



# **FACULTAD DE INGENIERÍA UAEMéx**



**Universidad Autónoma del Estado de  
México**

**Facultad de Ingeniería**

**Sistemas Gestores de Bases de Datos**

**Proyecto: Consultorio dental**

**Alumno:**

**Jiménez Rosas Victor Manuel**

## Índice

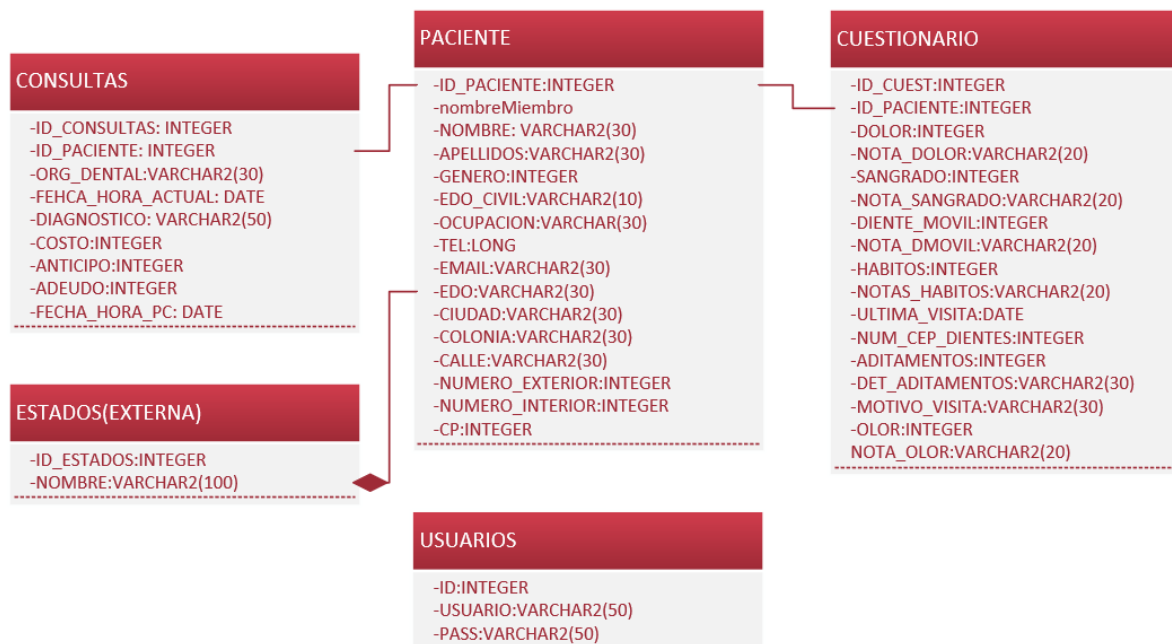
PROBLEMÁTICA.....	3
MODELO DE DATOS .....	3
DICCIONARIO DE DATOS.....	4
REQUERIMIENTOS DE HW Y SW .....	6
SCRIPTS .....	7
<i>SQLLoader:</i> .....	7
<i>Creación de objetos:</i> .....	7
<i>Manejo de datos:</i> .....	16
<i>Acceso</i> .....	16
<i>Ventanas</i> .....	16
ESTRUCTURA .....	20
LOGICA .....	20
FISICA .....	20
PERMISOS.....	21
BACKUP .....	22
EXTRACTOS DE CODIGO CON SENTENCIAS SQL .....	25
CODIGO COMPLETO .....	56
REFERENCIAS .....	182

## PROBLEMÁTICA

Cierto consultorio dental requiere una aplicación para la administración de las citas y pacientes de su consultorio, en la actualidad se encuentran realizando el registro por medio de una libreta, mantienen el registro tanto de sus pacientes regulares, adeudos de los mismos, y citas del día.

## MODELO DE DATOS

Para la creación del proyecto se utilizaron 5 tablas, 3 de ellas para el funcionamiento y almacenamiento de la información relacionada a la utilización del programa, 1 externa para cargar datos al programa y 1 para administrar la seguridad de la aplicación.



## DICCIONARIO DE DATOS

### Consultas:

COLUMN NAME	DATA TYPE	NULLABLE	DATA DEFAULT	COLUMN ID	COMMENTS
ID_CONSULTA	NUMBER(38,0)	No	null	1	null
ID_PACIENTE	NUMBER(38,0)	No	null	2	null
ORG_DENTAL	VARCHAR2(30 BYTE)	Yes	null	3	null
FECHA_HORA_ACTUAL	DATE	Yes	null	4	null
DIAGNOSTICO	VARCHAR2(50 BYTE)	Yes	null	5	null
COSTO	NUMBER	Yes	null	6	null
ANTICIPO	NUMBER	Yes	null	7	null
ADEUDO	NUMBER	Yes	null	8	null
FECHA_HORA_PC	DATE	Yes	null	9	null

### Cuestionario:

COLUMN NAME	DATA TYPE	NULLABLE	DATA DEFAULT	COLUMN ID	COMMENTS
ID_CUEST	NUMBER(38,0)	No	null	1	null
ID_PACIENTE	NUMBER(38,0)	No	null	2	null
DOLOR	NUMBER(38,0)	Yes	null	3	null
NOTA_DOLOR	VARCHAR2(20 BYTE)	Yes	null	4	null
SANGRADO	NUMBER(38,0)	Yes	null	5	null
NOTA_SANGRADO	VARCHAR2(20 BYTE)	Yes	null	6	null
DIENTE_MOVIL	NUMBER(38,0)	Yes	null	7	null
NOTA_DMOVIL	VARCHAR2(20 BYTE)	Yes	null	8	null
HABITOS	NUMBER(38,0)	Yes	null	9	null
NOTAS_HABITOS	VARCHAR2(20 BYTE)	Yes	null	10	null
ULTIMA_VISITA	DATE	Yes	null	11	null
NUM_CEP_DIENTES	NUMBER(38,0)	Yes	null	12	null
ADITAMENTOS	NUMBER(38,0)	Yes	null	13	null
DET_ADITAMENTOS	VARCHAR2(30 BYTE)	Yes	null	14	null
MOTIVO_VISITA	VARCHAR2(30 BYTE)	Yes	null	15	null
OLOR	NUMBER(38,0)	Yes	null	16	null
NOTA_OLOR	VARCHAR2(20 BYTE)	Yes	null	17	null

### *Paciente:*

<u>COLUMN NAME</u>	<u>DATA TYPE</u>	<u>NULLABLE</u>	<u>DATA DEFAULT</u>	<u>COLUMN ID</u>	<u>COMMENTS</u>
ID_PACIENTE	NUMBER(38,0)	No	null	1	null
NOMBRE	VARCHAR2(30 BYTE)	Yes	null	2	null
APELLIDOS	VARCHAR2(30 BYTE)	Yes	null	3	null
GENERO	NUMBER(38,0)	Yes	null	4	null
EDO_CIVIL	VARCHAR2(10 BYTE)	Yes	null	5	null
OCUPACION	VARCHAR2(30 BYTE)	Yes	null	6	null
TEL	LONG	Yes	null	7	null
EMAIL	VARCHAR2(30 BYTE)	Yes	null	8	null
EDO	VARCHAR2(30 BYTE)	Yes	null	9	null
CIUDAD	VARCHAR2(30 BYTE)	Yes	null	10	null
COLONIA	VARCHAR2(30 BYTE)	Yes	null	11	null
CALLE	VARCHAR2(30 BYTE)	Yes	null	12	null
NUMERO_EXTERIOR	NUMBER(38,0)	Yes	null	13	null
NUMERO_INTERIOR	NUMBER(38,0)	Yes	null	14	null
CP	NUMBER(38,0)	Yes	null	15	null

### *Estados:*

<u>COLUMN NAME</u>	<u>DATA TYPE</u>	<u>NULLABLE</u>	<u>DATA DEFAULT</u>	<u>COLUMN ID</u>	<u>COMMENTS</u>
ID_ESTADOS	NUMBER(38,0)	Yes	null	1	null
NOMBRE	VARCHAR2(100 BYTE)	Yes	null	2	null

### *Usuarios:*

<u>COLUMN NAME</u>	<u>DATA TYPE</u>	<u>NULLABLE</u>	<u>DATA DEFAULT</u>	<u>COLUMN ID</u>	<u>COMMENTS</u>
ID	NUMBER(38,0)	No	"DENTISTA","ISEQ\$\$_73533".nextval	1	null
USUARIO	VARCHAR2(50 BYTE)	Yes	null	2	null
PASS	VARCHAR2(50 BYTE)	Yes	null	3	null

## REQUERIMIENTOS DE HW Y SW

*Para el desarrollo de la aplicación se utilizaron los siguientes requerimientos:*

### *Hardware:*

- Computadora: Dell Inspiron 15 5575
- CPU: 8 núcleos, usado por BD: 2 núcleos
- RAM: 8 GB, usado por BD: 2GB
- HDD: HDD de 320GB, usado por BD: 80 GB
- SSD: 500GB

### *Software:*

- Sistema operativo: Aplicación sobre Windows 10 home, BD sobre Oracle Linux 7
- Base de datos: Oracle database express 18c
- IDE: Apache Netbeans 13
- Extras: Oracle SQL Developer, Oracle VirtualBox

## SCRIPTS

*Los siguientes scripts fueron usados para la consulta, manejo de los datos en la base de datos y creación de objetos pertenecientes a la base.*

### *SQLLoader:*

- sqlldr control='/home/oracle/Documentos/loader/control.txt'  
data='/home/oracle/Documentos/loader/usuarios.txt'  
log='/home/oracle/Documentos/loader/log.txt'  
bad='/home/oracle/Documentos/loader/bad.txt' userid='DENTISTA/123@xepdb1'

### *Creación de objetos:*

Schema:

- ALTER USER "DENTISTA"  
DEFAULT TABLESPACE "TSDENTISTA"  
TEMPORARY TABLESPACE "TSTEMPDENTISTA"  
ACCOUNT UNLOCK ;
- ALTER USER "DENTISTA" QUOTA UNLIMITED ON "TSDENTISTA";
- ALTER USER "DENTISTA" DEFAULT ROLE "RESOURCE";

### *Directorios:*

- create or replace directory tabla\_externa\_estados as  
'/home/oracle/Documentos/tabla\_externa'

### *Tablespaces:*

- CREATE SMALLFILE TABLESPACE TSDENTISTA  
DATAFILE  
'/opt/oracle/oradata/XE/XEPDB1/TsDentista01.dbf' SIZE 10485760  
AUTOEXTEND ON NEXT 10485760 MAXSIZE 34359721984  
BLOCKSIZE 8192  
DEFAULT NOCOMPRESS NO INMEMORY  
ONLINE  
SEGMENT SPACE MANAGEMENT AUTO  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
- CREATE SMALLFILE TEMPORARY TABLESPACE TSTEMPDENTISTA  
TEMPFILE  
'/opt/oracle/oradata/XE/XEPDB1/TsTempDentista01.dbf' SIZE 65011712  
AUTOEXTEND ON NEXT 655360 MAXSIZE 34359721984  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1048576;

### *Índices usados:*

- create bitmap index generoPacientes on paciente(genero);
- *create unique index "dentista"."consultas\_pk" on "dentista"."consultas" ("id\_consulta")*  
*pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" ;*
- *create unique index "dentista"."cuestionario\_pk" on "dentista"."cuestionario" ("id\_cuest")*  
*pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" ;*
- *create unique index "dentista"."paciente\_pk" on "dentista"."paciente" ("id\_paciente")*  
*pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" ;*
- *create unique index "dentista"."usuarios\_pk" on "dentista"."usuarios" ("id")*  
*pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" ;*

### *Secuencias:*

- create sequence cuestionario\_seq start with 1;
- create sequence consultas\_seq start with 1;



### *Triggers:*

- create or replace trigger cuestionario\_trigger\_seq before insert on cuestionario for each row  
begin  
select cuestionario\_seq.nextval  
into :new.id\_cuest  
from dual;  
end;
- create or replace trigger consultas\_trigger\_seq before insert on consultas for each row  
begin  
select consultas\_seq.nextval  
into :new.id\_consulta  
from dual;  
end;
- create or replace trigger pacientes\_trigger\_seq before insert on paciente for each row  
begin  
select pacientes\_seq.nextval  
into :new.id\_paciente  
from dual;  
end;

### *Funciones usadas:*

- create or replace function nombre\_completo(in\_paciente\_id in number)  
return varchar2 is  
nombrec varchar2(130);  
begin  
select paciente.nombre||' '||paciente.apellidos  
into nombrec  
from paciente  
where paciente.id\_paciente=in\_paciente\_id;  
return(nombrec);  
end nombre\_completo;
- *to\_char()*
- *to\_date()*

*Paquete y procedimientos:*

- CREATE OR REPLACE PACKAGE paquetepaciente AS  
PROCEDURE addpaciente(  
p\_nombre paciente.nombre%TYPE,  
p\_apellidos paciente.apellidos%TYPE,  
p\_genero paciente.genero%TYPE,  
p\_edo\_civil paciente.edo\_civil%TYPE,  
p\_ocupacion paciente.ocupacion%TYPE,  
p\_tel paciente.tel%TYPE,  
p\_email paciente.email%TYPE,  
p\_edo paciente.edo%TYPE,  
p\_ciudad paciente.ciudad%TYPE,  
p\_colonia paciente.colonia%TYPE,  
p\_calle paciente.calle%TYPE,  
p\_numero\_exterior paciente.numero\_exterior%TYPE,  
p\_numero\_interior paciente.numero\_interior%TYPE,  
p\_cp paciente.cp%TYPE  
);  
PROCEDURE delpaciente(id paciente.id\_paciente%TYPE);  
END paquetepaciente;  
CREATE OR REPLACE PACKAGE BODY paquetepaciente AS  
PROCEDURE addpaciente(  
p\_nombre paciente.nombre%TYPE,  
p\_apellidos paciente.apellidos%TYPE,  
p\_genero paciente.genero%TYPE,  
p\_edo\_civil paciente.edo\_civil%TYPE,  
p\_ocupacion paciente.ocupacion%TYPE,  
p\_tel paciente.tel%TYPE,  
p\_email paciente.email%TYPE,  
p\_edo paciente.edo%TYPE,  
p\_ciudad paciente.ciudad%TYPE,  
p\_colonia paciente.colonia%TYPE,  
p\_calle paciente.calle%TYPE,  
p\_numero\_exterior paciente.numero\_exterior%TYPE,  
p\_numero\_interior paciente.numero\_interior%TYPE,  
p\_cp paciente.cp%TYPE  
)  
IS

```
BEGIN
INSERT INTO paciente
(nombre,apellidos,genero,edo_civil,ocupacion,tel,email,edo,ciudad,colonia,calle,
numero_exterior,numero_interior,cp)
VALUES(p_nombre,p_apellidos,p_genero,p_edo_civil,p_ocupacion,p_tel,p_email,p_edo,p_ciudad,p_colonia,p_calle,p_numero_exterior,p_numero_interior,p_cp);
END addpaciente;
PROCEDURE delpaciente(id paciente.id_paciente%TYPE) IS
BEGIN
DELETE FROM consultas WHERE consultas.id_paciente=id;
DELETE FROM cuestionario WHERE cuestionario.id_paciente=id;
DELETE FROM paciente WHERE paciente.id_paciente = id;
END delpaciente;
END paquetepaciente;
```

*Tablas:*

**Consultas:**

- *create table "dentista"."consultas"*  
(  
    *"id\_consulta" number(\*,0) not null enable,*  
    *"id\_paciente" number(\*,0) not null enable,*  
    *"org\_dental" varchar2(30 byte),*  
    *"fecha\_hora\_actual" date,*  
    *"diagnostico" varchar2(50 byte),*  
    *"costo" number,*  
    *"anticipo" number,*  
    *"adeudo" number,*  
    *"fecha\_hora\_pc" date,*  
    *constraint "consultas\_pk" primary key ("id\_consulta")*  
    *using index pctfree 10 initrans 2 maxtrans 255 compute statistics*  
    *storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
    *pctincrease 0 freelists 1 freelist groups 1*  
    *buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
    *tablespace "tsdentista" enable,*  
    *constraint "consultas\_fk1" foreign key ("id\_paciente")*  
    *references "dentista"."paciente" ("id\_paciente") enable*  
    *) segment creation immediate*  
    *pctfree 10 pctused 40 initrans 1 maxtrans 255*  
    *nocompress logging*  
    *storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
    *pctincrease 0 freelists 1 freelist groups 1*  
    *buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
    *tablespace "tsdentista" ;*

**Cuestionarios:**

- *create table "dentista"."cuestionario"*  
(*"id\_cuest" number(\*,0) not null enable,*  
*"id\_paciente" number(\*,0) not null enable,*  
*"dolor" number(\*,0),*  
*"nota\_dolor" varchar2(20 byte),*  
*"sangrado" number(\*,0),*  
*"nota\_sangrado" varchar2(20 byte),*  
*"diente\_movil" number(\*,0),*  
*"nota\_dmovil" varchar2(20 byte),*  
*"habitos" number(\*,0),*  
*"notas\_habitos" varchar2(20 byte),*  
*"ultima\_visita" date,*  
*"num\_cep\_dientes" number(\*,0),*  
*"aditamentos" number(\*,0),*  
*"det\_aditamentos" varchar2(30 byte),*  
*"motivo\_visita" varchar2(30 byte),*  
*"olor" number(\*,0),*  
*"nota\_olor" varchar2(20 byte),*  
*constraint "cuestionario\_pk" primary key ("id\_cuest")*  
*using index pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" enable,*  
*constraint "cuestionario\_fk1" foreign key ("id\_paciente")*  
*references "dentista"."paciente" ("id\_paciente") enable*  
*) segment creation immediate*  
*pctfree 10 pctused 40 initrans 1 maxtrans 255*  
*nocompress logging*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" ;*

**Pacientes:**

- *create table "dentista"."paciente"*  
*(*       *"id\_paciente" number(\*,0) not null enable,*  
*"nombre" varchar2(30 byte),*  
*"apellidos" varchar2(30 byte),*  
*"genero" number(\*,0),*  
*"edo\_civil" varchar2(10 byte),*  
*"ocupacion" varchar2(30 byte),*  
*"tel" long,*  
*"email" varchar2(30 byte),*  
*"edo" varchar2(30 byte),*  
*"ciudad" varchar2(30 byte),*  
*"colonia" varchar2(30 byte),*  
*"calle" varchar2(30 byte),*  
*"numero\_exterior" number(\*,0),*  
*"numero\_interior" number(\*,0),*  
*"cp" number(\*,0),*  
*constraint "paciente\_pk" primary key ("id\_paciente")*  
*using index pctfree 10 initrans 2 maxtrans 255 compute statistics*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista" enable*  
*) segment creation immediate*  
*pctfree 10 pctused 40 initrans 1 maxtrans 255*  
*nocompress logging*  
*storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645*  
*pctincrease 0 freelists 1 freelist groups 1*  
*buffer\_pool default flash\_cache default cell\_flash\_cache default)*  
*tablespace "tsdentista"*  
*no inmemory ("tel") ;*

**Usuarios:**

- ```
○ create table "dentista"."usuarios"  
    ("id" number(*,0) generated always as identity minvalue 1 maxvalue  
99999999999999999999999999999999 increment by 1 start with 1 cache 20 noorder  
nocycle nokeep noscale not null enable,  
"usuario" varchar2(50 byte),  
"pass" varchar2(50 byte),  
constraint "usuarios_pk" primary key ("id")  
using index pctfree 10 initrans 2 maxtrans 255 compute statistics  
storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645  
pctincrease 0 freelists 1 freelist groups 1  
buffer_pool default flash_cache default cell_flash_cache default)  
tablespace "tsdentista" enable  
) segment creation immediate  
pctfree 10 pctused 40 initrans 1 maxtrans 255  
nocompress logging  
storage(initial 65536 next 1048576 minextents 1 maxextents 2147483645  
pctincrease 0 freelists 1 freelist groups 1  
buffer_pool default flash_cache default cell_flash_cache default)  
tablespace "tsdentista" ;
```

**Estados:**

- *create table estados  
(id\_estados number(\*, 0)  
, nombre varchar2(100))  
organization external  
( type oracle\_loader  
default directory tabla\_externa\_estados  
access parameters (  
records delimited by newline  
fields terminated by ','  
missing field values are null)  
location ( tabla\_externa\_estados: 'tabla\_externa.txt' ) )*

### *Manejo de datos:*

#### *Acceso*

**Conexión:** jdbc:oracle:thin:@server.oracle.com:1521/xepdb1

**Usuario:** *dentista*

**Password:** 123

#### *Ventanas*

##### **Ventana Login:**

- SELECT usuario from usuarios where usuario=? and pass=? and rownum=1

##### **Ventana Lobby:**

- SELECT nombre\_completo(id\_paciente) as nombre,to\_char(consultas.fecha\_hora\_pc,'HH:MI:SS') as hora FROM consultas NATURAL JOIN paciente where TO\_CHAR(consultas.fecha\_hora\_pc,'DD/MM/YY')=(select TO\_CHAR(current\_date,'DD/MM/YY') from dual)

##### **Ventana acceso a pacientes:**

- SELECT COUNT(id\_paciente) FROM paciente

##### **Ventana agregar pacientes:**

- select count(id\_paciente) as conteo from paciente
- {call paquetepaciente.addpaciente(?,?,?,?,?,?,?,?,?,?,?,?,?)}
- select \* from paciente where rownum=1 order by id\_paciente desc
- UPDATE paciente SET nombre=?,apellidos=?,genero=?,edo\_civil=?,ocupacion=?,tel=?,email=?,edo=?,ciudad=?,colonia=?,calle=?,numero\_exterior=?,numero\_interior=?,cp=? where id\_paciente=?
- SELECT \* from paciente where id\_paciente=? and rownum=1 order by id\_paciente DESC



### Ventana buscar paciente:

- {call paquetepaciente.delpaciente(?)}
- SELECT id\_paciente,nombre\_completo(id\_paciente) as nombre from paciente where upper(nombre) LIKE upper(?) or upper(apellidos) LIKE upper(?) order by id\_paciente DESC

### Ventana Citas:

- select org\_dental from consultas where fecha\_hora\_pc=to\_date(?, 'DD/MM/YYYY HH')
- insert into consultas(id\_paciente,org\_dental,fecha\_hora\_actual,diagnostico,costo,anticipo,adeudo,fecha\_hora\_pc) values(?,?,to\_date(?, 'DD/MM/YYYY'),?,?,?,to\_date(?, 'DD/MM/YYYY HH'))
- insert into consultas(id\_paciente,org\_dental,fecha\_hora\_actual,diagnostico,costo,anticipo,adeudo,fecha\_hora\_pc) values(?,?,to\_date(?, 'DD/MM/YYYY'),?,?,?,to\_date(?, 'DD/MM/YYYY HH'))
- select id\_paciente from consultas where fecha\_hora\_pc=to\_date(?, 'DD/MM/YYYY HH')
- UPDATE consultas SET org\_dental=?,diagnostico=?,costo=?,anticipo=?,adeudo=?,fecha\_hora\_pc=to\_date(?, 'DD/MM/YYYY HH') where id\_consulta=?
- SELECT nombre\_completo(id\_paciente) as nombre from paciente where id\_paciente=? and rownum=1 order by id\_paciente DESC
- SELECT id\_consulta,id\_paciente,org\_dental,to\_char(consultas.fecha\_hora\_actual,'dd/MM/yyyy') as fecha\_actual,diagnostico,costo,anticipo,adeudo,to\_char(consultas.fecha\_hora\_pc,'dd/MM/yyyy') as fecha\_pc,to\_char(consultas.fecha\_hora\_pc,'HH') as hora\_pc from consultas where id\_paciente=? and rownum=1 order by id\_paciente DESC

**Ventana buscar citas:**

- SELECT id\_paciente,nombre\_completo(id\_paciente) as nombre from paciente natural join consultas where upper(nombre) LIKE upper(?) or upper(apellidos) LIKE upper(?) order by id\_paciente DESC
- DELETE FROM consultas WHERE id\_paciente=?
- SELECT id\_paciente,nombre\_completo(id\_paciente) as nombre from paciente where upper(nombre) LIKE upper(?) or upper(apellidos) LIKE upper(?) order by id\_paciente DESC
- select count(id\_paciente) as conteo from consultas where id\_paciente=?

**Ventana crear usuarios:**

- SELECT usuario from usuarios where usuario=? and rownum=1
- insert into usuarios(usuario,pass) values(?,?)

**Ventana buscar usuarios:**

- SELECT id,usuario from usuarios where usuario LIKE ? and id!=1 and usuario!=? order by id DESC
- DELETE FROM usuarios WHERE id=?

**Ventana cuestionario:**

- INSERT INTO cuestionario  
(id\_paciente,dolor,olor,sangrado,diente\_movil,habitos,aditamentos,nota\_dolor,nota\_olor,nota\_sangrado,nota\_dmovil,notas\_habitos,det\_aditamentos,ultima\_visita,num\_cep\_dientes,motivo\_visita)  
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,to\_date(?,'DD/MM/YYYY'),?,?)

### **Ventana Reportes:**

- `SELECT nombre_completo(id_paciente) as nombre, fecha_hora_pc FROM consultas NATURAL JOIN paciente where consultas.fecha_hora_pc BETWEEN to_date(?, 'DD/MM/YY') and to_date(?, 'DD/MM/YY')+1 order by fecha_hora_pc ASC`
- `SELECT id_paciente, nombre_completo(id_paciente) as nombre, tel, motivo_visita FROM cuestionario NATURAL JOIN paciente`
- `select nombre_completo(paciente.id_paciente) as nombre, SUM(consultas.adeudo) AS adeudo FROM consultas LEFT JOIN paciente on consultas.id_paciente = paciente.id_paciente GROUP BY paciente.id_paciente, nombre, adeudo having adeudo > 0`
- `SELECT nombre_completo(id_paciente) as nombre, fecha_hora_pc FROM consultas NATURAL JOIN paciente where to_date(TO_CHAR(consultas.fecha_hora_pc, 'DD/MM/YY')) >= current_date order by fecha_hora_pc ASC`

## ESTRUCTURA

### LOGICA

*Para la realización del proyecto se creó el esquema “DENTISTA” para almacenar todos los objetos del proyecto, posteriormente se le otorgaron permisos básicos para comenzar con la creación de otros objetos.*

*Se creo el tablespace “TSDENTISTA” como espacio para alojar todas las tablas del programa, se creo a su vez un tablespace temporal llamado “TSTEMP DENTISTA” y se configuraron como los espacios por defecto.*

*Dentro del espacio de tablas “TSDENTISTA” se encuentran todas las tablas del proyecto.*

*Los triggers, índices y secuencias se encuentran en el mismo esquema que las tablas.*

### FISICA

*La base de datos se encuentra ubicada de forma física en:*


- *'/opt/oracle/oradata/XE/XEPDB1/TsDentista01.dbf'*
- *'/opt/oracle/oradata/XE/XEPDB1/TsTempDentista01.dbf'*

PERMISOS

| USERNAME | GRANTED_ROLE | ADM | DEL | DEF | OS_ | COM | INH |
|----------|--------------|-----|-----|-----|-----|-----|-----|
| DENTISTA | RESOURCE     | NO  | NO  | YES | NO  | NO  | NO  |

| USERNAME | PRIVILEGE            | ADM | COM | INH |
|----------|----------------------|-----|-----|-----|
| DENTISTA | CREATE SYNONYM       | NO  | NO  | NO  |
| DENTISTA | DROP ANY SYNONYM     | NO  | NO  | NO  |
| DENTISTA | DROP TABLESPACE      | NO  | NO  | NO  |
| DENTISTA | ALTER SESSION        | NO  | NO  | NO  |
| DENTISTA | CREATE SESSION       | NO  | NO  | NO  |
| DENTISTA | CREATE ANY DIRECTORY | NO  | NO  | NO  |
| DENTISTA | CREATE DATABASE LINK | NO  | NO  | NO  |
| DENTISTA | ALTER TABLESPACE     | NO  | NO  | NO  |
| DENTISTA | DROP ANY DIRECTORY   | NO  | NO  | NO  |
| DENTISTA | DROP ANY SEQUENCE    | NO  | NO  | NO  |
| DENTISTA | DROP ANY TRIGGER     | NO  | NO  | NO  |
| DENTISTA | CREATE SEQUENCE      | NO  | NO  | NO  |
| DENTISTA | CREATE TRIGGER       | NO  | NO  | NO  |
| DENTISTA | DROP ANY TABLE       | NO  | NO  | NO  |
| DENTISTA | DROP ANY VIEW        | NO  | NO  | NO  |
| DENTISTA | CREATE VIEW          | NO  | NO  | NO  |
| DENTISTA | DELETE ANY TABLE     | NO  | NO  | NO  |

| GRANTEE  | OWNER | TABLE_NAME            | GRANTOR  | PRIVILEGE          | GRA | HIE | COM | TYPE      | INH |
|----------|-------|-----------------------|----------|--------------------|-----|-----|-----|-----------|-----|
| DENTISTA | SYS   | TABLA_EXTERNA_ESTADOS | SYS      | EXECUTE            | YES | NO  | NO  | DIRECTORY | NO  |
| DENTISTA | SYS   | TABLA_EXTERNA_ESTADOS | SYS      | READ               | YES | NO  | NO  | DIRECTORY | NO  |
| DENTISTA | SYS   | TABLA_EXTERNA_ESTADOS | SYS      | WRITE              | YES | NO  | NO  | DIRECTORY | NO  |
| PUBLIC   | SYS   | DENTISTA              | DENTISTA | INHERIT PRIVILEGES | NO  | NO  | NO  | USER      | NO  |

| Privilegio           | Otorgado  |
|----------------------|----------------------------------------------------------------------------------------------|
| DROP ANY TRIGGER     | <input checked="" type="checkbox"/>                                                          |
| DROP ANY SEQUENCE    | <input checked="" type="checkbox"/>                                                          |
| CREATE TRIGGER       | <input checked="" type="checkbox"/>                                                          |
| CREATE ANY DIRECTORY | <input checked="" type="checkbox"/>                                                          |
| ALTER SESSION        | <input checked="" type="checkbox"/>                                                          |
| CREATE VIEW          | <input checked="" type="checkbox"/>                                                          |
| CREATE SESSION       | <input checked="" type="checkbox"/>                                                          |
| DELETE ANY TABLE     | <input checked="" type="checkbox"/>                                                          |
| DROP ANY TABLE       | <input checked="" type="checkbox"/>                                                          |
| DROP ANY SYNONYM     | <input checked="" type="checkbox"/>                                                          |
| CREATE SYNONYM       | <input checked="" type="checkbox"/>                                                          |
| CREATE SEQUENCE      | <input checked="" type="checkbox"/>                                                          |
| DROP ANY VIEW        | <input checked="" type="checkbox"/>                                                          |
| ALTER TABLESPACE     | <input checked="" type="checkbox"/>                                                          |
| CREATE DATABASE LINK | <input checked="" type="checkbox"/>                                                          |
| DROP ANY DIRECTORY   | <input checked="" type="checkbox"/>                                                          |
| DROP TABLESPACE      | <input checked="" type="checkbox"/>                                                          |

## BACKUP

Se realizó una copia de seguridad y restauración en frío de la base de datos.

```
[oracle@Oracle ~]$ sqlplus sys/Administrador1 as sysdba@xepdb1
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Thu Jun 9 13:42:49 2022  
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

Conectado a:

```
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production  
Version 18.4.0.0.0
```

```
SQL> shutdown immediate;  
Base de datos cerrada.  
Base de datos desmontada.  
Instancia ORACLE cerrada.  
SQL> startup mount  
Instancia ORACLE iniciada.
```

```
Total System Global Area 1241512896 bytes  
Fixed Size 8895424 bytes  
Variable Size 452984832 bytes  
Database Buffers 771751936 bytes  
Redo Buffers 7880704 bytes  
Base de datos montada.
```

```
SQL> rem Victor Manuel Jimenez Rosas  
[oracle@Oracle ~]$ rman
```

```
Recovery Manager: Release 18.0.0.0.0 - Production on Thu Jun 9 13:58:35 2022  
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle and/or its affiliates. All rights reserved.
```

```
RMAN> connect target /
```

```
connected to target database: XE (DBID=2982172347, not open)
```

```
RMAN> backup database;█
```

```

output file name=/opt/oracle/product/18c/dbhomeXE/dbs/data_D-XE_I-2982172347_TS-UNDOTBS1_FNO-4_090vkhff tag=T
AG20220609T140154 RECID=12 STAMP=1106920943
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00013 name=/opt/oracle/oradata/XE/XEPDB1/TsDentista01.dbf
output file name=/opt/oracle/product/18c/dbhomeXE/dbs/data_D-XE_I-2982172347_TS-TSDENTISTA_FNO-13_0a0vkhfg ta
g=TAG20220609T140154 RECID=13 STAMP=1106920944
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00007 name=/opt/oracle/oradata/XE/users01.dbf
output file name=/opt/oracle/product/18c/dbhomeXE/dbs/data_D-XE_I-2982172347_TS-USERS_FNO-7_0b0vkhfh tag=TAG2
0220609T140154 RECID=14 STAMP=1106920945
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00012 name=/opt/oracle/oradata/XE/XEPDB1/users01.dbf
output file name=/opt/oracle/product/18c/dbhomeXE/dbs/data_D-XE_I-2982172347_TS-USERS_FNO-12_0c0vkhfi tag=TAG
20220609T140154 RECID=15 STAMP=1106920946
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 09-JUN-22

```

```

Starting Control File and SPFILE Autobackup at 09-JUN-22
piece handle=/opt/oracle/product/18c/dbhomeXE/dbs/c-2982172347-20220609-00 comment=NONE
Finished Control File and SPFILE Autobackup at 09-JUN-22

```

```

RMAN> alter database open;

```

```

Statement processed

```

```

RMAN> █

```

Se borro el archivo de la base de datos “users01.dbf” y marca error al intentar abrir la base de datos.

