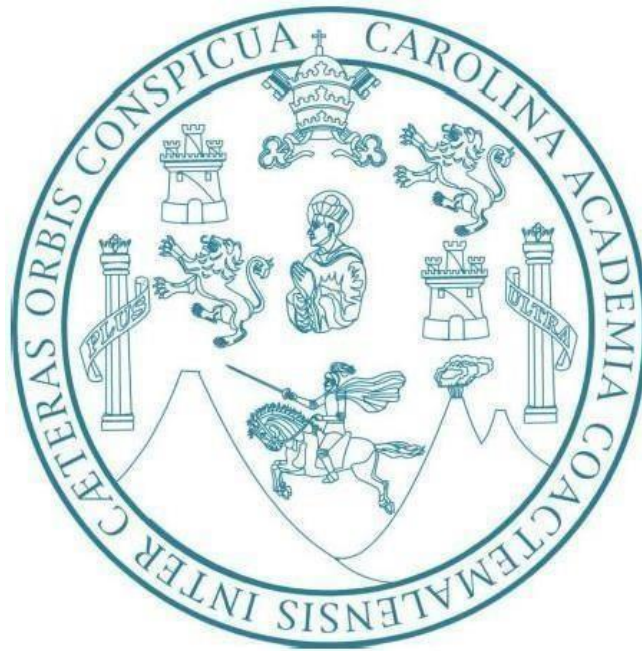


Facultad de Ingeniería
Introducción a la Programación y
Computación 1Sección: E
Cat. Ing. Neftalí De Jesús Calderón Méndez
Tutor académico: Rodrigo Antonio Porón De León



Manual Técnico

Proyecto#1

Universidad de San Carlos de Guatemala

INDICE

Contenido

INDICE	2
Lenguaje utilizado	3
Requisitos mínimos	3
Lógica utilizada	3
Contenido de las diferentes clases	4

Universidad de San Carlos de Guatemala

Lenguaje utilizado

El juego en consola esta desarrollado en Java, para su ejecución nada mas necesitamos el uso del JDK del kit 21.0, siendo esta la versión mas nueva de nuestro compilador de Java. Es de tomar en cuenta que el proyecto fue realizado en Gradle de java para un mejor manejo de paquetes y tener un manejo adecuado de dependencias.

Requisitos mínimos

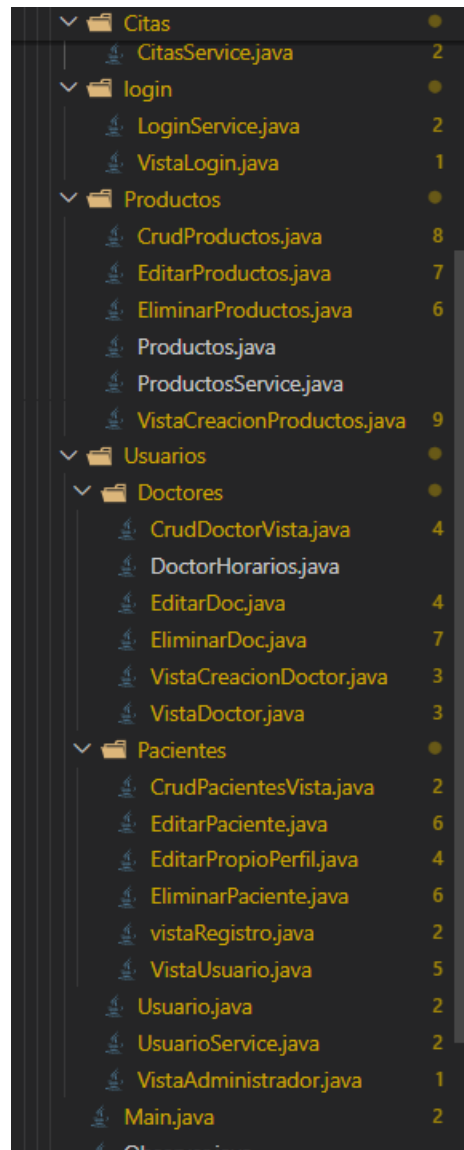
Necesitamos como mínimo en nuestro hardware para el correcto funcionamiento de nuestro programa:

