

# Pagina Hospital

08.10.2020

---

Carlos Benjamín Pac Flores

Centro Universitario de Occidente (CUNOC)

Ingeniería en Ciencias y Sistemas

Octubre 2020

<b>Introducción</b>	<b>3</b>
<b>Plataforma de Desarrollo:</b>	<b>4</b>
<b>Alcances Técnicos:</b>	<b>4</b>
<b>Requisitos De Software:</b>	<b>4</b>
<b>Base de Datos:</b>	<b>5</b>
Diagrama ER:	5
Mapeo Físico:	6
Código SQL de la generación de la base de datos:	6
<b>Archivo de carga del Sistema:</b>	<b>12</b>
Administrador:	12
Doctor:	13
Laboratorista:	14
Paciente:	15
Examen:	16
Informe:	16
Resultado:	17
Cita:	18
Consulta:	18
<b>Detalle del Sistema:</b>	<b>19</b>
<b>Diagrama de clases del proyecto:</b>	<b>19</b>
<b>Detalles de Clases:</b>	<b>20</b>
ConversionesVariables:	20
ConvercionesVariables:	20
DBManage:	21
ConnectionDB:	21
ConsultasDB:	22
DuplicidadDB:	29
ModificacionDB:	31
RegistroDB:	32
ExcepcionesDePrograma:	35
Atributos incompletos:	35

FormatoArchivoIncorrecto:	35
RegistroDuplicado:	36
LecturaXML:	36
Objetos	36
Admin:	36
Archivo:	37
Cita:	38
Consulta:	40
Doctor:	41
Examen:	43
Hospital:	45
HospitalHandler:	48
Laboratorista:	50
Paciente:	52
Persona:	55
Reporte:	56
Resultado:	58
SolicitudExamen:	61
Trabajador:	62
usuarioSistema:	63
Paths	64
obtenerNombreArchivo	64
QueryObjets	65
QueryCita:	65
QueryPaciente:	66
Servlets	67
VerificarContenido	74
VerificarContenido:	74

## Introducción

En el presente documento, encontrará un guía a nivel técnico del funcionamiento de la administración de personal y las acciones que pueden ser realizadas por cada una de las entidades que pueden pertenecer al programa, adjuntando la LÓGICA del programa para guia si se decide en un futuro cercano escalar el proyecto, o el mantenimiento del mismo durante su funcionamiento.

Se utilizará simplificaciones con fines de entendimiento y no sobrecargar la explicación de la lógica.



## **Plataforma de Desarrollo:**

- Lenguaje Java en su versión 11 LTS ORACLE.
- Lenguaje Java en su distribución OPEN-JDK
- Sistema de gestión de proyecto Apache Maven 3.6.3.
- Máquina Virtual (build 11.0.7+8-LTS) Windows.
- Máquina Virtual () Linux Ubuntu y derivados.
- Apache Tomcat 9.0.38

## **Alcances Técnicos:**

El alcance planteado para el programa es la solución al momento de realizar citas medicas así también llevar un control del mismo, dotando a sus usuario de facilidad para llevar las acciones y proveer de comodidad, proveer de información importante al usuario para la toma de decisiones y a la empresa poder agilizar de mejor manera su desenvolvimiento en el aspecto de salud regional.

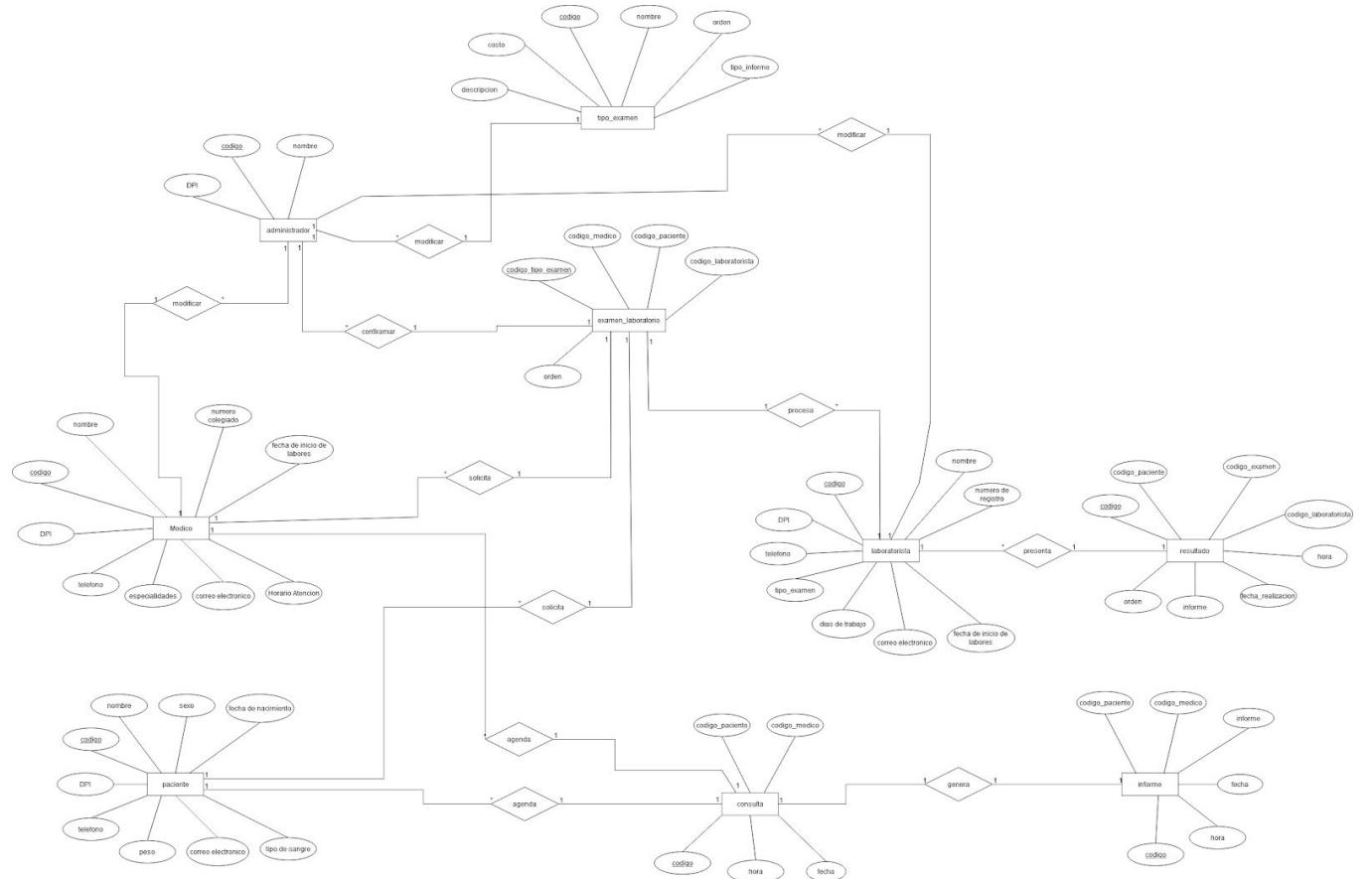
## **Requisitos De Software:**

- Windows 10 ó Linux en sus versiones de 64 bits.
- El requisito de ram es dependiente del crecimiento del programa.
- Los navegadores compatibles son Chrome, Mozilla, Safari.
- Base de Datos Mysql.



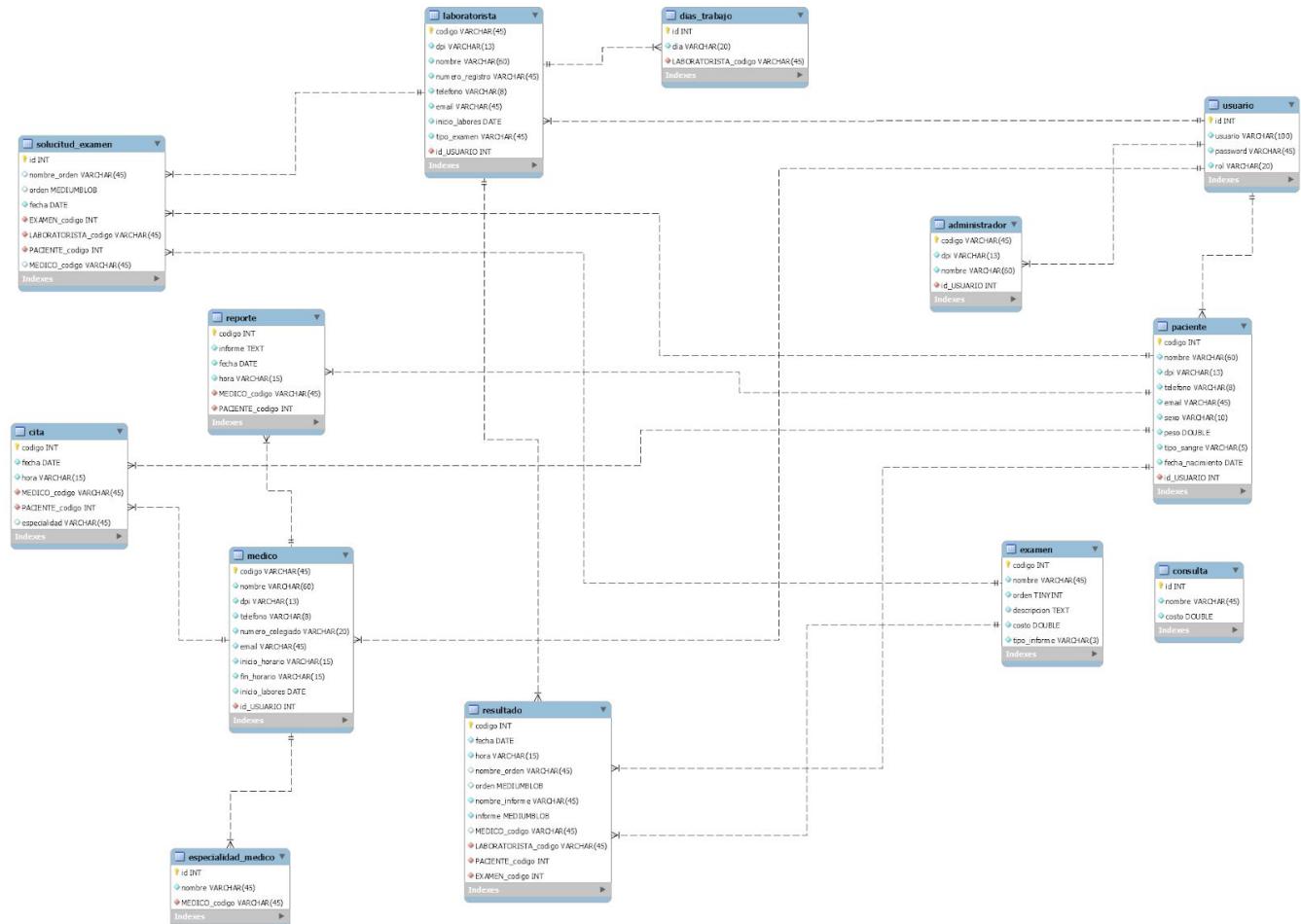
## Base de Datos:

### Diagrama ER:



(Con este documento también se pone a disposición la imagen en formato PNG, para su mejor visualización).

## Mapeo Físico:



(Con este documento también se pone a disposición la imagen en formato PNG, para su mejor visualización).

## Código SQL de la generación de la base de datos:

```
DROP DATABASE PROYECTO2;
```

```
CREATE SCHEMA IF NOT EXISTS PROYECTO2;
```

```
USE PROYECTO2;
```

```
CREATE TABLE IF NOT EXISTS USUARIO(  
    id INT NOT NULL AUTO_INCREMENT,  
    usuario VARCHAR(100) NOT NULL,  
    password VARCHAR(45) NOT NULL,  
    rol VARCHAR(20) NOT NULL,  
    PRIMARY KEY (id),  
    UNIQUE (id,usuario)  
);
```

```
CREATE TABLE IF NOT EXISTS ADMINISTRADOR(  
    codigo VARCHAR(45) NOT NULL ,  
    dpi VARCHAR(13) NOT NULL ,  
    nombre VARCHAR(60) NOT NULL ,  
    id_USUARIO INT NOT NULL ,  
    PRIMARY KEY (codigo),  
    FOREIGN KEY (id_USUARIO) REFERENCES USUARIO(id),  
    UNIQUE (codigo,dpi,id_USUARIO)  
);
```

```
CREATE TABLE IF NOT EXISTS CONSULTA(  
    id INT NOT NULL AUTO_INCREMENT ,  
    nombre VARCHAR (45) NOT NULL ,  
    costo DOUBLE NOT NULL ,  
    PRIMARY KEY (id),  
    UNIQUE (id,nombre)  
);
```

```
CREATE TABLE IF NOT EXISTS MEDICO(
    codigo VARCHAR (45) NOT NULL ,
    nombre VARCHAR (60) NOT NULL ,
    dpi VARCHAR (13) NOT NULL ,
    telefono VARCHAR (8) NOT NULL ,
    numero_colegiado VARCHAR (20) NOT NULL ,
    email VARCHAR (45) NOT NULL ,
    inicio_horario VARCHAR(15) NOT NULL ,
    fin_horario VARCHAR(15) NOT NULL,
    inicio_labores DATE NOT NULL,
    id_USUARIO INT NOT NULL ,
    PRIMARY KEY(codigo),
    FOREIGN KEY (id_USUARIO) REFERENCES USUARIO(id),
    UNIQUE (codigo,dpi,numero_colegiado,email,telefono,id_USUARIO)

);

CREATE TABLE IF NOT EXISTS ESPECIALIDAD_MEDICO(
    id INT NOT NULL AUTO_INCREMENT,
    nombre VARCHAR (45) NOT NULL ,
    MEDICO_codigo VARCHAR (45) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (MEDICO_codigo) REFERENCES MEDICO(codigo),
    UNIQUE (id)

);

CREATE TABLE IF NOT EXISTS PACIENTE(
    codigo INT NOT NULL AUTO_INCREMENT ,
    nombre VARCHAR (60) NOT NULL ,
```



```

dpi VARCHAR (13) NOT NULL ,
telefono VARCHAR (8) NOT NULL ,
email VARCHAR (45) NOT NULL ,
sexo VARCHAR (10) NOT NULL ,
peso DOUBLE NOT NULL ,
tipo_sangre VARCHAR (5) NOT NULL ,
fecha_nacimiento DATE NOT NULL ,
id_USUARIO INT NOT NULL ,
PRIMARY KEY (codigo),
FOREIGN KEY (id_USUARIO) REFERENCES USUARIO(id),
UNIQUE (codigo,dpi,email,telefono,id_USUARIO)
);

