



18 DE OCTUBRE DE 2024


PROYECTO ALGORITMOS

MANUAL TÉCNICO

SOL MARTINEZ

CARNET : 7590-24-4829

Sección "B"

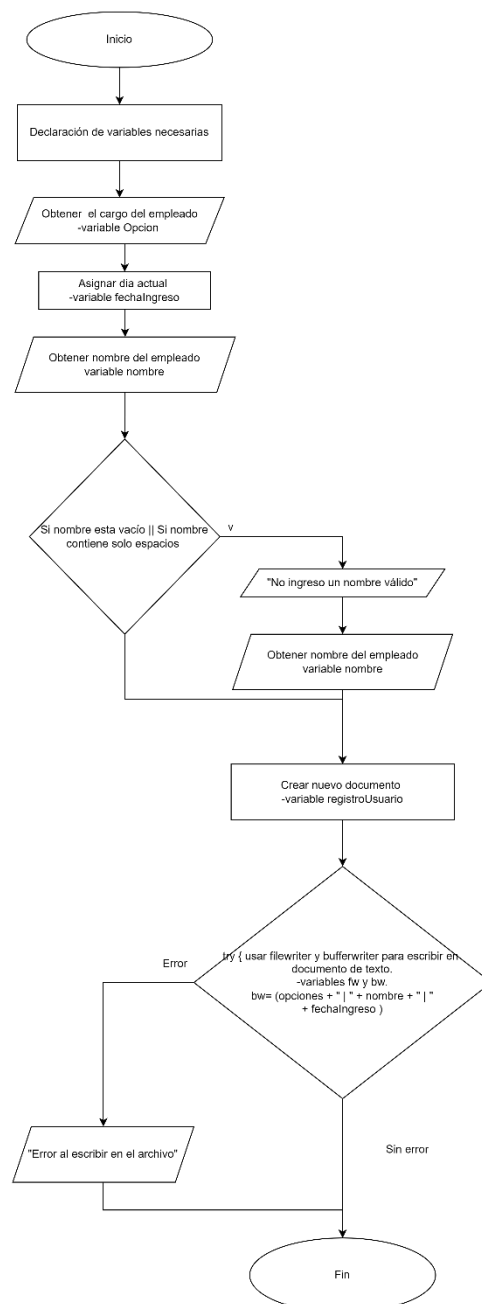


INTRODUCCIÓN:

Se buscó desarrollar un programa con la función de almacenar, modificar y llevar un control de los datos que se ingresaran, implementando medidas de seguridad de los datos al asignar a cada cargo del empleado una cierta disponibilidad de información.