- Windows 7 o posterior, Linux o IOS.
- JDK 21.0
- IDE

Universidad de San Carlos de Guatemala

Lógica utilizada

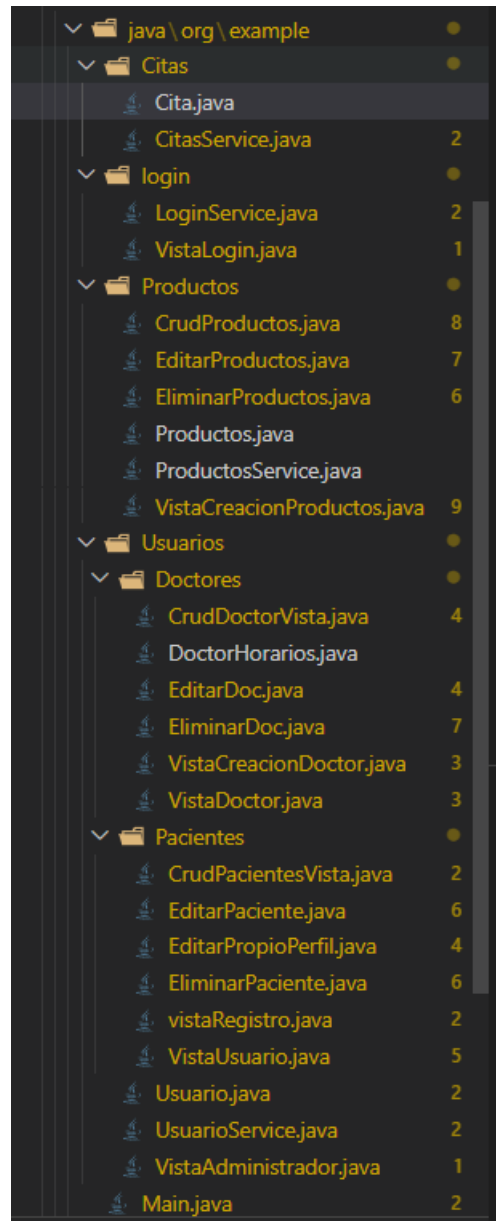
El programa este hecho con una lógica de las siguientes clases producida en gradle :



▼	Citas	●
	CitasService.java	2
▼	login	●
	LoginService.java	2
	VistaLogin.java	1
▼	Productos	●
	CrudProductos.java	8
	EditarProductos.java	7
	EliminarProductos.java	6
	Productos.java	
	ProductosService.java	
	VistaCreacionProductos.java	9
▼	Usuarios	●
▼	Doctores	●
	CrudDoctorVista.java	4
	DoctorHorarios.java	
	EditarDoc.java	4
	EliminarDoc.java	7
	VistaCreacionDoctor.java	3
	VistaDoctor.java	3
▼	Pacientes	●
	CrudPacientesVista.java	2
	EditarPaciente.java	6
	EditarPropioPerfil.java	4
	EliminarPaciente.java	6
	vistaRegistro.java	2
	VistaUsuario.java	5
	Usuario.java	2
	UsuarioService.java	2
	VistaAdministrador.java	1
	Main.java	2
	Observer.java	

Universidad de San Carlos de Guatemala

Lógica de las clases:



En el proyecto tenemos la división según en carpetas, donde se divide en citas, Login, productos, usuarios que tiene otras dos subdivisiones mas que son la de doctores y pacientes. Donde tenemos la denominación de Service para la lógica donde en esto sirve para la lógica del sistema.