```

```

CREATE TABLE IF NOT EXISTS LABORATORISTA(
codigo VARCHAR (45) NOT NULL ,
dpi VARCHAR (13) NOT NULL ,
nombre VARCHAR (60) NOT NULL ,
numero_registro VARCHAR (45) NOT NULL,
telefono VARCHAR (8) NOT NULL ,
email VARCHAR (45) NOT NULL ,
inicio_labores DATE NOT NULL ,
tipo_examen VARCHAR(45) NOT NULL,
id_USUARIO INT NOT NULL ,
PRIMARY KEY (codigo),
FOREIGN KEY (id_USUARIO) REFERENCES USUARIO(id),
UNIQUE (codigo,numero_registro,dpi,email,telefono,id_USUARIO)
);

```

```

CREATE TABLE IF NOT EXISTS DIAS_TRABAJO(

```

```
id INT NOT NULL AUTO_INCREMENT ,  
dia VARCHAR (20) NOT NULL ,  
LABORATORISTA_codigo VARCHAR (45) NOT NULL ,  
PRIMARY KEY (id),  
FOREIGN KEY (LABORATORISTA_codigo) REFERENCES LABORATORISTA(codigo),  
UNIQUE (id)  
);
```

```
CREATE TABLE IF NOT EXISTS EXAMEN(  
codigo INT NOT NULL AUTO_INCREMENT ,  
nombre VARCHAR (45) NOT NULL ,  
orden TINYINT NOT NULL ,  
descripcion TEXT NOT NULL ,  
costo DOUBLE NOT NULL ,  
tipo_informe VARCHAR (3) NOT NULL ,  
PRIMARY KEY (codigo),  
UNIQUE (codigo,nombre)  
);
```

```
CREATE TABLE IF NOT EXISTS SOLUCITUD_EXAMEN(  
id INT AUTO_INCREMENT NOT NULL ,  
nombre_orden VARCHAR (45),  
orden MEDIUMBLOB,  
fecha DATE NOT NULL ,  
EXAMEN_codigo INT NOT NULL ,  
LABORATORISTA_codigo VARCHAR (45) NOT NULL ,  
PACIENTE_codigo INT NOT NULL ,  
MEDICO_codigo VARCHAR (45),  
PRIMARY KEY (id),
```

```
FOREIGN KEY (EXAMEN_codigo) REFERENCES EXAMEN(codigo),
FOREIGN KEY (LABORATORISTA_codigo) REFERENCES LABORATORISTA(codigo),
FOREIGN KEY (PACIENTE_codigo) REFERENCES PACIENTE(codigo),
UNIQUE (id)
);
```

```
CREATE TABLE IF NOT EXISTS RESULTADO(
    codigo INT AUTO_INCREMENT NOT NULL ,
    fecha DATE NOT NULL ,
    hora VARCHAR(15) NOT NULL ,
    nombre_orden VARCHAR (45),
    orden MEDIUMBLOB,
    nombre_informe VARCHAR (45) NOT NULL ,
    informe MEDIUMBLOB NOT NULL,
    MEDICO_codigo VARCHAR (45),
    LABORATORISTA_codigo VARCHAR (45) NOT NULL ,
    PACIENTE_codigo INT NOT NULL ,
    EXAMEN_codigo INT NOT NULL ,
    PRIMARY KEY (codigo),
    FOREIGN KEY (LABORATORISTA_codigo) REFERENCES LABORATORISTA(codigo),
    FOREIGN KEY (PACIENTE_codigo) REFERENCES PACIENTE(codigo),
    FOREIGN KEY (EXAMEN_codigo) REFERENCES EXAMEN(codigo),
    UNIQUE (codigo)
);
```

```
CREATE TABLE IF NOT EXISTS CITA(
    codigo INT AUTO_INCREMENT NOT NULL ,
    fecha DATE NOT NULL ,
    hora VARCHAR(15) NOT NULL ,
    MEDICO_codigo VARCHAR (45) NOT NULL ,
```

```
PACIENTE_codigo INT NOT NULL ,  
especialidad VARCHAR (45),  
PRIMARY KEY (codigo),  
FOREIGN KEY (MEDICO_codigo) REFERENCES MEDICO(codigo),  
FOREIGN KEY (PACIENTE_codigo) REFERENCES PACIENTE(codigo),  
UNIQUE (codigo)  
);
```

```
CREATE TABLE IF NOT EXISTS REPORTE(  
codigo INT AUTO_INCREMENT NOT NULL ,  
informe TEXT NOT NULL,  
fecha DATE NOT NULL ,  
hora VARCHAR(15) NOT NULL ,  
MEDICO_codigo VARCHAR (45) NOT NULL ,  
PACIENTE_codigo INT NOT NULL ,  
PRIMARY KEY (codigo),  
FOREIGN KEY (MEDICO_codigo) REFERENCES MEDICO(codigo),  
FOREIGN KEY (PACIENTE_codigo) REFERENCES PACIENTE(codigo),  
UNIQUE (codigo)  
);
```

## Archivo de carga del Sistema:

Al inicio del lanzamiento del programa este requiere que haya un archivo de alimentación del mismo el cual está contenido en un archivo XML con el siguiente formato de información en su interior:

### Administrador:

```
<admin>
  <CODIGO>ADMIN1</CODIGO>
  <DPI>69712516133</DPI>
  <NOMBRE>Diego Domiguez</NOMBRE>
  <PASSWORD>dfDrVH~9a#-7)^w^P9pB</PASSWORD>
</admin>
```

- admin: etiqueta que identifica a los administradores.
- CODIGO: codigo dentro del sistema.
- DPI: Documento Unico de Identificacion Personal.
- NOMBRE: Nombre del administrador.
- PASSWORD: contraseña de acceso del administrador.

## Doctor:

```
<doctor>
  <CODIGO>MED-123</CODIGO>
  <NOMBRE>Alejandra Mendez</NOMBRE>
  <COLEGIADO>3890</COLEGIADO>
  <DPI>72463794587</DPI>
  <TELEFONO>31661250</TELEFONO>
  <ESPECIALIDAD>
    <TITULO>Pediatrica</TITULO>
    <TITULO>Ginecologia</TITULO>
  </ESPECIALIDAD>
  <CORREO>mail123@hospital.com</CORREO>
  <HORARIO>
    <INICIO>6:00</INICIO>
    <FIN>14:00</FIN>
  </HORARIO>
  <TRABAJO>2010-9-2</TRABAJO>
  <PASSWORD>~fc,X,9;</PASSWORD>
</doctor>
```

- doctor: etiqueta para identificar al doctor.
- CODIGO: id del doctor en el sistema.

- NOMBRE: nombre del doctor.
- COLEGIADO: número de colegiado.
- DPI: Documento Personal de Identificación.
- TELEFONO: Teléfono del doctor.
- ESPECIALIDAD: especialidades asociadas al doctor (Modifica solo el Admin).
  - TITULO: Identifica las especialidad del doctor, puede tener más de una.
- CORREO: correo electrónico del doctor.
- HORARIO: Horario de trabajo (Modifica solo el Admin).
  - INICIO: Inicio de la jornada laboral.
  - FIN: Fin de jornada laboral.
- TRABAJO: fecha en la que inició labores en el hospital (Modifica solo el Admin).
- PASSWORD: contraseña de acceso al sistema.

## Laboratorista:

```
<laboratorista>
  <CODIGO>LAB-123</CODIGO>
  <NOMBRE>Diego Mendez</NOMBRE>
  <REGISTRO>SALUD-2646322</REGISTRO>
  <DPI>50562544694</DPI>
  <TELEFONO>74719529</TELEFONO>
  <EXAMEN>X-ray</EXAMEN>
  <CORREO>mail123@hospital.com</CORREO>
  <TRABAJO>
    <DIA>Lunes</DIA>
    <DIA>Miercoles</DIA>
    <DIA>Jueves</DIA>
    <DIA>Sabado</DIA>
  </TRABAJO>
  <TRABAJOF>2010-9-2</TRABAJOF>
  <PASSWORD>7"UAYz~z</PASSWORD>
</laboratorista>
```

- Laboratorista: Etiqueta para identificar al laboratorista.
- CODIGO: Código dentro del sistema.
- NOMBRE: Nombre del laboratorista.
- REGISTRO: Registro ante el Ministerio de Salud.

- 
- DPI: Documento Personal de Identificación.
  - TELEFONO: Telefono del laboratorista
  - EXAMEN: Examen que puede realizar (Modifica solo el Admin).
  - CORREO: Correo del laboratorista.
  - TRABAJO: Etiqueta que encapsula los días de trabajo (Modifica solo el Admin).
    - DIA: Indica el día de trabajo.
  - TRABAJO: fecha en la que inició labores en el hospital (Modifica solo el Admin).
  - PASSWORD: contraseña de acceso al sistema.

## Paciente:

```
<paciente>
  <CODIGO>177840</CODIGO>
  <NOMBRE>Mario Diaz</NOMBRE>
  <SEXO>Hombre</SEXO>
  <BIRTH>1990-1-1</BIRTH>
  <DPI>89505278065</DPI>
  <TELEFONO>50422898</TELEFONO>
  <PESO>70</PESO>
  <SANGRE>A-</SANGRE>
  <CORREO>mail2848@othermail.com</CORREO>
  <PASSWORD>K^4U;q>=nx5WE954</PASSWORD>
</paciente>
```

- Paciente: etiqueta que identifica al paciente
- CODIGO: Código dentro del sistema.
- NOMBRE: nombre.
- SEXO: Puede ser hombre o mujer.
- BIRTH: Fecha de nacimiento.
- DPI: Documento Personal de Identificación.
- TELEFONO: telefono.
- PESO: Peso en Kg del paciente.
- SANGRE: Tipo de sangre.
- CORREO: Correo electrónico.
- PASSWORD: contraseña de acceso al sistema.



## Examen:

```
<examen>
  <CODIGO>234</CODIGO>
  <NOMBRE>Tomografias</NOMBRE>
  <ORDEN>TRUE</ORDEN>
  <DESCRIPCION>Esta puede ser una descripcion bastante grande</DESCRIPCION>
  <COSTO>2500</COSTO>
  <INFORME>IMG</INFORME>
</examen>
```

- Examen: etiqueta que identifica un examen.
- CODIGO: codigo en el sistema.
- NOMBRE: nombre del examen.
- ORDEN: Indica si requiere orden.
- DESCRIPCIÓN: La descripción del examen.
- COSTO: Costo del examen.
- INFORME: Indica si el archivo a almacenar es IMG (imagen) o PDF.

## Informe:

```
<reporte>
  <CODIGO>594740</CODIGO>
  <PACIENTE>977693</PACIENTE>
  <MEDICO>MED-684</MEDICO>
  <INFORME>Lorem ipsum dolor sit amet.</INFORME>
  <FECHA>2020-1-1</FECHA>
  <HORA>7:00</HORA>
</reporte>
```

- Reporte: etiqueta que identifica un reporte.
- CODIGO: codigo en el sistema.
- PACIENTE: Codigo que identifica al paciente.
- MEDICO: Codigo que identifica al medico.
- INFORME: Contiene el informe redactado por el médico este puede ser extenso.
- FECHA: Fecha en la que se realizó la consulta.
- HORA: Hora de la cita, en formato 24:00 hrs

## Resultado:

```
<resultado>
  <CODIGO>5748348</CODIGO>
  <PACIENTE>977693</PACIENTE>
  <MEDICO>MED-684</MEDICO>
  <EXAMEN>123</EXAMEN>
  <LABORATORISTA>LAB-123</LABORATORISTA>
  <ORDEN>ordenA.png</ORDEN>
  <INFORME>archivoA.png</INFORME>
  <FECHA>2020-9-6</FECHA>
  <HORA>8:00</HORA>
</resultado>
```

- Resultado: Etiqueta que identifica un resultado de examen de laboratorio.
- CODIGO: Codigo dentro del sistema.
- PACIENTE: ID del paciente al que se le practicó el examen.

- 
- MEDICO: Codigo que identifica al medico (Opcional).
  - EXAMEN: Codigo del examen que se realizó.
  - LABORATORISTA: Codigo del laboratorista.
  - ORDEN: De requerirse una orden, debe agregarse el nombre para ser cargado.
  - INFORME: Indica el nombre del archivo, ubicación junto al archivo XML.
  - FECHA: Fecha en la que se realizó el examen.
  - HORA: Hora de la cita, en formato 24:00 hrs.

