

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e ingeniería

Ingeniería en Computación

**Expediente Clínico Electrónico.
ECE**

Reporte Final de PT02

Alumno:

**Castrejon Morales Daniel
204358724**

Asesorado por:

**M.en C. Rafaela Blanca Silva López
No. Económico: 17114**

**Trimestre 10P
09 de Septiembre de 2010**

Índice

Índice.....	2
Ilustraciones.....	5
Introducción.....	6
Objetivos del Sistema.....	6
Objetivo general:.....	6
Objetivos Específicos:.....	6
Antecedentes:.....	7
Justificación:.....	9
Descripción Técnica:.....	10
Especificación Técnica:.....	11
Licencia de Software Libre – Creative Commons.....	13
Casos De Uso.....	14
Casos de Uso del Usuario.....	14
Caso de uso 1: Autenticación.....	15
Caso de uso 2: Selección de pacientes.....	15
Caso de uso 3: Alta de Pacientes.....	16
Caso de uso 4: Consulta de notas médicas.....	17
Caso de uso 5: Agregar notas médicas.....	18
Caso de uso 6: Agregar recomendaciones médicas.....	19
Caso de uso 7: Consulta de Recomendaciones Médicas.....	20
Caso de uso 8: Imprimir Notas Médicas.....	20
Caso de uso 9: Imprimir recomendaciones médicas.....	21
Caso de uso 10: Agregar Comentarios.....	22
Caso de uso 11: Consultar comentarios.....	23
Base De Datos.....	24
Diagrama Entidad – Relación.....	24
Esquema Físico de la BD.....	25
Implementación.....	67
ProyectoTerminal/ece/data.....	67
ComentarioDTO.java.....	67
ConsultorioDTO.java.....	68
DetalleRecommendacionesDTO.java.....	69
EscolaridadDTO.java.....	70
EspecialidadDTO.java.....	71
EstadoCivilDTO.java.....	72
FamiliarResponsableDTO.java.....	72
NacionalidadDTO.java.....	74
NotaAnestesiologiaDTO.java.....	75
NotaAnestesiologiaJoinDTO.java.....	75
NotaDTO.java.....	78
NotaIngresoDTO.java.....	80
NotaIngresoJoinDTO.java.....	81
NotaInterconsultaDTO.java.....	82
NotaInterconsultaJoinDTO.java.....	84
NotaJoinDTO.java.....	85
OcupacionDTO.java.....	87
PacienteDTO.java.....	88
PacienteHospitalizadoDTO.java.....	92

PacienteHospitalizadoJoinDTO.java	94
PacienteJoinDTO.java	96
RecomendacionesDTO.java	102
RecomendacionesJoinDTO.java	103
ReligionDTO.java	105
RolDTO.java	105
SexoDTO.java	106
SignosVitalesDTO.java	107
TurnoDTO.java	109
UnidadDTO.java	109
UsuarioDTO.java	111
UsurioJoinDTO.java	115
ValidarUsuarioDTO.java	119
ProyectoTerminal/ece/business	120
 AdministradorComentario.java	120
 AdministradorConsultorio.java	120
 AdministradorDetalleRecomendaciones.java	121
 AdministradorEscolaridad.java	121
 AdministradorEspecialidad.java	122
 AdministradorEstadoCivil.java	122
 AdministradorFamiliarResponsable.java	123
 AdministradorNacionalidad.java	123
 AdministradorNotaAnestesiologia.java	124
 AdministradorNotaIngreso.java	124
 AdministradorNotaInterconsulta.java	124
 AdministradorOcupacion.java	125
 AdministradorPacienteHospitalizado.java	125
 AdministradorPacientes.java	126
 AdministradorRecomendaciones.java	127
 AdministradorReligion.java	127
 AdministradorRoles.java	128
 AdministradorSexo.java	128
 AdministradorSignosVitales.java	129
 AdministradorTurno.java	129
 AdministradorUnidad.java	130
 AdministradorUsuario.java	130
.....	131
ProyectoTerminal/ece/business/local	132
 AdministradorComentarioImpl.java	132
 AdministradorConsultorioImpl.java	133
 AdministradorDetalleRecomendacionesImpl.java	134
 AdministradorEscolaridadImpl.java	135
 AdministradorEspecialidadImpl.java	136
 AdministradorEstadoCivil.java	138
 AdministradorFamiliarResponsableImpl.java	139
 AdministradorNacionalidadImpl.java	140
 AdministradorNotaAnestesiologiaImpl.java	141
 AdministradorNotaIngresoImpl.java	142
 AdministradorNotaInteconsultaImpl.java	143

<u>AdministradorOcupacionImpl.java</u>	144
<u>AdministradorPacienteHospitalizadoImpl.java</u>	145
<u>AdministradorPacienteimpl.java</u>	148
<u>AdministradorRecomendacionesImpl.java</u>	150
<u>AdministradorReligionImpl.java</u>	151
<u>AdministradorRolesImpl.java</u>	153
<u>AdministradorSexoImpl.java</u>	154
<u>AdministradorSignosVitales.java</u>	155
<u>AdministradorTurnoImpl.java</u>	156
<u>AdministradorUnidad.java</u>	157
<u>AdministradorUsuarioImpl.java</u>	159
<u>FiltroPacientes.java</u>	161
<u>FiltroUsuarios.java</u>	162
<u>ProyectoTerminal/ece/actions</u>	162
<u>ComentarioAction.java</u>	162
<u>DetalleRecomendacionesAction.java</u>	164
<u>FamiliarAction.java</u>	166
<u>LoginAction.java</u>	167
<u>NotaAnestesiologiaAction.java</u>	169
<u>NotaIngresoAction.java</u>	173
<u>NotaInterconsultaAction.java</u>	177
<u>NotasAction.java</u>	181
<u>PacienteAction.java</u>	183
<u>PacienteHospitalizadoActiona.java</u>	190
<u>RecomendacionesAction.java</u>	196
<u>SignosVitalesAction.java</u>	201
<u>UsuarioAction.java</u>	203
<u>Pruebas y Liberación</u>	213
.....	213
<u>Referencias</u>	214

Ilustraciones

Ilustración 1: Casos de Uso Usuario.....	14
Ilustración 2: Diagrama Entidad – Relación.....	24

Introducción

En la mayoría de los hospitales, el proceso de elaboración del expediente clínico, se lleva a cabo de manera manual. A continuación describimos el procedimiento: El paciente llega a la clínica y/u hospital, el médico toma sus datos personales y analiza los signos vitales y síntomas del paciente mediante una revisión e interrogatorio, respectivamente. Con los datos obtenidos se llena una primera nota del expediente clínico, y se realiza el mismo procedimiento de llenado en cada nota. Es decir, en cada nota nueva que se realice, sea igual a la primera o diferente se tiene que volver a poner los datos personales y signos vitales y síntomas, haciendo lento el servicio de atención a los pacientes. Lo mismo pasa con la elaboración de recetas de los pacientes.

Cabe mencionar que el expediente clínico está conformado por varias notas. Esto está especificado en la norma NOM-168 SSA1-1998[], del expediente clínico.

Objetivos del Sistema.

Objetivo general:

Realizar un sistema que permita la automatización en la elaboración de expedientes clínicos electrónicos con las especificaciones de Norma Oficial Mexicana NOM-168 SSA1-1998 [I], del Expediente Clínico."

Objetivos Específicos:

1. Analizar las necesidades del personal de una clínica y/u hospital en la elaboración de expedientes clínicos.
2. Investigar herramientas y tecnologías para la elaboración de expedientes clínicos electrónicos.
3. Diseñar el sistema para automatizar la elaboración de Expedientes Clínicos Electrónicos, con las especificaciones de la norma NOM-168, del expediente clínico de la SSA² que incluya los siguientes módulos:
 1. Registro de Pacientes.
 2. Registro de Personal.
 3. Consulta de expedientes clínicos electrónicos del paciente.
 4. Elaboración de recetas.
 5. Foro.
- MI. Diseñar el flujo de trabajo electrónico para la elaboración de expedientes clínicos electrónicos.
- MII. Implementar el sistema para la automatización de expedientes clínicos electrónicos con las especificaciones de la norma NOM-168, del expediente clínico.

² Secretaria de Salud.

- MIII. Realizar las pruebas para la liberación del sistema de automatización en la elaboración de expedientes clínicos electrónicos con las especificaciones de la norma NOM-168, del expediente clínico.
- MIV. Realizar la documentación del sistema de automatización en la elaboración de expedientes clínicos electrónicos con las especificaciones de la norma NOM-168, del expediente clínico.
- MV. Realizar manuales de usuario.

Antecedentes:

En la mayoría de los hospitales, el proceso de elaboración del expediente clínico, se lleva a cabo de manera manual. A continuación describimos el procedimiento: El paciente llega a la clínica y/u hospital, el médico toma sus datos personales y analiza los signos vitales y síntomas del paciente mediante una revisión e interrogatorio, respectivamente. Con los datos obtenidos se llena una primera nota del expediente clínico, y se realiza el mismo procedimiento de llenado en cada nota. Es decir, en cada nota nueva que se realice, sea igual a la primera o diferente se tiene que volver a poner los datos personales y signos vitales y síntomas, haciendo lento el servicio de atención a los pacientes. Lo mismo pasa con la elaboración de recetas de los pacientes.

Cabe mencionar que el expediente clínico está conformado por varias notas. Esto está especificado en la norma NOM-168, del expediente clínico.

En la UAM-A se encontró el siguiente proyecto:

- 3. “SISTEMA DE GESTIÓN DE INFORMACIÓN DE PACIENTES, PARA UNA CLÍNICA HOMEOPÁTICA”. El proyecto fue desarrollado por la estudiante de la licenciatura en ingeniería en computación, Manzanares Soriano Selene. El sistema permite registrar nuevos pacientes, programar citas, especificar antecedentes médicos de una persona, así como compartir el expediente médico del paciente entre varios médicos. Este sistema de gestión de información opera en un ambiente WEB² y maneja el expediente clínico electrónico para compartirse con los médicos que tengan acceso a dicho sistema [II].

Una búsqueda en la WEB tuvo como objetivo encontrar sistemas similares al que se plantea en este proyecto. A continuación se describen algunos sistemas, que ofrecen algunas de las funcionalidades que estamos planteando en este proyecto:

- 4. **VistA IMSS**³: Sistema que permite gestionar los expedientes clínicos de los pacientes y toda persona que esté registrada al seguro social. Se pueden realizar todas las notas que indica la norma NOM-168, del expediente clínico. Actualmente está instalado en algunos hospitales, y

²Sistema de documentos interconectados por enlaces de hipertexto, que se ejecutan en Internet.

³ Software utilizado por el Instituto Mexicano del Seguro Social.

centros médicos que pertenecen al IMSS⁴, se pretende cubrir a nivel nacional en unos cuantos años más [III].

5. **SAAM**: Solución *online* que permite administrar y gestionar la información médica del paciente, con total seguridad, desde cualquier lugar donde se encuentre el médico. [IV].
Permite a al médico tener un control de su agenda, la gestión de pacientes y expedientes clínicos, la configuración y emisión de documentos o plantillas, el manejo de catálogos, exámenes y vacunas y algunos reportes generales como agendas e historiales clínicos.
6. **Medica 1.0**: es un sistema de información personal, que puede usarse para almacenar información médica de personas, expedientes médicos hechos por anteriores doctores, y la información de las aseguradoras médicas [V].
Los doctores pueden usar esta aplicación para tener a la mano y en todo momento los informes de cada uno de sus pacientes, con la prescripción médica realizada hasta el momento, la cual puede ser útil para futuros diagnósticos. Además, puede usarse para configurar avisos de futuras citas concertadas con los pacientes.
7. **Sistema del ISSSTE⁵ Vía Web**: Funciona actualmente en las clínicas del ISSSTE permitiendo programar citas de acuerdo a los horarios disponibles para el día presente (fichas) o para días posteriores [VI].

En esta investigación, constatamos que:

El "SISTEMA DE GESTIÓN DE INFORMACIÓN DE PACIENTES, PARA UNA CLÍNICA HOMEAPÁTICA" :

1. Permite compartir el expediente clínico electrónico con médicos que tengan acceso al sistema.
2. Está planteado en una solución WEB y se puede acceder con cualquier servidor de internet.
3. No realiza todas las notas que conforman el expediente clínico, como lo especifica la norma NOM-168, del expediente clínico.

El sistema VistA IMSS:

2. Puede realizar todas las notas que conforman el expediente clínico, como lo especifica la norma NOM-168, del expediente clínico, pero presenta anomalías. Al momento de realizar una nueva nota, se pueden dejar sin actualizar algunos campos o usar los datos de una nota anterior, aunque estos datos hayan sido tomados uno o varios días atrás.
3. No está en un ambiente WEB, sino, que se utiliza un servidor y se accede a él por medio de un programa que está instalado en una máquina (cliente) que se encuentra en el hospital y/o la institución.

Los demás sistemas mencionados:

⁴ Instituto Mexicano del Seguro Social.

⁵ Sistema utilizado por Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado.

2. No cuentan con una funcionalidad que permita compartir el expediente clínico electrónico.
3. No operan bajo la WEB, sino que deben ser instalados en cada una de las máquinas personales del médico o clínica.
4. Son software propietarios, por lo tanto, no se pueden descargar y conocer su funcionalidad.

El sistema que se está plateando incluirá lo siguiente:

1. Todas las notas que conforman al expediente clínico, como lo especifica la norma NOM-168, del expediente clínico.
2. Toda nota que se elabore debe actualizar todos los datos que conforman dicha nota, es decir, la nota no se guardará hasta que el médico actualice todos los datos y de manera automática se llenarán datos personales de la institución, del médico, el paciente y demás datos que se repitan.
3. Será instalado en un servidor y se podrá tener acceso a él mediante la WEB, con cualquier servidor de internet y que tenga clave de acceso.
4. Nuevas funcionalidades, como es el módulo de elaboración de recetas de manera automatizada y un foro donde se pueden realizar comentarios y/o sugerencias sobre algún paciente, para algún otro médico que tenga acceso al expediente clínico del paciente.

Justificación:

Consideramos que el sistema de automatización en la elaboración de expedientes clínicos presenta varias ventajas para sus usuarios:

Primero, al médico no le llevará demasiado tiempo la elaboración de las diferentes notas, ya que los datos que se repiten en las notas serán llenados de manera automática al crear una nueva nota.

Segundo, el expediente clínico estará disponible para cualquier médico que tenga clave de acceso al sistema y al expediente clínico del paciente, así que, el expediente clínico del paciente podrá ser observado por cualquier médico, en cualquier otro hospital que pertenece a la institución.

Tercero, el sistema que funcione bajo ambiente Web:

8. No es necesaria la instalación en cada una de las computadoras personales.
9. El sistema puede residir en uno o varios servidores, proporcionando su servicio a varios usuarios simultáneamente. Esto reduce significativamente los costos de memoria y procesamiento.
10. Los usuarios pueden acceder desde cualquier computadora personal con un servidor de internet, autenticando su acceso.

11. El sistema puede funcionar independientemente de las plataformas Windows², Unix³ y Linux⁴.

Es importante mencionar que el sistema de automatización en la elaboración del expediente clínico, que se está proponiendo, no depende de los sistemas mencionados en la sección anterior. Estos son sistemas propietarios, por lo tanto sus códigos fuentes no están disponibles, exceptuando el proyecto realizado en esta casa de estudios. De manera que los módulos planeados en los objetivos se desarrollarán completamente para estar disponibles para esta casa de estudios.

La realización de este proyecto requiere conocimientos, relativamente avanzados de las siguientes ramas: ⁵

- Sistemas de Bases de Datos Relacionales (RDBMS).
- Redes.
- Sistemas Operativos.
- Sistemas distribuidos.
- Metodologías de análisis y diseño de sistemas de información.
- Ingeniería de software.
- Programación JSP⁶ (Temas Selectos de computación, impartida en el trimestre 09I).

Estos cursos son impartidos en nuestra casa de estudios, por lo cual, se considera que este proyecto no podría ser realizado por una persona que no sea ingeniero en computación.

En un desarrollo posterior, se podrían incluir otros módulos, como uno, que incluya notas de laboratorio, trabajo social, estadística de tipo administrativo, etc.

También podría desarrollarse la portabilidad del expediente, esto es, el paciente tiene en su poder su expediente clínico y cada vez que acuda a asistencia médica, puede consultar y/o actualizar su expediente clínico.

Descripción Técnica:

El sistema de automatización en la elaboración de expedientes clínicos se plantea como un sistema que mejorará considerablemente la atención al

² Familia de sistemas operativos desarrollados y comercializados por Microsoft.

³ Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

⁴ Es un término genérico para referirse a sistemas operativos similares a Unix basados en el núcleo de Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; normalmente todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la Licencia Pública General de GNU (GNU GPL) y otras licencias libres.

⁵ Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

⁶ Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

paciente que lo requiera, ya que será más rápida y se podrá dar seguimiento más oportuno al paciente al tener su expediente clínico disponible cada vez que se requiera.

Se plantea desarrollar el sistema utilizando plataforma JAVA², con el fin de que pueda ser instalado en servidores con características de sistema operativo y capacidad de procesamiento diferentes.

El sistema contará con los siguientes bloques o módulos:

3. Manejo de expedientes clínicos electrónicos
4. Manejo de Pacientes.
5. Manejo de personal.
6. Elaboración de recetas.
7. Foro

Manejo de expedientes clínicos electrónicos: Esta interfaz o módulo permitirá al médico realizar todas las notas que requiera el expediente clínico del paciente, así como observar notas anteriores del paciente.

Manejo de pacientes: Dentro de este módulo se dará de alta a los nuevos usuarios que requieran del servicio médico que ofrece el hospital y/o institución.

Manejo de personal: Dentro de este bloque se manejarán todos los datos del personal que tienen relación directa con el expediente clínico, en especial los médicos que laboran en el hospital y/o institución. También permitirá dar de alta nuevo personal que contrate el hospital o institución.

Elaboración de recetas: Dentro de este bloque se elaboran de manera automatizada las recetas del paciente para su impresión.

Foro: Este bloque permitirá tener una comunicación asíncrona con el personal sobre algún expediente, dejar comentarios y/o sugerencias de algún expediente especial del paciente.

El sistema se desarrollará utilizando plataforma abierta (Eclipse [VII]) sobre sistema operativo LINUX [VIII], ya que permite realizar el desarrollo sobre la parte que se desarrollará en plataforma Java como en la parte de interfase de usuario. También permite integrar servidores de aplicaciones abiertas, así como bases de datos relacionales, en general.

Especificación Técnica:

Se plantea utilizar el estándar *J2EE 1.4*² como base, con el fin de poder ser montado a futuro sobre diferentes servidores de portales.

² Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

² Java 2 Platform Enterprise Edition. Plataforma de programación.

El protocolo de comunicación que se utilizará entre clientes y servidor será HTTP³, debido a que es una aplicación que requiere ser ejecutada desde la red, para poder dar soporte a los diferentes clientes que pueden encontrarse en diferentes lugares de la República Mexicana.

Tanto los clientes como el servidor, podrán ser ejecutados sobre sistemas operativos distintos, con el fin de tener amplias posibilidades de implementación.

Las listas de servicios o reportes que se generen podrán ser impresos utilizando formatos comunes, como: PDF⁴ y Open Office [IX].

Dentro del servidor se tendrá ejecutando un *Java Runtime Environment V5.0*⁵, debido a que es una plataforma portable, abierta y ampliamente utilizada para este tipo de aplicaciones [X].

Para dar soporte a la aplicación Web, se manejará un contenedor de *struts*⁶. Dentro de dicho contenedor se ejecutarán los distintos módulos Web que compondrán a la aplicación. Este contenedor puede dar soporte a diferentes componentes Web como son *struts* o *servlets*⁷, por lo que en algún momento la comunicación entre ellos se realizará fácilmente.

El acceso a la base de datos se realizará con *Hibernate*⁸ versión 3.0 y el sistema se diseñará de tal manera que se pueda comunicar con cualquier base de datos relacional, que soporta el estándar SQL⁹ 92. Esto es con el fin de tener una gran variedad de opciones de implementación cuando el sistema se desee implementar en otra institución y/u hospital [XI].

La administración de los servicios de acceso a base de datos, las transacciones, la comunicación entre componentes y el ciclo de vida se llevará a cabo por medio de Struts Framework¹⁰.

La generación del XML¹¹ de entrada para la interfaz de usuario se programará con JSP¹² versión 2.1 y librerías estándares como JSTL¹³ y algunas otras etiquetas desarrolladas para este proyecto en particular.

Para la generación de reportes en Calc, se utilizará la librería abierta JXLS¹⁴, la cual internamente está basada en Jakarta POI¹⁵, que es el estándar de facto

³ Protocolo de transferencia de texto (*HyperText Transfer Protocol*).

⁴ **PDF** (acrónimo del inglés *Portable Document Format*, formato de documento portátil). Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

⁵ Entorno en tiempo de ejecución Java.

⁶ Componentes modulares de interfaz de usuario gestionadas y visualizadas en un portal Web.

⁷ Pequeña aplicación Java que se ejecuta en un servidor web.

⁸ Herramienta de Mapeo objeto-relacional para la plataforma de Java.

⁹ Lenguaje de consulta estructurado (*Structured Query Language*).

¹⁰ *Framework* de código abierto de desarrollo de aplicaciones para la plataforma Java.

¹¹ Lenguaje de marcas extensible (*Extensible Markup Language*).

¹² *Java Server Pages*. Tecnología Java que permite generar contenido dinámico para web.

¹³ *JavaServer Pages Standar Tag Libra*. Es un componente de Java EE.

¹⁴ Biblioteca de Java para la generación de archivos de Excel XLS.

¹⁵ API para Java que permite manipular diversos formatos de Microsoft (Excel, Word, Power Point).

para el manejo de Calc en Java. Para la generación de reportes en PDF, se hará uso de la familia de estándares XSL¹⁶ (XSL-T¹⁷, XSL-FO¹⁸, XPath¹⁹). Las librerías en Java que se utilizarán para esto son Apache²⁰ Xalan²¹, Apache [X] FOP²² y el API estándar de Java JAXP²³

El proyecto se dará por concluido cuando se tengan en funcionamiento los módulos siguientes:

1. Manejo de Pacientes.
2. Manejo de personal.
3. Manejo de expedientes clínicos electrónicos
4. Elaboración de recetas.
5. Foro

Licencia de Software Libre – Creative Commons.

Reconocimiento (Attribution): En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

Compartir Igual (Share alike): La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.



Reconocimiento - CompartirIgual (by-sa): Se permite la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

¹⁶ Lenguaje de hojas extensibles de estilo (*Extensible Stylesheet Language*).

¹⁷ Lenguaje de hojas extensibles de transformación (*Extensible Stylesheet Language Transformations*).

¹⁸ Lenguaje de hojas extensibles de formateo de objetos.

¹⁹ *XML Path lenguaje*, sintaxis para acceder o referirse a porciones de un documento XML.

²⁰ Servidor http de dominio público el cual está basado en el sistema operativo Linux.

²¹ Procesador XLST para transformar documentos XML en HTML.

²² *Formatting Objects Processor*. Aplicación Java.

²³ *Java Api for XML Processing*. Sirve para la manipulación y tratamiento de archivos XML.

Casos De Uso

Casos de Uso del Usuario

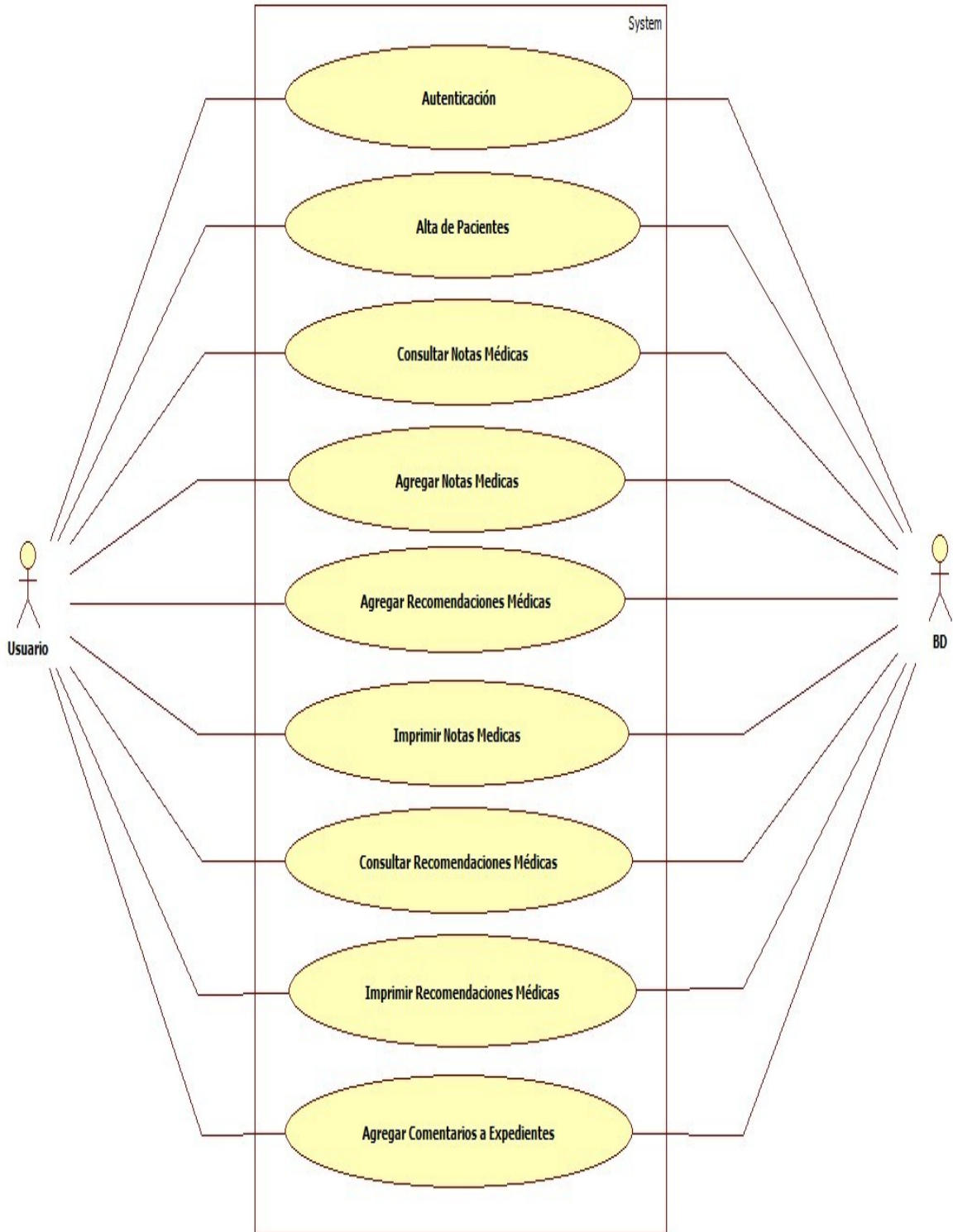


Ilustración 1: Casos de Uso Usuario

Caso de uso 1: Autenticación.

UC1: Autenticación.
Actor Principal: Usuario
Precondiciones: <ul style="list-style-type: none">• El actor debe tener acceso al Sistema.• El actor debe estar Registrado como Usuario del Sistema.
Post-condiciones: <ul style="list-style-type: none">• El actor podrá consultar y realizar expedientes clínicos.• El actor podrá agregar pacientes a la base de datos.• El actor podrá consultar y realizar recomendaciones médicas.• El actor podrá consultar y realizar comentarios a los expedientes clínicos.
Escenario Principal: <ol style="list-style-type: none">1. El Sistema despliega la Pantalla de Bienvenida.2. El Usuario ingresa su nombre de usuario y contraseña.3. El Usuario presiona aceptar.4. El Sistema verifica la información ingresada.5. EL Sistema despliega la Pantalla principal de Usuarios.
Flujos Alternativos: <ol style="list-style-type: none">3.1. El Usuario presiona limpiar.3.2. El Sistema limpia las casillas y se regresa al paso 2.4.1. Si los datos son incorrectos se informa al usuario y regresa al paso 1.

Caso de uso 2: Selección de pacientes.

UC2: Selección de pacientes.
Actor Principal: Usuario
Precondiciones: <ul style="list-style-type: none">• El actor debe estar autenticado.• El actor debe saber acerca del paciente que busca (nombre o numero de seguro).
Post-condiciones: <ul style="list-style-type: none">• El actor podrá agregar o consultar el expediente clínico.• El actor podrá agregar o consultar las recomendaciones médicas.• El actor podrá agregar o consultar comentarios acerca del paciente seleccionado.
Escenario Principal: <ol style="list-style-type: none">1. El Sistema despliega la pantalla principal de Usuarios.2. El Usuario selecciona “Buscar Pacientes”.3. El Sistema despliega la pantalla de Buscar Pacientes.4. El Usuario ingresa el número de seguridad social y presiona Buscar5. El Sistema despliega la pantalla de información del paciente y pregunta al usuario que confirme si es el paciente buscado.6. El Usuario confirma que es el paciente buscado.7. El Sistema despliega el menú principal de los pacientes.
Flujos Alternativos: <ol style="list-style-type: none">4.1 El Usuario ingresa el nombre del paciente y presiona buscar. Se ejecuta desde el paso 5.5.1 Si el paciente no existe, el Sistema informa al usuario y le pregunta si lo quiere

registrar.

5.2 Si el Usuario selecciona que si, se ejecuta el Caso de Uso agregar pacientes.

5.3 En caso contrario El Sistema despliega la pantalla principal de usuarios.

6.1 Si no es el paciente buscado, El Usuario selecciona buscar otro paciente.

Se repite desde el paso 1.

Caso de uso 3: Alta de Pacientes.

UC3: Alta de Paciente
Actor Principal: Usuario
Precondiciones: <ul style="list-style-type: none">• El actor debe estar autenticado en el Sistema.• El actor debe poder realizar altas de pacientes al Sistema• El paciente no debe estar registrado en el Sistema.
Post-condiciones: <ul style="list-style-type: none">• El actor podrá consultar todo lo relacionado con el paciente en el Sistema.• El actor podrá agregar notas al expediente clínico del nuevo paciente.• El actor podrá agregar recomendaciones médicas al nuevo paciente.• El actor podrá agregar comentarios al expediente clínico al nuevo paciente.
Escenario Principal: <ol style="list-style-type: none">1. El Sistema despliega el aviso “¿Quiere agregar un nuevo Paciente?, SI o No”2. El Usuario Presiona SI.3. El Sistema despliega la plantilla para alta de pacientes.4. El Usuario ingresa los datos del paciente.5. El Usuario presiona continuar.6. El Sistema verifica que los datos ingresados sean los correctos y los necesarios.7. El Sistema guarda los datos en la base de datos.8. EL Sistema Despliega el mensaje “Se realizo con éxito la alta del paciente, ¿Qué desea realizar?, dar de alta un nuevo paciente SI o No”.9. El Usuario presiona SI, se repite desde el paso 3.
Flujos Alternativos: <ol style="list-style-type: none">2.1. El Usuario presiona NO.2.2. El Sistema Despliega la pantalla Principal.6.1. El Sistema encuentra un error en los datos ingresados.6.2. El Sistema le informa al Usuario del error y pide que reingrese los datos erróneos y/o los datos faltantes.6.3. El sistema despliega la plantilla para alta de pacientes.8.1. El Usuario presiona NO.8.2. El Sistema despliega la pantalla Principal.

Caso de uso 4: Consulta de notas médicas.

UC4: Consulta de notas médicas.
Actor Principal: Usuario
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado en el Sistema. • El actor debe poder hacer consultas de pacientes en el Sistema • El paciente debe estar registrado en el Sistema.
<p>Post-condiciones:</p> <ul style="list-style-type: none"> • El actor podrá observar todas las notas médicas del paciente. • El actor podrá imprimir notas médicas del paciente. • El actor podrá consultar y realizar recomendaciones médicas al paciente. • El actor podrá consultar y realizar comentarios al expediente clínico del paciente.
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla de menú principal de paciente. 2. El Usuario selecciona agregar o consultar expediente clínico. 3. El Sistema despliega la pantalla de agregar o consultar notas médicas. 4. El Usuario selecciona consultar una nota ya realizada. 5. El Sistema despliega la plantilla con los datos guardados en la nota médica. 6. El Usuario selecciona unas de las diferentes opciones. <ol style="list-style-type: none"> 6.1 El Usuario Selecciona imprimir la nota médica seleccionada. 6.2 El Usuario Selecciona consultar o agregar recomendaciones médicas al paciente. 6.3 El Usuario selecciona consultar o realizar comentarios al expediente clínico del paciente. 6.4 El Sistema repite desde el paso 9. 7. El Usuario puede seleccionar otra nota o regresar al menú principal. 8. El Sistema repite desde el paso 10.
<p>Flujos Alternativos:</p> <ol style="list-style-type: none"> 4.1. Si no se encuentra el paciente, el Sistema le indica al Usuario, que si desea a darlo de alta y se ejecuta el caso de uso “Alta de Pacientes”. 6.1. Si el paciente no es el que se esta buscando, el Usuario selecciona buscar otro paciente. 6.2 El Sistema regresa desde el paso 3. <ol style="list-style-type: none"> 12.1.1. El Sistema manda a impresión la nota médica seleccionada. 12.2.1. El Sistema despliega la pantalla de consulta o realizar recomendaciones médicas al paciente, y se ejecuta el caso de uso de consulta o realizar recomendaciones médicas. 12.3.1. El Sistema despliega la pantalla de consulta o realizar comentarios al expediente clínico del paciente, y se ejecuta el caso de uso de consulta o realizar comentarios al expediente clínico del paciente.

Caso de uso 5: Agregar notas médicas.

UC5: Agregar notas médicas
<p>Actor Principal: Usuario</p>
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado. • El paciente debe estar registrado. • El actor debe escoger un paciente. • El actor debe poder agregar notas médicas al paciente.
<p>Post-condiciones:</p> <ul style="list-style-type: none"> • El actor podrá imprimir las nuevas notas médicas. • El actor podrá consultar las nuevas notas médicas.
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla de menú principal del paciente. 2. El Usuario Selecciona agregar o consultar notas médicas. 3. El Sistema despliega la pantalla agregar o consultar notas medicas. 4. El Usuario selecciona una nueva nota médica. 5. El Sistema pregunta al usuario que tipo de nota desea realizar. 6. El Usuario selecciona la nota médica a realizar. 7. El Sistema despliega la pantalla correspondiente a la plantilla de la nota médica seleccionada. 8. El Usuario llena el formulario y presiona guardar y firmar. <ol style="list-style-type: none"> 8.1. El Usuario podrá guardar temporalmente el formulario al presionar guardar sin firmar. 8.2. El Sistema guarda temporalmente el formulario y despliega la pantalla “agregar o consultar notas médicas”. 9. El Sistema valida los datos ingresados por el usuario, guarda la nota médica en la base de datos y despliega la pantalla “agregar o consultar notas medicas”.
<p>Flujos Alternativos:</p> <ol style="list-style-type: none"> 7.1 El usuario puede cancelar la nota y seleccionar otra. 7.2 El Sistema repite desde el paso 2. 8.1.1 El Usuario puede cancelar la nota siempre y cuando no la halla firmado. 8.1.2 El Sistema despliega la pantalla “agregar o consultar notas médicas”. 8.2.1 El Usuario puede cancelar la nota siempre y cuando no la halla firmado. 8.2.2 El Sistema despliega la pantalla “agregar o consultar notas médicas”. 9.1 Si los datos son incorrectos el Sistema despliega un mensaje indicando el error y pide al usuario que los corrija. 9.2 Se repite desde el paso 8.

Caso de uso 6: Agregar recomendaciones médicas

UC6: Agregar recomendaciones médicas
Actor Principal: Usuario
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado. • El paciente debe estar registrado. • El actor debe escoger un paciente. • El actor debe poder agregar recomendaciones médicas al paciente.
<p>Post-condiciones:</p> <ul style="list-style-type: none"> • El actor podrá imprimir las recomendaciones médicas. • El actor podrá consultar las nuevas recomendaciones médicas.
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla de menú principal del paciente. 2. El Usuario selecciona consultar o agregar recomendación médica. 3. El Sistema despliega la pantalla de consulta o agregar recomendación médica. 4. El Usuario selecciona agregar una nueva recomendación médica. 5. El Sistema despliega el formulario de llenado de recomendación médica. 6. El Usuario llena el formulario y presiona aceptar. 7. El Sistema verifica y guarda los datos ingresados y despliega la información contenida en la recomendación médica y pregunta al usuario si desea imprimir o terminar. 8. El usuario presiona imprimir la recomendación médica. 9. Se ejecuta el caso de uso imprimir recomendación médica.
<p>Flujos Alternativos:</p> <ol style="list-style-type: none"> 6.1. El Usuario puede guardar temporalmente la recomendación médica al presionar guardar sin firmar. 6.2. El Sistema guarda temporalmente la recomendación médica y despliega la pantalla de consulta o agregar recomendación médica. 6.3. El Usuario puede cancelar el llenado de la recomendación médica al presionar cancelar. 6.4 El Sistema despliega la pantalla de consulta o agregar recomendación médica. Se repite desde el paso 3. 7.1. El Sistema encuentra errores en el llenado del formulario, le informa al usuario y despliega nuevamente el formulario de llenado de recomendación médica. 7.2. El Usuario llena los campos faltantes o erróneos y vuelve presionar guardar. Se repite desde el paso 7. 8.1. El Usuario presiona terminar. 8.2. El Sistema despliega la pantalla de consulta o agregar recomendación médica. Se repite desde el paso 3.

