hecho 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. Configuracion del Proyecto

Bueno Ahora solo tenemos que crear un alias en Apache⁹ para nuestra carpeta donde colocaremos en mi caso la carpeta destino será:

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

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

```
/sgcm/
```

Tendremos que agregar las siguientes líneas 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 puede instalar una aplicación WSGI organizada por mod_wsgi Puede consultar <https://code.google.com/p/modwsgi/wiki/QuickInstallationGuide> si desea explorar otras opciones de configuración.

⁸El archivo httpd.conf esta en la siguiente ruta en el caso de mi instalación: C:\Apache\conf\httpd.conf

⁹Para mayor información de cómo crear alias en Apache consulte <http://httpd.apache.org/docs/2.2/urlmapping.html>

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 solo 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 si 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 aquí 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
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 aplicación ya debería funcionar aunque como verán no se cargan los archivos estáticos como imágenes 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 la siguiente información a http.conf:

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

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

También puedes usar *LocationMatch* para comparar con una expresión regular. Por ejemplo, esto configura Django en la raíz 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 estará funcionando correctamente, y podrá cargar imágenes y demás ficheros necesarios.

6.7. Configuración Inicial

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

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

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

Esta instrucción le dice a Django que sincronice los modelos con la base de datos y que cree las tablas necesarias en la misma para que esto funcione, además 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 configuración inicial.

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

```
$ python init_app.py
```

El mismo creara toda la información requerida por la aplicación y con ello quedara nuestro servidor en funcionamiento, solo hará falta reiniciar Apache. Con esto concluye todo lo referente a configuración del servidor.

Capítulo 7

Guia de Referencia

El presente capítulo no pretende ser un completo manual de usuario de la aplicación, el sistema en si es bastante intuitivo en cuanto a su funcionamiento por lo que en este capítulo se resumirá un poco algunas de las diferentes sesiones y funcionalidades de la aplicación.

7.1. Organización de la Aplicación

La aplicación se organiza de la siguiente manera, con las diferentes áreas bien definidas:

- 1 Menú Principal
- 2 Menú Secundario
- 3 Cuerpo de la Aplicación
- 4 Información de Usuario



Figura 7.1: Organización Espacial del contenido de la aplicación

7.1.1. Menú Principal

El contenido del menú principal depende del tipo de usuario que haya iniciado sesión en base a ello tendrá o no habilitadas diferentes funcionalidades de la aplicación, los únicos menús comunes son **Mensajes** y **Opciones**.

7.1.2. Menú Secundario

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

7.1.3. Cuerpo de la Aplicación

Aquí se localizará el contenido principal de la vista, ya sea un formulario para registrar alguna información, una lista para mostrar información etc.

7.1.4. Información de Usuario

Muestra información acerca de la sesión de usuario actual que se está ejecutando, tal información es el nombre del usuario ¹ y el tipo de usuario que puede ser (Paciente, Médico, Administrativo, Not Login en caso de no haber iniciado sesión)

7.2. Panel de Usuario No Registrado

Corresponde al panel que verá el usuario la primera vez que ingrese a la aplicación las funciones que se pueden hacer son restringidas y se limitan a:

- **Inicio:** Ir a la pantalla de Inicio
- **Listado de Médicos:** Mostrar información básica acerca de los especialistas con los que cuenta la institución.
- **Registrarse:** Permite al usuario mediante una serie de pasos registrarse como paciente.
- **Iniciar Sesión:** Iniciar una sesión con un usuario registrado.

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

La vista **Listado de Médicos** puede consultarla en la sesión del panel del Paciente, ya que la única 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 información.

7.2.1. Registrarse

Esta vista ofrece a los usuarios no registrados, un formulario donde deberán 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 (*) :	*****
Repite 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:

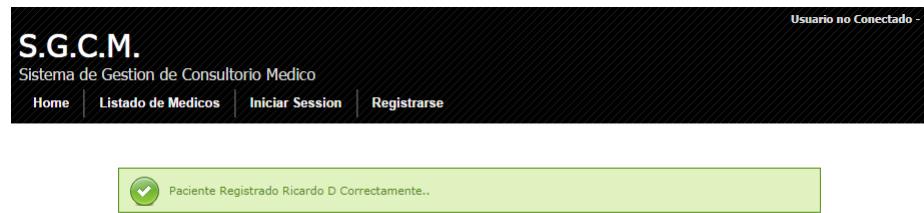


Figura 7.3: Formulario Registro Paciente

Paso siguiente deberemos revisar nuestra casilla de correo donde nos aparecerá el mensaje con la dirección del formulario para activación de usuario.²

