

ANEXO 1

**ESPECIFICACIÓN DE REQUISITOS DE
SOFTWARE**

Índice de Contenido

1. Introducción.....	3
2. Propósito	3
3. Definiciones y Acrónimos	3
4. Referencias	4
5. Visión General de la ERS	4

Índice de Figuras

Figura 1. Diagrama de Actores del Sistema.....	5
Figura 2. Paquete de Administración.....	6
Figura 3. Paquete de Generador de cronograma.	6
Figura 4. Paquete de Seguimiento.	7

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

1. Introducción

El presente documento es una Especificación de Requisitos de Software (ERS) del sistema REHABI que incluye la especificación de casos de uso del mismo.

2. Propósito

El propósito de este documento es definir cuáles son los requerimientos que debe tener un programa que permita agilizar el proceso de generar un cronograma de terapia física, en base a un diagnóstico que indica la periodicidad y las indicaciones de lo que se aplica en cada terapia.

El presente documento será fundamental, puesto que servirá de guía general para los involucrados en el desarrollo del sistema.

3. Definiciones y Acrónimos

Definiciones	
Definición	Concepto
Módulo	Comprende a un conjunto de funcionalidades similares del sistema desarrollado.
Paquete	Es un conjunto de casos de uso que son agrupados de acuerdo al módulo al que corresponden.
Usuario	Son personas que acceden al sistema para hacer uso de las funcionalidades desarrolladas.
Caso de Uso	Es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.
Pre condición de un caso de uso	Indican las condiciones que el sistema debe cumplir para que se pueda dar inicio a determinado caso de uso.

Post condición de un caso de uso	Indica el estado en el cual el sistema debe quedar al finalizar la ejecución del caso de uso.
----------------------------------	---

Acrónimos

ERS	Especificación de Requisitos de Software
-----	--

4. Referencias

IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998

Documento de Tesis

5. Visión General de la ERS

La visión del presente documento es brindar la explicación detallada de los requisitos que fueron estipulados como requerimientos del análisis. El documento permitirá revisar gráfica y textualmente el detalle de cada caso de uso, especificando a detalle el funcionamiento del sistema.

Este documento está estructurado en tres partes:

En la primera división o sección se incluye una introducción al documento y además se muestra una visión general de la especificación de recursos del sistema.

En la segunda sección podemos encontrar una descripción general del sistema. Con esta se podrán conocer las principales funcionalidades que tendrá, además del entorno en donde se trabajará; es decir, los datos, factores, restricciones, supuestos y dependencias que influirán en el desarrollo del sistema.

En la tercera sección se describe detalladamente cada una de las funcionalidades del sistema, en base a los requisitos que se debe satisfacer de acuerdo a las necesidades del cliente.

5.1 Descripción General

En esta sección se presenta una descripción a alto nivel del sistema. Se presentará el modelo de casos de uso (modelo que muestra la funcionalidad del sistema), las características de los usuarios, las suposiciones y dependencias del sistema.

5.2 Modelo de Casos de Uso

En esta sección se presentan los diagramas de casos de uso del sistema obtenidos durante el proceso de especificación de requisitos. Inicialmente, se detalla el catálogo de actores que interactúan con el sistema. Luego, la descripción de cada uno de los paquetes con sus respectivos diagramas de casos de uso.

Catálogo de Actores

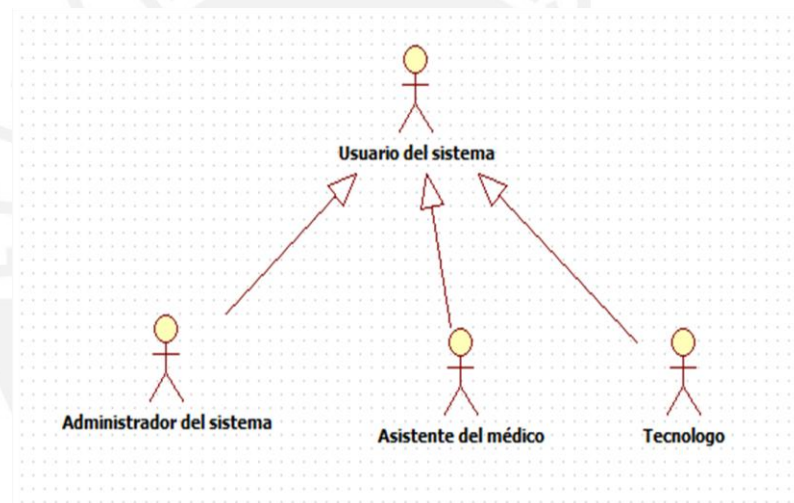


Figura 1. Diagrama de Actores del Sistema.

5.2.1 El sistema contará con los siguientes actores:

Asistente del Médico: es el encargado de registrar, modificar y dar de baja la información correspondiente al diagnóstico, indica la información técnica en cada terapia y el intervalo de tiempo de cada terapia, además del tipo de preferencia a la que pertenece.

Tecnólogo: es la persona que tendrá acceso a la información de cada paciente al cual le va a dar terapia, tiene acceso al cronograma de terapia de cada paciente. Verifica la asistencia del paciente a cada terapia.

Administrador del sistema: posee permisos de acceso a todas las pantallas y funciones del sistema y se encarga del registro de usuarios y la asignación de roles a los mismos de acuerdo al tipo de labor que realizarán. Además, es el encargado de la administración de seguridad del sistema.

5.2.2 Casos de uso por paquete

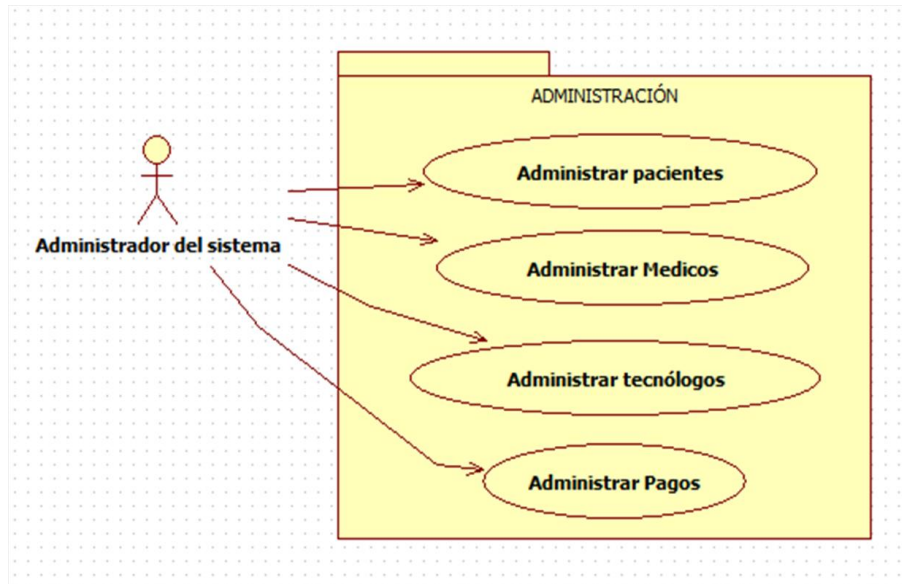


Figura 2. Paquete de Administración.

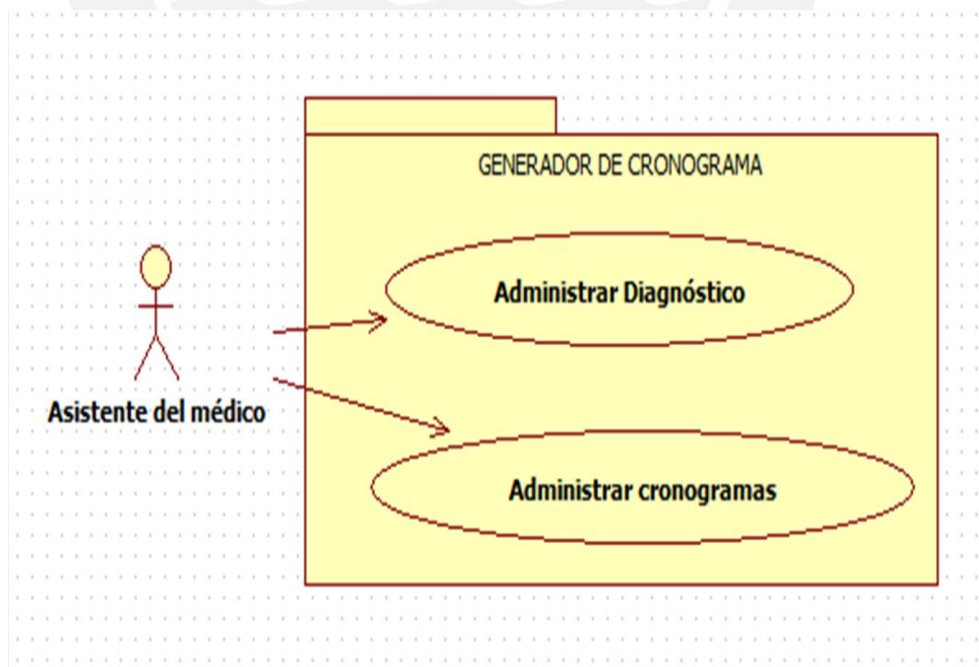


Figura 3. Paquete de Generador de cronograma.

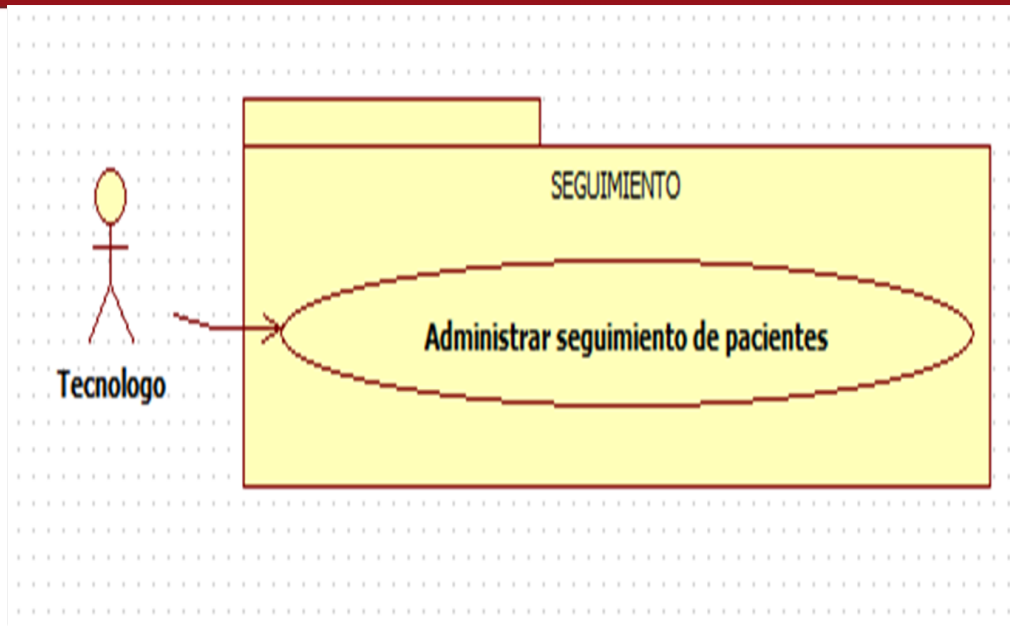


Figura 4. Paquete de Seguimiento.

5.3 Características de los Usuarios

El sistema a desarrollar está orientado al personal de un hospital. Dichos empleados están familiarizados con los procesos principales del negocio, mas no se encuentran totalmente familiarizados con la manipulación de computadores, por lo que la interfaz de usuario deberá ser sencilla de utilizar.

5.4 Suposiciones y Dependencias

Se describirá lo que significa cada una de estas ítems en el documento.

Suposiciones

Se asume que las exigencias plasmadas en el documento se mantienen incluso después de ser aprobadas por los usuarios finales. Además se asume que el sistema será utilizado sobre la plataforma Linux y contará con los recursos necesarios de memoria y almacenamiento para asegurar un funcionamiento adecuado.

Dependencias

La información de pacientes se encontrará en un archivo xml el cual será obtenido de la información histórica que actualmente se encuentre almacenada en el hospital y será procesado en el sistema mediante una interface. La consulta y modificación de los registros en la base de datos estarán sujetos a la conexión entre el equipo servidor y las terminales, las cuales se encontrarán en ambientes dentro del área de medicina física.

Especificación de Casos de Uso

En este punto se explicará la interacción de los usuarios con el sistema y todas las funcionalidades a las que está dispuesto.

6. Paquete de Administración

Id:	ADM-CU01
Nombre:	Administrar pacientes.
Actores:	Administrador del sistema.
Descripción General: Permite el registro de pacientes nuevos y su información, así como la modificación de los mismos y darlos de baja en casos especiales. También permite la carga masiva de información de pacientes de acuerdo a un formato de archivos xml establecido como datos de entrada.	
Pre-condición:	El actor debe estar registrado en el sistema antes de que inicie el caso de uso.
Flujo de eventos: Consultar pacientes El actor ingresa a la opción "Administración". El sistema muestra las opciones "Pacientes", "Médicos" y "Tecnólogos". El actor elige la opción "Pacientes". El sistema muestra la pantalla inicial de "Pacientes" con los siguientes campos como filtros de datos para realizar una búsqueda: DNI Apellido paterno Apellido materno	

<p>Nombre</p> <p>Dirección</p> <p>Correo electrónico</p> <p>Número telefónico</p> <p>Fecha de nacimiento</p> <p>Fecha de última atención en consultorio</p> <p>La última sesión de terapia.</p> <p>El actor ingresa o elige datos en los filtros que desee para realizar la búsqueda y elige la opción “Buscar”.</p> <p>El sistema realiza una búsqueda y muestra todos los pacientes que cumplan con estos filtros listados en una tabla.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	<p>Lista de pacientes consultados mostrada</p> <p>Pantalla “Pacientes”</p>
<p>Flujo de Eventos Alternativo 1</p> <p>Agregar pacientes</p> <p>El siguiente flujo de eventos alternos se da cuando el usuario elige la opción “Nuevo” en el punto 5 del Flujo Normal.</p> <p>El sistema muestra la pantalla de “Agregar pacientes” con los siguientes campos como datos requeridos a completar:</p> <p>DNI</p> <p>Apellido paterno</p> <p>Apellido materno</p> <p>Nombre</p> <p>Dirección</p> <p>Correo electrónico</p> <p>Número telefónico</p> <p>Fecha de nacimiento</p> <p>Fecha de última atención en consultorio</p> <p>La última sesión de terapia.</p> <p>El actor completa estos campos (todos son obligatorios) y elige la opción de “Guardar” el nuevo registro.</p> <p>El paciente queda registrado.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	<p>Paciente registrado.</p> <p>Pantalla “Agregar paciente”</p>

Flujo de Eventos Alternativo 2

Modificar pacientes

El siguiente flujo de eventos alternos se da cuando el usuario selecciona algún ambiente de la lista de resultados de la búsqueda en el punto 7 del Flujo Normal.

El sistema muestra la pantalla de “Detalle Paciente” con los siguientes campos Tipo de Documento, Número de Documento, Nombre, Apellido, Correo electrónico, Teléfono, Dirección del paciente seleccionado. Estos muestran la información del paciente que está guardada actualmente en el sistema y le permite modificarlos según la lógica del negocio.

El actor modifica los campos que desee y elige la opción “Guardar” los cambios.

El caso de uso termina exitosamente.

Post-condición:	Paciente modificado Pantalla “Detalle paciente”
------------------------	--

Flujo de Eventos Alternativo 3

Eliminar pacientes

El siguiente flujo de eventos alternos se da cuando el usuario selecciona algún paciente de la lista de resultados de la búsqueda en el punto 7 del Flujo Normal.

El sistema muestra la pantalla de “Detalles Paciente” con los siguientes campos Tipo de Documento, Número de Documento, Nombre, Apellido, Correo electrónico, Teléfono, Dirección del paciente seleccionado. Estos muestran la información del paciente que está guardada actualmente en el sistema y le permite modificarlos según la lógica del negocio.

El actor elige la opción “Eliminar”.

El sistema da de baja el paciente elegido por el usuario modificando su estado en el sistema.

El caso de uso termina exitosamente.

Post-condición:	Paciente eliminado. Pantalla “Detalle paciente”
------------------------	--

Flujo de Eventos Alternativo 4

Carga masiva de pacientes

El siguiente flujo de eventos alternos se da cuando el usuario escoge la opción de “Carga masiva de pacientes” en el punto 7 del Flujo Normal.

El usuario escoge la opción “Carga masiva de pacientes”.

<p>El sistema muestra una ventana de selección de archivos. Esta ventana solo mostrara carpetas y archivos en formato xml.</p> <p>El usuario deberá escoger el archivo en formato xml en el cual se encuentra la información de los pacientes provista por el hospital y luego escoge la opción “Aceptar”.</p> <p>El sistema muestra en una ventana la lista total de pacientes cargada.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición	<p>Lista de pacientes cargada exitosamente.</p> <p>Pantalla “Carga masiva de pacientes”</p>

Id:	ADM-CU02
Nombre:	Administrar Médicos.
Actores:	Administrador del sistema.
Descripción General:	
<p>Permite el registro de médicos nuevos y su información, así como la modificación de los mismos y darlos de baja en casos especiales. También permite la carga masiva de médicos de acuerdo a un formato establecido como datos de entrada.</p>	
Pre-condición:	<p>El actor debe estar registrado en el sistema antes de que inicie el caso de uso.</p>
Flujo de Eventos Alternativo 1	
Agregar médicos	
<p>El siguiente flujo de eventos alternos se da cuando el usuario elije la opción “Nuevo” en el punto 5 del Flujo Normal.</p> <p>El sistema muestra la pantalla de “Agregar médicos” con los siguientes campos como datos requeridos a completar:</p> <p>DNI</p> <p>Apellido paterno</p> <p>Apellido materno</p> <p>Nombre</p> <p>Dirección</p> <p>Número telefónico</p> <p>Fecha de nacimiento</p>	

Correo electrónico

El actor completa estos campos (todos son obligatorios) y elije la opción de “Guardar” el nuevo registro.