Caso de uso 7: Consulta de Recomendaciones Médicas

UC7: Consulta de recomendaciones médicas.
Actor Principal: Usuario
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado. • El paciente debe estar registrado. • El actor debe escoger un paciente. • El actor debe poder consultar recomendaciones médicas al paciente.
<p>Post-condiciones:</p> <ul style="list-style-type: none"> • El actor podrá imprimir las recomendaciones médicas. • El actor podrá consultar las recomendaciones médicas.
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla de menú principal del paciente. 2. El Usuario selecciona consultar o agregar recomendación médica. 3. El Sistema despliega la pantalla de consulta o agregar recomendación médica. 4. El Usuario selecciona una recomendación médica existente en el Sistema 5. El Sistema despliega la información contenida en la dicha recomendación. 6. El Usuario presiona imprimir recomendación médica. 7. Se ejecuta el caso de uso imprimir recomendación médica. 8. El Sistema despliega la pantalla de consulta o agregar recomendación médica. <p>Se repite desde el paso 3.</p>
<p>Flujos Alternativos:</p> <ol style="list-style-type: none"> 6.1. El Usuario puede seleccionar otra recomendación médica. 6.2. El Sistema despliega la pantalla de consulta o agregar recomendación médica. Se repite desde el paso 4. 8.1. El usuario puede terminar de realizar su consulta al presionar terminar. 8.2. El Sistema despliega el menú principal del paciente seleccionado.

Caso de uso 8: Imprimir Notas Médicas

UC8: Imprimir notas médicas.
Actor Principal: Usuario
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado. • El actor debe haber elegido un paciente. • El actor debe poder consultar las notas médicas del paciente. • El actor debe haber elegido una nota para imprimir.
<p>Post-condiciones:</p> <ul style="list-style-type: none"> •
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla con la información contenida en la nota médica previamente seleccionada. 2. El Usuario presiona imprimir la nota médica seleccionada.

<p>3. El Sistema manda a imprimir a la impresora la nota médica seleccionada y el despliega la pantalla de consultar o agregar notas médicas.</p> <p>4. Se ejecuta el caso de uso agregar nota médica o consultar nota médica. Se repite desde el paso 1.</p>
<p>Flujos Alternativos:</p> <p>2.1. El Usuario presiona cancelar.</p> <p>2.2. El Sistema despliega la pantalla de consultar o agregar notas médicas. Se repite desde el paso 1.</p> <p>3.1. El Sistema tiene problemas con la impresora, notifica al usuario y pide que vuelva a intentarlo. Se repite desde el paso 2.</p>

Caso de uso 9: Imprimir recomendaciones médicas

UC9: Imprimir recomendaciones médicas.
Actor Principal: Usuario
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El actor debe estar autenticado. • El actor debe haber elegido un paciente. • El actor debe poder consultar las recomendaciones médicas del paciente. • El actor debe haber elegido una recomendación para imprimir
<p>Post-condiciones:</p> <ul style="list-style-type: none"> •
<p>Escenario Principal:</p> <ol style="list-style-type: none"> 1. El Sistema despliega la pantalla con la información contenida en la recomendación médica previamente seleccionada. 2. El Usuario presiona imprimir la recomendación médica seleccionada. 3. El Sistema manda a imprimir a la impresora la recomendación médica seleccionada y el despliega la pantalla de consultar o agregar recomendaciones médicas. 4. Se ejecuta el caso de uso agregar recomendación médica o consultar recomendación médica. 5. Se repite desde el paso 1.
<p>Flujos Alternativos:</p> <p>2.1. El Usuario presiona cancelar.</p> <p>2.2. El Sistema despliega la pantalla de consultar o agregar recomendaciones médicas. Se repite desde el paso 1.</p> <p>3.1. El Sistema tiene problemas con la impresora, notifica al usuario y pide que vuelva a intentarlo. Se repite desde el paso 2.</p>

Caso de uso 10: Agregar Comentarios.

UC10: Agregar Comentarios(al expediente clínico del paciente)
Actor Principal: Usuario
Precondiciones: <ul style="list-style-type: none">• El actor debe estar autenticado.• El paciente debe estar registrado.• El actor debe escoger un paciente.• El actor debe poder agregar comentarios al expediente clínico del paciente.
Post-condiciones: <ul style="list-style-type: none">• El actor podrá consultar los nuevos comentarios al expediente clínico del paciente.
Escenario Principal: <ol style="list-style-type: none">1. El Sistema despliega la pantalla de menú principal del paciente.2. El Usuario selecciona consultar o agregar comentarios.3. El Sistema despliega la pantalla de consulta o agregar comentario.4. El Usuario selecciona agregar un nuevo comentario.5. El Sistema despliega el formulario de llenado de comentarios.6. El Usuario llena el formulario y presiona aceptar.7. El Sistema verifica y guarda los datos ingresados y despliega la información contenida en el comentario.8. El usuario presiona terminar.9. El Sistema despliega la pantalla de menú principal del paciente.
Flujos Alternativos: <ol style="list-style-type: none">6.1. El Usuario puede cancelar el llenado del comentario al presionar cancelar.6.4 El Sistema despliega la pantalla de consulta o agregar comentario. Se repite desde el paso 3.7.1. El Sistema encuentra errores en el llenado del formulario, le informa al usuario y despliega nuevamente el formulario de llenado de comentarios.7.2. El Usuario llena los campos faltantes o erróneos y vuelve presionar guardar. Se repite desde el paso7.

Caso de uso 11: Consultar comentarios.

UC11: Consultar comentarios (del expediente clínico del paciente)
Actor Principal: Usuario
Precondiciones: <ul style="list-style-type: none">• El actor debe estar autenticado.• El paciente debe estar registrado.• El actor debe escoger un paciente.• El actor debe poder consultar comentarios del paciente.
Post-condiciones: <ul style="list-style-type: none">• El actor podrá consultar los comentarios del paciente.
Escenario Principal: <ol style="list-style-type: none">1. El Sistema despliega la pantalla de menú principal del paciente.2. El Usuario selecciona consultar o agregar comentarios.3. El Sistema despliega la pantalla de consulta o agregar comentarios4. El Usuario selecciona un comentario que quiera revisar.5. El Sistema despliega la información contenida en el dicho comentario.6. El Usuario presiona terminar.7. El Sistema despliega el menú principal del paciente.
Flujos Alternativos: <ol style="list-style-type: none">6.1. El Usuario puede seleccionar otro comentario.6.2. El Sistema despliega la pantalla de consulta o agregar recomendación médica. Se repite desde el paso 4.

Esquema Físico de la BD

```
--
-- PostgreSQL database dump
--

-- Started on 2010-09-09 07:51:13

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

--
-- TOC entry 2017 (class 1262 OID 16444)
-- Name: DataBaseECE; Type: DATABASE; Schema: -; Owner: postgres
--

CREATE DATABASE "DataBaseECE" WITH TEMPLATE = template0 ENCODING
= 'UTF8' LC_COLLATE = 'Spanish_Mexico.1252' LC_CTYPE =
'Spanish_Mexico.1252';

ALTER DATABASE "DataBaseECE" OWNER TO postgres;

\connect "DataBaseECE"

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

--
-- TOC entry 389 (class 2612 OID 16386)
-- Name: plpgsql; Type: PROCEDURAL LANGUAGE; Schema: -; Owner:
postgres
--

CREATE PROCEDURAL LANGUAGE plpgsql;

ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;

SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
```

```

-- TOC entry 1626 (class 1259 OID 34305)
-- Dependencies: 3
-- Name: comentario; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE comentario (
    id_comentario integer NOT NULL,
    idusuario integer NOT NULL,
    idnota integer NOT NULL,
    comentario character varying(255) NOT NULL
);

ALTER TABLE public.comentario OWNER TO postgres;

--
-- TOC entry 1625 (class 1259 OID 34303)
-- Dependencies: 1626 3
-- Name: comentario_id_comentario_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE comentario_id_comentario_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.comentario_id_comentario_seq OWNER TO postgres;

--
-- TOC entry 2020 (class 0 OID 0)
-- Dependencies: 1625
-- Name: comentario_id_comentario_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE comentario_id_comentario_seq OWNED BY comentario.id_comentario;

--
-- TOC entry 1591 (class 1259 OID 16893)
-- Dependencies: 3
-- Name: consultorio; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE consultorio (
    idconsultorio integer NOT NULL,
    id_unidad integer,

```

```

        numero integer,
        medicoresponsable character varying
    );

ALTER TABLE public.consultorio OWNER TO postgres;

--
-- TOC entry 1590 (class 1259 OID 16891)
-- Dependencies: 1591 3
-- Name: consultorio_idconsultorio_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE consultorio_idconsultorio_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.consultorio_idconsultorio_seq OWNER TO post-
gres;

--
-- TOC entry 2021 (class 0 OID 0)
-- Dependencies: 1590
-- Name: consultorio_idconsultorio_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE consultorio_idconsultorio_seq OWNED BY
consultorio.idconsultorio;

--
-- TOC entry 1622 (class 1259 OID 34173)
-- Dependencies: 3
-- Name: detalle_recomendacion; Type: TABLE; Schema: public;
Owner: postgres; Tablespace:
--

CREATE TABLE detalle_recomendacion (
    id_detalle_recomendacion integer NOT NULL,
    idrecomendaciones integer NOT NULL,
    medicamento character varying(45),
    modouso character varying(45),
    recomendacion character varying(255)
);

ALTER TABLE public.detalle_recomendacion OWNER TO postgres;

--

```

```

-- TOC entry 1621 (class 1259 OID 34171)
-- Dependencies: 3 1622
-- Name: detalle_recomendacion_id_detalle_recomendacion_seq;
Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE
detalle_recomendacion_id_detalle_recomendacion_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.detalle_recomendacion_id_detalle_recomen-
dacion_seq OWNER TO postgres;

--

-- TOC entry 2022 (class 0 OID 0)
-- Dependencies: 1621
-- Name: detalle_recomendacion_id_detalle_recomendacion_seq;
Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE
detalle_recomendacion_id_detalle_recomendacion_seq OWNED BY de-
talle_recomendacion.id_detalle_recomendacion;

--

-- TOC entry 1577 (class 1259 OID 16617)
-- Dependencies: 3
-- Name: escolaridad; Type: TABLE; Schema: public; Owner: post-
gres; Tablespace:
--

CREATE TABLE escolaridad (
    idescolaridad integer NOT NULL,
    escolaridad character varying(45)
);

ALTER TABLE public.escolaridad OWNER TO postgres;

--

-- TOC entry 1576 (class 1259 OID 16615)
-- Dependencies: 1577 3
-- Name: escolaridad_idescolaridad_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE escolaridad_idescolaridad_seq
    START WITH 1
    INCREMENT BY 1

```

```
NO MAXVALUE
NO MINVALUE
CACHE 1;
```

```
ALTER TABLE public.escolaridad_idescolaridad_seq OWNER TO postgres;
```

```
--
-- TOC entry 2023 (class 0 OID 0)
-- Dependencies: 1576
-- Name: escolaridad_idescolaridad_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE escolaridad_idescolaridad_seq OWNED BY
escolaridad.idescolaridad;
```

```
--
-- TOC entry 1595 (class 1259 OID 33391)
-- Dependencies: 3
-- Name: especialidad; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE especialidad (
    idespecialidad integer NOT NULL,
    especialidad character varying(45)
);
```

```
ALTER TABLE public.especialidad OWNER TO postgres;
```

```
--
-- TOC entry 1594 (class 1259 OID 33389)
-- Dependencies: 1595 3
-- Name: especialidad_idespecialidad_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE especialidad_idespecialidad_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.especialidad_idespecialidad_seq OWNER TO
postgres;
```

```
--
-- TOC entry 2024 (class 0 OID 0)
-- Dependencies: 1594
```

```

-- Name: especialidad_idespecialidad_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE especialidad_idespecialidad_seq OWNED BY
especialidad.idespecialidad;

--

-- TOC entry 1579 (class 1259 OID 16625)
-- Dependencies: 3
-- Name: estadocivil; Type: TABLE; Schema: public; Owner: post-
gres; Tablespace:
--

CREATE TABLE estadocivil (
    idestadocivil integer NOT NULL,
    estadocivil character varying(45)
);

ALTER TABLE public.estadocivil OWNER TO postgres;

--

-- TOC entry 1578 (class 1259 OID 16623)
-- Dependencies: 1579 3
-- Name: estadocivil_idestadocivil_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE estadocivil_idestadocivil_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.estadocivil_idestadocivil_seq OWNER TO post-
gres;

--

-- TOC entry 2025 (class 0 OID 0)
-- Dependencies: 1578
-- Name: estadocivil_idestadocivil_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE estadocivil_idestadocivil_seq OWNED BY estado-
civil.idestadocivil;

--

-- TOC entry 1614 (class 1259 OID 34084)
-- Dependencies: 3

```

```
-- Name: familiar_responsable; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE familiar_responsable (
    idfamiliarresponsable integer NOT NULL,
    nombre character varying(45),
    apellidopaterno character varying(45),
    apellidomaterno character varying(45),
    direccion character varying(255),
    telefono character varying(20),
    version integer,
    parentezco character varying(45)
);
```

```
ALTER TABLE public.familiar_responsable OWNER TO postgres;
```

```
--
-- TOC entry 1613 (class 1259 OID 34082)
-- Dependencies: 1614 3
-- Name: familiar_responsable_idfamiliarresponsable_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE familiar_responsable_idfamiliarresponsable_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE
public.familiar_responsable_idfamiliarresponsable_seq OWNER TO
postgres;
```

```
--
-- TOC entry 2026 (class 0 OID 0)
-- Dependencies: 1613
-- Name: familiar_responsable_idfamiliarresponsable_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE familiar_responsable_idfamiliarresponsable_seq
OWNED BY familiar_responsable.idfamiliarresponsable;
```

```
--
-- TOC entry 1581 (class 1259 OID 16641)
-- Dependencies: 3
-- Name: nacionalidad; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```

CREATE TABLE nacionalidad (
    idnacionalidad integer NOT NULL,
    nacionalidad character varying(45)
);

ALTER TABLE public.nacionalidad OWNER TO postgres;

--
-- TOC entry 1580 (class 1259 OID 16639)
-- Dependencies: 1581 3
-- Name: nacionalidad_idnacionalidad_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE nacionalidad_idnacionalidad_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.nacionalidad_idnacionalidad_seq OWNER TO
postgres;

--
-- TOC entry 2027 (class 0 OID 0)
-- Dependencies: 1580
-- Name: nacionalidad_idnacionalidad_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE nacionalidad_idnacionalidad_seq OWNED BY
nacionalidad.idnacionalidad;

--
-- TOC entry 1607 (class 1259 OID 33685)
-- Dependencies: 3
-- Name: nota; Type: TABLE; Schema: public; Owner: postgres; Ta-
blespace:
--

CREATE TABLE nota (
    id_nota integer NOT NULL,
    idpacientehospitalizado integer NOT NULL,
    idusuario integer NOT NULL,
    fecharegistro timestamp with time zone NOT NULL,
    diagnostico character varying(45) NOT NULL,
    nombrenota character varying(45) NOT NULL,
    estadosalud character varying(255),
    pronostico character varying(255)
);

```



```

ALTER TABLE public.nota OWNER TO postgres;

--
-- TOC entry 1606 (class 1259 OID 33683)
-- Dependencies: 3 1607
-- Name: nota_id_nota_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE nota_id_nota_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.nota_id_nota_seq OWNER TO postgres;

--
-- TOC entry 2028 (class 0 OID 0)
-- Dependencies: 1606
-- Name: nota_id_nota_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE nota_id_nota_seq OWNED BY nota.id_nota;

--
-- TOC entry 1620 (class 1259 OID 34138)
-- Dependencies: 3
-- Name: notaanestesiologia; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE notaanestesiologia (
    id_nota_anestesiologia integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    horainicioanestesiologia character varying(45) NOT NULL,
    duracionanestesiologia character varying(45) NOT NULL,
    horainicioquirurgico character varying(45) NOT NULL,
    duracionquirurgico character varying(45) NOT NULL,
    contigecias character varying(255) NOT NULL,
    estadopaciente character varying(255) NOT NULL,
    medicamentos character varying(255) NOT NULL,
    tecnicautilizada character varying(255) NOT NULL,
    liquidosintravenosos character varying(255) NOT NULL,
    pronostico character varying(255) NOT NULL
);

ALTER TABLE public.notaanestesiologia OWNER TO postgres;

```

```

--
-- TOC entry 1619 (class 1259 OID 34136)
-- Dependencies: 1620 3
-- Name: notaanestesiologia_id_nota_anestesiologia_seq; Type:
SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE notaanestesiologia_id_nota_anestesiologia_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.notaanestesiologia_id_nota_anestesiologia_seq
OWNER TO postgres;

--
-- TOC entry 2029 (class 0 OID 0)
-- Dependencies: 1619
-- Name: notaanestesiologia_id_nota_anestesiologia_seq; Type:
SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE notaanestesiologia_id_nota_anestesiologia_seq
OWNED BY notaanestesiologia.id_nota_anestesiologia;

--
-- TOC entry 1601 (class 1259 OID 33584)
-- Dependencies: 3
-- Name: notacontrareferencia; Type: TABLE; Schema: public; Own-
er: postgres; Tablespace:
--

CREATE TABLE notacontrareferencia (
    id_nota_contrareferencia integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    motivovaloracion character varying(255),
    antecedentes character varying(255),
    padecimientoactual character varying(255),
    exploracionfisica character varying(255),
    estudios character varying(255),
    recomendaciones character varying(255)
);

ALTER TABLE public.notacontrareferencia OWNER TO postgres;

--
-- TOC entry 1600 (class 1259 OID 33582)
-- Dependencies: 1601 3

```

```

-- Name: notacontrareferencia_id_nota_contrareferencia_seq;
Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE
notacontrareferencia_id_nota_contrareferencia_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE
public.notacontrareferencia_id_nota_contrareferencia_seq OWNER
TO postgres;

--
-- TOC entry 2030 (class 0 OID 0)
-- Dependencies: 1600
-- Name: notacontrareferencia_id_nota_contrareferencia_seq;
Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE notacontrareferencia_id_nota_contrareferencia_seq
OWNED BY notacontrareferencia.id_nota_contrareferencia;

--
-- TOC entry 1603 (class 1259 OID 33621)
-- Dependencies: 3
-- Name: notaegreso; Type: TABLE; Schema: public; Owner: post-
gres; Tablespace:
--

CREATE TABLE notaegreso (
    id_nota_egreso integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    fechaegreso date,
    motivoegreso character varying(255),
    diagnostico character varying(255),
    resumenclinico character varying(255),
    estadoactual character varying(255),
    planmanejo character varying(255),
    foliocertificado character varying(45),
    necropsia boolean
);

ALTER TABLE public.notaegreso OWNER TO postgres;

--
-- TOC entry 1602 (class 1259 OID 33619)
-- Dependencies: 3 1603

```

```

-- Name: notaegreso_id_nota_egreso_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE notaegreso_id_nota_egreso_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.notaegreso_id_nota_egreso_seq OWNER TO post-
gres;

--
-- TOC entry 2031 (class 0 OID 0)
-- Dependencies: 1602
-- Name: notaegreso_id_nota_egreso_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE notaegreso_id_nota_egreso_seq OWNED BY notae-
greso.id_nota_egreso;

--
-- TOC entry 1618 (class 1259 OID 34117)
-- Dependencies: 3
-- Name: notaingreso; Type: TABLE; Schema: public; Owner: post-
gres; Tablespace:
--

CREATE TABLE notaingreso (
    id_nota_ingreso integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    motivoingreso character varying(255) NOT NULL,
    antecedentes character varying(255) NOT NULL,
    padecimiento character varying(255) NOT NULL,
    exploracionfisica character varying(255) NOT NULL,
    exámenes character varying(255) NOT NULL
);

ALTER TABLE public.notaingreso OWNER TO postgres;

--
-- TOC entry 1617 (class 1259 OID 34115)
-- Dependencies: 3 1618
-- Name: notaingreso_id_nota_ingreso_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE notaingreso_id_nota_ingreso_seq

```

```

START WITH 1
INCREMENT BY 1
NO MAXVALUE
NO MINVALUE
CACHE 1;

ALTER TABLE public.notaingreso_id_nota_ingreso_seq OWNER TO
postgres;

--
-- TOC entry 2032 (class 0 OID 0)
-- Dependencies: 1617
-- Name: notaingreso_id_nota_ingreso_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE notaingreso_id_nota_ingreso_seq OWNED BY
notaingreso.id_nota_ingreso;

--
-- TOC entry 1605 (class 1259 OID 33643)
-- Dependencies: 3
-- Name: notainterconsulta; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE notainterconsulta (
    id_nota_interconsulta integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    antecedentes character varying(255),
    padecimientoactual character varying(255),
    exploracionfisica character varying(255),
    estudios character varying(255),
    recomendaciones character varying(255)
);

ALTER TABLE public.notainterconsulta OWNER TO postgres;

--
-- TOC entry 1604 (class 1259 OID 33641)
-- Dependencies: 3 1605
-- Name: notainterconsulta_id_nota_interconsulta_seq; Type: SE-
QUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE notainterconsulta_id_nota_interconsulta_seq
START WITH 1
INCREMENT BY 1
NO MAXVALUE
NO MINVALUE
CACHE 1;

```

```

ALTER TABLE public.notainterconsulta_id_nota_interconsulta_seq
OWNER TO postgres;

--
-- TOC entry 2033 (class 0 OID 0)
-- Dependencies: 1604
-- Name: notainterconsulta_id_nota_interconsulta_seq; Type: SE-
QUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE notainterconsulta_id_nota_interconsulta_seq OWNED
BY notainterconsulta.id_nota_interconsulta;

--
-- TOC entry 1599 (class 1259 OID 33563)
-- Dependencies: 3
-- Name: notareferencia; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE notareferencia (
    id_nota_referencia integer NOT NULL,
    idnota integer NOT NULL,
    idvitales integer NOT NULL,
    motivoenvio character varying(255),
    hospitalenvia character varying(255),
    hospitalrecibe character varying(255),
    medicoenvia character varying(255),
    medicorecibe character varying(255),
    resumenclinico character varying(255),
    terapeutica character varying(255),
    recomendaciones character varying(255)
);

ALTER TABLE public.notareferencia OWNER TO postgres;

--
-- TOC entry 1598 (class 1259 OID 33561)
-- Dependencies: 3 1599
-- Name: notareferencia_id_nota_referencia_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE notareferencia_id_nota_referencia_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

```

```

ALTER TABLE public.notareferencia_id_nota_referencia_seq OWNER
TO postgres;

--
-- TOC entry 2034 (class 0 OID 0)
-- Dependencies: 1598
-- Name: notareferencia_id_nota_referencia_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE notareferencia_id_nota_referencia_seq OWNED BY
notareferencia.id_nota_referencia;

--
-- TOC entry 1583 (class 1259 OID 16649)
-- Dependencies: 3
-- Name: ocupacion; Type: TABLE; Schema: public; Owner: post-
gres; Tablespace:
--

CREATE TABLE ocupacion (
    idocupacion integer NOT NULL,
    ocupacion character varying(45)
);

ALTER TABLE public.ocupacion OWNER TO postgres;

--
-- TOC entry 1582 (class 1259 OID 16647)
-- Dependencies: 1583 3
-- Name: ocupacion_idocupacion_seq; Type: SEQUENCE; Schema: pub-
lic; Owner: postgres
--

CREATE SEQUENCE ocupacion_idocupacion_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.ocupacion_idocupacion_seq OWNER TO postgres;

--
-- TOC entry 2035 (class 0 OID 0)
-- Dependencies: 1582
-- Name: ocupacion_idocupacion_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE ocupacion_idocupacion_seq OWNED BY
ocupacion.idocupacion;

```

```
--  
-- TOC entry 1610 (class 1259 OID 33882)  
-- Dependencies: 3  
-- Name: paciente; Type: TABLE; Schema: public; Owner: postgres;  
Tablespace:  
--
```

```
CREATE TABLE paciente (  
    id_paciente bigint NOT NULL,  
    idturno integer NOT NULL,  
    idconsultorio integer NOT NULL,  
    idescolaridad integer NOT NULL,  
    idocupacion integer NOT NULL,  
    idestadocivil integer NOT NULL,  
    idreligion integer NOT NULL,  
    idfamiliarresponsable integer NOT NULL,  
    idnacionalidad integer NOT NULL,  
    nombre character varying(45),  
    apellidopaterno character varying(45),  
    apellidomaterno character varying(45),  
    fechanacimiento date,  
    edad integer,  
    idsexo integer NOT NULL,  
    direccion character varying(255),  
    telefono character varying(20),  
    lugarorigen character varying(45),  
    etnicidad character varying(45),  
    estatus boolean,  
    curp character varying(45),  
    estado boolean,  
    version integer  
);
```

```
ALTER TABLE public.paciente OWNER TO postgres;
```

```
--  
-- TOC entry 1609 (class 1259 OID 33880)  
-- Dependencies: 1610 3  
-- Name: paciente_id_paciente_seq; Type: SEQUENCE; Schema: public;  
Owner: postgres  
--
```

```
CREATE SEQUENCE paciente_id_paciente_seq  
    START WITH 1  
    INCREMENT BY 1  
    NO MAXVALUE  
    NO MINVALUE  
    CACHE 1;
```

```
ALTER TABLE public.paciente_id_paciente_seq OWNER TO postgres;
```



```

--
-- TOC entry 2036 (class 0 OID 0)
-- Dependencies: 1609
-- Name: paciente_id_paciente_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE paciente_id_paciente_seq OWNED BY paciente.id_pa-
ciente;

--
-- TOC entry 1616 (class 1259 OID 34092)
-- Dependencies: 3
-- Name: pacientehospitalizado; Type: TABLE; Schema: public;
Owner: postgres; Tablespace:
--

CREATE TABLE pacientehospitalizado (
    idpacientehospitalizado integer NOT NULL,
    id_usuario integer NOT NULL,
    id_especialidad integer NOT NULL,
    id_paciente integer NOT NULL,
    fechaadmission date,
    diagnostico character varying(255),
    cama integer
);

ALTER TABLE public.pacientehospitalizado OWNER TO postgres;

--
-- TOC entry 1615 (class 1259 OID 34090)
-- Dependencies: 1616 3
-- Name: pacientehospitalizado_idpacientehospitalizado_seq;
Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE
pacientehospitalizado_idpacientehospitalizado_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.pacientehospitalizado_idpacientehospitaliz-
ado_seq OWNER TO postgres;

--
-- TOC entry 2037 (class 0 OID 0)
-- Dependencies: 1615
-- Name: pacientehospitalizado_idpacientehospitalizado_seq;
Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

```

```

--

ALTER SEQUENCE pacientehospitalizado_idpacientehospitalizado_seq
OWNED BY pacientehospitalizado.idpacientehospitalizado;

--

-- TOC entry 1624 (class 1259 OID 34222)
-- Dependencies: 3
-- Name: recomendaciones; Type: TABLE; Schema: public; Owner:
postgres; Tablespace:
--

CREATE TABLE recomendaciones (
    id_recomendaciones integer NOT NULL,
    idusuario integer NOT NULL,
    idpaciente integer NOT NULL,
    fecha timestamp without time zone,
    estatus boolean
);

ALTER TABLE public.recomendaciones OWNER TO postgres;

--

-- TOC entry 1623 (class 1259 OID 34220)
-- Dependencies: 1624 3
-- Name: recomendaciones_id_recomendaciones_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE recomendaciones_id_recomendaciones_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.recomendaciones_id_recomendaciones_seq OWNER
TO postgres;

--

-- TOC entry 2038 (class 0 OID 0)
-- Dependencies: 1623
-- Name: recomendaciones_id_recomendaciones_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE recomendaciones_id_recomendaciones_seq OWNED BY
recomendaciones.id_recomendaciones;

--

-- TOC entry 1585 (class 1259 OID 16657)

```

```
-- Dependencies: 3
-- Name: religion; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE religion (
    idreligion integer NOT NULL,
    religion character varying(45)
);
```

```
ALTER TABLE public.religion OWNER TO postgres;
```

```
--
-- TOC entry 1584 (class 1259 OID 16655)
-- Dependencies: 1585 3
-- Name: religion_idreligion_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--
```

```
CREATE SEQUENCE religion_idreligion_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.religion_idreligion_seq OWNER TO postgres;
```

```
--
-- TOC entry 2039 (class 0 OID 0)
-- Dependencies: 1584
-- Name: religion_idreligion_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE religion_idreligion_seq OWNED BY religion.idreligion;
```

```
--
-- TOC entry 1593 (class 1259 OID 25114)
-- Dependencies: 3
-- Name: roles; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE roles (
    id_rol integer NOT NULL,
    rol character varying(45) NOT NULL
);
```

```
ALTER TABLE public.roles OWNER TO postgres;
```

```

--
-- TOC entry 1592 (class 1259 OID 25112)
-- Dependencies: 3 1593
-- Name: roles_id_rol_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE roles_id_rol_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.roles_id_rol_seq OWNER TO postgres;

--
-- TOC entry 2040 (class 0 OID 0)
-- Dependencies: 1592
-- Name: roles_id_rol_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE roles_id_rol_seq OWNED BY roles.id_rol;

--
-- TOC entry 1608 (class 1259 OID 33875)
-- Dependencies: 3
-- Name: sexo; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE sexo (
    id_sexo integer NOT NULL,
    sexo character varying(45)
);

ALTER TABLE public.sexo OWNER TO postgres;

--
-- TOC entry 1597 (class 1259 OID 33534)
-- Dependencies: 3
-- Name: signosvitaales; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE signosvitaales (
    id_signos_vitales integer NOT NULL,
    peso double precision NOT NULL,
    talla double precision NOT NULL,
    temperatura double precision NOT NULL,

```

```

    freccardiaca double precision NOT NULL,
    frecrespiratoria double precision NOT NULL,
    tensionarterial double precision NOT NULL,
    imc double precision NOT NULL
);

ALTER TABLE public.signosvitaales OWNER TO postgres;

--
-- TOC entry 1596 (class 1259 OID 33532)
-- Dependencies: 1597 3
-- Name: signosvitaales_id_signos_vitales_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE signosvitaales_id_signos_vitales_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.signosvitaales_id_signos_vitales_seq OWNER TO
postgres;

--
-- TOC entry 2041 (class 0 OID 0)
-- Dependencies: 1596
-- Name: signosvitaales_id_signos_vitales_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE signosvitaales_id_signos_vitales_seq OWNED BY si-
gnosvitaales.id_signos_vitales;

--
-- TOC entry 1587 (class 1259 OID 16665)
-- Dependencies: 3
-- Name: turno; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--

CREATE TABLE turno (
    idturno integer NOT NULL,
    turno character varying(20)
);

ALTER TABLE public.turno OWNER TO postgres;

--
-- TOC entry 1586 (class 1259 OID 16663)

```

```
-- Dependencies: 3 1587
-- Name: turno_idturno_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE turno_idturno_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.turno_idturno_seq OWNER TO postgres;
```

```
--
-- TOC entry 2042 (class 0 OID 0)
-- Dependencies: 1586
-- Name: turno_idturno_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE turno_idturno_seq OWNED BY turno.idturno;
```

```
--
-- TOC entry 1589 (class 1259 OID 16673)
-- Dependencies: 3
-- Name: unidad; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE unidad (
    idunidad integer NOT NULL,
    tipo character varying(45),
    nombre character varying(45),
    domicilio character varying(255),
    institucionperteneiente character varying(45),
    razonsocial character varying(45)
);
```

```
ALTER TABLE public.unidad OWNER TO postgres;
```

```
--
-- TOC entry 1588 (class 1259 OID 16671)
-- Dependencies: 1589 3
-- Name: unidad_idunidad_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE unidad_idunidad_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
```

```
NO MINVALUE
CACHE 1;
```

```
ALTER TABLE public.unidad_idunidad_seq OWNER TO postgres;
```

```
--
-- TOC entry 2043 (class 0 OID 0)
-- Dependencies: 1588
-- Name: unidad_idunidad_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--
```

```
ALTER SEQUENCE unidad_idunidad_seq OWNED BY unidad.idunidad;
```

```
--
-- TOC entry 1612 (class 1259 OID 34034)
-- Dependencies: 3
-- Name: usuario; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
```

```
CREATE TABLE usuario (
    id_usuario integer NOT NULL,
    idescolaridad integer NOT NULL,
    idunidad integer NOT NULL,
    idestadocivil integer NOT NULL,
    idturno integer NOT NULL,
    idnacionalidad integer NOT NULL,
    idrol integer NOT NULL,
    nombre character varying(45),
    apellidopaterno character varying(45),
    apellidomaterno character varying(45),
    cedula character varying(45),
    direccion character varying(255),
    telefono character varying(20),
    idsexo integer NOT NULL,
    usuario character varying(20) NOT NULL,
    contrasenia character varying(20) NOT NULL,
    fechanacimiento date,
    tipo character varying(45),
    version integer
);
```

```
ALTER TABLE public.usuario OWNER TO postgres;
```

```
--
-- TOC entry 1611 (class 1259 OID 34032)
-- Dependencies: 3 1612
-- Name: usuario_id_usuario_seq; Type: SEQUENCE; Schema: public;
Owner: postgres
--
```

```

CREATE SEQUENCE usuario_id_usuario_seq
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;

ALTER TABLE public.usuario_id_usuario_seq OWNER TO postgres;

--
-- TOC entry 2044 (class 0 OID 0)
-- Dependencies: 1611
-- Name: usuario_id_usuario_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE usuario_id_usuario_seq OWNED BY
usuario.id_usuario;

--
-- TOC entry 1928 (class 2604 OID 34308)
-- Dependencies: 1626 1625 1626
-- Name: id_comentario; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE comentario ALTER COLUMN id_comentario SET DEFAULT
nextval('comentario_id_comentario_seq'::regclass);

--
-- TOC entry 1911 (class 2604 OID 16896)
-- Dependencies: 1590 1591 1591
-- Name: id_consultorio; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE consultorio ALTER COLUMN id_consultorio SET DEFAULT
nextval('consultorio_id_consultorio_seq'::regclass);

--
-- TOC entry 1926 (class 2604 OID 34176)
-- Dependencies: 1621 1622 1622
-- Name: id_detalle_recomendacion; Type: DEFAULT; Schema: pub-
lic; Owner: postgres
--

ALTER TABLE detalle_recomendacion ALTER COLUMN id_detalle_re-
comendacion SET DEFAULT nextval('detalle_recomendacion_id_de-
talle_recomendacion_seq'::regclass);

```



```

--
-- TOC entry 1904 (class 2604 OID 16620)
-- Dependencies: 1576 1577 1577
-- Name: idescolaridad; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE escolaridad ALTER COLUMN idescolaridad SET DEFAULT
nextval('escolaridad_idescolaridad_seq'::regclass);

--
-- TOC entry 1913 (class 2604 OID 33394)
-- Dependencies: 1595 1594 1595
-- Name: idespecialidad; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE especialidad ALTER COLUMN idespecialidad SET DEFAULT
nextval('especialidad_idespecialidad_seq'::regclass);

--
-- TOC entry 1905 (class 2604 OID 16628)
-- Dependencies: 1579 1578 1579
-- Name: idestadocivil; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE estadocivil ALTER COLUMN idestadocivil SET DEFAULT
nextval('estadocivil_idestadocivil_seq'::regclass);

--
-- TOC entry 1922 (class 2604 OID 34087)
-- Dependencies: 1613 1614 1614
-- Name: idfamiliarresponsable; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE familiar_responsable ALTER COLUMN idfamiliarrespons-
able SET DEFAULT nextval('familiar_responsable_idfamiliarre-
sponsable_seq'::regclass);

--
-- TOC entry 1906 (class 2604 OID 16644)
-- Dependencies: 1580 1581 1581
-- Name: idnacionalidad; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE nacionalidad ALTER COLUMN idnacionalidad SET DEFAULT
nextval('nacionalidad_idnacionalidad_seq'::regclass);