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

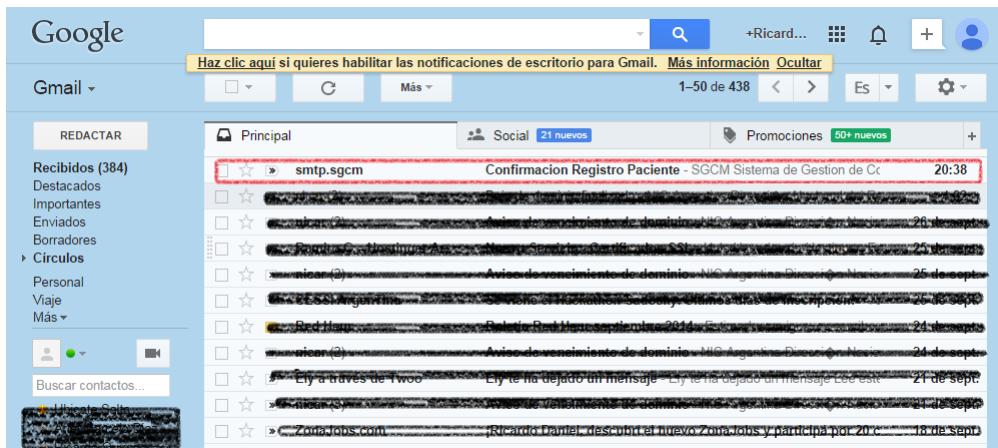


Figura 7.4: Bandeja de Correo con el mensaje

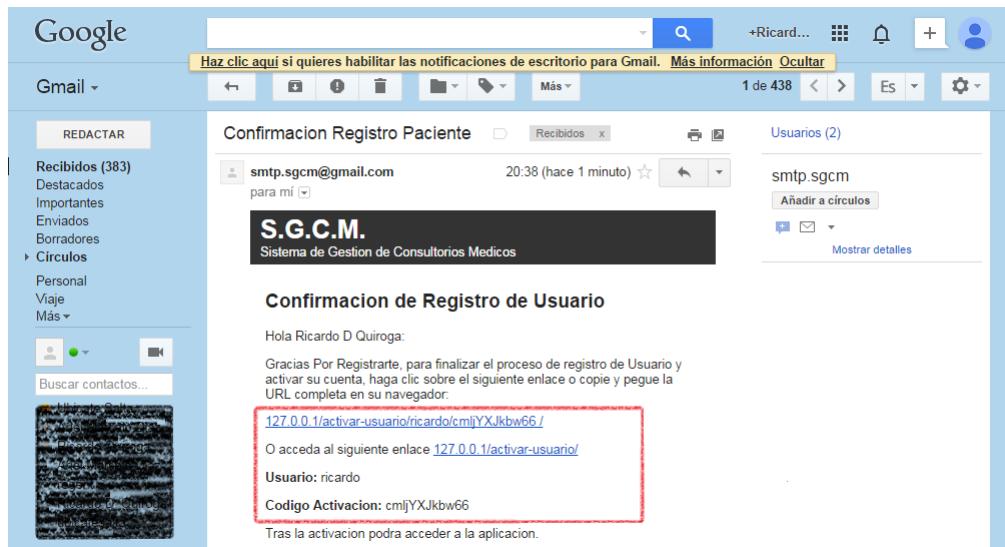


Figura 7.5: Cuerpo del mensaje con la información de activación de usuario

The screenshot shows a web page titled 'Activar Usuario'. It has a header 'S.G.C.M.' and 'Sistema de Gestión de Consultorios Medicos'. At the top right, it says 'Usuario no Conectado -'. Below the header, there are navigation links: 'Home', 'Listado de Medicos', 'Iniciar Sesión', and 'Registrarse'. The main content area is titled 'Activar Usuario' and contains a form with two input fields: 'Usuario:' (with value 'ricardo') and 'Codigo Activacion:' (with value 'cmjYXJkbw66'). A blue button labeled 'Activar Usuario' is at the bottom of the form.

Figura 7.6: Formulario de Activación de usuario

Luego de estos pasos el usuario estará registrado y activado, solo faltaría iniciar sesión para poder empezar a operar como paciente.

7.2.2. Iniciar Sesión

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

7.3. Panel de Usuario Paciente

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

- **Inicio:** Ir a la pantalla de Inicio
- **Listado de Médicos:** Mostrar información acerca de los especialistas.
- **Mensajes:** Casilla de Mensajes Internos.
- **Mis Turnos:** Información acerca del estado de los turnos del usuario.
- **Opciones:** Panel de Opciones

7.3.1. Listado de Médicos

Vista que permite seleccionar entre el listado de especialistas que componen el cuerpo médico de la institución consultar información, realizar una consulta rápidas y solicitar turno.



The screenshot displays the 'Listado de Médicos' (List of Doctors) page of the S.G.C.M. system. At the top, there is a header with the logo 'S.G.C.M.' and the text 'Sistema de Gestión de Consultorio Medico'. On the right side of the header, it says 'Quiroga, Ricardo D - (Paciente)'. Below the header is a navigation menu with links: Home, Listado de Medicos, Mensajes, Mis Turnos, and Opciones. The main content area is titled 'Listado de Medicos' and shows two doctor profiles: 'Quiroga, Ricardo' and 'Roldan, Juan'. Each profile has a small placeholder image and the doctor's name.

Figura 7.7: Listado de Médicos

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

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

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 Asignación de turno son similares, entre si no varían mucho por lo que se explicara en la parte del panel de Médico.

7.3.2. Mensajes

Como ya se menciono esto es una sesión común a todas los usuarios, se trata de un conjunto de vista donde funciona el sistema de mensajería interna entre los usuarios de la aplicación, su función está reducida en cuanto al usuario paciente ya que este solo puede visualizar y responder los mensajes que se le envían, para enviar un mensaje a un profesional se realiza mediante la opción

de realizar consulta online, por lo que solo puede enviar mensaje a los especialistas, las opciones disponibles 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:

La captura 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'. A la derecha de este cuadro hay un botón 'Enviar'.

