

UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE



Object-oriented programming

NRC: 9642

U2 Workshop N° 31

Topic: Project Revision Class

ING. Jorge Edison Lascano.

Main Rubric

GitHub	/100
GoogleDoc	/10
MongoDB vs CD	/10
Design (ClassDiagram&UseCae vs SRS)	/10
MVC	/10
Polymorphism	/10
Test Cases/UnitTest	/10
GUI	/20
CRUD/Bussiness Rule	/20
TOTAL:	/200
@everyone	

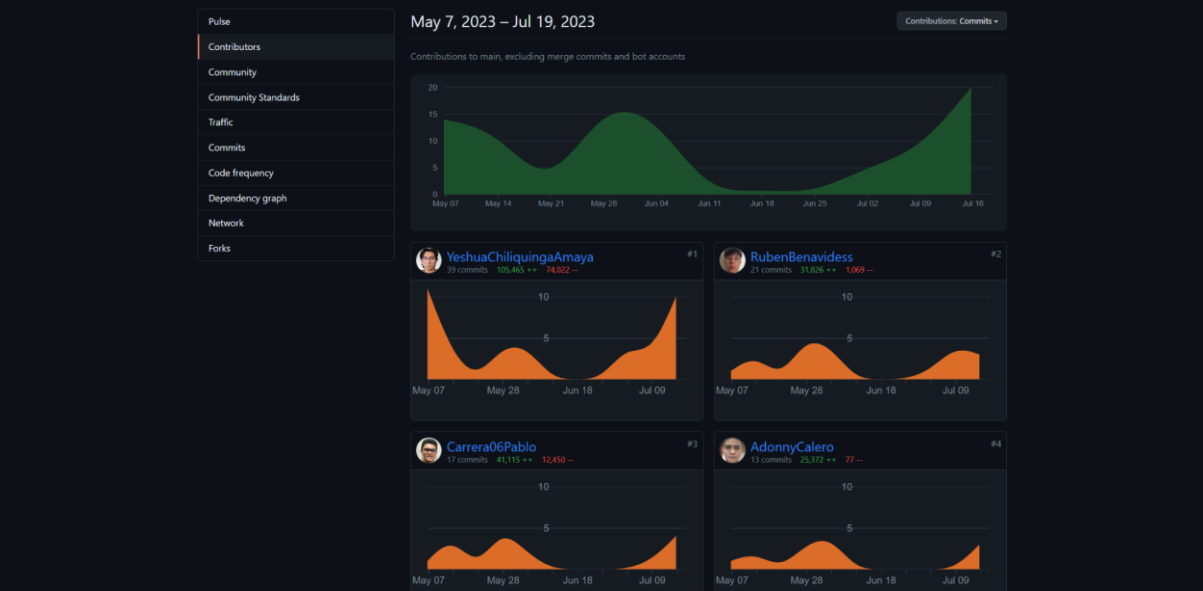
Team 1

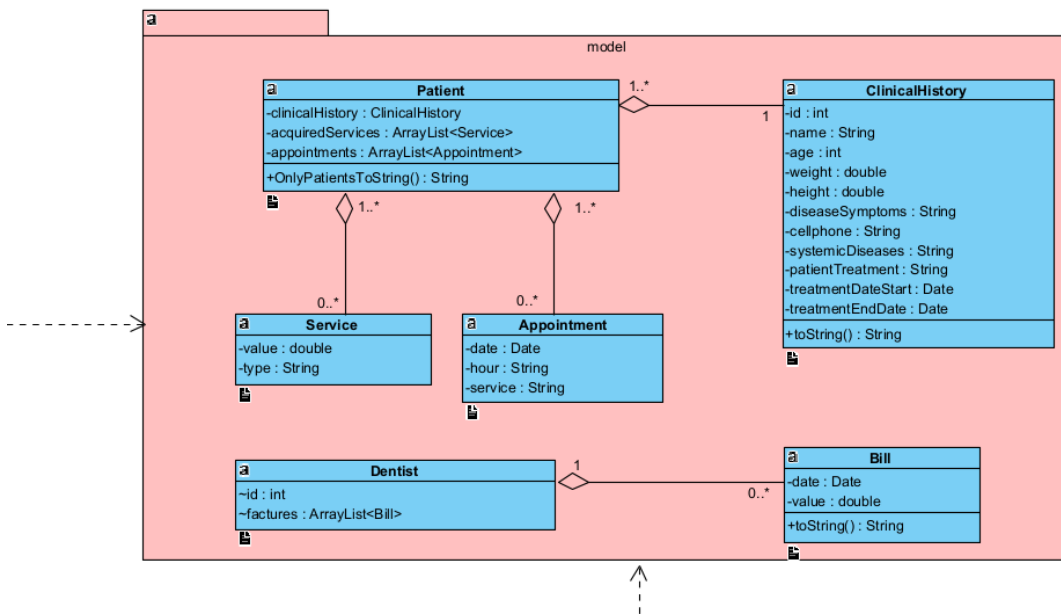
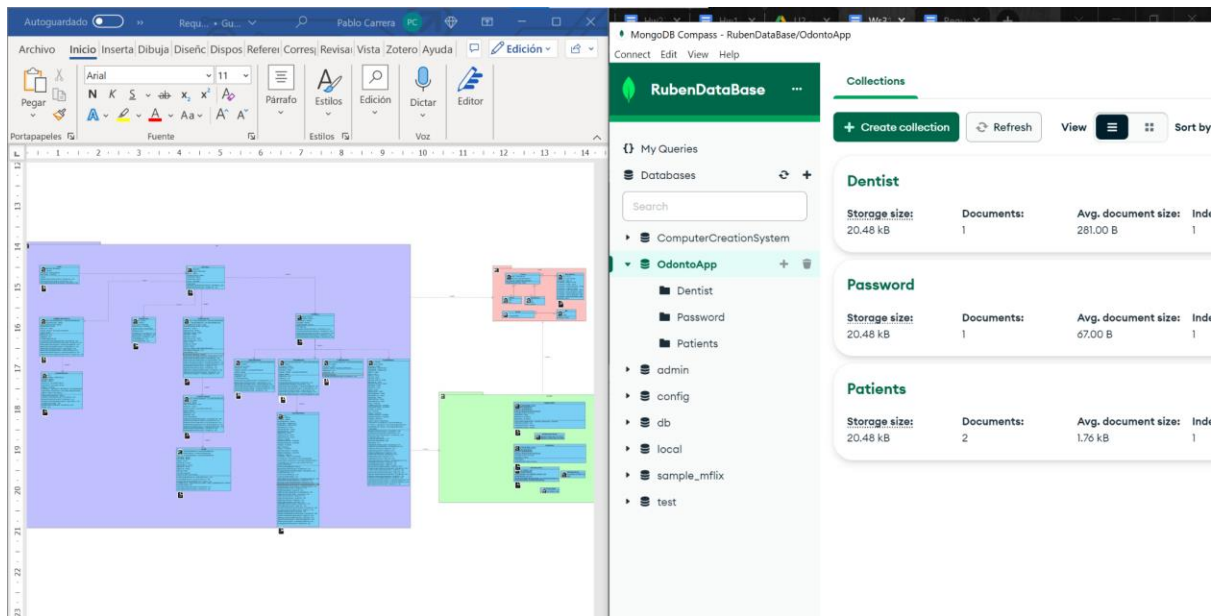
1. RUBEN DARIO BENAVIDES MACIAS
2. ADONNY MATEO CALERO ARGUELLO
3. PABLO ESTEBAN CARRERA QUINDE
4. YESHUA AMADOR CHILQUINGA AMAYA

Format:

Main Rubric

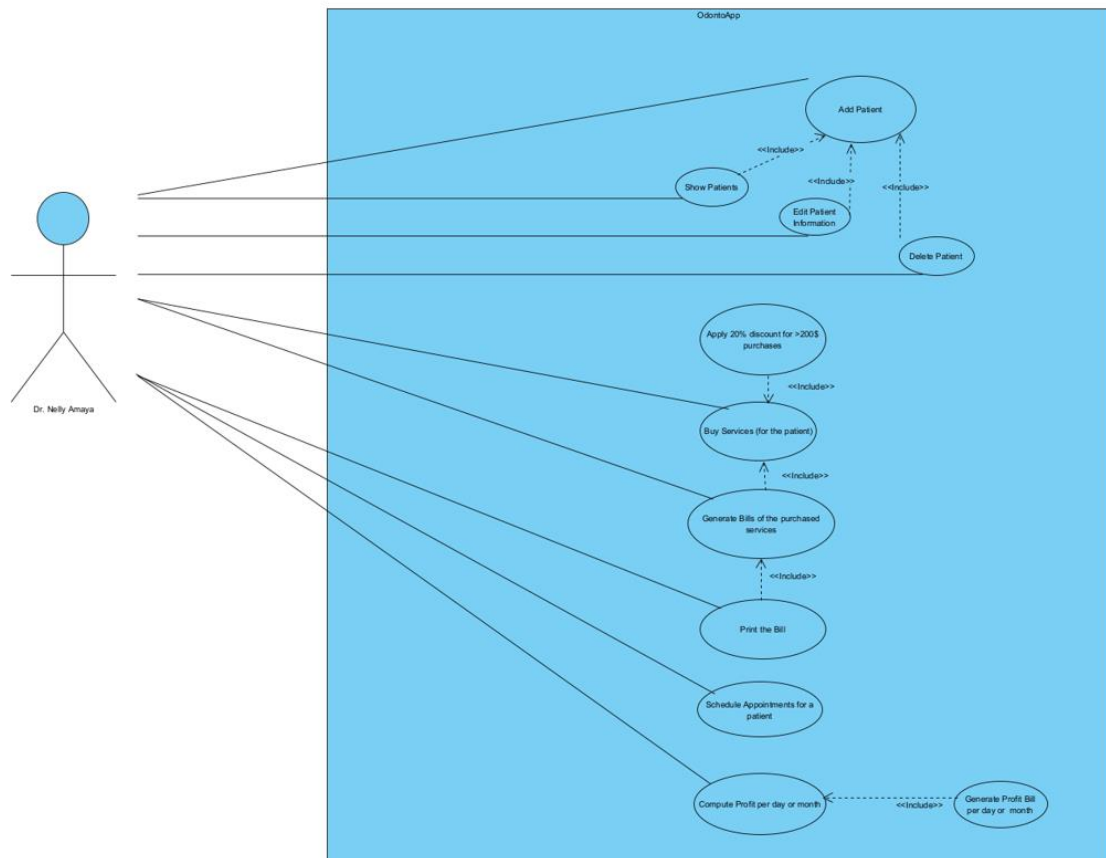
GitHub	100/100
GoogleDoc	9/10
MongoDB vs CD	10/10
Design (ClassDiagram&UseCae vs SRS)	/10
MVC	/10
Polymorphism	/10
Test Cases/UnitTest	/10
GUI	/20
CRUD/Bussiness Rule	/20
TOTAL:	/200
Evidence:	