```

```
--  
-- TOC entry 1919 (class 2604 OID 33688)  
-- Dependencies: 1607 1606 1607  
-- Name: id_nota; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE nota ALTER COLUMN id_nota SET DEFAULT  
nextval('nota_id_nota_seq'::regclass);
```

```
--  
-- TOC entry 1925 (class 2604 OID 34141)  
-- Dependencies: 1619 1620 1620  
-- Name: id_nota_anestesiologia; Type: DEFAULT; Schema: public;  
Owner: postgres  
--
```

```
ALTER TABLE notaanestesiologia ALTER COLUMN  
id_nota_anestesiologia SET DEFAULT  
nextval('notaanestesiologia_id_nota_anestesiologia_seq'::regclas  
s);
```

```
--  
-- TOC entry 1916 (class 2604 OID 33587)  
-- Dependencies: 1600 1601 1601  
-- Name: id_nota_contrareferencia; Type: DEFAULT; Schema:  
public; Owner: postgres  
--
```

```
ALTER TABLE notacontrareferencia ALTER COLUMN  
id_nota_contrareferencia SET DEFAULT  
nextval('notacontrareferencia_id_nota_contrareferencia_seq'::reg  
class);
```

```
--  
-- TOC entry 1917 (class 2604 OID 33624)  
-- Dependencies: 1602 1603 1603  
-- Name: id_nota_egreso; Type: DEFAULT; Schema: public; Owner:  
postgres  
--
```

```
ALTER TABLE notaegreso ALTER COLUMN id_nota_egreso SET DEFAULT  
nextval('notaegreso_id_nota_egreso_seq'::regclass);
```

```
--  
-- TOC entry 1924 (class 2604 OID 34120)  
-- Dependencies: 1618 1617 1618  
-- Name: id_nota_ingreso; Type: DEFAULT; Schema: public; Owner:  
postgres  
--
```

```
ALTER TABLE notaingreso ALTER COLUMN id_nota_ingreso SET DEFAULT
nextval('notaingreso_id_nota_ingreso_seq'::regclass);
```

```
--
```

```
-- TOC entry 1918 (class 2604 OID 33646)
-- Dependencies: 1605 1604 1605
-- Name: id_nota_interconsulta; Type: DEFAULT; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER TABLE notainterconsulta ALTER COLUMN id_nota_interconsulta
SET DEFAULT
nextval('notainterconsulta_id_nota_interconsulta_seq'::regclass)
;
```

```
--
```

```
-- TOC entry 1915 (class 2604 OID 33566)
-- Dependencies: 1598 1599 1599
-- Name: id_nota_referencia; Type: DEFAULT; Schema: public; Own-
er: postgres
```

```
--
```

```
ALTER TABLE notareferencia ALTER COLUMN id_nota_referencia SET
DEFAULT
nextval('notareferencia_id_nota_referencia_seq'::regclass);
```

```
--
```

```
-- TOC entry 1907 (class 2604 OID 16652)
-- Dependencies: 1583 1582 1583
-- Name: idocupacion; Type: DEFAULT; Schema: public; Owner:
postgres
```

```
--
```

```
ALTER TABLE ocupacion ALTER COLUMN idocupacion SET DEFAULT nex-
tval('ocupacion_idocupacion_seq'::regclass);
```

```
--
```

```
-- TOC entry 1920 (class 2604 OID 33885)
-- Dependencies: 1609 1610 1610
-- Name: id_paciente; Type: DEFAULT; Schema: public; Owner:
postgres
```

```
--
```

```
ALTER TABLE paciente ALTER COLUMN id_paciente SET DEFAULT nextv-
al('paciente_id_paciente_seq'::regclass);
```

```
--
```

```
-- TOC entry 1923 (class 2604 OID 34095)
-- Dependencies: 1616 1615 1616
```

```

-- Name: idpacientehospitalizado; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE pacientehospitalizado ALTER COLUMN
idpacientehospitalizado SET DEFAULT
nextval('pacientehospitalizado_idpacientehospitalizado_seq'::reg
class);

--

-- TOC entry 1927 (class 2604 OID 34225)
-- Dependencies: 1624 1623 1624
-- Name: id_recomendaciones; Type: DEFAULT; Schema: public; Own-
er: postgres
--

ALTER TABLE recomendaciones ALTER COLUMN id_recomendaciones SET
DEFAULT
nextval('recomendaciones_id_recomendaciones_seq'::regclass);

--

-- TOC entry 1908 (class 2604 OID 16660)
-- Dependencies: 1584 1585 1585
-- Name: idreligion; Type: DEFAULT; Schema: public; Owner: post-
gres
--

ALTER TABLE religion ALTER COLUMN idreligion SET DEFAULT nextv-
al('religion_idreligion_seq'::regclass);

--

-- TOC entry 1912 (class 2604 OID 25117)
-- Dependencies: 1593 1592 1593
-- Name: id_rol; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE roles ALTER COLUMN id_rol SET DEFAULT
nextval('roles_id_rol_seq'::regclass);

--

-- TOC entry 1914 (class 2604 OID 33537)
-- Dependencies: 1597 1596 1597
-- Name: id_signos_vitales; Type: DEFAULT; Schema: public; Own-
er: postgres
--

ALTER TABLE signosvitales ALTER COLUMN id_signos_vitales SET DE-
FAULT nextval('signosvitales_id_signos_vitales_seq'::regclass);

--

```

```
-- TOC entry 1909 (class 2604 OID 16668)
-- Dependencies: 1587 1586 1587
-- Name: idturno; Type: DEFAULT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE turno ALTER COLUMN idturno SET DEFAULT
nextval('turno_idturno_seq'::regclass);
```

```
--
-- TOC entry 1910 (class 2604 OID 16676)
-- Dependencies: 1589 1588 1589
-- Name: idunidad; Type: DEFAULT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE unidad ALTER COLUMN idunidad SET DEFAULT
nextval('unidad_idunidad_seq'::regclass);
```

```
--
-- TOC entry 1921 (class 2604 OID 34037)
-- Dependencies: 1612 1611 1612
-- Name: id_usuario; Type: DEFAULT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE usuario ALTER COLUMN id_usuario SET DEFAULT
nextval('usuario_id_usuario_seq'::regclass);
```

```
--
-- TOC entry 1984 (class 2606 OID 34310)
-- Dependencies: 1626 1626
-- Name: comentario_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY comentario
    ADD CONSTRAINT comentario_pkey PRIMARY KEY (id_comentario);
```

```
--
-- TOC entry 1944 (class 2606 OID 16901)
-- Dependencies: 1591 1591
-- Name: consultorio_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY consultorio
    ADD CONSTRAINT consultorio_pkey PRIMARY KEY (idconsultorio);
```

```
--
-- TOC entry 1980 (class 2606 OID 34178)
```

```

-- Dependencies: 1622 1622
-- Name: detalle_recomendacion_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY detalle_recomendacion
    ADD CONSTRAINT detalle_recomendacion_pkey PRIMARY KEY
(id_detalle_recomendacion);

--

-- TOC entry 1930 (class 2606 OID 16622)
-- Dependencies: 1577 1577
-- Name: escolaridad_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY escolaridad
    ADD CONSTRAINT escolaridad_pkey PRIMARY KEY (idescolaridad);

--

-- TOC entry 1948 (class 2606 OID 33396)
-- Dependencies: 1595 1595
-- Name: especialidad_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY especialidad
    ADD CONSTRAINT especialidad_pkey PRIMARY KEY (idespecialid-
ad);

--

-- TOC entry 1932 (class 2606 OID 16630)
-- Dependencies: 1579 1579
-- Name: estadocivil_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY estadocivil
    ADD CONSTRAINT estadocivil_pkey PRIMARY KEY (idestadocivil);

--

-- TOC entry 1972 (class 2606 OID 34089)
-- Dependencies: 1614 1614
-- Name: familiar_responsable_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY familiar_responsable
    ADD CONSTRAINT familiar_responsable_pkey PRIMARY KEY (idfa-
miliarresponsable);

```

```

--
-- TOC entry 1934 (class 2606 OID 16646)
-- Dependencies: 1581 1581
-- Name: nacionalidad_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY nacionalidad
    ADD CONSTRAINT nacionalidad_pkey PRIMARY KEY (idnacionalid-
ad);

--
-- TOC entry 1960 (class 2606 OID 33690)
-- Dependencies: 1607 1607
-- Name: nota_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY nota
    ADD CONSTRAINT nota_pkey PRIMARY KEY (id_nota);

--
-- TOC entry 1978 (class 2606 OID 34146)
-- Dependencies: 1620 1620
-- Name: notaanestesiologia_pkey; Type: CONSTRAINT; Schema: pub-
lic; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY notaanestesiologia
    ADD CONSTRAINT notaanestesiologia_pkey PRIMARY KEY
(id_nota_anestesiologia);

--
-- TOC entry 1954 (class 2606 OID 33592)
-- Dependencies: 1601 1601
-- Name: notacontrareferencia_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY notacontrareferencia
    ADD CONSTRAINT notacontrareferencia_pkey PRIMARY KEY
(id_nota_contrareferencia);

--
-- TOC entry 1956 (class 2606 OID 33629)
-- Dependencies: 1603 1603
-- Name: notaegreso_pkey; Type: CONSTRAINT; Schema: public; Own-
er: postgres; Tablespace:
--

```

```
ALTER TABLE ONLY notaegreso
    ADD CONSTRAINT notaegreso_pkey PRIMARY KEY (id_nota_egreso);
```

```
--
-- TOC entry 1976 (class 2606 OID 34125)
-- Dependencies: 1618 1618
-- Name: notaingreso_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY notaingreso
    ADD CONSTRAINT notaingreso_pkey PRIMARY KEY (id_nota_in-
greso);
```

```
--
-- TOC entry 1958 (class 2606 OID 33651)
-- Dependencies: 1605 1605
-- Name: notainterconsulta_pkey; Type: CONSTRAINT; Schema: pub-
lic; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY notainterconsulta
    ADD CONSTRAINT notainterconsulta_pkey PRIMARY KEY
(id_nota_interconsulta);
```

```
--
-- TOC entry 1952 (class 2606 OID 33571)
-- Dependencies: 1599 1599
-- Name: notareferencia_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY notareferencia
    ADD CONSTRAINT notareferencia_pkey PRIMARY KEY (id_nota_ref-
erencia);
```

```
--
-- TOC entry 1936 (class 2606 OID 16654)
-- Dependencies: 1583 1583
-- Name: ocupacion_pkey; Type: CONSTRAINT; Schema: public; Own-
er: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY ocupacion
    ADD CONSTRAINT ocupacion_pkey PRIMARY KEY (idocupacion);
```

```
--
-- TOC entry 1964 (class 2606 OID 33890)
-- Dependencies: 1610 1610
```



```
-- Name: paciente_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```
ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_pkey PRIMARY KEY (id_paciente);
```

```
--
-- TOC entry 1974 (class 2606 OID 34097)
-- Dependencies: 1616 1616
-- Name: pacientehospitalizado_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY pacientehospitalizado
    ADD CONSTRAINT pacientehospitalizado_pkey PRIMARY KEY (idpa-
cientehospitalizado);
```

```
--
-- TOC entry 1982 (class 2606 OID 34227)
-- Dependencies: 1624 1624
-- Name: recomendaciones_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--
```

```
ALTER TABLE ONLY recomendaciones
    ADD CONSTRAINT recomendaciones_pkey PRIMARY KEY (id_recomen-
daciones);
```

```
--
-- TOC entry 1938 (class 2606 OID 16662)
-- Dependencies: 1585 1585
-- Name: religion_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```
ALTER TABLE ONLY religion
    ADD CONSTRAINT religion_pkey PRIMARY KEY (idreligion);
```

```
--
-- TOC entry 1946 (class 2606 OID 25119)
-- Dependencies: 1593 1593
-- Name: roles_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--
```

```
ALTER TABLE ONLY roles
    ADD CONSTRAINT roles_pkey PRIMARY KEY (id_rol);
```

```
--
```

```

-- TOC entry 1962 (class 2606 OID 33879)
-- Dependencies: 1608 1608
-- Name: sexo_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY sexo
    ADD CONSTRAINT sexo_pkey PRIMARY KEY (id_sexo);

--

-- TOC entry 1950 (class 2606 OID 33539)
-- Dependencies: 1597 1597
-- Name: signosvitaless_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY signosvitaless
    ADD CONSTRAINT signosvitaless_pkey PRIMARY KEY (id_signos_vi-
tales);

--

-- TOC entry 1940 (class 2606 OID 16670)
-- Dependencies: 1587 1587
-- Name: turno_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY turno
    ADD CONSTRAINT turno_pkey PRIMARY KEY (idturno);

--

-- TOC entry 1942 (class 2606 OID 16678)
-- Dependencies: 1589 1589
-- Name: unidad_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY unidad
    ADD CONSTRAINT unidad_pkey PRIMARY KEY (idunidad);

--

-- TOC entry 1966 (class 2606 OID 34046)
-- Dependencies: 1612 1612
-- Name: usuario_contrasenia_key; Type: CONSTRAINT; Schema: pub-
lic; Owner: postgres; Tablespace:
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_contrasenia_key UNIQUE (contrasenia);

```

```

--
-- TOC entry 1968 (class 2606 OID 34042)
-- Dependencies: 1612 1612
-- Name: usuario_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres; Tablespace:
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario);

--
-- TOC entry 1970 (class 2606 OID 34044)
-- Dependencies: 1612 1612
-- Name: usuario_usuario_key; Type: CONSTRAINT; Schema: public;
Owner: postgres; Tablespace:
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_usuario_key UNIQUE (usuario);

--
-- TOC entry 2014 (class 2606 OID 34316)
-- Dependencies: 1959 1626 1607
-- Name: comentario_idnota_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY comentario
    ADD CONSTRAINT comentario_idnota_fkey FOREIGN KEY (idnota)
REFERENCES nota(id_nota);

--
-- TOC entry 2013 (class 2606 OID 34311)
-- Dependencies: 1612 1626 1967
-- Name: comentario_idusuario_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY comentario
    ADD CONSTRAINT comentario_idusuario_fkey FOREIGN KEY
(idusuario) REFERENCES usuario(id_usuario);

--
-- TOC entry 1985 (class 2606 OID 16902)
-- Dependencies: 1941 1591 1589
-- Name: consultorio_id_unidad_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY consultorio

```

```

        ADD CONSTRAINT consultorio_id_unidad_fkey FOREIGN KEY
(id_unidad) REFERENCES unidad(idunidad);

--
-- TOC entry 2010 (class 2606 OID 34147)
-- Dependencies: 1949 1620 1597
-- Name: notaanestesiologia_idvitaless_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY notaanestesiologia
        ADD CONSTRAINT notaanestesiologia_idvitaless_fkey FOREIGN KEY
(idvitaless) REFERENCES signosvitaless(id_signos_vitaless);

--
-- TOC entry 1987 (class 2606 OID 33598)
-- Dependencies: 1601 1949 1597
-- Name: notacontrareferencia_idvitaless_fkey; Type: FK CON-
STRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY notacontrareferencia
        ADD CONSTRAINT notacontrareferencia_idvitaless_fkey FOREIGN
KEY (idvitaless) REFERENCES signosvitaless(id_signos_vitaless);

--
-- TOC entry 1988 (class 2606 OID 33635)
-- Dependencies: 1949 1603 1597
-- Name: notaegreso_idvitaless_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY notaegreso
        ADD CONSTRAINT notaegreso_idvitaless_fkey FOREIGN KEY (idvi-
taless) REFERENCES signosvitaless(id_signos_vitaless);

--
-- TOC entry 2008 (class 2606 OID 34126)
-- Dependencies: 1607 1618 1959
-- Name: notaingreso_idnota_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY notaingreso
        ADD CONSTRAINT notaingreso_idnota_fkey FOREIGN KEY (idnota)
REFERENCES nota(id_nota);

--
-- TOC entry 2009 (class 2606 OID 34131)
-- Dependencies: 1618 1949 1597

```

```

-- Name: notaingreso_idvitaless_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY notaingreso
    ADD CONSTRAINT notaingreso_idvitaless_fkey FOREIGN KEY (idvitaless) REFERENCES signosvitaless(id_signos_vitaless);

--

-- TOC entry 1989 (class 2606 OID 33657)
-- Dependencies: 1597 1605 1949
-- Name: notainterconsulta_idvitaless_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY notainterconsulta
    ADD CONSTRAINT notainterconsulta_idvitaless_fkey FOREIGN KEY (idvitaless) REFERENCES signosvitaless(id_signos_vitaless);

--

-- TOC entry 1986 (class 2606 OID 33577)
-- Dependencies: 1599 1597 1949
-- Name: notareferencia_idvitaless_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY notareferencia
    ADD CONSTRAINT notareferencia_idvitaless_fkey FOREIGN KEY (idvitaless) REFERENCES signosvitaless(id_signos_vitaless);

--

-- TOC entry 1990 (class 2606 OID 33891)
-- Dependencies: 1610 1943 1591
-- Name: paciente_idconsultorio_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idconsultorio_fkey FOREIGN KEY (idconsultorio) REFERENCES consultorio(idconsultorio);

--

-- TOC entry 1991 (class 2606 OID 33896)
-- Dependencies: 1577 1610 1929
-- Name: paciente_idescolaridad_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idescolaridad_fkey FOREIGN KEY (idescolaridad) REFERENCES escolaridad(idescolaridad);

```

```

--
-- TOC entry 1992 (class 2606 OID 33901)
-- Dependencies: 1579 1931 1610
-- Name: paciente_idestadocivil_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idestadocivil_fkey FOREIGN KEY
(idestadocivil) REFERENCES estadocivil(idestadocivil);

--

-- TOC entry 1993 (class 2606 OID 33911)
-- Dependencies: 1610 1581 1933
-- Name: paciente_idnacionalidad_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idnacionalidad_fkey FOREIGN KEY (id-
nacionalidad) REFERENCES nacionalidad(idnacionalidad);

--

-- TOC entry 1994 (class 2606 OID 33916)
-- Dependencies: 1583 1935 1610
-- Name: paciente_idocupacion_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idocupacion_fkey FOREIGN KEY (idocu-
pacion) REFERENCES ocupacion(idocupacion);

--

-- TOC entry 1995 (class 2606 OID 33921)
-- Dependencies: 1937 1610 1585
-- Name: paciente_idreligion_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idreligion_fkey FOREIGN KEY (idreli-
gion) REFERENCES religion(idreligion);

--

-- TOC entry 1997 (class 2606 OID 33931)
-- Dependencies: 1610 1961 1608
-- Name: paciente_idsexo_fkey; Type: FK CONSTRAINT; Schema: pub-
lic; Owner: postgres

```

```

--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idsexo_fkey FOREIGN KEY (idsexo)
REFERENCES sexo(id_sexo);

--

-- TOC entry 1996 (class 2606 OID 33926)
-- Dependencies: 1939 1610 1587
-- Name: paciente_idturno_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY paciente
    ADD CONSTRAINT paciente_idturno_fkey FOREIGN KEY (idturno)
REFERENCES turno(idturno);

--

-- TOC entry 2006 (class 2606 OID 34103)
-- Dependencies: 1616 1595 1947
-- Name: pacientehospitalizado_id_especialidad_fkey; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY pacientehospitalizado
    ADD CONSTRAINT pacientehospitalizado_id_especialidad_fkey
FOREIGN KEY (id_especialidad) REFERENCES especialidad(idespe-
cialidad);

--

-- TOC entry 2007 (class 2606 OID 34108)
-- Dependencies: 1616 1610 1963
-- Name: pacientehospitalizado_id_paciente_fkey; Type: FK CON-
STRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY pacientehospitalizado
    ADD CONSTRAINT pacientehospitalizado_id_paciente_fkey FOR-
EIGN KEY (id_paciente) REFERENCES paciente(id_paciente);

--

-- TOC entry 2005 (class 2606 OID 34098)
-- Dependencies: 1967 1616 1612
-- Name: pacientehospitalizado_id_usuario_fkey; Type: FK CON-
STRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY pacientehospitalizado
    ADD CONSTRAINT pacientehospitalizado_id_usuario_fkey FOREIGN
KEY (id_usuario) REFERENCES usuario(id_usuario);

```

```
--
-- TOC entry 2011 (class 2606 OID 34228)
-- Dependencies: 1610 1624 1963
-- Name: recomendaciones_idpaciente_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY recomendaciones
    ADD CONSTRAINT recomendaciones_idpaciente_fkey FOREIGN KEY
(idpaciente) REFERENCES paciente(id_paciente);
```

```
--
-- TOC entry 2012 (class 2606 OID 34233)
-- Dependencies: 1967 1612 1624
-- Name: recomendaciones_idusuario_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY recomendaciones
    ADD CONSTRAINT recomendaciones_idusuario_fkey FOREIGN KEY
(idusuario) REFERENCES usuario(id_usuario);
```

```
--
-- TOC entry 1998 (class 2606 OID 34047)
-- Dependencies: 1929 1612 1577
-- Name: usuario_idescolaridad_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idescolaridad_fkey FOREIGN KEY
(idescolaridad) REFERENCES escolaridad(idescolaridad);
```

```
--
-- TOC entry 1999 (class 2606 OID 34052)
-- Dependencies: 1931 1579 1612
-- Name: usuario_idestadocivil_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idestadocivil_fkey FOREIGN KEY
(idestadocivil) REFERENCES estadocivil(idestadocivil);
```

```
--
-- TOC entry 2000 (class 2606 OID 34057)
-- Dependencies: 1933 1581 1612
-- Name: usuario_idnacionalidad_fkey; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```



```

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idnacionalidad_fkey FOREIGN KEY (id-
nacionalidad) REFERENCES nacionalidad(idnacionalidad);

--
-- TOC entry 2001 (class 2606 OID 34062)
-- Dependencies: 1945 1612 1593
-- Name: usuario_idrol_fkey; Type: FK CONSTRAINT; Schema: pub-
lic; Owner: postgres
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idrol_fkey FOREIGN KEY (idrol) REFER-
ENCES roles(id_rol);

--
-- TOC entry 2003 (class 2606 OID 34072)
-- Dependencies: 1608 1612 1961
-- Name: usuario_idsexo_fkey; Type: FK CONSTRAINT; Schema: pub-
lic; Owner: postgres
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idsexo_fkey FOREIGN KEY (idsexo) REF-
ERENCES sexo(id_sexo);

--
-- TOC entry 2002 (class 2606 OID 34067)
-- Dependencies: 1612 1587 1939
-- Name: usuario_idturno_fkey; Type: FK CONSTRAINT; Schema: pub-
lic; Owner: postgres
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idturno_fkey FOREIGN KEY (idturno)
REFERENCES turno(idturno);

--
-- TOC entry 2004 (class 2606 OID 34077)
-- Dependencies: 1941 1612 1589
-- Name: usuario_idunidad_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_idunidad_fkey FOREIGN KEY (idunidad)
REFERENCES unidad(idunidad);

--

```

```
-- TOC entry 2019 (class 0 OID 0)
-- Dependencies: 3
-- Name: public; Type: ACL; Schema: -; Owner: postgres
--
```

```
REVOKE ALL ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON SCHEMA public FROM postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO PUBLIC;
```

```
-- Completed on 2010-09-09 07:51:29
```

```
--
-- PostgreSQL database dump complete
--
```

Implementación

ProyectoTerminal/ece/data

ComentarioDTO.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table(name="comentario")
public class ComentarioDTO implements Serializable
{

    private long _id;
    private long _idUseruario;
    private int _idNota;
    private String _comentario;

    @Id @Column(name="id_comentario")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public long getId()
    {
        return _id;
    }
    @Column(name="idusuuario")
    public long getIdUsuario()
    {
        return _idUseruario;
    }
    @Column(name="idnota")
    public int getIdNota()
    {
        return _idNota;
    }
    @Column(name="comentario")
    public String getComentario()
    {
        return _comentario;
    }
    public void setId( long id )
    {
        _id = id;
    }
    public void setIdUsuario( long idUsuario )
    {
        _idUseruario = idUsuario;
    }
    public void setIdNota( int idNota )
    {

```

```

    _idNota = idNota;
}
public void setComentario( String comentario )
{
    _comentario = comentario;
}
}
}

```

ConsultorioDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table(name="consultorio")
public class ConsultorioDTO implements Serializable
{
    private int _id;
    private int _numero;
    private String _medico;
    private UnidadDTO _unidad;

    @Id @Column(name="idconsultorio")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column (name="numero")
    public int getNumero()
    {
        return _numero;
    }
    @Column (name="medicoresponsable")
    public String getMedico()
    {
        return _medico;
    }
    @OneToOne @JoinColumn(name="id_unidad")
    public UnidadDTO getUnidad()
    {
        return _unidad;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setNumero( int numero )
    {
        _numero = numero;
    }
}

```

```

    }
    public void setMedico( String medico )
    {
        _medico = medico;
    }
    public void setUnidad( UnidadDTO unidad )
    {
        _unidad = unidad;
    }
}

```

DetalleRecomendacionesDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.GenerationType.IDENTITY;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="detalle_recomendacion")
@Proxy( lazy=false )
public class DetalleRecomendacionesDTO implements Serializable
{
    private long _idDetalle;
    private int _idRecomendaciones;
    private String _medicamento;
    private String _modoUso;
    private String _recomendacion;

    @Id @Column(name="id_detalle_recomendacion") @GeneratedValue(
        strategy=IDENTITY )
    public long getIdDetalle()
    {
        return _idDetalle;
    }
    @Column(name="idrecomendaciones")
    public int getIdRecomendaciones()
    {
        return _idRecomendaciones;
    }
    @Column(name="medicamento")
    public String getMedicamento()
    {
        return _medicamento;
    }
    @Column(name="modouso")
    public String getModoUso()
    {

```

```

        return _modoUso;
    }
    @Column(name="recomendacion")
    public String getRecomendacion()
    {
        return _recomendacion;
    }
    public void setIdDetalle( long idDetalle )
    {
        _idDetalle = idDetalle;
    }
    public void setIdRecomendaciones( int idRecomendaciones )
    {
        _idRecomendaciones = idRecomendaciones;
    }
    public void setMedicamento( String medicamento )
    {
        _medicamento = medicamento;
    }
    public void setModoUso( String modoUso )
    {
        _modoUso = modoUso;
    }
    public void setRecomendacion( String recomendacion )
    {
        _recomendacion = recomendacion;
    }
}

```

EscolaridadDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="escolaridad")
public class EscolaridadDTO implements Serializable
{
    private int _id;
    private String _nombre;

    @Id @Column (name="idescolaridad")
    @GeneratedValue (strategy= GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column (name="escolaridad")
    public String getNombre()
    {
        return _nombre;
    }
}

```

```

    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }
}

```

EspecialidadDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="especialidad")
public class EspecialidadDTO implements Serializable
{
    private int _id;
    private String _especialidad;

    @Id @Column(name="idespecialidad")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }

    @Column (name="especialidad")
    public String getEspecialidad()
    {
        return _especialidad;
    }

    public void setId( int id )
    {
        _id = id;
    }

    public void setEspecialidad( String especialidad )
    {
        _especialidad = especialidad;
    }
}

```

EstadoCivilDTO.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table(name="estadocivil")
public class EstadoCivilDTO implements Serializable
{
    private int _id;
    private String _estadoCivil;

    @Id @Column(name="idestadocivil")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column(name="estadocivil")
    public String getEstadoCivil()
    {
        return _estadoCivil;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setEstadoCivil( String estadoCivil )
    {
        _estadoCivil = estadoCivil;
    }
}
```

FamiliarResponsableDTO.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table(name="familiar_responsable")
```



```

public class FamiliarResponsableDTO implements Serializable
{
    private int _id;
    private String _nombre;
    private String _apellidoPaterno;
    private String _apellidoMaterno;
    private String _direccion;
    private String _telefono;
    private String _parentezco;

    /**
     * @param nombreCompleto
     */

    @Id @Column(name="idfamiliarresponsable")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column (name="nombre")
    public String getNombre()
    {
        return _nombre;
    }
    @Column (name="apellidoPaterno")
    public String getApellidoPaterno()
    {
        return _apellidoPaterno;
    }
    @Column (name="apellidoMaterno")
    public String getApellidoMaterno()
    {
        return _apellidoMaterno;
    }
    @Column (name="direccion")
    public String getDireccion()
    {
        return _direccion;
    }
    @Column (name="telefono")
    public String getTelefono()
    {
        return _telefono;
    }
    @Column (name="parentezco")
    public String getParentezco()
    {
        return _parentezco;
    }
    public void setParentezco( String parentezco )
    {
        _parentezco = parentezco;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setNombre( String nombre )

```

```

    {
        _nombre = nombre;
    }
    public void setApellidoPaterno( String apellidoPaterno )
    {
        _apellidoPaterno = apellidoPaterno;
    }
    public void setApellidoMaterno( String apellidoMaterno )
    {
        _apellidoMaterno = apellidoMaterno;
    }
    public void setDireccion( String direccion )
    {
        _direccion = direccion;
    }
    public void setTelefono( String telefono )
    {
        _telefono = telefono;
    }
}

```

NacionalidadDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="nacionalidad")
public class NacionalidadDTO implements Serializable
{
    private int _id;
    private String _nacionalidad;

    @Id @Column (name="idnacionalidad")
    @GeneratedValue (strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }

    @Column (name="nacionalidad")
    public String getNacionalidad()
    {
        return _nacionalidad;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setNacionalidad( String nacionalidad )
    {

```

```

        _nacionalidad = nacionalidad;
    }

}

```

NotaAnestesiologiaDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 *
 */
@Entity @Table (name="nacionalidad")
public class NacionalidadDTO implements Serializable
{
    private int _id;
    private String _nacionalidad;

    @Id @Column (name="idnacionalidad")
    @GeneratedValue (strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }

    @Column (name="nacionalidad")
    public String getNacionalidad()
    {
        return _nacionalidad;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setNacionalidad( String nacionalidad )
    {
        _nacionalidad = nacionalidad;
    }

}

```

NotaAnestesiologiaJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

```

```

/**
 * @author Daniel
 *
 */
@Entity @Table (name="notaanestesiologia")
@PrimaryKeyJoinColumn (name="idnota")
@Proxy( lazy=false )
public class NotaAnestesiologiaJoinDTO extends NotaJoinDTO
{
    private SignosVitalesDTO _idVitales;
    private String _horaInicioAnestesiologia;
    private String _duracionAnestesiologia;
    private String _horaInicioQuirurgico;
    private String _duracionQuirurgico;
    private String _contigencias;
    private String _estadoPaciente;
    private String _medicamentos;
    private String _tecnicUtilizada;
    private String _liquidosIntravenosos;
    private String _pronostico;

    @OneToOne @JoinColumn (name="idvitales")
    public SignosVitalesDTO getIdVitales()
    {
        return _idVitales;
    }
    @Column (name="horainicioanestesiologia")
    public String getHoraInicioAnestesiologia()
    {
        return _horaInicioAnestesiologia;
    }
    @Column (name="duracionanestesiologia")
    public String getDuracionAnestesiologia()
    {
        return _duracionAnestesiologia;
    }
    @Column (name="horainicioquirurgico")
    public String getHoraInicioQuirurgico()
    {
        return _horaInicioQuirurgico;
    }
    @Column (name="duracionquirurgico")
    public String getDuracionQuirurgico()
    {
        return _duracionQuirurgico;
    }
    @Column (name="contigencias")
    public String getContigencias()
    {
        return _contigencias;
    }
    @Column (name="estadopaciente")
    public String getEstadoPaciente()
    {
        return _estadoPaciente;
    }
    @Column (name="medicamentos")
    public String getMedicamentos()
    {
        return _medicamentos;
    }
}

```

```

}
@Column (name="tecnicautilizada")
public String getTecnicaUtilizada()
{
    return _tecnicaUtilizada;
}
@Column (name="liquidosintravenosos")
public String getLiquidosIntravenosos()
{
    return _liquidosIntravenosos;
}

@Column (name="pronostico")
public String getPronostico()
{
    return _pronostico;
}

public void setIdVitales( SignosVitalesDTO idVitales )
{
    _idVitales = idVitales;
}
public void setHoraInicioAnestesiologia( String
    horaInicioAnestesiologia )
{
    _horaInicioAnestesiologia = horaInicioAnestesiologia;
}
public void setDuracionAnestesiologia( String duracionAnestesiologia
    )
{
    _duracionAnestesiologia = duracionAnestesiologia;
}
public void setHoraInicioQuirurgico( String horaInicioQuirurgico )
{
    _horaInicioQuirurgico = horaInicioQuirurgico;
}
public void setDuracionQuirurgico( String duracionQuirurgico )
{
    _duracionQuirurgico = duracionQuirurgico;
}
public void setContigencias( String contigencias )
{
    _contigencias = contigencias;
}
public void setEstadoPaciente( String estadoPaciente )
{
    _estadoPaciente = estadoPaciente;
}
public void setMedicamentos( String medicamentos )
{
    _medicamentos = medicamentos;
}
public void setTecnicaUtilizada( String tecnicaUtilizada )
{
    _tecnicaUtilizada = tecnicaUtilizada;
}
public void setLiquidosIntravenosos( String liquidosIntravenosos )
{
    _liquidosIntravenosos = liquidosIntravenosos;
}

```

```

    public void setPronostico( String pronostico )
    {
        _pronostico = pronostico;
    }

}

```

NotaDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.GenerationType.IDENTITY;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

/**
 * @author Daniel
 */

@Entity @Table (name="nota")
@Inheritance (strategy=InheritanceType.JOINED)
@Proxy( lazy=false )

public class NotaDTO implements Serializable
{

    private long _idNota;
    private long _idPacienteHospitalizado;
    private long _idUsuario;
    private Calendar _fechaRegistro = Calendar.getInstance();
    private String _diagnostico;
    private String _nombreNota;
    private String _estadoSalud;
    private String _pronostico;

    @Id @Column (name="id_nota") @GeneratedValue ( strategy=IDENTITY )
    public long getIdNota ()
    {
        return _idNota;
    }
    @Column (name="idpacientehospitalizado")
    public long getIdPacienteHospitalizado ()
    {
        return _idPacienteHospitalizado;
    }
    @Column (name="idusuario")
    public long getIdUsuario ()
    {
        return _idUsuario;
    }
    @Column (name="fecharegistro")

```

```

public Calendar getFechaRegistro()
{
    return _fechaRegistro;
}
@Column(name="diagnostico")
public String getDiagnostico()
{
    return _diagnostico;
}
@Column(name="nombrenota")
public String getNombreNota()
{
    return _nombreNota;
}

@Column(name="estadosalud")
public String getEstadoSalud()
{
    return _estadoSalud;
}
@Column(name="pronostico")
public String getPronostico()
{
    return _pronostico;
}
public void setEstadoSalud( String estadoSalud )
{
    _estadoSalud = estadoSalud;
}
public void setPronostico( String pronostico )
{
    _pronostico = pronostico;
}
public void setNombreNota( String nombreNota )
{
    _nombreNota = nombreNota;
}
public void setIdNota( long idNota )
{
    _idNota = idNota;
}
public void setIdPacienteHospitalizado( long idPacienteHospitalizado
)
{
    _idPacienteHospitalizado = idPacienteHospitalizado;
}
public void setIdUsuario( long idUsuario )
{
    _idUsuario = idUsuario;
}
public void setFechaRegistro( Calendar fechaRegistro )
{
    _fechaRegistro = fechaRegistro;
}
public void setDiagnostico( String diagnostico )
{
    _diagnostico = diagnostico;
}
}

```

NotaIngresoDTO.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="notaingreso")
@PrimaryKeyJoinColumn (name="idnota")
@Proxy( lazy=false )
public class NotaIngresoDTO extends NotaDTO
{

    private long _idVitales;
    private String _motivoIngreso;
    private String _antecedentes;
    private String _padecimiento;
    private String _exploracionFisica;
    private String _examenenes;

    @Column (name="idvitales")
    public long getIdVitales ()
    {
        return _idVitales;
    }
    @Column (name="motivoingreso")
    public String getMotivoIngreso ()
    {
        return _motivoIngreso;
    }
    @Column (name="antecedentes")
    public String getAntecedentes ()
    {
        return _antecedentes;
    }
    @Column (name="padecimiento")
    public String getPadecimiento ()
    {
        return _padecimiento;
    }
    @Column (name="exploracionfisica")
    public String getExploracionFisica ()
    {
        return _exploracionFisica;
    }
    @Column (name="examenenes")
    public String getExamenenes ()
    {
        return _examenenes;
    }
}
```



```

public void setIdVitales( long idVitales )
{
    _idVitales = idVitales;
}
public void setMotivoIngreso( String motivoIngreso )
{
    _motivoIngreso = motivoIngreso;
}
public void setAntecedentes( String antecedentes )
{
    _antecedentes = antecedentes;
}
public void setPadecimiento( String padecimiento )
{
    _padecimiento = padecimiento;
}
public void setExploracionFisica( String exploracionFisica )
{
    _exploracionFisica = exploracionFisica;
}
public void setExámenes( String exámenes )
{
    _exámenes = exámenes;
}
}

```

NotaIngresoJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 *
 */
@Entity @Table(name="notaingreso")
@PrimaryKeyJoinColumn(name="idnota")
@Proxy( lazy=false )
public class NotaIngresoDTO extends NotaDTO
{
    private long _idVitales;
    private String _motivoIngreso;
    private String _antecedentes;
    private String _padecimiento;
    private String _exploracionFisica;
    private String _exámenes;

    @Column (name="idvitales")
    public long getIdVitales()
    {
        return _idVitales;
    }
}