El médico queda registrado.

El caso de uso termina exitosamente.

Post-condición:	Médico registrado. Pantalla “Agregar médico”
------------------------	---

Flujo de Eventos Alternativo 2

Modificar médicos

El siguiente flujo de eventos alternos se da cuando el usuario selecciona algún ambiente de la lista de resultados de la búsqueda en el punto 7 del Flujo Normal.

El sistema muestra la pantalla de “Detalle Médico” con los siguientes campos Tipo de Documento, Número de Documento, Nombre, Apellido, Correo electrónico, Teléfono, Dirección del médico seleccionado. Estos muestran la información del médico que está guardada actualmente en el sistema y le permite modificarlos según la lógica del negocio.

El actor modifica los campos que desee y elije la opción “Guardar” los cambios.

El caso de uso termina exitosamente.

Post-condición:	Médico modificado Pantalla “Detalle médico”
------------------------	--

Flujo de Eventos Alternativo 3

Eliminar médicos

El siguiente flujo de eventos alternos se da cuando el usuario selecciona algún médico de la lista de resultados de la búsqueda en el punto 7 del Flujo Normal.

El sistema muestra la pantalla de “Detalles Médico” con los siguientes campos Tipo de Documento, Número de Documento, Nombre, Apellido, Correo electrónico, Teléfono, Dirección del médico seleccionado. Estos muestran la información del médico que está guardada actualmente en el sistema y le permite modificarlos según la lógica del negocio.

El actor elije la opción “Eliminar”.

El sistema da de baja el médico elegido por el usuario modificando su estado en el sistema.

El caso de uso termina exitosamente.

Post-condición:	Médico eliminado. Pantalla “Detalle médico”
Flujo de Eventos Alternativo <p>El actor hace clic en el botón “Guardar”.</p> <p>Si el actor no completó todos los campos obligatorios entonces el sistema impedirá que se registre al nuevo médico y le solicitará que complete los campos que faltan para que pueda completarse el registro. Si el actor hace modificaciones no permitidas, el sistema mostrará un mensaje de error cuando el actor intente finalizar la edición.</p> <p>El actor hace clic en “Cancelar”.</p> <p>Si el actor selecciona la opción “Cancelar” dentro del registro de un nuevo médico, el sistema cancelará el registro y lo llevará de vuelta a la página principal de médicos.</p>	
Post-condición:	No se crea o actualiza el registro.
Flujo de eventos: Consultar médicos <p>El actor ingresa a la opción “Administración”.</p> <p>El sistema muestra las opciones “Pacientes”, “Médicos” y “Tecnólogos”.</p> <p>El actor elige la opción “Médicos”.</p> <p>El sistema muestra la pantalla inicial de “Médicos” con los siguientes campos como filtros de datos para realizar una búsqueda: Nombre Apellido Tipo de documento Número de documento</p> <p>El actor ingresa o elige datos en los filtros que desee para realizar la búsqueda y elige la opción “Buscar”.</p> <p>El sistema realiza una búsqueda y muestra todos los médicos que cumplan con estos filtros listados en una tabla.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	Lista de médicos consultados mostrada Pantalla “Médicos”
Flujo de Eventos Alternativo Carga masiva de médicos <p>El siguiente flujo de eventos alternos se da cuando el usuario escoge la opción</p>	

<p>de “Carga masiva de médicos” en el punto 7 del Flujo Normal.</p> <p>El usuario escoge la opción “Carga masiva de médicos”.</p> <p>El sistema muestra una ventana de selección de archivos. Esta ventana solo mostrara carpetas y archivos en formato xml.</p> <p>El usuario deberá escoger el archivo en formato xml en el cual se encuentra la información de los pacientes provista por el hospital y luego escoge la opción “Aceptar”.</p> <p>El sistema muestra en una ventana la lista total de médicos cargada.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	<p>Lista de médicos cargada exitosamente.</p> <p>Pantalla “Carga masiva de médicos”</p>

Id:	ADM-CU03
Nombre:	Administrar tecnólogos.
Actores:	Administrador del sistema.
Descripción General:	
<p>Permite el registro de tecnólogos nuevos y su información, así como la modificación de los mismos y darlos de baja en casos especiales. También permite la carga masiva de información de tecnólogos de acuerdo a un formato establecido como datos de entrada.</p>	
Pre-condición:	El actor debe estar registrado en el sistema antes de que inicie el caso de uso.
Flujo de eventos:	
Consultar tecnólogos	
<p>El actor ingresa a la opción “Administración”.</p> <p>El sistema muestra las opciones “Pacientes”, “Médicos” y “Tecnólogos”.</p> <p>El actor elije la opción “Tecnólogos”.</p> <p>El sistema muestra la pantalla inicial de “Tecnólogos” con los siguientes campos como filtros de datos para realizar una búsqueda:</p> <p>Nombre</p> <p>Apellido</p> <p>DNI</p> <p>Apellido paterno</p> <p>Apellido materno</p> <p>Nombre</p>	

Dirección Correo electrónico Número telefónico Fecha de nacimiento Preferencia El actor ingresa o elige datos en los filtros que desee para realizar la búsqueda y elige la opción “Buscar”. El sistema realiza una búsqueda y muestra todos los tecnólogos que cumplan con estos filtros listados en una tabla. El caso de uso termina exitosamente.	
Post-condición:	Lista de tecnólogos consultados mostrada Pantalla “Tecnólogos”
Flujo de Eventos Alternativo Carga masiva de tecnólogos El siguiente flujo de eventos alternos se da cuando el usuario escoge la opción de “Carga masiva de tecnólogos” en el punto 7 del Flujo Normal. El usuario escoge la opción “Carga masiva de tecnólogos”. El sistema muestra una ventana de selección de archivos. Esta ventana solo mostrara carpetas y archivos en formato xml. El usuario deberá escoger el archivo en formato xml en el cual se encuentra la información de los tecnólogos provista por el hospital y luego escoge la opción “Aceptar”. El sistema muestra en una ventana la lista total de tecnólogos cargada. El caso de uso termina exitosamente.	
Post-condición:	Lista de tecnólogos cargada exitosamente. Pantalla “Carga masiva de tecnólogos”

Id:	GCR-CU04
Nombre:	Administrar Historia clínica.
Actores:	Asistente del médico.
Descripción General:	
Permite el registro de nuevos diagnóstico y su información, así como la modificación de los mismos.	
Pre-condición:	El actor debe estar registrados en el sistema antes de que inicie el caso de uso.

Flujo de eventos:**Consultar Diagnóstico**

El actor ingresa a la opción "Generador de cronograma".

El sistema muestra un submenú.

El actor elige la opción "Diagnóstico".

El sistema muestra la pantalla inicial de "Diagnóstico" con los campos código, descripción, indicaciones en cada terapia de un día, la cantidad de días entre cada terapia puesto de manera ordenada y la preferencia.

El actor ingresa o elige datos en los filtros que desee para realizar la búsqueda y elige la opción "Buscar".

El sistema realiza una búsqueda y muestra todos los Diagnóstico que cumplan con estos filtros listados en una tabla.

El caso de uso termina exitosamente.

Post-condición:

Lista de Diagnóstico consultados mostrada
 Pantalla "Diagnóstico"

Flujo de Eventos Alternativo 1**Agregar Diagnóstico**

El siguiente flujo de eventos alternos se da cuando el usuario elige la opción "Nuevo" en el punto 5 del Flujo Normal.

El sistema muestra la pantalla de "Agregar Diagnóstico" con los campos código, fecha, descripción del diagnóstico, indicaciones en cada día de terapia, la cantidad de días entre cada terapia puesto de manera ordenada y la preferencia como datos requeridos a completar. También tendrá Código CIE10 del diagnóstico, reseña de la enfermedad, examen clínico, tratamiento, evolución, recomendaciones.

El actor completa estos campos (todos son obligatorios) y elige la opción de "Guardar" el nuevo registro.

El actor elige la opción "Generador de cronograma".

El sistema muestra el diagnóstico con los campos código, descripción, las fechas de cada terapia y las indicaciones para cada una.

El Diagnóstico junto con el cronograma de terapias queda registrado.

El caso de uso termina exitosamente.

Post-condición:

Diagnóstico registrado.
 Pantalla "Agregar Diagnóstico"

Flujo de Eventos Alternativo 2**Modificar Diagnóstico**

El siguiente flujo de eventos alternos se da cuando el usuario selecciona algún diagnóstico de la lista de resultados de la búsqueda en el punto 7 del Flujo Normal.

El sistema muestra la pantalla de “Detalle de Diagnóstico” con los siguientes campos código, fecha, descripción del diagnóstico, indicaciones en cada día de terapia, la cantidad de días entre cada terapia puesto de manera ordenada y la preferencia. Estos muestran la información del diagnóstico que está guardada actualmente en el sistema y le permite modificarlo.

El actor elije la opción “Generador de cronograma”.

El sistema muestra el diagnóstico con los campos código, descripción, las fechas de cada terapia y las indicaciones para cada una.

El actor modifica los campos que desee y elije la opción “Guardar” los cambios.

El sistema muestra un mensaje de confirmación de la acción realizada por el actor.

El caso de uso termina exitosamente.

Post-condición:	Diagnóstico modificado Pantalla “Detalle de Diagnóstico”
------------------------	---

Id:	GCR-CU05
Nombre:	Administrar cronogramas.
Actores:	Asistente del médico.
Descripción General:	
Permite la actualización de cronogramas, por parte de la inasistencia de pacientes o de los tecnólogos.	
Pre-condición:	El actor debe estar registrados en el sistema antes de que inicie el caso de uso.
Flujo de eventos:	
Actualizar	
El actor ingresa a la opción “Generador de cronograma”.	
El sistema muestra un submenú.	
El actor elije la opción “Actualización”.	
El sistema muestra la pantalla para cargar archivos planos con la información	

<p>necesaria como datos de entrada para eliminar las citas de los pacientes con 3 inasistencias.</p> <p>El sistema actualiza la disponibilidad de cada tecnólogo y si la carga de trabajo es demasiado desigual se vuelve a planificar todos los cronogramas de terapia.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	Se actualiza la disponibilidad de cada tecnólogo.
<p>Flujo de Eventos Alternativo 1</p> <p>Actualizar por falta de personal</p> <p>El actor ingresa a la opción “Generador de cronograma”.</p> <p>El sistema muestra un submenú.</p> <p>El actor elige la opción “Actualizar por inasistencia del personal”.</p> <p>El sistema muestra la pantalla inicial de “Actualización de cronogramas”.</p> <p>El actor carga los archivos de texto planos con la información necesaria como datos de entrada para eliminar a los tecnólogos asignados en cada cronograma de terapia ya elaborado, debido a que faltarán por un periodo de tiempo.</p> <p>El sistema actualiza la asignación de los tecnólogos a cada cronograma de terapia, considerando un balance de carga casi equitativo para los tecnólogos disponibles durante ese periodo de tiempo.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición:	Se actualiza los cronogramas que lo requieran.

Id:	GCR-CU06
Nombre:	Administrar seguimiento de pacientes.
Actores:	Tecnólogo.
<p>Descripción General:</p> <p>Permite registrar la asistencia a cada paciente a sus respectivas terapias dadas en el cronograma, además, de registrar observaciones de cada terapia.</p>	
Pre condición:	El actor debe estar registrados en el sistema antes de que inicie el caso de uso.

Flujo de Eventos**Consultar Seguimiento**

El actor ingresa a la opción "Seguimiento".

El sistema muestra las opciones "Asistencia" y "Cronograma".

El actor elige la opción "Asistencia".

El sistema muestra la pantalla inicial de "Asistencia" con los siguientes campos como filtros de datos para realizar una búsqueda:

Fecha de inicio

Fecha de fin

El actor ingresa o elige datos en los filtros que desee para realizar la búsqueda y elige la opción "Buscar".

El sistema realiza una búsqueda y muestra todos los pacientes que cumplan con estos filtros listados en una tabla.

El caso de uso termina exitosamente.

Post-

Lista de pacientes consultados mostrada

condición:

Pantalla "Pacientes"

Flujo de Eventos Alternativo 1**Agregar seguimiento**

El siguiente flujo de eventos alternos se da cuando el usuario escoge la opción de "Cronograma" en el punto 2 del Flujo Normal.

El usuario escoge la opción "Cronograma".

El sistema muestra la pantalla inicial de "Pacientes" de todos los pacientes del actor con los siguientes campos como filtros de datos para realizar una búsqueda:

DNI

Apellido paterno

Apellido materno

Nombre

Dirección

Correo electrónico

Número telefónico

Fecha de nacimiento

Fecha de inicio

Fecha de fin

El actor ingresa o elige datos en los filtros que desee para realizar la

<p>búsqueda y elige la opción “Buscar”.</p> <p>El sistema realiza una búsqueda y muestra todos los pacientes que cumplan con estos filtros listados en una tabla.</p> <p>El actor selecciona un paciente del formulario mostrado.</p> <p>El sistema muestra la terapia con la fecha y las indicaciones para esa terapia.</p> <p>El actor registra la asistencia del paciente para esa terapia y las observaciones de la terapia.</p> <p>El caso de uso termina exitosamente.</p> <p>El caso de uso termina exitosamente.</p>	
Post-condición	Registrar asistencia y observaciones de cada terapia del paciente. Pantalla “Pacientes”.

Id:	SEG-CU01
Nombre:	Administrar Roles.
Actores:	Administrador del sistema.
Descripción General:	
<p>El presente caso de uso permite administrar los accesos y permisos de cada usuario mediante un esquema de roles por niveles, los cuales irán acordes al grado de confidencialidad para cada cliente, según su rol. Con ello se permite filtrar los accesos no autorizados.</p>	
Pre-condición:	<p>Para que se puedan administrar los roles se debe de haber ingresado con el usuario administrador del sistema.</p> <p>En el caso que se requiera editar la información de un rol, este debe estar previamente registrado.</p>
Flujo de eventos:	
<p>El actor selecciona la opción “Seguridad” que se encuentra en el menú principal.</p> <p>El actor selecciona la opción “Roles”.</p> <p>El sistema muestra una lista de Roles, con los siguientes botones “Modificar” y “Eliminar” así como un botón de “Nuevo Rol” en la parte superior de la Lista.</p> <p>Si el actor selecciona la Opción Nuevo Rol:</p> <p>El sistema muestra una lista de opciones que podrán ser asignadas al Rol,</p>	

<p>divididas por niveles: Menú Principal, Submenú, Botones.</p> <p>El actor selecciona los permisos que ha de asignar al nuevo rol siguiendo un esquema de cascada Menú Principal-> SubMenú-> Botones y escoge la opción “Guardar”.</p> <p>Si el actor selecciona la Opción Modificar Rol:</p> <p>El sistema muestra una lista de opciones que podrán ser asignadas al Rol, divididas por niveles: Menú Principal, Submenú, Botones.</p> <p>El actor selecciona los permisos que ha de asignar o desasignar al nuevo rol siguiendo un esquema de cascada Menú Principal-> SubMenú-> Botones y escoge la opción “Guardar”.</p> <p>Si el actor selecciona la Opción Eliminar Rol:</p> <p>El sistema muestra un mensaje de confirmación sobre la acción tomada.</p> <p>El actor Confirma la eliminación.</p> <p>El sistema actualiza la información del nuevo rol y lo muestra en la lista.</p>	
Post-condición:	El nuevo rol se encuentra correctamente registrado.
<p>Flujo de Eventos Alternativo</p> <p>El actor hace clic en “Cancelar”</p> <p>Si el actor selecciona la opción “Cancelar” dentro se regresa al listado de los roles.</p>	
Post-condición:	No se realiza ninguna modificación a los roles de usuario.
Requerimientos:	Administrar los accesos al sistema de acuerdo a los roles asignados a los usuarios. Registrar, modificar y eliminar roles.