```

[oracle@Oracle ~]$ sqlplus sys/Administrador1 as sysdba@xepdb1

```

```

SQL*Plus: Release 18.0.0.0.0 - Production on Thu Jun 9 14:13:33 2022
Version 18.4.0.0.0

```

```

Copyright (c) 1982, 2018, Oracle. All rights reserved.

```

```

Connected to an idle instance.

```

```

SQL> startup;
ORACLE instance started.

```

```

Total System Global Area 1241512896 bytes
Fixed Size                  8895424 bytes
Variable Size              452984832 bytes
Database Buffers          771751936 bytes
Redo Buffers               7880704 bytes

```

```

Base de datos montada.

```

```

ORA-01157: no se puede identificar/bloquear el archivo de datos 7 - consulte el
archivo de rastreo del DBWR

```

```

ORA-01110: archivo de datos 7: '/opt/oracle/oradata/XE/users01.dbf'

```

```

SQL> █

```

```
[oracle@Oracle ~]$ rman
```

```
Recovery Manager: Release 18.0.0.0.0 - Production on Thu Jun 9 14:21:28 2022  
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle and/or its affiliates. All rights reserved.
```

```
RMAN> connect target /
```

```
connected to target database: XE (DBID=2982172347, not open)
```

```
RMAN> restore database;■
```

```
172347_TS-USERS_FNO-12_0c0vkhfi  
destination for restore of datafile 00012: /opt/oracle/oradata/XE/XEPDB1/users01.dbf  
channel ORA_DISK_1: copied datafile copy of datafile 00012, elapsed time: 00:00:01  
output file name=/opt/oracle/oradata/XE/XEPDB1/users01.dbf RECID=0 STAMP=0  
channel ORA_DISK_1: restoring datafile 00013  
input datafile copy RECID=13 STAMP=1106920944 file name=/opt/oracle/product/18c/dbhomeXE/dbs/data_D-XE_I-2982  
172347_TS-TSDENTISTA_FNO-13_0a0vkhfg  
destination for restore of datafile 00013: /opt/oracle/oradata/XE/XEPDB1/TsDentista01.dbf  
channel ORA_DISK_1: copied datafile copy of datafile 00013, elapsed time: 00:00:01  
output file name=/opt/oracle/oradata/XE/XEPDB1/TsDentista01.dbf RECID=0 STAMP=0  
Finished restore at 09-JUN-22
```

```
RMAN> recover database;
```

```
Starting recover at 09-JUN-22  
using channel ORA_DISK_1
```

```
starting media recovery
```

```
archived log for thread 1 with sequence 9 is already on disk as file /opt/oracle/oradata/XE/redo03.log  
archived log file name=/opt/oracle/oradata/XE/redo03.log thread=1 sequence=9  
media recovery complete, elapsed time: 00:00:03  
Finished recover at 09-JUN-22
```

```
RMAN> alter database open resetlogs;
```

```
Statement processed
```

```
RMAN>
```

Después de hacer la restauración se probó que se pudiera abrir la base de datos.

```
[oracle@Oracle ~]$ sqlplus dentista/123@xepdb1
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Thu Jun 9 14:26:01 2022  
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Hora de Ultima Conexion Correcta: Vie Jun 03 2022 23:00:01 -05:00
```

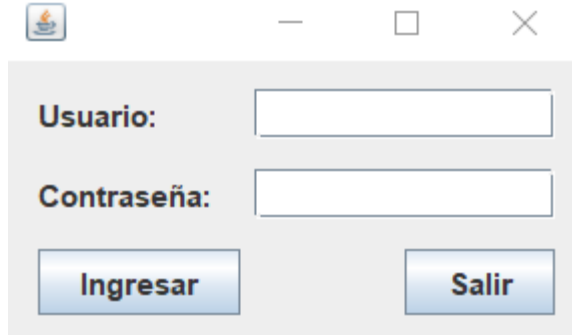
```
Conectado a:
```

```
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production  
Version 18.4.0.0.0
```

```
SQL>
```



## EXTRACTOS DE CODIGO CON SENTENCIAS SQL



```
private void ingresarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC  
    if (this.jTextField1.getText().length() > 0 && String.valueOf(this.jPasswordField1.getPassword()).length() > 0) {  
        try {  
            PreparedStatement st;  
            ResultSet rt;  
            ConexionBD con = new ConexionBD();  
            Connection cn = con.getConexion();  
            String sql = "SELECT usuario from usuarios where usuario=? and pass=? and rownum=1";  
            st = cn.prepareStatement(sql);  
            st.setString(1, this.jTextField1.getText());  
            String pas=String.valueOf(this.jPasswordField1.getPassword());  
            st.setString(2, pas);  
            rt = st.executeQuery();  
            if (rt.next()) {  
                if (rt.getString("usuario") != null) {  
                    System.out.println("acceso");  
                    this.superCtrl.setUser(this.jTextField1.getText());  
                    this.superCtrl.muestraVentanaLobby(true);  
                } else {  
                    JOptionPane.showMessageDialog(this, "El usuario o contraseña son incorrectos");  
                }  
            }  
        } catch (SQLException e) {  
            JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");  
            e.printStackTrace();  
        }  
    } else {  
        JOptionPane.showMessageDialog(this, "No debe dejar campos vacios");  
    }  
}
```

## Citas del Dia

| Nombre         | Hora     |
|----------------|----------|
| victor jimenez | 09:00:00 |
| A A            | 10:00:00 |
|                |          |

Citas

Pacientes

Reportes

Usuarios

Salir

```
public void consulta() {
    if(!"admin".equals(this.superCtrl.getUser())){
        this.users.setVisible(false);
    }else{
        this.users.setVisible(true);
    }
    try {

        this.modelo.setRowCount(0);
        PreparedStatement st = null;
        ResultSet rt = null;
        ConexionBD con = new ConexionBD();
        Connection cn = con.getConexion();
        String sql = "SELECT nombre_completo(id_paciente) as nombre,to_char(consultas.fecha_hora_pc,'HH:MI:SS') as hora FROM
consultas NATURAL JOIN paciente where TO_CHAR(consultas.fecha_hora_pc,'DD/MM/YY')=(select
TO_CHAR(current_date,'DD/MM/YY') from dual)";
        st = cn.prepareStatement(sql);
        rt = st.executeQuery();

        ResultSetMetaData rsMd = rt.getMetaData();
        int cantcolumnas = rsMd.getColumnCount();
        //System.out.println(rsMd.getColumnCount());
        //System.out.println(rsMd.toString());
        while (rt.next()) {
            //System.out.println("1");
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
                //System.out.println(filas[i]);
            }
            this.modelo.addRow(filas); }
    }
```

|                |                      |                                         |
|----------------|----------------------|-----------------------------------------|
| <b>Usuario</b> | <input type="text"/> | <input type="button" value="Crear"/>    |
| <b>Pass</b>    | <input type="text"/> | <input type="button" value="Regresar"/> |

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.con = new ConexionBD();
    Connection cn = this.con.getConnection();
    System.out.println(evt.getActionCommand());
    if(this.jTextField1.getText().length()<1||this.jTextField2.getText().length()<1){
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }else{
        try{
            String sql = "SELECT usuario from usuarios where usuario=? and rownum=1";
            st = cn.prepareStatement(sql);
            st.setString(1, this.jTextField1.getText());
            rt = st.executeQuery();
            if (rt.next()) {
                if (rt.getString("usuario") != null) {
                    JOptionPane.showMessageDialog(this, "El usuario ya existe");
                    this.jTextField1.setText(null);
                    this.jTextField2.setText(null);
                } else {
                    //JOptionPane.showMessageDialog(this, "El usuario o contraseña son incorrectos");
                    sql = "insert into usuarios(usuario,pass) values(?,?)";
                    (this.st = cn.prepareStatement(sql)).setString(1, this.jTextField1.getText());
                    this.st.setString(2, this.jTextField2.getText());
                    this.st.executeUpdate();
                    JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
                    this.jTextField1.setText(null);
                    this.jTextField2.setText(null);
                    this.superCtrl.muestraVentanaus(true);
                }
            }
        }else{
            sql = "insert into usuarios(usuario,pass) values(?,?)";
            (this.st = cn.prepareStatement(sql)).setString(1, this.jTextField1.getText());
            this.st.setString(2, this.jTextField2.getText());
            this.st.executeUpdate();
            JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
            this.jTextField1.setText(null);
            this.jTextField2.setText(null);
            this.superCtrl.muestraVentanaus(true);
        }
    }

    }catch(HeadlessException | NumberFormatException | SQLException ex4){
        JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
        JOptionPane.showMessageDialog(this, "Ocurrio un error al crear");
    }
}
}

```

**Paciente:**      **A A**

**Organo dental:**

**Diagnostico:**

**Costo:**

**Anticipo:**

**Adeudo:**      **0**

**Fecha de creacion:**      **09/06/2022**

**Proxima cita:**       **Ej. 26/01/2022**

**Hora:**   **Hrs.**

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.con = new ConexionBD();
    Connection cn = this.con.getConnection();
    System.out.println(evt.getActionCommand());
    if ("agregar".equals(evt.getActionCommand())) {

if((int)this.hora.getValue()<9||(int)this.hora.getValue()>17||this.proxcita.getText().length()<10||this.anticipo.getText().length()<1||this.c
osto.getText().length()<1||this.diagnostico.getText().length()<1||this.orgDental.getSelectedIndex() == 0){
    JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
}else if (Integer.parseInt(this.costo.getText())-Integer.parseInt(this.anticipo.getText())<0){
    JOptionPane.showMessageDialog(this, "El anticipo no puede ser mayor al costo", "Alerta", 2);
}else{
    try{
        String sql ="select org_dental from consultas where fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH')";
        (this.st = cn.prepareStatement(sql)).setString(1, this.proxcita.getText()+" "+(int)this.hora.getValue());
        rt = st.executeQuery();
        if (rt.next()) {
            if (rt.getString("org_dental") != null) {
                JOptionPane.showMessageDialog(this, "Ya existe una cita en ese dia y hora señalada");
            }else{
                sql = "insert into
consultas(id_paciente,org_dental,fecha_hora_actual,diagnostico,costo,anticipo,adeudo,fecha_hora_pc)
values(?,?,to_date(?, 'DD/MM/YYYY'),?,?,?,to_date(?, 'DD/MM/YYYY HH'))";
                (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
                this.st.setString(2, this.orgDental.getSelectedItem().toString());
                this.st.setString(3, this.fecha_actual.getText());
                this.st.setString(4, this.diagnostico.getText());
                this.st.setInt(5, Integer.parseInt(this.costo.getText()));
```

```

        this.st.setInt(6,Integer.parseInt(this.antipico.getText()));
        this.st.setInt(7,Integer.parseInt(this.adeudo.getText()));
        this.st.setString(8,this.proxcita.getText()+" "+(int)this.hora.getValue());
        this.st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
        limpiar();
        this.superCtrl.muestraVentanaCitas(true);
    }
} else{
    sql = "insert into
consultas(id_paciente,org_dental,fecha_hora_actual,diagnostico,costo,antipico,adeudo,fecha_hora_pc)
values(?,?,to_date('DD/MM/YYYY'),?,?,?,to_date('DD/MM/YYYY HH'))";
    (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
    this.st.setString(2, this.orgDental.getSelectedItem().toString());
    this.st.setString(3, this.fecha_actual.getText());
    this.st.setString(4, this.diagnostico.getText());
    this.st.setInt(5,Integer.parseInt(this.costo.getText()));
    this.st.setInt(6,Integer.parseInt(this.antipico.getText()));
    this.st.setInt(7,Integer.parseInt(this.adeudo.getText()));
    this.st.setString(8,this.proxcita.getText()+" "+(int)this.hora.getValue());
    this.st.executeUpdate();
    JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
    limpiar();
    this.superCtrl.muestraVentanaCitas(true);
}

}catch(HeadlessException | NumberFormatException | SQLException ex4){
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
    JOptionPane.showMessageDialog(this, "Ocurrio un error al crear");
}
}

}else if (JOptionPane.showConfirmDialog(null, "\u00bfEsta seguro que desea modificar?\nESTA ACCION NO SE PUEDE DESHACER", "Alerta!", 0, 2) == 0) {

if(((int)this.hora.getValue()<9||(int)this.hora.getValue()>17||this.proxcita.getText().length()<10||this.antipico.getText().length()<1||this.costo.getText().length()<1||this.diagnostico.getText().length()<1||this.orgDental.getSelectedIndex() == 0){
    JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
}else if (Integer.parseInt(this.costo.getText())-Integer.parseInt(this.antipico.getText())<0){
    JOptionPane.showMessageDialog(this, "El antipico no puede ser mayor al costo", "Alerta", 2);
}else{
    try{
        String sql ="select id_paciente from consultas where fecha_hora_pc=to_date('DD/MM/YYYY HH')";
        (this.st = cn.prepareStatement(sql)).setString(1,this.proxcita.getText()+" "+(int)this.hora.getValue());
        rt = st.executeQuery();
        if (rt.next()) {
            if ( !this.superCtrl.getClave().equals(rt.getInt("id_paciente")+"")) //{
                JOptionPane.showMessageDialog(this, "Ya existe una cita en ese dia y hora sealada");
            }else{
                sql = "UPDATE consultas SET
org_dental=?,diagnostico=?,costo=?,antipico=?,adeudo=?,fecha_hora_pc=to_date('DD/MM/YYYY HH') where id_consulta=?";
                (this.st = cn.prepareStatement(sql)).setString(1, this.orgDental.getSelectedItem().toString());
                this.st.setString(2, this.diagnostico.getText());
                this.st.setInt(3,Integer.parseInt(this.costo.getText()));
                this.st.setInt(4,Integer.parseInt(this.antipico.getText()));
                this.st.setInt(5,Integer.parseInt(this.adeudo.getText()));
                this.st.setString(6,this.proxcita.getText()+" "+(int)this.hora.getValue());
                this.st.setString(7, this.id_c+ "");

```

```

        this.st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Se modifico", "Atenci\u00f3n", 1);
        limpiar();
        this.superCtrl.muestraVentanaCitas(true);
    }
} else {
    sql = "UPDATE consultas SET
org_dental=?, diagnostico=?, costo=?, anticipo=?, adeudo=?, fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH') where id_consulta=?";
    (this.st = cn.prepareStatement(sql)).setString(1, this.orgDental.getSelectedItemId().toString());
    this.st.setString(2, this.diagnostico.getText());
    this.st.setInt(3, Integer.parseInt(this.costo.getText()));
    this.st.setInt(4, Integer.parseInt(this.anticipo.getText()));
    this.st.setInt(5, Integer.parseInt(this.adeudo.getText()));
    this.st.setString(6, this.proxcita.getText() + " +(int)this.hora.getValue());
    this.st.setString(7, this.id_c + "");
    this.st.executeUpdate();
    JOptionPane.showMessageDialog(this, "Se modifico", "Atenci\u00f3n", 1);
    limpiar();
    this.superCtrl.muestraVentanaCitas(true);
}

} catch (HeadlessException | NumberFormatException | SQLException ex4) {
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
    JOptionPane.showMessageDialog(this, "ocurrio un error al modificar");
}
}
}

}

public void modificar() {
    limpiar();
    this.jButton2.setText("Modificar");
    this.jButton2.setActionCommand("modificar");
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConnection();
        String sql = "SELECT nombre_completo(id_paciente) as nombre from paciente where id_paciente=? and rownum=1 order
by id_paciente DESC";
        //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.jLabel14.setText(this.rt.getString("nombre"));
        }
        sql = "SELECT id_consulta, id_paciente, org_dental, to_char(consultas.fecha_hora_actual, 'dd/MM/yyyy') as
fecha_actual, diagnostico, costo, anticipo, adeudo, to_char(consultas.fecha_hora_pc, 'dd/MM/yyyy') as
fecha_pc, to_char(consultas.fecha_hora_pc, 'HH') as hora_pc from consultas where id_paciente=? and rownum=1 order by
id_paciente DESC";
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.id_c = this.rt.getInt("id_consulta");
            this.superCtrl.setClave(this.rt.getString("id_paciente"));
            this.orgDental.setSelectedItem(this.rt.getString("org_dental"));
            this.diagnostico.setText(this.rt.getString("diagnostico"));
            this.costo.setText(this.rt.getInt("costo") + "");
            this.anticipo.setText(this.rt.getInt("anticipo") + "");

```

```

        this.adeudo.setText(this.rt.getInt("adeudo")+"" );
        this.fecha_actual.setText(this.rt.getString("fecha_actual")+"" );
        this.proxcita.setText(this.rt.getString("fecha_pc"));
        this.hora.setValue(Integer.parseInt(this.rt.getString("hora_pc")));
    }
}
catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
    e.printStackTrace();
}
}
public void agregar() {
    limpiar();
    Date now = new Date(System.currentTimeMillis());
    SimpleDateFormat date = new SimpleDateFormat("dd/MM/yyyy");
    SimpleDateFormat hour = new SimpleDateFormat("HH:mm:ss");
    this.fecha_actual.setText(date.format(now));
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConnection();
        final String sql = "SELECT nombre_completo(id_paciente) as nombre from paciente where id_paciente=? and rownum=1
order by id_paciente DESC";
        //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.jLabel14.setText(this.rt.getString("nombre"));
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
    }
    this.jButton2.setText("Agregar");
    this.jButton2.setActionCommand("agregar");
}
}

```

## Paciente

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente where upper(nombre) LIKE
upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";
    this.con = new ConexionBD();
    Connection cn = this.con.getConexion();
    st = null;
    ResultSet rt = null;

    /**
    try {
        //System.out.println("1");
        this.modelo.setRowCount(0);
        //System.out.println("2");
        st = cn.prepareStatement(sql);
        //System.out.println("3");
        st.setString(1, "%" + this(jTextField1.getText()) + "%");
        //System.out.println("4");
        st.setString(2, "%" + this(jTextField1.getText()) + "%");
        //System.out.println("5");
        //System.out.println(st.getParameterMetaData());
        rt = st.executeQuery();
        //System.out.println("6");
        ResultSetMetaData rsMd = rt.getMetaData();
        //System.out.println("7");
        int cantcolumnas = rsMd.getColumnCount();
        //System.out.println("8");
        //System.out.println(rsMd.getColumnCount());
        //System.out.println(rsMd.toString());
        //System.out.println(cantcolumnas);
        //System.out.println("aqui");
    }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```



```

int nfilas=0;
while (rt.next()) {
    //System.out.println("1");
    Object[] filas = new Object[cantcolumnas];
    for (int i = 0; i < cantcolumnas; i++) {
        filas[i] = rt.getObject(i + 1);
        nfilas++;
        //System.out.println(filas[i]);
    }
    //System.out.println("aqui2");
    this.modelo.addRow(filas);
}
if (nfilas>0){
    jTable1.changeSelection(0, 1, false, false);
}
//this.jTable1.setModel(this.modelo);
jTable1.getColumnModel().getColumn(0).setMinWidth(0);
jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

jTable1.getColumnModel().getColumn(1).setWidthToFit();

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

/**/
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

```

## Paciente

```
// TODO add your handling code here:
```

```
final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente natural join consultas where  
er(nombre) LIKE upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";
```

```
this.con = new ConexionBD();
Connection cn = this.con.getConexion();
st = null;
ResultSet rt = null;
```

```

/**
try {
    //System.out.println("1");
    this.modelo.setRowCount(0);
    //System.out.println("2");
    st = cn.prepareStatement(sql);
    //System.out.println("3");
    st.setString(1, "%" + this.jTextField1.getText() + "%");
    //System.out.println("4");
    st.setString(2, "%" + this.jTextField1.getText() + "%");
    //System.out.println("5");
    //System.out.println(st.getParameterMetaData());
    rt = st.executeQuery();
    //System.out.println("6");
    ResultSetMetaData rsMd = rt.getMetaData();
    //System.out.println("7");
    int cantcolumnas = rsMd.getColumnCount();
    //System.out.println("8");
    //System.out.println(rsMd.getColumnCount());
    //System.out.println(rsMd.toString());
    //System.out.println(cantcolumnas);
    //System.out.println("aqui");
}

```

```

int nfilas=0;
while (rt.next()) {
    //System.out.println("1");
    Object[] filas = new Object[cantcolumnas];
    for (int i = 0; i < cantcolumnas; i++) {
        filas[i] = rt.getObject(i + 1);
        nfilas++;
        //System.out.println(filas[i]);
    }
    //System.out.println("aqui2");
    this.modelo.addRow(filas);
}
if (nfilas>0){
    jTable1.changeSelection(0, 1, false, false);
}
//this.jTable1.setModel(this.modelo);
jTable1.getColumnModel().getColumn(0).setMinWidth(0);
jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

jTable1.getColumnModel().getColumn(1).setWidthToFit();

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

/**
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**
}

public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    if (e.getActionCommand().equals("Modificar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
        System.out.println(dato);

        //this.superCtrl.muestraVentanaModificarPacientes(true);
        this.superCtrl.setClave(dato);
        this.limpiar();
        //this.superCtrl.muestraVentanaCitas(false);
        this.superCtrl.muestraVentanaModificarCita(true);

    } else if (e.getActionCommand().equals("Eliminar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));

```

```

        System.out.println(dato);
        this.superCtrl.setClave(dato);
        if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar la cita?\nESTA ACCION NO SE PUEDE
DESHACER", "Alerta!", 0, 2) == 0) {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConexion();
            String sql = "DELETE FROM consultas WHERE id_paciente=?";
            try {
                (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
                this.st.executeQuery();
                cn.close();
                this.con.desconectar();
                this.st.close();
                this.limpiar();
                JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
                //this.superCtrl.muestraVentanaPacientes(true);
            } catch (HeadlessException | NumberFormatException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
            }
        }
    }
}

```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    final String sql = "SELECT id,usuario from usuarios where usuario LIKE ? and id!=1 and usuario!=? order by id DESC";
    this.con = new ConexionBD();
    Connection cn = this.con.getConexion();
    st = null;
    ResultSet rt = null;

    /**
    try {
        //System.out.println("1");
        this.modelo.setRowCount(0);
        //System.out.println("2");
        st = cn.prepareStatement(sql);
        //System.out.println("3");
        st.setString(1, "%" + this.jTextField1.getText() + "%");
        st.setString(2, this.superCtrl.getUser());
        //System.out.println("5");
        //System.out.println(st.getParameterMetaData());
        rt = st.executeQuery();
        //System.out.println("6");
        ResultSetMetaData rsMd = rt.getMetaData();
        //System.out.println("7");
        int cantcolumnas = rsMd.getColumnCount();
        //System.out.println("8");
        //System.out.println(rsMd.getColumnCount());
        //System.out.println(rsMd.toString());
        //System.out.println(cantcolumnas);
        //System.out.println("aqui");
        int nfilas=0;
        while (rt.next()) {
```

```

        //System.out.println("1");
        Object[] filas = new Object[cantcolumnas];
        for (int i = 0; i < cantcolumnas; i++) {
            filas[i] = rt.getObject(i + 1);
            nfilas++;
            //System.out.println(filas[i]);
        }
        //System.out.println("aqui2");
        this.modelo.addRow(filas);
    }
    if (nfilas>0){
        jTable1.changeSelection(0, 1, false, false);
    }
    //this.jTable1.setModel(this.modelo);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    jTable1.getColumnModel().getColumn(1).setWidthToFit();

    jTable1.getColumnModel().getColumn(1).setResizable(false);
    //jTable1.getColumnModel().getColumn(2).setResizable(false);
    //jTable1.setEnabled(false);

    /**/
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    if (e.getActionCommand().equals("Eliminar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
        System.out.println(dato);
        if (!"1".equals(dato)){
            if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar ?\nESTA ACCION NO SE PUEDE
DESHACER", "Alerta!", 0, 2) == 0) {
                this.con = new ConexionBD();
                final Connection cn = this.con.getConnection();
                String sql = "DELETE FROM usuarios WHERE id=?";
                try {
                    (this.st = cn.prepareStatement(sql)).setString(1, dato);
                    this.st.executeQuery();
                    cn.close();
                    this.con.desconectar();
                    this.st.close();
                    this.limpiar();
                    JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
                    //this.superCtrl.muestraVentanaPacientes(true);
                } catch (HeadlessException | NumberFormatException | SQLException ex) {

```

```
        JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
    }
}
}
```

Buscar por nombre

| Paciente |
|----------|
|----------|

```

public void actionPerformed(ActionEvent e) {

    if (e.getActionCommand().equals("Modificar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato=String.valueOf(this.modelo.getValueAt(table.getSelectedRow(),0));
        System.out.println(dato);

        //this.superCtrl.muestraVentanaModificarPacientes(true);
        this.superCtrl.setClave(dato);
        this.limpiar();
        this.superCtrl.muestraVentanaModificarPacientes(false);

    } else if (e.getActionCommand().equals("Eliminar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato=String.valueOf(this.modelo.getValueAt(table.getSelectedRow(),0));
        System.out.println(dato);
        this.superCtrl.setClave(dato);
        if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar al paciente?\nSe eliminaran todos los registros donde aparezca este paciente\nESTA ACCION NO SE PUEDE DESHACER", "Alerta!", 0, 2) == 0) {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConexion();
            //String sql = "DELETE FROM consultas WHERE id_paciente=?";
            String sql="{call paquetepaciente.delpaciente(?)}";
        }
    }
}