Figura 7.10: Redactar un Mensaje

La captura 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

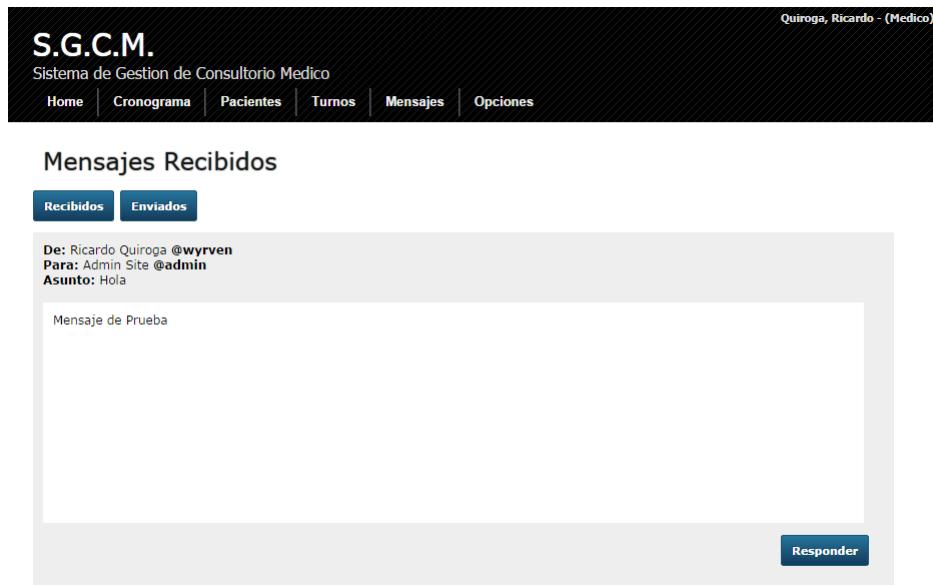


Figura 7.12: Mostrar Mensaje

7.3.3. Opciones

Otro menú común a todos los usuarios permite la administración de los datos e información del mismo, dentro de las funciones que permite este menú se encuentra:

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

7.4. Panel de Usuario Administrativo

Corresponde al panel de funciones al que tendrán acceso los usuarios administrativos .

- **Inicio:** Ir a la pantalla de Inicio
- **Pacientes:** Administrar Usuarios Pacientes
- **Médicos:** Administrar usuarios Médicos
- **Administrativos:** Administrar usuarios Administrativos
- **Especialidades:** Administrar especialidades medicas
- **Mensajes:** Casilla de mensajes Internos.
- **Opciones:** Panel de opciones

7.4.1. Pacientes, Médicos, 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 menú se enlista de acuerdo a los tipos de usuarios que se desea administrar, permitiendo según el sub menú la posibilidad de crear un tipo de usuario específico ³ Buscar un usuario, modificar 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

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 búsqueda de usuario, en este caso usuarios Pacientes

En el caso de los usuarios médicos además puede consultar y modificar el estado de los turnos que les fueron solicitados, definirles especialidades correspondiente.

³Los Usuarios registrados por el Administrador o el médico no requieren activación 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 right side of the header, it says "Site, Admin - (Administrativo)". Below the header, there is a sidebar titled "OPCIONES" containing links: Mostrar Datos, Modificar Datos, Cambiar Contraseña, Turnos Solicitados, and Agregar Especialidad. The main content area is titled "Datos del Medico" and displays the following data for a doctor named Ricardo Quiroga:

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

Below this, there is a section titled "Listado Expecialidades Medicas" (List of Medical Specialties) with a table:

Nombre	Opciones
Medico Clínico	Quitar

Figura 7.15: Mostrar Medico

The screenshot shows the S.G.C.M. interface again. The navigation bar and sidebar are identical to Figure 7.15. The main content area is titled "Turnos Solicitados" (Requested Appointments) and shows a list for Doctor Ricardo Quiroga. The table lists various appointment requests:

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

Figura 7.16: Vista Listado Turnos Solicitados al Medico.

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

The screenshot shows the S.G.C.M. application interface. At the top, there is a navigation bar with links: Home, Pacientes, Medicos, Administrativos, Especialidades, Mensajes, and Opciones. On the right side of the header, it says "Site Admin - (Administrativo)". Below the header, there is a sidebar titled "OPCIONES" containing 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 section is another titled "Turnos Solicitados" which lists scheduled appointments:

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 Médicos

Corresponde al panel de funciones al que tendrán acceso los usuarios médicos aunque comparte funcionalidades en común con los otros paneles, presenta vistas exclusivas para el uso por parte de los médicos.

- **Inicio:** Ir a la pantalla de Inicio
- **Cronograma:** Vista con las actividades pendiente para el día.
- **Pacientes:** Administrar Usuarios Pacientes
- **Turnos:** Información acerca del estado de los turnos del médico.
- **Mensajes:** Casilla de Mensajes Internos.
- **Opciones:** Panel de Opciones

7.5.1. Cronograma

La funcionalidad de esta vista es permitir un acceso rápido a la funciones comunes, mostrar los mensajes sin leer, y los turnos solicitados para la fecha.

