

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESTRUCTURA DE DATOS

CATEDRÁTICO: ING. HERMAN IGOR VELIZ LINARES

TUTOR ACADÉMICO: ESTEBAN HUMBERTO VALDEZ ENNATI



MANUAL TÉCNICO

Ottoniel Fabricio Vásquez Pineda

CARNÉ: 202307820

SECCIÓN: B

GUATEMALA, 9 DE MARZO DEL 2,024

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	1
OBJETIVOS	1
1. GENERAL	1
2. ESPECÍFICOS	1
ALCANCES DEL SISTEMA	1
ESPECIFICACIÓN TÉCNICA	1
• REQUISITOS DE HARDWARE	1
• REQUISITOS DE SOFTWARE	1
DESCRIPCIÓN DE LA SOLUCIÓN	2
LÓGICA DEL PROGRAMA	2
❖ NOMBRE DE LA CLASE	
Captura de las librerías usadas	2
➤ Librerías	2
➤ Variables Globales de la clase _(El nombre de su clase actual)	3
➤ Función Main	3
➤ Métodos y Funciones utilizadas	3

INTRODUCCIÓN

En este manual se quiere a dar a entender su funcionalidad y también que es lo que hay detrás de la aplicación, ya que muchas veces no se ve lo que hay detrás de una aplicación, también para enseñar como se hizo este programa.

OBJETIVOS

1. GENERAL

1.1. Enseñar a todos lo que hay detrás de una aplicación

2. ESPECÍFICOS

2.1. Objetivo 1: Lo que quiero enseñar en este código, es enseñarles algunas partes que se usaron en el código con la cual puede que sean esenciales.

2.2. Objetivo 2: Mostrarle a cada persona que fue lo que más se utilizo en el código y también enseñarles algo diferente y la complejidad que lleva hacer todo esto.

ALCANCES DEL SISTEMA

El objetivo de este manual es llevar a cabo enseñarles conocimiento de como hizo la aplicación en detrás de la aplicación y no solo eso sino que también mostrarles lo que puede llevar solo una aplicación en la cual a simple vista se mire que es fácil pero no lo es.

ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE**

- No hay requerimientos específicos de hardware más allá de los necesarios para ejecutar una aplicación Java estándar.

- **REQUISITOS DE SOFTWARE**

- JDK (Java Development Kit) instalado para compilar y ejecutar la aplicación.
- Un entorno de desarrollo integrado (IDE) NetBeans, o simplemente el compilador de línea de comandos de Java (javac) y el intérprete (java).
- Sistema operativo compatible con Java (Windows, Linux, macOS).

DESCRIPCIÓN DE LA SOLUCIÓN

- Lo primero que se tubo que ver, es saber que librerías se iban a utilizar, que funcionalidades iban a llevar al principio del programa, en el transcurso y al final del programa, esto se tenía que ver un poco con tan solo verlo y lo demás cuando se comenzó a programar para que nos saliera el programa de manera de que funcione y también para que nos dé una interfaz parecida a la que se ve y también las funcionalidades.

LÓGICA DEL PROGRAMA

❖ VentanaAdmin

```
package proyectol;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
import javax.swing.*;  
import org.jfree.chart.ChartFactory;  
import org.jfree.chart.ChartPanel;  
import org.jfree.chart.JFreeChart;  
import org.jfree.chart.plot.PlotOrientation;  
import org.jfree.data.category.DefaultCategoryDataset;
```

➤ Librerías

La librería AWT: la utilicé más que todo para los botones, diseños, meter imágenes y entre otras cosas.

La librería Util ArraList: la utilice más que todo para hacer arreglos y ponerlos en una lista.

La librería javax swing: me sirvió para crear interfaces graficas de usuario.

La librería Jfree: más que todo está librería me ayudo a poder hacer graficas.

➤ Variables Globales de la clase _(El nombre de su clase actual)

```
public class Proyecto1 {  
    public static int Codigos1 =500;  
    public static int codigo=200;  
    public static int Numero = 1;  
    public static int códigos=1;  
    public static int CódigoDoc;  
    public static int CódigoPac;  
  
    public static ArrayList<Doctores> listaDoctores = new ArrayList<>();  
    public static ArrayList<Doctores> listaDocHora = new ArrayList<>();  
    public static ArrayList<Pacientes> listaPacientes = new ArrayList<>();  
    public static ArrayList<Producto> listaProducto = new ArrayList<>();  
    public static ArrayList<CitasContudocotor> listaHorarioCitas2 = new ArrayList<>();  
  
    private static Map<Integer, List<String>> horariosPorDoctor = new HashMap<>();  
}
```

Más que todo estas variables me ayudaron para hacer arreglos o listas y también para trabajar con códigos y números el programa


➤ Función Main

En este caso use la función main para ir a una ventana de inicio de sesión.


```
public static void main(String[] args) {  
    Login ventana = new Login();  
}
```

➤ Métodos y Funciones utilizadas


A continuación, se dará una explicación general de lo que hace cada función:


-  `public static void Agregar_Doctores(i`


Esta función sirve para conseguir los datos de los doctores.


-  `public static Object[][] convertirDatosDoctores_tabla() {`


Esta función me sirve para obtener datos y ponerlos en un arreglo o en una tabla.


- 59  `public static ArrayList<Doctores> bubblesortEdadDoctores() {`


Esta función la utilicé para ordenar doctores de mayor a menor, con la cual se le conoce más como método burbuja.
- 96  `public static void Agregar_Pacientes(:`


Esta función más que todo me ayudo a obtener datos de pacientes.
- 104  `public static Object[][] convertirDatosPacientes_tabla()`


Esta función me sirvió a poder a poner datos en una tabla.
- 124  `public static void Agregar_Productos(.`

Use esta función para obtener datos y guardarlos.
- 131  `public static Object[][] convertirDatosProducto_tabla() {`

Use esta función para poner datos que había guardado y conseguido anteriormente.
- 152  `public static ArrayList<Producto> bubblesortCantidadProducto() {`

Use esta función para ordenar valores de Producto de mayor a menor.
- 189  `public static void Agregar_horarios1(String horario, i`

Utilice esta función para poder guardar horarios y tenerlos almacenados para utilizarlos después.
- 196  `public static Object[][] convertirHorarioDoc_tab`

Utilice esta función para poner los horarios obtenidos anteriormente, almacenarlos en una tabla.
- 211  `public static Object[][] convertirDatosCitas_tal`

Utilicé esta función para poner datos que obtuve estados de citas.