```



```

try {
    CallableStatement statement = cn.prepareCall(sql);
    statement.setString(1, this.superCtrl.getClave());
    boolean f;
    f=statement.execute();
    System.out.println(f);
    //proc.setString(1, "lisa");
    //cs.execute();
    //(this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
    //this.st.executeQuery();
    /*sql = "DELETE FROM cuestionario WHERE id_paciente=?";
    (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
    this.st.executeUpdate();
    sql = "DELETE FROM paciente WHERE id_paciente=? ";
    (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
    this.st.executeUpdate();*/
    cn.close();
    this.con.desconectar();
    this.st.close();
    this.limpiar();
    JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
    //this.superCtrl.muestraVentanaPacientes(true);
}
catch (HeadlessException | NumberFormatException | SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
}
}
}
}
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

        final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente where upper(nombre) LIKE
upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";

```

```

        this.con = new ConexionBD();
        Connection cn = this.con.getConexion();
        st = null;
        ResultSet rt = null;

```

```

/**
try {
    //System.out.println("1");
    this.modelo.setRowCount(0);
    //System.out.println("2");
    st = cn.prepareStatement(sql);
    //System.out.println("3");
    st.setString(1, "%" + this.jTextField1.getText() + "%");
    //System.out.println("4");
    st.setString(2, "%" + this.jTextField1.getText() + "%");
    //System.out.println("5");
    //System.out.println(st.getParameterMetaData());
    rt = st.executeQuery();
    //System.out.println("6");
    ResultSetMetaData rsMd = rt.getMetaData();

```

```

//System.out.println("7");
int cantcolumnas = rsMd.getColumnCount();
//System.out.println("8");
//System.out.println(rsMd.getColumnCount());
//System.out.println(rsMd.toString());
//System.out.println(cantcolumnas);
//System.out.println("aqui");
int nfilas=0;
while (rt.next()) {
    //System.out.println("1");
    Object[] filas = new Object[cantcolumnas];
    for (int i = 0; i < cantcolumnas; i++) {
        filas[i] = rt.getObject(i + 1);
        nfilas++;
        //System.out.println(filas[i]);
    }
    //System.out.println("aqui2");
    this.modelo.addRow(filas);
}
if (nfilas>0){
    jTable1.changeSelection(0, 1, false, false);
}
//this.jTable1.setModel(this.modelo);
jTable1.getColumnModel().getColumn(0).setMinWidth(0);
jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

jTable1.getColumnModel().getColumn(1).setWidthToFit();

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

/**/
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

```

**Ingrese intervalo de fecha (dd/mm/yyyy)**

entre fecha:  y fecha:

| Nombre         | Cita                  |
|----------------|-----------------------|
| victor jimenez | 2022-06-09 09:00:00.0 |
| A A            | 2022-06-09 10:00:00.0 |

```
private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.fechacompleta1.getText().length() < 1 || this.fechacompleta2.getText().length() < 1 ) {
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }
    else if (validafecha(this.fechacompleta1.getText()) && validafecha(this.fechacompleta2.getText()) &&
        fecha(this.fechacompleta1.getText(), this.fechacompleta2.getText())) {
        try {
            this.modelo.setRowCount(0);
            PreparedStatement st = null;
            ResultSet rt = null;
            ConexionBD con = new ConexionBD();
            Connection cn = con.getConexion();
            String sql = "SELECT nombre_completo(id_paciente) as nombre, fecha_hora_pc FROM consultas NATURAL JOIN paciente
where consultas.fecha_hora_pc BETWEEN to_date(?, 'DD/MM/YY') and to_date(?, 'DD/MM/YY')+1 order by fecha_hora_pc ASC";
            st = cn.prepareStatement(sql);
            st.setString(1, this.fechacompleta1.getText());
            st.setString(2, this.fechacompleta2.getText());
            // System.out.println("aqui1");
            rt = st.executeQuery();
            // System.out.println("aqui2");
            ResultSetMetaData rsMd = rt.getMetaData();
            int cantcolumnas = rsMd.getColumnCount();
        }
    }
}
```

```

        // System.out.println("aqui3");
        // System.out.println(cantcolumnas);
        while (rt.next()) {
            // System.out.println("aqui bucle1");
            Object[] filas = new Object[cantcolumnas];
            // System.out.println(filas);
            for (int i = 0; i < cantcolumnas; i++) {
                //System.out.println("aqui bucle2");
                filas[i] = rt.getObject(i + 1);
            }
            //System.out.println("aqui4");
            this.modelo.addRow(filas);
        }
        //System.out.println("aqui5");
        this.jTable1.setModel(this.modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    }
} else if (!validafecha(this.fechacompleta1.getText())) {
    JOptionPane.showMessageDialog(this, "Ingrese fecha inicial valida", "Alerta", 2);
} else if (!validafecha(this.fechacompleta2.getText())) {
    JOptionPane.showMessageDialog(this, "Ingrese fecha final valida", "Alerta", 2);
} else if (!fecha(this.fechacompleta1.getText(), this.fechacompleta2.getText())) {
    JOptionPane.showMessageDialog(this, "Ingrese fecha inicial menor a la final", "Alerta", 2);
}
}
}

public void paciente() {
    this.modelo.setColumnCount(0);
    this.jLabel1.setVisible(false);
    this.jLabel2.setVisible(false);
    this.jLabel3.setVisible(false);
    this.fechacompleta1.setVisible(false);
    this.fechacompleta2.setVisible(false);
    this.btnBuscar.setVisible(false);
    this.modelo.addColumn("Id");
    this.modelo.addColumn("Nombre");
    this.modelo.addColumn("Telefono");
    this.modelo.addColumn("Motivo de Visita");

    try {
        this.modelo.setRowCount(0);
        PreparedStatement st = null;
        ResultSet rt = null;
        ConexionBD con = new ConexionBD();
        Connection cn = con.getConnection();
        String sql = "SELECT id_paciente, nombre_completo(id_paciente) as nombre,tel,motivo_visita FROM cuestionario NATURAL
JOIN paciente";
        st = cn.prepareStatement(sql);
        rt = st.executeQuery();

        ResultSetMetaData rsMd = rt.getMetaData();
        int cantcolumnas = rsMd.getColumnCount();

        while (rt.next()) {
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
            }
        }
    }
}

```

```

        this.modelo.addRow(filas);
    }

    this.jTable1.setModel(this.modelo);

    TableColumnModel columnModel = this.jTable1.getColumnModel();

    columnModel.getColumn(0).setPreferredWidth(15);
    columnModel.getColumn(1).setPreferredWidth(250);
    columnModel.getColumn(2).setPreferredWidth(50);
    columnModel.getColumn(3).setPreferredWidth(250);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
}
}

public void adeudo() {
    this.modelo.setColumnCount(0);
    this.jLabel1.setVisible(false);
    this.jLabel2.setVisible(false);
    this.jLabel3.setVisible(false);
    this.fechacompleta1.setVisible(false);
    this.fechacompleta2.setVisible(false);
    this.btnBuscar.setVisible(false);
    this.modelo.addColumn("Nombre");
    this.modelo.addColumn("Adeudo");
    try {
        this.modelo.setRowCount(0);
        PreparedStatement st = null;
        ResultSet rt = null;
        ConexionBD con = new ConexionBD();
        Connection cn = con.getConexion();
        String sql = "select nombre_completo(paciente.id_paciente) as nombre,SUM(consultas.adeudo) AS adeudo FROM consultas
LEFT JOIN paciente on consultas.id_paciente = paciente.id_paciente GROUP BY paciente.id_paciente,nombre,adeudo having
adeudo>0";
        st = cn.prepareStatement(sql);
        rt = st.executeQuery();

        ResultSetMetaData rsMd = rt.getMetaData();
        int cantcolumnas = rsMd.getColumnCount();

        while (rt.next()) {
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
            }
            this.modelo.addRow(filas);
        }
        this.jTable1.setModel(this.modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    }
}

public void citas() {
    this.modelo.setColumnCount(0);
    this.jLabel1.setVisible(false);
    this.jLabel2.setVisible(false);
    this.jLabel3.setVisible(false);
    this.fechacompleta1.setVisible(false);

```

```

this.fechaCompleta2.setVisible(false);
this.btnBuscar.setVisible(false);
this.modelo.addColumn("Nombre");
this.modelo.addColumn("Cita");
try {
    this.modelo.setRowCount(0);
    PreparedStatement st = null;
    ResultSet rt = null;
    ConexionBD con = new ConexionBD();
    Connection cn = con.getConexion();
    String sql = "SELECT nombre_completo(id_paciente) as nombre, fecha_hora_pc FROM consultas NATURAL JOIN paciente
where to_date(TO_CHAR(consultas.fecha_hora_pc,'DD/MM/YY'))>=current_date order by fecha_hora_pc ASC";
    st = cn.prepareStatement(sql);
    rt = st.executeQuery();

    ResultSetMetaData rsMd = rt.getMetaData();
    int cantcolumnas = rsMd.getColumnCount();

    while (rt.next()) {
        Object[] filas = new Object[cantcolumnas];
        for (int i = 0; i < cantcolumnas; i++) {
            filas[i] = rt.getObject(i + 1);
        }
        this.modelo.addRow(filas);
    }
    this.jTable1.setModel(this.modelo);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
}
}

```



```
public void actionPerformed(ActionEvent e) {}
public void consulta() {
    //'SELECT COUNT(id_paciente) FROM paciente'
    //'COUNT(id_paciente)'
    final ConexionBD con = new ConexionBD();
    final Connection cn = con.getConexion();
    final String sql = "SELECT COUNT(id_paciente) FROM paciente";
    try {
        final PreparedStatement st = cn.prepareStatement(sql);
        final ResultSet rt = st.executeQuery();
        if (rt.next()) {
            this.lblNumeroPac.setText("" + rt.getInt("COUNT(id_paciente)"));
            cn.close();
            con.desconectar();
            rt.close();
            st.close();
        }
    }
    catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage(), "Error", 0);
    }
}
```

### Cuestionario inicial

Marque la casilla que corresponda a su situacion

☐ ¿Molestias o dolor de boca? Detalles:

☐ ¿Mal olor o sabor en la boca? Detalles:

☐ ¿Sangrado de encías? Detalles:

☐ ¿Tiene dientes moviles o le rechinan? Detalles:

☐ ¿Malos habitos orofaciales? (morder uñas/lapices/etc. chuparse los dedos, etc.) Detalles:

☐ Aparte de cepillo de dientes, ¿Utiliza algun otro aditamento de limpieza? ¿Cuales?

Ultima visita (dd/mm/yyyy):  En caso de primera visita ingrese fecha actual

Numero de limpiezas al dia:

Motivo de consulta:

```
private void btnAgregarPacienteCuestionarioActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.txtNoCepPacienteCuestionario.getText().length() < 1 || this.txtMotivoPacienteCuestionario.getText().length() < 1) {
        JOptionPane.showMessageDialog(this, "No puede dejar Motivo de consulta o numero de cepillados en blanco", "Alerta", 2);
    } else {
        try {
            if (validafecha(fechacompleta.getText())) {
                this.con = new ConexionBD();
                Connection cn = this.con.getConexion();
                try {
                    String sql = "INSERT INTO cuestionario
(id_paciente,dolor,olor,sangrado,diente_movil,habitos,aditamentos,nota_dolor,nota_olor,nota_sangrado,nota_dmovil,notas_habitos,
det_aditamentos,ultima_visita,num_cep_dientes,motivo_visita) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,to_date(?, 'DD/MM/YYYY'),?,?)";
                    this.st = cn.prepareStatement(sql);
                    this.st.setInt(1, Integer.parseInt(this.superCtrl.getClave()));
                    if (this.cbMolestia.isSelected()) {
                        this.st.setInt(2, 1);
                        this.st.setString(8, this.txtMolestiaDetalle.getText());
                    } else {
                        this.st.setInt(2, 0);
                        this.st.setNull(8, 4);
                    }
                }
                if (this.cbOlor.isSelected()) {
                    this.st.setInt(3, 1);
                    this.st.setString(9, this.txtOlorDetalle.getText());
                } else {
                    this.st.setInt(3, 0);
                    this.st.setNull(9, 4);
                }
            }
            if (this.cbSangrado.isSelected()) {
                this.st.setInt(4, 1);
                this.st.setString(10, this.txtSangradoDetalle.getText());
            } else {
                this.st.setInt(4, 0);
                this.st.setNull(10, 4);
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, ex.getMessage(), "Error", 2);
        }
    }
}
```



```

    }
    if (this.cbMovil.isSelected()) {
        this.st.setInt(5, 1);
        this.st.setString(11, this.txtMovilDetalle.getText());
    } else {
        this.st.setInt(5, 0);
        this.st.setNull(11, 4);
    }
    if (this.cbHbitos.isSelected()) {
        this.st.setInt(6, 1);
        this.st.setString(12, this.txtHbitosDetalle.getText());
    } else {
        this.st.setInt(6, 0);
        this.st.setNull(12, 4);
    }
    if (this.cbAditamentos.isSelected()) {
        this.st.setInt(7, 1);
        this.st.setString(13, this.txtAditamentosDetalle.getText());
    } else {
        this.st.setInt(7, 0);
        this.st.setNull(13, 4);
    }
    this.st.setString(14, fechacompleta.getText());
    this.st.setInt(15, Integer.parseInt(this.txtNoCepPacienteCuestionario.getText()));
    this.st.setString(16, this.txtMotivoPacienteCuestionario.getText());
    this.res = this.st.executeUpdate();
    if (this.res > 0) {
        this.con.desconectar();
        this.st.close();
        limpiar();
        this.superCtrl.muestraVentanaAgregarCuestionario(false);
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un problema al registrar el cuestionario, intente de nuevo.",
"Error", 0);
    }
    if (cn != null) {
        cn.close();
    }
} catch (Throwable throwable) {
    if (cn != null)
        try {
            cn.close();
        } catch (Throwable throwable1) {
            throwable.addSuppressed(throwable1);
        }
    throw throwable;
}
} else {
    JOptionPane.showMessageDialog(null, "Ingrese una fecha valida", "Alerta", 2);
}

} catch (HeadlessException | NumberFormatException | java.sql.SQLException e) {
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}
}

```

|                                                                                |                                                      |
|--------------------------------------------------------------------------------|------------------------------------------------------|
| <b>Nombre:</b>                                                                 | <input type="text"/>                                 |
| <b>Apellidos:</b>                                                              | <input type="text"/>                                 |
| <b>Genero:</b>                                                                 | <input type="button" value="v"/>                     |
| <b>Estado Civil:</b>                                                           | <input type="button" value="v"/>                     |
| <b>Ocupacion:</b>                                                              | <input type="text"/>                                 |
| <b>Telefono:</b>                                                               | <input type="text"/>                                 |
| <b>Correo:</b>                                                                 | <input type="text"/>                                 |
| <b>Estado:</b>                                                                 | <input type="button" value="v"/>                     |
| <b>Ciudad:</b>                                                                 | <input type="text"/>                                 |
| <b>Colonia:</b>                                                                | <input type="text"/>                                 |
| <b>Calle:</b>                                                                  | <input type="text"/>                                 |
| <b>No. Ext.</b>                                                                | <input type="text"/> <b>(Ingresa 0 si no aplica)</b> |
| <b>No. Int.</b>                                                                | <input type="text"/> <b>(Opcional)</b>               |
| <b>C.P.</b>                                                                    | <input type="text"/>                                 |
| <input type="button" value="Cancelar"/> <input type="button" value="Agregar"/> |                                                      |

```
private void btnAgregarPacienteAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if ("agregar".equals(evt.getActionCommand())) {
        if (this.cbEstadoPacienteAgregar.getSelectedIndex() == 0 || this.cbEdoCivilPacienteAgregar.getSelectedIndex() == 0 ||
            this.cbGeneroPacienteAgregar.getSelectedIndex() == 0 || this.txtCpPacienteAgregar.getText().length() < 1 ||
            this.txtNoExPacienteAgregar.getText().length() < 1 || this.txtCallePacienteAgregar.getText().length() < 1 ||
            this.txtColoniaPacienteAgregar.getText().length() < 1 || this.txtCiudadPacienteAgregar.getText().length() < 1 ||
            this.txtCorreoPacienteAgregar.getText().length() < 1 || this.txtTelefonoPacienteAgregar.getText().length() < 1 ||
            this.txtOcupacionPacienteAgregar.getText().length() < 1 || this.txtApellidosPacienteAgregar.getText().length() < 1 ||
            this.txtNombrePacienteAgregar.getText().length() < 1) {
            JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
        }
    }
}
```

```

else if (this.txtTelefonoPacienteAgregar.getText().length() <= 9) {
    JOptionPane.showMessageDialog(this, "Ingrese telefono a 10 digitos", "Alerta", 2);
}
else if (this.txtCpPacienteAgregar.getText().length() <= 4) {
    JOptionPane.showMessageDialog(this, "Ingrese un codigo postal valido a 5 digitos", "Alerta", 2);
}
else {
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConexion();
        int antes=0,despues=0;
        try {
            this.st = cn.prepareStatement("select count(id_paciente) as conteo from paciente");
            this.rt = this.st.executeQuery();
            if (this.rt.next()) {
                antes=this.rt.getInt("conteo");
            }
            //final String sql = "INSERT INTO paciente
(nombre,apellidos,genero,edo_civil,ocupacion,tel,email,edo,ciudad,colonia,calle,numero_exterior,numero_interior,cp)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
            final String sql="{call paquetepaciente.addpaciente(?,?,?,?,?,?,?,?,?,?,?,?,?)}";
            CallableStatement st = cn.prepareCall(sql);
            //st.setString(1, this.superCtrl.getClave());
            st.setString(1, this.txtNombrePacienteAgregar.getText());
            st.setString(2, this.txtApellidosPacienteAgregar.getText());
            if (this.cbGeneroPacienteAgregar.getSelectedIndex() == 1) {
                this.g = 1;
            }
            else {
                this.g = 0;
            }
            st.setInt(3, this.g);
            st.setString(4, this.cbEdoCivilPacienteAgregar.getSelectedItem().toString().toUpperCase());
            st.setString(5, this.txtOcupacionPacienteAgregar.getText());
            st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgregar.getText()));
            st.setString(7, this.txtCorreoPacienteAgregar.getText());
            st.setString(8, this.cbEstadoPacienteAgregar.getSelectedItem().toString().toUpperCase());
            st.setString(9, this.txtCiudadPacienteAgregar.getText());
            st.setString(10, this.txtColoniaPacienteAgregar.getText());
            st.setString(11, this.txtCallePacienteAgregar.getText());
            st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgregar.getText()));
            if (this.txtNoInPacienteAgregar.getText().length() < 1) {
                st.setInt(13, 0);
            }
            else {
                st.setInt(13, Integer.parseInt(this.txtNoInPacienteAgregar.getText()));
            }
            st.setInt(14, Integer.parseInt(this.txtCpPacienteAgregar.getText()));
            System.out.println("llego aqui 1");
            boolean f = st.execute();
            /*(this.st = cn.prepareStatement(sql)).setString(1, this.txtNombrePacienteAgregar.getText());
            this.st.setString(2, this.txtApellidosPacienteAgregar.getText());
            if (this.cbGeneroPacienteAgregar.getSelectedIndex() == 1) {
                this.g = 1;
            }
            else {
                this.g = 0;
            }
            }

```

```

this.st.setInt(3, this.g);
this.st.setString(4, this.cbEdoCivilPacienteAgregar.getSelectedItem().toString().toUpperCase());
this.st.setString(5, this.txtOcupacionPacienteAgregar.getText());
this.st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgregar.getText()));
this.st.setString(7, this.txtCorreoPacienteAgregar.getText());
this.st.setString(8, this.cbEstadoPacienteAgregar.getSelectedItem().toString().toUpperCase());
this.st.setString(9, this.txtCiudadPacienteAgregar.getText());
this.st.setString(10, this.txtColoniaPacienteAgregar.getText());
this.st.setString(11, this.txtCallePacienteAgregar.getText());
this.st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgregar.getText()));
if (this.txtNoInPacienteAgregar.getText().length() < 1) {
    this.st.setInt(13, 0);
}
else {
    this.st.setInt(13, Integer.parseInt(this.txtNoInPacienteAgregar.getText()));
}
this.st.setInt(14, Integer.parseInt(this.txtCpPacienteAgregar.getText()));
System.out.println("llego aqui 1");
this.res = this.st.executeUpdate();
/**/
this.rt = this.st.executeQuery("select count(id_paciente) as conteo from paciente");
if (this.rt.next()) {
    despues=this.rt.getInt("conteo");
}
System.out.println(antes);
System.out.println(despues);
if (antes!=despues) {
    System.out.println("llego aqui 2");
    this.rt = this.st.executeQuery("select * from paciente where rownum=1 order by id_paciente desc");
    System.out.println("llego aqui 2.5");
    //rt.beforeFirst();
    this.rt.next();
    System.out.println("llego aqui 3");

    this.last = this.rt.getInt(1);
    JOptionPane.showMessageDialog(this,"el id del paciente es: "+this.last, "Atenci\u00f3n", 1);
}
else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un problema al registrar", "Error", 0);
}
System.out.println("llego aqui 4");
if (cn != null) {
    cn.close();
}
}
catch (Throwable t) {
    if (cn != null) {
        try {
            cn.close();
        }
        catch (Throwable exception) {
            t.addSuppressed(exception);
        }
    }
    throw t;
}
this.con.desconectar();
this.rt.close();

```

```

        this.st.close();
        this.limpiar();
        this.superCtrl.setClave("" + this.last);
        this.superCtrl.muestraVentanaAgregarCuestionario(true);
    }
    catch (HeadlessException | NumberFormatException | SQLException ex4) {
        JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
    }
}

}

else if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea modificar al paciente?\nESTA ACCION NO SE PUEDE DESHACER", "Alerta!", 0, 2) == 0) {
    if (this.cbEstadoPacienteAgregar.getSelectedIndex() == 0 || this.cbEdoCivilPacienteAgregar.getSelectedIndex() == 0 ||
        this.cbGeneroPacienteAgregar.getSelectedIndex() == 0 || this.txtCpPacienteAgregar.getText().length() < 1 ||
        this.txtNoExPacienteAgregar.getText().length() < 1 || this.txtCallePacienteAgregar.getText().length() < 1 ||
        this.txtColoniaPacienteAgregar.getText().length() < 1 || this.txtCiudadPacienteAgregar.getText().length() < 1 ||
        this.txtCorreoPacienteAgregar.getText().length() < 1 || this.txtTelefonoPacienteAgregar.getText().length() < 1 ||
        this.txtOcupacionPacienteAgregar.getText().length() < 1 || this.txtApellidosPacienteAgregar.getText().length() < 1 ||
        this.txtNombrePacienteAgregar.getText().length() < 1) {
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }
    else if (this.txtTelefonoPacienteAgregar.getText().length() <= 9) {
        JOptionPane.showMessageDialog(this, "Ingresa telefono a 10 digitos", "Alerta", 2);
    }
    else if (this.txtCpPacienteAgregar.getText().length() <= 4) {
        JOptionPane.showMessageDialog(this, "Ingresa un codigo postal valido a 5 digitos", "Alerta", 2);
    }
    else {
        try {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConnection();
            try {
                String sql = "UPDATE paciente SET
nombre=?,apellidos=?,genero=?,edo_civil=?,ocupacion=?,tel=?,email=?,edo=?,ciudad=?,colonia=?,calle=?,numero_exterior=?,num
ero_interior=?,cp=? where id_paciente=?";
                (this.st = cn.prepareStatement(sql)).setString(1, this.txtNombrePacienteAgregar.getText());
                this.st.setString(2, this.txtApellidosPacienteAgregar.getText());
                if (this.cbGeneroPacienteAgregar.getSelectedIndex() == 1) {
                    this.g = 1;
                }
                else {
                    this.g = 0;
                }
                this.st.setInt(3, this.g);
                this.st.setString(4, this.cbEdoCivilPacienteAgregar.getSelectedItem().toString().toUpperCase());
                this.st.setString(5, this.txtOcupacionPacienteAgregar.getText());
                this.st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgregar.getText()));
                this.st.setString(7, this.txtCorreoPacienteAgregar.getText());
                this.st.setString(8, this.cbEstadoPacienteAgregar.getSelectedItem().toString().toUpperCase());
                this.st.setString(9, this.txtCiudadPacienteAgregar.getText());
                this.st.setString(10, this.txtColoniaPacienteAgregar.getText());
                this.st.setString(11, this.txtCallePacienteAgregar.getText());
                this.st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgregar.getText()));
                if (this.txtNolnPacienteAgregar.getText().length() < 1) {
                    this.st.setInt(13, 0);
                }
                else {
                    this.st.setInt(13, Integer.parseInt(this.txtNolnPacienteAgregar.getText()));
                }
            }
            catch (SQLException ex5) {
                JOptionPane.showMessageDialog(this, ex5.getMessage(), "Error", 2);
            }
        }
        catch (SQLException ex6) {
            JOptionPane.showMessageDialog(this, ex6.getMessage(), "Error", 2);
        }
    }
}

```

```

    }
    this.st.setInt(14, Integer.parseInt(this.txtCpPacienteAgregar.getText()));
    this.st.setString(15, this.superCtrl.getClave());
    this.st.executeUpdate();
    sql = "SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC";
    //select * from paciente where rownum=1 order by id_paciente desc;
    (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
    //(this.rt = this.st.executeQuery()).beforeFirst();
    this.rt = this.st.executeQuery();
    if (this.rt.next()) {
        this.last = this.rt.getInt(1);
        JOptionPane.showMessageDialog(this, "Se modifiko", "Atenci\u00f3n", 1);
    }
    else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error al modificar", "Error", 0);
    }
    if (cn != null) {
        cn.close();
    }
}
catch (Throwable t2) {
    if (cn != null) {
        try {
            cn.close();
        }
        catch (Throwable exception2) {
            t2.addSuppressed(exception2);
        }
    }
    throw t2;
}
this.con.desconectar();
this.rt.close();
this.st.close();
this.limpiar();
this.superCtrl.muestraVentanaPacientes(true);
}
catch (HeadlessException | NumberFormatException | SQLException ex4) {
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
}
}
}

}

public void modificar() {
    this.btnAgregarPacienteAgregar.setText("Modificar");
    this.btnAgregarPacienteAgregar.setActionCommand("modificar");
    this.btnCancelPacienteAgregar.setActionCommand("modificar");
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConnection();
        final String sql = "SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC";
        //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.txtNombrePacienteAgregar.setText(this.rt.getString("nombre"));
            this.txtApellidosPacienteAgregar.setText(this.rt.getString("apellidos"));

```

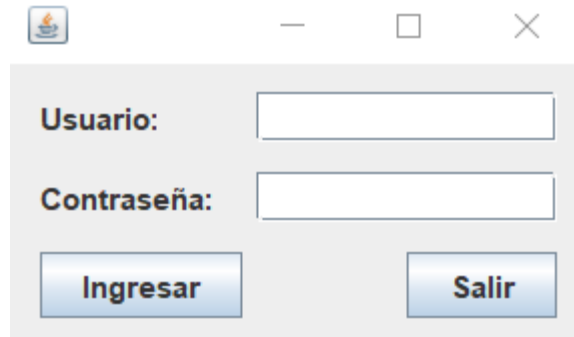
```

this.txtOcupacionPacienteAgrega.setText(this.rt.getString("ocupacion"));
this.txtTelefonoPacienteAgrega.setText(""+ this.rt.getLong("tel"));
this.txtCorreoPacienteAgrega.setText(this.rt.getString("email"));
this.txtCiudadPacienteAgrega.setText(this.rt.getString("ciudad"));
this.txtColoniaPacienteAgrega.setText(this.rt.getString("colonia"));
this.txtCallePacienteAgrega.setText(this.rt.getString("calle"));
this.txtNoExPacienteAgrega.setText(""+ this.rt.getInt("numero_exterior"));
if (this.rt.getInt("numero_interior") != 0) {
    this.txtNoInPacienteAgrega.setText(""+ this.rt.getInt("numero_interior"));
}
else {
    this.txtNoInPacienteAgrega.setText(null);
}
this.txtCpPacienteAgrega.setText(""+ this.rt.getInt("cp"));
if (this.rt.getInt("genero") == 1) {
    this.cbGeneroPacienteAgrega.setSelectedItem("H");
}
else {
    this.cbGeneroPacienteAgrega.setSelectedItem("M");
}
this.cbEdoCivilPacienteAgrega.setSelectedItem(this.formato(this.rt.getString("edo_civil")));
this.cbEstadoPacienteAgrega.setSelectedItem(this.formato(this.rt.getString("edo")));
}
}
catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
}
}

```

## CODIGO COMPLETO

*A continuación, se mostrarán todas las ventanas efectivas que conforman el proyecto en su totalidad y se mostrara el código que las hace funcionar.*



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package GUI;

import Conexion.ConexionBD;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author Jiro
 */
public class login extends VentanaControladaAbstract {

    /**
     * Creates new form login
     */
    public login() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
```



```

jLabel2 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jPasswordField1 = new javax.swing.JPasswordField();
ingresar = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("Usuario:");

jLabel2.setText("Contraseña:");

jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jTextField1KeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        jTextField1KeyTyped(evt);
    }
});

jPasswordField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jPasswordField1ActionPerformed(evt);
    }
});
jPasswordField1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jPasswordField1KeyReleased(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        jPasswordField1KeyTyped(evt);
    }
});

ingresar.setText("Ingresar");
ingresar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ingresarActionPerformed(evt);
    }
});

jButton2.setText("Salir");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(jLabel1)
                        .add(jLabel2)
                    )
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .add(jTextField1)
                )
                .add(jPasswordField1)
                .add(ingresar)
                .add(jButton2)
            )
            .addContainerGap())
        .add(layout.createSequentialGroup()
            .add(jButton2)
            .addContainerGap())
);

```