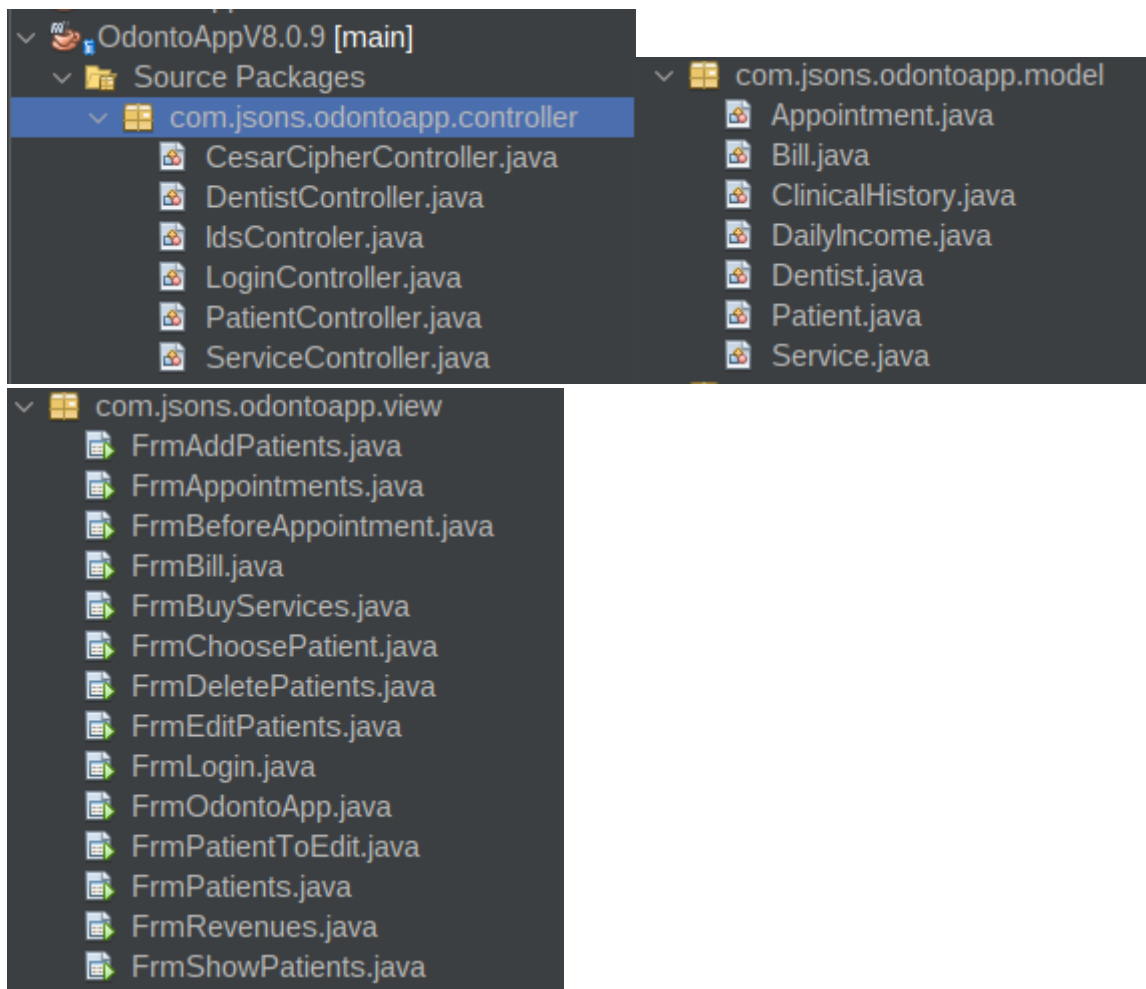
Design (ClassDiagram&UseCae vs SRS)

8/10



MVC

10/10



Polymorphism

8/10

```

public class DailyIncome {
    private Date date;
    private double income;

    @Override
    public String toString() {
        return "DailyIncome{" + "date=" + date + ", income=" + income + '}';
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }
}

```

```

    @Override
    public String toString() {
        return name + "\n" +
            "id: " + id + "\n" +
            age + "\n" +
            weight + "\n" +
            height + "\n" +
            diseaseSymptoms + "\n" +
            cellphone + "\n" +
            systemicDiseases + "\n" +
            patientTreatment + "\n" +
            "Fecha de Inicio de Tratamiento: " + treatmentDateStart.toString() +
            "Fecha de Finalización de Tratamiento: " + treatmentEndDate.toString()
    }
}