```

```

    }
    @Column (name="motivoIngreso")
    public String getMotivoIngreso()
    {
        return _motivoIngreso;
    }
    @Column (name="antecedentes")
    public String getAntecedentes()
    {
        return _antecedentes;
    }
    @Column (name="padecimiento")
    public String getPadecimiento()
    {
        return _padecimiento;
    }
    @Column (name="exploracionfisica")
    public String getExploracionFisica()
    {
        return _exploracionFisica;
    }
    @Column (name="examenes")
    public String getExamenes()
    {
        return _examenes;
    }

    public void setIdVitales( long idVitales )
    {
        _idVitales = idVitales;
    }
    public void setMotivoIngreso( String motivoIngreso )
    {
        _motivoIngreso = motivoIngreso;
    }
    public void setAntecedentes( String antecedentes )
    {
        _antecedentes = antecedentes;
    }
    public void setPadecimiento( String padecimiento )
    {
        _padecimiento = padecimiento;
    }
    public void setExploracionFisica( String exploracionFisica )
    {
        _exploracionFisica = exploracionFisica;
    }
    public void setExamenes( String examenes )
    {
        _examenes = examenes;
    }
}

```

NotaInterconsultaDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

```

```

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

/**
 * @author Daniel
 *
 */
@Entity @Table(name="notainterconsulta")
@PrimaryKeyJoinColumn(name="idnota")
@Proxy( lazy=false )
public class NotaInterconsultaDTO extends NotaDTO
{
    private String _antecedentes;
    private String _padecimientoActual;
    private String _exploracionFisica;
    private String _estudiosGabinete;
    private String _recomendacionesManejo;
    private long _idVitales;

    @Column (name="antecedentes")
    public String getAntecedentes()
    {
        return _antecedentes;
    }
    @Column (name="padecimientoactual")
    public String getPadecimientoActual()
    {
        return _padecimientoActual;
    }
    @Column (name="exploracionfisica")
    public String getExploracionFisica()
    {
        return _exploracionFisica;
    }
    @Column (name="estudios")
    public String getEstudiosGabinete()
    {
        return _estudiosGabinete;
    }
    @Column (name="recomendaciones")
    public String getRecomendacionesManejo()
    {
        return _recomendacionesManejo;
    }
    @Column (name="idvitales")
    public long getIdVitales()
    {
        return _idVitales;
    }
    public void setAntecedentes( String antecedentes )
    {
        _antecedentes = antecedentes;
    }
    public void setPadecimientoActual( String padecimientoActual )
    {
        _padecimientoActual = padecimientoActual;
    }
    public void setExploracionFisica( String exploracionFisica )
    {
        _exploracionFisica = exploracionFisica;
    }
}

```

```

    }
    public void setEstudiosGabinete( String estudiosGabinete )
    {
        _estudiosGabinete = estudiosGabinete;
    }
    public void setRecomendacionesManejo( String recomendacionesManejo )
    {
        _recomendacionesManejo = recomendacionesManejo;
    }
    public void setIdVitales( long idvitales )
    {
        _idVitales = idvitales;
    }
}

```

NotaInterconsultaJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

/**
 * @author Daniel
 */
@Entity @Table(name="notainterconsulta")
@PrimaryKeyJoinColumn(name="idnota")
@Proxy( lazy=false )
public class NotaInterconsultaJoinDTO extends NotaJoinDTO
{
    private String _antecedentes;
    private String _padecimientoActual;
    private String _exploracionFisica;
    private String _estudiosGabinete;
    private String _recomendacionesManejo;
    private SignosVitalesDTO _idVitales;

    @Column (name="antecedentes")
    public String getAntecedentes()
    {
        return _antecedentes;
    }
    @Column (name="padecimientoactual")
    public String getPadecimientoActual()
    {
        return _padecimientoActual;
    }
    @Column (name="exploracionfisica")
    public String getExploracionFisica()
    {
        return _exploracionFisica;
    }
    @Column (name="estudios")
    public String getEstudiosGabinete()

```

```

    {
        return _estudiosGabinete;
    }
    @Column (name="recomendaciones")
    public String getRecomendacionesManejo()
    {
        return _recomendacionesManejo;
    }
    @OneToOne @JoinColumn(name="idvitaless")
    public SignosVitalesDTO getIdVitales()
    {
        return _idVitales;
    }
    public void setAntecedentes( String antecedentes )
    {
        _antecedentes = antecedentes;
    }
    public void setPadecimientoActual( String padecimientoActual )
    {
        _padecimientoActual = padecimientoActual;
    }
    public void setExploracionFisica( String exploracionFisica )
    {
        _exploracionFisica = exploracionFisica;
    }
    public void setEstudiosGabinete( String estudiosGabinete )
    {
        _estudiosGabinete = estudiosGabinete;
    }
    public void setRecomendacionesManejo( String recomendacionesManejo )
    {
        _recomendacionesManejo = recomendacionesManejo;
    }
    public void setIdVitales( SignosVitalesDTO idvitaless )
    {
        _idVitales = idvitaless;
    }
}
}

```

NotaJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.GenerationType.IDENTITY;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

/**
 * @author Daniel
 *
 */

```

```

@Entity @Table (name="nota")
@Inheritance (strategy=InheritanceType.JOINED)
@Proxy( lazy=false )

public class NotaJoinDTO implements Serializable
{

    private long _idNota;
    private PacienteHospitalizadoJoinDTO _idPacienteHospitalizado;
    private UsuarioJoinDTO _idUser;
    private Calendar _fechaRegistro = Calendar.getInstance();
    private String _diagnostico;
    private String _nombreNota;
    private String _estadoSalud;
    private String _pronostico;

    @Id @Column(name="id_nota") @GeneratedValue( strategy=IDENTITY )
    public long getIdNota()
    {
        return _idNota;
    }
    @OneToOne @JoinColumn(name="idpacientehospitalizado")
    public PacienteHospitalizadoJoinDTO getIdPacienteHospitalizado()
    {
        return _idPacienteHospitalizado;
    }
    @OneToOne @JoinColumn(name="idusuario")
    public UsuarioJoinDTO getIdUsuario()
    {
        return _idUser;
    }
    @Column(name="fecharegistro")
    public Calendar getFechaRegistro()
    {
        return _fechaRegistro;
    }
    @Column(name="diagnostico")
    public String getDiagnostico()
    {
        return _diagnostico;
    }
    @Column (name="nombrenota")
    public String getNombreNota()
    {
        return _nombreNota;
    }

    @Column(name="estadosalud")
    public String getEstadoSalud()
    {
        return _estadoSalud;
    }
    @Column (name="pronostico")
    public String getPronostico()
    {
        return _pronostico;
    }
    public void setEstadoSalud( String estadoSalud )
    {
        _estadoSalud = estadoSalud;
    }
}

```

```

    }
    public void setPronostico( String pronostico )
    {
        _pronostico = pronostico;
    }
    public void setNombreNota( String nombreNota )
    {
        _nombreNota = nombreNota;
    }
    public void setIdNota( long idNota )
    {
        _idNota = idNota;
    }
    public void setIdPacienteHospitalizado( PacienteHospitalizadoJoinDTO
        idPacienteHospitalizado )
    {
        _idPacienteHospitalizado = idPacienteHospitalizado;
    }
    public void setIdUsuario( UsuarioJoinDTO idUsuario )
    {
        _idUsuario = idUsuario;
    }
    public void setFechaRegistro( Calendar fechaRegistro )
    {
        _fechaRegistro = fechaRegistro;
    }
    public void setDiagnostico( String diagnostico )
    {
        _diagnostico = diagnostico;
    }
}

```

OcupacionDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 */
@Entity @Table (name="ocupacion")
public class OcupacionDTO implements Serializable
{
    private int _id;
    private String _ocupacion;

    @Id @Column(name="idocupacion")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }

    @Column(name="ocupacion")

```

```

public String getOcupacion()
{
    return _ocupacion;
}
public void setId( int id )
{
    _id = id;
}
public void setOcupacion( String ocupacion )
{
    _ocupacion = ocupacion;
}
}

```

PacienteDTO.java

```

package mx.uam.azc.proyectoTerminal.ece.data;

import java.io.Serializable;
import java.util.Calendar;

import javax.persistence.*;

import org.hibernate.annotations.Proxy;
import static javax.persistence.GenerationType.IDENTITY;

@Entity @Table (name="paciente")
@Proxy( lazy=false )
public class PacienteDTO implements Serializable
{

    private long _idPaciente;
    private int _idNacionalidad;
    private int _idReligion;
    private int _idEscolaridad;
    private int _estadoCivil;
    private int _ocupacion;
    private int _consultorio;
    private int _turno;
    private int _familiarResponsable;
    private String _nombre;
    private String _apellidoPaterno;
    private String _apellidoMaterno;
    private Calendar _fechaNacimiento = Calendar.getInstance();
    private int _edad;
    private int _sexo;
    private String _direccion;
    private String _lugarOrigen;
    private String _etnicidad;
    private boolean _estatus;
    private boolean _estado = true;
    private String _telefono;
    private String _curp;
    private int _version;

    @Id @Column ( name="id_paciente" ) @GeneratedValue( strategy=IDENTITY )
    public long getIdPaciente()
    {

```



```

        return _idPaciente;
    }
    @Column(name="idnacionalidad")
    public int getIdNacionalidad()
    {
        return _idNacionalidad;
    }
    @Column(name="idreligion")
    public int getIdReligion()
    {
        return _idReligion;
    }
    @Column(name="idescolaridad")
    public int getIdEscolaridad()
    {
        return _idEscolaridad;
    }
    @Column(name="idestadocivil")
    public int getEstadoCivil()
    {
        return _estadoCivil;
    }
    @Column(name="idocupacion")
    public int getOcupacion()
    {
        return _ocupacion;
    }
    @Column(name="idconsultorio")
    public int getConsultorio()
    {
        return _consultorio;
    }
    @Column(name="idturno")
    public int getTurno()
    {
        return _turno;
    }
    @Column(name="idfamiliarresponsable")
    public int getFamiliarResponsable()
    {
        return _familiarResponsable;
    }

    @Column(name="nombre")
    public String getNombre()
    {
        return _nombre;
    }
    @Column(name="apellidopaterno")
    public String getApellidoPaterno()
    {
        return _apellidoPaterno;
    }
    @Column(name="apellidomaterno")
    public String getApellidoMaterno()
    {
        return _apellidoMaterno;
    }
    @Column(name="fechanacimiento")
    public Calendar getFechaNacimiento()
    {

```

```

    return _fechaNacimiento;
}
@Transient
public int getEdad()
{
    return _edad;
}
@Column (name="idsexo")
public int getSexo()
{
    return _sexo;
}
@Column (name="direccion")
public String getDireccion()
{
    return _direccion;
}
@Column (name="lugarorigen")
public String getLugarOrigen()
{
    return _lugarOrigen;
}
@Column (name="etnicidad")
public String getEtnicidad()
{
    return _etnicidad;
}
@Column (name="estado")
public boolean isEstado()
{
    return _estado;
}

@Column (name="estatus")
public boolean isEstatus()
{
    return _estatus;
}
@Column (name="telefono")
public String getTelefono()
{
    return _telefono;
}
@Column (name="curp")
public String getCurp()
{
    return _curp;
}

@Column(name="version")
public int getVersion()
{
    return _version;
}
public void setVersion( int version )
{
    _version = version;
}
public void setTelefono(String telefono)
{
    _telefono = telefono;
}

```

```

    }

    public void setCurp(String curp)
    {
        _curp = curp;
    }
public void setIdPaciente( long idPaciente )
{
    _idPaciente = idPaciente;
}
public void setIdNacionalidad( int idNacionalidad )
{
    _idNacionalidad = idNacionalidad;
}
public void setIdReligion( int idReligion )
{
    _idReligion = idReligion;
}
public void setIdEscolaridad( int idEscolaridad )
{
    _idEscolaridad = idEscolaridad;
}
public void setEstadoCivil( int estadoCivil )
{
    _estadoCivil = estadoCivil;
}
public void setOcupacion( int ocupacion )
{
    _ocupacion = ocupacion;
}
public void setConsultorio( int consultorio )
{
    _consultorio = consultorio;
}
public void setTurno( int turno )
{
    _turno = turno;
}

public void setNombre( String nombre )
{
    _nombre = nombre;
}
public void setApellidoPaterno( String apellidoPaterno )
{
    _apellidoPaterno = apellidoPaterno;
}
public void setApellidoMaterno( String apellidoMaterno )
{
    _apellidoMaterno = apellidoMaterno;
}
public void setFechaNacimiento( Calendar fechaNacimiento )
{
    _fechaNacimiento = fechaNacimiento;
}
public void setEdad( int edad )
{
    _edad = edad;
}
public void setSexo( int sexo )
{

```

```

    _sexo = sexo;
}
public void setDireccion( String direccion )
{
    _direccion = direccion;
}
public void setLugarOrigen( String lugarOrigen )
{
    _lugarOrigen = lugarOrigen;
}
public void setEtnicidad( String etnicidad )
{
    _etnicidad = etnicidad;
}
public void setEstatus( boolean estatus )
{
    _estatus = estatus;
}
public void setEstado(boolean estado)
{
    _estado = estado;
}
public void setFamiliarResponsable(int familiarResponsable)
{
    _familiarResponsable = familiarResponsable;
}
public int calcularEdad(Calendar fechaNacimiento)
{
    int años = 0;
    Calendar hoy = Calendar.getInstance();

    años = hoy.get( Calendar.YEAR ) -
    fechaNacimiento.get( Calendar.YEAR );

    if(hoy.get( Calendar.MONTH ) < fechaNacimiento.get( Calendar.-
    MONTH ))
        años = años - 1;
    else if(hoy.get( Calendar.MONTH ) == fechaNacimiento.get( Calendar.-
    MONTH ))
    {
        if (hoy.get( Calendar.DAY_OF_MONTH ) < fechaNacimiento.get( Cal-
        endar.DAY_OF_MONTH ))
            años = años - 1;
    }

    return años;
}
}

```

PacienteHospitalizadoDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

```

```

/**
 * @author Daniel
 *
 */
@Entity @Table(name="pacientehospitalizado")
public class PacienteHospitalizadoDTO implements Serializable
{

    private long _id;
    private long _idPaciente;
    private long _idUsuario;
    private int _idEspecialidad;
    private Calendar _fechaAdmision = Calendar.getInstance();
    private String _diagnostico;
    private int _cama;

    @Id @Column (name="idpacientehospitalizado")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public long getId()
    {
        return _id;
    }
    @Column (name="id_paciente")
    public long getIdPaciente()
    {
        return _idPaciente;
    }
    @Column (name="id_usuario")
    public long getIdUsuario()
    {
        return _idUsuario;
    }
    @Column (name="id_especialidad")
    public int getIdEspecialidad()
    {
        return _idEspecialidad;
    }
    @Column (name="fechaadmision")
    public Calendar getFechaAdmision()
    {
        return _fechaAdmision;
    }
    @Column (name="diagnostico")
    public String getDiagnostico()
    {
        return _diagnostico;
    }
    @Column (name="cama")
    public int getCama()
    {
        return _cama;
    }
    public void setId( long id )
    {
        _id = id;
    }
    public void setIdPaciente( long idPaciente )
    {
        _idPaciente = idPaciente;
    }
    public void setIdUsuario( long idUsuario )

```

```

    {
        _idUserario = idUsuario;
    }
    public void setIdEspecialidad( int idEspecialidad )
    {
        _idEspecialidad = idEspecialidad;
    }
    public void setFechaAdmision( Calendar fechaAdmision )
    {
        _fechaAdmision = fechaAdmision;
    }
    public void setDiagnostico( String diagnostico )
    {
        _diagnostico = diagnostico;
    }
    public void setCama( int cama )
    {
        _cama = cama;
    }
}

```

PacienteHospitalizadoJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.FetchType.LAZY;

import javax.persistence.*;

import java.io.Serializable;
import java.util.*;

/**
 * @author Daniel
 */
@Entity @Table(name="pacientehospitalizado")
public class PacienteHospitalizadoJoinDTO implements Serializable,
    Comparable<NotaDTO>
{
    private long _id;
    private PacienteJoinDTO _idPaciente;
    private UsuarioJoinDTO _idUserario;
    private EspecialidadDTO _idEspecialidad;
    private Calendar _fechaAdmision = Calendar.getInstance();
    private String _diagnostico;
    private int _cama;
    private Collection<NotaDTO> _notas = new ArrayList<NotaDTO>();

    @Id @Column (name="idpacientehospitalizado")
    @GeneratedValue (strategy= GenerationType.IDENTITY)
    public long getId()
    {
        return _id;
    }
    @OneToOne @JoinColumn (name="id_paciente")

```

```

public PacienteJoinDTO getIdPaciente()
{
    return _idPaciente;
}
@OneToOne @JoinColumn (name="id_usuario")
public UsuarioJoinDTO getIdUsuario()
{
    return _idUsuario;
}
@OneToOne @JoinColumn (name="id_especialidad")
public EspecialidadDTO getIdEspecialidad()
{
    return _idEspecialidad;
}
@Column (name="fechaadmission")
public Calendar getFechaAdmision()
{
    return _fechaAdmision;
}
@Column (name="diagnostico")
public String getDiagnostico()
{
    return _diagnostico;
}
@Column (name="cama")
public int getCama()
{
    return _cama;
}
public void setId( long id )
{
    _id = id;
}
public void setIdPaciente( PacienteJoinDTO idPaciente )
{
    _idPaciente = idPaciente;
}
public void setIdUsuario( UsuarioJoinDTO idUsuario )
{
    _idUsuario = idUsuario;
}
public void setIdEspecialidad( EspecialidadDTO idEspecialidad )
{
    _idEspecialidad = idEspecialidad;
}
public void setFechaAdmision( Calendar fechaAdmision )
{
    _fechaAdmision = fechaAdmision;
}
public void setDiagnostico( String diagnostico )
{
    _diagnostico = diagnostico;
}
public void setCama( int cama )
{
    _cama = cama;
}
@OneToMany (fetch=LAZY)
@OrderBy ("fechaRegistro DESC")
@JoinColumn (name="idpacientehospitalizado")
public Collection<NotaDTO> getNotas()

```

```

    {
        return _notas;
    }
    public void setNotas( Collection<NotaDTO> notas )
    {
        _notas = notas;
    }
    /*
     * @author Daniel
     */
    @Override
    public int compareTo( NotaDTO nota )
    {
        if(nota.getIdPacienteHospitalizado() != _id)
        {
            return new Integer("" + (_id -
            nota.getIdPacienteHospitalizado())).intValue();
        }

        return 0;
    }
}

```

PacienteJoinDTO.java

```

package mx.uam.azc.proyectoTerminal.ece.data;

import java.io.Serializable;
import java.util.*;

import javax.persistence.*;

import org.hibernate.annotations.Proxy;
import static javax.persistence.GenerationType.IDENTITY;
import static javax.persistence.FetchType.LAZY;

@Entity @Table (name="paciente")
@Proxy (lazy=false)
public class PacienteJoinDTO implements Serializable,
    Comparable<RecomendacionesDTO>
{

    private long _idPaciente;
    private NacionalidadDTO _nacionalidad;
    private ReligionDTO _religion;
    private EscolaridadDTO _escolaridad;
    private EstadoCivilDTO _estadoCivil;
    private OcupacionDTO _ocupacion;
    private ConsultorioDTO _consultorio;
    private TurnoDTO _turno;
    private FamiliarResponsableDTO _familiarResponsable;
    private String _nombre;
    private String _apellidoPaterno;
    private String _apellidoMaterno;
    private Calendar _fechaNacimiento = Calendar.getInstance();
    private int _edad;
    private SexoDTO _sexo;
    private String _direccion;
    private String _lugarOrigen;
    private String _etnicidad;
}

```



```

private boolean _estatus;
private boolean _estado;
private String _telefono;
private String _curp;
private String _familiarNombreCompleto;
private Collection<PacienteHospitalizadoDTO>
_pacienteHospitalizados = new
ArrayList<PacienteHospitalizadoDTO>();
private Collection<RecomendacionesDTO> _recomendaciones = new
ArrayList<RecomendacionesDTO>();
private int _version;

@Id @Column ( name="id_paciente" ) @GeneratedValue( strategy=IDEN-
TITY )
public long getIdPaciente()
{
return _idPaciente;
}
@OneToOne @JoinColumn(name="idnacionalidad")
public NacionalidadDTO getNacionalidad()
{
return _nacionalidad;
}
@OneToOne @JoinColumn(name="idreligion")
public ReligionDTO getReligion()
{
return _religion;
}

@OneToOne @JoinColumn(name="idescolaridad")
public EscolaridadDTO getEscolaridad()
{
return _escolaridad;
}

@OneToOne @JoinColumn(name="idestadocivil")
public EstadoCivilDTO getEstadoCivil()
{
return _estadoCivil;
}

@OneToOne @JoinColumn(name="idocupacion")
public OcupacionDTO getOcupacion()
{
return _ocupacion;
}

@OneToOne @JoinColumn(name="idconsultorio")
public ConsultorioDTO getConsultorio()
{
return _consultorio;
}

@OneToOne @JoinColumn(name="idturno")
public TurnoDTO getTurno()
{
return _turno;
}

@OneToOne @JoinColumn(name="idfamiliarresponsable")
public FamiliarResponsableDTO getFamiliarResponsable()

```

```

{
    return _familiarResponsable;
}

@Column (name="nombre")
public String getNombre()
{
    return _nombre;
}
@Column (name="apellidopaterno")
public String getApellidoPaterno()
{
    return _apellidoPaterno;
}
@Column (name="apellidomaterno")
public String getApellidoMaterno()
{
    return _apellidoMaterno;
}
@Column (name="fechanacimiento")
public Calendar getFechaNacimiento()
{
    return _fechaNacimiento;
}
@Transient
public int getEdad()
{
    return _edad;
}
@OneToOne @JoinColumn (name="idsexo")
public SexoDTO getSexo()
{
    return _sexo;
}
@Column (name="direccion")
public String getDireccion()
{
    return _direccion;
}
@Column (name="lugarorigen")
public String getLugarOrigen()
{
    return _lugarOrigen;
}
@Column (name="etnicidad")
public String getEtnicidad()
{
    return _etnicidad;
}
@Column (name="estado")
public boolean isEstado()
{
    return _estado;
}
@Column (name="estatus")
public boolean isEstatus()
{
    return _estatus;
}
@Column (name="telefono")

```

```

public String getTelefono()
{
    return _telefono;
}
@Column (name="curp")
public String getCurp()
{
    return _curp;
}
@Column(name="version")
public int getVersion()
{
    return _version;
}
@OneToMany (fetch=LAZY)
@OrderBy ("fechaAdmision DESC")
@JoinColumn(name="id_paciente")
public Collection<PacienteHospitalizadoDTO>
getPacienteHospitalizados()
{
    return _pacienteHospitalizados;
}
@OneToMany (fetch=LAZY)
@OrderBy ("fecha DESC")
@JoinColumn(name="idpaciente")
public Collection<RecomendacionesDTO> getRecomendaciones()
{
    return _recomendaciones;
}
public void setRecomendaciones( Collection<RecomendacionesDTO>
recomendaciones )
{
    _recomendaciones = recomendaciones;
}
public void setPacienteHospitalizados(
Collection<PacienteHospitalizadoDTO> pacienteHospitalizados )
{
    _pacienteHospitalizados = pacienteHospitalizados;
}
public void setVersion( int version )
{
    _version = version;
}
public void setTelefono(String telefono)
{
    _telefono = telefono;
}

public void setCurp(String curp)
{
    _curp = curp;
}
public void setIdPaciente( long idPaciente )
{
    _idPaciente = idPaciente;
}

public void setNacionalidad( NacionalidadDTO nacionalidad )
{
    _nacionalidad = nacionalidad;
}

```

```

}

public void setReligion( ReligionDTO religion )
{
    _religion = religion;
}

public void setEscolaridad( EscolaridadDTO escolaridad )
{
    _escolaridad = escolaridad;
}

public void setEstadoCivil( EstadoCivilDTO estadoCivil )
{
    _estadoCivil = estadoCivil;
}

public void setOcupacion( OcupacionDTO ocupacion )
{
    _ocupacion = ocupacion;
}

public void setConsultorio( ConsultorioDTO consultorio )
{
    _consultorio = consultorio;
}

public void setTurno( TurnoDTO turno )
{
    _turno = turno;
}

public void setFamiliarResponsable( FamiliarResponsableDTO
    familiarResponsable )
{
    _familiarResponsable = familiarResponsable;
}

public void setNombre( String nombre )
{
    _nombre = nombre;
}
public void setApellidoPaterno( String apellidoPaterno )
{
    _apellidoPaterno = apellidoPaterno;
}
public void setApellidoMaterno( String apellidoMaterno )
{
    _apellidoMaterno = apellidoMaterno;
}
public void setFechaNacimiento( Calendar fechaNacimiento )
{
    _fechaNacimiento = fechaNacimiento;
}
public void setEdad( int edad )
{
    _edad = edad;
}
public void setSexo( SexoDTO sexo )
{
    _sexo = sexo;
}

```

```

}
public void setDireccion( String direccion )
{
    _direccion = direccion;
}
public void setLugarOrigen( String lugarOrigen )
{
    _lugarOrigen = lugarOrigen;
}
public void setEtnicidad( String etnicidad )
{
    _etnicidad = etnicidad;
}
public void setEstatus( boolean estatus )
{
    _estatus = estatus;
}
public void setEstado( boolean estado)
{
    _estado = estado;
}
@Transient
public String getFamiliarNombreCompleto()
{
    return _familiarNombreCompleto;
}
public void setFamiliarNombreCompleto( String familiarNombreCompleto
)
{
    _familiarNombreCompleto = familiarNombreCompleto;
}

public int compareTo( PacienteHospitalizadoDTO pacienteHospitalizado
)
{
    if ( pacienteHospitalizado.getIdPaciente() != _idPaciente)

        return new Integer( ""+ ( _idPaciente -
pacienteHospitalizado.getIdPaciente()) ).intValue();

    return 0;
}
public int compareTo( RecomendacionesDTO recomendaciones)
{
    if(recomendaciones.getIdPaciente() != _idPaciente)

        return new Integer( "" + ( _idPaciente -
recomendaciones.getIdPaciente()) ).intValue();

    return 0;
}
public void addPacienteHospitalizado( PacienteHospitalizadoDTO
pacienteHospitalizado )
{
    _pacienteHospitalizados.add( pacienteHospitalizado );
    pacienteHospitalizado.setIdPaciente( _idPaciente );
}

public void addRecomendaciones( RecomendacionesDTO recomendaciones )
{

```

```

        _recomendaciones.add( recomendaciones );
        recomendaciones.setIdPaciente( _idPaciente );
    }
    public int calcularEdad(Calendar fechaNacimiento)
    {
        int años = 0;
        Calendar hoy = Calendar.getInstance();

        años = hoy.get( Calendar.YEAR ) -
            fechaNacimiento.get( Calendar.YEAR );

        if(hoy.get( Calendar.MONTH ) < fechaNacimiento.get( Calendar.-
            MONTH ))
            años = años - 1;
        else if(hoy.get( Calendar.MONTH ) == fechaNacimiento.get( Calendar.-
            MONTH ))
        {
            if (hoy.get( Calendar.DAY_OF_MONTH ) < fechaNacimiento.get( Cal-
                endar.DAY_OF_MONTH ))
                años = años - 1;
        }

        return años;
    }
}

```

RecomendacionesDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.GenerationType.IDENTITY;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

/**
 * @author Daniel
 */
@Entity @Table (name="recomendaciones")
@Proxy( lazy=false )
public class RecomendacionesDTO implements Serializable
{

    private long _id;
    private long _idUsuario;
    private long _idPaciente;
    private Calendar _fecha = Calendar.getInstance();
    private boolean _estatus;

    @Id @Column(name="id_recomendaciones") @GeneratedValue(
        strategy=IDENTITY )
    public long getId()
    {

```

```

        return _id;
    }
    @Column (name="idusuario")
    public long getIdUsuario ()
    {
        return _idUseruario;
    }
    @Column (name="idpaciente")
    public long getIdPaciente ()
    {
        return _idPaciente;
    }
    @Column (name="fecha")
    public Calendar getFecha ()
    {
        return _fecha;
    }
    @Column (name="estatus")
    public boolean isEstatus ()
    {
        return _estatus;
    }
    public void setId( long id )
    {
        _id = id;
    }
    public void setIdUsuario( long idUsuario )
    {
        _idUseruario = idUsuario;
    }
    public void setIdPaciente( long idPaciente )
    {
        _idPaciente = idPaciente;
    }
    public void setFecha( Calendar fecha )
    {
        _fecha = fecha;
    }
    public void setEstatus( boolean estatus )
    {
        _estatus = estatus;
    }
}

```

RecomendacionesJoinDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import static javax.persistence.GenerationType.IDENTITY;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

```

```

/**
 * @author Daniel
 *
 */
@Entity @Table (name="recomendaciones")
@Proxy( lazy=false )
public class RecomendacionesDTO implements Serializable
{

    private long _id;
    private long _idUserario;
    private long _idPaciente;
    private Calendar _fecha = Calendar.getInstance();
    private boolean _estatus;

    @Id @Column(name="id_recomendaciones") @Generated-
    Value( strategy=IDENTITY )
    public long getId()
    {
        return _id;
    }
    @Column (name="idusuario")
    public long getIdUsuario()
    {
        return _idUserario;
    }
    @Column(name="idpaciente")
    public long getIdPaciente()
    {
        return _idPaciente;
    }
    @Column(name="fecha")
    public Calendar getFecha()
    {
        return _fecha;
    }
    @Column(name="estatus")
    public boolean isEstatus()
    {
        return _estatus;
    }
    public void setId( long id )
    {
        _id = id;
    }
    public void setIdUsuario( long idUsuario )
    {
        _idUserario = idUsuario;
    }
    public void setIdPaciente( long idPaciente )
    {
        _idPaciente = idPaciente;
    }
    public void setFecha( Calendar fecha )
    {
        _fecha = fecha;
    }
    public void setEstatus( boolean estatus )
    {
        _estatus = estatus;
    }
}

```



```
}  
  
}
```

ReligionDTO.java

```
/**  
 * @author Daniel  
 */  
package mx.uam.azc.proyectoTerminal.ece.data;  
  
import javax.persistence.*;  
  
import java.io.Serializable;  
  
/**  
 * @author Daniel  
 *  
 */  
@Entity @Table(name="religion")  
public class ReligionDTO implements Serializable  
{  
    private int _id;  
    private String _religion;  
  
    @Id @Column(name="idreligion")  
    @GeneratedValue(strategy=GenerationType.IDENTITY)  
    public int getId()  
    {  
        return _id;  
    }  
  
    @Column(name="religion")  
    public String getReligion()  
    {  
        return _religion;  
    }  
    public void setId( int id )  
    {  
        _id = id;  
    }  
    public void setReligion( String religion )  
    {  
        _religion = religion;  
    }  
  
}
```

RoIDTO.java

```
/**  
 * @author Daniel  
 */  
package mx.uam.azc.proyectoTerminal.ece.data;  
  
import javax.persistence.*;  
  
import java.io.Serializable;
```

```

/**
 * @author Daniel
 *
 */
@Entity @Table(name="roles")
public class RolDTO implements Serializable
{
    private int _id;
    private String _rol;

    @Id @Column(name="id_rol")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column (name="rol")
    public String getRol()
    {
        return _rol;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setRol( String rol )
    {
        _rol = rol;
    }
}

```

SexoDTO.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;
import javax.persistence.*;
import java.io.Serializable;

/**
 * @author Daniel
 *
 */
@Entity @Table (name="sexo")
@Proxy (lazy=false)
public class SexoDTO implements Serializable
{
    private int _idSexo;
    private String _sexo;
    @Id @Column(name="id_sexo")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getIdSexo()
    {
        return _idSexo;
    }
}

```

```

    }
    @Column(name="sexo")
    public String getSexo()
    {
        return _sexo;
    }
    public void setIdSexo( int idSexo )
    {
        _idSexo = idSexo;
    }
    public void setSexo( String sexo )
    {
        _sexo = sexo;
    }
}
}

```

SignosVitalesDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import java.io.Serializable;

import javax.persistence.*;

/**
 * @author Daniel
 */
@Entity @Table (name="signosvitales")
public class SignosVitalesDTO implements Serializable
{
    private long _id;
    private float _peso;
    private float _talla;
    private float _temperatura;
    private float _frecuenciaCardiaca;
    private float _frecuenciaRespiratoria;
    private float _tensionArterial;
    private float _imc;

    @Id @Column(name="id_signos_vitales")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public long getId()
    {
        return _id;
    }
    @Column(name="peso")
    public float getPeso()
    {
        return _peso;
    }
    @Column(name="talla")
    public float getTalla()
    {
        return _talla;
    }
}

```

```

@Column(name="temperatura")
public float getTemperatura()
{
    return _temperatura;
}
@Column(name="freccardiaca")
public float getFrecuenciaCardiaca()
{
    return _frecuenciaCardiaca;
}
@Column(name="frecrespiratoria")
public float getFrecuenciaRespiratoria()
{
    return _frecuenciaRespiratoria;
}
@Column(name="tensionarterial")
public float getTensionArterial()
{
    return _tensionArterial;
}
@Column(name="imc")
public float getImc()
{
    return _imc;
}
public void setId( long id )
{
    _id = id;
}
public void setPeso( float peso )
{
    _peso = peso;
}
public void setTalla( float talla )
{
    _talla = talla;
}
public void setTemperatura( float temperatura )
{
    _temperatura = temperatura;
}
public void setFrecuenciaCardiaca( float frecuenciaCardiaca )
{
    _frecuenciaCardiaca = frecuenciaCardiaca;
}
public void setFrecuenciaRespiratoria( float
frecuenciaRespiratoria )
{
    _frecuenciaRespiratoria = frecuenciaRespiratoria;
}
public void setTensionArterial( float tensionArterial )
{
    _tensionArterial = tensionArterial;
}
public void setImc( float imc )
{
    _imc = imc;
}

public float calcularIMC(float peso, float talla)
{

```

```

        float imc = 0;

        imc = peso / ( talla * talla);

        return imc;
    }
}

```

TurnoDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;

/**
 * @author Daniel
 *
 */
@Entity @Table(name="turno")
public class TurnoDTO implements Serializable
{
    private int _id;
    private String turno;

    @Id @Column(name="idturno")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }

    @Column (name="turno")
    public String getTurno()
    {
        return turno;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setTurno( String turno )
    {
        this.turno = turno;
    }
}

```

UnidadDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

```

```

import java.io.Serializable;

/**
 * @author Daniel
 *
 */

@Entity @Table(name="unidad")
public class UnidadDTO implements Serializable
{
    private int _id;
    private String _tipo;
    private String _nombre;
    private String _domicilio;
    private String _institucionPerteneiente;
    private String _razonSocial;

    @Id @Column(name="idunidad")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public int getId()
    {
        return _id;
    }
    @Column(name="tipo")
    public String getTipo()
    {
        return _tipo;
    }
    @Column(name="nombre")
    public String getNombre()
    {
        return _nombre;
    }
    @Column(name="domicilio")
    public String getDomicilio()
    {
        return _domicilio;
    }
    @Column(name="institucionperteneiente")
    public String getInstitucionPerteneiente()
    {
        return _institucionPerteneiente;
    }
    @Column(name="razonsocial")
    public String getRazonSocial()
    {
        return _razonSocial;
    }
    public void setId( int id )
    {
        _id = id;
    }
    public void setTipo( String tipo )
    {
        _tipo = tipo;
    }
    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }
}

```

```

public void setDomicilio( String domicilio )
{
    _domicilio = domicilio;
}
public void setInstitucionPerteneiente( String
    institucionPerteneiente )
{
    _institucionPerteneiente = institucionPerteneiente;
}
public void setRazonSocial( String razonSocial )
{
    _razonSocial = razonSocial;
}
}

```

UsuarioDTO.java

```

/**
 *
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

/**
 * @author Daniel
 *
 */
@Entity @Table(name="usuario")
public class UsuarioDTO implements Serializable
{
    private long _id;
    private int _escolaridad;
    private int _unidad;
    private int _estadoCivil;
    private int _turno;
    private int _nacionalidad;
    private int _rol;
    private String _nombre;
    private String _apellidoPaterno;
    private String _apellidoMaterno;
    private String _cedula;
    private String _direccion;
    private String _telefono;
    private int _sexo;
    private Calendar _fechaNacimiento = Calendar.getInstance();
    private int _edad;
    private String _tipo;
    private String _usuario;
    private String _contrasenia;
    private int _version;

    @Id @Column(name="id_usuario")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    public long getId()
    {