Id:	SEG-CU02
Nombre:	Modificar Contraseña
Actores:	Administrador del Sistema. Medico Tecnólogo Paciente

Descripción General:	
El presente caso de uso permite el cambio de la contraseña de un usuario en particular la cual seguirá un criterio de encriptación en todo momento con el fin de mantener el sistema libre de accesos no autorizados.	
Pre-condición:	Para que se pueda modificar la contraseña se debe de haber ingresado como usuario del sistema.
Flujo de eventos:	
El actor selecciona la opción “Cuenta” que se encuentra en el menú principal.	
El actor selecciona la opción “Modificar Contraseña”.	
El sistema muestra un campo para ingresar la contraseña actual y otro para la contraseña nueva.	
El actor ingresa su contraseña actual y su contraseña nueva.	
El sistema verifica que la contraseña nueva vaya acorde a los factores: longitud de contraseña, cantidad de caracteres especiales, cantidad de caracteres alfanuméricos, contraseña no usada antes.	
El sistema actualiza la información de la contraseña.	
Post-condición:	La nueva contraseña se encuentra vigente.
Flujo de Eventos Alternativo	
El actor hace clic en “Cancelar”	
Si el actor selecciona la opción “Cancelar” se regresa al menú de “Cuenta”.	
El actor ingresa una contraseña que no cumple con los factores necesarios : longitud de contraseña, cantidad de caracteres especiales, cantidad de caracteres alfanuméricos, contraseña no usada antes.	
El sistema muestra información de cuál o cuáles son los factores que no se han completado.	
Post-condición:	La contraseña no es modificada.
Requerimientos:	<p>Modificar las contraseñas periódicamente o a solicitud del usuario evitando el uso de contraseñas repetidas en base a las últimas en su historial de contraseña.</p> <p>Las contraseñas se almacenarán bajo determinado criterio de encriptación</p> <p>Administrar de forma configurable reglas para la formación de contraseñas</p>

Id:	SEG-CU03
Nombre:	Recuperar Contraseña.
Actores:	Administrador del Sistema. Médico Tecnólogo Paciente
Descripción General: El presente caso de uso permite el restablecimiento de una nueva contraseña para un usuario, usando un medio de autenticación tal como el correo electrónico del cliente, en caso de olvido. La Transmisión de la nueva contraseña se mantendrá encriptado, con el fin de mantener el sistema libre de accesos no autorizados.	
Pre-condición:	En el caso que se requiera editar la contraseña de un usuario, este debe estar previamente registrado.
Flujo de eventos: El actor selecciona la opción “Olvide mi contraseña” que se encuentra en la pantalla de Acceso. El sistema muestra una pantalla que presenta 2 Opciones: Restablecer por Correo Electrónico y restablecer con información personal. Si el actor selecciona la Opción de Restablecer por correo electrónico: El sistema muestra una pantalla con una casilla para colocar la dirección de correo electrónico asociada a la cuenta. El actor ingresa su correo electrónico. El sistema manda un correo de verificación al correo electrónico. El actor verifica el correo electrónico. El sistema modifica la contraseña por una generada aleatoriamente siguiendo los siguientes factores: longitud de contraseña, cantidad de caracteres especiales, cantidad de caracteres alfanuméricos, contraseña no usada antes y la envía por correo electrónico al usuario. Si el actor selecciona la “Opción de Restablecer” con información personal: El sistema muestra un formulario con una lista de preguntas para resolver y una casilla para escribir la nueva contraseña.	

<p>El actor responde las preguntas postuladas.</p> <p>El sistema valida que la información ingresada sea correcta.</p> <p>El sistema El sistema verifica que la contraseña nueva vaya acorde a los factores: longitud de contraseña, cantidad de caracteres especiales, cantidad de caracteres alfanuméricos, contraseña no usada antes y la guarda.</p> <p>El sistema actualiza la información de la contraseña.</p>	
Post-condición:	La nueva contraseña se encuentra vigente.

Id:	SEG-CU04
Nombre:	Administrar Usuarios.
Actores:	Administrador del sistema.
Descripción General:	
<p>El presente caso de uso permite la creación, modificación y eliminación de los datos de un usuario o un grupo de usuarios del sistema.</p>	
Pre-condición:	<p>Para que se puedan administrar los usuarios se debe de haber ingresado con el usuario administrador del sistema.</p> <p>En el caso que se requiera editar la información de un usuario, este debe estar previamente registrado.</p>
Flujo de eventos:	
<p>El actor selecciona la opción “Administración” que se encuentra en el menú principal.</p> <p>El actor selecciona la opción “Usuario”.</p> <p>El sistema muestra una lista de Usuarios del sistema, con los siguientes botones “Modificar” y “Eliminar” así como un botón de “Nuevo Usuario” en la parte superior de la Lista.</p> <p>Si el actor selecciona la Opción Nuevo Usuario:</p> <p>El sistema muestra los siguientes campos a rellenar: Un código autogenerated, Nombre del usuario, Apellido del Usuario, Una contraseña Autogenerada, Teléfono fijo, Correo electrónico, Dirección domiciliaria y Rol de acceso.</p> <p>El actor rellena la información de los campos mencionados en el punto anterior.</p>	

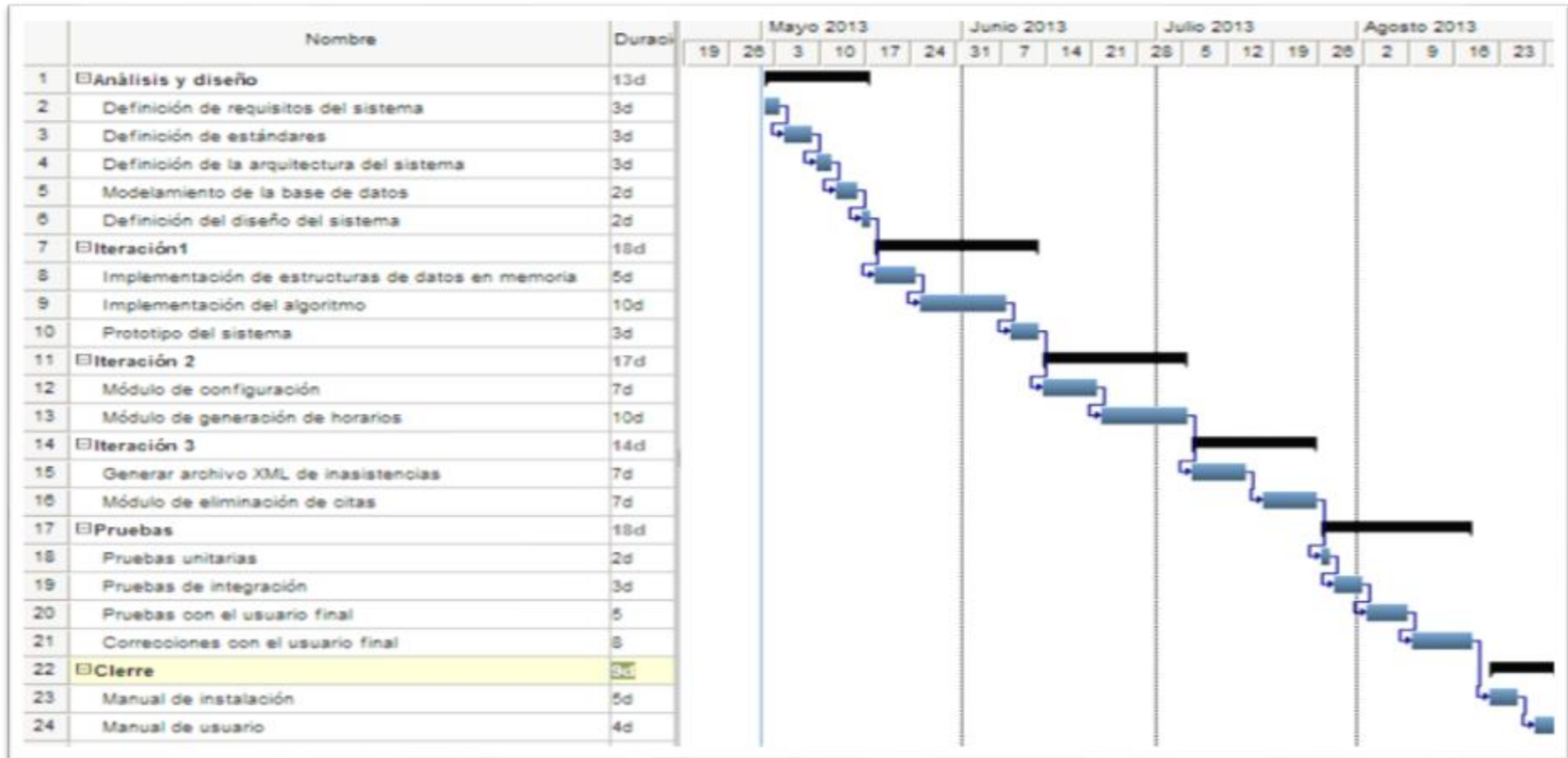
<p>El sistema muestra una pantalla de confirmación de la información ingresada.</p> <p>El cliente confirma la información ingresada.</p> <p>El sistema guarda la información del nuevo usuario y envía un correo electrónico con dicha información detallada junto a un login autogenerado.</p> <p>Si el actor selecciona la Opción Modificar Usuario:</p> <p>El sistema muestra los siguientes modificables del usuario a modificar: Nombre de usuario, Apellido del Usuario, Una contraseña Autogenerada, Teléfono fijo, Correo electrónico, Dirección domiciliaria y Rol de acceso.</p> <p>El actor modifica la información necesaria.</p> <p>El sistema muestra una pantalla de confirmación de la información modificada.</p> <p>El actor confirma la información modificada.</p> <p>El sistema actualiza la información del usuario.</p> <p>Si el actor selecciona la Opción Eliminar Usuarios:</p> <p>El sistema muestra un mensaje de confirmación sobre la acción tomada.</p> <p>El usuario Confirma la eliminación.</p> <p>El sistema actualiza la lista de Usuarios con el usuario eliminado.</p>	
Post-condición:	El Usuario fue creado, modificado o eliminado.
Flujo de Eventos Alternativo	
<p>El actor hace clic en "Cancelar".</p> <p>Si el actor selecciona la opción "Cancelar" se regresa al listado de Usuarios.</p>	
Post-condición:	No se realiza ningún cambio en los usuarios.
Requerimientos:	<p>Registrar, modificar y eliminar usuarios.</p> <p>Administrar de forma configurable reglas para la formación de contraseñas.</p>



ANEXO 2

DIAGRAMA DE GANTT DEL PROYECTO

DIAGRAMA DE GANTT DEL PROYECTO





ANEXO 3

**DESCRIPCIÓN DE LAS CLASES PARA EL
DESARROLLO DEL SOFTWARE**

Índice de Figuras

Figura 1. Clase Paciente.....	4
Figura 2. Declaración de constante TIPO_DOC.....	4
Figura 3. Script de creación de la tabla Paciente.....	7
Figura 4. Clase Personal.	8
Figura 5. Script de creación de la tabla Personal.....	8
Figura 6. Clase Turno.	9
Figura 7. Script de creación de la tabla Turno.	9
Figura 8. Clase Tarea.	10
Figura 9. Script de creación de la tabla Tarea.	10
Figura 10. Clase Conflicto.....	11
Figura 11. Generación de tabla Conflicto.....	11
Figura 12: Clase CIE10.....	11
Figura 13. Script de creación de la tabla CIE10.....	12
Figura 14. Clase Acto_Medico.....	13
Figura 15. Script de creación de la tabla Acto_Medico.....	14
Figura 16. Clase Medicamento.....	14
Figura 17. Script de creación de la tabla Medicamento.....	15
Figura 18. Clase Medicamento_por_Acto_Medico.	15
Figura 19. Script de creación de la tabla Medicamento_por_Acto_Medico.	16
Figura 20. Clase Preferencia.	16
Figura 21. Script de creación de la tabla Preferencia.	17
Figura 22. Clase Requisito.....	17
Figura 23. Script de creación de la tabla Requisito.....	18
Figura 24. Clase Capacidad.	18
Figura 25. Script de creación de la tabla Capacidad.	19
Figura 26. Clase Disponibilidad.	19

Figura 27. Script de creación de la tabla Disponibilidad. 20

Figura 28. Clase Grupo_Turno. 20

Figura 29. Script de creación de la tabla Grupo_Turno. 21

Figura 30. Clase Carga_Minima. 22

Figura 31. Script de creación de la tabla Carga_Minima. 22

Figura 32. Clase Carga_Maxima. 23

Figura 33. Script de creación de la tabla Carga_Maxima. 23

Figura 34. Clase Pago. 24

Figura 35. Script de creación de la tabla Pago. 25



▪ Clase Paciente

En la figura 1, se presenta el código necesario para la creación de la tabla de pacientes:

```
class Paciente(models.Model):
    """
        la pk es creada automáticamente por Django y denominada "id"
    """
    tipo_doc=models.IntegerField('Tipo documento', choices=TIPO_DOC, default=0)
    numero_doc=models.CharField('Número documento', max_length=11)
    ap_paterno=models.CharField('Apellido paterno', max_length=100)
    ap_materno=models.CharField('Apellido materno', max_length=100)
    nombres=models.CharField(max_length= 100)
    sexo=models.IntegerField(choices=SEXO, default=0)
    fecha_nac=models.DateField('Fecha nacimiento', null=True)
    lugar_nac=models.CharField('Lugar nacimiento', max_length=100, blank=True)
    procedencia=models.CharField('Procedencia', max_length=100, blank=True)
    estado_civil=models.IntegerField(choices=ESTADO_CIVIL, default=0) #0=soltero,
1=casado, 2=viudo, 3=divorciado
    centro_educativo=models.CharField(max_length=200, blank=True)
    grado_instruccion=models.IntegerField('Grado instrucción',
choices=GRADO_INSTRUCCION, default=4)
    nombre_padre=models.CharField(max_length= 100, blank=True)
    nombre_madre=models.CharField(max_length= 100, blank=True)
    tipo_seguro=models.IntegerField(choices=TIPO_SEGURO, default=0) #0=essalud
1=sis 2=soat
    numero_seguro=models.CharField('Número seguro', max_length=20, blank=True)
    ocupacion=models.CharField('Ocupación', max_length=30, blank=True)
    profesion=models.CharField('Profesión', max_length=30, blank=True)
    religion=models.IntegerField('Religión', choices=RELIGION, default=0) #0=catolico
1=otra
    fecha_ult_at_medica=models.DateField('Fecha última consulta médica',
default=timezone.now())
    fecha_ult_at_terapia=models.DateField('Fecha última terapia', null=True)
    edad=models.IntegerField(null= True)
    domicilio=models.CharField(max_length= 100, blank=True)
    telefono=models.CharField('Teléfono', max_length=50, blank=True)
```

```

email=models.EmailField(blank=True)

def _get_full_name(self):
    return '%s %s, %s' % (self.ap_paterno, self.ap_materno, self.nombres)

nombre_completo=property(_get_full_name)

def __unicode__(self):
    return self.nombre_completo

```

Figura 1. Clase Paciente.

El diseño de los campos definidos en esta clase, utiliza como punto de partida la información que es requerida en el formulario impreso de Historia Clínica que es utilizado en el Hospital San José.

Esta clase recibe como parámetro el objeto `models.Model` del framework, que es utilizado para definir los tipos de campos de la tabla. La clave principal o “pk”, cuando no es creada explícitamente, es generada en forma automática por Django, con el nombre “id”.

Algunos de los tipos de campo que son utilizados son:

- ***IntegerField***. Esta propiedad crea campos tipo Entero, de 11 posiciones, en la base de datos. En nuestro diseño, hemos definido el campo “tipo_doc”; es decir, tipo de documento, como una variable de este tipo a la que además se le ha especificado la etiqueta “Tipo documento” que será utilizada cada vez que se presente el campo en las interacciones con el usuario (pantallas, reportes). Para este campo también se ha especificado el parámetro *choices*, que limitará los valores que puede tomar la variable a únicamente los valores contenidos en la constante TIPO_DOC, declarada al inicio del módulo, y que contiene los siguientes valores como se muestra en la figura 2:

```

TIPO_DOC=(
    (0, 'DNI'),
    (1, 'RUC'),
    (2, 'Carnet de extranjería'),
    (3, 'Otro'),
)

```

Figura 2. Declaración de constante TIPO_DOC.

Otros campos de la clase Paciente que son de tipo IntegerField son “sexo” (asociado a la constante SEXO), “estado_civil” cuyos valores posibles están definidos en la constante ESTADO_CIVIL, “grado de instrucción” (GRADO_INSTRUCCIÓN), “tipo_seguro” (TIPO_SEGURO) y “religión” (RELIGION).

- **CharField.** Esta propiedad genera campos tipo “varchar” en la base de datos, con el tamaño o ancho indicado en el parámetro *max_length*. Este tipo de campo permite registrar valores alfanuméricos que describen a nuestro objeto. Para esta clase, los campos CharField que se han definido son: “numero_doc”, es decir número de documento del tipo que se eligió, “ap_paterno”, apellido paterno del paciente, con un ancho máximo de 100 posiciones, “ap_materno”, que registra el apellido materno de cada individuo y que tiene un ancho máximo de 100 posiciones, “nombres”, que como su nombre indica servirá para registrar los nombres del paciente, “procedencia”, en el que se coloca el nombre del distrito del cual proviene el paciente. Los campos “centro_educativo”, que aplica cuando el paciente es un escolar, “nombre_padre” y “nombre_madre” con datos de los padres del paciente, sobre todo menores de edad. En nuestro diseño, hemos definido el campo “tipo_doc”, es decir, tipo de documento, como una variable de este, “numero_seguro” que sirve para registrar el número o código del seguro (ESSALUD, SOAT, SIS) según lo indicado en el “tipo_seguro”. Son también campos del tipo Charfield, la “ocupación”, “profesión”, “domicilio” y “teléfono”. Como se puede apreciar en la figura 4.2, un parámetro de muy importante de esta propiedad, y que servirá para definir si la entrada de un campo es obligatoria o no es, se denomina *blank=True*, y cuando aparece significa que ese campo puede quedar en blanco. Si no aparece, Django asume por defecto que ese campo es obligatorio, y esto es tomado en cuenta en la generación de los formularios de mantenimiento de esta clase.
- Las fechas se indican en el diseño como campos **DateField**, y esta indicación permite además que en el momento de generación de los formularios web, se genera una clase adecuada para su validación y el

ingreso de formularios. Son este tipo de campos la `fecha_nac` (fecha de nacimiento), `fecha_ult_at_medica` (fecha de última atención médica o consulta), `fecha_ult_at_terapia` (fecha de última terapia).

- El último de los tipos de campo mostrados es el de tipo **EmailField**, que como su nombre indica se utiliza para registrar y validar direcciones de correo electrónico. En la clase `Paciente`, ese tipo de campo corresponde a "email".

Otros de los aspectos destacables en el diseño de la clase `Paciente` es la definición de las propiedades `__get_full_name` y `__unicode__` y de la propiedad `nombre_completo`, todos ellos utilizados para obtener una representación serializada de cada objeto de esta clase en diversas instancias de la capa de presentación.