```

        .addComponent(jLabel2)
        .addComponent(jLabel1))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jTextField1)
            .addComponent(jPasswordField1, javax.swing.GroupLayout.DEFAULT_SIZE, 120, Short.MAX_VALUE)))
        .addGroup(layout.createSequentialGroup())
        .addComponent(ingresar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 50, Short.MAX_VALUE)
        .addComponent(jButton2)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(ingresar)
                .addComponent(jButton2))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

pack();
} // </editor-fold>

```

```

private void ingresarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
    if (this(jTextField1.getText()).length() > 0 && String.valueOf(this.jPasswordField1.getPassword()).length() > 0) {
        try {
            PreparedStatement st;
            ResultSet rt;
            ConexionBD con = new ConexionBD();
            Connection cn = con.getConexion();
            String sql = "SELECT usuario from usuarios where usuario=? and pass=? and rownum=1";
            st = cn.prepareStatement(sql);
            st.setString(1, this(jTextField1.getText()));
            String pas=String.valueOf(this.jPasswordField1.getPassword());
            st.setString(2, pas);
            rt = st.executeQuery();
            if (rt.next()) {
                if (rt.getString("usuario") != null) {
                    System.out.println("acceso");
                    this.superCtrl.setUser(this(jTextField1.getText()));
                    this.superCtrl.muestraVentanaLobby(true);
                } else {
                    JOptionPane.showMessageDialog(this, "El usuario o contraseña son incorrectos");
                }
            }
        }
    }
}

```

```

    }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
        e.printStackTrace();
    }
} else {
    JOptionPane.showMessageDialog(this, "No debe dejar campos vacios");
}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jPasswordField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void jPasswordField1KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode()==KeyEvent.VK_ENTER){
        this.ingresar.doClick();
    }
    if(evt.getKeyCode()==KeyEvent.VK_ESCAPE){
        System.exit(0);
    }
}

private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode()==KeyEvent.VK_ENTER){
        //JOptionPane.showMessageDialog(this, "Has pulsado Enter");
        this(jTextField1).transferFocus();
    }
    if(evt.getKeyCode()==KeyEvent.VK_ESCAPE){
        System.exit(0);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

```

```

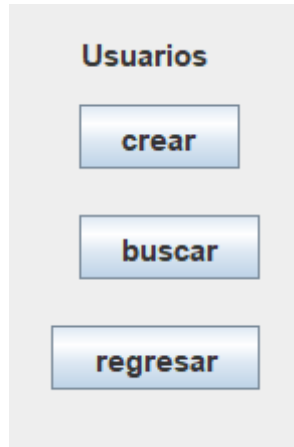
* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new login().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton ingresar;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JTextField jTextField1;
// End of variables declaration

@Override
public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package GUI;

import java.awt.event.ActionEvent;

/**
 *
 * @author Jiro
 */
public class usuarios extends VentanaControladaAbstract {

    /**
     * Creates new form usuarios
     */
    public usuarios() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jLabel1.setText("Usuarios");

        jButton1.setText("crear");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("buscar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jButton3.setText("regresar");
    jButton3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton3ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jButton3)
                        .addGap(18, 18, 18)
                        .addComponent(jButton2)
                        .addGap(18, 18, 18)
                        .addComponent(jButton1))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(25, 25, 25)))
                .addContainerGap())
            .addGroup(layout.createSequentialGroup()
                .addGap(29, 29, 29)
                .addComponent(jLabel1))
            .addGap(20, 20, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jButton2)
                .addGap(18, 18, 18)
                .addComponent(jButton3)
                .addGap(25, 25, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaLobby(true);
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestracreaus(true);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestrabuscars(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(usuarios.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(usuarios.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(usuarios.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(usuarios.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new usuarios().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
// End of variables declaration

@Override
public void actionPerformed(ActionEvent e) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

## Citas del Dia

| Nombre         | Hora     |
|----------------|----------|
| victor jimenez | 09:00:00 |
| A A            | 10:00:00 |
|                |          |

Citas

Pacientes

Reportes

Usuarios

Salir

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package GUI;
import Conexion.ConexionBD;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTable;
import javax.swing.LayoutStyle;
import javax.swing.table.DefaultTableModel;
import java.sql.SQLException;
import javax.swing.JOptionPane;
/**
 *
 * @author Jiro
 */
public class lobby extends VentanaControladaAbstract {
    private DefaultTableModel modelo = new DefaultTableModel();
    /**
     * Creates new form lobby2
     */
    public lobby() {
        this.modelo.addColumn("Nombre");
```



```

this.modelo.addColumn("Hora");
 initComponents();
 setLocationRelativeTo(null);
 this.users.setVisible(false);
 this.tblCitasDlaLobby.setEnabled(false);
 this.btnCitasLobby.setActionCommand("citas");
 this.btnCitasLobby.addActionListener(this);
 this.btnPacientesLobby.setActionCommand("pacientes");
 this.btnPacientesLobby.addActionListener(this);
 this.btnReportesLobby.setActionCommand("reportes");
 this.btnReportesLobby.addActionListener(this);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblCitasDlaLobby = new javax.swing.JTable();
    btnCitasLobby = new javax.swing.JButton();
    btnPacientesLobby = new javax.swing.JButton();
    btnReportesLobby = new javax.swing.JButton();
    btnSalir = new javax.swing.JButton();
    users = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);

    jLabel1.setText("Citas del Dia");

    tblCitasDlaLobby.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

            },
        new String [] {
            "Nombre", "Hora"
        }
    ) {
        Class[] types = new Class [] {
            java.lang.String.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    }

```

```
});
tblCitasDlaLobby.getTableHeader().setReorderingAllowed(false);
jScrollPane1.setViewportViewView(tblCitasDlaLobby);
if (tblCitasDlaLobby.getColumnModel().getColumnCount() > 0) {
    tblCitasDlaLobby.getColumnModel().getColumn(0).setMinWidth(300);
    tblCitasDlaLobby.getColumnModel().getColumn(0).setPreferredWidth(300);
    tblCitasDlaLobby.getColumnModel().getColumn(0).setMaxWidth(300);
    tblCitasDlaLobby.getColumnModel().getColumn(1).setResizable(false);
}

btnCitasLobby.setText("Citas");
btnCitasLobby.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCitasLobbyActionPerformed(evt);
    }
});

btnPacientesLobby.setText("Pacientes");
btnPacientesLobby.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnPacientesLobbyActionPerformed(evt);
    }
});

btnReportesLobby.setText("Reportes");
btnReportesLobby.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnReportesLobbyActionPerformed(evt);
    }
});

btnSalir.setText("Salir");
btnSalir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSalirActionPerformed(evt);
    }
});

users.setText("Usuarios");
users.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        usersActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jLabel1)
        .addContainerGap())
    .addGroup(layout.createSequentialGroup()
        .addComponent(btnCitasLobby)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnPacientesLobby)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnReportesLobby)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnSalir)
        .addContainerGap())
    .addGroup(layout.createSequentialGroup()
        .addComponent(users)
        .addContainerGap())
);
```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(btnPacientesLobby, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btnCitasLobby, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btnReportesLobby, javax.swing.GroupLayout.Alignment.TRAILING,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btnSalir, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
            .addComponent(users, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE))))
        .addContainerGap()
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 275,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(btnCitasLobby)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(btnPacientesLobby)
                    .addGap(18, 18, 18)
                    .addComponent(btnReportesLobby)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(users)
                    .addGap(18, 18, 18)
                    .addComponent(btnSalir)))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

    pack();
} // </editor-fold>

private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void btnCitasLobbyActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaCitas(true);
}

private void btnPacientesLobbyActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaPacientes(true);
}

private void btnReportesLobbyActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaReportes(true);
}

```

```

private void usersActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaus(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(lobby.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(lobby.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(lobby.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(lobby.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new lobby().setVisible(true);
        }
    });
}

public void consulta() {
    if(!"admin".equals(this.superCtrl.getUser())){
        this.users.setVisible(false);
    }else{
        this.users.setVisible(true);
    }
}

try {

    this.modelo.setRowCount(0);
    PreparedStatement st = null;
    ResultSet rt = null;
    ConexionBD con = new ConexionBD();
    Connection cn = con.getConexion();
    String sql = "SELECT nombre_completo(id_paciente) as nombre,to_char(consultas.fecha_hora_pc,'HH:MI:SS') as hora FROM
consultas NATURAL JOIN paciente where TO_CHAR(consultas.fecha_hora_pc,'DD/MM/YY')=(select
TO_CHAR(current_date,'DD/MM/YY') from dual)";
    st = cn.prepareStatement(sql);

```

```

rt = st.executeQuery();

ResultSetMetaData rsMd = rt.getMetaData();
int cantcolumnas = rsMd.getColumnCount();
//System.out.println(rsMd.getColumnCount());
//System.out.println(rsMd.toString());
while (rt.next()) {
    //System.out.println("1");
    Object[] filas = new Object[cantcolumnas];
    for (int i = 0; i < cantcolumnas; i++) {
        filas[i] = rt.getObject(i + 1);
        //System.out.println(filas[i]);
    }
    this.modelo.addRow(filas);
}

this.tblCitasDlaLobby.setModel(this.modelo);
tblCitasDlaLobby.getColumnModel().getColumn(0).setMinWidth(300);
tblCitasDlaLobby.getColumnModel().getColumn(0).setPreferredWidth(300);
tblCitasDlaLobby.getColumnModel().getColumn(0).setMaxWidth(300);
tblCitasDlaLobby.getColumnModel().getColumn(1).setResizable(false);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
}
}

// Variables declaration - do not modify
private javax.swing.JButton btnCitasLobby;
private javax.swing.JButton btnPacientesLobby;
private javax.swing.JButton btnReportesLobby;
private javax.swing.JButton btnSalir;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblCitasDlaLobby;
private javax.swing.JButton users;
// End of variables declaration

@Override
public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

|                |                      |                 |
|----------------|----------------------|-----------------|
| <b>Usuario</b> | <input type="text"/> | <b>Crear</b>    |
| <b>Pass</b>    | <input type="text"/> | <b>Regresar</b> |

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package GUI;

```

```

import Conexion.ConexionBD;
import java.awt.HeadlessException;
import java.awt.event.ActionEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

```

```

/**
 *
 * @author Jiro
 */
public class creaus extends VentanaControladaAbstract {
    private ConexionBD con;
    private PreparedStatement st;
    private ResultSet rt;
    /**
     * Creates new form creaus
     */
    public creaus() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jLabel1.setText("Usuario");

```

```

jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        jTextField1KeyTyped(evt);
    }
});

jLabel2.setText("Pass");

jButton1.setText("Crear");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Regresar");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(34, 34, 34)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jTextField1)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 150, Short.MAX_VALUE))
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addGap(18, 18, 18)
            .addComponent(jButton2)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel2)
            .addGap(34, 34, 34)
            .addComponent(jTextField1)
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addGap(18, 18, 18)
            .addComponent(jButton2)
            .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(34, 34, 34)
            .addComponent(jLabel1)
            .addGap(34, 34, 34)
            .addComponent(jTextField1)
            .addGap(34, 34, 34)
            .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34)
            .addComponent(jButton1)
            .addGap(34, 34, 34)
            .addComponent(jButton2)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel2)
            .addGap(34, 34, 34)
            .addComponent(jTextField1)
            .addGap(34, 34, 34)
            .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34)
            .addComponent(jButton1)
            .addGap(34, 34, 34)
            .addComponent(jButton2)
            .addContainerGap())
);

```

```

    );

    pack();
} // </editor-fold>

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(this(jTextField1.getText().length())>49){evt.consume();}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.con = new ConexionBD();
    Connection cn = this.con.getConnection();
    System.out.println(evt.getActionCommand());
    if(this(jTextField1.getText().length())<1||this(jTextField2.getText().length())<1){
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }else{
        try{
            String sql = "SELECT usuario from usuarios where usuario=? and rownum=1";
            st = cn.prepareStatement(sql);
            st.setString(1, this(jTextField1.getText()));
            rt = st.executeQuery();
            if (rt.next()) {
                if (rt.getString("usuario") != null) {
                    JOptionPane.showMessageDialog(this, "El usuario ya existe");
                    this(jTextField1.setText(null);
                    this(jTextField2.setText(null);
                } else {
                    //JOptionPane.showMessageDialog(this, "El usuario o contraseña son incorrectos");
                    sql = "insert into usuarios(usuario,pass) values(?,?)";
                    (this.st = cn.prepareStatement(sql)).setString(1, this(jTextField1.getText()));
                    this.st.setString(2, this(jTextField2.getText()));
                    this.st.executeUpdate();
                    JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
                    this(jTextField1.setText(null);
                    this(jTextField2.setText(null);
                    this.superCtrl.muestraVentanaus(true);
                }
            }
        }else{
            sql = "insert into usuarios(usuario,pass) values(?,?)";
            (this.st = cn.prepareStatement(sql)).setString(1, this(jTextField1.getText()));
            this.st.setString(2, this(jTextField2.getText()));
            this.st.executeUpdate();
            JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
            this(jTextField1.setText(null);
            this(jTextField2.setText(null);
            this.superCtrl.muestraVentanaus(true);
        }
    }

    }catch(HeadlessException | NumberFormatException | SQLException ex4){
        JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
        JOptionPane.showMessageDialog(this, "Ocurrio un error al crear");
    }
}

```



```

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.jTextField1.setText(null);
        this.jTextField2.setText(null);
        this.superCtrl.muestraVentanaus(true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(creaus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(creaus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(creaus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(creaus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new creaus().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration

    @Override
    public void actionPerformed(ActionEvent e) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}

```

|                                         |                                                                |                                        |
|-----------------------------------------|----------------------------------------------------------------|----------------------------------------|
| <b>Paciente:</b>                        | A A                                                            |                                        |
| <b>Organo dental:</b>                   | <input type="text"/>                                           | ▼                                      |
| <b>Diagnostico:</b>                     | <input type="text"/>                                           |                                        |
| <b>Costo:</b>                           | <input type="text" value="0"/>                                 |                                        |
| <b>Anticipo:</b>                        | <input type="text" value="0"/>                                 |                                        |
| <b>Adeudo:</b>                          | 0                                                              |                                        |
| <b>Fecha de creacion:</b>               | 09/06/2022                                                     |                                        |
| <b>Proxima cita:</b>                    | <input type="text"/>                                           | Ej. 26/01/2022                         |
| <b>Hora:</b>                            | <input type="text" value="9"/> <input type="text" value="00"/> | Hrs.                                   |
| <input type="button" value="Regresar"/> |                                                                | <input type="button" value="Agregar"/> |

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

```
package GUI;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.awt.EventQueue;
import javax.swing.UnsupportedLookAndFeelException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.UIManager;
import java.awt.Toolkit;
import java.sql.Connection;
import java.sql.SQLException;
import java.awt.HeadlessException;
import javax.swing.JOptionPane;
import javax.swing.LayoutStyle;
import java.awt.LayoutManager;
import javax.swing.GroupLayout;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import java.awt.event.KeyListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Component;
import javax.swing.JTextField;
```

```

import javax.swing.JLabel;
import javax.swing.JComboBox;
import javax.swing.JButton;
import java.util.Calendar;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import Conexion.ConexionBD;
import java.util.Date;
/**
 *
 * @author Jiro
 */
public class citas2 extends VentanaControladaAbstract {
    private ConexionBD con;
    private PreparedStatement st;
    private ResultSet rt;
    private int g;
    private int res;
    private int last;
    private int id_c=0;
    private Calendar c = Calendar.getInstance();

    /**
     * Creates new form citas2
     */
    public citas2() {

        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        orgDental = new javax.swing.JComboBox<>();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        diagnostico = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        costo = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        anticipo = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        adeudo = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        fecha_actual = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        hora = new javax.swing.JSpinner();
        jLabel12 = new javax.swing.JLabel();
        proxcita = new javax.swing.JFormattedTextField();

```

```

jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

orgDental.setModel(new DefaultComboBoxModel(new String[] { "N/A", "INCISIVO CENTRAL",
    "INCISIVO LATERAL",
    "CANINO",
    "PRIMER PREMOLAR",
    "SEGUNDO PREMOLAR",
    "PRIMER MOLAR",
    "SEGUNDO MOLAR",
    "TERCER MOLAR"}));
orgDental.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        orgDentalActionPerformed(evt);
    }
});

jLabel1.setText("Organo dental:");

jLabel2.setText("Diagnostico:");

diagnostico.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        diagnosticoActionPerformed(evt);
    }
});
diagnostico.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        diagnosticoKeyTyped(evt);
    }
});

jLabel3.setText("Costo:");

costo.setText("0");
costo.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        costoFocusLost(evt);
    }
});
costo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        costoKeyTyped(evt);
    }
});

jLabel4.setText("Anticipo:");

anticipo.setText("0");
anticipo.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        anticipoFocusLost(evt);
    }
}

```

```

});
anticipo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        anticipoKeyTyped(evt);
    }
});

jLabel5.setText("Adeudo:");

adeudo.setText("0");

jLabel7.setText("Fecha de creacion:");

fecha_actual.setText("22/05/22");

jLabel9.setText("Proxima cita:");

jLabel10.setText("Ej. 26/01/2022");

jLabel11.setText("Hora:");

hora.setModel(new javax.swing.SpinnerNumberModel(9, 9, 17, 1));
hora.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        horaKeyTyped(evt);
    }
});

jLabel12.setText("Hrs.");

proxcita.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.DateFormatter(new
java.text.SimpleDateFormat("dd/MM/yyyy"))));
proxcita.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        proxcitaFocusLost(evt);
    }
});
proxcita.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        proxcitaKeyTyped(evt);
    }
});

jButton1.setText("Regresar");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Crear");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jLabel13.setText("Paciente:");

```

```
jLabel14.setText("Victor Manuel Jimenez Rosas//");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jButton1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton2)
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel11)
                .addComponent(jLabel9)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel4)
                        .addComponent(jLabel5)
                        .addComponent(jLabel7))
                    .addGap(23, 23, 23)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(adeudo)
                        .addComponent(antecipo, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(hora, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jLabel12))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(proximita, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
                                javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(jLabel10))
                        .addComponent(fecha_actual, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
                            javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addComponent(jLabel2)
                        .addComponent(jLabel13)
                        .addComponent(jLabel3))
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(jLabel14)
                                .addGroup(layout.createSequentialGroup()
                                    .addGap(42, 42, 42)
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addComponent(diagnostico, javax.swing.GroupLayout.PREFERRED_SIZE, 188,
                                            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(costo, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(orgDental, javax.swing.GroupLayout.PREFERRED_SIZE, 188,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
        .addGap(83, 83, Short.MAX_VALUE))))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(13, 13, 13)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel13)
                .addComponent(jLabel14))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(orgDental, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(diagnostico, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(costo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(antico, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel5)
                .addComponent(adeudo))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel7)
                .addComponent(fecha_actual))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel9)
                .addComponent(jLabel10))
            .addComponent(proxcita, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel11)
                .addComponent(hora, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel12))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

```

```

);

pack();
} // </editor-fold>

private void orgDentalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void diagnosticoKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (this.diagnostico.getText().length() >=49) {
        evt.consume();
    }
}

private void proxcitaFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
}

private void proxcitaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (Character.isDigit(c)||c=='/') {
        if (this.proxcita.getText().length() < 10) {
            if (this.proxcita.getText().length() == 2&& c!='/') {
                this.proxcita.setText(this.proxcita.getText() + "/");
            } else if (this.proxcita.getText().length() == 5&& c!='/') {
                this.proxcita.setText(this.proxcita.getText() + "/");
            }
        } else if (this.proxcita.getText().length() > 9) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    } else {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.con = new ConexionBD();
    Connection cn = this.con.getConnection();
    System.out.println(evt.getActionCommand());
    if ("agregar".equals(evt.getActionCommand())) {

if((int)this.hora.getValue()<9||(int)this.hora.getValue()>17||this.proxcita.getText().length()<10||this.anticipo.getText().length()<1||this.c
osto.getText().length()<1||this.diagnostico.getText().length()<1||this.orgDental.getSelectedIndex() == 0){
    JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
}else if (Integer.parseInt(this.costo.getText())-Integer.parseInt(this.anticipo.getText())<0){
    JOptionPane.showMessageDialog(this, "El anticipo no puede ser mayor al costo", "Alerta", 2);
}else{
    try{
        String sql ="select org_dental from consultas where fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH')";
        (this.st = cn.prepareStatement(sql)).setString(1,this.proxcita.getText()+" "+(int)this.hora.getValue());
        rt = st.executeQuery();
        if (rt.next()) {

```



```

        if (rt.getString("org_dental") != null) {
            JOptionPane.showMessageDialog(this, "Ya existe una cita en ese dia y hora señalada");
        }else{
            sql = "insert into
consultas(id_paciente,org_dental,fecha_hora_actual,diagnostico,costo,anticipo,adeudo,fecha_hora_pc)
values(?,?,to_date(?, 'DD/MM/YYYY'),?,?,?,to_date(?, 'DD/MM/YYYY HH'))";
            (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
            this.st.setString(2, this.orgDental.getSelectedItem().toString());
            this.st.setString(3, this.fecha_actual.getText());
            this.st.setString(4, this.diagnostico.getText());
            this.st.setInt(5, Integer.parseInt(this.costo.getText()));
            this.st.setInt(6, Integer.parseInt(this.anticipo.getText()));
            this.st.setInt(7, Integer.parseInt(this.adeudo.getText()));
            this.st.setString(8, this.proxcita.getText()+" "+(int)this.hora.getValue());
            this.st.executeUpdate();
            JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
            limpiar();
            this.superCtrl.muestraVentanaCitas(true);
        }
    }else{
        sql = "insert into
consultas(id_paciente,org_dental,fecha_hora_actual,diagnostico,costo,anticipo,adeudo,fecha_hora_pc)
values(?,?,to_date(?, 'DD/MM/YYYY'),?,?,?,to_date(?, 'DD/MM/YYYY HH'))";
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.st.setString(2, this.orgDental.getSelectedItem().toString());
        this.st.setString(3, this.fecha_actual.getText());
        this.st.setString(4, this.diagnostico.getText());
        this.st.setInt(5, Integer.parseInt(this.costo.getText()));
        this.st.setInt(6, Integer.parseInt(this.anticipo.getText()));
        this.st.setInt(7, Integer.parseInt(this.adeudo.getText()));
        this.st.setString(8, this.proxcita.getText()+" "+(int)this.hora.getValue());
        this.st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Se creo", "Atenci\u00f3n", 1);
        limpiar();
        this.superCtrl.muestraVentanaCitas(true);
    }
}

}catch(HeadlessException | NumberFormatException | SQLException ex4){
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
    JOptionPane.showMessageDialog(this, "Ocurrio un error al crear");
}
}

}

}else if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea modificar?\nESTA ACCION NO SE PUEDE
DESHACER", "Alerta!", 0, 2) == 0) {

if((int)this.hora.getValue()<9||(int)this.hora.getValue()>17||this.proxcita.getText().length()<10||this.anticipo.getText().length()<1||this.c
osto.getText().length()<1||this.diagnostico.getText().length()<1||this.orgDental.getSelectedIndex() == 0){
    JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
}else if (Integer.parseInt(this.costo.getText())-Integer.parseInt(this.anticipo.getText())<0){
    JOptionPane.showMessageDialog(this, "El anticipo no puede ser mayor al costo", "Alerta", 2);
}else{
    try{
        String sql ="select id_paciente from consultas where fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH')";
        (this.st = cn.prepareStatement(sql)).setString(1, this.proxcita.getText()+" "+(int)this.hora.getValue());
        rt = st.executeQuery();
        if (rt.next()) {
            if ( !this.superCtrl.getClave().equals(rt.getInt("id_paciente")+"")) {

```

```

        JOptionPane.showMessageDialog(this, "Ya existe una cita en ese día y hora señalada");
    }else{
        sql = "UPDATE consultas SET
org_dental=?,diagnostico=?,costo=?,anticipo=?,adeudo=?,fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH') where id_consulta=?";
        (this.st = cn.prepareStatement(sql)).setString(1, this.orgDental.getSelectedItem().toString());
        this.st.setString(2, this.diagnostico.getText());
        this.st.setInt(3, Integer.parseInt(this.costo.getText()));
        this.st.setInt(4, Integer.parseInt(this.anticipo.getText()));
        this.st.setInt(5, Integer.parseInt(this.adeudo.getText()));
        this.st.setString(6, this.prox cita.getText() + " " + (int) this.hora.getValue());
        this.st.setString(7, this.id_c + "");
        this.st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Se modifico", "Atenci\u00f3n", 1);
        limpiar();
        this.superCtrl.muestraVentanaCitas(true);
    }
}
}else{
    sql = "UPDATE consultas SET
org_dental=?,diagnostico=?,costo=?,anticipo=?,adeudo=?,fecha_hora_pc=to_date(?, 'DD/MM/YYYY HH') where id_consulta=?";
    (this.st = cn.prepareStatement(sql)).setString(1, this.orgDental.getSelectedItem().toString());
    this.st.setString(2, this.diagnostico.getText());
    this.st.setInt(3, Integer.parseInt(this.costo.getText()));
    this.st.setInt(4, Integer.parseInt(this.anticipo.getText()));
    this.st.setInt(5, Integer.parseInt(this.adeudo.getText()));
    this.st.setString(6, this.prox cita.getText() + " " + (int) this.hora.getValue());
    this.st.setString(7, this.id_c + "");
    this.st.executeUpdate();
    JOptionPane.showMessageDialog(this, "Se modifico", "Atenci\u00f3n", 1);
    limpiar();
    this.superCtrl.muestraVentanaCitas(true);
}

}catch(HeadlessException | NumberFormatException | SQLException ex4){
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
    JOptionPane.showMessageDialog(this, "ocurrio un error al modificar");
}
}
}
}

private void anticipoFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (this.anticipo.getText().length() < 1) {
        this.anticipo.setText("0");
    }else{
        if (Integer.parseInt(this.costo.getText()) - Integer.parseInt(this.anticipo.getText()) <= 0){
            this.adeudo.setText("0");
        }else{
            this.adeudo.setText(Integer.parseInt(this.costo.getText()) - Integer.parseInt(this.anticipo.getText()) + "");
        }
    }
}

private void costoKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();

```

```

        if (!Character.isDigit(c)) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    }

    private void anticipoKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char c = evt.getKeyChar();
        if (!Character.isDigit(c)) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    }

    private void costoFocusLost(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        if (this.costo.getText().length() < 1) {
            this.costo.setText("0");
        } else {
            if (Integer.parseInt(this.costo.getText()) - Integer.parseInt(this.anticipo.getText()) <= 0) {
                this.adeudo.setText("0");
            } else {
                this.adeudo.setText(Integer.parseInt(this.costo.getText()) - Integer.parseInt(this.anticipo.getText()) + "");
            }
        }
    }

    private void horaKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char c = evt.getKeyChar();
        if (!Character.isDigit(c)) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        limpiar();
        this.superCtrl.muestraVentanaCitas(true);
    }

    private void diagnosticoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

```

```

        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(citas2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(citas2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(citas2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(citas2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
/*java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new citas2().setVisible(true);
    }
});*/
try {
    for (final UIManager.LookAndFeelInfo info : UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    Logger.getLogger(citas2.class.getName()).log(Level.SEVERE, null, ex);
}
} catch (InstantiationException ex2) {
    Logger.getLogger(citas2.class.getName()).log(Level.SEVERE, null, ex2);
}
} catch (IllegalAccessException ex3) {
    Logger.getLogger(citas2.class.getName()).log(Level.SEVERE, null, ex3);
}
} catch (UnsupportedLookAndFeelException ex4) {
    Logger.getLogger(citas2.class.getName()).log(Level.SEVERE, null, ex4);
}
}
EventQueue.invokeLater(new Runnable() {
    @Override
    public void run() {
        new citas2().setVisible(true);
    }
});
}
@Override
public void actionPerformed(ActionEvent e) {}

// Variables declaration - do not modify
private javax.swing.JLabel adeudo;
private javax.swing.JTextField anticipo;
private javax.swing.JTextField costo;
private javax.swing.JTextField diagnostico;
private javax.swing.JLabel fecha_actual;

```

```

private javax.swing.JSpinner hora;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel9;
private javax.swing.JComboBox<String> orgDental;
private javax.swing.JFormattedTextField proxcita;
// End of variables declaration

public void modificar() {
    limpiar();
    this.jButton2.setText("Modificar");
    this.jButton2.setActionCommand("modificar");
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConnection();
        String sql = "SELECT nombre_completo(id_paciente) as nombre from paciente where id_paciente=? and rownum=1 order
by id_paciente DESC";
        //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.jLabel14.setText(this.rt.getString("nombre"));
        }
        sql = "SELECT id_consulta,id_paciente,org_dental,to_char(consultas.fecha_hora_actual,'dd/MM/yyyy') as
fecha_actual,diagnostico,costo,anticipo,adeudo,to_char(consultas.fecha_hora_pc,'dd/MM/yyyy') as
fecha_pc,to_char(consultas.fecha_hora_pc,'HH') as hora_pc from consultas where id_paciente=? and rownum=1 order by
id_paciente DESC";
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.id_c=this.rt.getInt("id_consulta");
            this.superCtrl.setClave(this.rt.getString("id_paciente"));
            this.orgDental.setSelectedItem(this.rt.getString("org_dental"));
            this.diagnostico.setText(this.rt.getString("diagnostico"));
            this.costo.setText(this.rt.getInt("costo")+ "");
            this.anticipo.setText(this.rt.getInt("anticipo")+ "");
            this.adeudo.setText(this.rt.getInt("adeudo")+ "");
            this.fecha_actual.setText(this.rt.getString("fecha_actual")+ "");
            this.proxcita.setText(this.rt.getString("fecha_pc"));
            this.hora.setValue(Integer.parseInt(this.rt.getString("hora_pc")));
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
        e.printStackTrace();
    }
}

