UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESTRUCTURA DE DATOS

CATEDRÁTICO: ING. WILLIAM ESTUARDO ESCOBAR ARGUETA

TUTOR ACADÉMICO: JOSUÉ RODOLFO MORALES CASTILLO



**MANUAL TÉCNICO DE UHOSPITAL**

ENNER ESAÍ MENDIZABAL CASTRO

CARNÉ: 202302220

SECCIÓN: B

GUATEMALA, 6 DE MARZO DEL 2,024

# ÍNDICE

[**ÍNDICE 1**](#_k1cgegpfdrrl)

[**INTRODUCCIÓN 1**](#_574nc6ehlpl)

[**OBJETIVOS 1**](#_3bsxbi49eln7)

[1. GENERAL 1](#_38w1k6cukb5i)

[2. ESPECÍFICOS 1](#_poh928tpbmga)

[**ALCANCES DEL SISTEMA 1**](#_twvsj8hvwtb9)

[**ESPECIFICACIÓN TÉCNICA 1**](#_hxtoxxv43krt)

[● REQUISITOS DE HARDWARE 1](#_mfriwpcdfh1l)

[● REQUISITOS DE SOFTWARE 1](#_8h6kxoigku56)

[**DESCRIPCIÓN DE LA SOLUCIÓN 2**](#_14axd4p9n3ry)

[**LÓGICA DEL PROGRAMA 2**](#_ng9quldintk8)

[❖ NOMBRE DE LA CLASE  
Captura de las librerías usadas 2](#_xush55kxig5)

[➢ Librerías 2](#_h84kbu3eveip)

[➢ Variables Globales de la clase \_(El nombre de su clase actual) 3](#_jmrvbw5ti5o0)

[➢ Función Main 3](#_lg5om9s2qe42)

[➢ Métodos y Funciones utilizadas 3](#_xbejjcbelcz7)

# 

# INTRODUCCIÓN

Este manual fue creado con la finalidad de explicar los distinto métodos, clases y librerías que se usaron para la creación de UHOSPITAL utilizando un lenguaje y descripciones más técnicas para que aquella persona interesada en los constitución interna del programa puedo entenderlo.

# OBJETIVOS

## GENERAL

* 1. Proporcionar información técnica del programa

## ESPECÍFICOS

* 1. Objetivo 1: Dar un mayor entendimiento de los métodos que se usaron en el registro y sobre la lógica utilizada.
  2. Objetivo 2: Indicar las librerías que se utilizaron para la generación del entorno gráfica sobre el cual se ejecuta el programa.

# ALCANCES DEL SISTEMA

Este manual pretende dar a un usuario con conocimiento técnico sobre JAVA la compresión del funcionamiento y estructura del programa UHOSPITAL para así si está interesado en replicar algunos de los comportamiento del programa mismo, pueda realizarlo por medio de las explicaciones técnicas de las que se tratará este manual.

# ESPECIFICACIÓN TÉCNICA

## REQUISITOS DE HARDWARE

* + Procesador: 2 GHz o superior
  + Disco duro: Espacio se al menos 1.5 GB
  + RAM: 2GB o superior

## REQUISITOS DE SOFTWARE

* + Sistema operativo: Windows, macOS o Linux
  + Java development kit (JDK)
  + Librería de Jtattoo
  + Librería de JfreeChard
  + Entorno de desarrollo integrado (IDE) compatible con Java (se recomienda NetBeans)