- La ejecución del comando `python manage.py syncdb` mencionado anteriormente, generará el siguiente script de base de datos que se muestra en la figura 3, que será también ejecutado automáticamente:

```
CREATE TABLE `tesis_paciente` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `tipo_doc` integer NOT NULL,
  `numero_doc` varchar(11) NOT NULL,
  `ap_paterno` varchar(100) NOT NULL,
  `ap_materno` varchar(100) NOT NULL,
  `nombres` varchar(100) NOT NULL,
  `sexo` integer NOT NULL,
  `fecha_nac` date,
  `lugar_nac` varchar(100) NOT NULL,
  `procedencia` varchar(100) NOT NULL,
  `estado_civil` integer NOT NULL,
  `centro_educativo` varchar(200) NOT NULL,
  `grado_instruccion` integer NOT NULL,
  `nombre_padre` varchar(100) NOT NULL,
```

```

`nombre_madre` varchar(100) NOT NULL,
`tipo_seguro` integer NOT NULL,
`numero_seguro` varchar(20) NOT NULL,
`ocupacion` varchar(30) NOT NULL,
`profesion` varchar(30) NOT NULL,
`religion` integer NOT NULL,
`fecha_ult_at_medica` date NOT NULL,
`fecha_ult_at_terapia` date,
`edad` integer,
`domicilio` varchar(100) NOT NULL,
`telefono` varchar(50) NOT NULL,
`email` varchar(75) NOT NULL
)
;

```

Figura 3. Script de creación de la tabla Paciente.

▪ Clase Personal

En la tabla Personal que se muestra la figura 4, como se indicó en el capítulo anterior, se almacenarán los datos correspondientes al personal asistencial (Médicos, Tecnólogos, Enfermeras, Técnicos). Como se aprecia en el diseño mostrado a continuación, los datos requeridos son básicos y sólo para los médicos es importante indicar su número de registro en el Colegio Médico del Perú (“numero_cmp”) y en el caso de los médicos con especialidad, el número de su registro en el Registro Nacional de Especialistas (“RNE”).

Como vemos, también se han definido propiedades que facilitan la presentación de los datos en la capa correspondiente.

```

class Personal(models.Model):
    tipo_personal=models.IntegerField('Grupo ocupacional', choices=TIPO_PERSONAL,
    default=0) #0=medico, 1=tecnologo, 2=enfermera, 3=tecnico
    tipo_doc=models.IntegerField('Tipo documento', choices=TIPO_DOC, default=0)
    #tipo document, default=0 dni
    numero_doc=models.CharField('Numero documento', max_length=11)

```



```

ap_paterno=models.CharField('Apellido paterno', max_length=100)
ap_materno=models.CharField('Apellido materno', max_length=100)
nombres=models.CharField(max_length=100)
sexo=models.IntegerField(choices=SEXO, default=0) #0=masculino,
1=femenino
numero_cmp=models.CharField('CMP', max_length=10, blank=True)
numero_rne=models.CharField('RNE', max_length=10, blank=True)
especialidad=models.CharField(max_length=30, blank=True)
def _get_full_name(self):
    return '%s %s, %s'% (self.ap_paterno, self.ap_materno, self.nombres)

nombre_completo=property(_get_full_name)

def __unicode__(self):
    return self.nombre_completo

```

Figura 4. Clase Personal.

Esta clase personal generará el siguiente script correspondiente a MySQL como se muestra en la figura 5:

```

CREATE TABLE `tesis_personal` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `tipo_persona` integer NOT NULL,
  `tipo_doc` integer NOT NULL,
  `numero_doc` varchar(11) NOT NULL,
  `ap_paterno` varchar(100) NOT NULL,
  `ap_materno` varchar(100) NOT NULL,
  `nombres` varchar(100) NOT NULL,
  `sexo` integer NOT NULL,
  `numero_cmp` varchar(10) NOT NULL,
  `numero_rne` varchar(10) NOT NULL,
  `especialidad` varchar(30) NOT NULL
)

```

Figura 5. Script de creación de la tabla Personal.

▪ Clase Turno

La clase Turno que se observa en la figura 6 se utiliza para crear la estructura de datos que contendrá los datos de los diversos turnos de trabajo y que serán luego agrupados en la tabla Grupo. Estos turnos indican básicamente el día de la semana al que hacen referencia y la hora de inicio y fin (con un intervalo de una hora entre ambas).

La propiedad “__unicode__” presenta cada registro de Turno como una concatenación del día, la hora de inicio y la hora de finalización, por ejemplo: “Lunes – De 09:00 a 10:00 hrs”.

```
class Turno(models.Model):
    dia=models.IntegerField(choices=DIA_SEMANA, default=0) #0=lunes, 1=martes,...
    hora_inicio=models.TimeField()
    hora_final=models.TimeField()
    def _get_turno(self):
        dia_semana=dict((d[0], d[1]) for d in DIA_SEMANA)
        return '%s - De %s a %s hrs' % (dia_semana[self.dia],
self.hora_inicio.strftime("%H:%M"), self.hora_final.strftime("%H:%M"))
    def __unicode__(self):
        return self._get_turno()
```

Figura 6. Clase Turno.

El script de MySQL de la tabla turno que es generado por el framework se muestra en la figura 7:

```
CREATE TABLE `tesis_turno` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `dia` integer NOT NULL,
  `hora_inicio` time NOT NULL,
  `hora_final` time NOT NULL
)
```

Figura 7. Script de creación de la tabla Turno.

▪ Clase Tarea

La clase Tarea que se muestra en la figura 8 se utiliza para crear la estructura de datos que contendrá los datos de las diversas tareas o tipos de tratamiento que deben ser realizados por los tecnólogos.

Luego esta tabla servirá también para definir los tratamientos preferidos por cada profesional. La propiedad “__unicode__” presenta únicamente la descripción de la Tarea a realizar.

```
class Tarea(models.Model):
    descripcion=models.CharField(max_length=200)
    recursos=models.CharField(max_length=200)
    def __unicode__(self):
        return '%s' % (self.descripcion)
```

Figura 8. Clase Tarea.

El script para la generación de la correspondiente tabla en la base de datos se muestra en la figura 9:

```
CREATE TABLE `tesis_tarea` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `descripcion` varchar(200) NOT NULL,
  `recursos` varchar(200) NOT NULL
)
```

Figura 9. Script de creación de la tabla Tarea.

▪ Clase Conflicto

En esta tabla que se muestra en la figura 10 se representa el conflicto que podría existir en la selección de las tareas pendientes entre los diversos tecnólogos. En este modelo se presentan dos nuevos tipos de campos de Django, el campo **ForeignKey**, que representa una relación uno a muchos con la tabla a la que hace referencia, en este caso la tabla de Personal. El otro tipo de campo es el tipo **BooleanField**, que como su nombre indica puede contener valores booleanos; es decir, verdadero o falso.

```

Class Conflicto(models.Model):
    persona=models.ForeignKey(Persona)
    turno1=models.IntegerField()
    turno2=models.IntegerField()
    flag=models.BooleanField(default=False)
  
```

Figura 10. Clase Conflicto.

El código de generación de esta tabla en la base de datos se presenta a continuación en la figura 11:

```

CREATE TABLE `tesis_conflicto` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `persona_id` integer NOT NULL,
  `turno1` integer NOT NULL,
  `turno2` integer NOT NULL,
  `flag` bool NOT NULL
)
;
ALTER TABLE `tesis_conflicto` ADD CONSTRAINT
`persona_id_refs_id_8359fed7` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);
  
```

Figura 11. Generación de tabla Conflicto.

▪ Clase CIE10

En esta tabla que se muestra en la figura 12, se registrará la calificación de las enfermedades diagnosticadas, según la calificación internacional CIE10. Estos datos serán cargados en la base de datos a partir de un archivo XML, elaborado a partir de la tabla en Excel descargada desde la página web de CIE10 (<http://cie10.org>)

```

class Cie_10(models.Model):
    codigo=models.CharField(max_length=8)
    descripcion=models.CharField(max_length=300)
    def __unicode__(self):
        return '%s - %s'%(self.codigo, self.descripcion)
  
```

Figura 12: Clase CIE10.

Los comandos generados para generar esta tabla de la figura 13 se muestran a continuación:

```
CREATE TABLE `tesis_cie_10` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `codigo` varchar(8) NOT NULL,
  `descripcion` varchar(300) NOT NULL
)
```

Figura 13. Script de creación de la tabla CIE10.

▪ Clase Acto_Medico

Esta contiene la información de cada una de las atenciones realizadas a los pacientes y las indicaciones de tratamiento que se les ha realizado. Como vemos la figura 14 está asociada con las clases Paciente, Personal (un subconjunto que hemos denominado “Médico”), Turno y CIE10.

Un nuevo tipo de campo, denominado *TextField*, se ha utilizado en la definición de algunos campos de esta clase. Este tipo de campo permite el registro de textos sin limitarlos a un ancho predeterminado.

```
class Acto_Medico(models.Model):
    paciente=models.ForeignKey(Paciente)
    medico=models.ForeignKey(Persona)
    turno=models.ForeignKey(Turno)
    acto_medico=models.CharField('Número acto médico', max_length=10)
    fecha_consulta=models.DateField()
    consultorio=models.CharField(max_length=10, blank=True)
    cie_10=models.ForeignKey(Cie_10)
    resena_enfermedad=models.TextField('Reseña enfermedad', max_length=300,
    blank=True)
    examen_clinico=models.TextField('Examen clínico', max_length=300, blank=True)
    tratamiento=models.TextField(max_length=300, blank=True)
    evolucion=models.TextField('Evaluación', max_length=300, blank=True)
    recomendaciones=models.TextField(max_length=300, blank=True)
    sesiones_indicadas=models.IntegerField(default=10, null=True)
```

```

sesiones_efectuadas=models.IntegerField(default=0, null=True)

def __unicode__(self):
    return '%s: %s' % (self.acto_medico, self.paciente)

```

Figura 14. Clase Acto_Medico.

De la misma forma que para las clases precedentes, a continuación presentamos el código generado para la creación de la estructura de datos en la figura 15:

```

CREATE TABLE `tesis_acto_medico` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `paciente_id` integer NOT NULL,
  `medico_id` integer NOT NULL,
  `turno_id` integer NOT NULL,
  `acto_medico` varchar(10) NOT NULL,
  `fecha_consulta` date NOT NULL,
  `consultorio` varchar(10) NOT NULL,
  `cie_10_id` integer NOT NULL,
  `resena_enfermedad` longtext NOT NULL,
  `examen_clinico` longtext NOT NULL,
  `tratamiento` longtext NOT NULL,
  `evolucion` longtext NOT NULL,
  `recomendaciones` longtext NOT NULL,
  `sesiones_indicadas` integer,
  `sesiones_efectuadas` integer
)
;
ALTER TABLE `tesis_acto_medico` ADD CONSTRAINT
`turno_id_refs_id_78e4ff98` FOREIGN KEY (`turno_id`) REFERENCES
`tesis_turno` (`id`);
ALTER TABLE `tesis_acto_medico` ADD CONSTRAINT
`paciente_id_refs_id_6ec4d283` FOREIGN KEY (`paciente_id`)
REFERENCES `tesis_paciente` (`id`);
ALTER TABLE `tesis_acto_medico` ADD CONSTRAINT
`cie_10_id_refs_id_6756999d` FOREIGN KEY (`cie_10_id`) REFERENCES

```

```

`tesis_cie_10` (`id`);
ALTER TABLE `tesis_acto_medico` ADD CONSTRAINT
`medico_id_refs_id_6bd91bea` FOREIGN KEY (`medico_id`) REFERENCES
`tesis_persona` (`id`);

```

Figura 15. Script de creación de la tabla Acto_Medico.

▪ Clase Medicamento

Esta tabla de la figura 16, sirve para almacenar los datos del maestro de Medicamentos, que es usado como referencia para la información del servicio de Medicina Física y Rehabilitación del Hospital San José. Sin embargo, no interviene en el cálculo de los horarios óptimos.

```

class Medicamento(models.Model):
    lote=models.CharField('Número lote', max_length=10)
    nombre_generico=models.CharField('Nombre genérico', max_length=100)
    nombre_comercial=models.CharField('Nombre comercial', max_length=100, blank=True)
    unidad_compra=models.IntegerField(default=0)
    unidad_despacho=models.IntegerField(default=0)
    stock_actual=models.IntegerField(default=0)
    fecha_vencimiento=models.DateField()
    referencia_utilizacion=models.CharField('Referencia utilización', max_length=300,
blank=True)
    dosis_recomendada=models.CharField('Dosis recomendada', max_length=100,
blank=True)
    def __unicode__(self):
        return self.nombre_generico

```

Figura 16. Clase Medicamento.

El script de creación de la tabla en la base de datos se muestra en la figura 17:

```

CREATE TABLE `tesis_medicamento` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `lote` varchar(10) NOT NULL,
  `nombre_generico` varchar(100) NOT NULL,
  `nombre_comercial` varchar(100) NOT NULL,

```

```

`unidad_compra` integer NOT NULL,
`unidad_despacho` integer NOT NULL,
`stock_actual` integer NOT NULL,
`fecha_vencimiento` date NOT NULL,
`referencia_utilizacion` varchar(300) NOT NULL,
`dosis_recomendada` varchar(100) NOT NULL
)

```

Figura 17. Script de creación de la tabla Medicamento.

▪ Clase Medicamento_por_Acto_Medico

Esta clase que se muestra en la figura 18, como su nombre indica, registra los medicamentos que son recetados a los pacientes en cada consulta o acto médico. Sirve de referencia al personal asistencial para conocer que está consumiendo el Paciente en el momento de sus terapias y tener una idea de las posibles interacciones que podría tener su medicación con su tratamiento en el servicio.

```

class Medicamento_por_Acto_Medico(models.Model):
    """
    paciente=models.ForeignKey(Paciente)
    persona=models.ForeignKey(Persona)
    """
    acto_medico=models.ForeignKey(Acto_Medico)
    medicamento=models.ForeignKey(Medicamento)
    unidad_medida=models.IntegerField(default=0)
    frecuencia_dosis=models.IntegerField(default=0)
    duracion_tratamiento=models.IntegerField(default=0)
    def __unicode__(self):
        return '%s'%(self.medicamento)

    def total_dosis(self):
        return self.frecuencia_dosis*self.duracion_tratamiento

```

Figura 18. Clase Medicamento_por_Acto_Medico.

El código para la creación de esta tabla es presentado en la siguiente figura 19:


```

CREATE TABLE `tesis_medicamento_por_acto_medico` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `acto_medico_id` integer NOT NULL,
  `medicamento_id` integer NOT NULL,
  `unidad_medida` integer NOT NULL,
  `frecuencia_dosis` integer NOT NULL,
  `duracion_tratamiento` integer NOT NULL
)
;
ALTER TABLE `tesis_medicamento_por_acto_medico` ADD CONSTRAINT
`acto_medico_id_refs_id_756483e5` FOREIGN KEY (`acto_medico_id`)
REFERENCES `tesis_acto_medico` (`id`);

ALTER TABLE `tesis_medicamento_por_acto_medico` ADD CONSTRAINT
`medicamento_id_refs_id_c3d0b507` FOREIGN KEY (`medicamento_id`)
REFERENCES `tesis_medicamento` (`id`);

```

Figura 19. Script de creación de la tabla Medicamento_por_Acto_Medico.

▪ Clase Preferencia

Esta clase de la figura 20, registra los tratamientos preferidos por los tecnólogos. La asignación de tratamientos diferentes a estos generará un incremento en el costo de los horarios al ejecutarse el algoritmo de optimización:

```

class Preferencia(models.Model):
    persona=models.ForeignKey(Persona)
    tratamiento1=models.IntegerField(default=0)
    tratamiento2=models.IntegerField(default=1)
    tratamiento3=models.IntegerField(default=2)

```

Figura 20. Clase Preferencia.

El código necesario para crear esta tabla en MySQL se presenta en la siguiente figura 21:

```
CREATE TABLE `tesis_preferencia` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `persona_id` integer NOT NULL,
  `tratamiento1` integer NOT NULL,
  `tratamiento2` integer NOT NULL,
  `tratamiento3` integer NOT NULL
)
;
ALTER TABLE `tesis_preferencia` ADD CONSTRAINT
`persona_id_refs_id_7fa418f5` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);
```

Figura 21. Script de creación de la tabla Preferencia.

▪ Clase Requisito

Registra la ocurrencia de las diversas tareas con los turnos programados como se muestra en la figura 22.

```
class Requisito(models.Model):
    tarea=models.ForeignKey(Tarea)
    turno=models.ForeignKey(Turno)
    cantidad_ocurrencias=models.IntegerField(default=0)
```

Figura 22. Clase Requisito.

Y el script para la generación de esta tabla de la figura 23 en MySql es el siguiente:

```
CREATE TABLE `tesis_requisito` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `tarea_id` integer NOT NULL,
  `turno_id` integer NOT NULL,
  `cantidad_ocurrencias` integer NOT NULL
)
;
ALTER TABLE `tesis_requisito` ADD CONSTRAINT `turno_id_refs_id_46c55047`
FOREIGN KEY (`turno_id`) REFERENCES `tesis_turno` (`id`);
```

```
ALTER TABLE `tesis_requisito` ADD CONSTRAINT `tarea_id_refs_id_90e2c8bf`
FOREIGN KEY (`tarea_id`) REFERENCES `tesis_tarea` (`id`);
```

Figura 23. Script de creación de la tabla Requisito.

▪ Clase Capacidad

Indica si un determinado tecnólogo está en la capacidad técnica de efectuar un determinado tratamiento, independientemente de sus preferencias.