## Cita:

```
<cita>
  <CODIGO>866691</CODIGO>
  <PACIENTE>177840</PACIENTE>
  <MEDICO>MED-159</MEDICO>
  <ESPECIALIDAD>Pediatrica</ESPECIALIDAD>
  <FECHA>2020-09-02</FECHA>
  <HORA>7:00</HORA>
</cita>
```

- Cita: Etiqueta que identifica una cita para una cita.
- CODIGO: Codigo dentro del sistema.
- PACIENTE: Indica el codigo del paciente.
- Especialidad: Identifica la especialidad del doctor.
- MEDICO: Codigo que identifica al medico.
- FECHA: Fecha programada para la consulta.
- HORA: Hora establecida para la consulta.

## Consulta:

```
<consulta>
  <TIPO>Ginecologia</TIPO>
  <COSTO>150</COSTO>
</consulta>
```

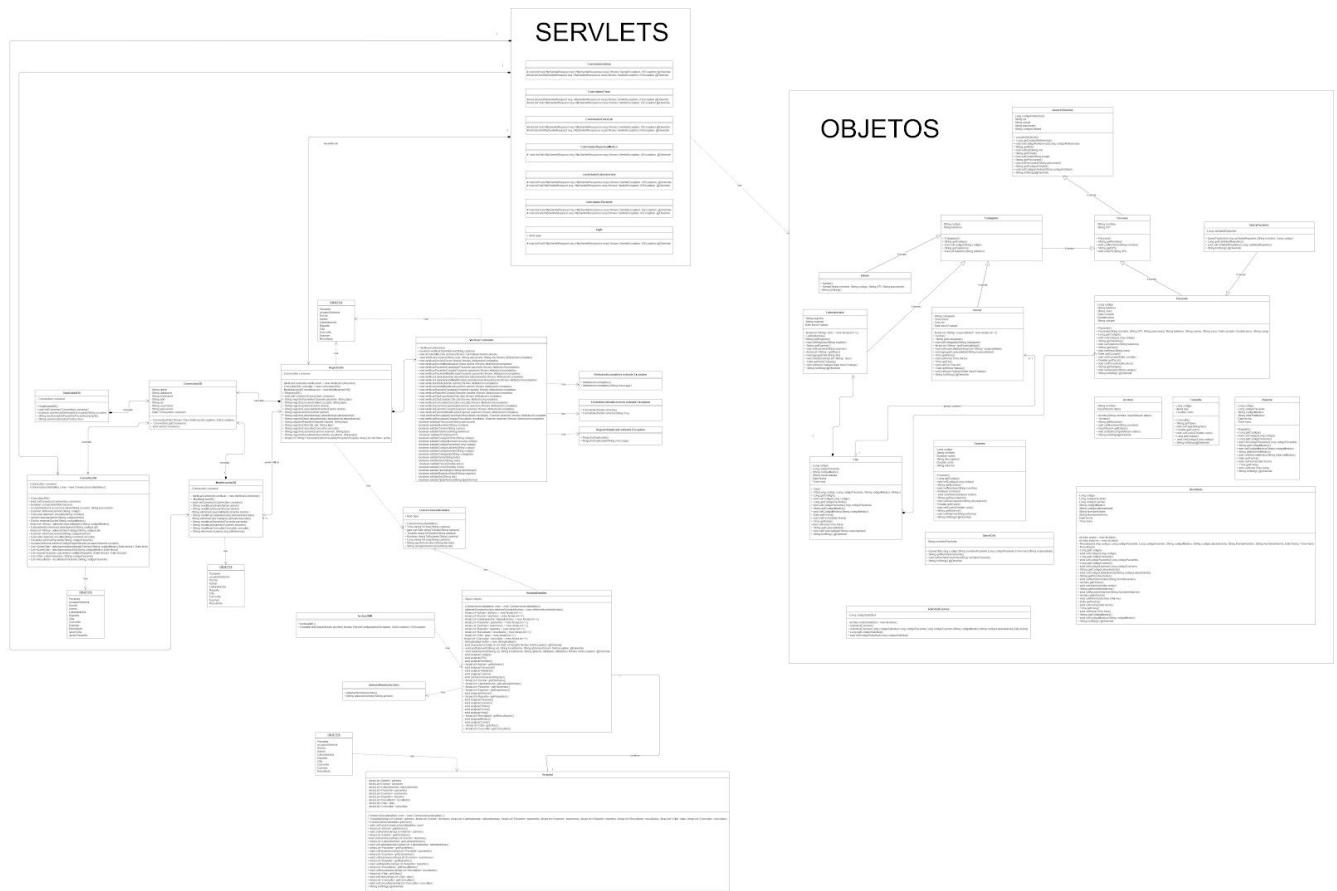
- Consulta: Etiqueta que indica que en este elemento se almacenan los tipos de consultas con su precio.

- TIPO: Nombre del tipo de consulta
- COSTO: Precio el cual se cobrará por cada consulta.

## Detalle del Sistema:

El programa está desarrollado bajo el paradigma POO, y dividido en clases para el trabajo conjunto de la aplicación del programa.

## Diagrama de clases del proyecto:



(Con este documento también se pone a disposición la imagen en formato PNG, para su mejor visualización).

## Detalles de Clases:

A continuación se presenta un detalle de las clases incluidas en el programa a nivel de código de JAVA:

### ConversionesVariables:

#### ConvercionesVariables:

**Method Summary**

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
java.lang.String	ajusteFormatos(java.lang.String formato)	AJUSTA EL NOMBRE DEL FORMATO PARA EL GUARDADO EN LA BASE DE DATOS CON UN ESTANDAR
java.lang.String	arregloDiaSemana(java.lang.String dia)	Introduce el dia en string y lo devuelve de una forma corregia lunes = Lunes o LuNes=Lunes
java.lang.Boolean	stringToBoolean(java.lang.String cadena)	Parcea una cadena String a boolean Si la cadena es valida devolvera un valor diferende de NULL
java.sql.Date	stringToDate(java.lang.String cadena)	Parcea una cadena String a Date con formato yyyy-MM-dd Si la cadena es valida devolvera un valor diferende de NULL
java.lang.Double	stringToDouble(java.lang.String cadena)	Parcea una cadena String a double Si la cadena es valida devolvera un valor diferende de NULL
java.lang.Long	stringToLong(java.lang.String cadena)	Parcea una cadena String a long Si la cadena es valida devolvera un valor diferende de NULL
java.sql.Time	stringToTime(java.lang.String cadena)	Parcea una cadena String a Time con formato HH:mm Si la cadena es valida devolvera un valor diferente de NULL

**Constructor Detail**

**ConversionesVariables**

```
public ConversionesVariables()
```

**Method Detail**

**stringToTime**

```
public java.sql.Time stringToTime(java.lang.String cadena)
Parcea una cadena String a Time con formato HH:mm Si la cadena es valida devolvera un valor diferente de NULL
Parameters:
cadena -
Returns:
```

**stringToDate**

```
public java.sql.Date stringToDate(java.lang.String cadena)
Parcea una cadena String a Date con formato yyyy-MM-dd Si la cadena es valida devolvera un valor diferende de NULL
Parameters:
cadena -
Returns:
```

**stringToDouble**

```
public java.lang.Double stringToDouble(java.lang.String cadena)
```

Parcea una cadena String a double Si la cadena es valida devolvera un valor differente de NULL

Parameters:

cadena -

Returns:

**stringToBoolean**

```
public java.lang.Boolean stringToBoolean(java.lang.String cadena)
```

Parcea una cadena String a boolean Si la cadena es valida devolvera un valor differente de NULL

Parameters:

cadena -

Returns:

**stringToLong**

```
public java.lang.Long stringToLong(java.lang.String cadena)
```

Parcea una cadena String a long Si la cadena es valida devolvera un valor differente de NULL

Parameters:

cadena -

Returns:

**ajusteFormatos**

```
public java.lang.String ajusteFormatos(java.lang.String formato)
```

AJUSTA EL NOMBRE DEL FORMATO PARA EL GUARDADO EN LA BASE DE DATOS CON UN ESTANDAR

Parameters:

formato -

Returns:

**arregloDiaSemana**

```
public java.lang.String arregloDiaSemana(java.lang.String dia)
```

Introduce el dia en string y lo devuelve de una forma corregia lunes = Lunes o LuNes=Lunes

Parameters:

dia -

Returns:

## DBManage:

### ConnectionDB:

#### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<a href="#">cerrarConexion()</a>	Cierra la conexion con la base de datos del programa
java.sql.Connection	<a href="#">getConexion()</a>	Retorna la conexion de la base de datos

**Constructor Detail****ConnectionDB**

```
public ConnectionDB()  
    throws java.lang.ClassNotFoundException,  
          java.sql.SQLException  
  
Constructor principal de la clase conexion Genera la conexion de la base de datos al principio  
Throws:  
java.lang.ClassNotFoundException  
java.sql.SQLException
```

**Method Detail****getConexion**

```
public java.sql.Connection getConexion()  
  
Retorna la conexion de la base de datos  
Returns:
```

**cerrarConexion**

```
public void cerrarConexion()  
  
Cierra la conexion con la base de datos del programa  
Throws:  
java.sql.SQLException
```

**ConsultasDB:**

**Method Summary**

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
usuarioSistema	accesoUsuario(java.lang.String usuario, java.lang.String password)	DEVULEVE EL ROL DEL USUARIO Y CONTRASENA INTRODUCIDOS	
java.util.List<java.lang.String[]>	CantidadExamenesRealizadosEnIntervalo (java.lang.String codigoPaciente, java.sql.Date fech1, java.sql.Date fech2)		
java.util.List<java.lang.String[]>	cantidadResultadosHechos(java.lang.String codigoLab)	LAS 10 FECHAS CON MAS TRABAJO	
java.util.List<java.lang.String[]>	cincoMedicosMenorCitas()	LOS 5 MEDICOS CON MENOR CANTIDAD DE CITAS	
java.util.List<QueryCita>	citasAgendadasDia(java.lang.String codigoMedico, java.sql.Date fecha)	RETORNA LAS CITAS SEGUN LA FECHA Y EL CODIGO DEL MEDICO QUIEN REALIZA LA CONSULTA	
java.util.List<QueryCita>	citasAgendadasIntervaloTiempo(java.lang.String codigoMedico, java.sql.Date fech1, java.sql.Date fech2)	RETORNA LAS CITAS EN UN INTERVALO DE TIEMPO DETERMINADO SEGUN EL MEDICO QUE REALIZA LA CONSULTA	
java.util.List<Cita>	citasPacientes(java.lang.String codigoPaciente)	RETORNA LAS CITAS DE UN PACIENTE EN ESPECIFICO SEGUN SU CODIGO DE IDENTIFICACION EN LA BASE DE DATOS	
DATOS			
boolean	comprobarInformacion()		
java.util.List<java.lang.String[]>	consultasConMedico(java.lang.String codigoPaciente, java.sql.Date fech1, java.sql.Date fech2)	CONSULTAS REALIZADAS CON UN MEDICO EN ESPECIFICO EN INTERVALO DE TIEMPO	
java.util.List<java.lang.String[]>	diezMedicosMasInforme()	RETORNA LOS MEDICOS CON MAS INFROMES EN EL HOSPITAL	
java.util.List<java.lang.String[]>	examenesMasDemandadosEnTiempo(java.sql.Date fech1, java.sql.Date fech2)	RETORNA LOS EXAMENES MAS DEMANDADOS EN UN INTERVALO DE TIEMPO	
java.util.List<java.lang.String[]>	medicosQueDemandanMasExamenes()	LOS MEDICOS QUE DEMANDAN MAS EXAMENES	
Consulta	obtenerConsulta(java.lang.String nombre)	Obtiene una consulta segun nombre de la consulta	
java.util.List<java.lang.String[]>	obtenerDoctores(java.lang.String nombre, java.lang.String especialidad, java.lang.String hora)		
java.util.ArrayList<java.lang.String>	obtenerEspecialidades(java.lang.String codigoMedico)	OBTIENE LAS ESPECIALIDADES DE UN MEDICO SEGUN SU CODIGO DE IDENTIFICACION	
Examen	obtenerExamen(java.lang.String codigo)	Obtiene un examen segun codigo del mismo examen	
java.util.List<Reporte>	obtenerReportePaciente(java.lang.String codigoPaciente)	REPORTES DE PACIENTE	
java.util.List<QueryPaciente>	pacientesConMasReportes(java.sql.Date fech1, java.sql.Date fech2)	RETORNA LOS PACIENTES CON MAYOR CANTIDAD DE REPORTES MEDICOS EN UN INTERVALO DE TIEMPO	
EXAMENES			
java.util.List<Resultado>	resultadosHechos(java.lang.String codigoLab, java.sql.Date fecha)	EXAMENES PROCESADOS SEGUN EL DIA Y LABORATORISTA DE ENTRADA	
java.util.List<Resultado>	resultadosPaciente(java.lang.String codigoPaciente)	RETORNO DE LOS RESULTADOS DE UN PACIENTE EN ESPECIFICO EN EL SISTEMA	
Consulta	retornaConsulta(java.lang.String nombreConsulta)	RETORNA LA CONSULTA SEGUN EL NOMBRE ASIGNADO	
Admin	retornarAdmin(java.lang.String codigoAdmin)	REALIZA LA BUSQUEDA DE UN ADMINISTRADOR SEGUN CODIGO DE IDENTIFICACION	
Cita	retornarCita(java.lang.String codigoCita)		
java.util.ArrayList<java.lang.String>	retornarCódigosSegunExamenLab(java.lang.String nombreExamen)	RETORNNA LOS CODIGOS DE LOS LABORATORISTA SEGUN EL EXAMEN QUE REALIZA	
Doctor	retornarDoctor(java.lang.String codigoMedico)	OBTINENE EL DOCTOR RESULTANTE DEPENDIENDO DEL CODIGO INTRODUCIDO	
Examen	retornarExamen(java.lang.String codigoExamen)	RETORNA UN EXAMEN SEGUN EL CODIGO DE INGRESO	
Laboratorista	retornarLaboratorista(java.lang.String codigoLab)		
Paciente	retornarPaciente(java.lang.String codigoPaciente)	RETORNA EL PACIENTE SEGUN EL CODIGO DEL PACIENTE	
java.util.List<Examen>	retornarTodosExamen()	RETORNA TODOS LOS EXAMENES QUE HAY EN EL HOSPITAL	
usuarioSistema	retornoCodigoDependiente(usuarioSistema usuario)	Retorna el codigo dependiente de usuario segun su	