```

```

    }
    private boolean validafecha(final String fecha) {
        String[] fecha2 = fecha.split("/");
        String dia = fecha2[0];
        String mes = fecha2[1];
        String annio = fecha2[2];
        System.out.println(dia);
        System.out.println(mes);
        System.out.println(annio);
        final SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
        try {
            formatoFecha.setLenient(false);
            formatoFecha.parse(dia + "/" + mes + "/" + annio);
            System.out.println("validafecha: fecha valida");
            return true;
        } catch (ParseException e) {
            System.out.println("validafecha: error al parcear fecha");
            return false;
        }
    }
    public void agregar() {
        limpiar();
        Date now = new Date(System.currentTimeMillis());
        SimpleDateFormat date = new SimpleDateFormat("dd/MM/yyyy");
        SimpleDateFormat hour = new SimpleDateFormat("HH:mm:ss");
        this.fecha_actual.setText(date.format(now));
        try {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConexion();
            final String sql = "SELECT nombre_completo(id_paciente) as nombre from paciente where id_paciente=? and rownum=1
order by id_paciente DESC";
            //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
            (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
            this.rt = this.st.executeQuery();
            if (this.rt.next()) {
                this.jLabel14.setText(this.rt.getString("nombre"));
            }
        }
        catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
        }
        this.jButton2.setText("Agregar");
        this.jButton2.setActionCommand("agregar");
    }
    private void limpiar(){
        this.jLabel14.setText(null);
        this.orgDental.setSelectedItem(null);
        this.diagnostico.setText(null);
        this.costeo.setText("0");
        this.anticipo.setText("0");
        this.adeudo.setText("0");
        this.fecha_actual.setText(null);
        this.proxcita.setText(null);
        this.hora.setValue(9);
    }
}

```

/\*

\* Click <nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt> to change this license

\* Click <nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java> to edit this template

\*/

```
package GUI;
```

```
import java.awt.event.ActionEvent;
import Conexion.ConexionBD;
import java.awt.Component;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSetMetaData;
import javax.swing.JMenuItem;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
```

/\*\*

\*

\* @author Jiro

\*/

```
public class busquedaCita1 extends VentanaControladaAbstract {
```

```

/**
 * Creates new form busquedaCita
 */
ConexionBD con;
private PreparedStatement st=null;
private ResultSet rt;
private DefaultTableModel modelo = new DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "ID", "Paciente"
        }
    ) {
        Class[] types = new Class [] {
            java.lang.Integer.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        @Override
        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        @Override
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    };
JPopupMenu popup;
public busquedaCita1() {
    initComponents();
    setLocationRelativeTo(null);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    popup = new JPopupMenu();
    JMenuItem menuItem3 = new JMenuItem("Crear");
    menuItem3.addActionListener(this);
    popup.add( menuItem3 );

    jTable1.addMouseListener( new MouseAdapter(){
        public void mousePressed(MouseEvent e)
        {
            //System.out.println("pressed");
        }

        public void mouseReleased(MouseEvent e)
        {
            if (e.isPopupTrigger())
            {
                JTable source = (JTable)e.getSource();
                int row = source.rowAtPoint( e.getPoint() );
                int column = source.columnAtPoint( e.getPoint() );
            }
        }
    });
}

```



```

        if (! source.isRowSelected(row))
            source.changeSelection(row, column, false, false);

        popup.show(e.getComponent(), e.getX(), e.getY());
    }
}
});
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton2 = new javax.swing.JButton();
    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);

    jLabel1.setText("Buscar por nombre");

    jTable1.setModel(this.modelo);
    jTable1.setColumnSelectionAllowed(true);
    jTable1.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(jTable1);

    jButton2.setText("Regresar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jTextField1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField1ActionPerformed(evt);
        }
    });
    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            jTextField1KeyTyped(evt);
        }
    });

    jButton1.setText("Buscar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

```

```

    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane1)
                    .addGroup(layout.createSequentialGroup()
                        .add(layout.createParallelGroup()
                            .addComponent(jLabel1)
                            .addGap(18, 18, 18)
                            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(jButton1)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 262, Short.MAX_VALUE)
                            .addComponent(jButton2)))
                        .addContainerGap())
                    .addGroup(layout.createSequentialGroup()
                        .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .add(layout.createParallelGroup()
                                .add(layout.createParallelGroup()
                                    .addContainerGap()
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addComponent(jButton1)
                                        .addComponent(jLabel1)
                                        .addComponent(jButton2))
                                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                .addContainerGap())
                            .addContainerGap())
                        .addContainerGap())
                    .addContainerGap())
                .addContainerGap())
            .addContainerGap())
        .addContainerGap()
);

pack();
} // </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.limpiar();
    this.superCtrl.muestraVentanaCitas(true);
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingresa solo letras", "Atencion", 1);
    }
}

```

```

    }else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

        final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente where upper(nombre) LIKE
upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";

```

```

        this.con = new ConexionBD();
        Connection cn = this.con.getConnection();
        st = null;
        ResultSet rt = null;

```

```

        /**
        try {
            //System.out.println("1");
            this.modelo.setRowCount(0);
            //System.out.println("2");
            st = cn.prepareStatement(sql);
            //System.out.println("3");
            st.setString(1, "%" + this.jTextField1.getText() + "%");
            //System.out.println("4");
            st.setString(2, "%" + this.jTextField1.getText() + "%");
            //System.out.println("5");
            //System.out.println(st.getParameterMetaData());
            rt = st.executeQuery();
            //System.out.println("6");
            ResultSetMetaData rsMd = rt.getMetaData();
            //System.out.println("7");
            int cantcolumnas = rsMd.getColumnCount();
            //System.out.println("8");
            //System.out.println(rsMd.getColumnCount());
            //System.out.println(rsMd.toString());
            //System.out.println(cantcolumnas);
            //System.out.println("aqui");
            int nfilas=0;
            while (rt.next()) {
                //System.out.println("1");
                Object[] filas = new Object[cantcolumnas];
                for (int i = 0; i < cantcolumnas; i++) {
                    filas[i] = rt.getObject(i + 1);
                    nfilas++;
                    //System.out.println(filas[i]);
                }
                //System.out.println("aqui2");
                this.modelo.addRow(filas);
            }
            if (nfilas>0){
                jTable1.changeSelection(0, 1, false, false);
            }
            //this.jTable1.setModel(this.modelo);
            jTable1.getColumnModel().getColumn(0).setMinWidth(0);
            jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
            jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

            jTable1.getColumnModel().getColumn(1).setWidthToFit();

```

```

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

    /**/
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(busquedaCita1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(busquedaCita1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(busquedaCita1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(busquedaCita1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new busquedaCita1().setVisible(true);
        }
    });
}

@Override
public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    if (e.getActionCommand().equals("Crear")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();

```

```

//System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
System.out.println(dato);
this.superCtrl.setClave(dato);
int antes = 0;
try {
    final Connection cn = this.con.getConexion();
    this.st = cn.prepareStatement("select count(id_paciente) as conteo from consultas where id_paciente=?");
    this.st.setString(1, this.superCtrl.getClave());
    this.rt = this.st.executeQuery();
    if (this.rt.next()) {
        antes = this.rt.getInt("conteo");
    }
    if (antes != 0) {
        JOptionPane.showMessageDialog(this, "Ya existe una cita para este paciente, modifiquela o eliminela", "Atencion", 1);
    } else if (antes == 0) {
        this.limpiar();
        this.superCtrl.muestracita(true);
    }
} catch (HeadlessException | NumberFormatException | SQLException ex4) {
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
}

}
}

private void limpiar() {
    this.modelo.setRowCount(0);
    this.jTextField1.setText(null);
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

$\ell^*$ 

\* Click <nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt> to change this license

\* Click <nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java> to edit this template

\*/

```
package GUI;
```

```
import java.awt.event.ActionEvent;
import Conexion.ConexionBD;
import java.awt.Component;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSetMetaData;
import javax.swing.JMenuItem;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
```

/\*\*

\*

\* @author Jiro

\* /

```
public class busquedaCita extends VentanaControladaAbstract {
```

```

/**
 * Creates new form busquedaCita
 */
ConexionBD con;
private PreparedStatement st=null;
private ResultSet rt;
private DefaultTableModel modelo = new DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "ID", "Paciente"
        }
    ) {
        Class[] types = new Class [] {
            java.lang.Integer.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        @Override
        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        @Override
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    };
JPopupMenu popup;
public busquedaCita() {
    initComponents();
    setLocationRelativeTo(null);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    popup = new JPopupMenu();
    JMenuItem menuItem = new JMenuItem("Modificar");
    menuItem.addActionListener(this);
    popup.add( menuItem );
    JMenuItem menuItem2 = new JMenuItem("Eliminar");
    menuItem2.addActionListener(this);
    popup.add( menuItem2 );

    jTable1.addMouseListener( new MouseAdapter(){
        public void mousePressed(MouseEvent e)
        {
            //System.out.println("pressed");
        }

        public void mouseReleased(MouseEvent e)
        {
            if (e.isPopupTrigger())
            {

```

```

        jTable.source = (JTable)e.getSource();
        int row = source.rowAtPoint( e.getPoint() );
        int column = source.columnAtPoint( e.getPoint() );

        if (! source.isRowSelected(row))
            source.changeSelection(row, column, false, false);

        popup.show(e.getComponent(), e.getX(), e.getY());
    }
}
});
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton2 = new javax.swing.JButton();
    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);

    jLabel1.setText("Buscar por nombre");

    jTable1.setModel(this.modelo);
    jTable1.setColumnSelectionAllowed(true);
    jTable1.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(jTable1);

    jButton2.setText("Regresar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jTextField1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField1ActionPerformed(evt);
        }
    });
    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            jTextField1KeyTyped(evt);
        }
    });

    jButton1.setText("Buscar");

```



```

jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton2)))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton3)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton3))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

pack();
} // </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.limpiar();
    this.superCtrl.muestraVentanaCitas(true);
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {

```

```

        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras", "Atencion", 1);
    }else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente natural join consultas where
upper(nombre) LIKE upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";
    this.con = new ConexionBD();
    Connection cn = this.con.getConnection();
    st = null;
    ResultSet rt = null;

    /**
    try {
        //System.out.println("1");
        this.modelo.setRowCount(0);
        //System.out.println("2");
        st = cn.prepareStatement(sql);
        //System.out.println("3");
        st.setString(1, "%" + this.jTextField1.getText() + "%");
        //System.out.println("4");
        st.setString(2, "%" + this.jTextField1.getText() + "%");
        //System.out.println("5");
        //System.out.println(st.getParameterMetaData());
        rt = st.executeQuery();
        //System.out.println("6");
        ResultSetMetaData rsMd = rt.getMetaData();
        //System.out.println("7");
        int cantcolumnas = rsMd.getColumnCount();
        //System.out.println("8");
        //System.out.println(rsMd.getColumnCount());
        //System.out.println(rsMd.toString());
        //System.out.println(cantcolumnas);
        //System.out.println("aqui");
        int nfilas=0;
        while (rt.next()) {
            //System.out.println("1");
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
                nfilas++;
                //System.out.println(filas[i]);
            }
            //System.out.println("aqui2");
            this.modelo.addRow(filas);
        }
        if (nfilas>0){
            jTable1.changeSelection(0, 1, false, false);
        }
        //this.jTable1.setModel(this.modelo);
        jTable1.getColumnModel().getColumn(0).setMinWidth(0);
        jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);

```

```

jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

jTable1.getColumnModel().getColumn(1).setWidthToFit();

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

/**
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(busquedaCita.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(busquedaCita.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(busquedaCita.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(busquedaCita.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new busquedaCita().setVisible(true);
    }
});
}

@Override
public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    if (e.getActionCommand().equals("Modificar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();

```

```

JPopupMenu popup = (JPopupMenu) c.getParent();
JTable table = (JTable) popup.getInvoker();
//System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
System.out.println(dato);

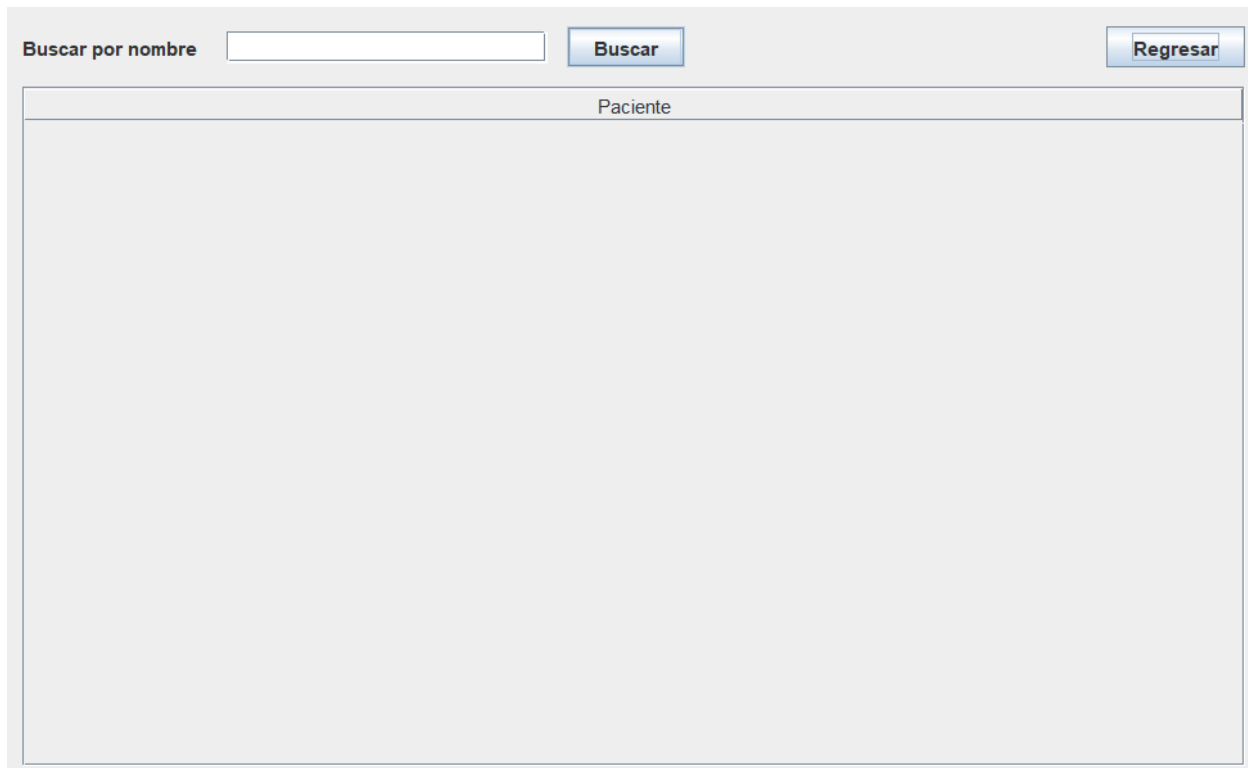
//this.superCtrl.muestraVentanaModificarPacientes(true);
this.superCtrl.setClave(dato);
this.limpiar();
//this.superCtrl.muestraVentanaCitas(false);
this.superCtrl.muestraVentanaModificarCita(true);

} else if (e.getActionCommand().equals("Eliminar")) {
    System.out.println(e.getActionCommand());
    Component c = (Component) e.getSource();
    JPopupMenu popup = (JPopupMenu) c.getParent();
    JTable table = (JTable) popup.getInvoker();
    //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
    String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
    System.out.println(dato);
    this.superCtrl.setClave(dato);
    if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar la cita?\nESTA ACCION NO SE PUEDE
DESHACER", "Alerta!", 0, 2) == 0) {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConexion();
        String sql = "DELETE FROM consultas WHERE id_paciente=?";
        try {
            (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
            this.st.executeQuery();
            cn.close();
            this.con.desconectar();
            this.st.close();
            this.limpiar();
            JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
            //this.superCtrl.muestraVentanaPacientes(true);
        } catch (HeadlessException | NumberFormatException | SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
        }
    }
}

private void limpiar() {
    this.modelo.setRowCount(0);
    this.jTextField1.setText(null);
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```



```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package GUI;

```

```

import java.awt.event.ActionEvent;
import Conexion.ConexionBD;
import java.awt.Component;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSetMetaData;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
/**
 *
 * @author Jiro
 */
public class buscarus extends VentanaControladaAbstract {

```

```

/**
 * Creates new form busquedaCita
 */
ConexionBD con;
private PreparedStatement st=null;
private ResultSet rt;
private DefaultTableModel modelo = new DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "ID", "usuario"
        }
    ) {
        Class[] types = new Class [] {
            java.lang.Integer.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        @Override
        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        @Override
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    };
JPopupMenu popup;
public buscarus() {
    initComponents();
    setLocationRelativeTo(null);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    popup = new JPopupMenu();
    JMenuItem menuItem2 = new JMenuItem("Eliminar");
    menuItem2.addActionListener(this);
    popup.add( menuItem2 );

    jTable1.addMouseListener( new MouseAdapter(){
        public void mousePressed(MouseEvent e)
        {
            //System.out.println("pressed");
        }

        public void mouseReleased(MouseEvent e)
        {
            if (e.isPopupTrigger())
            {
                JTable source = (JTable)e.getSource();
                int row = source.rowAtPoint( e.getPoint() );
                int column = source.columnAtPoint( e.getPoint() );
            }
        }
    });
}

```

```

        if (! source.isRowSelected(row))
            source.changeSelection(row, column, false, false);

        popup.show(e.getComponent(), e.getX(), e.getY());
    }
}
});
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton2 = new javax.swing.JButton();
    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);

    jLabel1.setText("Buscar por nombre");

    jTable1.setModel(this.modelo);
    jTable1.setColumnSelectionAllowed(true);
    jTable1.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(jTable1);

    jButton2.setText("Regresar");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jTextField1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField1ActionPerformed(evt);
        }
    });
    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            jTextField1KeyTyped(evt);
        }
    });

    jButton1.setText("Buscar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

```

```

    }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 400, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(10, 10, 10)
                        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(jButton1)
                        .addGap(10, 10, 10)
                        .addComponent(jButton2)))
                .addContainerGap(10, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(10, 10, 10)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1)
                    .addComponent(jLabel1)
                    .addComponent(jButton2))
                .addGap(10, 10, 10)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(10, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.limpiar();
    //this.superCtrl.muestraVentanaCitas(true);
    this.superCtrl.muestraVentanaAus(true);
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

```



```

        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras", "Atencion", 1);
    }else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    final String sql = "SELECT id,usuario from usuarios where usuario LIKE ? and id!=1 and usuario!=? order by id DESC";
    this.con = new ConexionBD();
    Connection cn = this.con.getConexion();
    st = null;
    ResultSet rt = null;

    /**
    try {
        //System.out.println("1");
        this.modelo.setRowCount(0);
        //System.out.println("2");
        st = cn.prepareStatement(sql);
        //System.out.println("3");
        st.setString(1, "%" + this.jTextField1.getText() + "%");
        st.setString(2, this.superCtrl.getUser());
        //System.out.println("5");
        //System.out.println(st.getParameterMetaData());
        rt = st.executeQuery();
        //System.out.println("6");
        ResultSetMetaData rsMd = rt.getMetaData();
        //System.out.println("7");
        int cantcolumnas = rsMd.getColumnCount();
        //System.out.println("8");
        //System.out.println(rsMd.getColumnCount());
        //System.out.println(rsMd.toString());
        //System.out.println(cantcolumnas);
        //System.out.println("aqui");
        int nfilas=0;
        while (rt.next()) {
            //System.out.println("1");
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
                nfilas++;
                //System.out.println(filas[i]);
            }
            //System.out.println("aqui2");
            this.modelo.addRow(filas);
        }
        if (nfilas>0){
            jTable1.changeSelection(0, 1, false, false);
        }
        //this.jTable1.setModel(this.modelo);
        jTable1.getColumnModel().getColumn(0).setMinWidth(0);
        jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
        jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

        jTable1.getColumnModel().getColumn(1).setWidthToFit();
    }
    */
}

```

```

jTable1.getColumnModel().getColumn(1).setResizable(false);
//jTable1.getColumnModel().getColumn(2).setResizable(false);
//jTable1.setEnabled(false);

    /**/
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    e.printStackTrace();
}
/**/
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(buscarus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(buscarus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(buscarus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(buscarus.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new buscarus().setVisible(true);
        }
    });
}

@Override
public void actionPerformed(ActionEvent e) {
    //throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    if (e.getActionCommand().equals("Eliminar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
    }
}

```

```

        String dato = String.valueOf(this.modelo.getValueAt(table.getSelectedRow(), 0));
        System.out.println(dato);
        if (!"1".equals(dato)){
            if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar ?\nESTA ACCION NO SE PUEDE DESHACER", "Alerta!", 0, 2) == 0) {
                this.con = new ConexionBD();
                final Connection cn = this.con.getConexion();
                String sql = "DELETE FROM usuarios WHERE id=?";
                try {
                    (this.st = cn.prepareStatement(sql)).setString(1, dato);
                    this.st.executeQuery();
                    cn.close();
                    this.con.desconectar();
                    this.st.close();
                    this.limpiar();
                    JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
                    //this.superCtrl.muestraVentanaPacientes(true);
                } catch (HeadlessException | NumberFormatException | SQLException ex) {
                    JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
                }
            }
        }
    }

    private void limpiar() {
        this.modelo.setRowCount(0);
        this.jTextField1.setText(null);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}

```

- \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
- \* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
- \* /

```
package GUI;
```

```
import ConexionBD;
import java.awt.Component;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import java.sql.ResultSetMetaData;
import javax.swing.JMenuItems;
import javax.swing.JPopupMenu;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import java.sql.CallableStatement;
/**
 *
 * @author Jiro
 */
public class buscar extends VentanaControladaAbstract {
    ConexionBD con;
```

```

PreparedStatement st = null;
private DefaultTableModel modelo = new DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "ID", "Paciente"
        }
    ){
        Class[] types = new Class [] {
            java.lang.Integer.class, java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false
        };

        @Override
        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        @Override
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    };
JPopupMenu popup;

/**
 * Creates new form buscar
 */
public buscar() {
    //this.modelo.addColumn("Nombre");
    //this.modelo.addColumn("Apellido");
    initComponents();
    setLocationRelativeTo(null);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    popup = new JPopupMenu();
    JMenuItem menuItem = new JMenuItem("Modificar");
    menuItem.addActionListener(this);
    popup.add( menuItem );
    JMenuItem menuItem2 = new JMenuItem("Eliminar");
    menuItem2.addActionListener(this);
    popup.add( menuItem2 );

    jTable1.addMouseListener( new MouseAdapter(){
        public void mousePressed(MouseEvent e)
        {
            //System.out.println("pressed");
        }

        public void mouseReleased(MouseEvent e)
        {
            if (e.isPopupTrigger())

```

```

        {
            JTable source = (JTable)e.getSource();
            int row = source.rowAtPoint( e.getPoint() );
            int column = source.columnAtPoint( e.getPoint() );

            if (! source.isRowSelected(row))
                source.changeSelection(row, column, false, false);

            popup.show(e.getComponent(), e.getX(), e.getY());
        }
    }
    /**jTable1.setPreferredScrollableViewportSize(jTable1.getPreferredSize());
    getContentPane().add( new JScrollPane(jTable1) );/**/

}
@Override
public void actionPerformed(ActionEvent e) {

    if (e.getActionCommand().equals("Modificar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato=String.valueOf(this.modelo.getValueAt(table.getSelectedRow(),0));
        System.out.println(dato);

        //this.superCtrl.muestraVentanaModificarPacientes(true);
        this.superCtrl.setClave(dato);
        this.limpiar();
        this.superCtrl.muestraVentanaModificarPacientes(false);

    } else if (e.getActionCommand().equals("Eliminar")) {
        System.out.println(e.getActionCommand());
        Component c = (Component) e.getSource();
        JPopupMenu popup = (JPopupMenu) c.getParent();
        JTable table = (JTable) popup.getInvoker();
        //System.out.println(table.getSelectedRow() + " : " + table.getSelectedColumn());
        String dato=String.valueOf(this.modelo.getValueAt(table.getSelectedRow(),0));
        System.out.println(dato);
        this.superCtrl.setClave(dato);
        if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea eliminar al paciente?\nSe eliminaran todos los registros
donde aparezca este paciente\nESTA ACCION NO SE PUEDE DESHACER", "Alerta!", 0, 2) == 0) {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConexion();
            //String sql = "DELETE FROM consultas WHERE id_paciente=?";
            String sql="{call paquetepaciente.delpaciente(?)}";

            try {
                CallableStatement statement = cn.prepareCall(sql);
                statement.setString(1, this.superCtrl.getClave());
                boolean f;
                f=statement.execute();
                System.out.println(f);
                //proc.setString(1, "lisa");
                //cs.execute();
            }
        }
    }
}

```

```

        //(this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        //this.st.executeQuery();
        /*sql = "DELETE FROM cuestionario WHERE id_paciente=?";
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.st.executeUpdate();
        sql = "DELETE FROM paciente WHERE id_paciente=? ";
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.st.executeUpdate();*/
        cn.close();
        this.con.desconectar();
        this.st.close();
        this.limpiar();
        JOptionPane.showMessageDialog(this, "Se ha eliminado con exito!", "Atencion", 1);
        //this.superCtrl.muestraVentanaPacientes(true);
    }
    catch (HeadlessException | NumberFormatException | SQLException ex) {
        JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", 0);
    }
}

}

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);

    jTextField1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField1ActionPerformed(evt);
        }
    });
    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            jTextField1KeyTyped(evt);
        }
    });

    jButton1.setText("Buscar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
}

```

```

jLabel1.setText("Buscar por nombre");

jTable1.setModel(this.modelo);
jTable1.setColumnSelectionAllowed(true);
jTable1.getTableHeader().setReorderingAllowed(false);
jScrollPane1.setViewportView(jTable1);
jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);

jButton2.setText("Regresar");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jButton1)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jButton2)
                            .addGap(18, 18, 18))
                        .addComponent(jLabel1)
                    )
                )
            .addGap(18, 18, 18)
        )
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jButton1)
                .addComponent(jButton2)
                .addComponent(jLabel1)
            )
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
        )
    );

pack();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```



```

        final String sql = "SELECT id_paciente,nombre_completo(id_paciente) as nombre from paciente where upper(nombre) LIKE
upper(?) or upper(apellidos) LIKE upper(?) order by id_paciente DESC";
        this.con = new ConexionBD();
        Connection cn = this.con.getConnection();
        st = null;
        ResultSet rt = null;

/**
try {
    //System.out.println("1");
    this.modelo.setRowCount(0);
    //System.out.println("2");
    st = cn.prepareStatement(sql);
    //System.out.println("3");
    st.setString(1, "%" + this.jTextField1.getText() + "%");
    //System.out.println("4");
    st.setString(2, "%" + this.jTextField1.getText() + "%");
    //System.out.println("5");
    //System.out.println(st.getParameterMetaData());
    rt = st.executeQuery();
    //System.out.println("6");
    ResultSetMetaData rsMd = rt.getMetaData();
    //System.out.println("7");
    int cantcolumnas = rsMd.getColumnCount();
    //System.out.println("8");
    //System.out.println(rsMd.getColumnCount());
    //System.out.println(rsMd.toString());
    //System.out.println(cantcolumnas);
    //System.out.println("aqui");
    int nfilas=0;
    while (rt.next()) {
        //System.out.println("1");
        Object[] filas = new Object[cantcolumnas];
        for (int i = 0; i < cantcolumnas; i++) {
            filas[i] = rt.getObject(i + 1);
            nfilas++;
            //System.out.println(filas[i]);
        }
        //System.out.println("aqui2");
        this.modelo.addRow(filas);
    }
    if (nfilas>0){
        jTable1.changeSelection(0, 1, false, false);
    }
    //this.jTable1.setModel(this.modelo);
    jTable1.getColumnModel().getColumn(0).setMinWidth(0);
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(0);
    jTable1.getColumnModel().getColumn(0).setMaxWidth(0);

    jTable1.getColumnModel().getColumn(1).setWidthToFit();

    jTable1.getColumnModel().getColumn(1).setResizable(false);
    //jTable1.getColumnModel().getColumn(2).setResizable(false);
    //jTable1.setEnabled(false);

```

```

    /**/
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
        e.printStackTrace();
    }
    /**/
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingresa solo letras", "Atenci\u2666n", 1);
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.limpiar();
    this.superCtrl.muestraVentanaPacientes(true);
}

private void limpiar() {
    this.modelo.setRowCount(0);
    this(jTextField1.setText(null);

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(buscar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(buscar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(buscar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(buscar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

```
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new buscar().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}
```

Ingrese intervalo de fecha (dd/mm/yyyy)

entre fecha:
y fecha:

| Nombre         | Cita                  |
|----------------|-----------------------|
| victor jimenez | 2022-06-09 09:00:00.0 |
| A A            | 2022-06-09 10:00:00.0 |

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package GUI;
//import Conexion.Conexion;
import Conexion.ConexionBD;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import javax.swing.GroupLayout;

```

```

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.LayoutStyle;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumnModel;
/**
 *
 * @author Jiro
 */
public class VentanaTablaReportes extends VentanaControladaAbstract {

    /**
     * Creates new form VentanaTablaReportes
     */
    private DefaultTableModel modelo;
    Calendar c;
    public VentanaTablaReportes() {
        initComponents();
        this.modelo = new DefaultTableModel();
        this.c = Calendar.getInstance();
        this.modelo.setRowCount(0);
        this.jTable1.setEnabled(false);
        setLocationRelativeTo(null);
        this.btnBuscar.setVisible(false);
    }

    this.lbl1.setVisible(false);
    this.lbl2.setVisible(false);
    this.lbl3.setVisible(false);
    this.lbl4.setVisible(false);
    this.lbl5.setVisible(false);
    this.lbl6.setVisible(false);
    this.txtfd1.setVisible(false);
    this.txtfd2.setVisible(false);
    this.txtfm1.setVisible(false);
    this.txtfm2.setVisible(false);
    this.txtfy1.setVisible(false);
    this.txtfy2.setVisible(false);