The screenshot shows the S.G.C.M. (Sistema de Gestión de Consultorio Médico) interface. At the top, it displays "Quiroga, Ricardo Daniel - (Médico)". Below the header, there's a navigation bar with links: Home, Cronograma, Pacientes, Turnos, Mensajes, and Opciones. The main content area is titled "Cronograma del Dia" and shows the date "Fecha: 3 Nov. 2014". A section titled "Consultas Sin Responder" indicates "Usted Tiene 1 Mensajes En su bandeja sin Responder" and provides a link "Ir a la Bandeja de Mensajes". Another section titled "Turnos Solicitados" lists three patients: Marcela Gutierrez, Pedro Marquez, and Pablo Aguirre, all marked as "Pendiente" (Pending) with "Mostrar" (Show) options.

Figura 7.18: Vista del Cronograma

7.5.2. Pacientes

La colección de vistas correspondientes a los pacientes son similares a las vistas que presenta el rol administrativo, como se mencionó antes el médico también puede registrar nuevos pacientes, lo que diferencia a las vistas de medico correspondiente a esta sesión es que ahora muestra un menú más amplio de opciones en comparación a la del administrador, se habilita el submenú para administrar lo referente a la Historia clínica del paciente.

The screenshot shows the S.G.C.M. interface for viewing a patient. The top navigation bar includes "Home", "Cronograma", "Pacientes" (highlighted in blue), "Turnos", "Mensajes", and "Opciones". On the left, a sidebar lists "INFORME GENERAL", "CONSULTAS MEDICAS", and "HISTORIAS CLINICAS" with various sub-options. The main content area is titled "Datos Paciente" for "Ortiz, Marcelo". It displays the patient's information in a table format:

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

Buttons for "Atras", "Modificar", and "Borrar" are located at the top right of the patient details box. There is also a placeholder for a patient profile picture.

Figura 7.19: Mostrar Paciente

La Primera parte corresponde al registro de información relevante obtenida durante la consultas medicas y prescripciones medicas ⁴ además de generar el comprobante para el paciente.

⁴Lo que comúnmente se conoce como receta médica.

Modificar Datos Consulta Medica

Medico: Quiroga, Ricardo
Paciente: Ortiz, Marcelo

Motivo: prueba

Examen Fisico: no detectado

Diagnóstico: resfriado

Observaciones: hacer reposo.

Guardar Cambios **Volver**

Prescripciones Medicas Asociadas

Agregar

Fecha	Medico	Principio Activo
22 Mayo 2014	Quiroga Ricardo	Amoxisilina

Figura 7.20: Consulta Médica

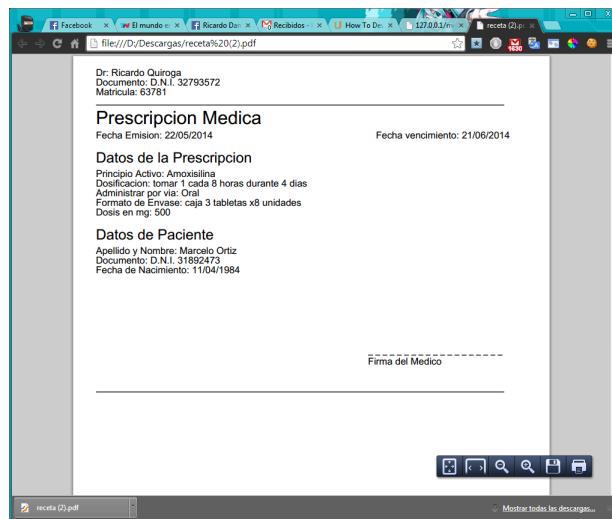


Figura 7.21: Imprimir Receta o Prescripción Medica

En cuanto a los estudios clínicos que se registran en la historia clínica, por lo general al seleccionarlo puede que se visualice:

La información del examen únicamente, con la opción de modificación si este es único, por ejemplo el caso de registro de antecedentes perinatales.

The screenshot shows the S.G.C.M. interface. At the top right, it says "Quiroga, Ricardo Daniel - (Medico)". The main title is "Antecedentes Perinatales". Below it, the patient's name is listed as "Gutierrez, Marcela". There is a "Modificar" (Modify) button. The form contains the following data:

Embarazo Numero:	0	Duracion del Embarazo:	0 Semanas
Controles durante embarazo:	NO	Parto Normal:	NO
Peso al nacer:	0 kg.	Talla al nacer:	0 cm.
Presento Patologias al Nacer:	NO	Requiro Atencion Medica:	NO

Below the table, there is a section for "Observaciones" (Observations) containing the text "Antecedentes Perinatales no Definidos".

The left sidebar lists various medical modules:

- INFORME GENERAL**
- Datos Personales
- Asignar Turno
- Turnos Solicitados
- CONSULTAS MEDICAS**
- Consultas Activas
- Recetas Medicas
- HISTORIAS CLINICAS**
- Grupo Familiar
- Antecedentes Perinatales
- Habitos Toxicos
- Examen Fisico Basico
- Examen de Cabeza
- Examen de Cuello
- Piel Faneras y Tejido Subcutaneo
- Sistema Osteo Articular
- Torax y Aparato Respiratorio
- Aparato Cardiovascular
- Imagenes

Figura 7.22: Examen Único

O también muestre un listado con los exámenes realizados donde se deberá seleccionar el examen específico que se desea consultar, o registrar un nuevo examen en caso de requerirlo, como por ejemplo es el caso del examen físico ya que al paciente pueden realizarle durante su vida varios exámenes de este tipo.