# DESCRIPCIÓN DE LA SOLUCIÓN

* **Creación de la interfaz gráfica:** Para la generación de las distintas pestañas, se usaron las librerías de AWT y SWING de java, con las cuales se generaron tanto pestañas como los botones, etiquetas, carillas, tablas y todo lo visualmente requeridos en este programa.
* **Creación de Usuario y doctores:** Para la creación de las cuentas de los usuarios y de los doctores, se utilizaron dos *arrayList* que contendrían las *clases* doctor y paciente, de esta forma tendría contenido de forma ordenada cada uno de los doctores con sus distintos *atributos* que se solicitarían a la hora de mostrarse la pestaña para el registro de cada uno. Por lo que, cuando se presiona el botón de registrar, se agrega al *arrayList* la nueva clase con la información de los campos ingresada en la nueva *instancia*.
* **Método de autenticación:** Para la autenticación del usuario en la pestaña de inicio, se recurrió a un método que se encargaría de buscar en cada una de las *ArrayList* que contiene los distintos pacientes y doctores, un doctor o paciente que coincida con el código y contraseña que se ingreso en el campo a rellenar, es decir, cuando se pulsa el botón para iniciar sesión, primero verifica si es le código y contraseña el administrador, en dado caso no sea así, verifica en el *ArrayList* de pacientes si hay alguno que posea la misma contraseña y código, y si en dado caso tampoco lo encuentre, verifica en la *ArrayList* de Doctores. Si no encuentra nada en ninguna, mostrará un mensaje de dialogo que indique que el usuario está mal escrito y si encuentra algún usuario que tenga la contraseña y código que se ingresó, se guardará la posición del ArrayList en donde se encontró al usuario y aparecerá la pestaña de pertinente al usuario en cuestión.
* **Generación del código único:** Para la generación del código único tanto de los doctores, de los pacientes y de los productos. Se utilizó una variable con un valor de 1000 y 100 en el caso de los productos, los cuales se asignarán a cada uno de los productos o usuarios y se sumará uno, para que de este modo, no puedan tener varios el mismo código. Los pacientes y Doctores comparten esta cuenta para evitar que se de la situación en la que tanto el doctor como el paciente tengan la misma contraseña y código.
* **Generación de Usuarios y productos:** Para la generación y creación de esto, se usa un método el cual se dedica a añadir un nuevo elemento de al arrayList de Usuario o producto y le ingresa los valores que se obtenidos de los campo de la ventana hecha para su creación.
* **Actualización de los datos de los Usuarios:** Para actualizar los datos de los usuarios, se puede realizar de dos formas, ya sea mediante el módulo del administrador o por medio de los módulos de cada usuario respectivamente. No importa como se haga, el funcionamiento solo se diferencia en que el administrador requerirá ingresar el código del usuario para poder modificarlo, lo cual no será necesario dado a que al ingresar, como se mencionó anteriormente, se guarda el índice del ArrayList de donde se obtuvo el el código y contraseña que concordaban, por lo que el programa ya conoce eso y no lo requiere. Una vez presionado el botón para actualizar datos, se modificará la instancia del objeto en el ArrayList en la posición del usuario indicada.
* **Eliminación de los usuarios y productos:** La única forma que se tiene para eliminar los usuarios y los productos, es por medio del módulo del administrador, en el cual aparecerá la opción para esta acción y solicitará el código. Una vez con el código, invocará el método que se encarga de hallar el índice por medio de su código, guardará este y lo borrará del arrayList.
* **Presentar los datos en tablas**: Para presentar los datos de los usuarios, de los productos y de los doctores en tablas, se usó un método que regresaría una matriz de tipo Object que contendría todos los valores que requiere la tabla mediante un ciclo que crearía esa matriz asignando cada valor en su lugar correspondiente, y luego de que se genere, se pasará a una tabla creado con SWING la cual lo mostrará. Para que este valor se actualicen lo valores de la tabla, se recurrió a que cada vez que se añadiera un usuario o producto, se cierre y abra nuevamente la venta, para que, de este modo, se realizan todo el procedimiento nuevamente y se genere la nueva tabla actualizada.
* **Generación de las citas:** Para generar las citas, se creó dentro de la clase personas, 3 arraysList en los cuales se guardarían las fechas, horas y estados de cada una de las citas disponibles, por lo que para llenarlas, se ingresaría en la posición del arrayList que ya tenemos y los colocamos en la arrayList de la instancia en esa posición. Adicionalmente, dentro de la clase doctores, se colocó un atributo ArrayList de tipo Integer que se encargaria de guardar la posición del arrayList en donde se encotraria el paciente que solicito la cita con dicho doctor.
* **Graficas**: Para la generación de las gráficas, se usó la librería de JfreeChard, con la cual se creó la gráfica ingresando en valores numéricos la frecuencia con la que había cada especialización y nombres de productos repetido en las ArrayList de los mismos...

# 

# LÓGICA DEL PROGRAMA

## NOMBRE DE LA CLASE *Captura de las librerías usadas*

### Librerías

Breve descripción del funcionamiento de cada librería usada y breve descripción de para qué la usó

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

*Captura de sus variables globales*

Breve descripción generalizada de el uso de sus variables globales (No es necesario especificar para qué se usó cada una)

### Función Main

Breve descripción de para qué usaron la función main

*Captura del código de su función main*

### Métodos y Funciones utilizadas

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

* *Captura del nombre de su función que muestre el número de línea*  
  Breve descripción del uso de la función o para qué sirve
*   
  Breve descripción del uso de la función o para qué sirve