usuarioSistema	retornoCodigoDependiente(usuarioSistema usuario)	Retorna el codigo dependiente de usuario segun su usuarioSistema
Reporte	returnReporte(java.lang.String codigoReporte)	RETORNA UN REPORTE CON UN CODIGO ESPECIFICO
void	setConexion(java.sql.Connection conexion)	
java.util.List<SolicitudExamen>	solicitudesExamen(java.lang.String codigoLab, java.sql.Date fecha)	RECUPERA LA SOLICITUDES PARA PROCESAR
SolicitudExamen	SolicitudExamen(java.lang.Long codigoSolicitud)	RETORNA LA SOLICITUD DE EXAMEN CON EL CODIGO DADO
java.util.List<java.lang.String[]>	TRESEXAMENESMasDemandadosEnTiempo(java.sql.Date fech1, java.sql.Date fech2)	LOS TRES EXAMENES MAS DEMANDADOS EN UN INTERVALO DE TIEMPO
java.util.List<Cita>	ultimas5CitasPacientes(java.lang.String codigoPaciente)	RETORNA LAS CITAS DE UN PACIENTE EN ESPECIFICO SEGUN SU CODIGO DE IDENTIFICACION EN LA BASE DE DATOS
java.util.List<Resultado>	ultimos5ResultadosPaciente(java.lang.String codigoPaciente)	RETORNO DE LOS RESULTADOS DE UN PACIENTE EN ESPECIFICO EN EL SISTEMA

**Constructor Detail****ConsultasDB**

```
public ConsultasDB()
```

**Method Detail****setConexion**

```
public void setConexion(java.sql.Connection conexion)
```

**comprobarInformacion**

```
public boolean comprobarInformacion()
```

**accesoUsuario**

```
public usuarioSistema accesoUsuario(java.lang.String usuario, java.lang.String password)
```

DEVULEVE EL ROL DEL USUARIO Y CONTRASENA INTRODUCIDOS

Parameters:

usuario -

password -

Returns:

Parameters:

usuario -

password -

Returns:

**obtenerExamen**

```
public Examen obtenerExamen(java.lang.String codigo)
```

Obtiene un examen segun codigo del mismo examen

Parameters:

codigo -

Returns:

**obtenerConsulta**

**obtenerConsulta**

```
public Consulta obtenerConsulta(java.lang.String nombre)
```

Obtiene una consulta segun nombre de la consulta

Parameters:

nombre -

Returns:

**retornarAdmin**

```
public Admin retornarAdmin(java.lang.String codigoAdmin)
```

REALIZA LA BUSQUEDA DE UN ADMINISTRADOR SEGUN CODIGO DE IDENTIFICACION

Parameters:

codigoAdmin -

Returns:

**retornarDoctor**

```
public Doctor retornarDoctor(java.lang.String codigoMedico)
```

OBTINENE EL DOCTOR RESULTANTE DEPENDIENDO DEL CODIGO INTRODUCIDO

OBTINENE EL DOCTOR RESULTANTE DEPENDIENDO DEL CODIGO INTRODUCIDO

Parameters:

codigoMedico -

Returns:

**obtenerEspecialidades**

```
public java.util.ArrayList<java.lang.String> obtenerEspecialidades(java.lang.String codigoMedico)
```

OBTIENE LAS ESPECIALIDADES DE UN MEDICO SEGUN SU CODIGO DE IDENTIFICACION

Parameters:

codigoMedico -

Returns:

**retornarLaboratorista**

```
public Laboratorista retornarLaboratorista(java.lang.String codigoLab)
```

**retornarCódigosSegunExamenLab**

```
public java.util.ArrayList<java.lang.String> retornarCódigosSegunExamenLab(java.lang.String nombreExamen)
```

RETORNA LOS CODIGOS DE LOS LABORATORISTA SEGUN EL EXAMEN QUE REALIZA

Parameters:

nombreExamen -

Returns:

**retornarCita**

```
public Cita retornarCita(java.lang.String codigoCita)
```

Parameters:

codigoCita -

**retornarExamen**

```
public Examen retornarExamen(java.lang.String codigoExamen)
```

RETORNA UN EXAMEN SEGUN EL CODIGO DE INGRESO

RETORNA UN EXAMEN SEGUN EL CODIGO DE INGRESO

Parameters:

codigoExamen -

Returns:

#### retornarTodosExamens

```
public java.util.List<Examen> retornarTodosExamens()
```

RETORNA TODOS LOS EXAMENES QUE HAY EN EL HOSPITAL

Parameters:

codigoExamen -

Returns:

#### retornaConsulta

```
public Consulta retornaConsulta(java.lang.String nombreConsulta)
```

RETORNA LA CONSULTA SEGUN EL NOMBRE ASIGNADO

Parameters:

nombreConsulta -

nombreConsulta -

Returns:

#### retornarPaciente

```
public Paciente retornarPaciente(java.lang.String codigoPaciente)
```

RETORNA EL PACIENTE SEGUN EL CODIGO DEL PACIENTE

Parameters:

codigoPaciente -

Returns:

#### retornoCodigoDependiente

```
public usuarioSistema retornoCodigoDependiente(usuarioSistema usuario)
```

Retorna el codigo dependiente de usuario segun su usuarioSistema

Returns:

#### citasAgendadasIntevaloTiempo

##### citasAgendadasIntevaloTiempo

```
public java.util.List<QueryCita> citasAgendadasIntevaloTiempo(java.lang.String codigoMedico, java.sql.Date fech1, java.sql.Date fech2)
```

RETORNA LAS CITAS EN UN INTERVALO DE TIEMPO DETERMINADO SEGUN EL MEDICO QUE REALIZA LA CONSULTA

Parameters:

codigoMedico -

fech1 -

fech2 -

Returns:

#### citasAgendadasDia

```
public java.util.List<QueryCita> citasAgendadasDia(java.lang.String codigoMedico, java.sql.Date fecha)
```

RETORNA LAS CITAS SEGUN LA FECHA Y EL CODIGO DEL MEDICO QUIEN REALIZA LA CONSULTA

Parameters:

codigoMedico -

fecha -

Returns:

**pacientesConMasReportes**

```
public java.util.List<QueryPaciente> pacientesConMasReportes(java.sql.Date fechal, java.sql.Date fecha2)
```

RETORNA LOS PACIENTES CON MAYOR CANTIDAD DE REPORTES MEDICOS EN UN INTERVALO DE TIEMPO

Parameters:

fechal -

fecha2 -

Returns:

**obtenerReportePaciente**

```
public java.util.List<Reporte> obtenerReportePaciente(java.lang.String codigoPaciente)
```

REPORTES DE PACIENTE

Parameters:

codigoPaciente -

Returns:

**returnReporte**

```
public Reporte returnReporte(java.lang.String codigoReporte)
```

RETORNA UN REPORTE CON UN CODIGO ESPECIFICO

Parameters:

codigoReporte -

Returns:

**citasPacientes**

```
public java.util.List<Cita> citasPacientes(java.lang.String codigoPaciente)
```

RETORNA LAS CITAS DE UN PACIENTE EN ESPECIFICO SEGUN SU CODIGO DE IDENTIFICACION EN LA BASE DE DATOS

Parameters:

codigoPaciente -

Returns:

**resultadosPaciente****resultadosPaciente**

```
public java.util.List<Resultado> resultadosPaciente(java.lang.String codigoPaciente)
```

RETORNO DE LOS RESULTADOS DE UN PACIENTE EN ESPECIFICO EN EL SISTEMA

Parameters:

codigoPaciente -

Returns:

**solicitudesExamen**

```
public java.util.List<SolicitudExamen> solicitudesExamen(java.lang.String codigoLab, java.sql.Date fecha)
```

RECUPERA LA SOLICITUDES PARA PROCESAR

Parameters:

codigoLab -

Returns:

**SolicitudExamen**

```
public SolicitudExamen SolicitudExamen(java.lang.Long codigoSolicitud)
```

RETORNA LA SOLICITUD DE EXAMEN CON EL CODIGO DADO

RETORNA LA SOLICITUD DE EXAMEN CON EL CODIGO DADO

Parameters:

codigoSolicitud -

Returns:

#### resultadosHechos

```
public java.util.List<Resultado> resultadosHechos(java.lang.String codigoLab, java.sql.Date fecha)
```

EXAMENES PROCESADOS SEGUN EL DIA Y LABORATORISTA DE ENTRADA

Parameters:

codigoLab -

fecha -

Returns:

#### cantidadResultadosHechos

```
public java.util.List<java.lang.String[]> cantidadResultadosHechos(java.lang.String codigoLab)
```

LAS 10 FECHAS CON MAS TRABAJO

LAS 10 FECHAS CON MAS TRABAJO

Parameters:

codigoLab -

Returns:

#### CantidadExamenesRealizadosEnIntervalo

```
public java.util.List<java.lang.String[]> CantidadExamenesRealizadosEnIntervalo(java.lang.String codigoPaciente, java.sql.Date fechal, java.sql.Date fecha2)
```

Parameters:

codigoPaciente -

fechal -

fecha2 -

Returns:

#### ultimos5ResultadosPaciente

```
public java.util.List<Resultado> ultimos5ResultadosPaciente(java.lang.String codigoPaciente)
```

RETORNO DE LOS RESULTADOS DE UN PACIENTE EN ESPECIFICO EN EL SISTEMA

Parameters:

#### ultimos5ResultadosPaciente

```
public java.util.List<Resultado> ultimos5ResultadosPaciente(java.lang.String codigoPaciente)
```

RETORNO DE LOS RESULTADOS DE UN PACIENTE EN ESPECIFICO EN EL SISTEMA

Parameters:

codigoPaciente -

Returns:

#### ultimas5CitasPacientes

```
public java.util.List<Cita> ultimas5CitasPacientes(java.lang.String codigoPaciente)
```

RETORNA LAS CITAS DE UN PACIENTE EN ESPECIFICO SEGUN SU CODIGO DE IDENTIFICACION EN LA BASE DE DATOS

Parameters:

codigoPaciente -

Returns:

#### consultasConMedico

```
public java.util.List<java.lang.String[]> consultasConMedico(java.lang.String codigoPaciente, java.sql.Date fechal, java.sql.Date fecha2)
```

```
public java.util.List<java.lang.String[]> consultasConMedico(java.lang.String codigoPaciente, java.sql.Date fechal, java.sql.Date fecha2)
```

CONSULTAS REALIZADAS CON UN MEDICO EN ESPECIFICO EN INTERVALO DE TIEMPO

Parameters:

codigoPaciente -

Returns:

#### diezMedicosMasInforme

```
public java.util.List<java.lang.String[]> diezMedicosMasInforme()
```

RETORNA LOS MEDICOS CON MAS INFROMES EN EL HOSPITAL

Returns:

#### cincoMedicosMenorCitas

```
public java.util.List<java.lang.String[]> cincoMedicosMenorCitas()
```

LOS 5 MEDICOS CON MENOR CANTIDAD DE CITAS

#### medicosQueDemandanMasExamenes

##### medicosQueDemandanMasExamenes

```
public java.util.List<java.lang.String[]> medicosQueDemandanMasExamenes()
```

LOS EMDICOS QUE DEMANDAN MAS EXAMENES

Returns:

#### examenesMasDemandadosEnTiempo

```
public java.util.List<java.lang.String[]> examenesMasDemandadosEnTiempo(java.sql.Date fechal, java.sql.Date fecha2)
```

RETORNA LOS EXAMENES MAS DEMANDADOS EN UN INTERVALO DE TIEMPO

Parameters:

fechal -

fecha2 -

Returns:

#### TREsexamenesMasDemandadosEnTiempo

```
public java.util.List<java.lang.String[]> TREsexamenesMasDemandadosEnTiempo(java.sql.Date fechal, java.sql.Date fecha2)
```

LOS TRES EXAMENES MAS DEMANDADOS ES UN INTERVALO DE TIEMPO

#### TRESexamenesMasDemandadosEnTiempo

```
public java.util.List<java.lang.String[]> TRESexamenesMasDemandadosEnTiempo(java.sql.Date fechal, java.sql.Date fecha2)
```

LOS TRES EXAMENES MAS DEMANDADOS ES UN INTERVALO DE TIEMPO

Parameters:

fechal -

fecha2 -

Returns:

#### obtenerDoctores

```
public java.util.List<java.lang.String[]> obtenerDoctores(java.lang.String nombre, java.lang.String especialidad, java.lang.String hora)
```

DuplicidadDB:

**Method Summary**

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
	java.lang.String <b>existenciaDePaciente</b> (Paciente paciente)	VERIFICA SI EXISTE UN YA EXISTE UN USUARIO EN LA BASE DE DATOS ,true = si existe, false = no existe
boolean	existenciaDeRegistroUsuario (java.lang.String usuario)	VERIFICA SI EXISTE UN YA EXISTE UN USUARIO EN LA BASE DE DATOS ,true = si existe, false = no existe
	java.lang.String <b>existenciaDoctor</b> (Doctor doc)	
void	setConexion(java.sql.Connection conexion)	Asigna la conexion