En esta clase se presenta un nuevo tipo de campo que es el *ManyToManyField*. Este indica una relación de muchos a muchos, en este caso significa que muchos tecnólogos pueden realizar muchos tratamientos y que muchos tratamientos pueden ser realizados por muchos tecnólogos. Como se muestra en la figura 24.

```
class Capacidad(models.Model):
    persona=models.ForeignKey(Persona)
    tarea=models.ManyToManyField(Tarea)
    def __unicode__(self):
        return '%s'%(self.persona)
```

Figura 24. Clase Capacidad.

El script generado por Django para la creación de la tabla Capacidad en la base de datos se muestra en la figura 25. Como podemos apreciar, el framework genera automáticamente las tablas necesarias para expresar adecuadamente el tipo de relación que estamos estableciendo en el modelo.

```
CREATE TABLE `tesis_capacidad_tarea` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `capacidad_id` integer NOT NULL,
  `tarea_id` integer NOT NULL,
  UNIQUE (`capacidad_id`, `tarea_id`)
)
;
ALTER TABLE `tesis_capacidad_tarea` ADD CONSTRAINT
`tarea_id_refs_id_bfee1438` FOREIGN KEY (`tarea_id`) REFERENCES
```

```

`tesis_tarea` (`id`);

CREATE TABLE `tesis_capacidad` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `persona_id` integer NOT NULL
)
;
ALTER TABLE `tesis_capacidad` ADD CONSTRAINT
`persona_id_refs_id_de790e63` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);
ALTER TABLE `tesis_capacidad_tarea` ADD CONSTRAINT
`capacidad_id_refs_id_80e6d426` FOREIGN KEY (`capacidad_id`) REFERENCES
`tesis_capacidad` (`id`);

```

Figura 25. Script de creación de la tabla Capacidad.

▪ Clase Disponibilidad

La clase Disponibilidad que se muestra en la figura 26, es utilizada para registrar la disponibilidad de un determinado personal asistencial en un turno determinado. Nuevamente, la relación del campo Turno es de mucho a muchos para indicar que en un turno puede haber muchos profesionales y que cada profesional puede tener muchos turnos disponibles.

```

class Disponibilidad(models.Model):
    persona=models.ForeignKey(Persona)
    turno=models.ManyToManyField(Turno)
    def __unicode__(self):
        return '%s'%(self.persona)

```

Figura 26. Clase Disponibilidad.

El script de generación de estas relaciones en la base de datos se presenta a continuación en la figura 27:

```

CREATE TABLE `tesis_disponibilidad_turno` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `disponibilidad_id` integer NOT NULL,

```

```

`turno_id` integer NOT NULL,
    UNIQUE (`disponibilidad_id`, `turno_id`)
)
;
ALTER TABLE `tesis_disponibilidad_turno` ADD CONSTRAINT
`turno_id_refs_id_d5e9f3c0` FOREIGN KEY (`turno_id`) REFERENCES
`tesis_turno` (`id`);
CREATE TABLE `tesis_disponibilidad` (
    `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
    `persona_id` integer NOT NULL
)
;
ALTER TABLE `tesis_disponibilidad` ADD CONSTRAINT
`persona_id_refs_id_074902d5` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);
ALTER TABLE `tesis_disponibilidad_turno` ADD CONSTRAINT
`disponibilidad_id_refs_id_3f6ac030` FOREIGN KEY (`disponibilidad_id`)
REFERENCES `tesis_disponibilidad` (`id`);

```

Figura 27. Script de creación de la tabla Disponibilidad.

▪ Clase Grupo_Turno

Esta clase que se muestra en la figura 28 permite la agrupación de los diversos grupos horarios en bloques de horarios, por ejemplo, el bloque de la mañana de los lunes, miércoles y viernes, o el bloque de las tardes los martes, jueves y sábados.

```

class Grupo_Turno(models.Model):
    """
    Grupo de turnos
    """
    nombre=models.CharField(max_length=100)
    turno=models.ManyToManyField(Turno)
    def __unicode__(self):
        return '%s'%(self.nombre)

```

Figura 28. Clase Grupo_Turno.

El código de MySQL para la creación de esta tabla se muestra en la figura 29:

```
CREATE TABLE `tesis_grupo_turno_turno` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `grupo_turno_id` integer NOT NULL,
  `turno_id` integer NOT NULL,
  UNIQUE (`grupo_turno_id`, `turno_id`)
)
;
ALTER TABLE `tesis_grupo_turno_turno` ADD CONSTRAINT
`turno_id_refs_id_08b25b63` FOREIGN KEY (`turno_id`) REFERENCES
`tesis_turno` (`id`);
CREATE TABLE `tesis_grupo_turno` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `nombre` varchar(100) NOT NULL
)
;
ALTER TABLE `tesis_grupo_turno_turno` ADD CONSTRAINT
`grupo_turno_id_refs_id_f847d281` FOREIGN KEY (`grupo_turno_id`)
REFERENCES `tesis_grupo_turno` (`id`);
```

Figura 29. Script de creación de la tabla Grupo_Turno.

▪ Clase Carga_Minima

Esta clase que se muestra en la figura 30 simplemente registra la carga mínima, es decir el número mínimo de atenciones, que debe efectuar un determinado profesional por cada turno.

```
class Carga_Minima(models.Model):
    """
    Límite inferior de número de tareas que se puede asignar a cada
    empleado. Otro límite es asignar un número limitado total de tareas por horario
    y también un limitado número de asignaciones específicas que debe hacer el
    empleado.
    Por lo tanto, un conjunto de turnos G1, ..., Gs, donde cada conjunto especifica un
    conjunto de turnos. Luego se define dos matrices de enteros Vms y Wms tal que
    Ei empleado sea asignado por lo menos en los turnos Vik y a lo máximo en los
```

turnos Wik del grupo Gk. El problema es encontrar cualquier asignación que satisfice todas las restricciones anteriores. Otras definiciones incluyen también restricciones de la función objetivo (a ser minimizada) y un justo balanceo de carga de trabajo para los empleados.

```

'''
persona=models.ForeignKey(Persona)
grupo=models.ForeignKey(Grupo_Turno)
turno=models.ForeignKey(Turno)
def __unicode__(self):
    return '%s'%(self.persona)

```

Figura 30. Clase Carga_Minima.

El código Sql para la creación de la tabla se muestra en la figura 31:

```

CREATE TABLE `tesis_carga_minima` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `persona_id` integer NOT NULL,
  `grupo_id` integer NOT NULL,
  `turno_id` integer NOT NULL
)
;
ALTER TABLE `tesis_carga_minima` ADD CONSTRAINT
`grupo_id_refs_id_b7c080b1` FOREIGN KEY (`grupo_id`) REFERENCES
`tesis_grupo_turno` (`id`);
ALTER TABLE `tesis_carga_minima` ADD CONSTRAINT
`turno_id_refs_id_99d379f4` FOREIGN KEY (`turno_id`) REFERENCES
`tesis_turno` (`id`);
ALTER TABLE `tesis_carga_minima` ADD CONSTRAINT
`persona_id_refs_id_704ecbed` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);

```

Figura 31. Script de creación de la tabla Carga_Minima.

▪ Clase Carga_Maxima

Esta clase mostrada en la figura 32, complementa a la anterior y registra la carga máxima, que se puede asignar a un determinado profesional por cada turno.


```
class Carga_Maxima(models.Model):
    persona=models.ForeignKey(Persona)
    grupo=models.ForeignKey(Grupo_Turno)
    turno=models.ForeignKey(Turno)
    def __unicode__(self):
        return '%s'%(self.persona)
```

Figura 32. Clase Carga_Maxima.

Los comandos de MySQL generados por Django se presentan en la figura 33:

```
CREATE TABLE `tesis_carga_maxima` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `persona_id` integer NOT NULL,
  `grupo_id` integer NOT NULL,
  `turno_id` integer NOT NULL
)
;
ALTER TABLE `tesis_carga_maxima` ADD CONSTRAINT
`grupo_id_refs_id_98bae32c` FOREIGN KEY (`grupo_id`) REFERENCES
`tesis_grupo_turno` (`id`);
ALTER TABLE `tesis_carga_maxima` ADD CONSTRAINT
`turno_id_refs_id_015bfdcc` FOREIGN KEY (`turno_id`) REFERENCES
`tesis_turno` (`id`);
ALTER TABLE `tesis_carga_maxima` ADD CONSTRAINT
`persona_id_refs_id_7328403e` FOREIGN KEY (`persona_id`) REFERENCES
`tesis_persona` (`id`);
```

Figura 33. Script de creación de la tabla Carga_Maxima.

▪ Clase Pago

Esta clase registra los pagos que ha realizado el paciente a lo largo de su tratamiento. Los pagos deben realizarse antes de su terapia, por lo que la no comprobación del pago correspondiente es utilizada para poder determinar los pacientes que han desertado de su tratamiento.

En esta clase se hace uso de un nuevo tipo de campo, DecimalField, que es utilizado para registrar el monto de los pagos correspondientes. Como se muestra en la figura 4.34.

```
class Pago(models.Model):
    paciente=models.ForeignKey(Paciente)
    persona=models.ForeignKey(Persona)
    turno=models.ForeignKey(Turno)
    numero_sesion=models.IntegerField()
    fecha_pago=models.DateField('Fecha ultimo pago')
    monto_pago=models.DecimalField(max_digits=10, decimal_places=2) #pago sin igv
    monto_igv=models.DecimalField(max_digits=10, decimal_places=2) #pago igv
    monto_total=models.DecimalField(max_digits=10, decimal_places=2) #pago total
    numero_doc_pago=models.CharField(max_length=50)
```

Figura 34. Clase Pago.


El script de creación de esta tabla en la base de datos MySql, es el presentado en la figura 35:

```
CREATE TABLE `tesis_pago` (
  `id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `paciente_id` integer NOT NULL,
  `persona_id` integer NOT NULL,
  `turno_id` integer NOT NULL,
  `numero_sesion` integer NOT NULL,
  `fecha_pago` date NOT NULL,
  `monto_pago` numeric(10, 2) NOT NULL,
  `monto_igv` numeric(10, 2) NOT NULL,
  `monto_total` numeric(10, 2) NOT NULL,
  `numero_doc_pago` varchar(50) NOT NULL
)
;
ALTER TABLE `tesis_pago` ADD CONSTRAINT `turno_id_refs_id_49009c1d`
FOREIGN KEY (`turno_id`) REFERENCES `tesis_turno` (`id`);
ALTER TABLE `tesis_pago` ADD CONSTRAINT `paciente_id_refs_id_155370cf`
```

```
FOREIGN KEY (`paciente_id`) REFERENCES `tesis_paciente` (`id`);  
ALTER TABLE `tesis_pago` ADD CONSTRAINT `persona_id_refs_id_c6b15ffd`  
FOREIGN KEY (`persona_id`) REFERENCES `tesis_persona` (`id`);
```

Figura 35: Script de creación de la tabla Pago.





ANEXO 4

**DESCRIPCIÓN DE LAS PANTALLAS DEL
SOFTWARE**

Índice de figuras

Figura 1. Relación de actos médicos.....	2
Figura 2. Detalle del Acto médico, parte 1.	3
Figura 3. Detalle del Acto médico, parte 2.	3
Figura 4. Mantenimiento de la tabla de Capacidad.....	4
Figura 5: Mantenimiento de la tabla de CIE10.	4
Figura 6: Detalle de la tabla de CIE10.....	5
Figura 7. Pantalla de mantenimiento de Grupos de turnos.	5
Figura 8. Detalle de la tabla de Medicamento por acto médico.....	6
Figura 9. Pantalla de mantenimiento del maestro de Medicamentos.....	7
Figura 10. Pantalla de detalle de Medicamentos.	7
Figura 11. Pantalla de pacientes (lista).	8
Figura 12. Pantalla de detalle de pacientes mostrando datos generales.....	9
Figura 13: Pantalla de detalle de pacientes mostrando datos de educación.	10
Figura 14. Pantalla de detalle de pacientes mostrando datos de los padres.....	10
Figura 15: Pantalla de detalle de pacientes mostrando datos de las atenciones. .	11
Figura 16. Pantalla de detalle de Pagos.....	11

DESCRIPCION DE LAS PANTALLAS DEL SOFTWARE

1. Pantalla de Actos Médicos

En esta pantalla de la figura 1, el médico registrará los datos correspondientes a la consulta que está efectuando a un determinado paciente. Como se puede apreciar, en un primer lugar se presenta la relación de los pacientes, el médico que los atiende y las fecha de consulta.

Si se desea ver el detalle de un determinado acto médico, se debe hacer clic en la fila que corresponda.

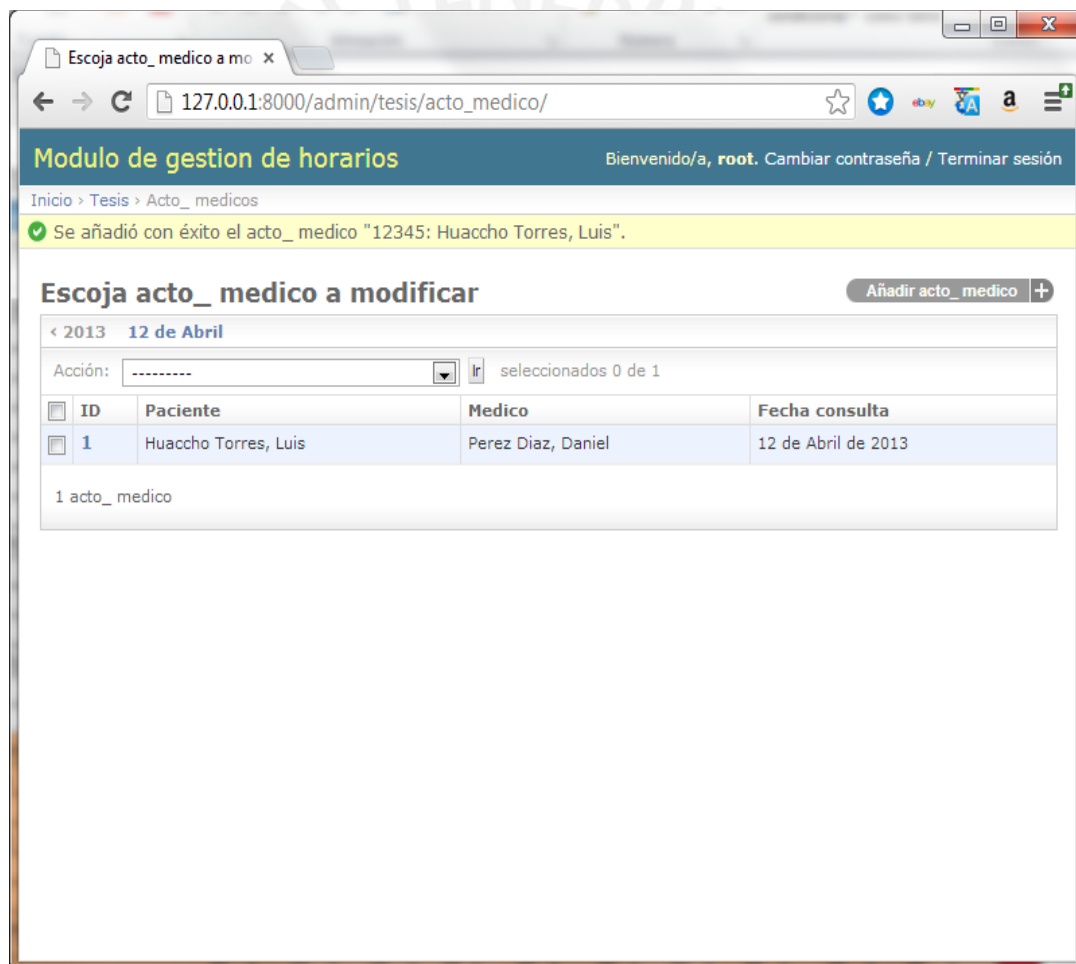


Figura 1. Relación de actos médicos.

Una vez que se ha seleccionado un acto médico para consultar, se presenta el siguiente detalle de la figura 2 y 3:

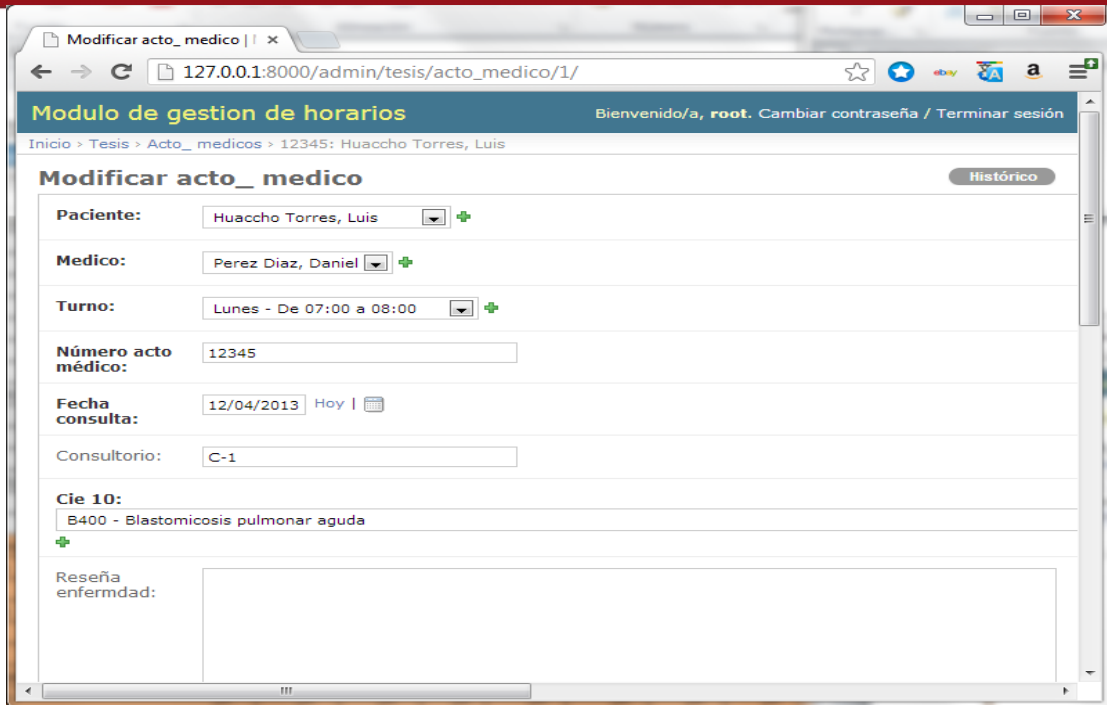


Figura 2. Detalle del Acto médico, parte 1.