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        fechacompleta1 = new javax.swing.JFormattedTextField();
        jLabel2 = new javax.swing.JLabel();

```

```

jLabel3 = new javax.swing.JLabel();
fechacompleta2 = new javax.swing.JFormattedTextField();
btnBuscar = new javax.swing.JButton();
btnVolver = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

jLabel1.setText("Ingrese intervalo de fecha (dd/mm/yyyy)");

fechacompleta1.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.DateFormatter(new
java.text.SimpleDateFormat("dd/MM/yyyy"))));
fechacompleta1.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        fechacompleta1KeyTyped(evt);
    }
});

jLabel2.setText("entre fecha:");

jLabel3.setText("y fecha:");

fechacompleta2.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.DateFormatter(new
java.text.SimpleDateFormat("dd/MM/yyyy"))));
fechacompleta2.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        fechacompleta2KeyTyped(evt);
    }
});

btnBuscar.setText("Buscar");
btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(evt);
    }
});

btnVolver.setText("Volver");
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnVolverActionPerformed(evt);
    }
});

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {},
        {},
        {},
        {}
    },
    new String [] {

    }
));
jScrollPane1.setViewportView(jTable1);

```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel2)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(fechaCompleta1, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(fechaCompleta2, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(btnBuscar)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 113, Short.MAX_VALUE)
                        .addComponent(btnVolver)))
                    .addGap(10, 10, 10))
            .addContainerGap())
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(fechaCompleta1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3)
                        .addComponent(fechaCompleta2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(btnBuscar)
                        .addComponent(btnVolver))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 448,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap())
    );

pack();
} // </editor-fold>

private void btnVolverActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.fechaCompleta1.setText(null);
    this.fechaCompleta2.setText(null);
    this.superCtrl.muestraVentanaReportes(true);
}

```

```

private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.fechacompleta1.getText().length() < 1 || this.fechacompleta2.getText().length() < 1 ) {
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }
    else if (validafecha(this.fechacompleta1.getText()) && validafecha(this.fechacompleta2.getText()) &&
        fecha(this.fechacompleta1.getText(), this.fechacompleta2.getText())) {
        try {
            this.modelo.setRowCount(0);
            PreparedStatement st = null;
            ResultSet rt = null;
            ConexionBD con = new ConexionBD();
            Connection cn = con.getConexion();
            String sql = "SELECT nombre_completo(id_paciente) as nombre, fecha_hora_pc FROM consultas NATURAL JOIN paciente
where consultas.fecha_hora_pc BETWEEN to_date(?, 'DD/MM/YY') and to_date(?, 'DD/MM/YY')+1 order by fecha_hora_pc ASC";
            st = cn.prepareStatement(sql);
            st.setString(1, this.fechacompleta1.getText());
            st.setString(2, this.fechacompleta2.getText());
            // System.out.println("aqui1");
            rt = st.executeQuery();
            // System.out.println("aqui2");
            ResultSetMetaData rsMd = rt.getMetaData();
            int cantcolumnas = rsMd.getColumnCount();
            // System.out.println("aqui3");
            // System.out.println(cantcolumnas);
            while (rt.next()) {
                // System.out.println("aqui bucle1");
                Object[] filas = new Object[cantcolumnas];
                // System.out.println(filas);
                for (int i = 0; i < cantcolumnas; i++) {
                    //System.out.println("aqui bucle2");
                    filas[i] = rt.getObject(i + 1);
                }
                //System.out.println("aqui4");
                this.modelo.addRow(filas);
            }
            //System.out.println("aqui5");
            this.jTable1.setModel(this.modelo);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
        }
    }
    else if (!validafecha(this.fechacompleta1.getText())) {
        JOptionPane.showMessageDialog(this, "Ingreso fecha inicial valida", "Alerta", 2);
    }
    else if (!validafecha(this.fechacompleta2.getText())) {
        JOptionPane.showMessageDialog(this, "Ingreso fecha final valida", "Alerta", 2);
    }
    else if (!fecha(this.fechacompleta1.getText(), this.fechacompleta2.getText())) {
        JOptionPane.showMessageDialog(this, "Ingreso fecha inicial menor a la final", "Alerta", 2);
    }
}

private void fechacompleta1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (Character.isDigit(c) || c=='/') {
        if (this.fechacompleta1.getText().length() < 10) {
            if (this.fechacompleta1.getText().length() == 2 && c!='/') {
                this.fechacompleta1.setText(this.fechacompleta1.getText() + "/");
            }
            else if (this.fechacompleta1.getText().length() == 5&& c!='/') {

```



```

        this.fechacompleta1.setText(this.fechacompleta1.getText() + "/");
    }
} else if (this.fechacompleta1.getText().length() > 9) {
    evt.consume();
    Toolkit.getDefaultToolkit().beep();
}
} else {
    evt.consume();
    Toolkit.getDefaultToolkit().beep();
}
}

private void fechacompleta2KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (Character.isDigit(c) || c=='/') {
        if (this.fechacompleta2.getText().length() < 10) {
            if (this.fechacompleta2.getText().length() == 2 && c!='/') {
                this.fechacompleta2.setText(this.fechacompleta2.getText() + "/");
            } else if (this.fechacompleta2.getText().length() == 5 && c!='/') {
                this.fechacompleta2.setText(this.fechacompleta2.getText() + "/");
            }
        } else if (this.fechacompleta2.getText().length() > 9) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    } else {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(VentanaTablaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(VentanaTablaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(VentanaTablaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(VentanaTablaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VentanaTablaReportes().setVisible(true);
    }
});
}

private boolean validafecha(final String fecha) {
    String[] fecha2 = fecha.split("/");
    String dia = fecha2[0];
    String mes = fecha2[1];
    String annio = fecha2[2];
    System.out.println(dia);
    System.out.println(mes);
    System.out.println(annio);
    /*Date dt = new Date();
    int dia2 = dt.getDate();
    int mes2 = dt.getMonth() + 1;
    int annio2 = dt.getYear() + 1900;*/
    final SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
    try {
        formatoFecha.setLenient(false);
        formatoFecha.parse(dia + "/" + mes + "/" + annio);
        System.out.println("validafecha: fecha valida");
        return true;
    } catch (ParseException e) {
        System.out.println("validafecha: error al parcear fecha");
        return false;
    }
    /*
    if (Integer.parseInt(annio) <= annio2) {
        if (Integer.parseInt(annio) == annio2) {
            if (Integer.parseInt(mes) > mes2) {
                System.out.println("validafecha: mes ingresado mayor al actual");
                return false;
            }
            if (Integer.parseInt(dia) > dia2) {
                System.out.println("validafecha: dia ingresado mayor al actual");
                return false;
            }
        } else if (Integer.parseInt(annio) < annio2 - 100) {
            System.out.println("validafecha: año ingresado menor a 1900");
            return false;
        }
        System.out.println("validafecha: año ingresado menor al actual");
        return true;
    }
    return false;*/
}

private boolean fecha(String fecha1, String fecha2) {
    String[] fecha1a = fecha1.split("/");
    String dia1 = fecha1a[0];
    String mes1 = fecha1a[1];
    String annio1 = fecha1a[2];
    String[] fecha2a = fecha2.split("/");
    String dia2 = fecha2a[0];
    String mes2 = fecha2a[1];

```

```

        String annio2 = fehab[2];
        if (Integer.parseInt(annio1) <= Integer.parseInt(annio2)) {
            if (Integer.parseInt(annio1) == Integer.parseInt(annio2)) {
                if (Integer.parseInt(mes1) > Integer.parseInt(mes2)) {
                    System.out.println("fecha: mes1 mayor al mes2");
                    return false;
                } else if (Integer.parseInt(mes1) == Integer.parseInt(mes2) && Integer.parseInt(dia1) > Integer.parseInt(dia2)) {
                    System.out.println("fecha: mes1 igual al mes2 y dia1 mayor a dia2");
                    return false;
                }
            } else if (Integer.parseInt(annio1) < Integer.parseInt(annio2) - 100 || Integer.parseInt(annio2) > Integer.parseInt(annio1) + 100) {
                System.out.println("fecha: jamas deberia entrar aqui");
                return false;
            }
        } else {
            System.out.println("fecha: año1 mayor o igual al año2");
            return false;
        }
        System.out.println("fecha: intervalo correcto");
        return true;
    }

    public void paciente() {
        this.modelo.setColumnCount(0);
        this.jLabel1.setVisible(false);
        this.jLabel2.setVisible(false);
        this.jLabel3.setVisible(false);
        this.fechacompleta1.setVisible(false);
        this.fechacompleta2.setVisible(false);
        this.btnBuscar.setVisible(false);
        this.modelo.addColumn("Id");
        this.modelo.addColumn("Nombre");
        this.modelo.addColumn("Telefono");
        this.modelo.addColumn("Motivo de Visita");

        try {
            this.modelo.setRowCount(0);
            PreparedStatement st = null;
            ResultSet rt = null;
            ConexionBD con = new ConexionBD();
            Connection cn = con.getConnection();
            String sql = "SELECT id_paciente, nombre_completo(id_paciente) as nombre,tel,motivo_visita FROM cuestionario NATURAL
JOIN paciente";
            st = cn.prepareStatement(sql);
            rt = st.executeQuery();

            ResultSetMetaData rsMd = rt.getMetaData();
            int cantcolumnas = rsMd.getColumnCount();

            while (rt.next()) {
                Object[] filas = new Object[cantcolumnas];
                for (int i = 0; i < cantcolumnas; i++) {
                    filas[i] = rt.getObject(i + 1);
                }
                this.modelo.addRow(filas);
            }

            this.jTable1.setModel(this.modelo);

```

```

TableModel columnModel = this.jTable1.getColumnModel();

columnModel.getColumnModel().getColumn(0).setPreferredWidth(15);
columnModel.getColumnModel().getColumn(1).setPreferredWidth(250);
columnModel.getColumnModel().getColumn(2).setPreferredWidth(50);
columnModel.getColumnModel().getColumn(3).setPreferredWidth(250);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
}
}

public void adeudo() {
    this.modelo.setColumnCount(0);
    this.jLabel1.setVisible(false);
    this.jLabel2.setVisible(false);
    this.jLabel3.setVisible(false);
    this.fechaCompleta1.setVisible(false);
    this.fechaCompleta2.setVisible(false);
    this.btnBuscar.setVisible(false);
    this.modelo.addColumn("Nombre");
    this.modelo.addColumn("Adeudo");
    try {
        this.modelo.setRowCount(0);
        PreparedStatement st = null;
        ResultSet rt = null;
        ConexionBD con = new ConexionBD();
        Connection cn = con.getConnection();
        String sql = "select nombre_completo(paciente.id_paciente) as nombre,SUM(consultas.adeudo) AS adeudo FROM consultas LEFT JOIN paciente on consultas.id_paciente = paciente.id_paciente GROUP BY paciente.id_paciente,nombre,adeudo having adeudo>0";
        st = cn.prepareStatement(sql);
        rt = st.executeQuery();

        ResultSetMetaData rsMd = rt.getMetaData();
        int cantcolumnas = rsMd.getColumnCount();

        while (rt.next()) {
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
            }
            this.modelo.addRow(filas);
        }
        this.jTable1.setModel(this.modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    }
}

public void citas() {
    this.modelo.setColumnCount(0);
    this.jLabel1.setVisible(false);
    this.jLabel2.setVisible(false);
    this.jLabel3.setVisible(false);
    this.fechaCompleta1.setVisible(false);
    this.fechaCompleta2.setVisible(false);
    this.btnBuscar.setVisible(false);
    this.modelo.addColumn("Nombre");
    this.modelo.addColumn("Cita");
    try {

```

```

        this.modelo.setRowCount(0);
        PreparedStatement st = null;
        ResultSet rt = null;
        ConexionBD con = new ConexionBD();
        Connection cn = con.getConexion();
        String sql = "SELECT nombre_completo(id_paciente) as nombre,fecha_hora_pc FROM consultas NATURAL JOIN paciente
where to_date(TO_CHAR(consultas.fecha_hora_pc,'DD/MM/YY'))>=current_date order by fecha_hora_pc ASC";
        st = cn.prepareStatement(sql);
        rt = st.executeQuery();

        ResultSetMetaData rsMd = rt.getMetaData();
        int cantcolumnas = rsMd.getColumnCount();

        while (rt.next()) {
            Object[] filas = new Object[cantcolumnas];
            for (int i = 0; i < cantcolumnas; i++) {
                filas[i] = rt.getObject(i + 1);
            }
            this.modelo.addRow(filas);
        }
        this.jTable1.setModel(this.modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta");
    }
}

public void citasxfecha() {
    this.modelo.setColumnCount(0);
    this.modelo.setRowCount(0);
    this.jLabel1.setVisible(true);
    this.jLabel2.setVisible(true);
    this.jLabel3.setVisible(true);
    this.fechacompleta1.setVisible(true);
    this.fechacompleta2.setVisible(true);
    this.btnBuscar.setVisible(true);
    this.modelo.addColumn("Nombre");
    this.modelo.addColumn("Cita");
}

@Override
public void actionPerformed(ActionEvent e) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
// Variables declaration - do not modify
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnVolver;
private javax.swing.JFormattedTextField fechacompleta1;
private javax.swing.JFormattedTextField fechacompleta2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```



```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package GUI;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.LayoutStyle;
/**
 *
 * @author Jiro
 */
public class VentanaReportes extends VentanaControladaAbstract {

    /**
     * Creates new form VentanaReportes
     */
    public VentanaReportes() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
    }

```

```

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
jLabel1.setText("Reportes");

jToggleButton1.setText("Pacientes");
jToggleButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton1ActionPerformed(evt);
    }
});

jToggleButton2.setText("Citas");
jToggleButton2.setToolTipText("");
jToggleButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton2ActionPerformed(evt);
    }
});

jToggleButton3.setText("Adeudos");
jToggleButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jToggleButton3ActionPerformed(evt);
    }
});

jButton1.setText("Volver");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(26, 26, 26)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addGroup(layout.createSequentialGroup()
                    .addGap(6, 6, 6)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jButton1)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(jToggleButton1, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jToggleButton2, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jToggleButton3, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                    .addGap(27, Short.MAX_VALUE))
            .addContainerGap())
);

```

```

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jToggleButton1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jToggleButton2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jToggleButton3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton1)
            .addContainerGap(16, Short.MAX_VALUE))
        );

pack();
} // </editor-fold>

private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraTablaVentanaReportes("pacientes");
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaLobby(true);
}

private void jToggleButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    final String[] options = { "Citas pendientes", "Citas por fecha", "Cancelar" };
    final int seleccion = JOptionPane.showOptionDialog(null, "Es necesario que seleccione una opcion", "Citas", -1, 3, null, options,
options[0]);
    if (seleccion == 0) {
        this.superCtrl.muestraTablaVentanaReportes("cita1");
    }
    else if (seleccion == 1) {
        this.superCtrl.muestraTablaVentanaReportes("cita2");
    }
}

private void jToggleButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraTablaVentanaReportes("adeudo");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

```



```

        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(VentanaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(VentanaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(VentanaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(VentanaReportes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VentanaReportes().setVisible(true);
    }
});
}

public void actionPerformed(ActionEvent e) { throw new UnsupportedOperationException("Not supported yet."); }
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JToggleButton jToggleButton1;
private javax.swing.JToggleButton jToggleButton2;
private javax.swing.JToggleButton jToggleButton3;
// End of variables declaration
}

```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package GUI;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.LayoutStyle;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import Conexion.ConexionBD;
import java.awt.LayoutManager;

import java.awt.Component;

/**
 *
 * @author Jiro
 */
public class VentanaPacientes extends VentanaControladaAbstract {

    /**
     * Creates new form VentanaPacientes
     */
    public VentanaPacientes() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
}
```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    btnAgregarPaciente = new javax.swing.JButton();
    btnBuscarPaciente = new javax.swing.JButton();
    btnVolverPaciente = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    lblNumeroPac = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    setPreferredSize(null);

    btnAgregarPaciente.setText("Nuevo paciente");
    btnAgregarPaciente.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnAgregarPacienteActionPerformed(evt);
        }
    });

    btnBuscarPaciente.setText("Buscar");
    btnBuscarPaciente.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnBuscarPacienteActionPerformed(evt);
        }
    });

    btnVolverPaciente.setText("Volver");
    btnVolverPaciente.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnVolverPacienteActionPerformed(evt);
        }
    });

    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
    jLabel1.setText("Menu Pacientes");

    jLabel2.setText("pacientes:");

    lblNumeroPac.setText("0");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(btnAgregarPaciente)
                    .add(btnVolverPaciente)
                    .add(btnBuscarPaciente)
                    .add(jLabel1)
                    .add(jLabel2)
                    .add(lblNumeroPac))
                .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(layout.createSequentialGroup()
                .add(btnAgregarPaciente)
                .add(btnVolverPaciente)
                .add(btnBuscarPaciente)
                .add(jLabel1)
                .add(jLabel2)
                .add(lblNumeroPac)
                .addContainerGap())
    );
}

```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(30, 30, 30)
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(lblNumeroPac))))
    .addGroup(layout.createSequentialGroup())
    .addGap(35, 35, 35)
    .addComponent(btnVolverPaciente, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap(16, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(jLabel1)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(lblNumeroPac))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(btnAgregarPaciente)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(btnBuscarPaciente)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(btnVolverPaciente)
    .addContainerGap(12, Short.MAX_VALUE))
);

pack();
} // </editor-fold>

private void btnAgregarPacienteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaAgregarPacientes(true);
}

private void btnBuscarPacienteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaBuscarP(true);
}

private void btnVolverPacienteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.superCtrl.muestraVentanaLobby(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

```

```

        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(VentanaPacientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(VentanaPacientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(VentanaPacientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(VentanaPacientes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VentanaPacientes().setVisible(true);
    }
});
}

public void actionPerformed(ActionEvent e) {}
public void consulta() {
    //SELECT COUNT(id_paciente) FROM paciente'
    //'COUNT(id_paciente)'
    final ConexionBD con = new ConexionBD();
    final Connection cn = con.getConnection();
    final String sql = "SELECT COUNT(id_paciente) FROM paciente";
    try {
        final PreparedStatement st = cn.prepareStatement(sql);
        final ResultSet rt = st.executeQuery();
        if (rt.next()) {
            this.lblNumeroPac.setText("" + rt.getInt("COUNT(id_paciente)"));
            cn.close();
            con.desconectar();
            rt.close();
            st.close();
        }
    }
    catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage(), "Error", 0);
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btnAgregarPaciente;
private javax.swing.JButton btnBuscarPaciente;
private javax.swing.JButton btnVolverPaciente;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel lblNumeroPac;
// End of variables declaration
}

```

### Cuestionario inicial

Marque la casilla que corresponda a su situacion

☐ ¿Molestias o dolor de boca? Detalles:

☐ ¿Mal olor o sabor en la boca? Detalles:

☐ ¿Sangrado de encías? Detalles:

☐ ¿Tiene dientes moviles o le rechinan? Detalles:

☐ ¿Malos habitos orofaciales? (morder uñas/lapices/etc. chuparse los dedos, etc.) Detalles:

☐ Aparte de cepillo de dientes, ¿Utiliza algun otro aditamento de limpieza? ¿Cuales?

Ultima visita (dd/mm/yyyy):  En caso de primera visita ingrese fecha actual

Numero de limpiezas al dia:

Motivo de consulta:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package GUI;

```
import Conexion.ConexionBD;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.awt.EventQueue;
import javax.swing.UnsupportedLookAndFeelException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.UIManager;
import java.sql.Connection;
import java.sql.SQLException;
import java.awt.HeadlessException;
import javax.swing.JOptionPane;
import java.awt.Toolkit;
import javax.swing.LayoutStyle;
import java.awt.LayoutManager;
import javax.swing.GroupLayout;
import java.awt.Font;
import java.awt.event.KeyListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Component;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import java.sql.PreparedStatement;
import java.util.Calendar;
```

```

import java.util.Date;

/**
 *
 * @author Jiro
 */
public class VentanaCuestionario extends VentanaControladaAbstract {

    private ConexionBD con;
    private PreparedStatement st;
    private int res;
    private Calendar c = Calendar.getInstance();

    /**
     * Creates new form VentanaCuestionario
     */
    public VentanaCuestionario() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        cbMolestia = new javax.swing.JCheckBox();
        cbOlor = new javax.swing.JCheckBox();
        cbSangrado = new javax.swing.JCheckBox();
        cbMovil = new javax.swing.JCheckBox();
        cbHabitos = new javax.swing.JCheckBox();
        cbAditamentos = new javax.swing.JCheckBox();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        txtMolestiaDetalle = new javax.swing.JTextField();
        txtOlorDetalle = new javax.swing.JTextField();
        txtSangradoDetalle = new javax.swing.JTextField();
        txtMovilDetalle = new javax.swing.JTextField();
        txtHabitosDetalle = new javax.swing.JTextField();
        txtAditamentosDetalle = new javax.swing.JTextField();
        jLabel9 = new javax.swing.JLabel();
        fechacompleta = new javax.swing.JFormattedTextField();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        txtNoCepPacienteCuestionario = new javax.swing.JFormattedTextField();
        jLabel12 = new javax.swing.JLabel();
        txtMotivoPacienteCuestionario = new javax.swing.JTextField();
        btnAgregarPacienteCuestionario = new javax.swing.JButton();
    }

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jLabel1.setText("Cuestionario inicial");

cbMolestia.setText("¿Molestias o dolor de boca?");
cbMolestia.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbMolestiaActionPerformed(evt);
    }
});

cbOlor.setText("¿Mal olor o sabor en la boca?");
cbOlor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbOlorActionPerformed(evt);
    }
});

cbSangrado.setText("¿Sangrado de encías?");
cbSangrado.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbSangradoActionPerformed(evt);
    }
});

cbMovil.setText("¿Tiene dientes móviles o le rechinan?");
cbMovil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbMovilActionPerformed(evt);
    }
});

cbHabititos.setText("¿Malos hábitos orofaciales? (morder uñas/lápices/etc. chuparse los dedos, etc.)");
cbHabititos.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbHabititosActionPerformed(evt);
    }
});

cbAditamentos.setText("Aparte de cepillo de dientes, ¿Utiliza algún otro aditamento de limpieza?");
cbAditamentos.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbAditamentosActionPerformed(evt);
    }
});

jLabel2.setText("Marque la casilla que corresponda a su situación");

jLabel3.setText("Detalles:");

jLabel4.setText("Detalles:");

jLabel5.setText("Detalles:");

jLabel6.setText("Detalles:");

```



```

jLabel7.setText("Detalles:");

jLabel8.setText("¿Cuales?");

txtMolestiaDetalle.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtMolestiaDetalleActionPerformed(evt);
    }
});
txtMolestiaDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMolestiaDetalleKeyTyped(evt);
    }
});

txtOlorDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtOlorDetalleKeyTyped(evt);
    }
});

txtSangradoDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtSangradoDetalleKeyTyped(evt);
    }
});

txtMovilDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtMovilDetalleKeyTyped(evt);
    }
});

txtHabitosDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtHabitosDetalleKeyTyped(evt);
    }
});

txtAditamentosDetalle.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtAditamentosDetalleKeyTyped(evt);
    }
});

jLabel9.setText("Ultima visita (dd/mm/yyyy):");

fechacompleta.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new javax.swing.text.DateFormatter(new
java.text.SimpleDateFormat("dd/MM/yyyy"))));
fechacompleta.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        fechacompletaFocusLost(evt);
    }
});
fechacompleta.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        fechacompletaKeyTyped(evt);
    }
});

```



```

        .addGap(18, 18, 18)
        .addComponent(jLabel6)
        .addGap(18, 18, 18)
        .addComponent(txtMovilDetalle, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(cbHabitos)
        .addGap(18, 18, 18)
        .addComponent(jLabel7)
        .addGap(18, 18, 18)
        .addComponent(txtHabitosDetalle, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(cbAditamentos)
        .addGap(18, 18, 18)
        .addComponent(jLabel8)
        .addGap(18, 18, 18)
        .addComponent(txtAditamentosDetalle, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel2)
        .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false))
        .addGroup(layout.createSequentialGroup())
        .addComponent(cbOlor)
        .addGap(18, 18, 18)
        .addComponent(jLabel4))
        .addGroup(layout.createSequentialGroup())
        .addComponent(cbMolestia)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel3)))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(txtMolestiaDetalle, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtOlorDetalle, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel9)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(fechacompleta, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel10))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel11)
        .addGap(18, 18, 18)
        .addComponent(txtNoCepPacienteCuestionario, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel12)
        .addGap(18, 18, 18)
        .addComponent(txtMotivoPacienteCuestionario, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup())
        .addGap(314, 314, 314)
        .addComponent(jLabel1))
        .addGroup(layout.createSequentialGroup())

```

```

        .addGap(377, 377, 377)
        .addComponent(btnAgregarPacienteCuestionario)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addGap(22, 22, 22)
            .addComponent(jLabel2)
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbMolestia)
                .addComponent(jLabel3)
                .addComponent(txtMolestiaDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbOlor)
                .addComponent(jLabel4)
                .addComponent(txtOlorDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbSangrado)
                .addComponent(jLabel5)
                .addComponent(txtSangradoDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbMovil)
                .addComponent(jLabel6)
                .addComponent(txtMovilDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbHabitos)
                .addComponent(jLabel7)
                .addComponent(txtHabitosDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(cbAditamentos)
                .addComponent(jLabel8)
                .addComponent(txtAditamentosDetalle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(9, 9, 9)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel9)
                .addComponent(fechacompleta, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel10))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel11)
                .addComponent(txtNoCepPacienteCuestionario, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel12)
            .addComponent(txtMotivoPacienteCuestionario, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(btnAgregarPacienteCuestionario)
        .addContainerGap(32, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void cbMolestiaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

private void cbMovilActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

private void cbHabitosActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

private void fechacompletaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (Character.isDigit(c)||c=='/') {
        if (this.fechacompleta.getText().length() < 10) {
            if (this.fechacompleta.getText().length() == 2&& c!='/') {
                this.fechacompleta.setText(this.fechacompleta.getText() + "/");
            } else if (this.fechacompleta.getText().length() == 5&& c!='/') {
                this.fechacompleta.setText(this.fechacompleta.getText() + "/");
            }
        } else if (this.fechacompleta.getText().length() > 9) {
            evt.consume();
            Toolkit.getDefaultToolkit().beep();
        }
    } else {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void cbOlorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

private void cbSangradoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

```

```

private void cbAditamentosActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    validacheck(evt.getActionCommand());
}

private void txtMotivoPacienteCuestionarioActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtMotivoPacienteCuestionarioKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras");
    } else if (this.txtMotivoPacienteCuestionario.getText().length() >= 30) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void txtNoCepPacienteCuestionarioActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtNoCepPacienteCuestionarioKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isLetter(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo numeros");
    } else if (this.txtNoCepPacienteCuestionario.getText().length() >= 2) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void btnAgregarPacienteCuestionarioActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.txtNoCepPacienteCuestionario.getText().length() < 1 || this.txtMotivoPacienteCuestionario.getText().length() < 1) {
        JOptionPane.showMessageDialog(this, "No puede dejar Motivo de consulta o numero de cepillados en blanco", "Alerta", 2);
    } else {
        try {
            if (validafecha(fechacompleta.getText())) {
                this.con = new ConexionBD();
                Connection cn = this.con.getConnection();
                try {
                    String sql = "INSERT INTO cuestionario
(id_paciente,dolor,olor,sangrado,diente_movil,habitos,aditamentos,nota_dolor,nota_olor,nota_sangrado,nota_dmovil,notas_habitos,
det_aditamentos,ultima_visita,num_cep_dientes,motivo_visita) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,to_date('DD/MM/YYYY'),?,?)";
                    this.st = cn.prepareStatement(sql);
                    this.st.setInt(1, Integer.parseInt(this.superCtrl.getClave()));
                    if (this.cbMolestia.isSelected()) {
                        this.st.setInt(2, 1);

```

```

        this.st.setString(8, this.txtMolestiaDetalle.getText());
    } else {
        this.st.setInt(2, 0);
        this.st.setNull(8, 4);
    }
    if (this.cbOlor.isSelected()) {
        this.st.setInt(3, 1);
        this.st.setString(9, this.txtOlorDetalle.getText());
    } else {
        this.st.setInt(3, 0);
        this.st.setNull(9, 4);
    }
    if (this.cbSangrado.isSelected()) {
        this.st.setInt(4, 1);
        this.st.setString(10, this.txtSangradoDetalle.getText());
    } else {
        this.st.setInt(4, 0);
        this.st.setNull(10, 4);
    }
    if (this.cbMovil.isSelected()) {
        this.st.setInt(5, 1);
        this.st.setString(11, this.txtMovilDetalle.getText());
    } else {
        this.st.setInt(5, 0);
        this.st.setNull(11, 4);
    }
    if (this.cbHabitos.isSelected()) {
        this.st.setInt(6, 1);
        this.st.setString(12, this.txtHabitosDetalle.getText());
    } else {
        this.st.setInt(6, 0);
        this.st.setNull(12, 4);
    }
    if (this.cbAditamentos.isSelected()) {
        this.st.setInt(7, 1);
        this.st.setString(13, this.txtAditamentosDetalle.getText());
    } else {
        this.st.setInt(7, 0);
        this.st.setNull(13, 4);
    }
    this.st.setString(14, fechaCompleta.getText());
    this.st.setInt(15, Integer.parseInt(this.txtNoCepPacienteCuestionario.getText()));
    this.st.setString(16, this.txtMotivoPacienteCuestionario.getText());
    this.res = this.st.executeUpdate();
    if (this.res > 0) {
        this.con.desconectar();
        this.st.close();
        limpiar();
        this.superCtrl.muestraVentanaAgregarCuestionario(false);
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un problema al registrar el cuestionario, intente de nuevo.",
"Error", 0);
    }
    if (cn != null) {
        cn.close();
    }
} catch (Throwable throwable) {
    if (cn != null)