The screenshot shows the S.G.C.M. interface. At the top right, it says "Quiroga, Ricardo - (Medico)". The main title is "Examenes de Aparato Cardiovascular". Below it, the patient's name is listed as "Diaz, Jose". There is a "Agregar" (Add) button. A table is displayed:

Fecha del Examen	Opciones
9 Oct. 2014	Mostrar Borrar

The left sidebar lists various medical modules:

- INFORME GENERAL**
- Datos Personales
- Asignar Turno
- Turnos Solicitados
- CONSULTAS MEDICAS**
- Consultas Activas
- Recetas Medicas
- HISTORIAS CLINICAS**
- Grupo Familiar
- Antecedentes Perinatales
- Habitos Toxicos
- Examen Fisico Basico
- Examen de Cabeza
- Examen de Cuello
- Piel Faneras y Tejido Subcutaneo
- Sistema Osteo Articular
- Torax y Aparato Respiratorio
- Aparato Cardiovascular
- Imagenes

Figura 7.23: Listado de Exámenes Aparato Cardiovascular

Sistema de Gestión de Consultorio Médico

Home | Cronograma | Pacientes | Turnos | Mensajes | Opciones

INFORME GENERAL

Datos Personales
Asignar Turno
Turnos Solicitados

CONSULTAS MEDICAS

Consultas Activas
Recetas Medicas

HISTORIAS CLINICAS

Grupo Familiar
Antecedentes Perinatales
Habitos Toxicos
Examen Fisico Basico
Examen de Cabeza
Examen de Cuello
Piel Faneras y Tejido Subcutaneo
Sistema Osteo Articular
Torax y Aparato Respiratorio
Aparato Cardiovascular
Imagenes

Detalle de Examen Aparato Cardiovascular

Díaz, José

Atras

Fecha del Examen:	9 Oct. 2014
Latidos:	412
Choques de Punta:	42
R1:	56
R2:	43
R3:	76
R4:	32
Murmullo Vesicular:	no
Chasquidos:	12
Pulso Carotídeo:	56
Pulso Humeral :	89
Pulso Radial :	54
Pulso Femoral :	67
Pulso Popliteo:	89
Pulso Tibial posterior :	64
Pulso Pedio:	56
Observaciones:	estos son solo datos introducidos al azar

Figura 7.24: Vista Muestra Datos Examen Cardiovascular

S.G.C.M.

Sistema de Gestión de Consultorio Médico

Quiroga, Ricardo - (Médico)

Home | Cronograma | Pacientes | Turnos | Mensajes | Opciones

INFORME GENERAL

Datos Personales
Asignar Turno
Turnos Solicitados

CONSULTAS MEDICAS

Consultas Activas
Recetas Medicas

HISTORIAS CLINICAS

Grupo Familiar
Antecedentes Perinatales
Habitos Toxicos
Examen Fisico Basico
Examen de Cabeza
Examen de Cuello
Piel Faneras y Tejido Subcutaneo
Sistema Osteo Articular
Torax y Aparato Respiratorio
Aparato Cardiovascular
Imagenes

Nuevo Examen Físico

Díaz, José

Presión Arterial Sistólica:	<input type="text"/>	Presión Arterial Dastólica:	<input type="text"/>
Frecuencia Respiratoria:	<input type="text"/>	Temperatura Corporal:	<input type="text"/>
Peso Promedio (kg):	<input type="text"/>	Peso (kg):	<input type="text"/>
Altura Promedio (cm):	<input type="text"/>	Altura (cm):	<input type="text"/>
Índice de Masa Corporal:	<input type="text"/>		
Impresión General:	<input type="text"/>		

Registrar | Cancelar

Figura 7.25: Nuevo Examen Físico

7.5.3. Turnos

7.5.3.1. Registrar nuevo Turno

El procedimiento relacionado al registro de un nuevo turno es similar en todos los casos solo tiene pequeñas variaciones dependiendo el rol del usuario, si se es paciente por ejemplo se seleccionara el médico al que se desea solicitar el turno, en caso de ser medico seleccionara el paciente, y si es administrativo seleccionara ambos (médico y paciente), luego el sistema le mostrara un calendario con las posibles fechas (fig:622) donde puede solicitar un turno dependiendo de los días de atención que el médico haya estipulado.