```

```

    return _id;
}
@Column (name="idescolaridad")
public int getEscolaridad()
{
    return _escolaridad;
}
@Column (name="idunidad")
public int getUnidad()
{
    return _unidad;
}
@Column (name="idestadocivil")
public int getEstadoCivil()
{
    return _estadoCivil;
}
@Column (name="idturno")
public int getTurno()
{
    return _turno;
}
@Column (name="idnacionalidad")
public int getNacionalidad()
{
    return _nacionalidad;
}
@Column (name="idrol")
public int getRol()
{
    return _rol;
}
@Column (name="nombre")
public String getNombre()
{
    return _nombre;
}
@Column (name="apellidopaterno")
public String getApellidoPaterno()
{
    return _apellidoPaterno;
}
@Column (name="apellidomaterno")
public String getApellidoMaterno()
{
    return _apellidoMaterno;
}
@Column (name="cedula")
public String getCedula()
{
    return _cedula;
}
@Column (name="direccion")
public String getDireccion()
{
    return _direccion;
}
@Column (name="telefono")
public String getTelefono()
{
    return _telefono;
}

```



```

}
@Column (name="idsexo")
public int getSexo()
{
    return _sexo;
}
@Column (name="fechanacimiento")
public Calendar getFechaNacimiento()
{
    return _fechaNacimiento;
}

@Transient
public int getEdad()
{
    return _edad;
}
@Column (name="tipo")
public String getTipo()
{
    return _tipo;
}
@Column (name="usuario")
public String getUsuario()
{
    return _usuario;
}
@Column (name="contrasenia")
public String getContrasenia()
{
    return _contrasenia;
}
@Version
@Column (name="version")
public int getVersion()
{
    return _version;
}
public void setVersion( int version )
{
    _version = version;
}
public void setId( long id )
{
    _id = id;
}
public void setEscolaridad( int escolaridad )
{
    _escolaridad = escolaridad;
}
public void setUnidad( int unidad )
{
    _unidad = unidad;
}
public void setEstadoCivil( int estadoCivil )
{
    _estadoCivil = estadoCivil;
}
public void setTurno( int turno )
{
    _turno = turno;
}

```

```

}
public void setNacionalidad( int nacionalidad )
{
    _nacionalidad = nacionalidad;
}
public void setRol( int rol )
{
    _rol = rol;
}
public void setNombre( String nombre )
{
    _nombre = nombre;
}
public void setApellidoPaterno( String apellidoPaterno )
{
    _apellidoPaterno = apellidoPaterno;
}
public void setApellidoMaterno( String apellidoMaterno )
{
    _apellidoMaterno = apellidoMaterno;
}
public void setCedula( String cedula )
{
    _cedula = cedula;
}
public void setDireccion( String direccion )
{
    _direccion = direccion;
}
public void setTelefono( String telefono )
{
    _telefono = telefono;
}
public void setSexo( int sexo )
{
    _sexo = sexo;
}
public void setFechaNacimiento( Calendar fechaNacimiento )
{
    _fechaNacimiento = fechaNacimiento;
}
public void setEdad( int edad )
{
    _edad = edad;
}
public void setTipo( String tipo )
{
    _tipo = tipo;
}
public void setUsuario( String usuario )
{
    _usuario = usuario;
}
public void setContrasenia( String contrasenia )
{
    _contrasenia = contrasenia;
}

public int calcularEdad(Calendar fechaNacimiento)
{
    int años = 0;
}

```

```

    Calendar hoy = Calendar.getInstance();

    años = hoy.get( Calendar.YEAR ) -
    fechaNacimiento.get( Calendar.YEAR );

    if(hoy.get( Calendar.MONTH ) < fechaNacimiento.get( Calendar.-
    MONTH ))
        años = años - 1;
    else if(hoy.get( Calendar.MONTH ) == fechaNacimiento.get( Calendar.-
    MONTH ))
    {
        if (hoy.get( Calendar.DAY_OF_MONTH ) < fechaNacimiento.get( Cal-
        endar.DAY_OF_MONTH ))
            años = años - 1;
        }

    return años;
}
}

```

UsuarioJoinDTO.java

```

/**
 *
 */
package mx.uam.azc.proyectoTerminal.ece.data;

import org.hibernate.annotations.Proxy;

import javax.persistence.*;

import java.io.Serializable;
import java.util.Calendar;

import static javax.persistence.GenerationType.IDENTITY;
/**
 * @author Daniel
 *
 */
@Entity @Table(name="usuario")
@Proxy( lazy=false )
public class UsuarioJoinDTO implements Serializable
{
    private long _id;
    private EscolaridadDTO _escolaridad;
    private UnidadDTO _unidad;
    private EstadoCivilDTO _estadoCivil;
    private TurnoDTO _turno;
    private NacionalidadDTO _nacionalidad;
    private RolDTO _rol;
    private String _nombre;
    private String _apellidoPaterno;
    private String _apellidoMaterno;
    private String _cedula;
    private String _direccion;
    private String _telefono;
    private SexoDTO _sexo;
    private Calendar _fechaNacimiento;
    private int _edad;
    private String _tipo;
    private String _usuario;
}

```

```

private String _contrasenia;
private int _version;

@Id @Column(name="id_usuario")
@GeneratedValue(strategy=GenerationType.IDENTITY)
public long getId()
{
    return _id;
}
@OneToOne @JoinColumn (name="idescolaridad")
public EscolaridadDTO getEscolaridad()
{
    return _escolaridad;
}
@OneToOne @JoinColumn (name="idunidad")
public UnidadDTO getUnidad()
{
    return _unidad;
}
@OneToOne @JoinColumn (name="idestadocivil")
public EstadoCivilDTO getEstadoCivil()
{
    return _estadoCivil;
}
@OneToOne @JoinColumn (name="idturno")
public TurnoDTO getTurno()
{
    return _turno;
}
@OneToOne @JoinColumn (name="idnacionalidad")
public NacionalidadDTO getNacionalidad()
{
    return _nacionalidad;
}
@OneToOne @JoinColumn (name="idrol")
public RolDTO getRol()
{
    return _rol;
}
@Column (name="nombre")
public String getNombre()
{
    return _nombre;
}
@Column (name="apellidopaterno")
public String getApellidoPaterno()
{
    return _apellidoPaterno;
}
@Column (name="apellidomaterno")
public String getApellidoMaterno()
{
    return _apellidoMaterno;
}
@Column (name="cedula")
public String getCedula()
{
    return _cedula;
}
@Column (name="direccion")
public String getDireccion()

```

```

{
    return _direccion;
}
@Column (name="telefono")
public String getTelefono()
{
    return _telefono;
}
@OneToOne @JoinColumn (name="idsexo")
public SexoDTO getSexo()
{
    return _sexo;
}
@Column (name="fechanacimiento")
public Calendar getFechaNacimiento()
{
    return _fechaNacimiento;
}
@Transient
public int getEdad()
{
    return _edad;
}
@Column (name="tipo")
public String getTipo()
{
    return _tipo;
}
@Column (name="usuario")
public String getUsuario()
{
    return _usuario;
}
@Column (name="contrasenia")
public String getContrasenia()
{
    return _contrasenia;
}
@Column (name="version")
public int getVersion()
{
    return _version;
}
public void setVersion( int version )
{
    _version = version;
}
public void setId( long id )
{
    _id = id;
}
public void setEscolaridad( EscolaridadDTO escolaridad )
{
    _escolaridad = escolaridad;
}
public void setUnidad( UnidadDTO unidad )
{
    _unidad = unidad;
}
public void setEstadoCivil( EstadoCivilDTO estadoCivil )
{

```

```

    _estadoCivil = estadoCivil;
}
public void setTurno( TurnoDTO turno )
{
    _turno = turno;
}
public void setNacionalidad( NacionalidadDTO nacionalidad )
{
    _nacionalidad = nacionalidad;
}
public void setRol( RolDTO rol )
{
    _rol = rol;
}
public void setNombre( String nombre )
{
    _nombre = nombre;
}
public void setApellidoPaterno( String apellidoPaterno )
{
    _apellidoPaterno = apellidoPaterno;
}
public void setApellidoMaterno( String apellidoMaterno )
{
    _apellidoMaterno = apellidoMaterno;
}
public void setCedula( String cedula )
{
    _cedula = cedula;
}
public void setDireccion( String direccion )
{
    _direccion = direccion;
}
public void setTelefono( String telefono )
{
    _telefono = telefono;
}
public void setSexo( SexoDTO sexo )
{
    _sexo = sexo;
}
public void setFechaNacimiento( Calendar fechaNacimiento )
{
    _fechaNacimiento = fechaNacimiento;
}
public void setEdad( int edad )
{
    _edad = edad;
}
public void setTipo( String tipo )
{
    _tipo = tipo;
}
public void setUsuario( String usuario )
{
    _usuario = usuario;
}
public void setContrasenia( String contrasenia )
{
    _contrasenia = contrasenia;
}

```

```

}

public int calcularEdad(Calendar fechaNacimiento)
{
    int años = 0;

    Calendar hoy = Calendar.getInstance();

    años = hoy.get( Calendar.YEAR ) -
    fechaNacimiento.get( Calendar.YEAR );

    if(hoy.get( Calendar.MONTH ) < fechaNacimiento.get( Calendar.-
    MONTH ))
        años = años - 1;
    else if(hoy.get( Calendar.MONTH ) == fechaNacimiento.get( Calendar.-
    MONTH ))
    {
        if (hoy.get( Calendar.DAY_OF_MONTH ) < fechaNacimiento.get( Cal-
        endar.DAY_OF_MONTH ))
            años = años - 1;
        }

    return años;
}
}

```

ValidarUsuarioDTO.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.data;

/**
 * @author Daniel
 *
 */
public class ValidarUsuarioDTO
{
    private String _usuario;
    private String _contrasenia;

    public String getUsuario()
    {
        return _usuario;
    }
    public String getContrasenia()
    {
        return _contrasenia;
    }
    public void setUsuario( String usuario )
    {
        _usuario = usuario;
    }
    public void setContrasenia( String contrasenia )
    {
        _contrasenia = contrasenia;
    }
}

```

ProyectoTerminal/ece/business

AdministradorComentario.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.ComentarioDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorComentario
{
    public long insertarComentario(ComentarioDTO comentario);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public ComentarioDTO leerComentario(long id);
}
```

AdministradorConsultorio.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.ConsultorioDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorConsultorio
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<ConsultorioDTO> leerConsultorio();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarConsultorio();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<Map<String, Object>> cargarCatalogoConsulotrio();
}
```



```
}
```

AdministradorDetalleRecomendaciones.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import mx.uam.azc.proyectoTerminal.ece.data.DetalleRecomendacionesDTO;

/**
 * @author Daniel
 */
@Transactional (rollbackFor = Exception.class)
public interface AdministradorDetalleRecomendaciones
{
    public long insertarDetalle(DetalleRecomendacionesDTO
        detalleRecomendaciones);

    @Transactional (propagation=Propagation.SUPPORTS, readOnly=true)
    public DetalleRecomendacionesDTO leerDetalle (long id);
}
```

AdministradorEscolaridad.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import mx.uam.azc.proyectoTerminal.ece.data.EscolaridadDTO;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorEscolaridad
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<EscolaridadDTO> leerEscolaridad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarEscolaridad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<Map<String, Object>>cargarCatalogoEscolaridad();
}
```

AdministradorEspecialidad.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.EspecialidadDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorEspecialidad
{

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<EspecialidadDTO> leerEspecialidad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarEspecialidad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<Map<String, Object>> cargarCatalogoEspecialidad();

}
```

AdministradorEstadoCivil.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.EstadoCivilDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorEstadoCivil
{

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<EstadoCivilDTO> leerEstadoCivil();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarEstadoCivil();

}
```

```
}
```

AdministradorFamiliarResponsable.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.FamiliarResponsableDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorFamiliarResponsable
{
    public int insertarFamiliarResponsable(FamiliarResponsableDTO
        familiarResponsable);

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<FamiliarResponsableDTO> leerFamiliarResponsable();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarFamiliarResponsable();
}
}
```

AdministradorNacionalidad.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.NacionalidadDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorNacionalidad
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<NacionalidadDTO> leerNacionalidad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarNacionalidad();
}
```

```
}
```

AdministradorNotaAnestesiologia.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.NotaAnestesiologiaDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaAnestesiologiaJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorNotaAnestesiologia
{

    public long insertarNotaAnestesiologia (NotaAnestesiologiaDTO
        notaAnestesiologia);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public NotaAnestesiologiaJoinDTO leerNotaAnestesiologia(long id);
}

```

AdministradorNotaIngreso.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import mx.uam.azc.proyectoTerminal.ece.data.NotaIngresoDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaIngresoJoinDTO;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorNotaIngreso
{

    public long insertarNotaIngreso (NotaIngresoDTO notaIngreso);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public NotaIngresoJoinDTO leerNotaIngreso(long id);
}

```

AdministradorNotaInterconsulta.java

```
/**
 * @author Daniel
 */

```

```

package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.NotaInterconsultaDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaInterconsultaJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorNotaInterconsulta
{
    public long insertarNotaInterconsulta (NotaInterconsultaDTO
        notaInterconsulta);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public NotaInterconsultaJoinDTO leerNotaInterconsulta(long id);
}

```

AdministradorOcupacion.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.OcupacionDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
public interface AdministradorOcupacion
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<OcupacionDTO> leerOcupacion();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarOcupacion();
}

```

AdministradorPacienteHospitalizado.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import com.acsinet_solutions.cetia4.data.PagedResults;

import mx.uam.azc.proyectoTerminal.ece.business.local.FiltroPacientes;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteHospitalizadoDTO;

```

```

import
    mx.uam.azc.proyectoTerminal.ece.data.PacienteHospitalizadoJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

/**
 * @author Daniel
 *
 */
@Transactional (rollbackFor=Exception.class)
public interface AdministradorPacienteHospitalizado
{
    public long hospitalizarPaciente( PacienteHospitalizadoDTO
        pacienteHospitalizado );

    public void actualizarPacienteHospitalizado();

    public void borrarPacienteHospitalizado (PacienteHospitalizadoDTO
        pacienteHospitalizado);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public PacienteHospitalizadoJoinDTO leerPacienteHospitalizado(long
        id);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizadosQuery();

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizados(FiltroPacientes filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public PagedResults<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizadosPaginados( FiltroPacientes filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteHospitalizadoJoinDTO>
        buscarPacienteHospitalizadoCriterioNombre(FiltroPacientes filter);
}

```

AdministradorPacientes.java

```

package mx.uam.azc.proyectoTerminal.ece.business;

import com.acsinet_solutions.cetia4.data.PagedResults;

import java.util.List;

import mx.uam.azc.proyectoTerminal.ece.business.local.FiltroPacientes;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteDTO;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

@Transactional (rollbackFor=Exception.class)

```

```

public interface AdministradorPacientes {

    public long insertarPaciente( PacienteDTO paciente );

    public void actualizarPaciente(PacienteDTO paciente);

    public void borrarPaciente (long id);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public PacienteJoinDTO leerPaciente(long id);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteJoinDTO> leerPacientesQuery();

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteJoinDTO> leerPacientes(FiltroPacientes filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public PagedResults<PacienteJoinDTO>
    leerPacientesPaginados( FiltroPacientes filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<PacienteJoinDTO>
    buscarPacienteCriterioNombre(FiltroPacientes filter);
}

```

AdministradorRecomendaciones.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.RecomendacionesDTO;
import mx.uam.azc.proyectoTerminal.ece.data.RecomendacionesJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

/**
 * @author Daniel
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorRecomendaciones
{
    public long insertarRecomendaciones(RecomendacionesDTO
    recomendaciones);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public RecomendacionesJoinDTO leerRecomendaciones(long id);

    public void actualizarRecomendacion(RecomendacionesDTO
    recomendaciones);
}

```

AdministradorReligion.java

```

/**
 * @author Daniel

```

```

 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.ReligionDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
public interface AdministradorReligion
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<ReligionDTO> leerReligion();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarReligion();
}

```

AdministradorRoles.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.ConsultorioDTO;
import mx.uam.azc.proyectoTerminal.ece.data.RolDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
public interface AdministradorRoles
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<RolDTO> leerRoles();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> cargarRoles();
}

```

AdministradorSexo.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

```



```

import mx.uam.azc.proyectoTerminal.ece.data.SexoDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorSexo
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<SexoDTO> leerSexo();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarSexo();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<Map<String, Object>> cargarCatalogoSexo();
}

```

AdministradorSignosVitales.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.SignosVitalesDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorSignosVitales
{
    public long insertarSignosVitales(SignosVitalesDTO signosVitales);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public SignosVitalesDTO leerSignosVitales(long id);

    public void borrarSignosVitales(long id);

    public void actualizarSignosVitales(long id);
}

```

AdministradorTurno.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

```

```

import mx.uam.azc.proyectoTerminal.ece.data.TurnoDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
public interface AdministradorTurno
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<TurnoDTO> leerTurno();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarTurno();
}

```

AdministradorUnidad.java

```

/**
 * @author Daniel
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import mx.uam.azc.proyectoTerminal.ece.data.UnidadDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;
import java.util.Map;

/**
 * @author Daniel
 *
 */
public interface AdministradorUnidad
{
    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<UnidadDTO> leerUnidad();

    @Transactional ( propagation=Propagation.SUPPORTS, readOnly = true )
    public Map<Long, String> CargarUnidad();
}

```

AdministradorUsuario.java

```

/**
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business;

import com.acsinet_solutions.cetia4.data.PagedResults;

```

```

import mx.uam.azc.proyectoTerminal.ece.business.local.FiltroUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioDTO;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioJoinDTO;

import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

/**
 * @author Daniel
 *
 */
@Transactional(rollbackFor=Exception.class)
public interface AdministradorUsuarios
{
    public long insertarUsuario (UsuarioDTO usuario);

    public void actualizarUsuario(UsuarioDTO usuario);

    public void borrarUsuario (long id);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public UsuarioJoinDTO leerUsuario(long id);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<UsuarioJoinDTO> leerUsuarioQuery();

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<UsuarioJoinDTO> leerUsuarios(FiltroUsuarios filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public PagedResults<UsuarioJoinDTO>
        leerUsuariosPaginados( FiltroUsuarios filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<UsuarioJoinDTO>
        buscarUsuarioCriterioNombre(FiltroUsuarios filter);

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public List<UsuarioDTO> validarUsuario(String user, String pass);
}

```

ProyectoTerminal/ece/business/local

AdministradorComentarioImpl.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorComentario;
import mx.uam.azc.proyectoTerminal.ece.data.ComentarioDTO;

/**
 * @author Daniel
 */
public class AdministradorComentarioImpl implements
    AdministradorComentario
{
    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public long insertarComentario( ComentarioDTO comentario )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( comentario );

        return 0;
    }

    /**
     * @author Daniel
     */
    @Override
    public ComentarioDTO leerComentario( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        return ( ComentarioDTO )session.get( ComentarioDTO.class, id );
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}
```

AdministradorConsultorioImpl.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorConsultorio;
import mx.uam.azc.proyectoTerminal.ece.data.ConsultorioDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorConsultorioImpl implements
    AdministradorConsultorio
{
    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarConsultorio()
    {
        // TODO Auto-generated method stub
        List <ConsultorioDTO> c = leerConsultorio();
        Iterator<ConsultorioDTO> i = c.iterator();
        ConsultorioDTO cc = new ConsultorioDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getMedico() );
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<ConsultorioDTO> leerConsultorio()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_consultorio" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }
}
```

```

    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    /*
     * @author Daniel
     */
    @Override
    public List<Map<String, Object>> cargarCatalogoConsulotrio()
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        Query query = session.createQuery( "SELECT new map (c.id " +
            "AS idconsultorio, c.numero AS numero ) " +
            "FROM ConsultorioDTO c" );
        return query.list();
    }
}

```

AdministradorDetalleRecomendacionesImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorDetalleRecomendaciones;
import mx.uam.azc.proyectoTerminal.ece.data.DetalleRecomendacionesDTO;

/**
 * @author Daniel
 */
public class AdministradorDetalleRecomendacionesImpl implements
    AdministradorDetalleRecomendaciones
{
    private SessionFactory _sessionFactory;
    /*
     * @author Daniel
     */
    @Override
    public long insertarDetalle( DetalleRecomendacionesDTO
        detalleRecomendaciones )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( detalleRecomendaciones );
        return detalleRecomendaciones.getIdDetalle();
    }

    /*
     * @author Daniel

```

```

    */
    @Override
    public DetalleRecomendacionesDTO leerDetalle( long id )
    {
        // TODO Auto-generated method stub
        return null;
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorEscolaridadImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.apache.derby.impl.sql.compile.QueryTreeNode;
import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorEscolaridad;
import mx.uam.azc.proyectoTerminal.ece.data.EscolaridadDTO;

/**
 * @author Daniel
 */
public class AdministradorEscolaridadImpl implements
    AdministradorEscolaridad
{
    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarEscolaridad()
    {
        // TODO Auto-generated method stub
        List <EscolaridadDTO> e = leerEscolaridad();
        Iterator<EscolaridadDTO> i = e.iterator();
        EscolaridadDTO ee = new EscolaridadDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            ee = i.next();
            mapa.put( new Long (ee.getId()), ee.getNombre() );
        }
    }
}

```

```

        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<EscolaridadDTO> leerEscolaridad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_escolaridad" );

        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    /**
     * @author Daniel
     */
    @Override
    public List<Map<String, Object>> cargarCatalogoEscolaridad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.createQuery( "SELECT new map (t.id " +
            "AS idescolaridad, t.nombre AS escolaridad ) " +
            "FROM EscolaridadDTO t" );

        return query.list();
    }
}

```

AdministradorEspecialidadImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorEspecialidad;
import mx.uam.azc.proyectoTerminal.ece.data.EspecialidadDTO;

/**
 * @author Daniel
 *
 */

```



```

public class AdministradorEspecialidadImpl implements
    AdministradorEspecialidad
{

    private SessionFactory _sessionFactory;
    /*
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarEspecialidad()
    {
        // TODO Auto-generated method stub
        List <EspecialidadDTO> e = leerEspecialidad();
        Iterator<EspecialidadDTO> i = e.iterator();
        EspecialidadDTO es = new EspecialidadDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            es = i.next();
            mapa.put( new Long (es.getId()), es.getEspecialidad() );
        }
        return mapa;
    }

    /*
     * @author Daniel
     */
    @Override
    public List<Map<String, Object>> cargarCatalogoEspecialidad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        Query query = session.createQuery( "SELECT new map (e.id " +
            "AS idespecialidad, e.especialidad AS especialidad ) " +
            "FROM EspecialidadDTO e" );
        return query.list();
    }

    /*
     * @author Daniel
     */
    @Override
    public List<EspecialidadDTO> leerEspecialidad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_especialidad" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

```
}  
}
```

AdministradorEstadoCivil.java

```
/**  
 * @author Daniel  
 */  
package mx.uam.azc.proyectoTerminal.ece.business.local;  
  
import org.hibernate.*;  
  
import java.util.*;  
  
import  
    mx.uam.azc.proyectoTerminal.ece.business.AdministradorEstadoCivil;  
import mx.uam.azc.proyectoTerminal.ece.data.EstadoCivilDTO;  
  
/**  
 * @author Daniel  
 *  
 */  
public class AdministradorEstadoCivilImpl implements  
    AdministradorEstadoCivil  
{  
    private SessionFactory _sessionFactory;  
    /**  
     * @author Daniel  
     */  
    @Override  
    public Map<Long, String> CargarEstadoCivil()  
    {  
        // TODO Auto-generated method stub  
        List <EstadoCivilDTO> e = leerEstadoCivil();  
        Iterator<EstadoCivilDTO> i = e.iterator();  
        EstadoCivilDTO cc = new EstadoCivilDTO();  
  
        Map <Long, String> mapa = new TreeMap<Long, String>();  
  
        while(i.hasNext())  
        {  
            cc = i.next();  
            mapa.put( new Long (cc.getId()), cc.getEstadoCivil());  
        }  
        return mapa;  
    }  
  
    /**  
     * @author Daniel  
     */  
    @Override  
    public List<EstadoCivilDTO> leerEstadoCivil()  
    {  
        // TODO Auto-generated method stub  
        Session session = getSession();  
        Query query = session.getNamedQuery( "leer_estado_civil" );  
        return query.list();  
    }  
}
```

```

public void setSessionFactory (SessionFactory sessionFactory)
{
    _sessionFactory = sessionFactory;
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}
}

```

AdministradorFamiliarResponsableImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorFamiliarRespo
nsable;
import mx.uam.azc.proyectoTerminal.ece.data.FamiliarResponsableDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorFamiliarResponsableImpl implements
    AdministradorFamiliarResponsable
{

    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarFamiliarResponsable()
    {
        // TODO Auto-generated method stub
        List <FamiliarResponsableDTO> e = leerFamiliarResponsable();
        Iterator<FamiliarResponsableDTO> i = e.iterator();
        FamiliarResponsableDTO cc = new FamiliarResponsableDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            String nom = cc.getNombre() + " " + cc.getApellidoPaterno() + "
" + cc.getApellidoMaterno();
            mapa.put( new Long (cc.getId()), nom);
        }
        return mapa;
    }

    /**
     * @author Daniel

```

```

    */
    @Override
    public List<FamiliarResponsableDTO> leerFamiliarResponsable()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_familiar_respons-
able" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    /*
    * @author Daniel
    */
    @Override
    public int insertarFamiliarResponsable(
        FamiliarResponsableDTO familiarResponsable )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( familiarResponsable );
        return familiarResponsable.getId();
    }
}

```

AdministradorNacionalidadImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorNacionalidad;
import mx.uam.azc.proyectoTerminal.ece.data.NacionalidadDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorNacionalidadImpl implements
    AdministradorNacionalidad
{

    private SessionFactory _sessionFactory;
    /*

```

```

    * @author Daniel
    */
    @Override
    public Map<Long, String> CargarNacionalidad()
    {
        // TODO Auto-generated method stub
        List <NacionalidadDTO> e = leerNacionalidad();
        Iterator<NacionalidadDTO> i = e.iterator();
        NacionalidadDTO cc = new NacionalidadDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getNacionalidad());
        }
        return mapa;
    }

    /*
    * @author Daniel
    */
    @Override
    public List<NacionalidadDTO> leerNacionalidad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_nacionalidad" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorNotaAnestesiologiaImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorNotaAnestesiologia;
import mx.uam.azc.proyectoTerminal.ece.data.NotaAnestesiologiaDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaAnestesiologiaJoinDTO;
/**

```

```

* @author Daniel
*
*/
public class AdministradorNotaAnestesiologiaImpl implements
    AdministradorNotaAnestesiologia
{
    private SessionFactory _sessionFactory;
    /*
    * @author Daniel
    */
    @Override
    public long insertarNotaAnestesiologia( NotaAnestesiologiaDTO
        notaAnestesiologia )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( notaAnestesiologia );
        return 0;
    }

    /*
    * @author Daniel
    */
    @Override
    public NotaAnestesiologiaJoinDTO leerNotaAnestesiologia( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        return ( NotaAnestesiologiaJoinDTO )session.get( NotaAnestesiolo-
            giaJoinDTO.class, id );
    }
    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorNotaIngresoImpl.java

```

/**
* @author Daniel
*/
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import
    mx.uam.azc.proyectoTerminal.ece.business.AdministradorNotaIngreso;
import mx.uam.azc.proyectoTerminal.ece.data.NotaIngresoDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaIngresoJoinDTO;

/**
* @author Daniel
*
*/

```

```

public class AdministradorNotaIngresoImpl implements
    AdministradorNotaIngreso
{
    /*
     * @author Daniel
     */
    private SessionFactory _sessionFactory;

    @Override
    public long insertarNotaIngreso( NotaIngresoDTO notaIngreso )
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        session.save( notaIngreso );
        _sessionFactory.close();
        return notaIngreso.getIdNota();
    }

    /*
     * @author Daniel
     */
    @Override
    public NotaIngresoJoinDTO leerNotaIngreso( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        NotaIngresoJoinDTO nota = ( NotaIngresoJoinDTO )session.get( No-
            taIngresoJoinDTO.class, id );
        _sessionFactory.close();
        return nota ;
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorNotaInteconsultaImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorNotaIntercons-
    ulta;
import mx.uam.azc.proyectoTerminal.ece.data.NotaInterconsultaDTO;
import mx.uam.azc.proyectoTerminal.ece.data.NotaInterconsultaJoinDTO;

```

```

/**
 * @author Daniel
 *
 */
public class AdministradorNotaInterconsultaImpl implements
    AdministradorNotaInterconsulta
{
    /**
     * @author Daniel
     */
    private SessionFactory _sessionFactory;
    @Override
    public long insertarNotaInterconsulta( NotaInterconsultaDTO
        notaInterconsulta )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( notaInterconsulta );
        return notaInterconsulta.getIdNota();
    }

    /**
     * @author Daniel
     */
    @Override
    public NotaInterconsultaJoinDTO leerNotaInterconsulta( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        return ( NotaInterconsultaJoinDTO )session.get( NotaInterconsulta-
            JoinDTO.class, id);
    }
    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorOcupacionImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import
    mx.uam.azc.proyectoTerminal.ece.business.AdministradorOcupacion;
import
    mx.uam.azc.proyectoTerminal.ece.data.OcupacionDTO;

```



```

/**
 * @author Daniel
 */
public class AdministradorOcupacionImpl implements
    AdministradorOcupacion
{

    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarOcupacion()
    {
        // TODO Auto-generated method stub
        List <OcupacionDTO> e = leerOcupacion();
        Iterator<OcupacionDTO> i = e.iterator();
        OcupacionDTO cc = new OcupacionDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getOcupacion());
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<OcupacionDTO> leerOcupacion()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_ocupacion" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorPacienteHospitalizadoImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

```

```

import com.acsinet_solutions.cetia4.data.PagedResults;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import java.util.List;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorPacienteHospitalizado;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteHospitalizadoDTO;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteHospitalizadoJoinDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorPacienteHospitalizadoImpl implements
    AdministradorPacienteHospitalizado
{

    private SessionFactory _sessionFactory;
    /*
     * @author Daniel
     */
    @Override
    public void actualizarPacienteHospitalizado()
    {
        // TODO Auto-generated method stub
    }

    /*
     * @author Daniel
     */
    @Override
    public void borrarPacienteHospitalizado(
        PacienteHospitalizadoDTO pacienteHospitalizado )
    {
        // TODO Auto-generated method stub
    }

    /*
     * @author Daniel
     */
    @Override
    public List<PacienteHospitalizadoJoinDTO>
        buscarPacienteHospitalizadoCriterioNombre(
            FiltroPacientes filter )
    {
        // TODO Auto-generated method stub
        return null;
    }

    /*
     * @author Daniel

```

```

    */
    @Override
    public long hospitalizarPaciente(
        PacienteHospitalizadoDTO pacienteHospitalizado )
    {
        // TODO Auto-generated method stub
        Session session = _sessionFactory.getCurrentSession();
        session.save( pacienteHospitalizado );

        return pacienteHospitalizado.getId();
    }

    /*
     * @author Daniel
     */
    @Override
    public PacienteHospitalizadoJoinDTO leerPacienteHospitalizado( long
        id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        PacienteHospitalizadoJoinDTO p = ( PacienteHospitalizadoJoinDTO )
        session.get( PacienteHospitalizadoJoinDTO.class, id );

        p.getNotas().size();

        return ( PacienteHospitalizadoJoinDTO )session.get( PacienteHos-
        pitalizadoJoinDTO.class, id );
    }

    /*
     * @author Daniel
     */
    @Override
    public List<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizados(
            FiltroPacientes filter )
    {
        // TODO Auto-generated method stub
        return null;
    }

    /*
     * @author Daniel
     */
    @Override
    public PagedResults<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizadosPaginados(
            FiltroPacientes filter )
    {
        // TODO Auto-generated method stub
        return null;
    }

    /*
     * @author Daniel
     */
    @Override
    public List<PacienteHospitalizadoJoinDTO>
        leerPacientesHospitalizadosQuery()
    {

```

```

    // TODO Auto-generated method stub
    return null;
}

public void setSessionFactory (SessionFactory sessionFactory)
{
    _sessionFactory = sessionFactory;
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}
}

```

AdministradorPacienteimpl.java

```

package mx.uam.azc.proyectoTerminal.ece.business.local;

import com.acsinet_solutions.cetia4.data.DTOUtils;
import com.acsinet_solutions.cetia4.data.PagedResults;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorPacientes;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteDTO;
import mx.uam.azc.proyectoTerminal.ece.data.PacienteJoinDTO;

import org.hibernate.*;
import org.hibernate.criterion.Order;
import org.hibernate.criterion.Property;

import java.util.ArrayList;
import java.util.List;

public class AdministradorPacientesImpl implements
    AdministradorPacientes
{
    private SessionFactory _sessionFactory;

    @Override
    public long insertarPaciente(PacienteDTO paciente)
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save(paciente);
        return paciente.getIdPaciente();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

```

@Override
public void actualizarPaciente(PacienteDTO paciente) {
    // TODO Auto-generated method stub
    Session session = getSession();
    session.update(paciente);
}

@Override
public void borrarPaciente(long id) {
    // TODO Auto-generated method stub
    Session session = getSession();
    PacienteDTO paciente = (PacienteDTO)
session.load(PacienteDTO.class, id);
    session.delete(paciente);
}

@Override
public PacienteJoinDTO leerPaciente(long id) {
    // TODO Auto-generated method stub
    Session session = getSession();
    PacienteJoinDTO p = ( PacienteJoinDTO )session.get( Paciente-
JoinDTO.class, id );
    p.getPacienteHospitalizados().size();
    p.getRecomendaciones().size();
    return (PacienteJoinDTO) session.get(PacienteJoinDTO.class, id);
}

@Override
public List<PacienteJoinDTO> leerPacientesQuery() {
    // TODO Auto-generated method stub
    Session session = getSession();
    Query query = session.createQuery("SELECT p FROM PacienteDTO
p");
    return query.list();
}

/*
 * @author Daniel
 */
@Override
public List<PacienteJoinDTO> leerPacientes( FiltroPacientes filter )
{
    // TODO Auto-generated method stub
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_pacientes_filtro" );
    query.setString( "nombre", "%" + filter.getNombre() + "%" );
    return query.list();
}

```

```

/*
 * @author Daniel
 */
@Override
public PagedResults<PacienteJoinDTO> leerPacientesPaginados(
    FiltroPacientes filter )
{
    // TODO Auto-generated method stub
    List<PacienteJoinDTO> lista = new ArrayList<PacienteJoinDTO>();

    lista = leerPacientes( filter );
    filter.setOrderBy( "idPaciente" );
    filter.setMaxResults( 3 );
    return DTOUtils.applyPaging (lista, filter);
}

/*
 * @author Daniel
 */
@Override
public List<PacienteJoinDTO>
    buscarPacienteCriterioNombre( FiltroPacientes filter )
{
    // TODO Auto-generated method stub
    Criteria criteria = getSession().createCriteria( PacienteDTO.class
    );

    String nombre = "%" + filter.getNombre() + "%";
    criteria.add( Property.forName( "nombre" ).like( nombre ) );
    int maxResults = filter.getMaxResults();

    if (maxResults > 0)
        criteria.setMaxResults( maxResults );
    criteria.addOrder( Order.asc( "id" ) );
    return criteria.list();
}
}
}

```

AdministradorRecomendacionesImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorRecomendaciones;
import mx.uam.azc.proyectoTerminal.ece.data.RecomendacionesDTO;
import mx.uam.azc.proyectoTerminal.ece.data.RecomendacionesJoinDTO;
/**

```

```

* @author Daniel
*
*/
public class AdministradorRecomendacionesImpl implements
    AdministradorRecomendaciones
{

    private SessionFactory _sessionFactory;
    /*
    * @author Daniel
    */
    @Override
    public long insertarRecomendaciones( RecomendacionesDTO
        recomendaciones )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( recomendaciones );
        return recomendaciones.getId();
    }

    @Override
    public RecomendacionesJoinDTO leerRecomendaciones( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        RecomendacionesJoinDTO reco = ( RecomendacionesJoinDTO )
        session.get( RecomendacionesJoinDTO.class, id );
        reco.getDetalleRecomendaciones().size();
        return ( RecomendacionesJoinDTO )session.get(Recomendaciones-
        JoinDTO.class, id);
    }

    public void actualizarRecomendacion(RecomendacionesDTO
        recomendacion)
    {
        Session session = getSession();

        session.update( recomendacion );
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorReligionImpl.java

```

/**
* @author Daniel
*/
package mx.uam.azc.proyectoTerminal.ece.business.local;

```

```

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorReligion;
import mx.uam.azc.proyectoTerminal.ece.data.ReligionDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorReligionImpl implements
    AdministradorReligion
{

    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarReligion()
    {
        // TODO Auto-generated method stub
        List <ReligionDTO> c = leerReligion();
        Iterator<ReligionDTO> i = c.iterator();
        ReligionDTO cc = new ReligionDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getReligion());
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<ReligionDTO> leerReligion()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_religion" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```