```

```

        try {
            cn.close();
        } catch (Throwable throwable1) {
            throwable.addSuppressed(throwable1);
        }
        throw throwable;
    }
} else {
    JOptionPane.showMessageDialog(null, "Ingrese una fecha valida", "Alerta", 2);
}

} catch (HeadlessException | NumberFormatException | java.sql.SQLException e) {
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}
}
}

```

```

private void txtMolestiaDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

}

```

```

private void txtMolestiaDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras");
    } else if (this.txtMolestiaDetalle.getText().length() >= 20) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}
}

```

```

private void txtOlorDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras");
    } else if (this.txtOlorDetalle.getText().length() >= 20) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}
}

```

```

private void txtSangradoDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
    }
}

```



```

        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras");
    } else if (this.txtSangradoDetalle.getText().length() >= 20) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void txtMovilDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras");
    } else if (this.txtMovilDetalle.getText().length() >= 20) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void txtHabitosDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras");
    } else if (this.txtHabitosDetalle.getText().length() >= 20) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void txtAditamentosDetalleKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras");
    } else if (this.txtAditamentosDetalle.getText().length() >= 30) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

private void fechacompletaFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
}

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(VentanaCuestionario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(VentanaCuestionario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(VentanaCuestionario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(VentanaCuestionario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new VentanaCuestionario().setVisible(true);
        }
    });
}

private void validacheck(String cadena) {
    if (null != cadena) {
        String str = cadena;
        byte b = -1;
        switch (str.hashCode()) {
            case -389776300:
                if (str.equals("molestia")) {
                    b = 0;
                }
                break;
            case 3414144:
                if (str.equals("olor")) {
                    b = 1;
                }
                break;
            case 1996058625:
                if (str.equals("sangrado")) {
                    b = 2;
                }
                break;
            case 1243699749:
                if (str.equals("moviles")) {

```

```

        b = 3;
    }
    break;
case 681878104:
    if (str.equals("habitos")) {
        b = 4;
    }
    break;
case -2072936683:
    if (str.equals("aditamentos")) {
        b = 5;
    }
    break;
}
switch (b) {
    case 0:
        if (this.txtMolestiaDetalle.isEnabled()) {
            this.txtMolestiaDetalle.setEnabled(false);
            this.txtMolestiaDetalle.setText(null);
            break;
        }
        this.txtMolestiaDetalle.setEnabled(true);
        break;
    case 1:
        if (this.txtOlorDetalle.isEnabled()) {
            this.txtOlorDetalle.setEnabled(false);
            this.txtOlorDetalle.setText(null);
            break;
        }
        this.txtOlorDetalle.setEnabled(true);
        break;
    case 2:
        if (this.txtSangradoDetalle.isEnabled()) {
            this.txtSangradoDetalle.setEnabled(false);
            this.txtSangradoDetalle.setText(null);
            break;
        }
        this.txtSangradoDetalle.setEnabled(true);
        break;
    case 3:
        if (this.txtMovilDetalle.isEnabled()) {
            this.txtMovilDetalle.setEnabled(false);
            this.txtMovilDetalle.setText(null);
            break;
        }
        this.txtMovilDetalle.setEnabled(true);
        break;
    case 4:
        if (this.txtHabitosDetalle.isEnabled()) {
            this.txtHabitosDetalle.setEnabled(false);
            this.txtHabitosDetalle.setText(null);
            break;
        }
        this.txtHabitosDetalle.setEnabled(true);
        break;
    case 5:
        if (this.txtAditamentosDetalle.isEnabled()) {
            this.txtAditamentosDetalle.setEnabled(false);

```

```

        this.txtAditamentosDetalle.setText(null);
        break;
    }
    this.txtAditamentosDetalle.setEnabled(true);
    break;
}
}
}
}

```

```

private void limpiar() {
    this.cbMolestia.setSelected(false);
    this.txtMolestiaDetalle.setEnabled(true);
    this.txtMolestiaDetalle.setText(null);
    this.txtMolestiaDetalle.setEnabled(false);

    this.cbHabitos.setSelected(false);
    this.txtHabitosDetalle.setEnabled(true);
    this.txtHabitosDetalle.setText(null);
    this.txtHabitosDetalle.setEnabled(false);

    this.cbMovil.setSelected(false);
    this.txtMovilDetalle.setEnabled(true);
    this.txtMovilDetalle.setText(null);
    this.txtMovilDetalle.setEnabled(false);

    this.cbAditamentos.setSelected(false);
    this.txtAditamentosDetalle.setEnabled(true);
    this.txtAditamentosDetalle.setText(null);
    this.txtAditamentosDetalle.setEnabled(false);

    this.cbOlor.setSelected(false);
    this.txtOlorDetalle.setEnabled(true);
    this.txtOlorDetalle.setText(null);
    this.txtOlorDetalle.setEnabled(false);

    this.cbSangrado.setSelected(false);
    this.txtSangradoDetalle.setEnabled(true);
    this.txtSangradoDetalle.setText(null);
    this.txtSangradoDetalle.setEnabled(false);

    this.fechacompleta.setText(null);
    this.txtNoCepPacienteCuestionario.setText(null);
    this.txtMotivoPacienteCuestionario.setText(null);
}

```

```

private boolean validafecha(final String fecha) {
    if (fechacompleta.getText().length() < 10) {
        return false;
    }
    String[] fecha2 = fecha.split("/");
    String dia = fecha2[0];
    String mes = fecha2[1];
    String annio = fecha2[2];
    Date dt = new Date();
    int dia2 = dt.getDate();
    int mes2 = dt.getMonth() + 1;
    int annio2 = dt.getYear() + 1900;
}

```

```

final SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
try {
    formatoFecha.setLenient(false);
    formatoFecha.parse(dia + "/" + mes + "/" + annio);
} catch (ParseException e) {
    return false;
}
if (Integer.parseInt(annio) <= annio2) {
    if (Integer.parseInt(annio) == annio2) {
        if (Integer.parseInt(mes) > mes2) {
            return false;
        }
        if (Integer.parseInt(dia) > dia2) {
            return false;
        }
    } else if (Integer.parseInt(annio) < annio2 - 100) {
        return false;
    }
    return true;
}
return false;
}

```

```

private String aFecha(String dia, String mes, String annio) {
    if (dia.length() < 2) {
        dia = "0" + dia;
    } else if (mes.length() < 2) {
        mes = "0" + mes;
    }
    //SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
    return dia + "/" + mes + "/" + annio;
}

```

```

public void actionPerformed(ActionEvent e) {
}

```

```

// Variables declaration - do not modify
private javax.swing.JButton btnAgregarPacienteCuestionario;
private javax.swing.JCheckBox cbAditamentos;
private javax.swing.JCheckBox cbHabitos;
private javax.swing.JCheckBox cbMolestia;
private javax.swing.JCheckBox cbMovil;
private javax.swing.JCheckBox cbOlor;
private javax.swing.JCheckBox cbSangrado;
private javax.swing.JFormattedTextField fechacompleta;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;

```

```
private javax.swing.JTextField txtAditamentosDetalle;  
private javax.swing.JTextField txtHabitosDetalle;  
private javax.swing.JTextField txtMolestiaDetalle;  
private javax.swing.JTextField txtMotivoPacienteCuestionario;  
private javax.swing.JTextField txtMovilDetalle;  
private javax.swing.JFormattedTextField txtNoCepPacienteCuestionario;  
private javax.swing.JTextField txtOlorDetalle;  
private javax.swing.JTextField txtSangradoDetalle;  
// End of variables declaration  
}
```



```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package GUI;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.LayoutStyle;
/**
 *
 * @author Jiro
 */
public class VentanaCitas extends VentanaControladaAbstract {

    /**
     * Creates new form VentanaCitas
     */
    public VentanaCitas() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        newcita = new javax.swing.JButton();
        buscarcita = new javax.swing.JButton();
        volver = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

```

```

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
jLabel1.setText("Citas");

newcita.setText("Agendar cita");
newcita.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        newcitaActionPerformed(evt);
    }
});

buscascitas.setText("Buscar");
buscascitas.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        buscascitasActionPerformed(evt);
    }
});
volver.setText("volver");
volver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volverActionPerformed(evt);
    }
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(42, 42, 42)
            .addComponent(jLabel1)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                .addComponent(newcita, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(buscascitas, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(volver, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(20, 20, 20))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(newcita)
            .addGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(buscascitas)
            .addGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(volver)
            .addGap(0, 9, Short.MAX_VALUE))
        );
pack();
} // </editor-fold>

private void volverActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```



```

        this.superCtrl.muestraVentanaLobby(true);
    }
    private void buscartitasActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.superCtrl.muestraVentanabusquedacita(true);
    }
    private void newcitaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        this.superCtrl.muestraVentanabusquedacita1(true);
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(VentanaCitas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(VentanaCitas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(VentanaCitas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(VentanaCitas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new VentanaCitas().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton buscartitas;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JButton newcita;
    private javax.swing.JButton volver;
    // End of variables declaration

    @Override
    public void actionPerformed(ActionEvent e) {
        //throw new UnsupportedOperationException("Not supported yet."); // Generated from
        nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}

```

|                                                  |                                               |
|--------------------------------------------------|-----------------------------------------------|
| Nombre:                                          | <input type="text"/>                          |
| Apellidos:                                       | <input type="text"/>                          |
| Genero:                                          | <input type="text"/>                          |
| Estado Civil:                                    | <input type="text"/>                          |
| Ocupacion:                                       | <input type="text"/>                          |
| Telefono:                                        | <input type="text"/>                          |
| Correo:                                          | <input type="text"/>                          |
| Estado:                                          | <input type="text"/>                          |
| Ciudad:                                          | <input type="text"/>                          |
| Colonia:                                         | <input type="text"/>                          |
| Calle:                                           | <input type="text"/>                          |
| No. Ext.                                         | <input type="text"/> (Ingresa 0 si no aplica) |
| No. Int.                                         | <input type="text"/> (Opcional)               |
| C.P.                                             | <input type="text"/>                          |
| <div><div>Cancelar</div><div>Agregar</div></div> |                                               |

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template

\*/

```
package GUI;  
import Conexion.ConexionBD;
```

```
import Control.ValidaCorreo;  
//import com.sun.jdi.connect.spi.Connection;  
//HeadlessException  
import java.sql.Connection;  
import java.awt.Toolkit;
```

```

import java.sql.SQLException;
import java.sql.PreparedStatement;
import java.awt.event.ActionEvent;
import java.awt.HeadlessException;
import java.awt.event.ActionListener;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.DefaultComboBoxModel;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.LayoutStyle;
import java.sql.CallableStatement;

/**
 *
 * @author Jiro
 */
public class VentanaAgregarPaciente extends VentanaControladaAbstract {
    private ConexionBD con;

    private final ValidaCorreo valida = new ValidaCorreo();
    private PreparedStatement st;
    private ResultSet rt;
    private int g;
    private int res;
    private int last;

    /**
     * Creates new form VentanaAgregarPaciente
     */
    public VentanaAgregarPaciente() {
        initComponents();
        setLocationRelativeTo(null);
        llenar_combo_actividad();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lblNombrePacienteAgregar = new javax.swing.JLabel();
        lblApellidosPacienteAgregar = new javax.swing.JLabel();
        lblGeneroPacienteAgregar = new javax.swing.JLabel();
        lblEdoCivilPacienteAgregar = new javax.swing.JLabel();

```

```

lblOcupacionPacienteAgrega = new javax.swing.JLabel();
lblTelefonoPacienteAgrega = new javax.swing.JLabel();
lblCorreoPacienteAgrega = new javax.swing.JLabel();
lblCpPacienteAgrega = new javax.swing.JLabel();
lblNoInPacienteAgrega = new javax.swing.JLabel();
lblNoExPacienteAgrega = new javax.swing.JLabel();
lblCallePacienteAgrega = new javax.swing.JLabel();
lblColoniaPacienteAgrega = new javax.swing.JLabel();
lblCiudadPacienteAgrega = new javax.swing.JLabel();
lblEstadoPacienteAgrega = new javax.swing.JLabel();
btnCancelarPacienteAgrega = new javax.swing.JButton();
btnAgregaPacienteAgrega = new javax.swing.JButton();
txtNombrePacienteAgrega = new javax.swing.JTextField();
txtApellidosPacienteAgrega = new javax.swing.JTextField();
txtOcupacionPacienteAgrega = new javax.swing.JTextField();
txtTelefonoPacienteAgrega = new javax.swing.JTextField();
txtCorreoPacienteAgrega = new javax.swing.JTextField();
cbGeneroPacienteAgrega = new javax.swing.JComboBox<>();
cbEdoCivilPacienteAgrega = new javax.swing.JComboBox<>();
cbEstadoPacienteAgrega = new javax.swing.JComboBox<>();
txtCiudadPacienteAgrega = new javax.swing.JTextField();
txtColoniaPacienteAgrega = new javax.swing.JTextField();
txtCallePacienteAgrega = new javax.swing.JTextField();
txtNoExPacienteAgrega = new javax.swing.JTextField();
txtNoInPacienteAgrega = new javax.swing.JTextField();
txtCpPacienteAgrega = new javax.swing.JTextField();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
setResizable(false);

lblNombrePacienteAgrega.setText("Nombre:");

lblApellidosPacienteAgrega.setText("Apellidos:");

lblGeneroPacienteAgrega.setText("Genero:");

lblEdoCivilPacienteAgrega.setText("Estado Civil:");

lblOcupacionPacienteAgrega.setText("Ocupacion:");

lblTelefonoPacienteAgrega.setText("Telefono:");

lblCorreoPacienteAgrega.setText("Correo:");

lblCpPacienteAgrega.setText("C.P.");

lblNoInPacienteAgrega.setText("No. Int.");

lblNoExPacienteAgrega.setText("No. Ext.");

lblCallePacienteAgrega.setText("Calle:");

lblColoniaPacienteAgrega.setText("Colonia:");

lblCiudadPacienteAgrega.setText("Ciudad:");

```

```

lblEstadoPacienteAgregar.setText("Estado:");

btnCancelarPacienteAgregar.setText("Cancelar");
btnCancelarPacienteAgregar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarPacienteAgregarActionPerformed(evt);
    }
});

btnAgregarPacienteAgregar.setText("Aceptar");
btnAgregarPacienteAgregar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAgregarPacienteAgregarActionPerformed(evt);
    }
});

txtNombrePacienteAgregar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNombrePacienteAgregarActionPerformed(evt);
    }
});
txtNombrePacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNombrePacienteAgregarKeyTyped(evt);
    }
});

txtApellidosPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtApellidosPacienteAgregarKeyTyped(evt);
    }
});

txtOcupacionPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtOcupacionPacienteAgregarKeyTyped(evt);
    }
});

txtTelefonoPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtTelefonoPacienteAgregarKeyTyped(evt);
    }
});

txtCorreoPacienteAgregar.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        txtCorreoPacienteAgregarFocusLost(evt);
    }
});
txtCorreoPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCorreoPacienteAgregarKeyTyped(evt);
    }
});

cbGeneroPacienteAgregar.setModel(new DefaultComboBoxModel(new String[] { "", "H", "M" }));

```

```
cbEdoCivilPacienteAgregar.setModel(new DefaultComboBoxModel(new String[] { "", "Soltero", "Casado", "Divorciado" }));

cbEstadoPacienteAgregar.setModel(new DefaultComboBoxModel(new String[] {""}));

txtCiudadPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCiudadPacienteAgregarKeyTyped(evt);
    }
});

txtColoniaPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtColoniaPacienteAgregarKeyTyped(evt);
    }
});

txtCallePacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCallePacienteAgregarKeyTyped(evt);
    }
});

txtNoExPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNoExPacienteAgregarKeyTyped(evt);
    }
});

txtNolnPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtNolnPacienteAgregarKeyTyped(evt);
    }
});

txtCpPacienteAgregar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCpPacienteAgregarKeyTyped(evt);
    }
});

jLabel17.setText("(Ingrese 0 si no aplica)");

jLabel18.setText("(Opcional)");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtCallePacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtColoniaPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(lblCiudadPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblEstadoPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblCorreoPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblTelefonoPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblApellidosPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblGeneroPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblOcupacionPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblEdoCivilPacienteAgregar))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(cbGeneroPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cbEdoCivilPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(txtCallePacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtColoniaPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtCiudadPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cbEstadoPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtCorreoPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtTelefonoPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtOcupacionPacienteAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 151, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtApellidosPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGroup(layout.createSequentialGroup())
        .addComponent(lblNombrePacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(txtNombrePacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup())
        .addComponent(btnCancelarPacienteAgregar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnAgregarPacienteAgregar))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup())
        .addComponent(lblCpPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(txtCpPacienteAgregar, javax.swing.GroupLayout.DEFAULT_SIZE, 69, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup())

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(lblNoExPacienteAgrega, javax.swing.GroupLayout.DEFAULT_SIZE, 64,
Short.MAX_VALUE)
            .addComponent(lblNoInPacienteAgrega, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(txtNoExPacienteAgrega, javax.swing.GroupLayout.DEFAULT_SIZE, 69,
Short.MAX_VALUE)
            .addComponent(txtNoInPacienteAgrega))))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel17)
            .addComponent(jLabel18, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblNombrePacienteAgrega)
                .addComponent(txtNombrePacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblApellidosPacienteAgrega)
                .addComponent(txtApellidosPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblGeneroPacienteAgrega)
                .addComponent(cbGeneroPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblEdoCivilPacienteAgrega)
                .addComponent(cbEdoCivilPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblOcupacionPacienteAgrega)
                .addComponent(txtOcupacionPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblTelefonoPacienteAgrega)
                .addComponent(txtTelefonoPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblCorreoPacienteAgrega)
                .addComponent(txtCorreoPacienteAgrega, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(33, 33, 33)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(lblEstadoPacienteAgrega)

```



```

        .addComponent(cbEstadoPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblCiudadPacienteAgregar)
        .addComponent(txtCiudadPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblColoniaPacienteAgregar)
        .addComponent(txtColoniaPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblCallePacienteAgregar)
        .addComponent(txtCallePacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblNoExPacienteAgregar)
        .addComponent(txtNoExPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel17))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblNoInPacienteAgregar)
        .addComponent(txtNoInPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel18))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblCpPacienteAgregar)
        .addComponent(txtCpPacienteAgregar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnCancelarPacienteAgregar)
        .addComponent(btnAgregarPacienteAgregar))
        .addContainerGap()
    );

    pack();
} // </editor-fold>

```

```

private void btnCancelarPacienteAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    if ("agregar".equals(evt.getActionCommand())) {
        limpiar();
        this.superCtrl.muestraVentanaPacientes(true);
    } else {
        limpiar();
        this.superCtrl.muestraVentanaModificarPacientes(true);
    }
}

```

```

private void txtCorreoPacienteAgregarFocusLost(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    if (!this.valida.Valida(this.txtCorreoPacienteAregar.getText())) {
        JOptionPane.showMessageDialog(this, "Ingrese un correo valido", "Alerta", 2);
    }
}

```

```
}  
}
```

```
private void txtNombrePacienteAgregarKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char validar = evt.getKeyChar();  
    if (Character.isDigit(validar)) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras", "Atenci❖n", 1);  
    } else if (this.txtNombrePacienteAgregar.getText().length() >= 30) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
    } else if (Character.isLowerCase(validar)) {  
        evt.setKeyChar(Character.toUpperCase(validar));  
    }  
}
```

```
private void txtApellidosPacienteAgregarKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char validar = evt.getKeyChar();  
    if (Character.isDigit(validar)) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras", "Atenci❖n", 1);  
    } else if (this.txtApellidosPacienteAgregar.getText().length() >= 30) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
    } else if (Character.isLowerCase(validar)) {  
        evt.setKeyChar(Character.toUpperCase(validar));  
    }  
}
```

```
private void txtOcupacionPacienteAgregarKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char validar = evt.getKeyChar();  
    if (Character.isDigit(validar)) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
        JOptionPane.showMessageDialog(this.rootPane, "Ingreso solo letras", "Atenci❖n", 1);  
    } else if (this.txtOcupacionPacienteAgregar.getText().length() >= 30) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
    } else if (Character.isLowerCase(validar)) {  
        evt.setKeyChar(Character.toUpperCase(validar));  
    }  
}
```

```
private void txtCorreoPacienteAgregarKeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char validar = evt.getKeyChar();  
    if (this.txtCorreoPacienteAgregar.getText().length() >= 30) {  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
    }  
}
```

```
private void txtTelefonoPacienteAgregarKeyTyped(java.awt.event.KeyEvent evt) {
```

```

// TODO add your handling code here:
char validar = evt.getKeyChar();
if (Character.isLetter(validar)) {
    evt.consume();
    Toolkit.getDefaultToolkit().beep();
    JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo numeros", "Atenci♦n", 1);
} else if (this.txtTelefonoPacienteAgrega.getText().length() >= 10) {
    evt.consume();
    Toolkit.getDefaultToolkit().beep();
}
}

```

```

private void txtCiudadPacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras", "Atenci♦n", 1);
    } else if (this.txtCiudadPacienteAgrega.getText().length() >= 30) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

```

```

private void txtColoniaPacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isDigit(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo letras");
    } else if (this.txtColoniaPacienteAgrega.getText().length() >= 30) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

```

```

private void txtCallePacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (this.txtCallePacienteAgrega.getText().length() >= 30) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    } else if (Character.isLowerCase(validar)) {
        evt.setKeyChar(Character.toUpperCase(validar));
    }
}

```

```

private void txtNoExPacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isLetter(validar)) {
        evt.consume();
    }
}

```

```

        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo numeros");
    } else if (this.txtNoExPacienteAgrega.getText().length() >= 11) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void txtNolnPacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isLetter(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo numeros");
    } else if (this.txtNolnPacienteAgrega.getText().length() >= 11) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void txtCpPacienteAgregaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char validar = evt.getKeyChar();
    if (Character.isLetter(validar)) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this.rootPane, "Ingrese solo numeros");
    } else if (this.txtCpPacienteAgrega.getText().length() >= 5) {
        evt.consume();
        Toolkit.getDefaultToolkit().beep();
    }
}

private void txtNombrePacienteAgregaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnAgregaPacienteAgregaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if ("agregar".equals(evt.getActionCommand())) {
        if (this.cbEstadoPacienteAgrega.getSelectedIndex() == 0 || this.cbEdoCivilPacienteAgrega.getSelectedIndex() == 0 ||
            this.cbGeneroPacienteAgrega.getSelectedIndex() == 0 || this.txtCpPacienteAgrega.getText().length() < 1 ||
            this.txtNoExPacienteAgrega.getText().length() < 1 || this.txtCallePacienteAgrega.getText().length() < 1 ||
            this.txtColoniaPacienteAgrega.getText().length() < 1 || this.txtCiudadPacienteAgrega.getText().length() < 1 ||
            this.txtCorreoPacienteAgrega.getText().length() < 1 || this.txtTelefonoPacienteAgrega.getText().length() < 1 ||
            this.txtOcupacionPacienteAgrega.getText().length() < 1 || this.txtApellidosPacienteAgrega.getText().length() < 1 ||
            this.txtNombrePacienteAgrega.getText().length() < 1) {
            JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
        }
        else if (this.txtTelefonoPacienteAgrega.getText().length() <= 9) {
            JOptionPane.showMessageDialog(this, "Ingrese telefono a 10 digitos", "Alerta", 2);
        }
        else if (this.txtCpPacienteAgrega.getText().length() <= 4) {
            JOptionPane.showMessageDialog(this, "Ingrese un codigo postal valido a 5 digitos", "Alerta", 2);
        }
        else {
            try {

```

```

this.con = new ConexionBD();
final Connection cn = this.con.getConexion();
int antes=0,despues=0;
try {
    this.st = cn.prepareStatement("select count(id_paciente) as conteo from paciente");
    this.rt = this.st.executeQuery();
    if (this.rt.next()) {
        antes=this.rt.getInt("conteo");
    }
    //final String sql = "INSERT INTO paciente
(nombre,apellidos,genero,edo_civil,ocupacion,tel,email,edo,ciudad,colonia,calle,numero_exterior,numero_interior,cp)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
    final String sql="{call paquetepaciente.addpaciente(?,?,?,?,?,?,?,?,?,?,?,?,?)}";
    CallableStatement st = cn.prepareCall(sql);
    //st.setString(1, this.superCtrl.getClave());
    st.setString(1, this.txtNombrePacienteAgrega.getText());
    st.setString(2, this.txtApellidosPacienteAgrega.getText());
    if (this.cbGeneroPacienteAgrega.getSelectedIndex() == 1) {
        this.g = 1;
    }
    else {
        this.g = 0;
    }
    st.setInt(3, this.g);
    st.setString(4, this.cbEdoCivilPacienteAgrega.getSelectedItem().toString().toUpperCase());
    st.setString(5, this.txtOcupacionPacienteAgrega.getText());
    st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgrega.getText()));
    st.setString(7, this.txtCorreoPacienteAgrega.getText());
    st.setString(8, this.cbEstadoPacienteAgrega.getSelectedItem().toString().toUpperCase());
    st.setString(9, this.txtCiudadPacienteAgrega.getText());
    st.setString(10, this.txtColoniaPacienteAgrega.getText());
    st.setString(11, this.txtCallePacienteAgrega.getText());
    st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgrega.getText()));
    if (this.txtNoInPacienteAgrega.getText().length() < 1) {
        st.setInt(13, 0);
    }
    else {
        st.setInt(13, Integer.parseInt(this.txtNoInPacienteAgrega.getText()));
    }
    st.setInt(14, Integer.parseInt(this.txtCpPacienteAgrega.getText()));
    System.out.println("llego aqui 1");
    boolean f = st.execute();
    /*(this.st = cn.prepareStatement(sql)).setString(1, this.txtNombrePacienteAgrega.getText());
    this.st.setString(2, this.txtApellidosPacienteAgrega.getText());
    if (this.cbGeneroPacienteAgrega.getSelectedIndex() == 1) {
        this.g = 1;
    }
    else {
        this.g = 0;
    }
    this.st.setInt(3, this.g);
    this.st.setString(4, this.cbEdoCivilPacienteAgrega.getSelectedItem().toString().toUpperCase());
    this.st.setString(5, this.txtOcupacionPacienteAgrega.getText());
    this.st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgrega.getText()));
    this.st.setString(7, this.txtCorreoPacienteAgrega.getText());
    this.st.setString(8, this.cbEstadoPacienteAgrega.getSelectedItem().toString().toUpperCase());
    this.st.setString(9, this.txtCiudadPacienteAgrega.getText());
    this.st.setString(10, this.txtColoniaPacienteAgrega.getText());

```

```

this.st.setString(11, this.txtCallePacienteAgregar.getText());
this.st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgregar.getText()));
if (this.txtNolnPacienteAgregar.getText().length() < 1) {
    this.st.setInt(13, 0);
}
else {
    this.st.setInt(13, Integer.parseInt(this.txtNolnPacienteAgregar.getText()));
}
this.st.setInt(14, Integer.parseInt(this.txtCpPacienteAgregar.getText()));
System.out.println("llego aqui 1");
this.res = this.st.executeUpdate();
/**/
this.rt = this.st.executeQuery("select count(id_paciente) as conteo from paciente");
if (this.rt.next()) {
    despues=this.rt.getInt("conteo");
}
System.out.println(antes);
System.out.println(despues);
if (antes!=despues) {
    System.out.println("llego aqui 2");
    this.rt = this.st.executeQuery("select * from paciente where rownum=1 order by id_paciente desc");
    System.out.println("llego aqui 2.5");
    //rt.beforeFirst();
    this.rt.next();
    System.out.println("llego aqui 3");

    this.last = this.rt.getInt(1);
    JOptionPane.showMessageDialog(this, "el id del paciente es: "+this.last, "Atenci\u00f3n", 1);
}
else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un problema al registrar", "Error", 0);
}
System.out.println("llego aqui 4");
if (cn != null) {
    cn.close();
}
}
catch (Throwable t) {
    if (cn != null) {
        try {
            cn.close();
        }
        catch (Throwable exception) {
            t.addSuppressed(exception);
        }
    }
    throw t;
}
this.con.desconectar();
this.rt.close();
this.st.close();
this.limpiar();
this.superCtrl.setClave(""+this.last);
this.superCtrl.muestraVentanaAgregarCuestionario(true);
}
catch (HeadlessException | NumberFormatException | SQLException ex4) {
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
}
}

```

```

    }
}
else if (JOptionPane.showConfirmDialog(null, "¿Esta seguro que desea modificar al paciente?\nESTA ACCION NO SE
PUEDE DESHACER", "Alerta!", 0, 2) == 0) {
    if (this.cbEstadoPacienteAgregar.getSelectedIndex() == 0 || this.cbEdoCivilPacienteAgregar.getSelectedIndex() == 0 ||
this.cbGeneroPacienteAgregar.getSelectedIndex() == 0 || this.txtCpPacienteAgregar.getText().length() < 1 ||
this.txtNoExPacienteAgregar.getText().length() < 1 || this.txtCallePacienteAgregar.getText().length() < 1 ||
this.txtColoniaPacienteAgregar.getText().length() < 1 || this.txtCiudadPacienteAgregar.getText().length() < 1 ||
this.txtCorreoPacienteAgregar.getText().length() < 1 || this.txtTelefonoPacienteAgregar.getText().length() < 1 ||
this.txtOcupacionPacienteAgregar.getText().length() < 1 || this.txtApellidosPacienteAgregar.getText().length() < 1 ||
this.txtNombrePacienteAgregar.getText().length() < 1) {
        JOptionPane.showMessageDialog(this, "No puede dejar campos en blanco", "Alerta", 2);
    }
    else if (this.txtTelefonoPacienteAgregar.getText().length() <= 9) {
        JOptionPane.showMessageDialog(this, "Ingresa telefono a 10 digitos", "Alerta", 2);
    }
    else if (this.txtCpPacienteAgregar.getText().length() <= 4) {
        JOptionPane.showMessageDialog(this, "Ingresa un codigo postal valido a 5 digitos", "Alerta", 2);
    }
    else {
        try {
            this.con = new ConexionBD();
            final Connection cn = this.con.getConnection();
            try {
                String sql = "UPDATE paciente SET
nombre=?,apellidos=?,genero=?,edo_civil=?,ocupacion=?,tel=?,email=?,edo=?,ciudad=?,colonia=?,calle=?,numero_exterior=?,num
ero_interior=?,cp=? where id_paciente=?";
                (this.st = cn.prepareStatement(sql)).setString(1, this.txtNombrePacienteAgregar.getText());
                this.st.setString(2, this.txtApellidosPacienteAgregar.getText());
                if (this.cbGeneroPacienteAgregar.getSelectedIndex() == 1) {
                    this.g = 1;
                }
                else {
                    this.g = 0;
                }
                this.st.setInt(3, this.g);
                this.st.setString(4, this.cbEdoCivilPacienteAgregar.getSelectedItem().toString().toUpperCase());
                this.st.setString(5, this.txtOcupacionPacienteAgregar.getText());
                this.st.setLong(6, Long.parseLong(this.txtTelefonoPacienteAgregar.getText()));
                this.st.setString(7, this.txtCorreoPacienteAgregar.getText());
                this.st.setString(8, this.cbEstadoPacienteAgregar.getSelectedItem().toString().toUpperCase());
                this.st.setString(9, this.txtCiudadPacienteAgregar.getText());
                this.st.setString(10, this.txtColoniaPacienteAgregar.getText());
                this.st.setString(11, this.txtCallePacienteAgregar.getText());
                this.st.setInt(12, Integer.parseInt(this.txtNoExPacienteAgregar.getText()));
                if (this.txtNolnPacienteAgregar.getText().length() < 1) {
                    this.st.setInt(13, 0);
                }
                else {
                    this.st.setInt(13, Integer.parseInt(this.txtNolnPacienteAgregar.getText()));
                }
                this.st.setInt(14, Integer.parseInt(this.txtCpPacienteAgregar.getText()));
                this.st.setString(15, this.superCtrl.getClave());
                this.st.executeUpdate();
                sql = "SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC";
                //select * from paciente where rownum=1 order by id_paciente desc;
                (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
                //(this.rt = this.st.executeQuery()).beforeFirst();
            }
        }
    }
}

```

```

        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.last = this.rt.getInt(1);
            JOptionPane.showMessageDialog(this, "Se modifico", "Atenci\u00f3n", 1);
        }
        else {
            JOptionPane.showMessageDialog(this, "Ha ocurrido un error al modificar", "Error", 0);
        }
        if (cn != null) {
            cn.close();
        }
    }
    catch (Throwable t2) {
        if (cn != null) {
            try {
                cn.close();
            }
            catch (Throwable exception2) {
                t2.addSuppressed(exception2);
            }
        }
        throw t2;
    }
    this.con.desconectar();
    this.rt.close();
    this.st.close();
    this.limpiar();
    this.superCtrl.muestraVentanaPacientes(true);
}
catch (HeadlessException | NumberFormatException | SQLException ex4) {
    JOptionPane.showMessageDialog(null, ex4.getMessage(), "Error", 0);
}
}
}

}
/**
 * @param args the command line arguments
 */

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
    catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(VentanaAgregarPaciente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(VentanaAgregarPaciente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    catch (IllegalAccessException ex) {

```



```

        java.util.logging.Logger.getLogger(VentanaAgregarPaciente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(VentanaAgregarPaciente.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VentanaAgregarPaciente().setVisible(true);
    }
});
}

public void actionPerformed(ActionEvent e) {

//"INSERT INTO paciente
(nombre,apellidos,genero,edo_civil,ocupacion,tel,email,edo,ciudad,colonia,calle,numero_exterior,numero_interior,cp)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?)'