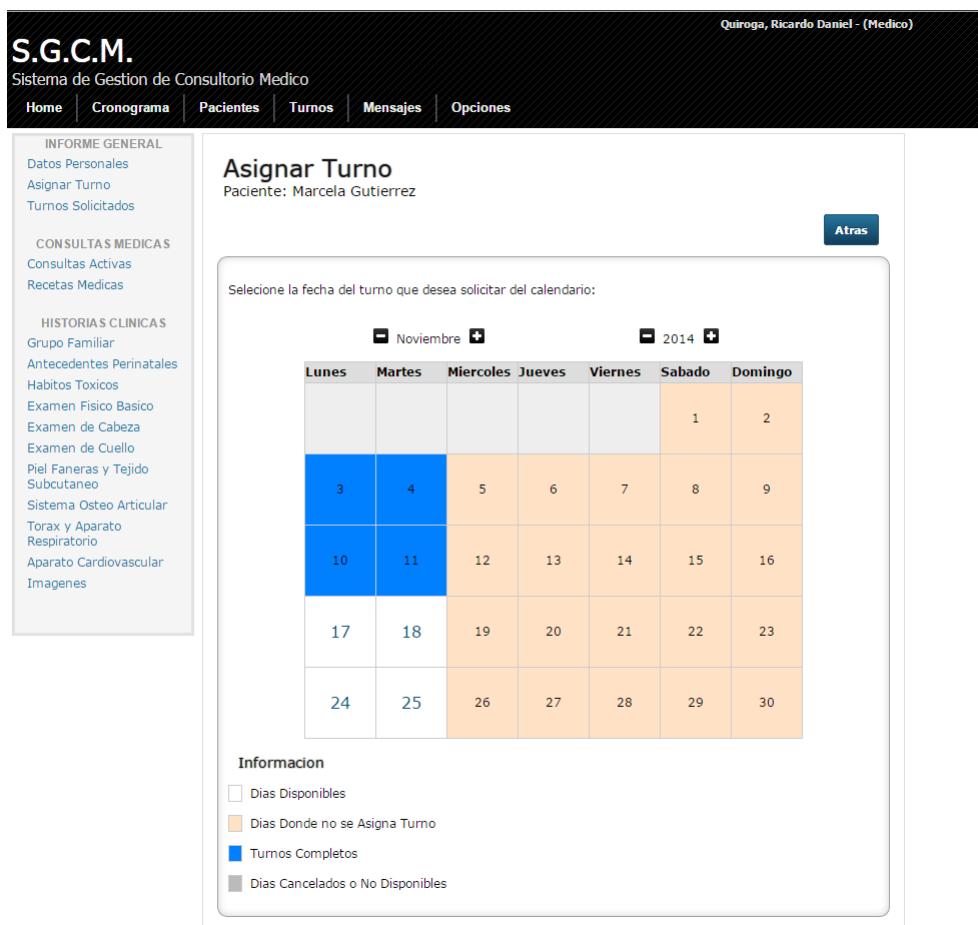


Figura 7.26: Vista para Selección de día de calendario para Asignar turno

Luego de seleccionada la fecha entre las disponibles se muestra una vista con el resumen de la información del paciente, medico, la fecha ⁵ donde se debe confirmar el registro del turno, este paso es obligatorio para evitar registrar accidentalmente un turno.

⁵El horario recién se define cuando el turno es registrado.

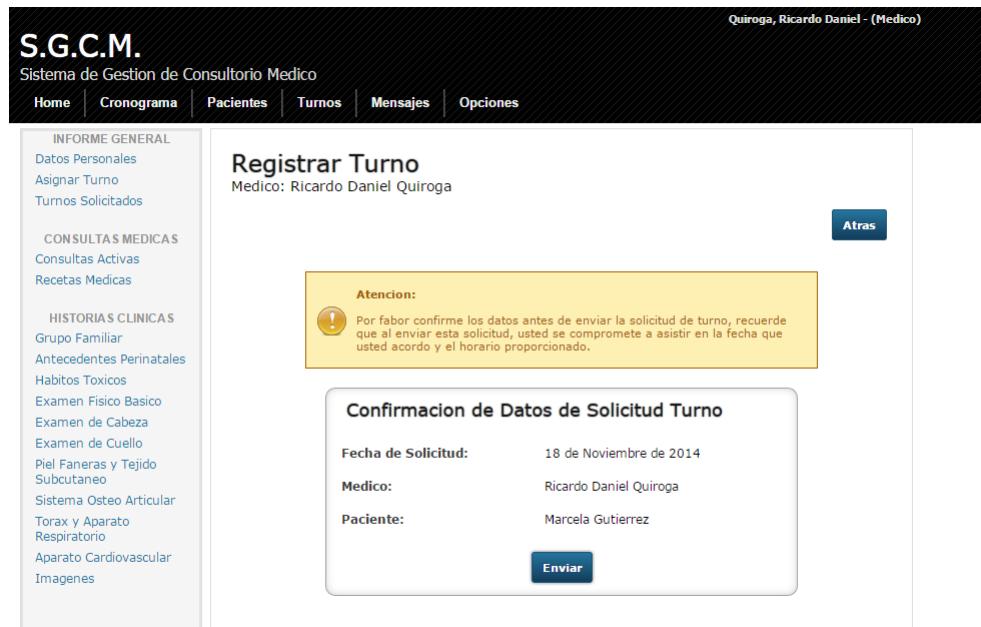


Figura 7.27: Vista de Confirmación de los datos del Turno

Después de confirmado, se mostraran los datos completo del turno, así como la opción de imprimir un comprobante.