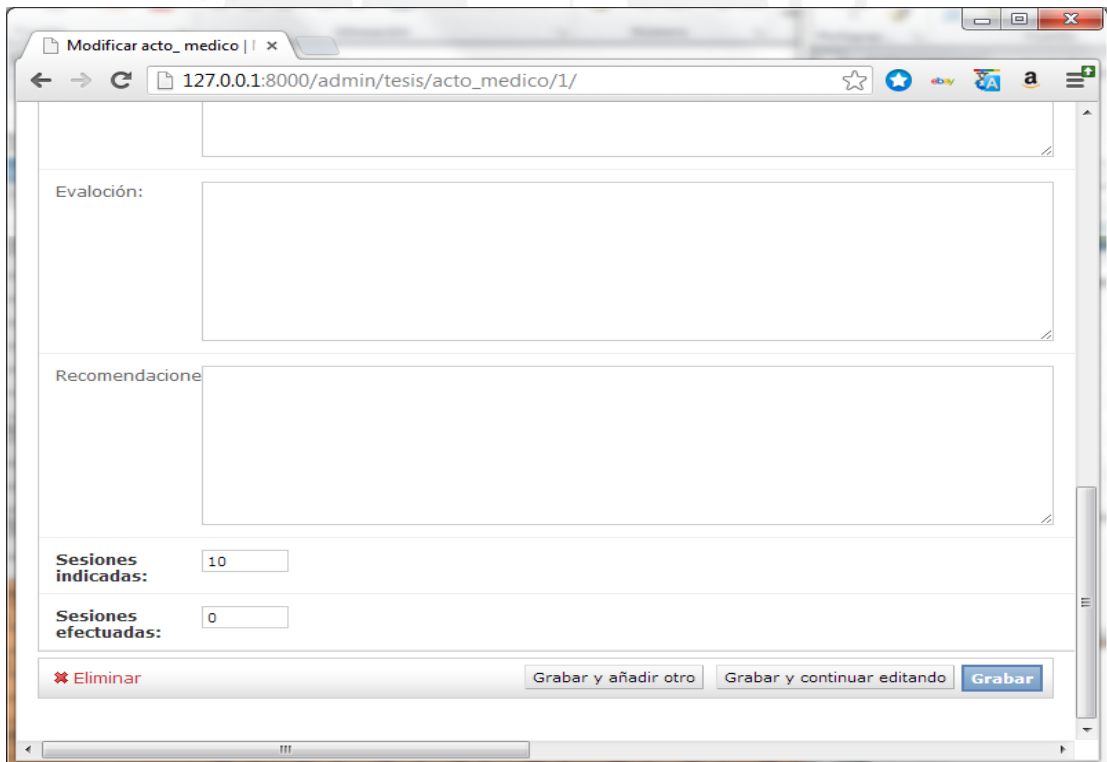


Figura 3. Detalle del Acto médico, parte 2.

En esta pantalla (mostrada en dos partes) se presenta el detalle del acto médico efectuado, seleccionando el paciente, el médico, el turno, el código CIE10, entre otros datos relevantes. En caso de que no existiera aún un registro del paciente o

del médico, se hace clic en el signo de suma “+” para abrir la pantalla correspondiente.

2. Pantalla de Capacidad

En esta pantalla de la figura 4 se da mantenimiento a la tabla de Capacidad; es decir, se registran las tareas que cada profesional es capaz de efectuar.

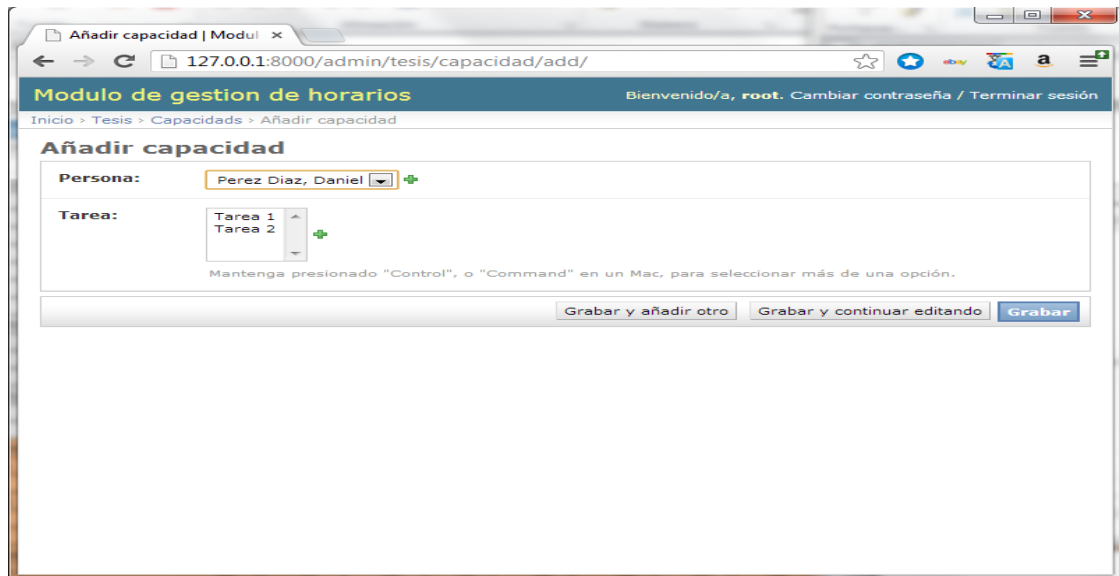


Figura 4. Mantenimiento de la tabla de Capacidad.

3. Pantalla de CIE10

En esta pantalla de la figura 5, se realiza el mantenimiento de la tabla de códigos de identificación de enfermedades.



Figura 5: Mantenimiento de la tabla de CIE10.

Al hacer clic en alguna de las filas de esta pantalla, se presenta la pantalla de la figura 6:

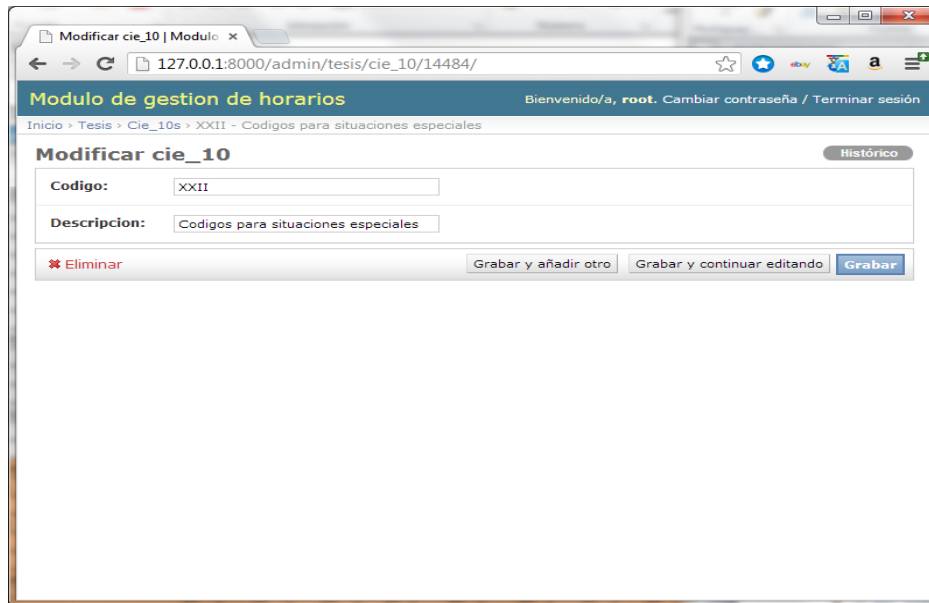


Figura 6: Detalle de la tabla de CIE10.

4. Pantalla de Grupo de turno

En esta pantalla de la figura 7, se realiza la creación o mantenimiento de los grupos de turnos. Como en el diseño se determinó que existía una relación de muchos a muchos con la tabla de turnos, el framework genera una pantalla adecuada a esta situación:

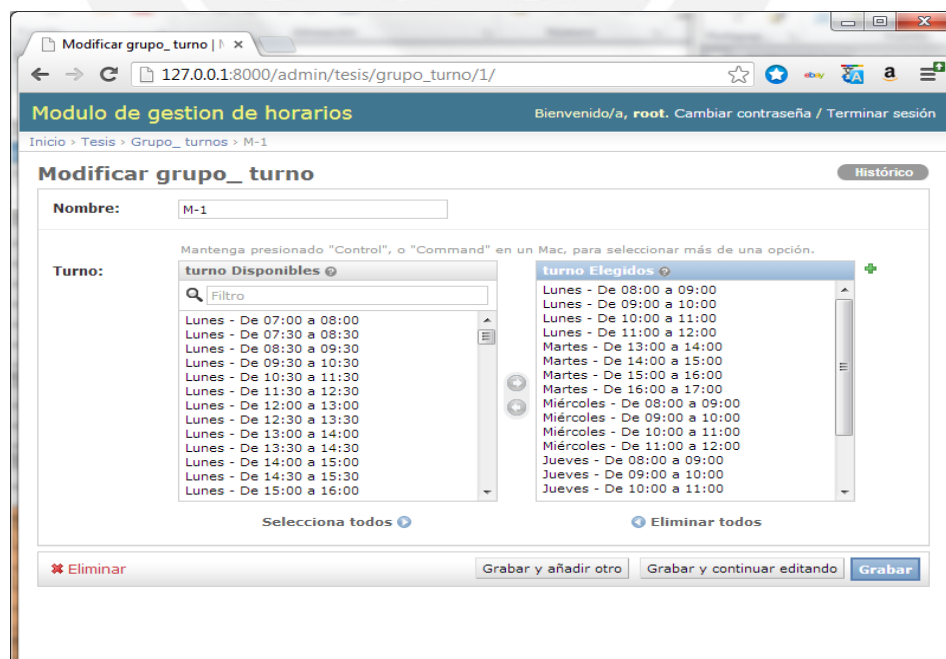
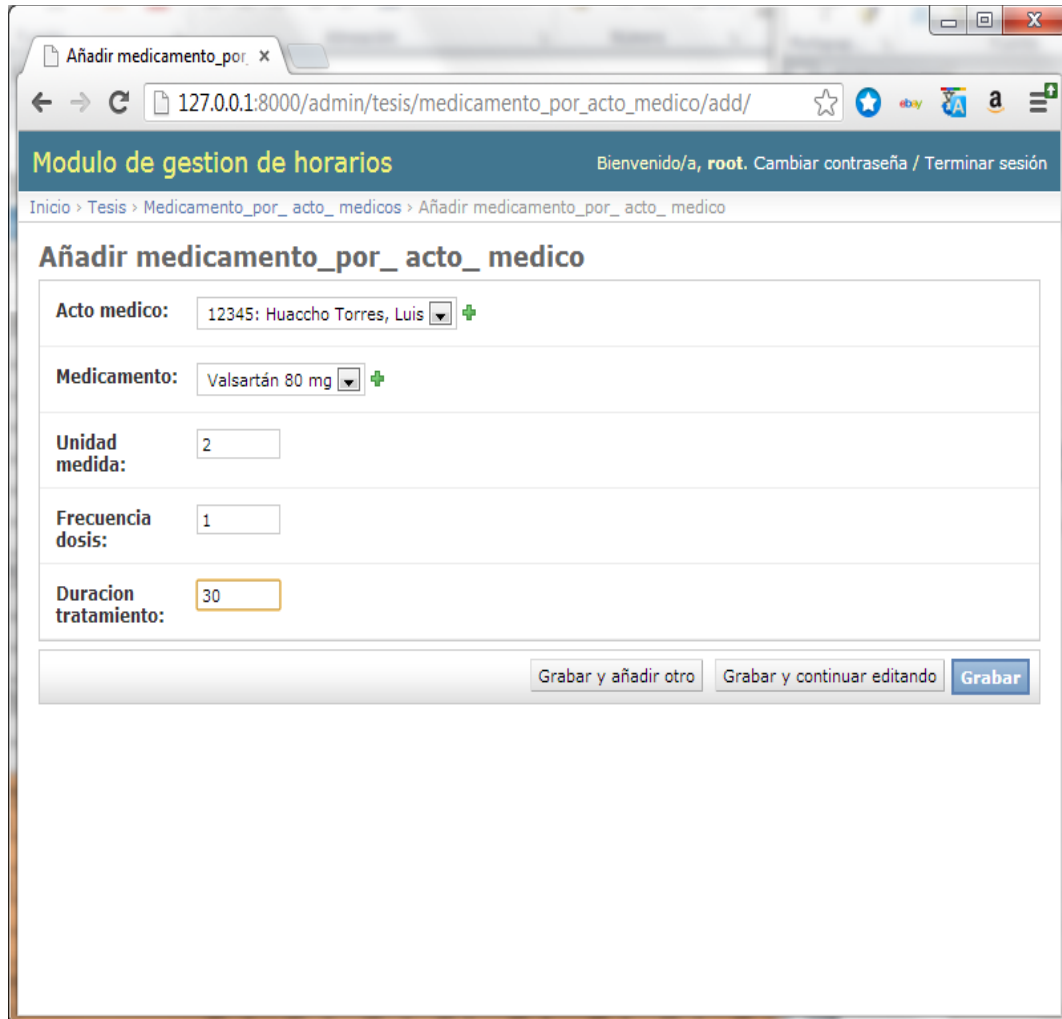


Figura 7. Pantalla de mantenimiento de Grupos de turnos.

5. Pantalla de Medicamentos por acto médico

En esta pantalla de la figura 8, se realiza el registro de los medicamentos indicados en cada acto médico. De forma similar a otras pantallas, en esta se puede seleccionar actos médicos ya registrados o crear nuevos registros haciendo clic en el ícono correspondiente. Lo mismo ocurre en el caso de los medicamentos disponibles.



Añadir medicamento_por_ x

127.0.0.1:8000/admin/tesis/medicamento_por_acto_medico/add/

Modulo de gestion de horarios Bienvenido/a, root. Cambiar contraseña / Terminar sesión

Inicio > Tesis > Medicamento_por_acto_medicos > Añadir medicamento_por_acto_medico

Añadir medicamento_por_acto_medico

Acto medico:	12345: Huaccho Torres, Luis +
Medicamento:	Valsartán 80 mg +
Unidad medida:	2
Frecuencia dosis:	1
Duracion tratamiento:	30

Grabar y añadir otro Grabar y continuar editando Grabar

Figura 8. Detalle de la tabla de Medicamento por acto médico

6. Pantalla de Medicamentos

En esta pantalla de la figura 9, se puede realizar el mantenimiento del maestro de Medicamentos.

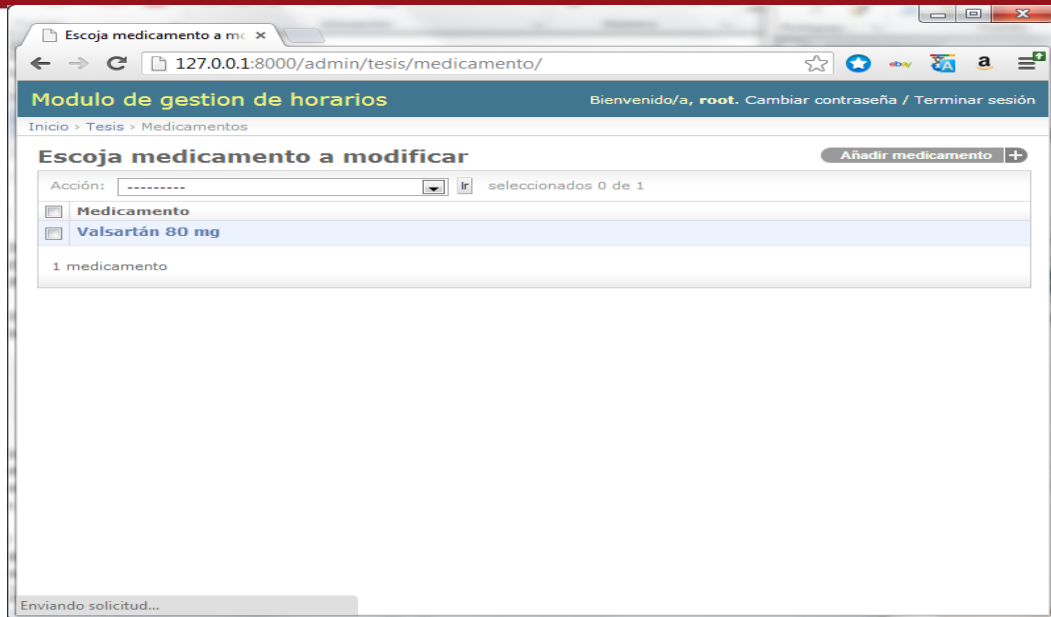


Figura 9. Pantalla de mantenimiento del maestro de Medicamentos.