**Constructor Detail****DuplicidadDB**

```
public DuplicidadDB()
CONSTRUCTOR VACIO
```

**Method Detail****setConexion**

```
public void setConexion(java.sql.Connection conexion)
Asgina la conexion
Parameters:
conexion -
```

**existenciaDeRegistroUsuario**

```
public boolean existenciaDeRegistroUsuario(java.lang.String usuario)
VERIFICA SI EXISTE UN YA EXISTE UN USUARIO EN LA BASE DE DATOS ,true = si existe, false = no existe
Parameters:
usuario -
Returns:
```

**existenciaDePaciente**

```
public java.lang.String existenciaDePaciente(Paciente paciente)
VERIFICA SI EXISTE UN YA EXISTE UN USUARIO EN LA BASE DE DATOS ,true = si existe, false = no existe
Parameters:
codigo -
dpi -
email -
telefono -
Returns:
```

**existenciaDoctor**

```
public java.lang.String existenciaDoctor(Doctor doc)
```

## ModificacionDB:

### Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
java.lang.String	eliminarSolicitud(java.lang.String codigoSolicitud)	ELIMINA UNA SOLICITUD DESPUES DE PROCESARLA	
java.lang.String	eliminarUsuario(java.lang.Long idReferencia)	ELIMINAR EL USUARIO SEGUN EL ID DE REFERENCIA	
java.lang.String	modificarAdmin(Admin admin)	REALIZA LA MODIFICACION DE LOS DATOS DEL ADMIN	
java.lang.String	modificarConsulta(Consulta consulta)	MODIFICACION DE LA CONSULTA DEPENDIENTO DE LA CONSULTA DE ENTRADA	
java.lang.String	modificarDoctor(Doctor doctor)	REALIZA LA MODIFICACION DEL DOCTOR INGRESADO	
java.lang.String	modificarExamen(Examen examen)	MODIFICA UN EXAMEN SEGUN EL EXAMEN DE ENTRADA	
java.lang.String	modificarLaboratorista(Laboratorista labo)		
java.lang.String	modificarPaciente(Paciente paciente)	MODIFICA LA INFORMACION DEL PACIENTE INGRESADO EN EL SISTEMA	
void	setConexion(java.sql.Connection conexion)		

### Constructor Detail

#### ModificacionDB

```
public ModificacionDB()
```

### Method Detail

#### setConexion

```
public void setConexion(java.sql.Connection conexion)
```

#### modificarAdmin

```
public java.lang.String modificarAdmin(Admin admin)
```

REALIZA LA MODIFICACION DE LOS DATOS DEL ADMIN

Parameters:

admin -

Returns:

#### modificarDoctor

```
public java.lang.String modificarDoctor(Doctor doctor)
```

REALIZA LA MODIFICACION DEL DOCTOR INGRESADO

REALIZA LA MODIFICACION DEL DOCTOR INGRESADO

Parameters:

doctor -

Returns:

#### eliminarSolicitud

```
public java.lang.String eliminarSolicitud(java.lang.String codigoSolicitud)
```

ELIMINA UNA SOLICITUD DESPUES DE PROCESARLA

Parameters:

codigoSolicitud -

Returns:

#### modificarLaboratorista

```
public java.lang.String modificarLaboratorista(Laboratorista labo)
```

```
modificarPaciente

public java.lang.String modificarPaciente(Paciente paciente)
MODIFICA LA INFORMACION DEL PACIENTE INGRESADO EN EL SISTEMA
Parameters:
paciente -
Returns:
```

```
modificarExamen

public java.lang.String modificarExamen(Examen examen)
MODIFICA UN EXAMEN SEGUN EL EXAMEN DE ENTRADA
Parameters:
examen -
Returns:
```

```
modificarConsulta

public java.lang.String modificarConsulta(Consulta consulta)
```

```
modificarConsulta

public java.lang.String modificarConsulta(Consulta consulta)
MODIFICACION DE LA CONSULTA DEPENDIENTO DE LA CONSULTA DE ENTRADA
Parameters:
consulta -
Returns:
```

```
eliminarUsuario

public java.lang.String eliminarUsuario(java.lang.Long idReferencia)
ELIMINAR EL USUARIO SEGUN EL ID DE REFERENCIA
Parameters:
idReferencia -
Returns:
```

## RegistroDB:

### Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method		Description
java.lang.String	registroAdmin(Admin admin)		
java.lang.String	registroCita(Cita cita, java.lang.String tipo)		
java.lang.String	registroConsulta(Consulta consulta)		REGISTRO DEL TIPO DE CONSULTAS QUE HAY EN EL HOSPITAL
java.lang.String	registroDiasLaboratorista(Laboratorista laboratorista)		REGISTRO DE LOS DIAS DE TRABAJO DEL LABORATORISTA
java.lang.String	registroDoctor(Doctor doctor)		REALIZA EL REGISTRO DEL DOCTOR ADJUNTADO
java.lang.String	registroEspecialidadDoctor(Doctor doctor)		REGISTRO DE ESPECIALIDAD DE DOCTOR EN BASE DE DATOS
java.lang.String	registroExamen(Examen examen, java.lang.String tipo)		REGISTRO DEL TIPO DE EXAMENES QUE SUE PUEDEN REALIZAR EN EL HOSPITAL
java.lang.String	registroLaboratorista(Laboratorista laboratorista)		REGISTRO DE LOS LABORATORISTA DEL HOSPITAL
java.lang.String	registroPaciente(Paciente paciente, java.lang.String tipo)		
java.lang.String	registroReporte(Reporte reporte, java.lang.String tipo)		REGISTRO DE REPORTE tipo="exportado" traslado existente tipo="nuevo" de nuevo ingreso
java.lang.String	registroResultado(Resultado resultado, java.lang.String tipo)		REGISTRO DE LOS RESULTADO EMITIDOS POR EL HOSPITAL

java.lang.String	registroResultado(Resultado resultado, java.lang.String tipo)	REGISTRO DE LOS RESULTADO EMITIDOS POR EL HOSPITAL
java.lang.String	registroSolicitudExamen(SolicitudExamen solicitud)	REGISTRO DE UNA SOLICITUD DE EXAMEN EN LA BASE DE DATOS DEL HOSPITAL
java.lang.String	registroUsuario(java.lang.Object usuario, java.lang.String tipo)	REALIZA EL REGISTRO DE USUARIOS PARA EL INGRESO AL SISTEMA
void	setConexion(java.sql.Connection conexion)	
java.util.ArrayList<java.lang.String>	trasladarDatosHospital(Hospital hospital, java.util.ArrayList<Archivo> archivos)	METODO DE INTROUCCION DE DATOS DEL HOSPITAL EN LA BASE DE DATOS

***Constructor Detail*****RegistroDB**

```
public RegistroDB()
```

***Method Detail*****setConexion**

```
public void setConexion(java.sql.Connection conexion)
```

**registroPaciente**

```
public java.lang.String registroPaciente(Paciente paciente, java.lang.String tipo)
```

**registroUsuario**

```
public java.lang.String registroUsuario(java.lang.Object usuario, java.lang.String tipo)
```

REALIZA EL REGISTRO DE USUARIOS PARA EL INGRESO AL SISTEMA

Parameters:

usuario -

tipo -

usuario -

tipo -

Returns:

**registroDoctor**

```
public java.lang.String registroDoctor(Doctor doctor)
```

REALIZA EL REGISTRO DEL DOCTOR ADJUNTADO

Parameters:

doctor -

Returns:

**registroEspecialidadDoctor**

```
public java.lang.String registroEspecialidadDoctor(Doctor doctor)
```

REGISTRO DE ESPECIALIDAD DE DOCTOR EN BASE DE DATOS

Parameters:

doctor -

Returns:

Returns:

#### registroAdmin

```
public java.lang.String registroAdmin(Admin admin)
```

#### registroLaboratorista

```
public java.lang.String registroLaboratorista(Laboratorista laboratorista)
```

REGISTRO DE LOS LABORATORISTA DEL HOSPITAL

Parameters:

laboratorista -

Returns:

#### registroDiasLaboratorista

```
public java.lang.String registroDiasLaboratorista(Laboratorista laboratorista)
```

REGISTRO DE LOS DIAS DE TRABAJO DEL LABORATORISTA

REGISTRO DE LOS DIAS DE TRABAJO DEL LABORATORISTA

Parameters:

laboratorista -

Returns:

#### registroReporte

```
public java.lang.String registroReporte(Reporte reporte, java.lang.String tipo)
```

REGISTRO DE REPORTE tipo="exportado" traslado existente tipo="nuevo" de nuevo ingreso

Parameters:

reporte -

tipo -

Returns:

#### registroCita

```
public java.lang.String registroCita(Cita cita, java.lang.String tipo)
```

#### registroConsulta

```
public java.lang.String registroConsulta(Consulta consulta)
```

REGISTRO DEL TIPO DE CONSULTAS QUE HAY EN EL HOSPITAL

Parameters:

consulta -

Returns:

#### registroExamen

```
public java.lang.String registroExamen(Examen examen, java.lang.String tipo)
```

REGISTRO DEL TIPO DE EXAMENES QUE SUE PUEDEN REALIZAR EN EL HOSPITAL

Parameters:

examen -

tipo -

Returns:

#### registroResultado

```

registroResultado

public java.lang.String registroResultado(Resultado resultado, java.lang.String tipo)
REGISTRO DE LOS RESULTADO EMITIDOS POR EL HOSPITAL
Parameters:
resultado-
tipo-
Returns:

trasladarDatosHospital

public java.util.ArrayList<java.lang.String> trasladarDatosHospital(Hospital hospital, java.util.ArrayList<Archivo> archivos) throws
java.io.FileNotFoundException

METODO DE INTROUCCION DE DATOS DEL HOSPITAL EN LA BASE DE DATOS
Parameters:
hospital -
archivos -
Returns:
Throws:
java.io.FileNotFoundException
java.io.FileNotFoundException

registroSolicitudExamen

public java.lang.String registroSolicitudExamen(SolicitudExamen solicitud)
REGISTRO DE UNA SOLICITUD DE EXAMEN EN LA BASE DE DATOS DEL HOSPITAL
Returns:

```

## ExcepcionesDePrograma:

### Atributos incompletos:

*Constructor Detail*

```

AtributosIncompletos

public AtributosIncompletos()

AtributosIncompletos

public AtributosIncompletos(java.lang.String message)

```

### FormatoArchivoIncorrecto:

*Constructor Detail*

```

FormatoArchivoIncorrecto

public FormatoArchivoIncorrecto()

FormatoArchivoIncorrecto

public FormatoArchivoIncorrecto(java.lang.String msg)

```

## RegistroDuplicado:

### *Constructor Detail*

#### RegistroDuplicado

```
public RegistroDuplicado()
```

#### RegistroDuplicado

```
public RegistroDuplicado(java.lang.String message)
```

## LecturaXML

## LecturaXML:

### *Method Summary*

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
Hospital	leer(java.io.InputStream archivo)	La lectura del documento XML da como resultado un hospital de forma abstracta

### *Constructor Detail*

#### lecturaXML

```
public lecturaXML()
```

CONSTRUCTOR VACIO DE LA CLASE LEER XML

### *Method Detail*

#### leer

```
public Hospital leer(java.io.InputStream archivo) throws javax.xml.parsers.ParserConfigurationException, org.xml.sax.SAXException, java.io.IOException
```

La lectura del documento XML da como resultado un hospital de forma abstracta

Parameters:

archivo -

Returns:

Throws:

javax.xml.parsers.ParserConfigurationException

org.xml.sax.SAXException

java.io.IOException

## Objetos

## Admin:

### *Method Summary*

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
java.lang.String	toString()	

***Constructor Detail*****Admin**

```
public Admin()
CONSTRUCTOR VACIO PARA ADMINISTRADOR
```

**Admin**

```
public Admin(java.lang.String nombre,
            java.lang.String codigo,
            java.lang.String DPI,
            java.lang.String password)
CONSTRUCTOR CON PARAMETROS PARA ADMINISTRADOR
```

**Parameters:**

nombre -  
codigo -  
DPI -  
password -

***Method Detail*****toString**

```
public java.lang.String toString()
Overrides:
toString in class usuarioSistema
```

## Archivo:

***Method Summary*****All Methods****Instance Methods****Concrete Methods**

Modifier and Type	Method	Description
java.lang.String	getContentType()	
java.io.InputStream	getDatos()	
java.lang.String	getNombre()	
void	setContentType(java.lang.String ContentType)	
void	setDatos(java.io.InputStream datos)	
void	setNombre(java.lang.String nombre)	
java.lang.String	toString()	

***Constructor Detail*****Archivo**

```
public Archivo(java.lang.String nombre,
              java.io.InputStream datos)
```

**Archivo**

```
public Archivo()
```

**Method Detail**

**getNombre**

```
public java.lang.String getNombre()
```

**setNombre**

```
public void setNombre(java.lang.String nombre)
```

**getDatos**

```
public java.io.InputStream getDatos()
```

**setDatos**

```
public void setDatos(java.io.InputStream datos)
```

```
public void setDatos(java.io.InputStream datos)
```