AdministradorRolesImpl.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorRoles;
import mx.uam.azc.proyectoTerminal.ece.data.ConsultorioDTO;
import mx.uam.azc.proyectoTerminal.ece.data.RolDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorRolesImpl implements AdministradorRoles
{

    /**
     * @author Daniel
     */
    private SessionFactory _sessionFactory;

    @Override
    public Map<Long, String> cargarRoles()
    {
        // TODO Auto-generated method stub
        List <RolDTO> r = leerRoles();
        Iterator<RolDTO> i = r.iterator();
        RolDTO rol = new RolDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            rol = i.next();
            mapa.put( new Long (rol.getId()), rol.getRol() );
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<RolDTO> leerRoles()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_rol" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {

```

```

    _sessionFactory = sessionFactory;
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}
}

```

AdministradorSexoImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorSexo;
import mx.uam.azc.proyectoTerminal.ece.data.ConsultorioDTO;
import mx.uam.azc.proyectoTerminal.ece.data.SexoDTO;

/**
 * @author Daniel
 */
public class AdministradorSexoImpl implements AdministradorSexo
{
    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarSexo()
    {
        // TODO Auto-generated method stub
        List <SexoDTO> c = leerSexo();
        Iterator<SexoDTO> i = c.iterator();
        SexoDTO sex = new SexoDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while (i.hasNext())
        {
            sex = i.next();
            mapa.put( new Long (sex.getIdSexo()), sex.getSexo() );
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<Map<String, Object>> cargarCatalogoSexo()
    {
        // TODO Auto-generated method stub
        Session session = getSession();

```

```

        Query query = session.createQuery( "SELECT new map (s.idSexo " +
            "AS id_sexo, s.sexo AS sexo ) " +
            "FROM SexoDTO s" );
        return query.list();
    }

    /**
     * @author Daniel
     */
    @Override
    public List<SexoDTO> leerSexo()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_sexo" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorSignosVitales.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorSignosVitales
;
import mx.uam.azc.proyectoTerminal.ece.data.SignosVitalesDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorSignosVitalesImpl implements
    AdministradorSignosVitales
{
    private SessionFactory _sessionFactory;

    /**
     * @author Daniel
     */
    @Override
    public void actualizarSignosVitales( long id )
    {

```

```

// TODO Auto-generated method stub

}

/*
 * @author Daniel
 */
@Override
public void borrarSignosVitales( long id )
{
// TODO Auto-generated method stub

}

/*
 * @author Daniel
 */
@Override
public long insertarSignosVitales( SignosVitalesDTO signosVitales )
{
// TODO Auto-generated method stub
Session session = getSession();
session.save( signosVitales );

return signosVitales.getId();
}

/*
 * @author Daniel
 */
@Override
public SignosVitalesDTO leerSignosVitales( long id )
{
// TODO Auto-generated method stub
SignosVitalesDTO signos = new SignosVitalesDTO();
Session session = getSession();
signos = ( SignosVitalesDTO )session.get( SignosVitalesDTO.class,
id);
return signos;
}

public void setSessionFactory (SessionFactory sessionFactory)
{
_sessionFactory = sessionFactory;
}

private Session getSession()
{
return _sessionFactory.getCurrentSession();
}
}

```

AdministradorTurnoImpl.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

```

```

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorTurno;
import mx.uam.azc.proyectoTerminal.ece.data.TurnoDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorTurnoImpl implements AdministradorTurno
{

    private SessionFactory _sessionFactory;
    /**
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarTurno()
    {
        // TODO Auto-generated method stub
        List <TurnoDTO> c = leerTurno();
        Iterator<TurnoDTO> i = c.iterator();
        TurnoDTO cc = new TurnoDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getTurno() );
        }
        return mapa;
    }

    /**
     * @author Daniel
     */
    @Override
    public List<TurnoDTO> leerTurno()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_turno" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorUnidad.java

```
/**
```

```

    * @author Daniel
    */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUnidad;
import mx.uam.azc.proyectoTerminal.ece.data.UnidadDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorUnidadImpl implements AdministradorUnidad
{
    private SessionFactory _sessionFactory;
    /*
     * @author Daniel
     */
    @Override
    public Map<Long, String> CargarUnidad()
    {
        // TODO Auto-generated method stub
        List <UnidadDTO> c = leerUnidad();
        Iterator<UnidadDTO> i = c.iterator();
        UnidadDTO cc = new UnidadDTO();

        Map <Long, String> mapa = new TreeMap<Long, String>();

        while(i.hasNext())
        {
            cc = i.next();
            mapa.put( new Long (cc.getId()), cc.getNombre() );
        }
        return mapa;
    }

    /*
     * @author Daniel
     */
    @Override
    public List<UnidadDTO> leerUnidad()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_unidad" );
        return query.list();
    }

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }
}

```

AdministradorUsuarioImpl.java

```
/**
 *
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import com.acsinet_solutions.cetia4.data.DTOUtils;
import com.acsinet_solutions.cetia4.data.PagedResults;

import org.hibernate.*;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioDTO;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioJoinDTO;

/**
 * @author Daniel
 *
 */
public class AdministradorUsuariosImpl implements
    AdministradorUsuarios
{
    private SessionFactory _sessionFactory;

    public void setSessionFactory (SessionFactory sessionFactory)
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    /**
     * @author Daniel
     */
    @Override
    public void actualizarUsuario( UsuarioDTO usuario )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.update( usuario );
    }

    /**
     * @author Daniel
     */
    @Override
    public void borrarUsuario( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        UsuarioDTO usuario = ( UsuarioDTO )session.load( UsuarioDTO.class,
            id );
    }
}
```

```

        session.delete( usuario );
    }

    /**
     * @author Daniel
     */
    @Override
    public List<UsuarioJoinDTO>
        buscarUsuarioCriterioNombre( FiltroUsuarios filter )
    {
        // TODO Auto-generated method stub
        return null;
    }

    /**
     * @author Daniel
     */
    @Override
    public long insertarUsuario( UsuarioDTO usuario )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        session.save( usuario );
        return usuario.getId();
    }

    /**
     * @author Daniel
     */
    @Override
    public UsuarioJoinDTO leerUsuario( long id )
    {
        // TODO Auto-generated method stub
        Session session = getSession();

        return ( UsuarioJoinDTO )session.get( UsuarioJoinDTO.class, id );
    }

    /**
     * @author Daniel
     */
    @Override
    public List<UsuarioJoinDTO> leerUsuarioQuery()
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.createQuery("SELECT u FROM UsuarioDTO u");
        return query.list();
    }

    /**
     * @author Daniel
     */
    @Override
    public List<UsuarioJoinDTO> leerUsuarios( FiltroUsuarios filter )
    {
        // TODO Auto-generated method stub
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_usuarios_filtro" );
    }

```



```

        query.setString( "nombre", "%" + filter.getNombre() + "%" );
        return query.list();
    }

    /**
     * @author Daniel
     */
    @Override
    public PagedResults<UsuarioJoinDTO>
        leerUsuariosPaginados( FiltroUsuarios filter )
    {
        // TODO Auto-generated method stub
        List<UsuarioJoinDTO> lista = new ArrayList<UsuarioJoinDTO>();
        lista = leerUsuarios( filter );
        filter.setMaxResults( 3 );

        return DTOUtils.applyPaging (lista, filter);
    }

    /**
     * @author Daniel
     */
    @Override
    public List<UsuarioDTO> validarUsuario( String user, String pass )
    {
        // TODO Auto-generated method stub
        Session session = getSession();
        Query query = session.createQuery( "SELECT u FROM UsuarioDTO u
        WHERE usuario='" +
            user + "' AND contrasenia='" + pass + "'");

        return query.list();
    }
}

```

FiltroPacientes.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import com.acsinet_solutions.cetia4.data.PagedSearchFilter;

/**
 * @author Daniel
 */
public class FiltroPacientes extends PagedSearchFilter
{
    private String _nombre;

    public FiltroPacientes()
    {
        setMaxResults( 3 );
    }

    public String getNombre()
    {

```

```

        return _nombre;
    }

    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }
}

```

FiltroUsuarios.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.business.local;

import com.acsinet_solutions.cetia4.data.PagedSearchFilter;

/**
 * @author Daniel
 *
 */
public class FiltroUsuarios extends PagedSearchFilter
{
    private String _nombre;

    public void FiltroUsuarios()
    {
        setMaxResults( 3 );
    }
    public String getNombre()
    {
        return _nombre;
    }
    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }
}

```

ProyectoTerminal/ece/actions

ComentarioAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import java.util.Map;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorComentario;
import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.ComentarioDTO;

```

```

import mx.uam.azc.proyectoTerminal.ece.data.UsuarioJoinDTO;

/**
 * @author Daniel
 *
 */
public class ComentarioAction extends ActionSupport implements Ses-
    sionAware
{
    private Map _sessionData;

    private int _idNota;
    private ComentarioDTO _comentario;

    private UsuarioJoinDTO _usuarioJoin;
    private AdministradorUsuarios _administradorUsuarios;

    private AdministradorComentario _administradorComentario;

    public String forma_insertar()
    {
        return "forma_insertar";
    }

    public String insertar()
    {
        _comentario.setIdNota( _idNota );
        long idUsuario = (( Long)_sessionData.get( "idUsuario" )).long-
            Value();
        _comentario.setIdUsuario(idUsuario );
        _administradorComentario.insertarComentario( _comentario );
        return "ver_redirect";
    }

    public String ver()
    {
        return "ver";
    }

    public String listar()
    {
        return "listar";
    }

    /**
     * @author Daniel
     */
    public void setSession( Map sessionData )
    {
        _sessionData = sessionData;
    }

    public Map getSessionData()
    {
        return _sessionData;
    }
}

```

```

public int getIdNota ()
{
    return _idNota;
}

public ComentarioDTO getComentario ()
{
    return _comentario;
}

public void setIdNota ( int idNota )
{
    _idNota = idNota;
}

public void setComentario ( ComentarioDTO comentario )
{
    _comentario = comentario;
}

public void setAdministradorComentario (
    AdministradorComentario administradorComentario )
{
    _administradorComentario = administradorComentario;
}
}

```

DetalleRecomendacionesAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorDetalleRecomendaciones;
import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.DetalleRecomendacionesDTO;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioJoinDTO;

/**
 * @author Daniel
 */
public class DetalleRecomendacionesAction extends ActionSupport
{
    private int _idRecomendacion;
    private long _idPaciente;

    private UsuarioJoinDTO _usuarioJoin;
    private AdministradorUsuarios _administradorUsuarios;

    private DetalleRecomendacionesDTO _detalle;
}

```

```

private AdministradorDetalleRecomendaciones
    _administradorDetalleRecomendaciones;

public String forma_insertar()
{
    return "forma_insertar";
}

public String insertar()
{
    _detalle.setIdRecomendaciones( _idRecomendacion );
    System.out.println( _idPaciente);

    _administradorDetalleRecomendaciones.insertarDetalle( _detalle );

    return "recomendaciones_redirect";
}

public int getIdRecomendacion()
{
    return _idRecomendacion;
}

public void setIdRecomendacion( int idRecomendacion )
{
    _idRecomendacion = idRecomendacion;
}

public DetalleRecomendacionesDTO getDetalle()
{
    return _detalle;
}

public void setDetalle( DetalleRecomendacionesDTO detalle )
{
    _detalle = detalle;
}

public void setAdministradorDetalleRecomendaciones (
    AdministradorDetalleRecomendaciones
    administradorDetalleRecomendaciones )
{
    _administradorDetalleRecomendaciones =
    administradorDetalleRecomendaciones;
}

public long getIdPaciente()
{
    return _idPaciente;
}

public void setIdPaciente( long idPaciente )
{
    _idPaciente = idPaciente;
}
}

```

FamiliarAction.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorFamiliarResponsible;
import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.*;

import org.apache.struts2.interceptor.SessionAware;

import java.util.Map;

/**
 * @author Daniel
 */
public class FamiliarAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;
    private PacienteDTO _paciente;
    private AdministradorFamiliarResponsible
        _administradorFamiliarResponsible;
    private FamiliarResponsibleDTO _familiarResponsible;

    private UsuarioJoinDTO _usuarioJoin;
    private AdministradorUsuarios _administradorUsuarios;

    public String forma_insertar()
    {
        _paciente = ( PacienteDTO )_sessionData.get( "paciente" );
        System.out.println("Nombre: " + _paciente.getNombre());
        return "forma_insertar";
    }

    public String insertar()
    {
        int idFamiliar;
        _sessionData.put( "paciente", _paciente );

        idFamiliar =
            _administradorFamiliarResponsible.insertarFamiliarResponsible( _familiarResponsible );
        _sessionData.put( "idFamiliar", idFamiliar );
        return "forma_insertar_paciente_redirect";
    }

    public void setSession( Map sessionData )
    {
        // TODO Auto-generated method stub
        _sessionData = sessionData;
    }

    public Map getSessionData()

```

```

    {
        return _sessionData;
    }

    public PacienteDTO getPaciente()
    {
        return _paciente;
    }

    public void setPaciente( PacienteDTO paciente )
    {
        _paciente = paciente;
    }

    public void setAdministradorFamiliarResponsable(
        AdministradorFamiliarResponsable
        administradorFamiliarResponsable )
    {
        _administradorFamiliarResponsable =
        administradorFamiliarResponsable;
    }

    public FamiliarResponsableDTO getFamiliarResponsable()
    {
        return _familiarResponsable;
    }

    public void setFamiliarResponsable( FamiliarResponsableDTO
        familiarResponsable )
    {
        _familiarResponsable = familiarResponsable;
    }
}

```

LoginAction.java

```

/**
 *
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.*;

/**
 * @author Daniel
 *
 */
public class LoginAction extends ActionSupport implements SessionAware
{

    private AdministradorUsuarios _administradorUsuarios;
    private UsuarioDTO _usuario;
    private Map _sessionData;
}

```

```

private String _login;
private String _contrasenia;
private UsuarioJoinDTO _usuarioJoin;

public String inicio()
{
    return "inicio";
}

public String validar()
{
    List<UsuarioDTO> list = _administradorUsuarios.validarUsuario(
        _login, _contrasenia);
    if (list.size() == 0)
    {
        addActionMessage("Usuario o contraseña incorrecta.");
        return "input";
    }
    Iterator<UsuarioDTO> i = list.iterator();
    while(i.hasNext())
    {
        _usuario = i.next();
    }

    if(_usuario.getRol() == 1 || _usuario.getRol() == 2)
    {
        _sessionData.put("idUsuario", _usuario.getId());
        _usuarioJoin = _administradorUsuarios.leerUsuario( ((Long)_ses-
            sionData.get("idUsuario")).longValue());
        return "principal_usuario";
    }

    return "input";
}

public String principal_usuario()
{
    _usuarioJoin = _administradorUsuarios.leerUsuario( ((Long)_ses-
        sionData.get("idUsuario")).longValue());

    return "principal_usuario";
}

public String cerrar_sesion()
{
    _sessionData.clear();

    return "terminar";
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public UsuarioDTO getUsuario()
{
    return _usuario;
}

```



```

    }
    public void setUsuario( UsuarioDTO usuario )
    {
        _usuario = usuario;
    }
    /*
     * @author Daniel
     */
    @Override
    public void setSession( Map sessionData )
    {
        // TODO Auto-generated method stub
        _sessionData = sessionData;
    }
    public String getContrasenia()
    {
        return _contrasenia;
    }

    public String getLogin()
    {
        return _login;
    }
    public void setLogin( String login )
    {
        _login = login;
    }
    public void setContrasenia( String contrasenia )
    {
        _contrasenia = contrasenia;
    }

    public UsuarioJoinDTO getUsuarioJoin()
    {
        return _usuarioJoin;
    }

    public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
    {
        _usuarioJoin = usuarioJoin;
    }
}

```

NotaAnestesiologiaAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.data.*;

import java.util.Calendar;
import java.util.Map;

```

```

/**
 * @author Daniel
 *
 */
public class NotaAnestesiologiaAction extends ActionSupport implements
    SessionAware
{
    private Calendar _fechaActual = Calendar.getInstance();
    private Map _sessionData;

    private long _idUserio;
    private long _idVitales;
    private long _idNota;
    private int _idExpediente;

    private NotaAnestesiologiaDTO _notaAnestesiologia;
    private NotaAnestesiologiaJoinDTO _notaAnestesiologiaJoin;

    private PacienteHospitalizadoJoinDTO _pacienteHospitalizadoJoin;

    private UsuarioJoinDTO _usuarioJoin;
    private SignosVitalesDTO _signosVitales;

    private AdministradorNotaAnestesiologia
        _administradorNotaAnestesiologia;
    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorPacienteHospitalizado
        _administradorPacienteHospitalizado;
    private AdministradorSignosVitales _administradorSignosVitales;

    public String forma_insertar()
    {
        _pacienteHospitalizadoJoin =
            _administradorPacienteHospitalizado.leerPacienteHospitalizado( _idE
                xpediente );
        _idUserio = ((Long)_sessionData.get( "idUserio" )).longValue();
        _usuarioJoin = _administradorUsuarios.leerUsuario( _idUserio );

        _idVitales = ((Long)_sessionData.get( "idVitales" )).longValue();
        _signosVitales = _administradorSignosVitales.leerSignosVitales(
            _idVitales );
        _fechaActual = Calendar.getInstance();

        return "forma_insertar";
    }

    public String insertar()
    {
        System.out.println( _fechaActual.getTime() );
        _notaAnestesiologia.setFechaRegistro( _fechaActual );
        _notaAnestesiologia.setDiagnostico( " " );
        _notaAnestesiologia.setNombreNota( "Nota de Anestesiologia" );
        _notaAnestesiologia.setIdVitales( _idVitales );
        _notaAnestesiologia.setIdPacienteHospitalizado( _idExpediente );
        _notaAnestesiologia.setIdUsuario( _idUserio );

        _administradorNotaAnestesiologia.insertarNotaAnestesiologia( _notaAn
            estesiologia );

        return "ver_expediente_redirect";
    }
}

```

```

}

public String ver()
{
    _notaAnestesiologiaJoin =
    _administradorNotaAnestesiologia.leerNotaAnestesiologia( _idNota );

    return "ver";
}

public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}
public Map getSessionData()
{
    return _sessionData;
}

public Calendar getFechaActual()
{
    return _fechaActual;
}

public long getIdUsuario()
{
    return _idUsuario;
}

public long getIdVitales()
{
    return _idVitales;
}

public long getIdNota()
{
    return _idNota;
}

public int getIdExpediente()
{
    return _idExpediente;
}

public PacienteHospitalizadoJoinDTO getPacienteHospitalizadoJoin()
{
    return _pacienteHospitalizadoJoin;
}

public UsuarioJoinDTO getUsuarioJoin()
{
    return _usuarioJoin;
}

public SignosVitalesDTO getSignosVitales()
{
    return _signosVitales;
}

```

```

public void setFechaActual( Calendar fechaActual )
{
    _fechaActual = fechaActual;
}

public void setIdUsuario( long idUsuario )
{
    _idUsuario = idUsuario;
}

public void setIdVitales( long idVitales )
{
    _idVitales = idVitales;
}

public void setIdNota( long idNota )
{
    _idNota = idNota;
}

public void setIdExpediente( int idExpediente )
{
    _idExpediente = idExpediente;
}

public void setPacienteHospitalizadoJoin(
    PacienteHospitalizadoJoinDTO pacienteHospitalizadoJoin )
{
    _pacienteHospitalizadoJoin = pacienteHospitalizadoJoin;
}

public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
{
    _usuarioJoin = usuarioJoin;
}

public void setSignosVitales( SignosVitalesDTO signosVitales )
{
    _signosVitales = signosVitales;
}

public void setAdministradorNotaAnestesiologia(
    AdministradorNotaAnestesiologia
    administradorNotaAnestesiologia )
{
    _administradorNotaAnestesiologia =
    administradorNotaAnestesiologia;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public void setAdministradorPacienteHospitalizado(
    AdministradorPacienteHospitalizado
    administradorPacienteHospitalizado )
{
    _administradorPacienteHospitalizado =
    administradorPacienteHospitalizado;
}

```

```

}

public void setAdministradorSignosVitales(
    AdministradorSignosVitales administradorSignosVitales )
{
    _administradorSignosVitales = administradorSignosVitales;
}

public NotaAnestesiologiaDTO getNotaAnestesiologia()
{
    return _notaAnestesiologia;
}

public void setNotaAnestesiologia( NotaAnestesiologiaDTO
    notaAnestesiologia )
{
    _notaAnestesiologia = notaAnestesiologia;
}

public NotaAnestesiologiaJoinDTO getNotaAnestesiologiaJoin()
{
    return _notaAnestesiologiaJoin;
}

public void setNotaAnestesiologiaJoin(
    NotaAnestesiologiaJoinDTO notaAnestesiologiaJoin )
{
    _notaAnestesiologiaJoin = notaAnestesiologiaJoin;
}
}

```

NotalIngresoAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.data.*;

import org.apache.struts2.interceptor.SessionAware;

import java.util.Calendar;
import java.util.Map;

/**
 * @author Daniel
 */
public class NotaIngresoAction extends ActionSupport implements Ses-
    sionAware
{
    private SignosVitalesDTO _signosVitales;
    private Calendar _fechaActual = Calendar.getInstance();
    private Map _sessionData;
}

```

```

private long _idUserio;
private long _idVitales;
private long _idNota;
private int _idExpediente;

private PacienteHospitalizadoJoinDTO _pacienteHospitalizadoJoin;
private UsuarioJoinDTO _usuarioJoin;
private NotaIngresoDTO _notaIngreso;

private NotaIngresoJoinDTO _notaIngresoJoin;

private AdministradorUsuarios _administradorUsuarios;
private AdministradorPacienteHospitalizado
    _administradorPacienteHospitalizado;
private AdministradorSignosVitales _administradorSignosVitales;
private AdministradorNotaIngreso _administradorNotaIngreso;

public String forma_insertar()
{
    _pacienteHospitalizadoJoin =
        _administradorPacienteHospitalizado.leerPacienteHospitalizado( _idE
            xpediente );
    _idUserio = ((Long)_sessionData.get( "idUserio" )).longValue();
    _usuarioJoin = _administradorUsuarios.leerUsuario( _idUserio );

    _idVitales = ((Long)_sessionData.get( "idVitales" )).longValue();
    _signosVitales = _administradorSignosVitales.leerSignosVitales(
        _idVitales );
    _fechaActual = Calendar.getInstance();

    return"forma_insertar";
}

public String insertar()
{
    _notaIngreso.setFechaRegistro( _fechaActual );
    _notaIngreso.setNombreNota( "Nota de Ingreso" );
    _notaIngreso.setIdVitales( _idVitales );
    _notaIngreso.setIdPacienteHospitalizado( _idExpediente );
    _notaIngreso.setIdUsuario( _idUserio );

    _administradorNotaIngreso.insertarNotaIngreso( _notaIngreso );

    return"ver_expediente_redirect";
}

public String ver()
{
    _notaIngresoJoin = _administradorNotaIngreso.leerNotaIngreso(
        _idNota );

    return "ver";
}

public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}

```

```

}
public Map getSessionData ()
{
    return _sessionData;
}

public int getIdExpediente ()
{
    return _idExpediente;
}

public void setIdExpediente ( int idExpediente )
{
    _idExpediente = idExpediente;
}

public long getIdUsuario ()
{
    return _idUsuario;
}

public PacienteHospitalizadoJoinDTO getPacienteHospitalizadoJoin ()
{
    return _pacienteHospitalizadoJoin;
}

public UsuarioJoinDTO getUsuarioJoin ()
{
    return _usuarioJoin;
}

public void setIdUsuario ( long idUsuario )
{
    _idUsuario = idUsuario;
}

public void setPacienteHospitalizadoJoin (
    PacienteHospitalizadoJoinDTO pacienteHospitalizadoJoin )
{
    _pacienteHospitalizadoJoin = pacienteHospitalizadoJoin;
}

public void setUsuarioJoin ( UsuarioJoinDTO usuarioJoin )
{
    _usuarioJoin = usuarioJoin;
}

public void setAdministradorUsuarios (
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public void setAdministradorPacienteHospitalizado (
    AdministradorPacienteHospitalizado
    administradorPacienteHospitalizado )
{
    _administradorPacienteHospitalizado =
    administradorPacienteHospitalizado;
}

```

```

public NotaIngresoDTO getNotaIngreso()
{
    return _notaIngreso;
}

public void setNotaIngreso( NotaIngresoDTO notaIngreso )
{
    _notaIngreso = notaIngreso;
}

public Calendar getFechaActual()
{
    return _fechaActual;
}

public void setFechaActual( Calendar fechaActual )
{
    _fechaActual = fechaActual;
}

public SignosVitalesDTO getSignosVitales()
{
    return _signosVitales;
}

public void setSignosVitales( SignosVitalesDTO signosVitales )
{
    _signosVitales = signosVitales;
}

public void setAdministradorSignosVitales(
    AdministradorSignosVitales administradorSignosVitales )
{
    _administradorSignosVitales = administradorSignosVitales;
}

public long getIdVitales()
{
    return _idVitales;
}

public void setIdVitales( long idVitales )
{
    _idVitales = idVitales;
}

public void setAdministradorNotaIngreso(
    AdministradorNotaIngreso administradorNotaIngreso )
{
    _administradorNotaIngreso = administradorNotaIngreso;
}

public long getIdNota()
{
    return _idNota;
}

public void setIdNota( long idNota )
{
    _idNota = idNota;
}

```



```

    }

    public NotaIngresoJoinDTO getNotaIngresoJoin()
    {
        return _notaIngresoJoin;
    }

    public void setNotaIngresoJoin( NotaIngresoJoinDTO notaIngresoJoin )
    {
        _notaIngresoJoin = notaIngresoJoin;
    }
}

```

NotaInterconsultaAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.data.*;

import java.util.Calendar;
import java.util.Map;

/**
 * @author Daniel
 */
public class NotaInterconsultaAction extends ActionSupport implements
    SessionAware
{
    private Calendar _fechaActual = Calendar.getInstance();
    private Map _sessionData;
    private long _idUsuario;
    private long _idVitales;
    private long _idNota;
    private int _idExpediente;

    private PacienteHospitalizadoJoinDTO _pacienteHospitalizadoJoin;
    private UsuarioJoinDTO _usuarioJoin;
    private NotaInterconsultaDTO _notaInterconsulta;
    private NotaInterconsultaJoinDTO _notaInterconsultaJoin;
    private SignosVitalesDTO _signosVitales;

    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorPacienteHospitalizado
        _administradorPacienteHospitalizado;
    private AdministradorSignosVitales _administradorSignosVitales;
    private AdministradorNotaInterconsulta
        _administradorNotaInterconsulta;

    public String forma_insertar()
    {

```

```

    _pacienteHospitalizadoJoin =
    _administradorPacienteHospitalizado.leerPacienteHospitalizado( _idE
xpediente );
    _idUserario = ((Long)_sessionData.get( "idUserario" )).longValue();
    _usuarioJoin = _administradorUsuarios.leerUsuario( _idUserario );

    _idVitales = ((Long)_sessionData.get( "idVitales" )).longValue();
    _signosVitales = _administradorSignosVitales.leerSignosVitales(
_idVitales );
    _fechaActual = Calendar.getInstance();

    return"forma_insertar";
}

public String insertar()
{

    _notaInterconsulta.setFechaRegistro( _fechaActual );
    _notaInterconsulta.setNombreNota( "Nota de Interconsulta" );
    _notaInterconsulta.setIdVitales( _idVitales );
    _notaInterconsulta.setIdPacienteHospitalizado( _idExpediente );
    _notaInterconsulta.setIdUsuario( _idUserario );

    _administradorNotaInterconsulta.insertarNotaInterconsulta( _notaInt
erconsulta );

    return"ver_expediente_redirect";

}

public String ver()
{
    _notaInterconsultaJoin =
    _administradorNotaInterconsulta.leerNotaInterconsulta( _idNota );

    return "ver";
}

public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}
public Map getSessionData()
{
    return _sessionData;
}

public int getIdExpediente()
{
    return _idExpediente;
}

public void setIdExpediente( int idExpediente )
{
    _idExpediente = idExpediente;
}

public long getIdUsuario()

```

```

{
    return _idUserario;
}

public PacienteHospitalizadoJoinDTO getPacienteHospitalizadoJoin()
{
    return _pacienteHospitalizadoJoin;
}

public UsuarioJoinDTO getUsuarioJoin()
{
    return _usuarioJoin;
}

public void setIdUsuario( long idUsuario )
{
    _idUserario = idUsuario;
}

public void setPacienteHospitalizadoJoin(
    PacienteHospitalizadoJoinDTO pacienteHospitalizadoJoin )
{
    _pacienteHospitalizadoJoin = pacienteHospitalizadoJoin;
}

public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
{
    _usuarioJoin = usuarioJoin;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public void setAdministradorPacienteHospitalizado(
    AdministradorPacienteHospitalizado
    administradorPacienteHospitalizado )
{
    _administradorPacienteHospitalizado =
    administradorPacienteHospitalizado;
}

public Calendar getFechaActual()
{
    return _fechaActual;
}

public void setFechaActual( Calendar fechaActual )
{
    _fechaActual = fechaActual;
}

public SignosVitalesDTO getSignosVitales()
{
    return _signosVitales;
}

public void setSignosVitales( SignosVitalesDTO signosVitales )

```

```

{
    _signosVitales = signosVitales;
}

public void setAdministradorSignosVitales(
    AdministradorSignosVitales administradorSignosVitales )
{
    _administradorSignosVitales = administradorSignosVitales;
}

public long getIdVitales()
{
    return _idVitales;
}

public void setIdVitales( long idVitales )
{
    _idVitales = idVitales;
}

public NotaInterconsultaDTO getNotaInterconsulta()
{
    return _notaInterconsulta;
}

public void setNotaInterconsulta( NotaInterconsultaDTO
    notaInterconsulta )
{
    _notaInterconsulta = notaInterconsulta;
}

public void setAdministradorNotaInterconsulta(
    AdministradorNotaInterconsulta administradorNotaInterconsulta )
{
    _administradorNotaInterconsulta = administradorNotaInterconsulta;
}

public long getIdNota()
{
    return _idNota;
}

public void setIdNota( long idNota )
{
    _idNota = idNota;
}

public NotaInterconsultaJoinDTO getNotaInterconsultaJoin()
{
    return _notaInterconsultaJoin;
}

public void setNotaInterconsultaJoin(
    NotaInterconsultaJoinDTO notaInterconsultaJoin )
{
    _notaInterconsultaJoin = notaInterconsultaJoin;
}
}

```

NotasAction.java

```
/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorPacienteHospitalizado;
import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.*;

/**
 * @author Daniel
 */
public class NotasAction extends ActionSupport
{
    private long _idExpediente;
    private long _idNota;
    private String _nombre;
    private PacienteHospitalizadoJoinDTO _pacienteHospitalizadoJoin;
    private AdministradorPacienteHospitalizado
        _administradorPacienteHospitalizado;
    private List<NotaDTO> _notas = new ArrayList<NotaDTO>();

    private UsuarioJoinDTO _usuarioJoin;
    private AdministradorUsuarios _administradorUsuarios;

    public String listar()
    {
        _pacienteHospitalizadoJoin =
            _administradorPacienteHospitalizado.leerPacienteHospitalizado( _idExpediente );

        Collection<NotaDTO> notes = _pacienteHospitalizadoJoin.getNotas();
        Iterator<NotaDTO> iterator = notes.iterator();

        while(iterator.hasNext())
        {
            _notas.add( iterator.next() );
        }

        return "listar";
    }

    public String ver()
    {
        if(_nombre.compareTo( "Nota de Ingreso" ) == 0)
            return "ingreso";
        else if(_nombre.compareTo( "Nota de Interconsulta" ) == 0)
            return "interconsulta";
        else if ( _nombre.compareTo( "Nota de Anestesiología" ) == 0)

```

```

        return "anestesiologia";
    else
        return "listar";
    }
    public long getIdExpediente()
    {
        return _idExpediente;
    }

    public void setIdExpediente( long idExpediente )
    {
        _idExpediente = idExpediente;
    }

    public void setAdministradorPacienteHospitalizado(
        AdministradorPacienteHospitalizado
        administradorPacienteHospitalizado )
    {
        _administradorPacienteHospitalizado =
        administradorPacienteHospitalizado;
    }

    public PacienteHospitalizadoJoinDTO getPacienteHospitalizadoJoin()
    {
        return _pacienteHospitalizadoJoin;
    }

    public void setPacienteHospitalizadoJoin(
        PacienteHospitalizadoJoinDTO pacienteHospitalizadoJoin )
    {
        _pacienteHospitalizadoJoin = pacienteHospitalizadoJoin;
    }

    public List<NotaDTO> getNotas()
    {
        return _notas;
    }

    public void setNotas( List<NotaDTO> notas )
    {
        _notas = notas;
    }

    public long getIdNota()
    {
        return _idNota;
    }

    public void setIdNota( long idNota )
    {
        _idNota = idNota;
    }

    public String getNombre()
    {
        return _nombre;
    }

    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }

```

```
}  
}
```

PacienteAction.java

```
/**  
 *  
 */  
package mx.uam.azc.proyectoTerminal.ece.actions;  
  
import com.opensymphony.xwork2.ActionSupport;  
import com.opensymphony.xwork2.Preparable;  
import com.acsinet_solutions.cetia4.data.PagedResults;  
  
import org.apache.struts2.ServletActionContext;  
import org.apache.struts2.interceptor.SessionAware;  
  
import javax.servlet.ServletContext;  
import javax.servlet.http.HttpServletRequest;  
  
import mx.uam.azc.proyectoTerminal.ece.business.*;  
import mx.uam.azc.proyectoTerminal.ece.business.local.FiltroPacientes;  
import mx.uam.azc.proyectoTerminal.ece.data.*;  
  
import java.util.*;  
  
/**  
 * @author Daniel  
 *  
 */  
public class PacienteAction extends ActionSupport implements SessionAware, Preparable  
{  
    private Map _sessionData;  
  
    private long _idPaciente;  
  
    private PacienteDTO _paciente;  
    private PacienteJoinDTO _pacienteJoin;  
    private PagedResults<PacienteJoinDTO> _pacientes;  
  
    private UsuarioJoinDTO _usuarioJoin;  
    //private List<PacienteJoinDTO> _pacientes = new  
        ArrayList<PacienteJoinDTO>();  
  
    private FiltroPacientes _filtro;  
  
    private AdministradorPacientes _administradorPacientes;  
    private AdministradorConsultorio _administradorConsultorio;  
    private AdministradorEscolaridad _administradorEscolaridad;  
    private AdministradorEstadoCivil _administradorEstadoCivil;  
    private AdministradorNacionalidad _administradorNacionalidad;  
    private AdministradorOcupacion _administradorOcupacion;  
    private AdministradorReligion _administradorReligion;  
    private AdministradorTurno _administradorTurno;  
    private AdministradorUnidad _administradorUnidad;  
    private AdministradorRoles _administradorRoles;  
    private AdministradorSexo _administradorSexo;
```

```

private AdministradorUsuarios _administradorUsuarios;

private String buttonName;

public String listar()
{
    //_filtro.setNombre( "" );

    obtenerUsuario();

    _pacientes = _administradorPacientes.leerPacientesPaginados(
        _filtro );

    return "listar";
}
public String ver()
{
    obtenerUsuario();

    _pacienteJoin = _administradorPacientes.leerPaciente(
        _idPaciente );
    _pacienteJoin.setEdad( _pacienteJoin.calcularEdad( _pacienteJoin
        .getFechaNacimiento() ) );

    return "ver";
}
public String forma_insertar()
{
    obtenerUsuario();

    cargarCatalogosContexto();

    _paciente = ( PacienteDTO )_sessionData.get( "paciente" );

    return "forma_insertar";
}
//inserta a la base de datos
public String insertar()
{
    if("Registrar Familiar".equals( buttonName ))
    {
        _sessionData.put( "paciente", _paciente );
        return "forma_insertar_familiar";
    }
    else if("Registrar Paciente".equals( buttonName ))
    {
        if ( (_sessionData.get( "idFamiliar" )) == null)
        {
            addActionMessage( "Debe de Registrar el Familiar del paciente"
                );
            return "input";
        }
        int idFamiliar = ((Integer)_sessionData.get( "idFamiliar" ));

        _paciente.setFamiliarResponsable( idFamiliar );
        if(_paciente.getSexo() == 0)
        {
            addActionMessage( "Debe de seleccionar sexo" );
            return "input";
        }
    }
}