Al hacer clic en un determinado medicamento de la lista, se abre la pantalla de detalle que se presenta a continuación en la figura 10:

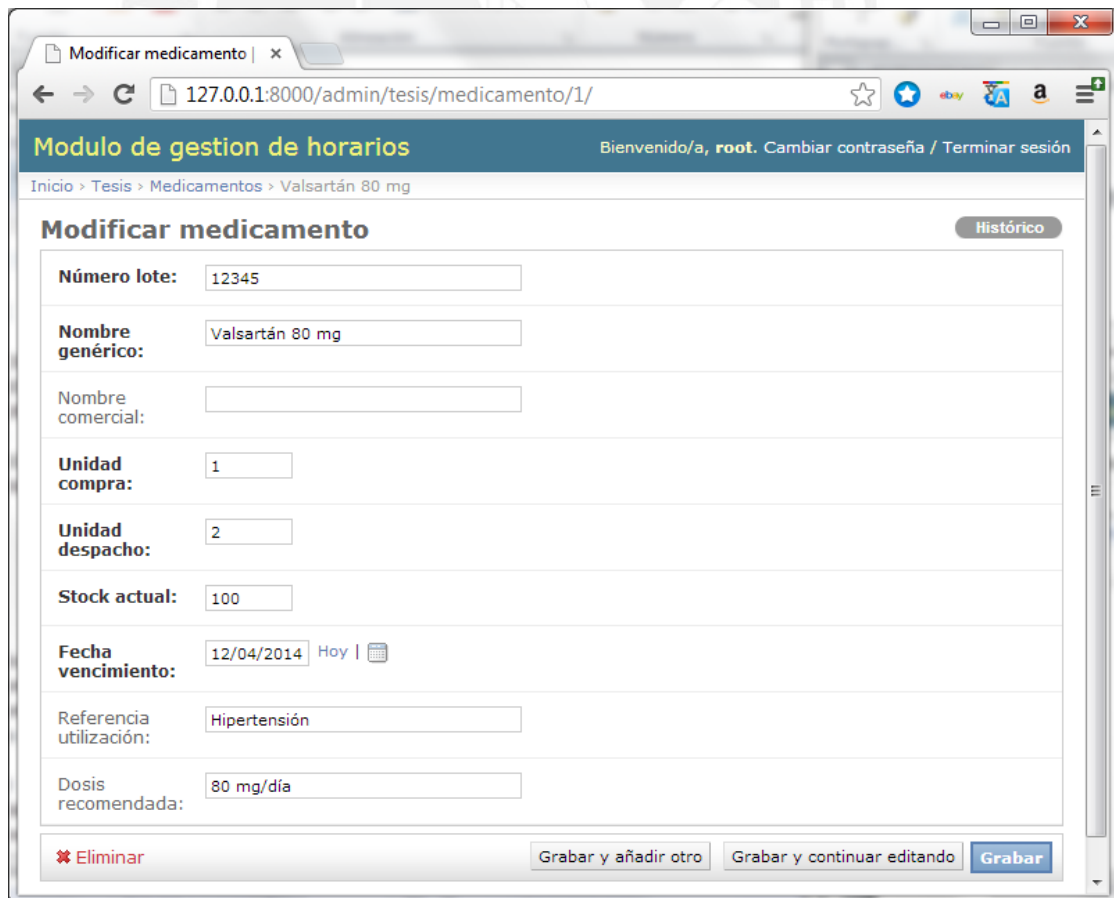


Figura 10. Pantalla de detalle de Medicamentos.

7. Pantalla de Pacientes

En esta pantalla de la figura 11 se listan los pacientes existentes en la base de datos. Como se aprecia, en esta pantalla se puede buscar a un determinado paciente o aplicar filtros para poder seleccionarlos.

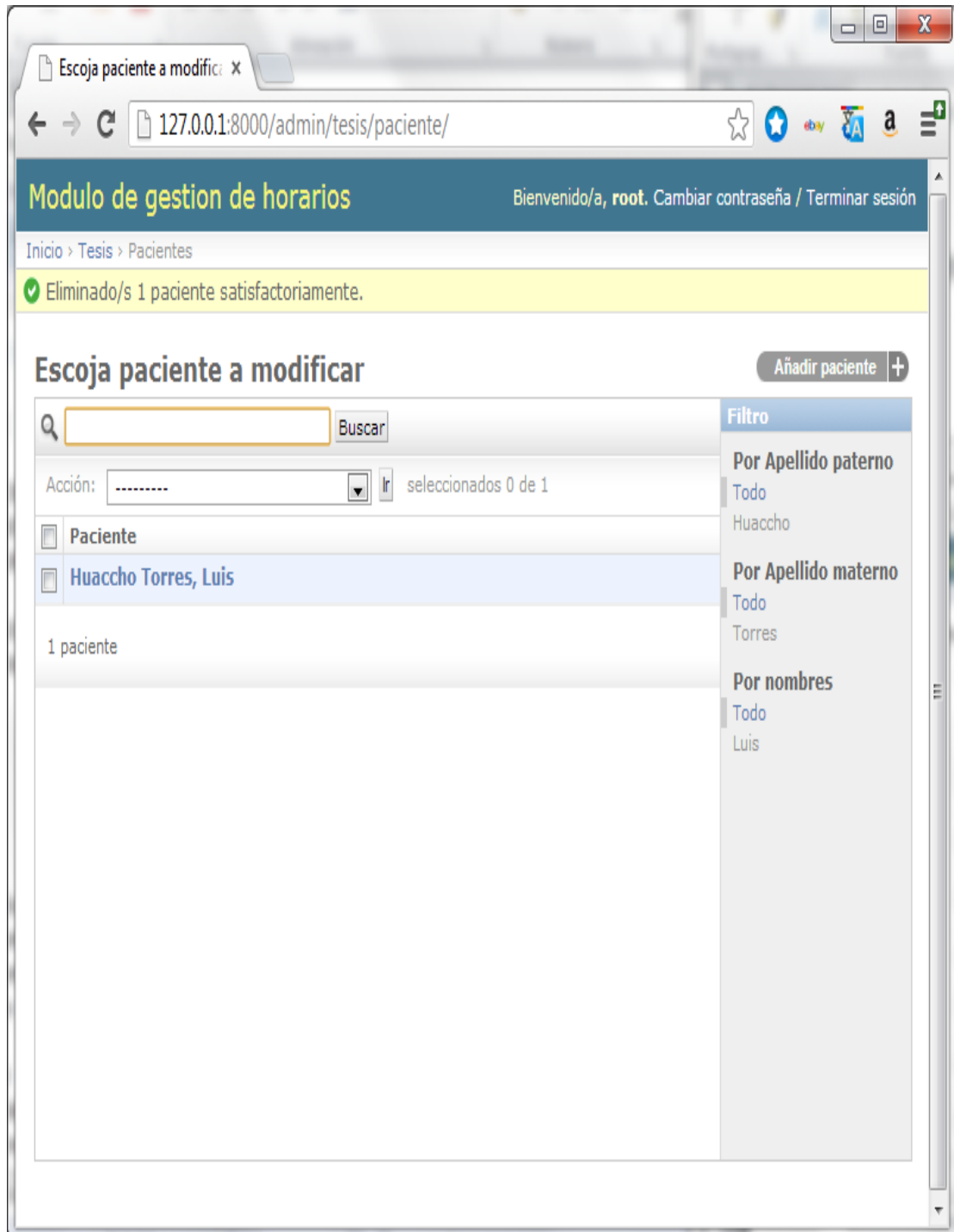
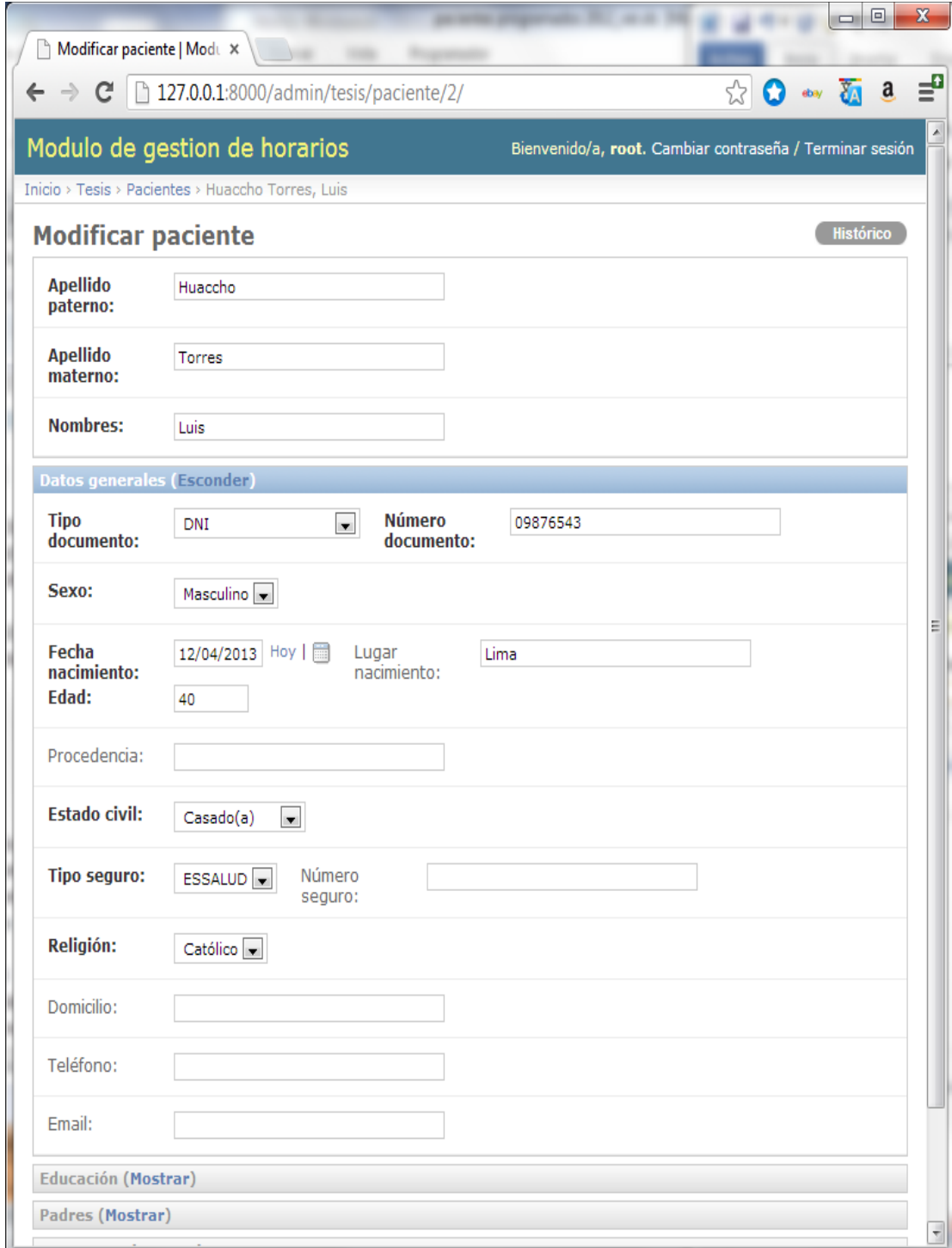


Figura 11. Pantalla de pacientes (lista).

El detalle de cada paciente se muestra a continuación en figura 12, 13, 14 y 15. Esta pantalla tiene la posibilidad de agrupar los datos y mostrarlos según sea conveniente:



Modificar paciente | Mod: x

127.0.0.1:8000/admin/tesis/paciente/2/

Modulo de gestion de horarios Bienvenida/a, root. Cambiar contraseña / Terminar sesión

Inicio > Tesis > Pacientes > Huaccho Torres, Luis

Modificar paciente Historico

Apellido paterno:	<input type="text" value="Huaccho"/>
Apellido materno:	<input type="text" value="Torres"/>
Nombres:	<input type="text" value="Luis"/>

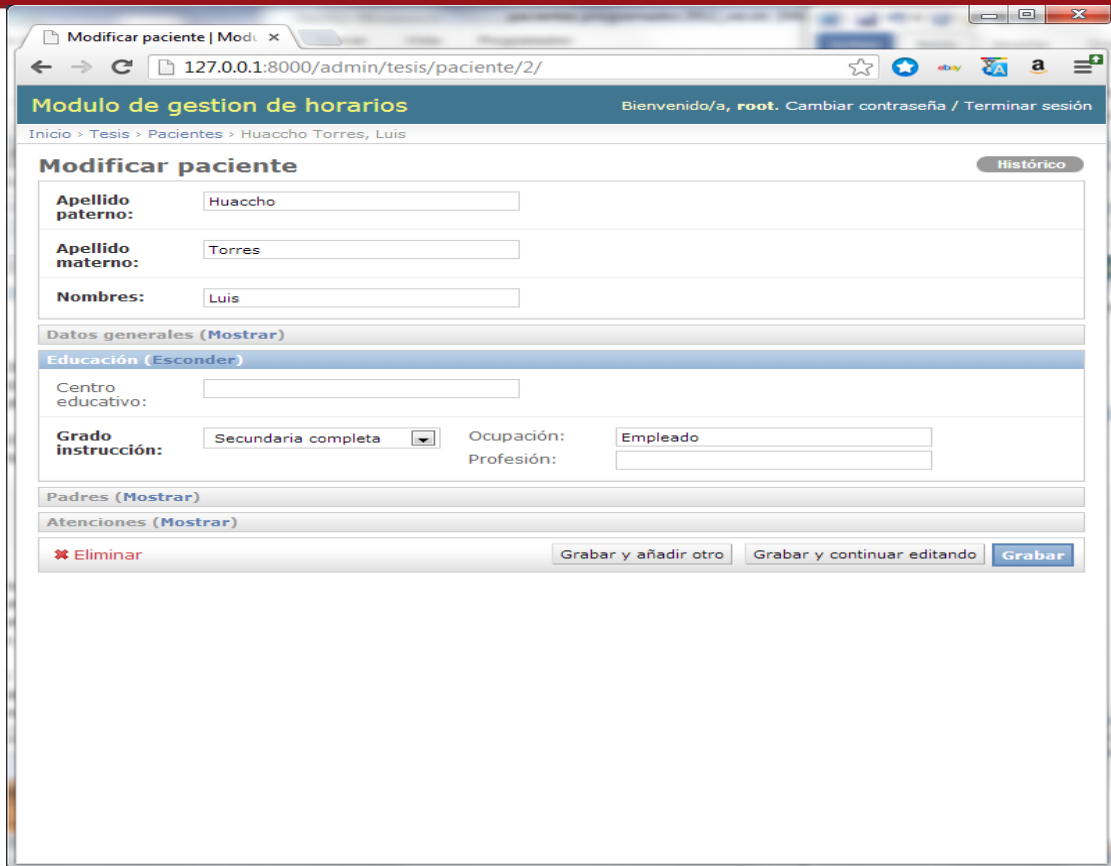
Datos generales (Esconder)

Tipo documento:	<input type="text" value="DNI"/>	Número documento:	<input type="text" value="09876543"/>
Sexo:	<input type="text" value="Masculino"/>		
Fecha nacimiento:	<input type="text" value="12/04/2013"/> Hoy <input type="text" value="12/04/2013"/>	Lugar nacimiento:	<input type="text" value="Lima"/>
Edad:	<input type="text" value="40"/>		
Procedencia:	<input type="text"/>		
Estado civil:	<input type="text" value="Casado(a)"/>		
Tipo seguro:	<input type="text" value="ESSALUD"/>	Número seguro:	<input type="text"/>
Religión:	<input type="text" value="Católico"/>		
Domicilio:	<input type="text"/>		
Teléfono:	<input type="text"/>		
Email:	<input type="text"/>		

Educación (Mostrar)

Padres (Mostrar)

Figura 12. Pantalla de detalle de pacientes mostrando datos generales.



Modificar paciente | Mod: x
127.0.0.1:8000/admin/tesis/paciente/2/

Modulo de gestion de horarios Bienvenido/a, root. Cambiar contraseña / Terminar sesión

Inicio > Tesis > Pacientes > Huaccho Torres, Luis

Modificar paciente

Apellido paterno:

Apellido materno:

Nombres:

Datos generales (Mostrar)

Educación (Esconder)

Centro educativo:

Grado instrucción: Ocupación:

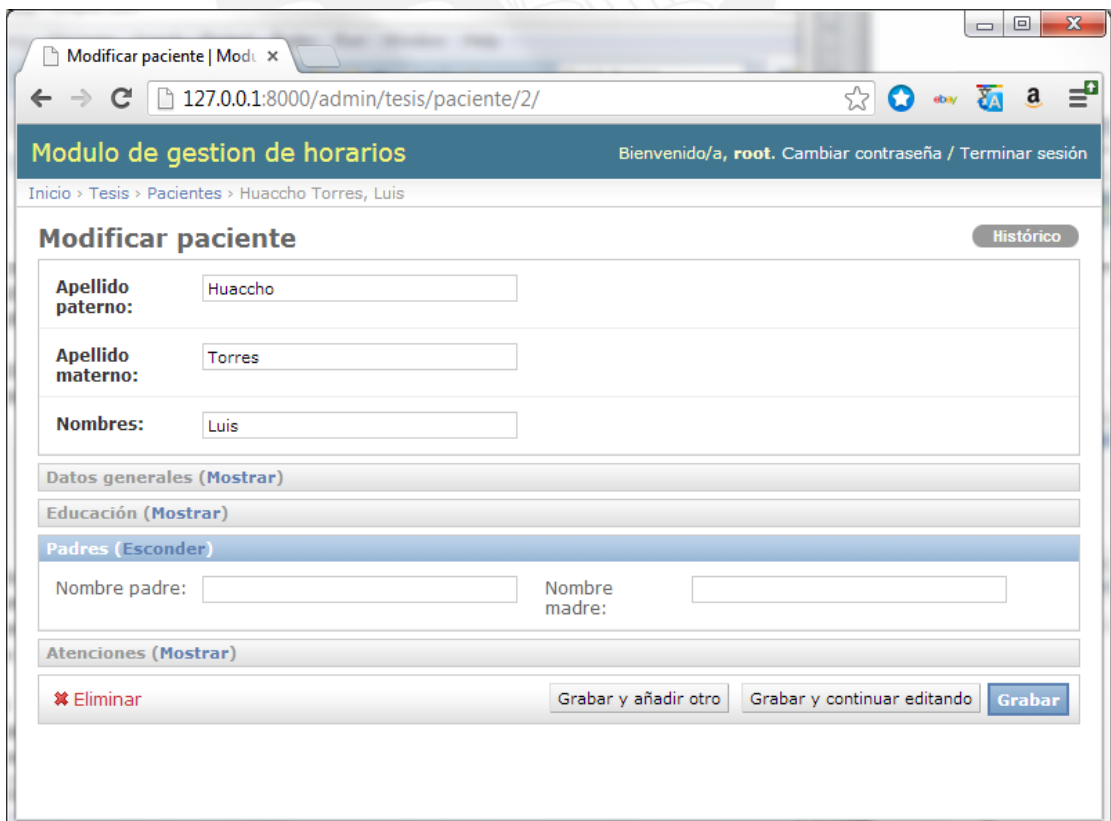
Profesión:

Padres (Mostrar)

Atenciones (Mostrar)

✘ Eliminar

Figura 13: Pantalla de detalle de pacientes mostrando datos de educación.