**setContentType**

```
public void setContentType(java.lang.String ContentType)
```

**getContentType**

```
public java.lang.String getContentType()
```

**toString**

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

**Cita:****Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
java.lang.Long	getCodigo()		
java.lang.String	getCodigoMedico()		
java.lang.Long	getCodigoPaciente()		
java.lang.String	getEspecialidad()		
java.sql.Date	getFecha()		
java.sql.Time	getHora()		
void	setCodigo(java.lang.Long codigo)		
void	setCodigoMedico(java.lang.String codigoMedico)		
void	setCodigoPaciente(java.lang.Long codigoPaciente)		
void	setEspecialidad(java.lang.String especialidad)		
void	setFecha(java.sql.Date fecha)		
void	setHora(java.sql.Time hora)		
java.lang.String	toString()		

**Constructor Detail****Cita**

```
public Cita()
```

**Cita**

```
public Cita(java.lang.Long codigo,
           java.lang.Long codigoPaciente,
           java.lang.String codigoMedico,
           java.lang.String especialidad,
           java.sql.Date fecha,
           java.sql.Time hora)
```

**Method Detail****getCodigo**

```
public java.lang.Long getCodigo()
```

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

**getCodigoPaciente**

```
public java.lang.Long getCodigoPaciente()
```

**setCodigoPaciente**

```
public void setCodigoPaciente(java.lang.Long codigoPaciente)
public void setCodigoPaciente(java.lang.Long codigoPaciente)
```

**getCodigoMedico**

```
public java.lang.String getCodigoMedico()
```

**setCodigoMedico**

```
public void setCodigoMedico(java.lang.String codigoMedico)
```

**getFecha**

```
public java.sql.Date getFecha()
```

**setFecha**



```

setFecha
public void setFecha(java.sql.Date fecha)

getHora
public java.sql.Time getHora()

setHora
public void setHora(java.sql.Time hora)

getEspecialidad
public java.lang.String getEspecialidad()

setEspecialidad
public void setEspecialidad(java.lang.String especialidad)

toString
public java.lang.String toString()
Overrides:
toString in class java.lang.Object

```

## Consulta:

### Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type		Method	Description
java.lang.Long		getCodigo()	
java.lang.Double		getCosto()	
java.lang.String		getTipo()	
void		setCodigo(java.lang.Long codigo)	
void		setCosto(java.lang.Double costo)	
void		setTipo(java.lang.String tipo)	
java.lang.String		toString()	

### Constructor Detail

Consulta
public Consulta()

**Method Detail**

**getTipo**

```
public java.lang.String getTipo()
```

**setTipo**

```
public void setTipo(java.lang.String tipo)
```

**getCosto**

```
public java.lang.Double getCosto()
```

**public java.lang.Double getCosto()**

**setCosto**

```
public void setCosto(java.lang.Double costo)
```

**getCodigo**

```
public java.lang.Long getCodigo()
```

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

**toString**

**toString**

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

**Doctor:****Method Summary**

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method		Description
void	agregarEspecialidad(java.lang.String especialidad)		Asigna un especialidad al doctor
java.lang.String	getColegiado()		Retorna el numero de colegiado
java.util.ArrayList<java.lang.String>	getEspecialidad()		Retorna las especialidades del doctor
java.sql.Time	getFin()		Retorna el fin de horario del doctor
java.sql.Time	getInicio()		Retorna el inicio de horario del doctor
java.sql.Date	getInicioTrabajo()		Retorna el inicio de trabajo del doctor
void	setColegiado(java.lang.String colegiado)		Asigna el numero de colegiado
void	setEspecialidad(java.util.ArrayList<java.lang.String> especialidad)		Asigna las especialidades del doctor
void	setFin(java.sql.Time fin)		Asigna el fin de horario del doctor
void	setInicio(java.sql.Time inicio)		Asigna el inicio de horario del doctor
void	setInicioTrabajo(java.sql.Date inicioTrabajo)		asigna el inicio de trabajo del doctor
java.lang.String	toString()		

**Constructor Detail****Doctor**

```
public Doctor()
```

**Method Detail****getColegiado**

```
public java.lang.String getColegiado()
```

Retorna el numero de colegiado

Returns:

**setColegiado**

```
public void setColegiado(java.lang.String colegiado)
```

Asigna el numero de colegiado

Parameters:

colegiado -

**getEspecialidad**

```
public java.util.ArrayList<java.lang.String> getEspecialidad()
```

Retorna las especialidades del doctor

Returns:

**setEspecialidad**

```
public void setEspecialidad(java.util.ArrayList<java.lang.String> especialidad)
```

Asigna las especialidades del doctor

Parameters:

especialidad -

**agregarEspecialidad**

```
public void agregarEspecialidad(java.lang.String especialidad)
```

Asigna un especialidad al doctor

Parameters:

Asigna un especialidad al doctor

Parameters:

especialidad -

**getInicio**

```
public java.sql.Time getInicio()
```

Retorna el inicio de horario del doctor

Returns:

**setInicio**

```
public void setInicio(java.sql.Time inicio)
```

Asigna el inicio de horario del doctor

Parameters:

inicio -

**getFin**

```
getFin
```

public java.sql.Time getFin()

Retorna el fin de horario del doctor

Returns:

```
setFin
```

public void setFin(java.sql.Time fin)

Asigna el fin de horario del doctor

Parameters:

fin -

```
getInicioTrabajo
```

public java.sql.Date getInicioTrabajo()

Retorna el inicio de trabajo del doctor

Returns:

Returns:

```
setInicioTrabajo
```

public void setInicioTrabajo(java.sql.Date inicioTrabajo)

asigna el inicio de trabajo del doctor

Parameters:

inicioTrabajo -

```
toString
```

public java.lang.String toString()

Overrides:

toString in class `usuarioSistema`

## Examen:

### Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
java.lang.Long	getCodigo()	Retorna el codigo del examen	
java.lang.Double	getCosto()	Retorna el costo del examen	
java.lang.String	getDescripcion()	Retorna la descripcion del examen	
java.lang.String	getInforme()	Retorna el tipo de informe del examen	
java.lang.String	getNombre()	Retorna el nombre del examen	
java.lang.Boolean	isOrden()	Retorna si el examen necesita orden	
void	setCodigo(java.lang.Long codigo)	Asigna el codigo del examen	
void	setCosto(java.lang.Double costo)	Asigna el costo del examen	
void	setDescripcion(java.lang.String descripcion)	Asigna la descripcion del examen	
void	setInforme(java.lang.String informe)	Asigna el tipo de informe al examen	
void	setNombre(java.lang.String nombre)	Asigna el nombre del examen	
void	setOrden(java.lang.Boolean orden)	Asigna si el examen necesita orden	
java.lang.String	toString()		

**Constructor Detail****Examen**

```
public Examen()
```

**Method Detail****getCodigo**

```
public java.lang.Long getCodigo()
```

Retorna el codigo del examen

Returns:

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

Asigna el codigo del examen

Parameters:

codigo -

**getNombre**

```
public java.lang.String getNombre()
```

Retorna el nombre del examen

Returns:

**setNombre**

```
public void setNombre(java.lang.String nombre)
```

Asigna el nombre del examen

Parameters:

nombre -

**isOrden**

```
public java.lang.Boolean isOrden()
```

Retorna si el examen necesita orden

Returns:

**setOrden**

```
public void setOrden(java.lang.Boolean orden)
```

Asigna si el examen necesita orden

Parameters:

orden -

**getDescripcion**

```
public java.lang.String getDescripcion()
```

Retorna la descripcion del examen

Returns:

**setDescripcion**

```
public void setDescripcion(java.lang.String descripcion)
```

Asigna la descripcion del examen

```
public void setDescripcion(java.lang.String descripcion)
```

Asigna la descripción del examen

Parameters:

descripcion -

#### getCosto

```
public java.lang.Double getCosto()
```

Retorna el costo del examen

Returns:

#### setCosto

```
public void setCosto(java.lang.Double costo)
```

Asigna el costo del examen

Parameters:

costo -

#### getInforme

```
public java.lang.String getInforme()
```

Retorna el tipo de informe del examen

Returns:

#### setInforme

```
public void setInforme(java.lang.String informe)
```

Asigna el tipo de informe al examen

Parameters:

informe -

#### toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

## Hospital:

### Method Summary

All Methods	Instance Methods	Concrete Methods	
Modifier and Type		Method	Description
<code>java.util.ArrayList&lt;Admin&gt;</code>		<code>getAdmins()</code>	
<code>java.util.ArrayList&lt;Cita&gt;</code>		<code>getCitas()</code>	
<code>java.util.ArrayList&lt;Consulta&gt;</code>		<code>getConsultas()</code>	
<code>ConveracionesVariables</code>		<code>getConv()</code>	
<code>java.util.ArrayList&lt;Doctor&gt;</code>		<code>getDoctores()</code>	
<code>java.util.ArrayList&lt;Examen&gt;</code>		<code>getExamenes()</code>	
<code>java.util.ArrayList&lt;Laboratorista&gt;</code>		<code>getLaboratoristas()</code>	
<code>java.util.ArrayList&lt;Paciente&gt;</code>		<code>getPacientes()</code>	
<code>java.util.ArrayList&lt;Reporte&gt;</code>		<code>getReportes()</code>	
<code>java.util.ArrayList&lt;Resultado&gt;</code>		<code>getResultados()</code>	
<code>void</code>		<code>setAdmins(java.util.ArrayList&lt;Admin&gt; admins)</code>	
<code>void</code>		<code>setCitas(java.util.ArrayList&lt;Cita&gt; citas)</code>	
<code>void</code>		<code>setConsultas(java.util.ArrayList&lt;Consulta&gt; consultas)</code>	

```

void           setAdmins(java.util.ArrayList<Admin> admins)
void           setCitas(java.util.ArrayList<Cita> citas)
void           setConsultas(java.util.ArrayList<Consulta> consultas)
void           setConv(ConvercionesVariables conv)
void           setDoctores(java.util.ArrayList<Doctor> doctores)
void           setExamenes(java.util.ArrayList<Examen> examenes)
void           setLaboratoristas(java.util.ArrayList<Laboratorista> laboratoristas)
void           setPacientes(java.util.ArrayList<Paciente> pacientes)
void           setReportes(java.util.ArrayList<Reporte> reportes)
void           setResultados(java.util.ArrayList<Resultado> resultados)
java.lang.String toString()

```

#### **Constructor Detail**

##### **Hospital**

```

public Hospital(java.util.ArrayList<Admin> admins,
                java.util.ArrayList<Doctor> doctores,
                java.util.ArrayList<Laboratorista> laboratoristas,
                java.util.ArrayList<Paciente> pacientes,
                java.util.ArrayList<Examen> examenes,
                java.util.ArrayList<Reporte> reportes,
                java.util.ArrayList<Resultado> resultados,
                java.util.ArrayList<Cita> citas,
                java.util.ArrayList<Consulta> consultas)

```

#### **Method Detail**

##### **getConv**

```
public ConvercionesVariables getConv()
```

##### **setConv**

```
public void setConv(ConvercionesVariables conv)
```

##### **getAdmins**

```
public java.util.ArrayList<Admin> getAdmins()
```

##### **setAdmins**

```
public void setAdmins(java.util.ArrayList<Admin> admins)
```

##### **getDoctores**

```
public java.util.ArrayList<Doctor> getDoctores()
```

##### **setDoctores**

```
public void setDoctores(java.util.ArrayList<Doctor> doctores)
```

##### **getLaboratoristas**

```
public java.util.ArrayList<Laboratorista> getLaboratoristas()
```

##### **setLaboratoristas**

```
setLaboratoristas  
public void setLaboratoristas(java.util.ArrayList<Laboratorista> laboratoristas)  
  
getPacientes  
public java.util.ArrayList<Paciente> getPacientes()  
  
setPacientes  
public void setPacientes(java.util.ArrayList<Paciente> pacientes)  
  
getExamenes  
public java.util.ArrayList<Examen> getExamenes()  
  
setExamenes  
public void setExamenes(java.util.ArrayList<Examen> examenes)  
  
getReportes  
public java.util.ArrayList<Reporte> getReportes()  
  
setReportes  
public void setReportes(java.util.ArrayList<Reporte> reportes)  
  
getResultados  
public java.util.ArrayList<Resultado> getResultados()
```

```

setResultados
public void setResultados(java.util.ArrayList<Resultado> resultados)

getCitas
public java.util.ArrayList<Cita> getCitas()

setCitas
public void setCitas(java.util.ArrayList<Cita> citas)

getConsultas
public java.util.ArrayList<Consulta> getConsultas()

setConsultas
public void setConsultas(java.util.ArrayList<Consulta> consultas)

toString
public java.lang.String toString()
Overrides:
toString in class java.lang.Object

```

## HospitalHandler:

### *Method Summary*

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
void	characters(char[] ch, int start, int length)		
void	endElement(java.lang.String uri, java.lang.String localName, java.lang.String qName)		
java.util.ArrayList<Admin>	getAdmins()		
java.util.ArrayList<Cita>	getCitas()		
java.util.ArrayList<Consulta>	getConsultas()		
java.util.ArrayList<Doctor>	getDoctores()		
java.util.ArrayList<Examen>	getExamenes()		
java.util.ArrayList<Laboratorista>	getLaboratoristas()		
java.util.ArrayList<Paciente>	getPacientes()		
java.util.ArrayList<Reporte>	getReportes()		
java.util.ArrayList<Resultado>	getResultados()		
void	startElement(java.lang.String uri, java.lang.String localName, java.lang.String qName, org.xml.sax.Attributes attributes)		

**Method Detail****characters**

```
public void characters(char[] ch, int start, int length) throws org.xml.sax.SAXException  
Specified by:  
characters in interface org.xml.sax.ContentHandler  
Overrides:  
characters in class org.xml.sax.helpers.DefaultHandler  
Throws:  
org.xml.sax.SAXException
```