```

```

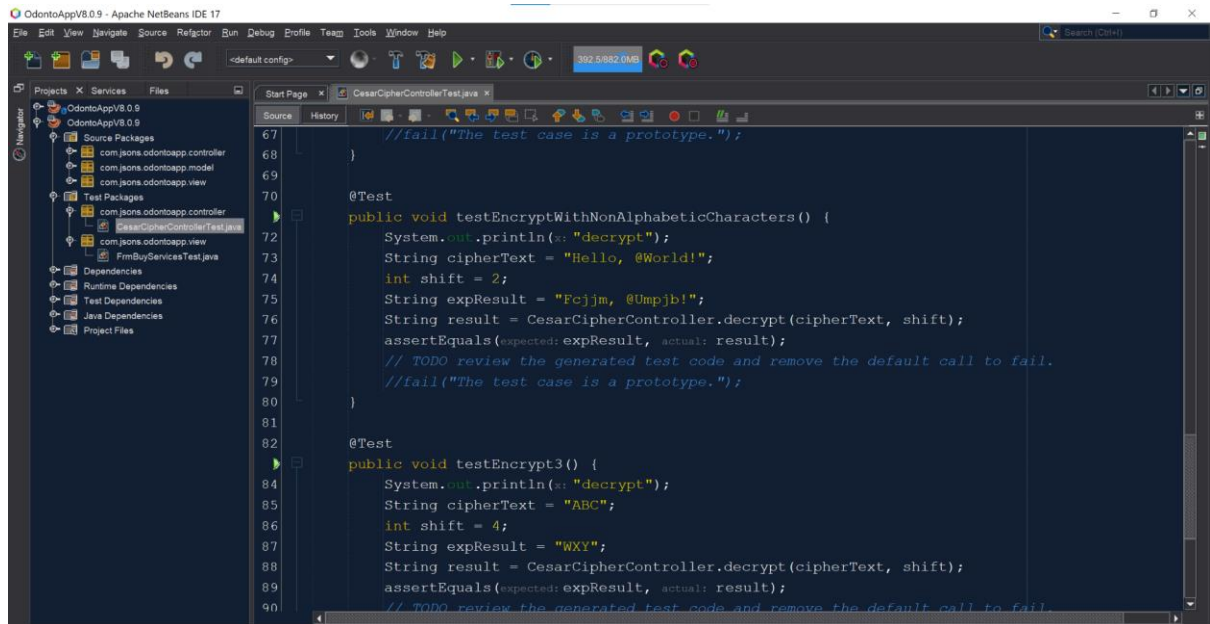
public class Bill {
    private Date date;
    private double value;

    @Override
    public String toString() {
        return "Bill{" + "date=" + getDate() + ", value=" + getValue() + '}';
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
    }
}

```



GUI

10/10



Odonto App

By Jsons Systems

Inicio de se...

Usuario:

Contraseña:

☐

Mostrar Contraseña

Entrar



ODONTO APP

By J S O N S S Y S T E M S

ID	Nombre	Edad	Peso	Altura	Síntomas de En...	Número de con...	Enfermedades ...	Tratamiento del...	Fecha de Inicio ...	Fecha de Finaliz...
2	Lucía	22	60.5	150.0	Dolor muela su...	123123123	Diabetes	Profilaxis	Sat Jul 08 20:50...	Tue Jul 25 20:5...
12345	Dario Gomez	30	50.69	1.8	Incrustacion	1234567890	Diabetes Tipo II	Extraccion Molar	Mon Jul 17 00:0...	Fri Jul 21 00:00:...

[Regresar](#)

Servicio

Profilaxis (limpieza dental): \$25

Profilaxis + Fluorización (especialmente para niños): \$20

Restauraciones simples: \$20

Restauraciones complejas (Las restauraciones pueden ser con resina o amalgama): \$25

Restauraciones para niños (empastes para niños): \$15

Extracciones (extracción de un diente): \$20

Extracciones complicadas: \$30

Tratamiento de un conducto (incisivos o dientes frontales): \$60

Tratamiento de dos conductos (premolares): \$80

Tratamiento de tres conductos (molares): \$100

Añadir al carrito

Añadido

Servicio

Tratamiento de tres conductos (molares)

Restauraciones complejas

Tratamiento de tres conductos (molares)

Eliminar del carrito

Comprar

Descuento de 20% Disponible!

Regresar

Advertencia de compra

?