```



```

    }
    _administradorPacientes.insertarPaciente( _paciente );
}

    return "listar_redirect";
}
public String forma_actualizar()
{
    obtenerUsuario();

    cargarCatalogosContexto();
    _pacienteJoin = _administradorPacientes.leerPaciente(
        _pacienteJoin.getIdPaciente() );
    _paciente = cargarPaciente(_pacienteJoin);
    return "forma_actualizar";
}
public String actualizar()
{
    //actualiza a la base de datos

    return "ver_redirect";
}

public String borrar()
{
    //borra a la base de datos
    _administradorPacientes.borrarPaciente( _idPaciente );

    return "listar_redirect";
}

private PacienteDTO cargarPaciente(PacienteJoinDTO pacienteJoin)
{
    PacienteDTO paciente = new PacienteDTO();

    paciente.setApellidoMaterno( pacienteJoin.getApellidoMaterno() );
    paciente.setApellidoPaterno( pacienteJoin.getApellidoPaterno() );
    paciente.setConsultorio( pacienteJoin.getConsultorio().getId() );
    paciente.setCurp( pacienteJoin.getCurp() );
    paciente.setDireccion( pacienteJoin.getDireccion() );
    paciente.setEdad( pacienteJoin.getEdad() );
    paciente.setEstado( pacienteJoin.isEstado() );
    paciente.setEstadoCivil( pacienteJoin.getEstadoCivil().getId() );
    paciente.setEstatus( pacienteJoin.isEstatus() );
    paciente.setEtnicidad( pacienteJoin.getEtnicidad() );
    paciente.setFamiliarResponsable( pacienteJoin.getFamiliarResponsable().getId() );
    paciente.setFechaNacimiento( pacienteJoin.getFechaNacimiento() );
    paciente.setIdEscolaridad( pacienteJoin.getIdEscolaridad().getId() );
    ;
    paciente.setIdNacionalidad( pacienteJoin.getIdNacionalidad().getId() );
    ;
    paciente.setIdPaciente( pacienteJoin.getIdPaciente() );
    paciente.setIdReligion( pacienteJoin.getIdReligion().getId() );
    paciente.setLugarOrigen( pacienteJoin.getIdLugarOrigen() );
    paciente.setNombre( pacienteJoin.getIdNombre() );
    paciente.setOcupacion( pacienteJoin.getIdOcupacion().getId() );
    paciente.setSexo( pacienteJoin.getIdSexo().getIdSexo() );
    paciente.setTelefono( pacienteJoin.getIdTelefono() );
    paciente.setTurno( pacienteJoin.getIdTurno().getId() );
    paciente.setVersion( pacienteJoin.getIdVersion() );
}

```

```

    return paciente;
}

private void cargarCatalogosContexto()
{
    // TODO Auto-generated method stub
    HttpServletRequest request = ServletActionContext.getRequest();
    ServletContext context = request.getSession().getServletContext();

    cargarConsultorio(context);
    cargarEscolaridad (context);
    cargarEstadoCivil (context);
    cargarNacionalidad (context);
    cargarOcupacion (context);
    cargarReligion (context);
    cargarTurno (context);
    cargarUnidad(context);
    cargarRoles(context);
    cargarSexo(context);
}

private void cargarSexo(ServletContext context)
{
    // TODO Auto-generated method stub
    Map<Long, String> sexo = _administradorSexo.CargarSexo();
    context.setAttribute( "sexos", sexo );
}
/**
 * @param context
 */
private void cargarRoles( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> roles = _administradorRoles.cargarRoles();
    context.setAttribute( "roles", roles );
}

/**
 * @param context
 */
private void cargarUnidad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> unidad = _administradorUnidad.CargarUnidad();
    context.setAttribute( "unidad", unidad );
}

/**
 * @param context
 */
private void cargarTurno( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> turno = _administradorTurno.CargarTurno();
    context.setAttribute( "turno", turno );
}

/**
 * @param context
 */

```

```

private void cargarReligion( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> religion = _administradorReli-
        gion.CargarReligion();
    context.setAttribute( "religion", religion );
}

/**
 * @param context
 */
private void cargarOcupacion( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> ocupacion =
        _administradorOcupacion.CargarOcupacion();
    context.setAttribute( "ocupacion", ocupacion );
}

/**
 * @param context
 */
private void cargarNacionalidad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> nacionalidad =
        _administradorNacionalidad.CargarNacionalidad();
    context.setAttribute( "nacionalidad", nacionalidad );
}

/**
 * @param context
 */
private void cargarEstadoCivil( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> estadoCivil =
        _administradorEstadoCivil.CargarEstadoCivil();
    context.setAttribute( "estadoCivil", estadoCivil );
}

/**
 * @param context
 */
private void cargarConsultorio( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> consultorios =
        _administradorConsultorio.CargarConsultorio();
    context.setAttribute( "consultorios", consultorios );
}

/**
 * @param context
 */
private void cargarEscolaridad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> escolaridades =
        _administradorEscolaridad.CargarEscolaridad();
    context.setAttribute( "escolaridad", escolaridades );
}

```

```

}

/*****
*****/
public void setSession( Map sessionData )
{
// TODO Auto-generated method stub
_sessionData = sessionData;
}
public Map getSessionData()
{
return _sessionData;
}
public long getIdPaciente()
{
return _idPaciente;
}
public PacienteDTO getPaciente()
{
return _paciente;
}
public PacienteJoinDTO getPacienteJoin()
{
return _pacienteJoin;
}

public FiltroPacientes getFiltro()
{
return _filtro;
}
public void setIdPaciente( long idPaciente )
{
_idPaciente = idPaciente;
}
public void setPaciente( PacienteDTO paciente )
{
_paciente = paciente;
}
public void setPacienteJoin( PacienteJoinDTO pacienteJoin )
{
_pacienteJoin = pacienteJoin;
}
public void setFiltro( FiltroPacientes filtro )
{
_filtro = filtro;
}
public void setAdministradorPacientes(
AdministradorPacientes administradorPacientes )
{
_administradorPacientes = administradorPacientes;
}
public void setAdministradorConsultorio(
AdministradorConsultorio administradorConsultorio )
{
_administradorConsultorio = administradorConsultorio;
}
public void setAdministradorEscolaridad(
AdministradorEscolaridad administradorEscolaridad )
{
_administradorEscolaridad = administradorEscolaridad;
}

```

```

public void setAdministradorEstadoCivil(
    AdministradorEstadoCivil administradorEstadoCivil )
{
    _administradorEstadoCivil = administradorEstadoCivil;
}
public void setAdministradorNacionalidad(
    AdministradorNacionalidad administradorNacionalidad )
{
    _administradorNacionalidad = administradorNacionalidad;
}
public void setAdministradorOcupacion(
    AdministradorOcupacion administradorOcupacion )
{
    _administradorOcupacion = administradorOcupacion;
}
public void setAdministradorReligion(
    AdministradorReligion administradorReligion )
{
    _administradorReligion = administradorReligion;
}
public void setAdministradorTurno( AdministradorTurno
    administradorTurno )
{
    _administradorTurno = administradorTurno;
}
public void setAdministradorUnidad( AdministradorUnidad
    administradorUnidad )
{
    _administradorUnidad = administradorUnidad;
}
public void setAdministradorRoles( AdministradorRoles
    administradorRoles )
{
    _administradorRoles = administradorRoles;
}
public void setAdministradorSexo( AdministradorSexo
    administradorSexo )
{
    _administradorSexo = administradorSexo;
}
public void setButtonName( String buttonName )
{
    this.buttonName = buttonName;
}

public PagedResults<PacienteJoinDTO> getPacientes()
{
    return _pacientes;
}
public void setPacientes( PagedResults<PacienteJoinDTO> pacientes )
{
    _pacientes = pacientes;
}

public UsuarioJoinDTO getUsuarioJoin()
{
    return _usuarioJoin;
}
public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
{

```

```

    _usuarioJoin = usuarioJoin;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

/**
 * Obtenemos el nombre y el rol de nuestro usuario en sesion
 * utilizados en los layouts y que se muestran en cada pagina jsp.
 */
private void obtenerUsuario()
{
    _usuarioJoin = _administradorUsuarios.leerUsuario( ((Long)_ses-
        sionData.get( "idUserio" )).longValue() );
}

/*
 * @author Daniel
 */
@Override
public void prepare() throws Exception
{
    // TODO Auto-generated method stub
    _filtro = ( FiltroPacientes )_sessionData.get( "filtroPaciente" );

    if ( _filtro == null )
    {
        _filtro = new FiltroPacientes();
        _sessionData.put( "filtroPaciente", _filtro );
    }
}
}
}

```

PacienteHospitalizadoActiona.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.data.*;

import org.apache.struts2.ServletActionContext;
import org.apache.struts2.interceptor.SessionAware;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;

import java.util.*;

```

```

/**
 * @author Daniel
 *
 */
public class PacienteHospitalizadoAction extends ActionSupport implements SessionAware
{
    private long _idUsuario;
    private long _idPaciente;
    private long _idExpediente;

    private PacienteHospitalizadoDTO _pacienteHospitalizado;
    private PacienteHospitalizadoJoinDTO _pacienteHospitalizadoJoin;
    private PacienteJoinDTO _pacienteJoin;
    private UsuarioJoinDTO _usuarioJoin;
    private PacienteDTO _paciente;

    private AdministradorPacientes _administradorPacientes;
    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorPacienteHospitalizado
        _administradorPacienteHospitalizado;
    private AdministradorEspecialidad _administradorEspecialidad;

    private List<PacienteHospitalizadoDTO> _expedientes = new
        ArrayList<PacienteHospitalizadoDTO>();

    private Map _sessionData;

    public String forma_hospitalizar()
    {
        _idUsuario = ((Long)_sessionData.get( "idUsuario" )).longValue();
        _pacienteJoin = _administradorPacientes.leerPaciente(
            _idPaciente );
        _usuarioJoin = _administradorUsuarios.leerUsuario( _idUsuario );

        cargarCatalogosContexto();
        return "forma_hospitalizar";
    }

    public String hospitalizar()
    {
        _idPaciente = _pacienteJoin.getIdPaciente();
        _idUsuario = ((Long)_sessionData.get( "idUsuario" )).longValue();
        _pacienteHospitalizado.setIdPaciente( _pacienteJoin.getIdPaciente(
        ) );
        _pacienteJoin = _administradorPacientes.leerPaciente(
            _pacienteJoin.getIdPaciente() );
        _pacienteHospitalizado.setIdUsuario( _idUsuario );

        _paciente = cargarPaciente( _pacienteJoin );
        _paciente.setEstatus( true );

        _administradorPacienteHospitalizado.hospitalizarPaciente( _pacient
eHospitalizado );
        _administradorPacientes.actualizarPaciente( _paciente );
    }
}

```

```

    return "ver_paciente_redirect";
}

public String listar_expedientes()
{
    _pacienteJoin = _administradorPacientes.leerPaciente(
        _idPaciente );

    Collection<PacienteHospitalizadoDTO> exp =
        _pacienteJoin.getPacienteHospitalizados();
    Iterator<PacienteHospitalizadoDTO> iterator = exp.iterator();

    while(iterator.hasNext())
    {
        _expedientes.add( iterator.next() );
    }
    return "listar_expedientes";
}

public String ver_pacienteHospitalizado()
{
    _pacienteHospitalizadoJoin =
        _administradorPacienteHospitalizado.leerPacienteHospitalizado( _idE
            xpediente );
    _idPaciente
        =_pacienteHospitalizadoJoin.getIdPaciente().getIdPaciente();

    return "ver_pacienteHospitalizado";
}
/*****
*****/

/**
 * @param void
 * @return void
 * Carga los catalogos a utilizar en los jsp's
 * de la base de datos,
 */
private void cargarCatalogosContexto()
{
    HttpServletRequest request = ServletActionContext.getRequest();
    ServletContext context = request.getSession().getServletContext();

    cargarEspecialidad(context);
}

/**
 * @param context
 * @return void
 * Carga el catalogo de especialidades de la base de datos,
 * para que se encuentren en la vista de los jsp's.
 */
private void cargarEspecialidad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> especialidad =
        _administradorEspecialidad.CargarEspecialidad();
    context.setAttribute( "especialidad", especialidad );
}

```



```

public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}

private PacienteDTO cargarPaciente(PacienteJoinDTO pacienteJoin)
{
    PacienteDTO paciente = new PacienteDTO();

    paciente.setApellidoMaterno( pacienteJoin.getApellidoMaterno() );
    paciente.setApellidoPaterno( pacienteJoin.getApellidoPaterno() );
    paciente.setConsultorio( pacienteJoin.getConsultorio().getNumero()
);
    paciente.setCurp( pacienteJoin.getCurp() );
    paciente.setDireccion( pacienteJoin.getDireccion() );
    paciente.setEdad( pacienteJoin.getEdad() );
    paciente.setEstado( pacienteJoin.isEstado());
    paciente.setEstadoCivil( pacienteJoin.getEstadoCivil().getId() );
    paciente.setEstatus( pacienteJoin.isEstatus() );
    paciente.setEtnicidad( pacienteJoin.getEtnicidad() );
    paciente.setFamiliarResponsable( pacienteJoin.getFamiliarResponsable().getId() );
    paciente.setFechaNacimiento( pacienteJoin.getFechaNacimiento() );
    paciente.setIdEscolaridad( pacienteJoin.getEscolaridad().getId() )
;
    paciente.setIdNacionalidad( pacienteJoin.getNacionalidad().getId()
);
    paciente.setIdPaciente( pacienteJoin.getIdPaciente() );
    paciente.setIdReligion( pacienteJoin.getReligion().getId() );
    paciente.setLugarOrigen( pacienteJoin.getLugarOrigen() );
    paciente.setNombre( pacienteJoin.getNombre() );
    paciente.setOcupacion( pacienteJoin.getOcupacion().getId() );
    paciente.setSexo( pacienteJoin.getSexo().getIdSexo() );
    paciente.setTelefono( pacienteJoin.getTelefono() );
    paciente.setTurno( pacienteJoin.getTurno().getId() );
    paciente.setVersion( pacienteJoin.getVersion() );
    return paciente;
}

/*****
*****/
public List<PacienteHospitalizadoDTO> getExpedientes()
{
    return _expedientes;
}

public void setExpedientes( List<PacienteHospitalizadoDTO>
expedientes )
{
    _expedientes = expedientes;
}

public long getIdUsuario()
{
    return _idUsuario;
}

```

```

public long getIdPaciente()
{
    return _idPaciente;
}

public PacienteHospitalizadoDTO getPacienteHospitalizado()
{
    return _pacienteHospitalizado;
}

public PacienteJoinDTO getPacienteJoin()
{
    return _pacienteJoin;
}

public UsuarioJoinDTO getUsuarioJoin()
{
    return _usuarioJoin;
}

public PacienteDTO getPaciente()
{
    return _paciente;
}

public Map getSessionData()
{
    return _sessionData;
}

public void setIdUsuario( long idUsuario )
{
    _idUsuario = idUsuario;
}

public void setIdPaciente( long idPaciente )
{
    _idPaciente = idPaciente;
}

public void setPacienteHospitalizado(
    PacienteHospitalizadoDTO pacienteHospitalizado )
{
    _pacienteHospitalizado = pacienteHospitalizado;
}

public void setPacienteJoin( PacienteJoinDTO pacienteJoin )
{
    _pacienteJoin = pacienteJoin;
}

```

```

public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
{
    _usuarioJoin = usuarioJoin;
}

public void setPaciente( PacienteDTO paciente )
{
    _paciente = paciente;
}

public void setAdministradorPacientes(
    AdministradorPacientes administradorPacientes )
{
    _administradorPacientes = administradorPacientes;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public void setAdministradorPacienteHospitalizado(
    AdministradorPacienteHospitalizado
    administradorPacienteHospitalizado )
{
    _administradorPacienteHospitalizado =
    administradorPacienteHospitalizado;
}

public void setAdministradorEspecialidad(
    AdministradorEspecialidad administradorEspecialidad )
{
    _administradorEspecialidad = administradorEspecialidad;
}

public long getIdExpediente ()
{
    return _idExpediente;
}

public PacienteHospitalizadoJoinDTO getPacienteHospitalizadoJoin()
{
    return _pacienteHospitalizadoJoin;
}

public void setIdExpediente( long idExpediente )
{
    _idExpediente = idExpediente;
}

```

```

    public void setPacienteHospitalizadoJoin(
        PacienteHospitalizadoJoinDTO pacienteHospitalizadoJoin )
    {
        _pacienteHospitalizadoJoin = pacienteHospitalizadoJoin;
    }
}

```

RecomendacionesAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import java.util.*;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.data.*;

/**
 * @author Daniel
 */

public class RecomendacionesAction extends ActionSupport implements
    SessionAware
{
    private RecomendacionesDTO _recomendacion;
    private Map _sessionData;

    private PacienteJoinDTO _pacienteJoin;
    private UsuarioJoinDTO _usuarioJoin;
    private RecomendacionesJoinDTO _recomendacionesJoin;

    private long _idPaciente;
    private long _idUsuario;
    private long _idRecomendacion;

    private List<DetalleRecomendacionesDTO> _detalles = new
        ArrayList<DetalleRecomendacionesDTO>();
    private List<RecomendacionesDTO> _recomendaciones = new
        ArrayList<RecomendacionesDTO>();

    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorPacientes _administradorPacientes;
    private AdministradorRecomendaciones _administradorRecomendaciones;

    public String insertar()
    {
        _idUsuario = ((Long)_sessionData.get( "idUsuario" )).longValue();

        _recomendacion = new RecomendacionesDTO();
        _recomendacion.setEstatus( false );
    }
}

```

```

    _recomendacion.setFecha( Calendar.getInstance() );
    _recomendacion.setIdPaciente( _idPaciente );
    _recomendacion.setIdUsuario( _idUsuario );

    _idRecomendacion =
    _administradorRecomendaciones.insertarRecomendaciones( _recomendacion );

    _sessionData.put( "idRecomendacion", _idRecomendacion);

    return "insertar";
}

public String listar()
{
    _pacienteJoin = _administradorPacientes.leerPaciente(
    _idPaciente );

    Collection<RecomendacionesDTO> rec =
    _pacienteJoin.getRecomendaciones();
    Iterator<RecomendacionesDTO> iterator = rec.iterator();

    while(iterator.hasNext())
    {
        _recomendaciones.add( iterator.next() );
    }

    return "listar";
}

public String forma_insertar()
{
    _idUsuario = ((Long)_sessionData.get( "idUsuario" )).longValue();
    _idRecomendacion = ((Long)_sessionData.get( "idRecomendacion" )).longValue();
    _recomendacionesJoin =
    _administradorRecomendaciones.leerRecomendaciones( _idRecomendacion );

    Collection<DetalleRecomendacionesDTO> detail =
    _recomendacionesJoin.getDetalleRecomendaciones();
    Iterator<DetalleRecomendacionesDTO> iterator = detail.iterator();

    while(iterator.hasNext())
    {

        _detalles.add( iterator.next());

    }

    return "forma_insertar";
}

public String ver()
{
    _recomendacionesJoin =
    _administradorRecomendaciones.leerRecomendaciones( _idRecomendacion );

    Collection<DetalleRecomendacionesDTO> detail =
    _recomendacionesJoin.getDetalleRecomendaciones();

```

```

Iterator<DetalleRecomendacionesDTO> iterator = detail.iterator();

while(iterator.hasNext())
{

    _detalles.add( iterator.next());

}

return "ver";
}
/*public String forma_insertar_recomendaciones2()
{

    _idUserario = ((Long)_sessionData.get( "idUserario" )).longValue();
    _recomendacionesJoin =
    _administradorRecomendaciones.leerRecomendaciones( _idRecomendacion
);

    Collection<DetalleRecomendacionesDTO> detail =
    _recomendacionesJoin.getDetalleRecomendaciones();
    Iterator<DetalleRecomendacionesDTO> iterator = detail.iterator();

    while(iterator.hasNext())
    {

        _detalles.add( iterator.next());

    }

    return "forma_insertar_recomendaciones";
}*/

public String finalizar()
{
    System.out.println(_idPaciente);
    _idRecomendacion = ((Long)_sessionData.get( "idRecomen-
dacion" )).longValue();
    _recomendacionesJoin =
    _administradorRecomendaciones.leerRecomendaciones( _idRecomendacion
);

    _recomendacion = CargarRecomendacion(_recomendacionesJoin);

    _administradorRecomendaciones.actualizarRecomendacion( _recomendacion );
    _sessionData.remove( "idRecomendacion" );
    _sessionData.remove( "idPaciente" );
    return "ver_paciente_redirect";
}

/**
 * @param recomendacionesJoin
 * @return
 */
private RecomendacionesDTO CargarRecomendacion(
    RecomendacionesJoinDTO recomendacionesJoin )
{
    // TODO Auto-generated method stub
    RecomendacionesDTO recomendacion = new RecomendacionesDTO();

```

```

recomendacion.setEstatus( true );
recomendacion.setFecha( recomendacionesJoin.getFecha() );
recomendacion.setId( recomendacionesJoin.getId() );
recomendacion.setIdPaciente( recomendacionesJoin.getIdPaciente().get
    IdPaciente());
recomendacion.setIdUsuario( recomendacionesJoin.getIdUsuario().getId
    () );

return recomendacion;
}

public RecomendacionesDTO getRecomendacion()
{
    return _recomendacion;
}

public Map getSessionData()
{
    return _sessionData;
}

public void setRecomendacion( RecomendacionesDTO recomendacion )
{
    _recomendacion = recomendacion;
}

@Override
public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}

public long getIdPaciente()
{
    return _idPaciente;
}

public long getIdUsuario()
{
    return _idUsuario;
}

public void setIdPaciente( long idPaciente )
{
    _idPaciente = idPaciente;
}

public void setIdUsuario( long idUsuario )
{
    _idUsuario = idUsuario;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public void setAdministradorPacientes(

```

```

        AdministradorPacientes administradorPacientes )
    {
        _administradorPacientes = administradorPacientes;
    }

    public void setAdministradorRecomendaciones (
        AdministradorRecomendaciones administradorRecomendaciones )
    {
        _administradorRecomendaciones = administradorRecomendaciones;
    }

    public PacienteJoinDTO getPacienteJoin()
    {
        return _pacienteJoin;
    }

    public UsuarioJoinDTO getUsuarioJoin()
    {
        return _usuarioJoin;
    }

    public void setPacienteJoin( PacienteJoinDTO pacienteJoin )
    {
        _pacienteJoin = pacienteJoin;
    }

    public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
    {
        _usuarioJoin = usuarioJoin;
    }

    public long getIdRecomendacion()
    {
        return _idRecomendacion;
    }

    public void setIdRecomendacion( long idRecomendacion )
    {
        _idRecomendacion = idRecomendacion;
    }

    public RecomendacionesJoinDTO getRecomendacionesJoin()
    {
        return _recomendacionesJoin;
    }

    public void setRecomendacionesJoin( RecomendacionesJoinDTO
        recomendacionesJoin )
    {
        _recomendacionesJoin = recomendacionesJoin;
    }

    public List<DetalleRecomendacionesDTO> getDetalles()
    {
        return _detalles;
    }

    public void setDetalles( List<DetalleRecomendacionesDTO> detalles )
    {
        _detalles = detalles;
    }

```



```

    public List<RecomendacionesDTO> getRecomendaciones()
    {
        return _recomendaciones;
    }

    public void setRecomendaciones( List<RecomendacionesDTO>
recomendaciones )
    {
        _recomendaciones = recomendaciones;
    }
}

```

SignosVitalesAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

import com.opensymphony.xwork2.ActionSupport;

import org.apache.struts2.interceptor.SessionAware;

import java.util.Map;

import mx.uam.azc.proyectoTerminal.ece.business.AdministradorSignosVitales
;
import mx.uam.azc.proyectoTerminal.ece.business.AdministradorUsuarios;
import mx.uam.azc.proyectoTerminal.ece.data.SignosVitalesDTO;
import mx.uam.azc.proyectoTerminal.ece.data.UsuarioJoinDTO;

/**
 * @author Daniel
 *
 */
public class SignosVitalesAction extends ActionSupport implements Ses-
sionAware
{
    private SignosVitalesDTO _signosVitales;

    private long _idExpediente;
    private long _idVitales;
    private int _tipoNota;

    private float _imc;
    private Map _sessionData;

    private UsuarioJoinDTO _usuarioJoin;
    private AdministradorUsuarios _administradorUsuarios;

    AdministradorSignosVitales _administradorSignosVitales;

    public String forma_insertar()
    {

```

```

    return "forma_insertar";
}

public String insertar()
{
    String tipo = "input";
    _imc = _signosVitales.calcularIMC( _signosVitales.getPeso(),
    _signosVitales.getTalla() );
    _signosVitales.setImc( _imc );

    switch ( _tipoNota)
    {
        case 1:
            tipo = "forma_insertar_ingreso";
            break;
        case 2:
            tipo = "forma_insertar_interconsulta";
            break;
        case 3:
            tipo="forma_insertar_anestesiologia";
            break;
        case 4:
            tipo="forma_insertar_evolucion";
            break;
        case 5:
            tipo="forma_insertar_contrareferencia";
            break;
        default:
            tipo = "error";
            break;
    }
    if(tipo.compareTo( "error" ) == 0)
        return tipo;

    _idVitales = _administradorSignosVitales.insertarSignosVitales(
    _signosVitales );
    _sessionData.put( "idVitales", _idVitales );
    return tipo;
}

/*
 * @author Daniel
 */
@Override
public void setSession( Map sessionData )
{
    // TODO Auto-generated method stub
    _sessionData = sessionData;
}

public SignosVitalesDTO getSignosVitales()
{
    return _signosVitales;
}

public long getIdExpediente()
{
    return _idExpediente;
}

public int getTipoNota()

```

```

    {
        return _tipoNota;
    }

    public void setSignosVitales( SignosVitalesDTO signosVitales )
    {
        _signosVitales = signosVitales;
    }

    public void setIdExpediente( long idExpediente )
    {
        _idExpediente = idExpediente;
    }

    public void setTipoNota( int tipoNota )
    {
        _tipoNota = tipoNota;
    }

    public Map getSessionData()
    {
        return _sessionData;
    }

    public float getImc()
    {
        return _imc;
    }

    public void setImc( float imc )
    {
        _imc = imc;
    }

    public void setAdministradorSignosVitales(
        AdministradorSignosVitales administradorSignosVitales )
    {
        _administradorSignosVitales = administradorSignosVitales;
    }

    public long getIdVitales()
    {
        return _idVitales;
    }

    public void setIdVitales( long idVitales )
    {
        _idVitales = idVitales;
    }

}

```

UsuarioAction.java

```

/**
 * @author Daniel
 */
package mx.uam.azc.proyectoTerminal.ece.actions;

```

```

import com.acsinet_solutions.cetia4.data.PagedResults;
import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.Preparable;

import mx.uam.azc.proyectoTerminal.ece.business.*;
import mx.uam.azc.proyectoTerminal.ece.business.local.FiltroUsuarios;

import mx.uam.azc.proyectoTerminal.ece.data.*;

import org.apache.struts2.ServletActionContext;
import org.apache.struts2.interceptor.SessionAware;
import org.hibernate.HibernateException;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;

import java.util.*;

/**
 * @author Daniel
 *
 */
public class UsuarioAction extends ActionSupport implements Session-
    Aware, Preparable
{
    private Map _sessionData;
    private long _idUsuario;
    private String _contrasenia;
    private String _contraseniaActual;
    private String _contraseniaUsuario;

    private UsuarioDTO _usuario;
    private UsuarioJoinDTO _usuarioJoin;
    private PagedResults<UsuarioJoinDTO> _usuarios;
    //private List<UsuarioJoinDTO> _usuarios = new ArrayList<Usuario-
        JoinDTO>();

    private FiltroUsuarios _filtro;

    private AdministradorConsultorio _administradorConsultorio;
    private AdministradorEscolaridad _administradorEscolaridad;
    private AdministradorEstadoCivil _administradorEstadoCivil;
    private AdministradorNacionalidad _administradorNacionalidad;
    private AdministradorOcupacion _administradorOcupacion;
    private AdministradorReligion _administradorReligion;
    private AdministradorTurno _administradorTurno;
    private AdministradorUnidad _administradorUnidad;
    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorRoles _administradorRoles;
    private AdministradorSexo _administradorSexo;

    public String menu()
    {
        _usuarioJoin = _administradorUsuarios.leerUsuario( ((Long)_ses-
            sionData.get( "idUsuario" )).longValue() );
        return "menu";
    }

    public String listar()

```

```

{
    //_filtroUsuarios.setNombre( "" );
    _usuarios = _administradorUsuarios.leerUsuariosPaginados(
        _filtro );

    return "listar";
}
public String ver()
{
    _usuarioJoin = _administradorUsuarios.leerUsuario( _idUsuario );
    _usuarioJoin.setEdad( _usuarioJoin.calcularEdad( _usuarioJoin.getF
        echaNacimiento() ) );

    _sessionData.put( "idUsuario", _idUsuario );
    return "ver";
}

public String ver2()
{
    _idUsuario = ( ( Long )_sessionData.get( "idUsuar-
        io" ) ).longValue();
    _usuarioJoin = _administradorUsuarios.leerUsuario( _idUsuario );

    return "ver";
}

public String forma_insertar()
{
    cargarCatalogosContexto();
    return "forma_insertar";
}

public String insertar()
{
    if ( _usuario.getSexo() == 0 )
    {
        addActionMessage( "Debe seleccionar sexo." );
        return "input";
    }
    if ( ( _contrasenia.compareTo( _usuario.getContrasenia() ) ) == 0 )
    {
        try
        {
            _administradorUsuarios.insertarUsuario( _usuario );
            return "listar_redirect";
        }
        catch (HibernateException ex)
        {
            addActionMessage( "El usuario ya existe" );
            return "input";
        }
    }
    else
    {
        addActionMessage( "Las contraseñas no coinciden." );
        return "forma_insertar_redirect";
    }
}

public String forma_actualizar()

```

```

{
    cargarCatalogosContexto();

    _usuarioJoin = _administradorUsuarios.leerUsuario(
        _usuarioJoin.getId());
    _usuarioJoin.setEdad( _usuarioJoin.calcularEdad( _usuarioJoin.getFechaNacimiento() ) );
    _sessionData.put( "contrasenia", _usuarioJoin.getContrasenia() );
    _usuario = cargarUsuario(_usuarioJoin);

    return "forma_actualizar";
}

//actualiza a la base de datos
public String actualizar()
{

    _contraseniaUsuario = ( String )_sessionData.get( "contrasenia" );
    System.out.println( _usuario.getFechaNacimiento() );
    if(( _contraseniaActual.compareTo( _contraseniaUsuario ) != 0)
    {
        addActionMessage( "Contraseña incorrecta" );
        return "forma_actualizar";
    }
    else if ( ( _contrasenia.compareTo( _usuario.getContrasenia() ) ) == 0)
    {
        _administradorUsuarios.actualizarUsuario( _usuario );
        return "ver_redirect";
    }
    else
    {
        addActionMessage( "Las contraseñas no coinciden." );
        return "forma_actualizar";
    }
}

public String borrar()
{
    _administradorUsuarios.borrarUsuario( _idUser );

    return "listar_redirect";
}

/**
 *
 */
private void cargarCatalogosContexto()
{
    // TODO Auto-generated method stub
    HttpServletRequest request = ServletActionContext.getRequest();
    ServletContext context = request.getSession().getServletContext();

    cargarConsultorio(context);
    cargarEscolaridad (context);
    cargarEstadoCivil (context);
    cargarNacionalidad (context);
    cargarOcupacion (context);
    cargarReligion (context);
    cargarTurno (context);
}

```

```

    cargarUnidad(context);
    cargarRoles(context);
    cargarSexo(context);
}

private void cargarSexo(ServletContext context)
{
    // TODO Auto-generated method stub
    Map<Long, String> sexo = _administradorSexo.CargarSexo();
    context.setAttribute( "sexos", sexo );
}
/**
 * @param context
 */
private void cargarRoles( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> roles = _administradorRoles.cargarRoles();
    context.setAttribute( "roles", roles );
}

/**
 * @param context
 */
private void cargarUnidad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> unidad = _administradorUnidad.CargarUnidad();
    context.setAttribute( "unidad", unidad );
}

/**
 * @param context
 */
private void cargarTurno( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> turno = _administradorTurno.CargarTurno();
    context.setAttribute( "turno", turno );
}

/**
 * @param context
 */
private void cargarReligion( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> religion = _administradorReligion.CargarReligion();
    context.setAttribute( "religion", religion );
}

/**
 * @param context
 */
private void cargarOcupacion( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> ocupacion =
        _administradorOcupacion.CargarOcupacion();
    context.setAttribute( "ocupacion", ocupacion );
}

```

```

}

/**
 * @param context
 */
private void cargarNacionalidad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> nacionalidad =
        _administradorNacionalidad.CargarNacionalidad();
    context.setAttribute( "nacionalidad", nacionalidad );
}

/**
 * @param context
 */
private void cargarEstadoCivil( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> estadoCivil =
        _administradorEstadoCivil.CargarEstadoCivil();
    context.setAttribute( "estadoCivil", estadoCivil );
}

/**
 * @param context
 */
private void cargarConsultorio( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> consultorios =
        _administradorConsultorio.CargarConsultorio();
    context.setAttribute( "consultorios", consultorios );
}

/**
 * @param context
 */
private void cargarEscolaridad( ServletContext context )
{
    // TODO Auto-generated method stub
    Map<Long, String> escolaridades =
        _administradorEscolaridad.CargarEscolaridad();
    context.setAttribute( "escolaridad", escolaridades );
}

private UsuarioDTO cargarUsuario(UsuarioJoinDTO uj)
{
    // TODO Auto-generated method stub
    UsuarioDTO user = new UsuarioDTO();
    user.setApellidoMaterno( uj.getApellidoMaterno() );
    user.setApellidoPaterno( uj.getApellidoPaterno() );
    user.setCedula( uj.getCedula() );
    user.setContrasenia( uj.getCedula() );
    user.setDireccion( uj.getDireccion() );
    user.setEdad( uj.getEdad() );
    user.setEscolaridad( uj.getEscolaridad().getId() );
    user.setEstadoCivil( uj.getEstadoCivil().getId() );
    user.setFechaNacimiento( uj.getFechaNacimiento() );
    user.setId( uj.getId() );
    user.setNacionalidad( uj.getNacionalidad().getId() );
}

```



```

user.setNombre( uj.getNombre() );
user.setRol( uj.getRol().getId() );
user.setSexo( uj.getSexo().getIdSexo() );
user.setTelefono( uj.getTelefono() );
user.setTipo( uj.getTipo() );
user.setTurno( uj.getTurno().getId() );
user.setUnidad( uj.getUnidad().getId() );
user.setUsuario( uj.getUsuario() );
user.setVersion( uj.getVersion() );

return user;
}
/*****Getters and
Setters*****/

/*
 * @author Daniel
 */
@Override
public void setSession( Map sessionData )
{
// TODO Auto-generated method stub
_sessionData = sessionData;
}
public Map getSessionData()
{
return _sessionData;
}

public PagedResults<UsuarioJoinDTO> getUsuarios()
{
return _usuarios;
}

public FiltroUsuarios getFiltro()
{
return _filtro;
}

public void setUsuarios(PagedResults<UsuarioJoinDTO> usuarios )
{
_usuarios = usuarios;
}

public void setFiltro( FiltroUsuarios filtro )
{
_filtro = filtro;
}

public void setAdministradorUsuarios(
AdministradorUsuarios administradorUsuarios )
{
_administradorUsuarios = administradorUsuarios;
}

public long getIdUsuario()
{
return _idUsuario;
}

public UsuarioJoinDTO getUsuarioJoin()

```

```

{
    return _usuarioJoin;
}

public void setIdUsuario( long idUsuario )
{
    _idUsuario = idUsuario;
}

public void setUsuarioJoin( UsuarioJoinDTO usuarioJoin )
{
    _usuarioJoin = usuarioJoin;
}

public UsuarioDTO getUsuario()
{
    return _usuario;
}

public void setUsuario( UsuarioDTO usuario )
{
    _usuario = usuario;
}

public void setAdministradorConsultorio(
    AdministradorConsultorio administradorConsultorio )
{
    _administradorConsultorio = administradorConsultorio;
}

public void setAdministradorEscolaridad(
    AdministradorEscolaridad administradorEscolaridad )
{
    _administradorEscolaridad = administradorEscolaridad;
}

public void setAdministradorEstadoCivil(
    AdministradorEstadoCivil administradorEstadoCivil )
{
    _administradorEstadoCivil = administradorEstadoCivil;
}

public void setAdministradorNacionalidad(
    AdministradorNacionalidad administradorNacionalidad )
{
    _administradorNacionalidad = administradorNacionalidad;
}

public void setAdministradorOcupacion(
    AdministradorOcupacion administradorOcupacion )
{
    _administradorOcupacion = administradorOcupacion;
}

public void setAdministradorReligion(
    AdministradorReligion administradorReligion )
{
    _administradorReligion = administradorReligion;
}

```