**endElement**

```
public void endElement(java.lang.String uri, java.lang.String localName, java.lang.String qName) throws org.xml.sax.SAXException  
Specified by:  
endElement in interface org.xml.sax.ContentHandler  
Overrides:  
Overrides:  
endElement in class org.xml.sax.helpers.DefaultHandler  
Throws:  
org.xml.sax.SAXException
```

**startElement**

```
public void startElement(java.lang.String uri, java.lang.String localName, java.lang.String qName, org.xml.sax.Attributes attributes) throws org.xml.sax.SAXException  
Specified by:  
startElement in interface org.xml.sax.ContentHandler  
Overrides:  
startElement in class org.xml.sax.helpers.DefaultHandler  
Throws:  
org.xml.sax.SAXException
```

**getAdmins**

```
public java.util.ArrayList<Admin> getAdmins()
```

**getDoctores**

```
public java.util.ArrayList<Doctor> getDoctores()
```

**getLaboratoristas**

```
public java.util.ArrayList<Laboratorista> getLaboratoristas()
```

**getPacientes**

```
public java.util.ArrayList<Paciente> getPacientes()
```

```

getExamenes
public java.util.ArrayList<Examen> getExamenes()

getReportes
public java.util.ArrayList<Reporte> getReportes()

getResultados
public java.util.ArrayList<Resultado> getResultados()

getCitas
public java.util.ArrayList<Cita> getCitas()

getConsultas
public java.util.ArrayList<Consulta> getConsultas()

```

## Laboratorista:

### *Method Summary*

All Methods	Instance Methods	Concrete Methods	
Modifier and Type	Method	Description	
void	agregarDia (java.lang.String dia)	Agrega un dia de trabajo al laboratorista	
java.util.ArrayList<java.lang.String>	getDias()	Retorna los dias de trabajo del laboratorista	
java.lang.String	getExamen()	Retrona el tipo de examen asociado al laboratorista	
java.sql.Date	getInicioTrabajo()	Retorna el inicio de labores del laboratorista	
java.lang.String	getRegistro()	Retorna el registro de salud del laboratorista	
void	setDias (java.util.ArrayList<java.lang.String> dias)	Agrega los dias de trabajo al laboratorista	
void	setExamen (java.lang.String examen)	Asigna el tipo de examen asociado al laboratorista	
void	setInicioTrabajo (java.sql.Date inicioTrabajo)	asigna el inicio de labores al laboratorista	
void	setRegistro (java.lang.String registro)	Asigna el registro de salud del laboratorista	
java.lang.String	toString()		

### *Constructor Detail*

```

Laboratorista
public Laboratorista()

CONSTRUCTOR VACIO DE LA CLASE LABORATORISTA

```

**Method Detail****getRegistro**

```
public java.lang.String getRegistro()
```

Retorna el registro de salud del laboratorista

Returns:

**setRegistro**

```
public void setRegistro(java.lang.String registro)
```

Asigna el registro de salud del laboratorista

Parameters:

registro -

**getExamen**

```
public java.lang.String getExamen()
```

Retorna el tipo de examen asociado al laboratorista

Returns:

**setExamen**

```
public void setExamen(java.lang.String examen)
```

Asigna el tipo de examen asociado al laboratorista

Parameters:

examen -

**getDias**

```
public java.util.ArrayList<java.lang.String> getDias()
```

Retorna los dias de trabajo del laboratorista

Returns:

```
agregarDia

public void agregarDia(java.lang.String dia)
Agrega un dia de trabajo al laboratorista
Parameters:
dia -
```

```
setDias

public void setDias(java.util.ArrayList<java.lang.String> dias)
Agrega los dias de trabajo al laboratorista
Parameters:
dias -
```

```
getInicioTrabajo

public java.sql.Date getInicioTrabajo()
public java.sql.Date getInicioTrabajo()
Retorna el inicio de labores del laboratorista
Returns:
```

```
setInicioTrabajo

public void setInicioTrabajo(java.sql.Date inicioTrabajo)
asigna el inicio de labores al laboratorista
Parameters:
inicioTrabajo -
```

```
toString

public java.lang.String toString()
Overrides:
toString in class usuarioSistema
```

## Paciente:

<b>Method Summary</b>		
<a href="#">All Methods</a>	<a href="#">Instance Methods</a>	<a href="#">Concrete Methods</a>
Modifier and Type	Method	Description
java.lang.Long	getCodigo()	
java.sql.Date	getCumple()	Retorna la fecha de cumpleaños del paciente
java.lang.Double	getPeso()	Retorna el peso del paciente
java.lang.String	getSangre()	Retorna el tipo de sangre del paciente
java.lang.String	getSexo()	Retorna el sexo del paciente
java.lang.String	getTelefono()	
void	setCodigo(java.lang.Long codigo)	
void	setCumple(java.sql.Date cumple)	Asigna la fecha de cumpleaños del paciente
void	setPeso(java.lang.Double peso)	Asigna el peso del paciente
void	setSangre(java.lang.String sangre)	Asigna el tipo de sangre del paciente
void	setSexo(java.lang.String sexo)	Asigna el sexo del paciente
void	setTelefono(java.lang.String telefono)	
java.lang.String	toString()	

**Constructor Detail****Paciente**

```
public Paciente()  
CONSTRUCTOR DE LA CLASE PACIETNE
```

**Paciente**

```
public Paciente(java.lang.String nombre,  
                java.lang.String DPI,  
                java.lang.String password,  
                java.lang.String telefono,  
                java.lang.String correo,  
                java.lang.String sexo,  
                java.sql.Date cumple,  
                java.lang.Double peso,  
                java.lang.String sangre)
```

CONSTRUCTOR DE PACIENTE CON ATRIBUTOS

Parameters:  
codigo -

codigo -  
nombre -  
DPI -  
password -  
telefono -  
correo -  
sexo -  
cumple -  
peso -  
sangre -

**Method Detail****getCodigo**

```
public java.lang.Long getCodigo()
```

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

**getTelefono**

```
public java.lang.String getTelefono()
```



```
setTelefono
public void setTelefono(java.lang.String telefono)

getSexo
public java.lang.String getSexo()
Retorna el sexo del paciente
Returns:
```

```
setSexo
public void setSexo(java.lang.String sexo)
Asigna el sexo del paciente
Parameters:
sexo -
```

```
getCumple
public java.sql.Date getCumple()
Retorna la fecha de cumpleaños del paciente
Returns:
```

```
setCumple
public void setCumple(java.sql.Date cumple)
Asigna la fecha de cumpleaños del paciente
Parameters:
cumple -
```

```
getPeso
public java.lang.Double getPeso()
Retorna el peso del paciente
Returns:
```

Returns:

```
setPeso
```

public void setPeso(java.lang.Double peso)

Asigna el peso del paciente

Parameters:

peso -

```
getSangre
```

public java.lang.String getSangre()

Retorna el tipo de sangre del paciente

Returns:

```
setSangre
```

public void setSangre(java.lang.String sangre)

```
setSangre
```

public void setSangre(java.lang.String sangre)

Asigna el tipo de sangre del paciente

Parameters:

sangre -

```
toString
```

public java.lang.String toString()

Overrides:

toString in class [usuarioSistema](#)

## Persona:

<b>Method Summary</b>		
<a href="#">All Methods</a>	<a href="#">Instance Methods</a>	<a href="#">Concrete Methods</a>
Modifier and Type	Method	Description
java.lang.String	getDPI()	
java.lang.String	getNombre()	
void	setDPI(java.lang.String dPI)	
void	setNombre(java.lang.String nombre)	

**Constructor Detail****Persona**

```
public Persona()
```

**Method Detail****getNombre**

```
public java.lang.String getNombre()
```

**setNombre**

```
public void setNombre(java.lang.String nombre)
```

**getDPI**

```
public java.lang.String getDPI()
```

**setDPI**

```
public void setDPI(java.lang.String dPI)
```

## Reporte:

**Method Summary**

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
java.lang.Long	getCodigo()	Retorna el codigo de reporte
java.lang.String	getCodigoMedico()	Retorna el codigo del medico
java.lang.Long	getCodigoPaciente()	Retorne ale codigo del paciente
java.sql.Date	getFecha()	Retorna la fecha de reporte
java.sql.Time	getHora()	Retorna la hora del reporte
java.lang.String	getInformeMedico()	Retrona el informe medico
void	setCodigo(java.lang.Long codigo)	Asigna el codigo de reporte
void	setCodigoMedico(java.lang.String codigoMedico)	Asigna el codigo del medico
void	setCodigoPaciente(java.lang.Long codigoPaciente)	Asigna el codigo del paciente
void	setFecha(java.sql.Date fecha)	Asigna la fecha de reporte
void	setHora(java.sql.Time hora)	Asigna la hora del reporte
void	setInformeMedico(java.lang.String informeMedico)	Asigna el informe medico
java.lang.String	toString()	

**Constructor Detail****Reporte**

```
public Reporte()
```

**Method Detail****getCodigo**

```
public java.lang.Long getCodigo()
```

Retorna el codigo de reporte

Returns:

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

Asigna el codigo de reporte

Parameters:

codigo -

**getCodigoPaciente**

```
public java.lang.Long getCodigoPaciente()
```

Retorne ale codigo del paciente

Returns:

**setCodigoPaciente**

```
public void setCodigoPaciente(java.lang.Long codigoPaciente)
```

Asigna el codigo del paciente

Parameters:

codigoPaciente -

**getCodigoMedico**

```
public java.lang.String getCodigoMedico()
```

Retorna el codigo del medico

Returns:

**setCodigoMedico**

```
public void setCodigoMedico(java.lang.String codigoMedico)
```

Asigna el codigo del medico

Parameters:

codigoMedico -

**getInformeMedico**

```
public java.lang.String getInformeMedico()
```

Retrona el informe medico

Returns:

**setInformeMedico**

```
public void setInformeMedico(java.lang.String informeMedico)
```

Asigna el informe medico

Parameters:  
informeMedico -

---

**getFecha**

```
public java.sql.Date getFecha()
```

Retorna la fecha de reporte

Returns:

---

**setFecha**

```
public void setFecha(java.sql.Date fecha)
```

Asigna la fecha de reporte

Parameters:  
fecha -

---

**getHora**

```
public java.sql.Time getHora()
```

Retorna la hora del reporte

Returns:

---

**setHora**

```
public void setHora(java.sql.Time hora)
```

Asigna la hora del reporte

Parameters:  
hora -

---

**toString**

```
public java.lang.String toString()
```

Overrides:  
toString in class java.lang.Object

## Resultado:

### Method Summary

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
java.lang.Long	getCodigo()		
java.lang.Long	getCodigoExamen()		
java.lang.String	getCodigoLaboratorista()		
java.lang.String	getCodigoMedico()		
java.lang.Long	getCodigoPaciente()		
java.sql.Date	getFecha()		
java.sql.Time	getHora()		
Archivo	getInforme()		
java.lang.String	getNombreInforme()		
java.lang.String	getNombreOrden()		
Archivo	getOrden()		
void	setCodigo(java.lang.Long codigo)		
void	setCodigoExamen(java.lang.Long codigoExamen)		
void	setCodigoLaboratorista(java.lang.String codigoLaboratorista)		

```
void           setCodigo(java.lang.Long codigo)
void           setCodigoExamen(java.lang.Long codigoExamen)
void           setCodigoLaboratorista(java.lang.String codigoLaboratorista)
void           setCodigoMedico(java.lang.String codigoMedico)
void           setCodigoPaciente(java.lang.Long codigoPaciente)
void           setFecha(java.sql.Date fecha)
void           setHora(java.sql.Time hora)
void           setInforme(Archivo informe)
void           setNombreInforme(java.lang.String Nombreinforme)
void           setNombreOrden(java.lang.String Nombreorden)
void           setOrden(Archivo orden)
java.lang.String  toString()
```

**Constructor Detail****Resultado**

```
public Resultado(java.lang.Long codigo,
                 java.lang.Long codigoPaciente,
                 java.lang.Long codigoExamen,
                 java.lang.String codigoMedico,
                 java.lang.String codigoLaboratorista,
                 java.lang.String NombreOrden,
                 java.lang.String NombreInforme,
                 java.sql.Date fecha,
                 java.sql.Time hora)
```

**Resultado**

```
public Resultado()
```

**Method Detail****getCodigo**

```
public java.lang.Long getCodigo()
```

**setCodigo**

```
public void setCodigo(java.lang.Long codigo)
```

**getCodigoPaciente**

```
public java.lang.Long getCodigoPaciente()
```



```
setCodigoPaciente
public void setCodigoPaciente(java.lang.Long codigoPaciente)

getCodigoExamen
public java.lang.Long getCodigoExamen()

setCodigoExamen
public void setCodigoExamen(java.lang.Long codigoExamen)

getCodigoLaboratorista
public java.lang.String getCodigoLaboratorista()

setCodigoLaboratorista
public void setCodigoLaboratorista(java.lang.String codigoLaboratorista)

getNombreOrden
public java.lang.String getNombreOrden()

setNombreOrden
public void setNombreOrden(java.lang.String Nombreorden)

getOrden
public Archivo getOrden()

setOrden
public void setOrden(Archivo orden)

getNombreInforme
public java.lang.String getNombreInforme()

setNombreInforme
public void setNombreInforme(java.lang.String Nombreinforme)

getInforme
public Archivo getInforme()
```

```

getInforme
public Archivo getInforme()

setInforme
public void setInforme(Archivo informe)

getFecha
public java.sql.Date getFecha()

setFecha
public void setFecha(java.sql.Date fecha)

getHora
public java.sql.Time getHora()

setHora
public void setHora(java.sql.Time hora)

getCodigoMedico
public java.lang.String getCodigoMedico()

setCodigoMedico
public void setCodigoMedico(java.lang.String codigoMedico)

toString
public java.lang.String toString()

Overrides:
toString in class java.lang.Object