Universidad de San Carlos de Guatemala

```
import java.util.ArrayList;
import org.example.Observer;
import java.util.Date;
import java.util.List;
import org.example.Usuarios.Doctores.DoctorHorarios;

import java.util.Random;
import java.util.stream.Collectors;
import java.io.*;

public class UsuarioService {

    private List<Usuario> usuarios;
    private static Random random = new Random();
    private List<Observer> observers = new ArrayList<>();
    private static Usuario doctorLogueado;

    public void addObserver(Observer observer) {
        observers.add(observer);
    }

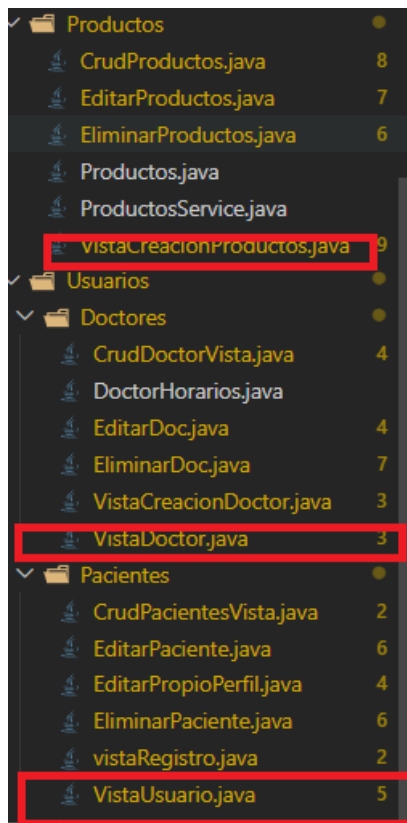
    public Usuario buscarPorId(String id) {
        List<Usuario> usuarios = getUsuarios();
        for (Usuario usuario : usuarios) {
            if (usuario != null && String.valueOf(usuario.getId()).equals(id)) {
                return usuario;
            }
        }
        return null;
    }

    public static Usuario setUsuarioLogueado(Usuario usuario) {
        doctorLogueado = usuario;
        return usuario;
    }
}
```

En el servicio definimos los métodos que vamos a usar para hacer las diferentes operaciones en nuestra aplicación, como puede ser añadir, buscar por x dato, eliminar y editar.