Modificar paciente | Mod: x
127.0.0.1:8000/admin/tesis/paciente/2/

Modulo de gestion de horarios Bienvenido/a, root. Cambiar contraseña / Terminar sesión

Inicio > Tesis > Pacientes > Huaccho Torres, Luis

Modificar paciente

Apellido paterno:

Apellido materno:

Nombres:

Datos generales (Mostrar)

Educación (Mostrar)

Padres (Esconder)

Nombre padre: Nombre madre:

Atenciones (Mostrar)

✘ Eliminar

Figura 14: Pantalla de detalle de pacientes mostrando datos de los padres.

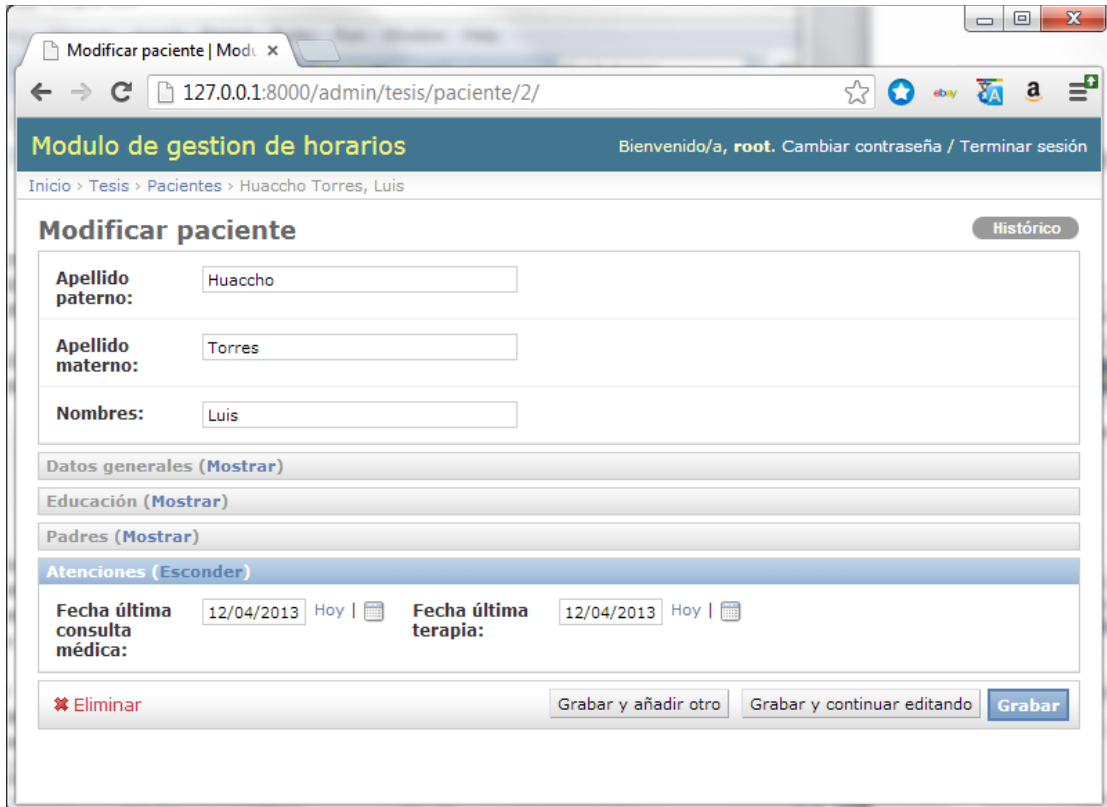


Figura 15: Pantalla de detalle de pacientes mostrando datos de las atenciones.

8. Pantalla de Pago

En esta pantalla de la figura 16 se registran los pagos realizados efectuados por los pacientes.

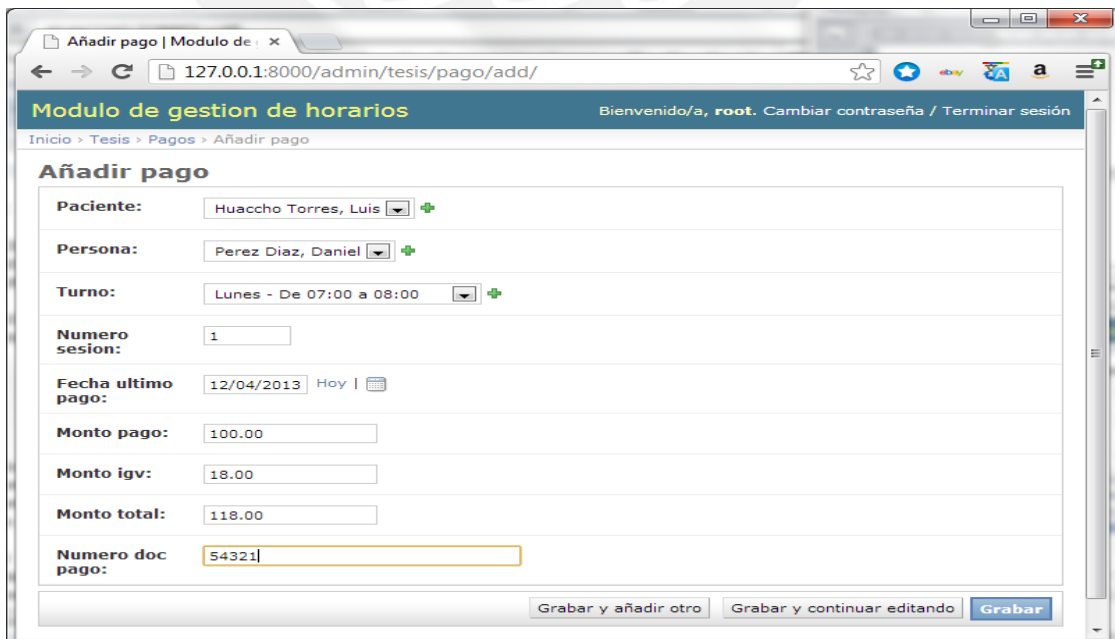


Figura 16. Pantalla de detalle de Pagos.



Índice de contenido

1. Pantalla de personal	2
2. Pantalla de detalle de terapia	5
3. Pantalla de detalle de turno	6
4. Pantalla de configuración de horarios	8
5. Pantalla de Generación de horarios	9
6. Pruebas de Casos de uso	10
6.1. Eliminación de terapias	10
6.2. Replanificación de carga laboral	11



1. Pantalla de personal

Condición de entrada	Clases válidas	Clases no válidas
Grupo ocupacional	1. Selección de médico 2. Selección de tecnólogo 3. Selección de otros	4. Vacío
Tipo documento	5. Selección de DNI 6. Selección de carnet de extranjería 7. Selección de otro	8. Vacío
Número documento	9. Cadena de 8 y 15 caracteres	10. Vacío
Apellido paterno	11. Cadena entre 1 y 100 caracteres	12. Vacío
Apellido materno	13. Cadena entre 1 y 100 caracteres	14. Vacío
Nombres	15. Cadena entre 1 y 100 caracteres	16. Vacío
Sexo	17. Selección de masculino 18. Selección de femenino	19. Vacío

Pruebas unitarias

Prueba 1	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Personal.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	<p>En la interfaz de entrada introducir los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Grupo ocupacional: Selección de Médico ▪ Tipo documento: Selección de DNI ▪ Número documento: Cadena de 8 y 15 caracteres ▪ Apellido paterno: Cadena entre 1 y 100 caracteres ▪ Apellido materno: Cadena entre 1 y 100 caracteres ▪ Nombres: Cadena entre 1 y 100 caracteres ▪ Sexo: Selección de masculino
Resultados Esperados:	Se muestra el mensaje: Se añadió con éxito el personal asistencial "xxx "

Prueba 2	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Personal.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	<p>En la interfaz de entrada introducir los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Grupo ocupacional: Selección de Médico ▪ Tipo documento: Selección de DNI ▪ Número documento: Vacío ▪ Apellido paterno: Cadena entre 1 y 100 caracteres ▪ Apellido materno: Cadena entre 1 y 100 caracteres ▪ Nombres: Cadena entre 1 y 100 caracteres ▪ Sexo: Selección de masculino
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

Prueba 3	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Personal.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	<p>En la interfaz de entrada introducir los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Grupo ocupacional: Selección de Médico ▪ Tipo documento: Selección de DNI ▪ Número documento: Cadena de 8 y 15 caracteres ▪ Apellido paterno: Vacío ▪ Apellido materno: Cadena entre 1 y 100 caracteres ▪ Nombres: Cadena entre 1 y 100 caracteres ▪ Sexo: Selección de masculino
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

Prueba 4	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Personal.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	<p>En la interfaz de entrada introducir los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Grupo ocupacional: Selección de Médico ▪ Tipo documento: Selección de DNI ▪ Número documento: Cadena de 8 y 15 caracteres ▪ Apellido paterno: Cadena entre 1 y 100 caracteres ▪ Apellido materno: Vacío ▪ Nombres: Cadena entre 1 y 100 caracteres ▪ Sexo: Selección de masculino
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

Prueba 5	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Personal.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	<p>En la interfaz de entrada introducir los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Grupo ocupacional: Selección de Médico ▪ Tipo documento: Selección de DNI ▪ Número documento: Cadena de 8 y 15 caracteres ▪ Apellido paterno: Cadena entre 1 y 100 caracteres ▪ Apellido materno: Cadena entre 1 y 100 caracteres ▪ Nombres: Vacío ▪ Sexo: Selección de masculino
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

2. Pantalla de detalle de terapia

Condición de entrada	Clases válidas	Clases no válidas
Descripción	1. Cadena entre 1 y 100 caracteres	2. Vacío
Recursos	3. Cadena entre 1 y 100 caracteres	4. Vacío

Pruebas unitarias

Prueba 1	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Terapia.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Descripción: Cadena entre 1 y 100 caracteres ▪ Recursos: Cadena entre 1 y 100 caracteres
Resultados Esperados:	Se muestra el mensaje: Se añadió con éxito la terapia "xxx "

Prueba 2	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Terapia.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Descripción: Vacío ▪ Recursos: Cadena entre 1 y 100 caracteres
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

Prueba 3	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Terapia.
Precondición:	Se ha accedido al sistema como usuario

Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Descripción: Cadena entre 1 y 100 caracteres ▪ Recursos: Vacío
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

3. Pantalla de detalle de turno

Condición de entrada	Clases válidas	Clases no válidas
Día	1. Selección de Lunes 2. Selección de Martes 3. Selección de Miércoles 4. Selección de Jueves 5. Selección de Viernes 6. Selección de Sábado 7. Selección de Domingo	8. Vacío 9. Otro valor
Hora inicio	10. Caracteres numéricos y dos puntos (:) de 5 caracteres.	11. Vacío 12. Caracteres del abecedario
Hora final	13. Caracteres numéricos y dos puntos (:) de 5 caracteres.	14. Vacío 15. Caracteres del abecedario

Pruebas unitarias

Prueba 1	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Turno.
Precondición:	Se ha accedido al sistema como usuario

Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Día: Selección de Lunes ▪ Hora inicio: Caracteres numéricos y dos puntos (:) de 5 caracteres. ▪ Hora final: Caracteres numéricos y dos puntos (:) de 5 caracteres.
Resultados Esperados:	Se muestra el mensaje: Se añadió con éxito el turno "xxx".

Prueba 2	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Turno.
Precondición:	Se ha accedido al sistema como usuario.
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Día: Selección de Lunes ▪ Hora inicio: Vacío ▪ Hora final: Caracteres numéricos y dos puntos (:) de 5 caracteres.
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio.

Prueba 3	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Turno.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Día: Selección de Lunes ▪ Hora inicio: Caracteres numéricos y dos puntos (:) de 5 caracteres. ▪ Hora final: Vacío
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

4. Pantalla de configuración de horarios

Condición de entrada	Clases válidas	Clases no válidas
Día	1. Selección de Lunes 2. Selección de Martes 3. Selección de Miércoles 4. Selección de Jueves 5. Selección de Viernes 6. Selección de Sábado 7. Selección de Domingo	8. Vacío 9. Otro valor
Hora inicio	10. Caracteres numéricos y dos puntos (:) de 5 caracteres.	11. Vacío 12. Caracteres del abecedario
Hora final	13. Caracteres numéricos y dos puntos (:) de 5 caracteres.	14. Vacío 15. Caracteres del abecedario

Pruebas unitarias

Prueba 1	
Objetivo Prueba:	Verificar que se ingrese los datos correctos de Turno.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Día: Selección de Lunes ▪ Hora inicio: Caracteres numéricos y dos puntos (:) de 5 caracteres. ▪ Hora final: Caracteres numéricos y dos puntos (:) de 5 caracteres.
Resultados Esperados:	Se muestra el mensaje: Se añadió con éxito el turno "xxx "

5. Pantalla de Generación de horarios

Condición de entrada	Clases válidas	Clases no válidas
Día	1. Selección de turnos del Lunes 2. Selección de turnos del Martes 3. Selección de turnos del Miércoles 4. Selección de turnos del Jueves 5. Selección de turnos del Viernes 6. Selección de turnos del Sábado 7. Todo vacío.	
Nro de Documento	8. Caracteres numéricos de Nro de documento válido asociado a un acto médico.	9. Vacío

Pruebas unitarias

Prueba 1	
Objetivo Prueba:	Verificar los parámetros necesarios para la realización del cronograma de terapias que se desea generar de acuerdo al acto médico relacionado al Nro de Documento.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> ▪ Día: Seleccionar los turnos que se desea por día para la elaboración de la terapia ▪ Nro de Documento válido asociado a un acto medico
Resultados Esperados:	Se genera un cronograma con fechas y turnos de terapias donde por cada día de los escogidos solo hay un turno escogido. Y un tecnólogo asociado a un turno de terapia.

Prueba 2	
Objetivo Prueba:	Verificar los parámetros necesarios para la realización del cronograma de terapias que se desea generar de acuerdo al acto médico relacionado al Nro de Documento.

Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> • Día: Seleccionar los turnos que se desea por día para la elaboración de la terapia • Nro de Documento vacío
Resultados Esperados:	Se muestra el mensaje: Por favor, corrija el siguiente error: Este campo es obligatorio

Prueba 3	
Objetivo Prueba:	Verificar los parámetros necesarios para la realización del cronograma de terapias que se desea generar de acuerdo al acto médico relacionado al Nro de Documento.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	En la interfaz de entrada introducir los siguientes datos: <ul style="list-style-type: none"> • Día: Todo vacío • Nro de Documento válido asociado a un acto medico
Resultados Esperados:	Se genera un cronograma con fechas y turnos de terapias del acto médico donde por cada día solo hay un turno escogido. Y un tecnólogo asociado a un turno de terapia.

6. Pruebas de Casos de uso

6.1. Eliminación de terapias

Prueba	
Objetivo Prueba:	Verificar la eliminación de terapias después de tres inasistencias.
Precondición:	Se ha accedido al sistema como usuario
Descripción de la prueba:	Buscar las terapias de un paciente y borrar la asistencia de tres terapias, seleccionar eliminación de terapias.
Resultados Esperados:	Se eliminan todas las terapias que le quedan al paciente después de la tercera no asistida.

6.2. Replanificación de carga laboral.

Prueba	
Objetivo	Verificar la replanificación de terapias después de varias
Prueba:	Eliminaciones de cronogramas de terapia, se mantenga igual carga laboral diaria.
Precondición:	Se ha accedido al sistema como usuario. No hay información en el sistema. Se crean 3 terapistas con los mismos turnos de trabajo con la misma preferencia por determinada enfermedad, se generan 12 cronogramas de terapia seleccionando en el sistema los turnos en días iguales para la generación de cada cronograma.
Descripción de la prueba:	Replanificar los cronogramas de terapia con igual carga laboral para los terapistas sin modificar los turnos de terapia establecidos para cada paciente sino solo al terapeuta.
Resultados Esperados:	Después de eliminar todas las terapias asignadas para un terapeuta. Se replanifica la carga laboral de cada terapeuta tendrán aproximadamente la misma cantidad de terapias asignados como 16 o 17.