Está segura de que el paciente Lucía desea comprar estos servicios:
Servicio: Tratamiento de tres conductos (molares), Valor: 100.0
Servicio: Restauraciones complejas, Valor: 25.0
Servicio: Tratamiento de tres conductos (molares), Valor: 100.0
+DESCUENTO DEL 20% AL TOTAL DE LAS COMPRAS

Cancel

No

Yes

SourceRefactorRunDebugProfileTeamToolsWindowHelp

<default confi...347.0/454.0MB

Factura

Fec... 19/07/2023

Nombre del paciente: Lucía

Compras Realizadas

Servicio	Costo
Tratamiento de tres conductos (molares)	100.0
Restauraciones complejas	25.0
Tratamiento de tres conductos (molares)	100.0

Total

225.0

Descu...

20%

Precio Final: 180.0

Imprimir Factura

Regresar Al Menu Principal

oved(Ancestoi[Bill{date=Mon Jul 17 23:09:20 ECT 2023, value=20.0}, Bill{date=Mon Jul 17 23:19:10 L[Bill{date=Mon Jul 17 23:09:20 ECT 2023, value=20.0}, Bill{date=Mon Jul 17 23:19:10

<-- Regresar

Ganancias al Día

Calculadora de Ganancia

July2023

SunMonTueWedThuFriSat

261

272345678

289101112131415

2916171819202122

3023242526272829

313031

☒ Activar Ganancias al Mes

July2023July2023

SunMonTueWedThuFriSatSunMonTueWedThuFriSat

261261

272345678272345678

289101112131415289101112131415

29161718192021222916171819202122

30232425262728293023242526272829

313031313031

Factura de ganancias del Jul 17, 2023 al Jul 19, 2023

Servicio	Monto
Servicio	\$20.0
Servicio	\$208.0
Servicio	\$224.0
Servicio	\$240.0
Servicio	\$180.0

Total ganancias: \$872.0

Mostrar Factura de gancias

Generar Cita
Ver cita Pendiente

ID	Nombre	Edad	Peso	Altura	Sínto...	Núme...	Enfer...	Trata...	Fech...	Fech...
2	Lucía	22	60.5	150.0	Dolor ...	1231...	Diabe...	Profil...	Sat Ju...	Tue J...
12345	Dario ...	30	50.69	1.8	Incru...	1234...	Diabe...	Extra...	Mon J...	Fri Jul...

Citas del Paciente: Lucía

Fecha: Fri Jul 28 00:00:00 ECT 2023

Hora: 3:00 pm

Servicio: Extracciones complicadas: \$30

=====

OK

Regresar
Siguiete Página

Agendamiento de Citas a pacientes

Día

Información del Paciente:

ID: 12345
Nombre: Dario Gomez

Hora

3:00 pm

Tratamiento

Extracciones (extracción de un diente): \$20

Agendar Cita

Regresar

Detalles de la Cita

Detalles de la Cita:

Fecha: Fri Jul 21 00:00:00 ECT 2023

Hora: 3:00 pm

Servicio: Extracciones (extracción de un diente): \$20

OK

```

public class PatientController {

    public static void add(Patient patient){

        String connectionString = "mongodb+srv://RBenavides:RBenavides@cluster0.js2ve9m.mongodb.net/";

        ServerApi serverApi = ServerApi.builder()
            .version(version: ServerApiVersion.V1)
            .build();

    public static ArrayList<Patient> show(){

        String connectionString = "mongodb+srv://RBenavides:RBenavides@cluster0.js2ve9m.mongodb.net/";

        ServerApi serverApi = ServerApi.builder()

    public static void update(int patientId, Patient newPatient) {

        String connectionString = "mongodb+srv://RBenavides:RBenavides@cluster0.js2ve9m.mongodb.net/";

        ServerApi serverApi = ServerApi.builder()
            .version(version: ServerApiVersion.V1)
            .build();

    public static void delete(int patientId){

        String connectionString = "mongodb+srv://RBenavides:RBenavides@cluster0.js2ve9m.mongodb.net/";

        ServerApi serverApi = ServerApi.builder()

private void btnAddPatientActionPerformed(java.awt.event.ActionEvent evt) {
    readTreatmentStartDate();
    readTreatmentEndDate();
    patient.setClinicalHistory(clinicalHistory);
    patient.setAppointments(appointments);
    patient.setAcquiredServices(acquiredServices);
    int option = JOptionPane.showConfirmDialog(parentComponent: this, "Está segura de que desea añac
    if(option == 0){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "saved, saving..");
        PatientController.add(patient);
        clearFields();
    }
    if(option == 1){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "not saved");
        clearFields();
    }
    if(option == 2){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "canceled");
        clearFields();
    }
}
}