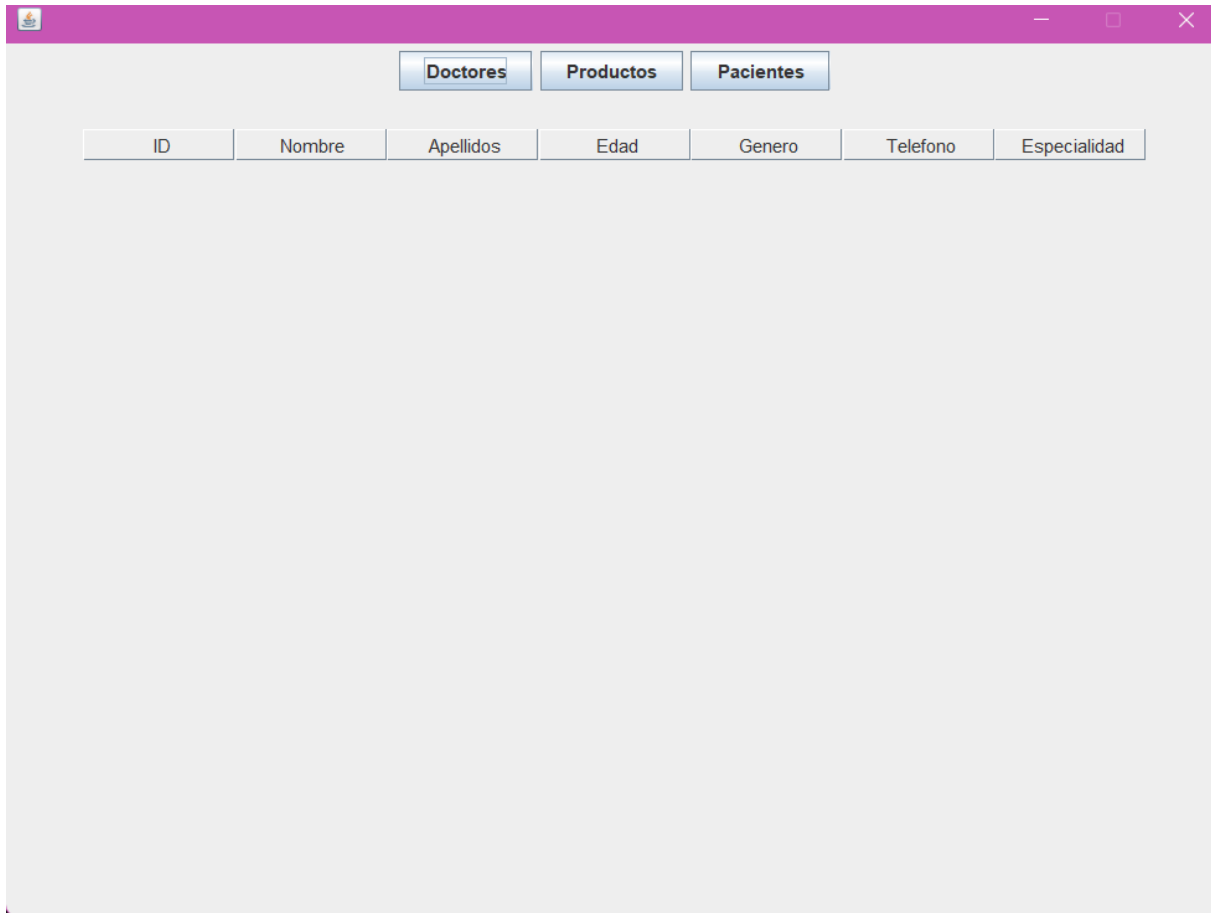
Universidad de San Carlos de Guatemala

La aplicación se divide en tres roles, que tenemos el módulo de administrador, el modulo de doctores y el módulo de pacientes, cada uno tiene su vista que podemos encontrarla en sus respectivas carpetas como:

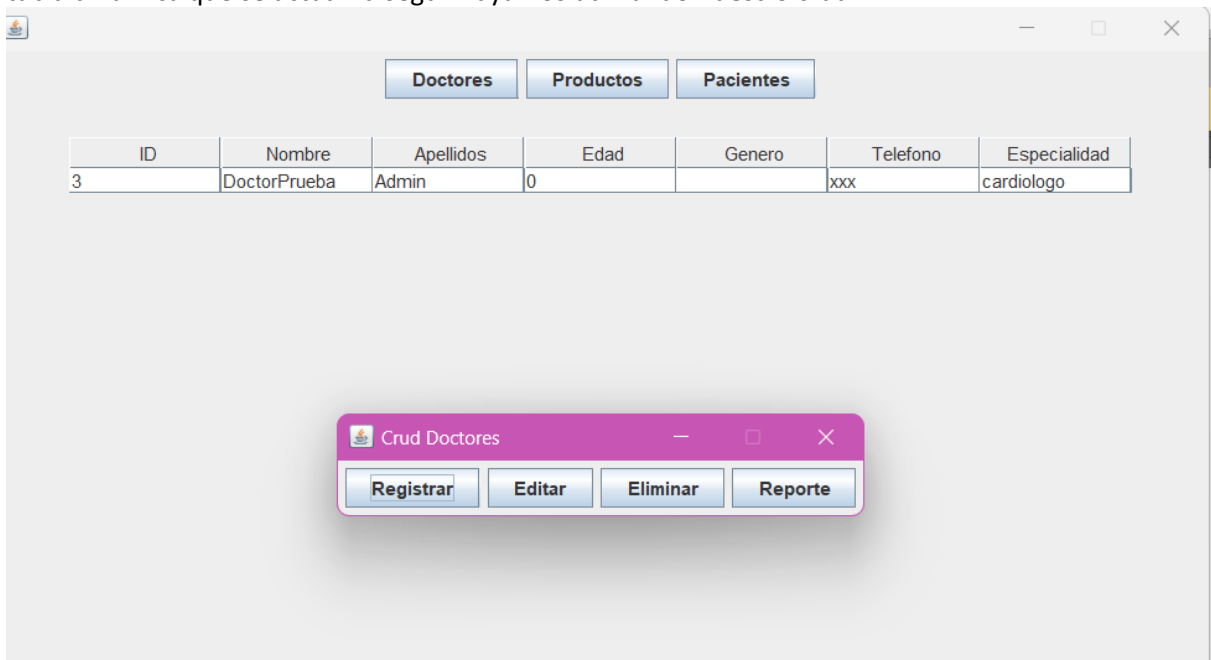


Aca tenemos los botones principales y la funcionalidad de cada uno de los elementos que podemos ver en nuestra interfaz grafica, para un mayor control se recomienda la separación adecuada de lo elementos del mismo.