```

## SolicitudExamen:

<i>Method Summary</i>		
<a href="#">All Methods</a>	<a href="#">Instance Methods</a>	<a href="#">Concrete Methods</a>
Modifier and Type	Method	Description
java.lang.Long	getCodigoSolicitud()	
void	setCodigoSolicitud(java.lang.Long codigoSolicitud)	

***Constructor Detail*****SolicitudExamen**

```
public SolicitudExamen()
```

**SolicitudExamen**

```
public SolicitudExamen(java.lang.Long codigoSolicitud,
                      java.lang.Long codigoPaciente,
                      java.lang.Long codigoExamen,
                      java.lang.String codigoMedico,
                      java.lang.String codigoLaboratorista,
                      java.sql.Date fecha)
```

***Method Detail*****getCodigoSolicitud**

```
public java.lang.Long getCodigoSolicitud()
```

**setCodigoSolicitud**

```
public void setCodigoSolicitud(java.lang.Long codigoSolicitud)
```

## Trabajador:

***Method Summary***

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
java.lang.String	getCodigo()		
java.lang.String	getTelefono()		
void	setCodigo(java.lang.String codigo)		
void	setTelefono(java.lang.String telefono)		

***Constructor Detail*****Trabajador**

```
public Trabajador()
```

**Method Detail****getCodigo**

```
public java.lang.String getCodigo()
```

**setCodigo**

```
public void setCodigo(java.lang.String codigo)
```

**getTelefono**

```
public java.lang.String getTelefono()
```

**setTelefono****setTelefono**

```
public void setTelefono(java.lang.String telefono)
```

## usuarioSistema:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
java.lang.String	getCodigoEntidad()		
java.lang.Long	getCodigoReferencia()		
java.lang.String	getEmail()		
java.lang.String	getPassword()		
java.lang.String	getRol()		
void	setCodigoEntidad(java.lang.String codigoEntidad)		
void	setCodigoReferencia(java.lang.Long codigoReferencia)		
void	setEmail(java.lang.String email)		
void	setPassword(java.lang.String password)		
void	setRol(java.lang.String rol)		
java.lang.String	toString()		

**Constructor Detail****usuarioSistema**

```
public usuarioSistema()
```

Constructor vacio del usuario del sistema

**Method Detail****getCodigoReferencia**

```
public java.lang.Long getCodigoReferencia()
```

**setCodigoReferencia**

```
public void setCodigoReferencia(java.lang.Long codigoReferencia)
```

**getRol**

```
public java.lang.String getRol()
```

**setRol**

```
public void setRol(java.lang.String rol)
```

**getEmail**

```
public java.lang.String getEmail()
```

**setEmail**

```
public void setEmail(java.lang.String email)
```

**getPassword**

```
public java.lang.String getPassword()
```

**setPassword**

```
public void setPassword(java.lang.String password)
```

**getCodigoEntidad****getCodigoEntidad**

```
public java.lang.String getCodigoEntidad()
```

**setCodigoEntidad**

```
public void setCodigoEntidad(java.lang.String codigoEntidad)
```

**toString**

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

## Paths

obtenerNombreArchivo

**Method Summary**

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
java.lang.String	obtenerNombre (java.lang.String archivo)	SE INTRODUCE UNA DIRECCION Y EL RESULTADO ES EL NOMBRE EN SI DEL ARCHIVO

**Constructor Detail**

obtenerNombreArchivo
public obtenerNombreArchivo()
CLASE UTILIZADA EN WINDOWS PARA OBTENER EL NOMBRE DE UN ARCHIVO DE UN PATH

**Method Detail**

obtenerNombre
public java.lang.String obtenerNombre (java.lang.String archivo)
SE INTRODUCE UNA DIRECCION Y EL RESULTADO ES EL NOMBRE EN SI DEL ARCHIVO
Parameters:
archivo -
Returns:

## QueryObjets

### QueryCita:

**Method Summary**

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
java.lang.String	getNombrePaciente ()	
void	setNombrePaciente (java.lang.String nombrePaciente)	
java.lang.String	toString ()	

**Constructor Detail****QueryCita**

```
public QueryCita(java.lang.Long codigo,
                 java.lang.String nombrePaciente,
                 java.lang.Long codigoPaciente,
                 java.sql.Time hora,
                 java.lang.String especialidad)
```

**Method Detail****getNombrePaciente**

```
public java.lang.String getNombrePaciente()
```

**setNombrePaciente**

```
public void setNombrePaciente(java.lang.String nombrePaciente)
```

**toString**

```
public java.lang.String toString()
```

Overrides:

`toString` in class `Cita`

## QueryPaciente:

**Method Summary**

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
java.lang.Long	getCantidadReportes()	
void	setCantidadReportes(java.lang.Long cantidadReportes)	
java.lang.String	toString()	

**Constructor Detail****QueryPaciente**

```
public QueryPaciente(java.lang.Long cantidadReportes,
                     java.lang.String nombre,
                     java.lang.Long codigo)
```

**Method Detail****getCantidadReportes**

```
public java.lang.Long getCantidadReportes()
```

**setCantidadReportes**

```
public void setCantidadReportes(java.lang.Long cantidadReportes)
```

**toString**

```
public java.lang.String toString()
```

Overrides:

`toString` in class `Paciente`

## Servlets

### ControladorAdmin:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail**

ControladorAdmin
public ControladorAdmin()

**Method Detail**

<b>doPost</b>
<pre>protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException, java.io.IOException  Overrides: doPost in class javax.servlet.http.HttpServlet  Throws: javax.servlet.ServletException java.io.IOException</pre>
<b>doGet</b>
<pre>protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException, java.io.IOException  Overrides: doGet in class javax.servlet.http.HttpServlet  Throws: javax.servlet.ServletException java.io.IOException</pre>

### ControladorCitas:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail**

ControladorCitas
public ControladorCitas()

**Method Detail****doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**ControladorCitasLab:****Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail****ControladorCitasLab**

```
public ControladorCitasLab()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException

Overrides:
doPost in class javax.servlet.http.HttpServlet

Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException

Overrides:
doGet in class javax.servlet.http.HttpServlet

Throws:
javax.servlet.ServletException
java.io.IOException
```

**ControladorCitasLabMedico:****Method Summary**

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)	
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)	

**Constructor Detail****ControladorCitasLabMedico**

```
public ControladorCitasLabMedico()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**ControladorGenerarCitaMedico:****Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail****ControladorGenerarCitaMedico**

```
public ControladorGenerarCitaMedico()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

## ControladorLaboratorista:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail**

```
controladorLaboratorista
public controladorLaboratorista()
```

**Method Detail**

**doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

## ControladorPaciente:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail**

```
controladorPaciente
public controladorPaciente()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

## ControladorReportesMedico:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail****ControladorReportesMedico**

```
public ControladorReportesMedico()
```

**Method Detail****doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

## Login:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail****login**

```
public login()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**ProcesadoExamen:****Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail****ProcesadoExamen**

```
public ProcesadoExamen()
```

**Method Detail****doPost**

```
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

**doGet**

```
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException,
java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException
```

## RedactarReporte:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
protected void	doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		
protected void	doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp)		

**Constructor Detail**

redactarReporte
public redactarReporte()

**Method Detail**

doPost
protected void doPost(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException, java.io.IOException
Overrides:
doPost in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException

doGet
protected void doGet(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse resp) throws javax.servlet.ServletException, java.io.IOException
Overrides:
doGet in class javax.servlet.http.HttpServlet
Throws:
javax.servlet.ServletException
java.io.IOException

## VerificarContenido

### VerificarContenido:

**Method Summary**

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
void	archivoXML(java.io.File archivo)		
void	validarSolicitudExamen(SolicitudExamen solicitud, java.lang.Boolean aceptacionOrden)		
boolean	validarTipoSangre(java.lang.String tipoSangre)		
void	verificarAdmin(Admin admin)		
void	verificarAdminModificado(Admin admin)		VERIFICA SI LOS ATRIBUTOS ESTAN CORRECTOS EN LA MODIFICACION DEL ADMINISTRADOR
void	verificarCitaCreada(Cita cita)		
void	verificarCitaEsportada(Cita cita)		
void	verificarConsulta(Consulta consulta)		
void	verificarDoctor(Doctor doctor)		
void	verificarDoctorModificado(Doctor doctor)		VERIFICA SI ESTA CORRECTA LA MODIFICACION DEL MEDICO
void	verificarExamenCreado(Examen examen)		
void	verificarExamenExportado(Examen examen)		

void	verificarDoctor(Doctor doctor)	
void	verificarDoctorModificado(Doctor doctor)	VERIFICA SI ESTA CORRECTA LA MODIFICACION DEL MEDICO
void	verificarExamenCreado(Examen examen)	
void	verificarExamenExportado(Examen examen)	
void	verificarExamenModificado(Examen examen)	
void	verificarLaboratorista(Laboratorista laboratorista)	
void	verificarLaboratoristaModificado(Laboratorista laboratorista)	
void	verificarPacienteCreado(Paciente paciente)	
void	verificarPacienteExportado(Paciente paciente)	
void	verificarPacienteModificado(Paciente paciente)	
void	verificarReporteCreado(Reporte reporte)	
void	verificarReporteExportado(Reporte reporte)	
void	verificarResultadoCreado(Resultado resultado, Examen examen)	PARA REALIZAR LA VERIFICACION DE LA INFORMACION EN NECESARIO SABER A QUE TIPO DE EXAMEN HACEMOS REFERENCIA
void	verificarResultadoExportado(Resultado resultado, Examen examen)	PARA REALIZAR LA VERIFICACION DE LA INFORMACION EN NECESARIO SABER A QUE TIPO DE EXAMEN HACEMOS REFERENCIA
boolean	verificarTipoInforme(java.lang.String cadena)	
void	verificarUsuario(java.lang.String user, java.lang.String password, java.lang.String rol)	

**Constructor Detail****VerificarContenido**

```
public VerificarContenido()
```

**Method Detail****verificarTipoInforme**

```
public boolean verificarTipoInforme(java.lang.String cadena)
```

**archivoXML**

```
public void archivoXML(java.io.File archivo) throws FormatoArchivoIncorrecto
```

Throws:

FormatoArchivoIncorrecto

**verificarUsuario**

```
public void verificarUsuario(java.lang.String user, java.lang.String password, java.lang.String rol) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarDoctor**

```
public void verificarDoctor(Doctor doctor) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarDoctorModificado**

```
public void verificarDoctorModificado(Doctor doctor) throws AtributosIncompletos
```

VERIFICA SI ESTA CORRECTA LA MODIFICACION DEL MEDICO

Parameters:

doctor -

Throws:

AtributosIncompletos

**verificarPacienteExportado**

```
public void verificarPacienteExportado(Paciente paciente) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarPacienteCreado**

```
public void verificarPacienteCreado(Paciente paciente) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarPacienteModificado**

```
public void verificarPacienteModificado(Paciente paciente) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarLaboratorista**

```
public void verificarLaboratorista(Laboratorista laboratorista) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarLaboratoristaModificado**

```
public void verificarLaboratoristaModificado(Laboratorista laboratorista) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarAdmin**

```
public void verificarAdmin(Admin admin) throws AtributosIncompletos
```

Throws:

AtributosIncompletos

**verificarAdminModificado**

```
public void verificarAdminModificado(Admin admin) throws AtributosIncompletos
```

VERIFICA SI LOS ATRIBUTOS ESTAN CORRECTOS EN LA MODIFICACION DEL ADMINISTRADOR

Parameters:

admin -

Throws:

Throws:  
AtributosIncompletos

**verificarReporteExportado**

```
public void verificarReporteExportado(Reporte reporte) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarReporteCreado**

```
public void verificarReporteCreado(Reporte reporte) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarCitaEsportada**

```
verificarCitaEsportada  
public void verificarCitaEsportada(Cita cita) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarCitaCreada**

```
public void verificarCitaCreada(Cita cita) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarConsulta**

```
public void verificarConsulta(Consulta consulta) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarExamenExportado**

```
public void verificarExamenExportado(Examen examen) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarExamenCreado**

```
public void verificarExamenCreado(Examen examen) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarExamenModificado**

```
public void verificarExamenModificado(Examen examen) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**verificarResultadoExportado**

```
public void verificarResultadoExportado(Resultado resultado, Examen examen) throws AtributosIncompletos  
PARA REALIZAR LA VERIFICACION DE LA INFORMACION EN NECESARIO SABER A QUE TIPO DE EXAMEN HACEMOS REFERENCIA  
Parameters:  
resultado -  
examen -  
Throws:  
com.hospital.proyecto2.excepcionesDePrograma.AtributosIncompletos  
AtributosIncompletos
```

**verificarResultadoCreado**

```
public void verificarResultadoCreado(Resultado resultado, Examen examen) throws AtributosIncompletos  
PARA REALIZAR LA VERIFICACION DE LA INFORMACION EN NECESARIO SABER A QUE TIPO DE EXAMEN HACEMOS REFERENCIA  
Parameters:  
resultado -  
resultado -  
examen -  
Throws:  
AtributosIncompletos
```

**validarSolicitudExamen**

```
public void validarSolicitudExamen(SolicitudExamen solicitud, java.lang.Boolean aceptacionOrden) throws AtributosIncompletos  
Throws:  
AtributosIncompletos
```

**validarTipoSangre**

```
public boolean validarTipoSangre(java.lang.String tipoSangre)
```