//UPDATE paciente SET
nombre=?,apellidos=?,genero=?,edo_civil=?,ocupacion=?,tel=?,email=?,edo=?,ciudad=?,colonia=?,calle=?,numero_exterior=?,num
ero_interior=?,cp=? where id_paciente=?'
//SELECT * from paciente where id_paciente=? order by id_paciente DESC LIMIT 1'

}

private void limpiar() {
    this.txtNombrePacienteAgregar.setText(null);
    this.txtApellidosPacienteAgregar.setText(null);
    this.txtOcupacionPacienteAgregar.setText(null);
    this.txtTelefonoPacienteAgregar.setText(null);
    this.txtCorreoPacienteAgregar.setText(null);
    this.txtCiudadPacienteAgregar.setText(null);
    this.txtColoniaPacienteAgregar.setText(null);
    this.txtCallePacienteAgregar.setText(null);
    this.txtNoExPacienteAgregar.setText(null);
    this.txtNoInPacienteAgregar.setText(null);
    this.txtCpPacienteAgregar.setText(null);
    this.cbEdoCivilPacienteAgregar.setSelectedIndex(0);
    this.cbEstadoPacienteAgregar.setSelectedIndex(0);
    this.cbGeneroPacienteAgregar.setSelectedIndex(0);
}

private void llenar_combo_actividad() {
    try {
        final String sql="select nombre from estados";
        ConexionBD con2 =new ConexionBD();
        final Connection cn = con2.getConnection();
        PreparedStatement st= cn.prepareStatement(sql);
        ResultSet rt = st.executeQuery();
        while(rt.next()){
            cbEstadoPacienteAgregar.addItem(rt.getString(1));
        }
        con2.desconectar();
    }catch(Exception e){
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta de estados", "Alerta", 2);
        e.printStackTrace();
    }
}

```

```

}
public void modificar() {
    this.btnAgregarPacienteAgregar.setText("Modificar");
    this.btnAgregarPacienteAgregar.setActionCommand("modificar");
    this.btnCancelarPacienteAgregar.setActionCommand("modificar");
    try {
        this.con = new ConexionBD();
        final Connection cn = this.con.getConnection();
        final String sql = "SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC";
        //SELECT * from paciente where id_paciente=? and rownum=1 order by id_paciente DESC
        (this.st = cn.prepareStatement(sql)).setString(1, this.superCtrl.getClave());
        this.rt = this.st.executeQuery();
        if (this.rt.next()) {
            this.txtNombrePacienteAgregar.setText(this.rt.getString("nombre"));
            this.txtApellidosPacienteAgregar.setText(this.rt.getString("apellidos"));
            this.txtOcupacionPacienteAgregar.setText(this.rt.getString("ocupacion"));
            this.txtTelefonoPacienteAgregar.setText("" + this.rt.getLong("tel"));
            this.txtCorreoPacienteAgregar.setText(this.rt.getString("email"));
            this.txtCiudadPacienteAgregar.setText(this.rt.getString("ciudad"));
            this.txtColoniaPacienteAgregar.setText(this.rt.getString("colonia"));
            this.txtCallePacienteAgregar.setText(this.rt.getString("calle"));
            this.txtNoExPacienteAgregar.setText("" + this.rt.getInt("numero_exterior"));
            if (this.rt.getInt("numero_interior") != 0) {
                this.txtNoInPacienteAgregar.setText("" + this.rt.getInt("numero_interior"));
            }
            else {
                this.txtNoInPacienteAgregar.setText(null);
            }
            this.txtCpPacienteAgregar.setText("" + this.rt.getInt("cp"));
            if (this.rt.getInt("genero") == 1) {
                this.cbGeneroPacienteAgregar.setSelectedItem("H");
            }
            else {
                this.cbGeneroPacienteAgregar.setSelectedItem("M");
            }
            this.cbEdoCivilPacienteAgregar.setSelectedItem(this.formato(this.rt.getString("edo_civil")));
            this.cbEstadoPacienteAgregar.setSelectedItem(this.formato(this.rt.getString("edo")));
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Ocurrio un error durante la consulta", "Alerta", 2);
    }
}

public void agregar() {
    this.btnAgregarPacienteAgregar.setText("Agregar");
    this.btnAgregarPacienteAgregar.setActionCommand("agregar");
    this.btnCancelarPacienteAgregar.setActionCommand("agregar");
}

private String formato(String string) {
    string = string.toLowerCase();
    char[] caracteres = string.toCharArray();
    caracteres[0] = Character.toUpperCase(caracteres[0]);

    for (int i = 0; i < string.length() - 2; i++) {
        if (caracteres[i] == ' ' || caracteres[i] == '.' || caracteres[i] == ',') {

```

```

        caracteres[i + 1] = Character.toUpperCase(caracteres[i + 1]);
    }
}
return new String(caracteres);
}

```

```

// Variables declaration - do not modify
private javax.swing.JButton btnAgregarPacienteAgregar;
private javax.swing.JButton btnCancelarPacienteAgregar;
private javax.swing.JComboBox<String> cbEdoCivilPacienteAgregar;
private javax.swing.JComboBox<String> cbEstadoPacienteAgregar;
private javax.swing.JComboBox<String> cbGeneroPacienteAgregar;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel lblApellidosPacienteAgregar;
private javax.swing.JLabel lblCallePacienteAgregar;
private javax.swing.JLabel lblCiudadPacienteAgregar;
private javax.swing.JLabel lblColoniaPacienteAgregar;
private javax.swing.JLabel lblCorreoPacienteAgregar;
private javax.swing.JLabel lblCpPacienteAgregar;
private javax.swing.JLabel lblEdoCivilPacienteAgregar;
private javax.swing.JLabel lblEstadoPacienteAgregar;
private javax.swing.JLabel lblGeneroPacienteAgregar;
private javax.swing.JLabel lblNoExPacienteAgregar;
private javax.swing.JLabel lblNoInPacienteAgregar;
private javax.swing.JLabel lblNombrePacienteAgregar;
private javax.swing.JLabel lblOcupacionPacienteAgregar;
private javax.swing.JLabel lblTelefonoPacienteAgregar;
private javax.swing.JTextField txtApellidosPacienteAgregar;
private javax.swing.JTextField txtCallePacienteAgregar;
private javax.swing.JTextField txtCiudadPacienteAgregar;
private javax.swing.JTextField txtColoniaPacienteAgregar;
private javax.swing.JTextField txtCorreoPacienteAgregar;
private javax.swing.JTextField txtCpPacienteAgregar;
private javax.swing.JTextField txtNoExPacienteAgregar;
private javax.swing.JTextField txtNoInPacienteAgregar;
private javax.swing.JTextField txtNombrePacienteAgregar;
private javax.swing.JTextField txtOcupacionPacienteAgregar;
private javax.swing.JTextField txtTelefonoPacienteAgregar;
// End of variables declaration
}

```

A continuación, se mostrará el código de entidades que no tienen ventanas.

```
package GUI;

import Control.ControlInter;
import java.awt.event.ActionListener;
import javax.swing.JFrame;

public abstract class VentanaControladaAbstract
    extends JFrame
    implements ActionListener
{
    protected ControlInter superCtrl;

    public VentanaControladaAbstract() {}

    public VentanaControladaAbstract(String t) { super(t); }

    public void setSuperCtrl(ControlInter ctrl) { this.superCtrl = ctrl; }
}
```

---

```
package Control;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class ValidaCorreo
{
    public boolean Valida(String email)
    {
        Pattern pattern = Pattern.compile("[a-z0-9]+(\\.[a-z0-9]+)*+@[a-z]+\\.([a-z]+)+");
        Matcher matcher = pattern.matcher(email);
        if (matcher.find() == true) {
            return true;
        }
        return false;
    }
}
```

---

```
package Control;

import GUI.VentanaAgregarPaciente;
import GUI.VentanaBuscarPaciente;
import GUI.VentanaCitas;
import GUI.VentanaCuestionario;
import GUI.VentanaPacientes;
import GUI.VentanaReportes;
import GUI.VentanaTablaReportes;
import GUI.citas2;
import GUI.lobby;
import GUI.buscar;
import GUI.buscarus;
import GUI.búsquedaCita;
import GUI.búsquedaCita1;
import GUI.creaus;
import GUI.login;
import GUI.usuarios;

public abstract interface ControlInter
{
}
```

```

public abstract void setVentanaLogin(login c);
public abstract void setVentanabusquedacita(busquedaCita c);
public abstract void muestraVentanaModificarCita(boolean b);
public abstract void muestraVentanabusquedacita(boolean b);
public abstract void setVentanabusquedacita1(busquedaCita1 c);
public abstract void muestraVentanabusquedacita1(boolean b);
public abstract void muestraVentanaLogin(boolean b);
public abstract void muestraVentanaBuscarP(buscar paramVentanaBuscarPaciente);
public abstract void muestraVentanaBuscarP(boolean paramBoolean);
public abstract void setVentanaLobby(lobby paramlobby);
public abstract void muestraVentanaLobby(boolean paramBoolean);
public abstract void setVentanaCitas(VentanaCitas paramVentanaCitas);
public abstract void muestraVentanaCitas(boolean paramBoolean);
public abstract void setVentanaPacientes(VentanaPacientes paramVentanaPacientes);
public abstract void muestraVentanaPacientes(boolean paramBoolean);
public abstract void setVentanaAgregarPaciente(VentanaAgregarPaciente paramVentanaAgregarPaciente);
public abstract void muestraVentanaAgregarPacientes(boolean paramBoolean);
public abstract void setVentanaAgregarCuestionario(VentanaCuestionario paramVentanaCuestionario);
public abstract void muestraVentanaAgregarCuestionario(boolean paramBoolean);
public abstract void setVentanaBuscarPaciente(VentanaBuscarPaciente paramVentanaBuscarPaciente);
public abstract void muestraVentanaBuscarPacientes(boolean paramBoolean);
public abstract void muestraVentanaEliminarPacientes(boolean paramBoolean);
public abstract void muestraVentanaModificarPacientes(boolean paramBoolean);
public abstract void setVentanaReportes(VentanaReportes paramVentanaReportes);
public abstract void muestraVentanaReportes(boolean paramBoolean);
public abstract void setVentanaTablaReportes(VentanaTablaReportes paramVentanaTablaReportes);
public abstract void muestraTablaVentanaReportes(String paramString);
public abstract void setcita(citas2 paramcitas2);
public abstract void muestracita(boolean paramBoolean);
public abstract void setClave(String paramString);
public abstract String getClave();
public abstract void setUser(String user);
public abstract String getUser();
public abstract void setVentanaus(usuarios c);
public abstract void muestraVentanaus(boolean b);
public abstract void setbuscarus(buscarus c);
public abstract void muestrabuscarus(boolean b);
public abstract void setcreaus(creaus c);
public abstract void muestracreaus(boolean b);

```

```

}

```

---

```

package Control;

```

```

import GUI.VentanaAgregarPaciente;
import GUI.VentanaBuscarPaciente;
import GUI.VentanaCitas;
import GUI.VentanaCuestionario;
import GUI.VentanaPacientes;
import GUI.VentanaReportes;
import GUI.VentanaTablaReportes;
import GUI.buscar;
import GUI.buscarus;
import GUI.busquedaCita;
import GUI.busquedaCita1;
import GUI.citas2;
import GUI.creaus;
import GUI.lobby;
import GUI.login;

```

```

import GUI.usuarios;

public class ControlApp
implements ControlInter
{
    private busquedaCita vbcita;
    private busquedaCita1 vbcita1;
    private VentanaCitas vcitas;
    private VentanaPacientes vpaciente;
    private VentanaAgregarPaciente vapaciente;
    private VentanaBuscarPaciente vbpaciente;
    private VentanaCuestionario vCuestionario;
    private VentanaReportes vReportes;
    private VentanaTablaReportes vtReportes;
    private buscar vbuscar;
    private citas2 cita;
    private lobby vlobby;
    private String clave;
    private String user="user";
    private login vpass;
    private usuarios usuarios;
    private buscarus b;
    private creaus u;
    public void setVentanaus(usuarios c)
    {
        this.usuarios = c;
        this.usuarios.setSuperCtrl(this);
    }

    public void muestraVentanaus(boolean b)
    {
        this.vlobby.setVisible(!b);
        this.b.setVisible(!b);
        this.u.setVisible(!b);
        this.usuarios.setVisible(b);
    }
    public void setbuscarus(buscarus c)
    {
        this.b = c;
        this.b.setSuperCtrl(this);
    }

    public void muestrabuscarus(boolean b)
    {
        this.usuarios.setVisible(!b);
        this.b.setVisible(b);
    }
    public void setcreaus(creaus c)
    {
        this.u = c;
        this.u.setSuperCtrl(this);
    }

    public void muestracreaus(boolean b)
    {
        this.usuarios.setVisible(!b);
        this.u.setVisible(b);
    }
}

```

```
public void setVentanaLogin(login c)
{
    this.vpass = c;
    this.vpass.setSuperCtrl(this);
}

public void muestraVentanaLogin(boolean b)
{
    this.vpass.setVisible(b);
}

public void setVentanaLobby(lobby c)
{
    this.vlobby = c;
    this.vlobby.setSuperCtrl(this);
}

public void muestraVentanaLobby(boolean b)
{
    this.v citas.setVisible(!b);
    this.cita.setVisible(!b);
    this.vpaciente.setVisible(!b);
    this.vReportes.setVisible(!b);
    this.vpass.setVisible(!b);
    this.vlobby.setVisible(b);
    this.vlobby.consulta();
    this.usuarios.setVisible(!b);
}

public void setVentanaPacientes(VentanaPacientes c)
{
    this.vpaciente = c;
    this.vpaciente.setSuperCtrl(this);
}

public void muestraVentanaPacientes(boolean b)
{
    this.vlobby.setVisible(!b);
    this.vpaciente.consulta();
    this.vpaciente.setVisible(b);
    this.vapaciente.setVisible(!b);
    this.vbpaciente.setVisible(!b);
    this.vbuscar.setVisible(!b);
    this.vCuestionario.setVisible(!b);
}

public void setVentanaAgregarPaciente(VentanaAgregarPaciente c)
{
    this.vapaciente = c;
    this.vapaciente.setSuperCtrl(this);
}

public void muestraVentanaAgregarPacientes(boolean b)
{
    this.vapaciente.setVisible(b);
    this.vpaciente.setVisible(!b);
}
```

```

        this.vapaciente.agregar();
    }

    public void setVentanaBuscarPaciente(VentanaBuscarPaciente c)
    {
        this.vbpaciente = c;
        this.vbpaciente.setSuperCtrl(this);
    }

    public void muestraVentanaBuscarPacientes(boolean b)
    {
        this.vbpaciente.setVisible(b);
        this.vpaciente.setVisible(!b);
        this.vbpaciente.buscar(b);
    }

    public void setVentanaBuscarP(buscar c)
    {
        this.vbuscar = c;
        this.vbuscar.setSuperCtrl(this);
    }

    public void muestraVentanaBuscarP(boolean b)
    {
        this.vbuscar.setVisible(b);
        this.vpaciente.setVisible(!b);
        //this.vbuscar.buscar(b);
    }

    public void muestraVentanaEliminarPacientes(boolean b)
    {
        this.vbpaciente.eliminar(b);
        this.vbpaciente.setVisible(b);
        this.vpaciente.setVisible(!b);
    }

    public void muestraVentanaModificarPacientes(boolean b)
    {
        if (b)
        {
            //this.vbpaciente.modificar(b);

            //this.vbpaciente.setVisible(b);
            this.vbuscar.setVisible(b);
            this.vpaciente.setVisible(!b);
            this.vapaciente.setVisible(!b);
        }
        else
        {
            //this.vbpaciente.setVisible(b);
            this.vbuscar.setVisible(b);
            this.vapaciente.modificar();
            this.vapaciente.setVisible(!b);
            this.vpaciente.setVisible(b);
        }
    }

    public void setClave(String clave)
    {
        this.clave = clave;
    }

```



```

    }

    public String getClave()
    {
        return this.clave;
    }
    public void setUser(String user)
    {
        this.user = user;
    }

    public String getUser()
    {
        return this.user;
    }
    public void setVentanaAgregarCuestionario(VentanaCuestionario c)
    {
        this.vCuestionario = c;
        this.vCuestionario.setSuperCtrl(this);
    }

    public void muestraVentanaAgregarCuestionario(boolean b)
    {
        if (b)
        {
            this.vpaciente.setVisible(!b);
            this.vCuestionario.setVisible(b);
        }
        else
        {
            this.vCuestionario.setVisible(b);
            muestraVentanaPacientes(!b);
        }
    }

    public void setVentanaReportes(VentanaReportes c)
    {
        this.vReportes = c;
        this.vReportes.setSuperCtrl(this);
    }

    public void muestraVentanaReportes(boolean b)
    {
        this.vlobby.setVisible(!b);
        this.vtReportes.setVisible(!b);
        this.vReportes.setVisible(b);
    }

    public void setVentanaTablaReportes(VentanaTablaReportes c)
    {
        this.vtReportes = c;
        this.vtReportes.setSuperCtrl(this);
    }

    public void muestraTablaVentanaReportes(String ventana)
    {
        /**
        if (ventana != null) {

```

```

switch (ventana)
{
case "pacientes" -> {
    this.vtReportes.paciente();
    this.vReportes.setVisible(false);
    this.vtReportes.setVisible(true);
}
case "adeudo" -> {
    this.vtReportes.adeudo();
    this.vReportes.setVisible(false);
    this.vtReportes.setVisible(true);
}
case "cita1" -> {
    this.vtReportes.citas();
    this.vReportes.setVisible(false);
    this.vtReportes.setVisible(true);
}
case "cita2" -> {
    this.vtReportes.citasxfecha();
    this.vReportes.setVisible(false);
    this.vtReportes.setVisible(true);
}
}
}

/**/
}

public void setcita(citas2 c)
{
    this.cita = c;
    this.cita.setSuperCtrl(this);
}

public void muestracita(boolean b)
{
    //this.vlobby.setVisible(!b);
    this.vcitas.setVisible(!b);
    //this.vbcita.setVisible(!b);
    this.vbcita1.setVisible(!b);
    this.cita.agregar();
    this.cita.setVisible(b);
}

public void muestraVentanaModificarCita(boolean b)
{
    if (b)
    {
        this.vbcita.setVisible(!b);
        this.vbcita1.setVisible(!b);
        this.cita.modificar();
        this.cita.setVisible(b);
    }
    /*else
    {
        //this.vbpaciente.setVisible(b);
        this.vbuscar.setVisible(b);
        this.vapaciente.modificar();
    }
    */
}

```

```

        this.vapaciente.setVisible(!b);
        this.vpaciente.setVisible(b);
    }/*/
}

public void setVentanabusquedacita(busquedaCita c)
{
    this.vbcita = c;
    this.vbcita.setSuperCtrl(this);
}

public void muestraVentanabusquedacita(boolean b)
{
    this.vcitas.setVisible(!b);
    this.vbcita.setVisible(b);
}

public void setVentanabusquedacita1(busquedaCita1 c)
{
    this.vbcita1 = c;
    this.vbcita1.setSuperCtrl(this);
}

public void muestraVentanabusquedacita1(boolean b)
{
    this.vcitas.setVisible(!b);
    this.vbcita1.setVisible(b);
}

public void setVentanaCitas(VentanaCitas c)
{
    this.vcitas = c;
    this.vcitas.setSuperCtrl(this);
}

}

public void muestraVentanaCitas(boolean b)
{
    this.vcitas.setVisible(b);
    this.cita.setVisible(!b);
    this.vbcita.setVisible(!b);
    this.vbcita1.setVisible(!b);
    this.vlobby.setVisible(!b);
}
}

```

---

```
package Conexion;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

```

```
public class ConexionBD {
```

```

    private final String stringConexion = "jdbc:oracle:thin:@server.oracle.com:1521/xepdb1";
    private final String User = "dentista";
    private final String Pass = "123";
    private Connection conexion;
    //Metodo para acceder a la BD

```

```
    public ConexionBD() {
```

```

        conexion = null;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conexion = DriverManager.getConnection(stringConexion, User, Pass);
    } catch (ClassNotFoundException e) {
        System.out.println("No se encuentra el driver");
    } catch (SQLException e) {
        System.out.println("Error en la conexion");
        e.printStackTrace();
    }
    //return conexion;
}

public Connection getConexion() {

    return this.conexion;
}

public void desconectar() {
    this.conexion = null;
}

public static void main(String[] args) {
    if (new ConexionBD().getConexion() != null) {
        System.out.println("Conexion exitosa");
    } else {
        System.out.println("Conexion fallida");
    }
}
}

```

---

```

package Dentista;

```

```

import Control.ControlApp;
import Control.ControlInter;
import GUI.VentanaAgregarPaciente;
import GUI.VentanaBuscarPaciente;
import GUI.VentanaCitas;
import GUI.VentanaCuestionario;
import GUI.VentanaPacientes;
import GUI.VentanaReportes;
import GUI.VentanaTablaReportes;
import GUI.buscar;
import GUI.buscarus;
import GUI.búsquedaCita;
import GUI.búsquedaCita1;
import GUI.citas2;
import GUI.creaus;
import GUI.lobby;
import GUI.login;
import GUI.usuarios;

public class Main
{
    public static void main(String[] args)
    {

```

```

lobby vLobby = new lobby();
VentanaPacientes vPaciente = new VentanaPacientes();
VentanaCitas vCita = new VentanaCitas();
VentanaBuscarPaciente vbPaciente = new VentanaBuscarPaciente();
buscar vbuscar=new buscar();
VentanaAgregarPaciente vaPaciente = new VentanaAgregarPaciente();
VentanaCuestionario vCuestionario = new VentanaCuestionario();
VentanaReportes vReportes = new VentanaReportes();
VentanaTablaReportes vtReportes = new VentanaTablaReportes();
login vpass=new login();
citas2 cita = new citas2();
busquedaCita bcita=new busquedaCita();
busquedaCita1 bcita1=new busquedaCita1();
usuarios u=new usuarios();
buscarus b=new buscarus();
creaus cu =new creaus();
cu.pack();
b.pack();
u.pack();
bcita.pack();
bcita1.pack();
cita.pack();
vtReportes.pack();
vReportes.pack();
vCuestionario.pack();
vaPaciente.pack();
vbPaciente.pack();
vbuscar.pack();
vLobby.pack();
vPaciente.pack();
vCita.pack();
vpass.pack();

ControlInter ctrlPrincipal = new ControlApp();
ctrlPrincipal.setVentanabusquedacita(bcita);

ctrlPrincipal.setVentanabusquedacita1(bcita1);
ctrlPrincipal.setVentanaLogin(vpass);
ctrlPrincipal.setVentanaus(u);
ctrlPrincipal.setVentanaBuscarP(vbuscar);
ctrlPrincipal.setVentanaLobby(vLobby);
ctrlPrincipal.setVentanaCitas(vCita);
ctrlPrincipal.setbuscarus(b);
ctrlPrincipal.setcreaus(cu);
ctrlPrincipal.setVentanaPacientes(vPaciente);
ctrlPrincipal.setVentanaBuscarPaciente(vbPaciente);
ctrlPrincipal.setVentanaAgregarPaciente(vaPaciente);
ctrlPrincipal.setVentanaAgregarCuestionario(vCuestionario);
ctrlPrincipal.setVentanaReportes(vReportes);
ctrlPrincipal.setVentanaTablaReportes(vtReportes);
ctrlPrincipal.setcita(cita);

ctrlPrincipal.muestraVentanaLogin(true);
}
}

```

## REFERENCIAS

- Archivos y presentaciones de Moodle
  - Architecture.pdf
  - Backups.pdf
  - BR\_concepts.pdf
  - Data.pdf
  - DetectingLocks.pdf
  - Instance.pdf
  - Network.pdf
  - Object\_Type.pdf
  - Performance.pdf
  - plsql.pdf
  - prac\_PLSQL.pdf
  - prac\_Schemas.pdf
  - prac\_Storage.pdf
  - prac\_UsingSQL.pdf
  - practica\_Users.pdf
  - Schema.pdf
  - Storage.pdf
  - Tuning.pdf
  - Users.pdf
- <https://docs.oracle.com/en/cloud/paas/exadata-express-cloud/csdbp/create-database-schemas.html>
- [CREATE TABLESPACE \(oracle.com\)](#)
- [Oracle 12c: crear un tablespace \(pchardwarepro.com\)](#)
- [¿Cómo mostrar todos los privilegios de un usuario en Oracle? \(qastack.mx\)](#)
- [SQL\\*Loader - Step by Step Guide How to Load a Datafile Into a Table \(oracletutorial.com\)](#)
- [SQL\\*Loader Concepts \(oracle.com\)](#)
- [SQL\\*Loader Command-Line Reference \(oracle.com\)](#)
- [How to Restore Oracle Database using RMAN \(with Examples\) \(thegeekstuff.com\)](#)
- <https://docs.oracle.com/en/database/oracle/oracle-database/18/bradv/getting-started-rman.html>
- [El Modo Archivado \(ArchiveLog Mode\) \(oraxedatabase.blogspot.com\)](#)
- [Realizar copia de seguridad con RMAN \(oraxedatabase.blogspot.com\)](#)
- [PreparedStatement \(Java Platform SE 7 \) \(oracle.com\)](#)
- [CallableStatement \(Java Platform SE 7 \) \(oracle.com\)](#)
- [javadoc \(oracle.com\)](#)