Universidad de San Carlos de Guatemala

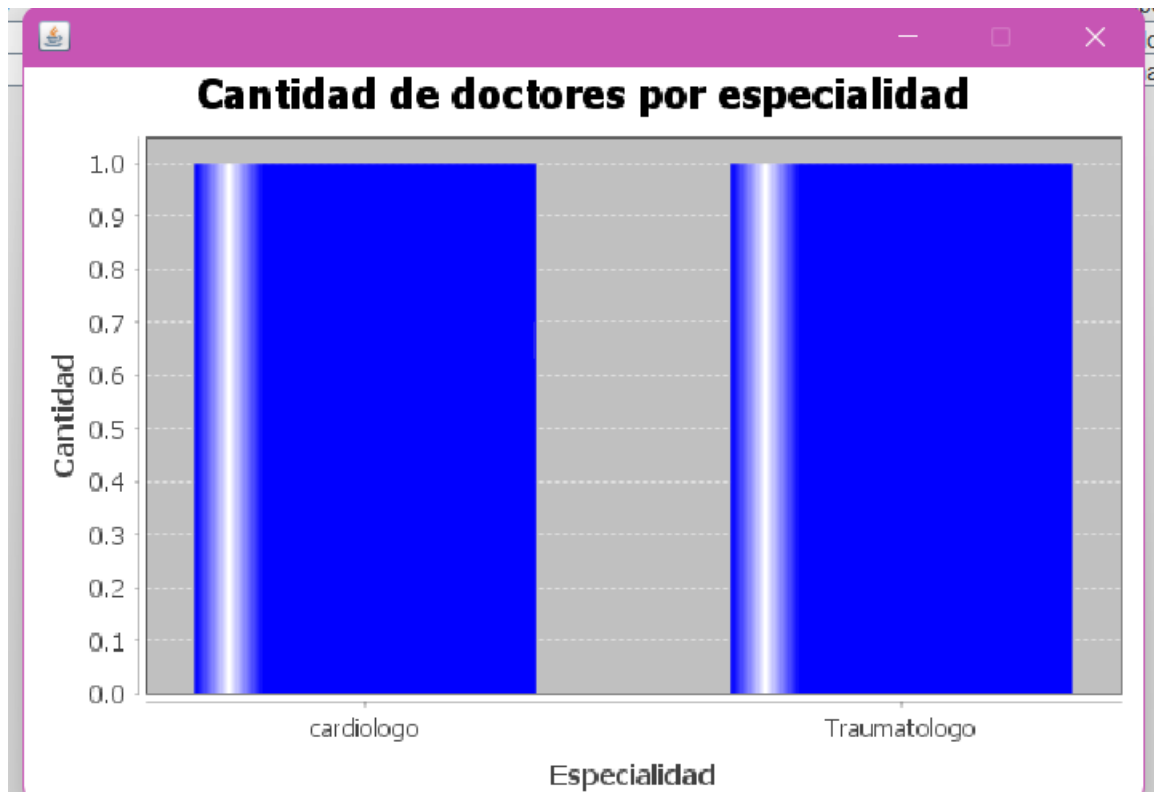


En el modulo de administración podemos encontrar las funciones que consisten en un CRUD de las diferentes listas creadas, en este caso serian las de Usuarios y las de Productos, donde tenemos una tabla dinámica que se actualiza según vayamos utilizando nuestro crud

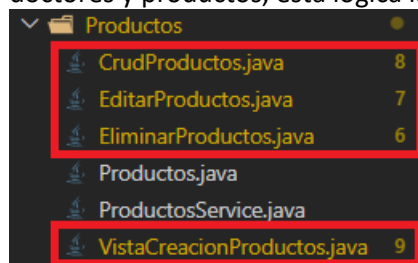


Universidad de San Carlos de Guatemala

Acompañado de un botón que se llama reporte que podemos crear una grafica de barras con el top de doctores, productos y usuarios, acá podemos encontrar la grafica que fue creada con FreeChart.



Estas mismas funcionalidades la tenemos para los diferentes objetos, como puede ser pacientes, doctores y productos, esta lógica la podemos encontrar en sus respectivas carpetas.



En todas las otras carpetas tenemos la misma lógica y nombres de las clases. Así mismo donde podemos gestionar la vista de usuarios y doctores para su funcionalidad, donde podemos agregar citas, asignar los diferentes horarios. Fue utilizado la implementación del observer, aunque en las versiones más nuevas de java no es tan recomendado su uso podemos decir que para uso práctico de esta aplicación fue de gran ayuda para mantener actualizadas las tablas.