```

```

public class FrmShowPatients extends javax.swing.JFrame {

    DefaultTableModel tablePatients = new DefaultTableModel();
    ArrayList<Patient> patients = new ArrayList<Patient>();

    private void fillPatients(){
        patients = PatientController.show();
    }
}

```

```

public FrmShowPatients() {
    initComponents();
    Toolkit toolkit = getToolkit();
    Dimension size = toolkit.getScreenSize();
    setLocation(size.width / 2 - getWidth() / 2, size.height / 2 - getHeight() / 2);

    fillPatients();
    setModel();
}

private void btnUpdatePatientActionPerformed(java.awt.event.ActionEvent evt) {
    readTreatmentStartDate();
    readTreatmentEndDate();
    newPatient.setClinicalHistory(clinicalHistory);
    newPatient.setAcquiredServices(acquiredServices);
    newPatient.setAppointments(appointments);
    int option = JOptionPane.showConfirmDialog(parentComponent: this, "Está segura de que desea actualizar el paciente?");
    if(option == 0){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "saved, saving..");
        PatientController.update(patientId, newPatient);
    }
    if(option == 1){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "not saved");
    }
    if(option == 2){
        JOptionPane.showMessageDialog(parentComponent: rootPane, message: "canceled");
    }
}

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    if(tblPatients.getSelectedRow() == -1){
        lblEditWarning.setText(text: "Selecciona un paciente.");
        lblEditWarning.setForeground(fg: Color.RED);
    }else{
        lblEditWarning.setVisible(aFlag: false);
        patient = patients.get(index: tblPatients.getSelectedRow());
        int option = JOptionPane.showConfirmDialog(parentComponent: this, "Está segura de que desea eliminar al paciente?");
        if(option == 0){
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Paciente Eliminado.");
            PatientController.delete(patientId: patient.getClinicalHistory().getId());
        }
        if(option == 1){
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Acción Negada.");
        }
        if(option == 2){
            JOptionPane.showMessageDialog(parentComponent: rootPane, message: "Acción Cancelada.");
        }
    }
}

```

Evidence

Walkthrough

Quito, 18 June, 7:30 am - 7:35 : Git-Hub

Order of the commits:

- Joshua Chiliquinga
- Rubén Benaúdes
- Pablo Carrera
- Adonny Calero

Quito, June 19, 7:35 - 7:40 : Google Docs

The Google Docs isn't updating since June 11, Ruben has done more changes

Quito, June 19, 7:40 - 7:45 : Mongo DB vs Class Diagram

Mongo DB has three types of collection that is equals to class Diagram, the types are: Patients, Password, Dentist

Quito, June 19, 7:45 - 7:50 : Class Diagram vs Software Requirements.

Quito, June 19, 7:50 - 8:00 : MVC

The program has the MVC packages and the JFrame, Java class and controllers inside, one error in this topic is that the controllers don't have called functions, the functions are inside the code.

Quito, June 19, 8:00 - 8:05 : Polymorphism

The only polymorphism application inside the code is the @Override to string

Quito, June 19, 8:05 - 8:10 : Test Cases / Unit Test

The only Unit Test that the project has is the encrypt password

Quito, June 19, 8:10 - 8:20 : GUI

The project has a login window, Menu window, Forms, tables and printout but the buttons has code inside

Quto, June 19, 8:20 - 8:25: CRUD

The project control all the CRUD operations and are called in the buttons.