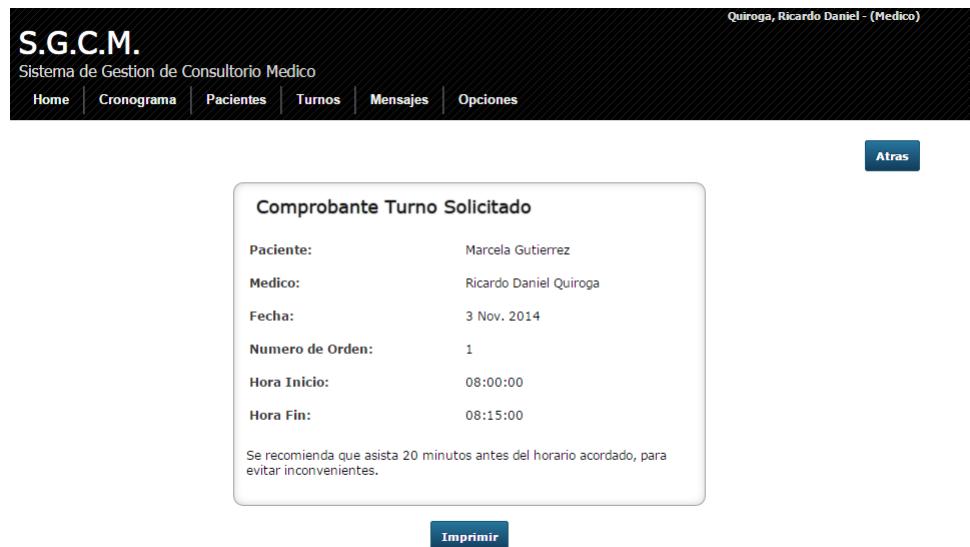


Figura 7.28: Vista muestra la informacion del Turno

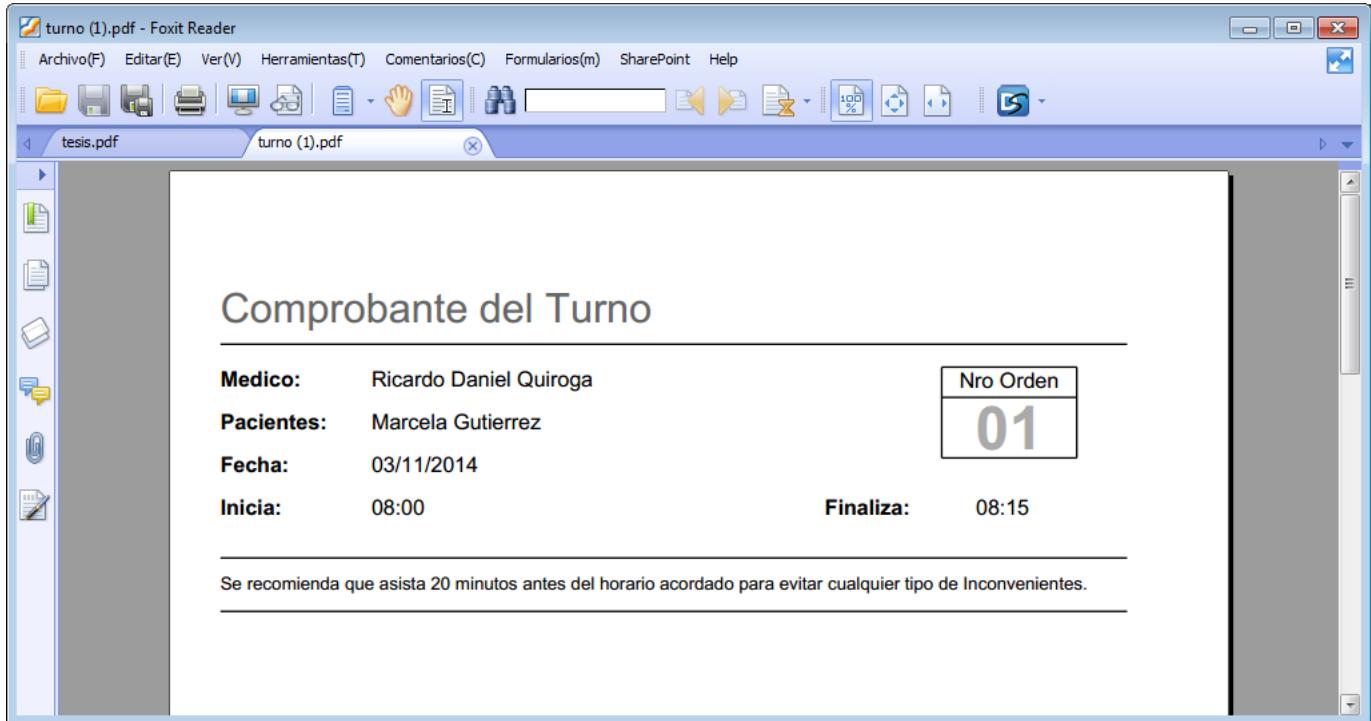


Figura 7.29: Ejemplo de Comprobante de turno

Con esto último, se concluye esta pequeña guía de uso del sistema.

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 atraves 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 funcionamiento de la mayoría de las instituciones, 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.
- 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 ahorro 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 Object Relational Model o ORM que forma parte de Django el cual es sencillo de aprender.
- Aunque no está relacionado en si 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 podes hacer preguntas y/o responderlas sobre cuestiones de programación, 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 rápida.