```

public void setAdministradorTurno( AdministradorTurno
    administradorTurno )
{
    _administradorTurno = administradorTurno;
}

public void setAdministradorUnidad( AdministradorUnidad
    administradorUnidad )
{
    _administradorUnidad = administradorUnidad;
}

public void setAdministradorRoles( AdministradorRoles
    administradorRoles )
{
    _administradorRoles = administradorRoles;
}

public String getContrasenia()
{
    return _contrasenia;
}

public void setContrasenia( String contrasenia )
{
    _contrasenia = contrasenia;
}

public String getContraseniaActual()
{
    return _contraseniaActual;
}

public void setContraseniaActual( String contraseniaActual )
{
    _contraseniaActual = contraseniaActual;
}

public String getContraseniaUsuario()
{
    return _contraseniaUsuario;
}

public void setContraseniaUsuario( String contraseniaUsuario )
{
    _contraseniaUsuario = contraseniaUsuario;
}

public void setAdministradorSexo( AdministradorSexo
    administradorSexo )
{
    _administradorSexo = administradorSexo;
}

/*
 * @author Daniel
 */
@Override
public void prepare() throws Exception
{
    // TODO Auto-generated method stub

```

```
    _filtro = ( FiltroUsuarios)_sessionData.get( "filtroUsuario" );  
  
    if (_filtro == null)  
    {  
        _filtro = new FiltroUsuarios();  
        sessionData.put( "filtroUsuario", _filtro );  
    }  
}  
  
}
```

Pruebas y Liberación

Referencias

- I. "Norma Oficial Mexicana NOM-168-SSA1-1998, del Expediente Clínico".
<http://www.ssa.org.mx>

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e ingeniería

Ingeniería en Computación

**Expediente Clínico Electrónico.
ECE**

Manual de Usuario

Alumno:

**Castrejon Morales Daniel
204358724**

Asesorado por:

**M. en C. Rafaela Blanca Silva López
No. Económico: 17114**

**Trimestre 10P
09 de Septiembre de 2010**

Autenticarse en el sistema.

Para poder acceder al sistema y poder realizar las notas médicas o revisar notas de los pacientes registrados.

HOSPITAL PILOTO

Expediente Clínico Electrónico

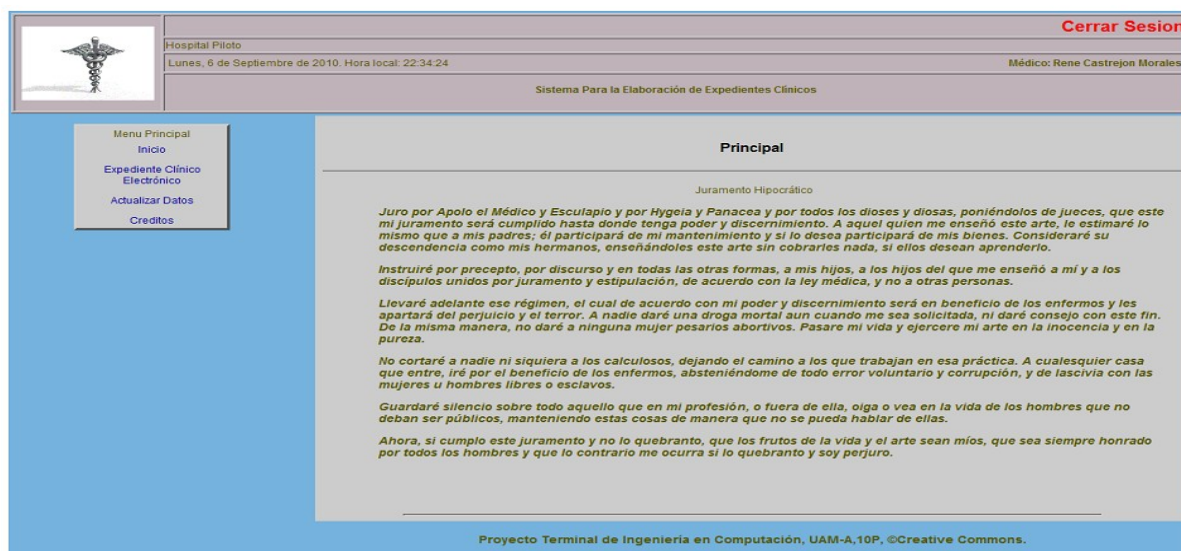
Usuario:

Contraseña:

Validar Limpicar

Ilustración : Inicio

1. Ingrese su nombre de usuario que registro en el sistema cuando tomaron sus datos.
2. Ingrese su contraseña que registro en el sistema cuando tomaron sus datos.
3. Presione el botón validar para autenticarse en el sistema.
4. Presione limpiar solo cuando quiera borrar los datos ingresados en los campos de usuario y/o contraseña.



Hospital Piloto
Lunes, 6 de Septiembre de 2010. Hora local: 22:34:24
Médico: Rene Castrejon Morales
Sistema Para la Elaboración de Expedientes Clínicos

Cerrar Sesión

Menu Principal
Inicio
Expediente Clínico Electrónico
Actualizar Datos
Creditos

Principal

Juramento Hipocrático

Juro por Apolo el Médico y Esculapio y por Hygeia y Panacea y por todos los dioses y diosas, poniéndolos de jueces, que este mi juramento será cumplido hasta donde tenga poder y discernimiento. A aquel quien me enseñó este arte, le estimaré lo mismo que a mis padres; él participará de mi mantenimiento y si lo desea participará de mis bienes. Consideraré su descendencia como mis hermanos, enseñándoles este arte sin cobrarles nada, si ellos desean aprenderlo.

Instruiré por precepto, por discurso y en todas las otras formas, a mis hijos, a los hijos del que me enseñó a mí y a los discípulos unidos por juramento y estipulación, de acuerdo con la ley médica, y no a otras personas.

Llevaré adelante ese régimen, el cual de acuerdo con mi poder y discernimiento será en beneficio de los enfermos y les apartará del perjuicio y el terror. A nadie daré una droga mortal aun cuando me sea solicitada, ni daré consejo con este fin. De la misma manera, no daré a ninguna mujer pesarios abortivos. Pasare mi vida y ejercere mi arte en la inocencia y en la pureza.

No cortaré a nadie ni siquiera a los calculosos, dejando el camino a los que trabajan en esa práctica. A cualesquier casa que entre, iré por el beneficio de los enfermos, absteniéndome de todo error voluntario y corrupción, y de lascivia con las mujeres u hombres libres o esclavos.

Guardaré silencio sobre todo aquello que en mi profesión, o fuera de ella, oiga o vea en la vida de los hombres que no deban ser públicos, manteniendo estas cosas de manera que no se pueda hablar de ellas.

Ahora, si cumplo este juramento y no lo quebranto, que los frutos de la vida y el arte sean míos, que sea siempre honrado por todos los hombres y que lo contrario me ocurra si lo quebranto y soy perjuro.

Proyecto Terminal de Ingeniería en Computación, UAM-A,10P, ©Creative Commons.

esta registrado aparecerá la pantalla principal, en caso contrario el sistema volverá desplegar la pantalla de inicio con un mensaje de error.

Buscar Paciente Registrado.

El usuario (administrador o médico) debe autenticarse en el sistema, ver [autenticarse](#).

Hospital Piloto
Lunes, 6 de Septiembre de 2010. Hora local: 22:34:24
Médico: Rene Castrejon Morales

Sistema Para la Elaboración de Expedientes Clínicos

Menu Principal
Inicio
Expediente Clínico Electrónico
Actualizar Datos
Creditos

Principal

Juramento Hipocrático

Juro por Apolo el Médico y Esculapio y por Hygeia y Panacea y por todos los dioses y diosas, poniéndolos de jueces, que este mi juramento será cumplido hasta donde tenga poder y discernimiento. A aquel quien me enseñó este arte, le estimaré lo mismo que a mis padres; él participará de mi mantenimiento y si lo desea participará de mis bienes. Consideraré su descendencia como mis hermanos, enseñándoles este arte sin cobrarles nada, si ellos desean aprenderlo.

Instruiré por precepto, por discurso y en todas las otras formas, a mis hijos, a los hijos del que me enseñó a mí y a los discípulos unidos por juramento y estipulación, de acuerdo con la ley médica, y no a otras personas.

Llevaré adelante ese régimen, el cual de acuerdo con mi poder y discernimiento será en beneficio de los enfermos y les apartará del perjuicio y el terror. A nadie daré una droga mortal aun cuando me sea solicitada, ni daré consejo con este fin. De la misma manera, no daré a ninguna mujer pesarios abortivos. Pasare mi vida y ejerceré mi arte en la inocencia y en la pureza.

No cortaré a nadie ni siquiera a los calculosos, dejando el camino a los que trabajan en esa práctica. A cualesquier casa que entre, iré por el beneficio de los enfermos, absteniéndome de todo error voluntario y corrupción, y de lascivia con las mujeres u hombres libres o esclavos.

Guardaré silencio sobre todo aquello que en mi profesión, o fuera de ella, oiga o vea en la vida de los hombres que no deban ser públicos, manteniendo estas cosas de manera que no se pueda hablar de ellas.

Ahora, si cumplo este juramento y no lo quebranto, que los frutos de la vida y el arte sean míos, que sea siempre honrado por todos los hombres y que lo contrario me ocurra si lo quebranto y soy perjuro.

Proyecto Terminal de Ingeniería en Computación, UAM-A, 10P, ©Creative Commons.

principal, en la parte de menú principal, dar click en el link *Expediente Clínico Electrónico*.

1. En la pantalla buscar paciente, ingresar el nombre o parte del nombre del paciente que se está buscando o se requiere obtener información.
2. Presionar el botón buscar, en esta misma pantalla aparecerá una lista de pacientes que su nombre coincide con la el nombre o parte de nombre ingresado en 2, además de información más relevante del paciente.
3. Si encontramos en la lista al paciente que estamos buscando, dar click al link de identificador del paciente y nos llevara a la pantalla datos del paciente, donde tenemos la información del paciente y un menú de opciones para manipular ya sea información del paciente, y así como realizar servicios médicos que necesite.
4. En caso de no encontrar el paciente en esta lista, y si existen más registros, podemos seguir buscando en nuevas listas presionando los botones para ir a la siguiente lista (5.a) o hasta la última lista (5.b), o presionando el numero de lista en la que queramos buscar (5.c). Esta etiqueta nos indica el número de listas encontradas (5.d).

Hospital Piloto
Lunes, 6 de Septiembre de 2010. Hora local: 23:17:37
Médico: Rene Castrejon Morales

Sistema Para la Elaboración de Expedientes Clínicos

Menu Principal
Inicio
Expediente Clínico Electrónico
Actualizar Datos
Creditos

BUSCAR PACIENTES

2 3 buscar

Pacientes - (Pagina 1 de 2) 5d

ID	CURP	Nombres	Apellido Paterno	Apellido Materno	Estatus	Vive
1	CAMD821121HGRSRN05	Daniel	Castrejon	Morales	true	true
2	PEGP010100HGRSRN02	Pedro	Perez	Gomez	false	true
3	LOAP130709HGRSRN09	Pancho	Lopez	Alvarez	true	true

4


Pagina 1 > 2 5c

Insertar Paciente

5a 5b

Proyecto Terminal de Ingeniería en Computación, UAM-A,10P, ©Creative Commons.

- Si no encontramos el paciente buscado en ninguna de las listas, revisar el nombre escrito en 2, si esta correcto y aun no encontramos al paciente es que aun no está registrado en el sistema y tendremos que registrar al paciente, ver [Registrar Paciente](#).



Cerrar Sesión
Hospital Piloto
Lunes, 6 de Septiembre de 2010. Hora local: 22:47:41
Médico: Rene Castrejon Morales

Sistema Para la Elaboración de Expedientes Clínicos

Menu Principal


Inicio

Expediente Clínico
Electrónico

Actualizar Datos


Creditos

Paciente

Hospitalizar
Ver Expediente
Nueva Recomendacion
Listar Recomendaciones


Datos del Paciente	
Identificador:	2
Curp:	PEGP010100HGRSRN02
Nombre:	Pedro
Apellido Paterno:	Perez
Apellido Materno:	Gomez
Fecha de Nacimiento:	24/06/2010
Edad:	0
Sexo:	Masculino
Dirección:	Direccion
Telefono Particular:	12345678
Lugar de Origen:	Guerrero
Etnicidad:	
Familiar Responsable:	Pancho Lopez Lopez
Telefono Familiar:	12345678
Nacionalidad:	Mexicana
Religion:	Catolico
Estado Civil:	Soltero
Ocupación:	Estudiante
Escolaridad:	Primaria Completa
Unidad YU Hospital:	Hospital Piloto
Consultorio:	1
Turno:	Matutino
Estatus:	false
Vive:	true

Actualizar Datos

Listar notas
Listar Recomendaciones


Proyecto Terminal de Ingeniería en Computación, UAM-A,10P, ©Creative Commons.

Ilustración : Datos del paciente

Hospitalizar Paciente.

Debemos tener la pantalla datos del paciente, el paciente no debe estar hospitalizado, ver [Buscar Paciente Registrado](#). Hasta el punto número 5.



The screenshot shows a web interface titled "Paciente". At the top, there is a navigation bar with four buttons: "Hospitalizar", "Ver Expediente", "Nueva Recomendacion", and "Listar Recomendaciones". The "Hospitalizar" button is highlighted with a red circle and the number "1". Below the navigation bar is a table titled "Datos del Paciente" with the following data:

Datos del Paciente	
Identificador:	2
Curp:	PEGP010100HGRSRN02
Nombre:	Pedro
Apellido Paterno:	Perez
Apellido Materno:	Gomez
Fecha de Nacimiento:	24/06/2010
Edad:	0
Sexo:	Masculino
Dirección:	Direccion
Telefono Particular:	12345678
Lugar de Origen:	Guerrero

Ilustración : Datos del paciente

1. Presionar el botón hospitalizar. Este botón no es visible en el caso que el paciente ya este hospitalizado, el sistema desplegara la pantalla de hospitalizar paciente.

Hospitalizar Paciente

Los campos con * son Requeridos

Datos del Paciente	
Identificador del Paciente:	2
Nombre del Paciente:	Pedro Perez Gomez
Datos del Medico	
Identificador del Medico:	2
Nombre del Medico:	Rene Castrejon Morales
Insertar Datos	
*Fecha de Admisión:	<input type="text" value=""/>
*Especialidad:	Urgencias
*Diagnostico:	<input style="height: 40px;" type="text"/>
*No. de Cama:	<input type="text" value=""/>
<input type="button" value="Hospitalizar"/>	

Ilustración : Hospitalizar paciente

2. En esta pantalla podemos observar datos del paciente, así como también datos del médico, que está realizando la hospitalización.
3. Presionar el calendario y se desplegara un calendario, buscar y seleccionar la fecha de admisión del paciente.
4. Buscar y seleccionar la especialidad donde será hospitalizado el paciente.
5. Ingresar diagnostico del paciente a hospitalizar.
6. Ingresar el número de cama en cual se encontrara el paciente.
7. Presionar hospitalizar para finalizar el registro, se guarda los datos y nos muestra la pantalla de datos del paciente.

Ver Expediente Clínico

Debemos tener la pantalla de datos del paciente, ver [Buscar Paciente Registrado](#). U [Hospitalizar Paciente](#). Para poder llegar a esta pantalla. El expediente nos muestra los datos de la hospitalización del paciente ya sea anteriores o la actual y en caso de que aun este hospitalizado se podrá generar notas medicas.

Paciente

Ver Expediente Nueva Recomendacion Listar Recomendaciones

1

Datos del Paciente	
Identificador:	2
Curp:	PEGP010100HGRSRN02
Nombre:	Pedro
Apellido Paterno:	Perez
Apellido Materno:	Gomez
Fecha de Nacimiento:	24/06/2010

Ilustración : Datos del Paciente

1. Presionar el botón ver expediente, el sistema desplegara la pantalla Expedientes del paciente con una lista histórica de las hospitalizaciones del paciente de la más reciente a la más antigua.

Expedientes del Paciente

Expedientes				
# Expediente	Identificador Paciente	Identificador Usuario	Diagnostico	Fecha de Admision
13	2	2	Diagnostico	07/09/2010
11	2	1	Diagnostico	21/07/2010
2	2	2	hjk	05/07/2010

2

Ilustración : Expedientes del paciente

2. Dar click al número de expediente, el sistema desplegara la pantalla datos de la hospitalización, si la hospitalización es la reciente y el paciente este aun hospitalizado podrá agregar nuevas notas médicas.

Expediente No. 11

Datos	
Nombre Paciente:	Pedro Perez Gomez
Especialidad	Terapia Intensiva
Fecha de Admision	07/21/2010
Diagnostico	Diagnostico
Cama	5
Elaboro	Daniel Castrejon Morales
Notas Anteriores	
Insertar Nuevas notas	
Nota de Ingreso	
Nota de Interconsulta	
Nota Médica de Anestesiología	
Nota de Evolucion	
Nota de Contrereferencia	

Ilustración : Datos del expediente

3. Dar click al *ver notas anteriores*, y el sistema mostrara la pantalla notas del expediente con una lista de todas las notas medicas realizadas en ese expediente, ir [Ver Notas Medicas](#) para mayor detalle de esta pantalla.
4. Dar click en el nombre de alguna nota para iniciar el registro de una nueva nota, ver [Insertar Nota Médica](#) para mayor información de registro de nueva nota.

Notas		
# Nota	Tipo nota	Fecha Creacion
30	Nota de Anestesiología	17/08/2010 23:02:12
29	Nota de Ingreso	17/08/2010 22:33:44
26	Nota de Anestesiología	17/08/2010 00:00:00
28	Nota de Ingreso	17/08/2010 00:00:00
27	Nota de Anestesiología	17/08/2010 00:00:00
22	Nota de Interconsulta	21/07/2010 00:00:00
25	Nota de Anestesiología	21/07/2010 00:00:00
17	Nota de Interconsulta	18/07/2010 23:29:25
16	Nota de Ingreso	18/07/2010 22:58:03
15	Nota de Interconsulta	18/07/2010 14:07:27
14	Nota de Interconsulta	18/07/2010 13:27:31
13	Nota de Interconsulta	18/07/2010 13:03:24
12	Nota de Ingreso	18/07/2010 00:00:00

Ilustración : Notas del expediente

Ver Notas Médicas

Debemos tener la pantalla notas del expediente, ir [Ver Expediente Clínico](#) punto número 3. Podemos observar en esta pantalla el historial de las notas médicas realizadas en el expediente de la más reciente hasta la más antigua.

Notas		
# Nota	Tipo nota	Fecha Creacion
30	Nota de Anestesiología	17/08/2010 23:02:12
29	Nota de Ingreso	17/08/2010 22:33:44
26	Nota de Anestesiología	17/08/2010 00:00:00
28	Nota de Ingreso	17/08/2010 00:00:00
27	Nota de Anestesiología	17/08/2010 00:00:00
22	Nota de Interconsulta	21/07/2010 00:00:00
25	Nota de Anestesiología	21/07/2010 00:00:00
17	Nota de Interconsulta	18/07/2010 23:29:25
16	Nota de Ingreso	18/07/2010 22:58:03
15	Nota de Interconsulta	18/07/2010 14:07:27
14	Nota de Interconsulta	18/07/2010 13:27:31
13	Nota de Interconsulta	18/07/2010 13:03:24
12	Nota de Ingreso	18/07/2010 00:00:00

Ilustración : Notas del Expediente

1. Seleccionar un número de nota para observar la información correspondiente a cada tipo o nombre de la nota, el sistema desplegará la pantalla correspondiente a cada tipo de nota.
2. En la pantalla de la nota elegida podemos ver signos vitales del paciente, datos de la hospitalización y los registros correspondientes a la nota.
3. Presionar para realizar un comentario a la nota, el sistema desplegará la pantalla de registro de comentarios ir insertar comentario.
4. Presionar para ver los comentarios realizados a la nota en cuestión, ir ver comentarios.
5. Repetir desde el paso número 1 para ver otras notas del expediente.

	Hospital Piloto	Hospital:	Hospital Piloto
	EXPEDIENTE CLÍNICO ELECTRÓNICO	Médico Tratante:	Rene Castrejon Morales
	Nota de Ingreso	Nombre del Paciente:	Pedro Perez Gomez
		Servicio:	Urgencias
	Expediente:	2	
	Cama:	5	


Fecha y Hora: 18/07/2010 22:58:03	Motivo de Ingreso: <input type="text" value="motivo ingreso"/>
SIGNOS VITALES Peso: <input type="text" value="80.0"/> Kg Talla: <input type="text" value="1.79"/> m Temperatura: <input type="text" value="36.0"/> °C Frec. Cardíaca: <input type="text" value="89.0"/> lpm Frec. Respiratoria: <input type="text" value="78.0"/> rpm Tension Arterial: <input type="text" value="99.0"/> mmHg IMC: <input type="text" value="24.96801"/> Kg/m2	Antecedentes <input type="text" value="antecedentes"/>
	Padecimiento Actual <input type="text" value="padecimiento actual"/>
	Exploración Física <input type="text" value="exploracion fisica"/>
	EXAMENES DE LABORATORIO Y GABINETE <input type="text" value="examens de laboratorio y gabinete"/>
	Impresión Diagnóstica: <input type="text" value="impresion diagnostica"/>
Estado de Salud: <input type="text" value="estado de salud"/>	Pronóstico: <input type="text" value="pronostico"/>
Médico: Rene Castrejon Morales	Identificador: 2

3

4

Insertar nota médica.

Debemos de tener la pantalla datos de la hospitalización ir ver expediente punto número 2, dependiendo del tipo de nota que queramos realizar será la pantalla que el sistema despliegue, cada nota pide datos diferentes, solo se explicara la nota de ingreso las demás notas son similares e intuitivas.



Datos	
Nombre Paciente:	Pedro Perez Gomez
Especialidad	Terapia Intensiva
Fecha de Admision	07/21/2010
Diagnostico	Diagnostico
Cama	5
Elaboro	Daniel Castrejon Morales

[Notas Anteriores](#)

Insertar Nuevas notas

- [Nota de Ingreso](#) **1**
- [Nota de Interconsulta](#)
- [Nota Médica de Anestesiología](#)
- [Nota de Evolucion](#)
- [Nota de Contrereferencia](#)

⏪

Ilustración : Datos de la hospitalización

1. Seleccionar el tipo de nota a registrar al paciente seleccionado, el sistema desplegara la pantalla de registro de signos vitales.

Insertar Signos Vitales

Los campos con * son requeridos

SIGNOS VITALES	
*Peso:	<input type="text"/> Kg.
*Talla:	<input type="text"/> mts.
*Temperatura:	<input type="text"/> °C.
*Frec. Cardiaca:	<input type="text"/> lpm.
*Frec. Respiratoria:	<input type="text"/> rpm.
*Tension Arterial:	<input type="text"/> mmHg.
<input type="button" value="Aceptar"/>	

Ilustración : Signos vitales

2. Ingresar el peso del paciente.
3. Ingresar la talla del paciente.
4. Ingresar la temperatura del paciente.
5. Ingresar la frecuencia cardiaca.
6. Ingresar la frecuencia respiratoria.
7. Ingresar tensión arterial.
8. Presionar aceptar y el sistema desplegara la pantalla correspondiente al registro de la nota médica correspondiente.
9. Llenar los campos obligatorios de la nota para poder continuar.
10. Presionar terminar la nota para guardar los datos ingresados de la nota, el sistema desplegara la pantalla de datos de la hospitalización.
11. Repetir desde el paso numero uno para realizar una nueva nota

	Hospital Piloto	<div style="border: 2px solid orange; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto;"> 9 </div> <div style="border-left: 2px solid orange; border-right: 2px solid orange; width: 50px; height: 80px; margin: 0 auto; position: relative;"> <div style="position: absolute; bottom: -10px; left: 50%; transform: translate(-50%, 50%); font-size: 24px; font-weight: bold;">↓</div> </div>	Hospital: Hospital Piloto
	EXPEDIENTE CLÍNICO ELECTRÓNICO		Médico Tratante: Rene Castrejon Morales
	NOTA DE INGRESO		Nombre del Paciente: Pedro Perez Gomez
			Servicio: Urgencias
	Expediente: 2	Cama: 5	

Fecha y Hora: 07/09/2010 01:17:25	Motivo de Ingreso: <input type="text"/>
SIGNOS VITALES Peso: <input type="text" value="80.0"/> Kg Talla: <input type="text" value="80.0"/> m Temperatura: <input type="text" value="37.0"/> °C Frec. Cardíaca: <input type="text" value="79.0"/> lpm Frec. Respiratoria: <input type="text" value="69.0"/> rpm Tension Arterial: <input type="text" value="56.0"/> mmHg IMC: <input type="text" value="0.0125"/> Kg/m2	Antecedentes <div style="border: 1px solid gray; height: 80px; width: 100%;"></div>
	Padecimiento Actual <div style="border: 1px solid gray; height: 80px; width: 100%;"></div>
	Exploración Física <div style="border: 1px solid gray; height: 80px; width: 100%;"></div>
	EXAMENES DE LABORATORIO Y GABINETE <div style="border: 1px solid gray; height: 80px; width: 100%;"></div>
	Impresión Diagnóstica: <div style="border: 1px solid gray; height: 80px; width: 100%;"></div>
Estado de Salud: <input type="text"/> Médico: Rene Castrejon Morales	Pronóstico: <input type="text"/> Identificador: 2 <div style="text-align: right;"> <input type="button" value="Firmar y Terminar"/> <div style="border: 2px solid orange; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-left: 10px;"> 10 </div> </div>

Ilustración : Nota Ingreso

Registrar Paciente.

Debemos tener la pantalla de buscar paciente ir buscar paciente registrado, para el registro de pacientes si se trata de de una persona desconocida, debe de ingresar datos propicios para poder identificar al paciente y darle la atención médica que necesite.

BUSCAR PACIENTES

Nombre:

Pacientes - (Pagina 1 de 2)

ID	CURP	Nombres	Apellido Paterno	Apellido Materno	Estatus	Vive
1	CAMD821121HGRSRN05	Daniel	Castrejon	Morales	true	true
2	PEGP010100HGRSRN02	Pedro	Perez	Gomez	false	true
3	LOAP130709HGRSRN09	Pancho	Lopez	Alvarez	true	true

Pagina 1 > 2

1

Ilustración : Insertar paciente

1. En la pantalla de buscar paciente, presionar el botón insertar paciente, el sistema desplegara la pantalla para registrar paciente nuevo.

Registrar Paciente

Los Campos con * son obligatorios.
Nota: si desconoce algun campo coloque "desconocido".

Datos	
*Nombre:	<input type="text"/>
*Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
*Fecha de Nacimiento:	<input type="text"/>
*CURP:	<input type="text"/>
*Sexo:	<input type="radio"/> Masculino <input type="radio"/> Femenin
*Dirección:	<input type="text"/>
*Telefono:	<input type="text"/>
*Familiar Responsable:	<input type="text"/> <input type="button" value="Registrar Familiar"/>
Lugar de Origen:	<input type="text"/>
Etnicidad:	<input type="text"/>
*Nacionalidad:	Mexicana
*Estado Civil:	Soltero
*Escolaridad:	Sin Estudios
*Ocupación:	Estudiante
*Religión:	Catolico
Unidad:	
*Consultorio:	Rene Castrejon Morales
*Turno:	Matutino
Hospitalizado:	Habilitado <input type="checkbox"/>

*Indicar que solo el paciente está hospitalizado.

Ilustración : Registrar paciente

2. Ingresar los datos, como son nombre, apellido paterno, apellido materno, fecha de nacimiento, CURP, sexo, dirección, teléfono del paciente a registrar.
3. Presionar el botón registrar familiar responsable, y el sistema desplegara la pantalla para insertar el familiar responsable del paciente.

Registrar Familiar Responsable

Los Campos con * son obligatorios.
Nota: si desconoce algun campo coloque "desconocido".

Datos	
*Nombre:	<input type="text"/>
*Apellido Paterno	<input type="text"/>
Apellido Materno:	<input type="text"/>
*Dirección:	<input type="text"/>
*Telefono:	<input type="text"/>
*Parentesco	<input type="text"/>

Registrar Familiar

Ilustración : Insertar familiar responsable

4. Insertar los datos como son nombre, apellido paterno, apellido materno, dirección, teléfono y parentesco con el paciente a registrar del familiar responsable.
5. Presionar registrar familiar y el sistema desplegara la pantalla de registro de paciente, para continuar con el llenado del registro.
6. Ingresar los datos faltantes en el registro del paciente, los datos proporcionados en 2, se quedan guardados.
7. Presionar el botón registrar paciente, el sistema realizara el registro del paciente y desplegara la pantalla de buscar paciente, si se busca el paciente recién ingresado debe ya de existir el registro.

Insertar Recomendación

Debemos tener la pantalla datos del paciente, para poder realizar recomendaciones a los pacientes registrados en el sistema.

Paciente

Hospitalizar Ver Expediente Nueva Recomendación Listar Recomendaciones

Datos del Paciente	
Identificador:	2
Curp:	PEGP010100HGRSRN02
Nombre:	Pedro
Apellido Paterno:	Perez
Apellido Materno:	Gomez
Fecha de Nacimiento:	24/06/2010

Ilustración : Datos del paciente

1. Presionar el botón nueva recomendación, el sistema desplegara la pantalla registro de recomendación, nos muestra información del paciente así como del médico que está realizando la recomendación.

Recomendación

Recomendaciones									
Numero	31								
Nombre del Paciente	Pedro Perez Gomez								
Hospital	Hospital Piloto								
Fecha y hora	09/09/2010 01:11:49								
Insertar Recomendaciones									
	<table border="1"> <thead> <tr> <th>Numero</th> <th>Medicamento</th> <th>Mode de Uso</th> <th>Recomendacion</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="text-align: center;">Insertar recomendación</td> </tr> </tbody> </table>	Numero	Medicamento	Mode de Uso	Recomendacion	Insertar recomendación			
Numero	Medicamento	Mode de Uso	Recomendacion						
Insertar recomendación									
Nombre del médico	Rene Castrejon Morales								
Finalizar									

Ilustración : Registro de recomendación

2. Presionar insertar recomendación y el sistema desplegara la pantalla para ingresar los datos correspondientes a la recomendación.

Detalle de las Recomendaciones

Los campos con * son requeridos.

Detalle	
Numero Receta:	31
Medicamento:	<input type="text"/>
Modo de uso:	<input type="text"/>
*Recomendacion:	<input type="text"/>
<input type="button" value="Insertar"/>	

Ilustración : Detalle de recomendación

3. Ingresar el nombre del medicamento.
4. Ingresar el modo de uso del medicamento.
5. Insertar alguna recomendación con respecto al medicamento.
6. Presionar insertar para guardar esa recomendación, el sistema desplegara la pantalla registro de recomendación.

Recomendación

Recomendaciones													
Numero	32												
Nombre del Paciente	Pedro Perez Gomez												
Hospital	Hospital Piloto												
Fecha y hora	09/09/2010 01:22:23												
Insertar Recomendaciones													
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #d9e1f2;"> <th style="width: 10%;">Numero</th> <th style="width: 20%;">Medicamento</th> <th style="width: 20%;">Mode de Uso</th> <th style="width: 45%;">Recomendacion</th> </tr> </thead> <tbody> <tr> <td>27</td> <td>Paracetamol</td> <td>Tomese una cada 8 horas</td> <td>Si se presenta alergia, suspendase</td> </tr> <tr> <td colspan="4" style="text-align: center;"> <input type="button" value="Insertar recomendacion"/> </td> </tr> </tbody> </table>	Numero	Medicamento	Mode de Uso	Recomendacion	27	Paracetamol	Tomese una cada 8 horas	Si se presenta alergia, suspendase	<input type="button" value="Insertar recomendacion"/>			
Numero	Medicamento	Mode de Uso	Recomendacion										
27	Paracetamol	Tomese una cada 8 horas	Si se presenta alergia, suspendase										
<input type="button" value="Insertar recomendacion"/>													
Nombre del médico	Rene Castrejon Morales												
<input type="button" value="Finalizar"/>													

Ilustración : Insertar Recomendación

7. Podemos observar que la nueva recomendación aparece en la pantalla, si queremos insertar otra nueva recomendación, repetir desde 2, presionar el botón finalizar para insertar el registro de las recomendaciones registradas, el sistema desplegara la pantalla de datos del paciente.

Ver recomendaciones

Debemos tener la pantalla datos del paciente, para poder realizar recomendaciones a los pacientes registrados en el sistema.

1. Presionar el botón listar recomendaciones, el sistema desplegara la pantalla con la lista de recomendaciones hechas al paciente, de fecha más reciente a la mas antigua.



The screenshot shows a web interface titled "Paciente". At the top, there is a navigation bar with four buttons: "Hospitalizar", "Ver Expediente", "Nueva Recomendacion", and "Listar Recomendaciones". To the right of these buttons is a purple circular icon with a white arrow. Below the navigation bar is a table titled "Datos del Paciente" with the following data:

Datos del Paciente	
Identificador:	2
Curp:	PEGP010100HGRSRN02
Nombre:	Pedro
Apellido Paterno:	Perez
Apellido Materno:	Gomez
Fecha de Nacimiento:	24/06/2010

An orange arrow points from the "Listar Recomendaciones" button to a circled number "1" on the right side of the table.

Ilustración : Datos del paciente

Recomendaciones del Paciente			
Recomendaciones			
# Recomendacion	Nombre Paciente	Identificador Usuario	Fecha Creacion
32	2	2	09/09/2010 01:22:23
31	2	2	09/09/2010 01:11:49
29	2	1	21/07/2010 18:05:15
26	2	1	15/07/2010 14:47:31
25	2	1	12/07/2010 23:35:07
24	2	2	05/07/2010 22:47:00
23	2	2	05/07/2010 18:24:05
22	2	2	05/07/2010 18:22:38
21	2	2	05/07/2010 18:19:38
20	2	2	05/07/2010 18:18:29
19	2	2	05/07/2010 18:17:59
18	2	2	05/07/2010 18:17:21
17	2	2	05/07/2010 18:15:52
16	2	2	05/07/2010 18:13:23
15	2	2	05/07/2010 18:09:28
9	2	2	05/07/2010 17:30:32

Ilustración : Lista de Recomendaciones

2. Presionar en el link para el detalle de esa recomendación, el sistema desplegara la pantalla con el detalle de la recomendación seleccionada.

Recomendación				
Recomendaciones				
Numero	32			
Nombre del Paciente	Pedro Perez Gomez			
Hospital	Hospital Piloto			
Fecha y hora	09/09/2010 01:22:23			
Recomendaciones				
	Numero	Medicamento	Mode de Uso	Recomendacion
	27	Paracetamol	Tomese una cada 8 horas	Si se presenta alergia, suspendase
Nombre del médico	Rene Castrejon Morales			

Ilustración : Detalle recomendación

Actualizar Datos Personales

Para poder actualizar datos personales debe autenticarse, ir [autenticarse](#), en el menú principal tenemos la liga para realizar cambios a sus datos personales.

Proyecto Terminal de Ingeniería en Computación, UAM-A, 10P, ©Creative commons.



Ilustración : Menú principal

1. Presionar en link *Actualizar datos personales* y el sistema desplegara la pantalla datos del usuario, donde nos muestra información del usuario registrado.

Usuario

Datos del usuario	
Identificador:	2
Cedula:	s
Nombre:	Rene
Apellido Paterno:	Castrejon
Apellido Materno:	Morales
Fecha de Nacimiento:	12/11/1974
Edad:	35
Sexo:	Masculino
Dirección:	Puente de Vigas
Telefono:	5553949432
Estado Civil:	Casado
Escolaridad:	Postgrado
Nacionalidad:	Mexicana
Unidad:	Hospital Piloto
Turno:	Matutino
Rol:	medico
Tipo:	Medico Internista
Usuario:	rene
Contraseña:	Contraseña Oculta




Ilustración : Datos del usuario

2. Presionar el botón actualizar usuario, el sistema desplegara la pantalla actualizar usuario.

Actualizar Usuario

Datos	
Nombre:	Rene
Apellido Paterno	Castrejon
Apellido Materno:	Morales
Fecha de Nacimiento:	12/11/1974
Edad:	35
Cedula:	s
Sexo:	<input checked="" type="radio"/> Masculino <input type="radio"/> Femenino
Dirección:	Puente de Vigas
Telefono:	5553949432
Nacionalidad:	Mexicana
Estado Civil:	Casado
Escolaridad:	Postgrado
Unidad:	Hospital Piloto
Turno:	Matutino
Rol:	medico
Tipo	Medico Internista
Usuario	rene
Contraseña Actual	
Contraseña Nueva	
Repita Contraseña Nueva	
<input type="button" value="Actualizar"/>	

Ilustración : Actualizar datos del usuario

3. Insertar los datos nuevos del usuario puede cambiar todos los registros o dejar los que ya se tienen y modificar algunos.
4. Ingresar la contraseña actual, la que se utilizo para autenticarse.
5. Ingresar la contraseña nueva, la que se utilizara de ahora en adelante, si no requiere cambiar contraseña vuelva a escribir su contraseña actual.
6. Ingresar la misma contraseña ingresada en 5.
7. Presionar el botón actualizar, si la contraseña ingresada en 4 es diferente a la contraseña con la que se autentifico no podrá realizar la actualización de los datos, si las contraseñas ingresadas en 5 y 6 son diferentes no podrá realizar la actualización de los datos, en caso contrario, el sistema hace la actualización de sus datos, y desplegara la pantalla principal.