- Aprender a usar un sistema de control de versiones para el código fuente como GIT de mi proyecto fue de gran utilidad ya que el desarrollo de esta aplicación 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 fácil como cuando instalas LAMP, pierdes mucho tiempo intentando configurar el servidor, la documentación existente sobre la misma es muy poca y normalmente incompleta.
- En su mayoría la documentación sobre las librerías y demás herramientas se encuentra escrita en inglés, no lo consideraría en si una desventaja pero lo menciono en este apartado por mi bajo nivel en lo que respecta a lectura y comprensión de texto en inglés.

8.2. Futuras Mejoras

El sistema podría evolucionar de varias maneras, al ser un sistema diseñado mediante plantillas la principal evolución del mismo es que se podría adaptar las interfaces a los navegadores de los dispositivos móviles inteligentes mediante un diseño responsive.

Otra mejora común al sistema, seria que pueda integrarse con otros estudios como poder registrar análisis de laboratorio, odontograma, integración con el sistema vademécum para que sea más sencillo elaborar una receta médica, y la posibilidad de importar y/o exportar la historia clínica a formatos conocidos como archivos PDF para permitir ser exportado a papel.

¹Implementar un servidor de producción con Apache, Python, Django y PosgreSQL, mod_wsgi

Glosario

Glosario de términos y nomenclaturas especiales ordenados en orden alfabético que se utilizaron durante la redacción de este Informe.

A

Antecedentes Perinatales: Información referente a la atención durante el embarazo y parto.

Aplicación: Durante el informe se utilizó el nombre Aplicación o Sistema indistintamente para referirse al sistema informático desarrollado, en el cual se basa el presente informe.

Aplicación Informática: En informática, una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos. Esto lo diferencia principalmente de otros tipos de programas como los sistemas operativos, las utilidades (que realizan tareas de mantenimiento o de uso general), y los lenguajes de programación (con el cual se crean los programas informáticos).

B

Baterías Incluidas: El término hace referencia al tamaño de la biblioteca estándar (conjunto de módulos y bibliotecas con que cuenta Python por defecto) ya que la instalación básica incluye bibliotecas para casi todos tipo de tareas y las cuales pueden ser extendidas.

C

Consulta Médica: Se refiere a la cita que el paciente tiene con el especialista.

D

DER: DER es el acrónico de Diagrama de Entidad Relación, es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

Deploy: En este informe hace referencia al proceso de implementación y todas las actividades necesarias para que el sistema software esté disponible para su uso.

E

Especialista: En este informe usamos el término especialista para referirnos a los médicos.

M

MVC: son las siglas de Modelo Vista Controlador, 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 arquitectura de software 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.

N

No Pythonico: es lo contrario del término Pythonico. Ósea código fuente es ofuscado o tiene problemas de legibilidad, es un término de programación.

NoSQL: En informática, NoSQL (a veces llamado "no sólo SQL") es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales (RDBMS) en aspectos importantes, el más destacado es que no usan SQL como el principal lenguaje de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad), y habitualmente Escalan bien horizontalmente.

Por lo general, los investigadores académicos se refieren a las bases de dato SQL como bases de datos con almacenamiento estructurado, término que abarca también las bases de datos relacionales clásicas. A menudo, las bases de datos NoSQL se clasifican según su forma de almacenar los datos, y comprenden categorías como clave-valor, las implementaciones de BigTable, bases de datos documentales, y Bases de datos orientadas a grafos.

O

Ofuscado: La ofuscación se refiere a encubrir el significado de una comunicación haciéndola más confusa y complicada de interpretar. En computación, la ofuscación se refiere al acto deliberado de realizar un cambio no destructivo, ya sea en el código fuente de un programa informático o código máquina cuando el programa está en forma compilada o binaria, con el fin de que no sea fácil de entender o leer.

P

PEPs: Las PEPs o (Python Enhancement Proposals) son un conjunto de propuestas de mejoras del lenguaje de programación Python.

Prescripción Médica: (también conocida como receta médica) es el documento legal por medio del cual los médicos legalmente capacitados prescriben la medicación al paciente para su dispensación por parte del farmacéutico.

Pythonico: Es una filosofía de los programadores de Python para referirse a la legibilidad del código basándose en las especificaciones de la PEP 8.

S

SQL El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla

información de interés de bases de datos, así como hacer cambios en ellas. En el Informe se utiliza el término más que nada para referirse a los motores de base de datos que lo utilizan.

Sistema Durante el informe se utilizo el nombre Aplicación o Sistema indistintamente para referirse al sistema en cual se basa el presente informe. Aunque el termino sistema se utiliza generalmente para referirse a un objeto complejo cuyos componentes se relacionan con al menos algún otro componente, el cual puede ser material o conceptual.

Super Usuario Es una cuenta de usuario preconfigurada para facilitar la administración del Sistema.

U

UML El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado de sistemas software.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Z

Zen 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. Estos principios fueron famosamente descritos por el desarrollador de Python Tim Peters en El Zen de Python que comprende un conjunto de recomendaciones sobre buenas prácticas.

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%A1nica
- [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
- [27] TEORIA GENERAL DE SISTEMAS *Enfoque Sistematico* http://es.wikipedia.org/wiki/Teor%C3%ADa_de_sistemas