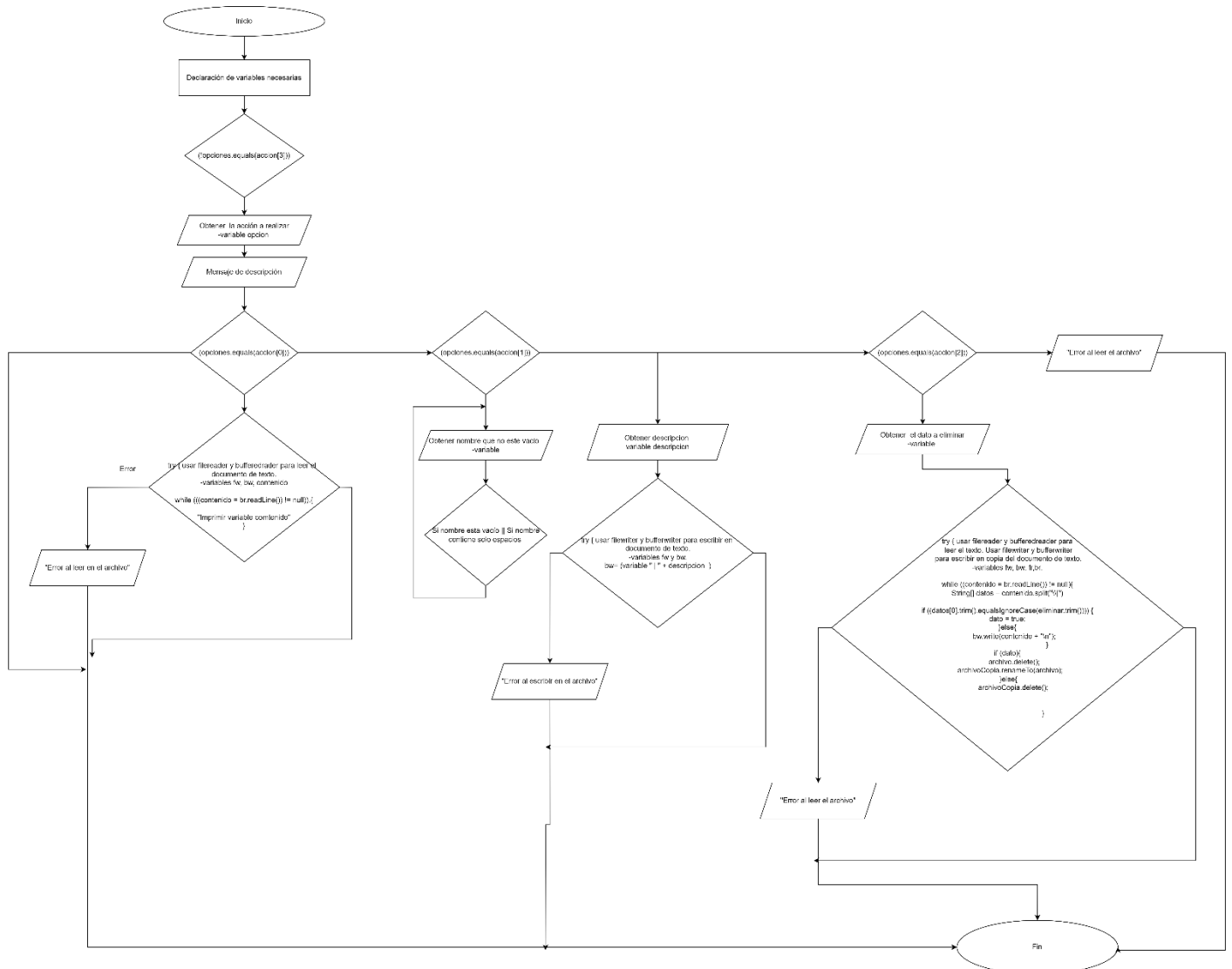
DIAGRAMA DE FLUJO

Inicio.



Categorías, Especificaciones, Características:

Estas tres clases comparten un diagrama porque tienen los mismos procesos y el mismo funcionamiento.



ALGORITMOS:

Clase (Inicio)

Importar clases necesarias

1. Declarar variables

String opciones;

LocalDate fechaIngreso;

```
String nombre;
```

```
String[] cargo = {"Administrador","Vendedor", "EncargadodeBodega"};
```

2. Obtener datos del usuario (nombre, cargo, fecha de ingreso)

```
Leer opciones;
```

```
fechaIngreso = LocalDate.now();
```

```
Escribir(fechaIngreso);
```

```
Leer nombre;
```

3. Condicionar la variable para escribir correctamente el nombre

```
if (nombre.isEmpty() || nombre.isBlank()){
```

```
    Escribir "No ingreso ningun dato, ingreselo de nuevo.");
```

```
    Leer nombre;
```

```
}
```

4. Guardar datos en archivo de texto

a) Crear un nuevo archivo

```
File registroUsuario = new File ("Historial_de_movimientos.txt");
```

b) Escribir en el archivo

```
try {
```

```
    FileWriter fw = new FileWriter(registroUsuario, true);
```

```
    BufferedWriter bw = new BufferedWriter(fw);
```

```
    bw.write(opciones + " | " + nombre + " | " + fechaIngreso + "\n");
```

```
    bw.close();
```

```
} catch (IOException ex) {
```

```
    Escribir ("Error al escribir en el archivo")
```

```
}
```

Clases (características, especificaciones, categorías)

Importar clases necesarias

1. Declarar variables

```
String opciones = "";
```

```
String (caracteristica, especificacion o categoría, dependiendo de la clase) = "";
```

```
String descripcion;
```

```
String eliminar = "";
String contenido;
Boolean dato = false;
String[] accion = { "Ver ", "Agregar", "Eliminar", "Regresar a Inicio" };
```

2. Crear archivos de texto

```
File archivo = new File("Caracteristicas.txt");
File archivoCopia = new File("Caracteristicas_copia.txt");
```

3. Acciones a tomar (ver Características, agregar, modificar, eliminar, regresar)

3.1) Condicionar para asignar un proceso a la variable

```
while (!opciones.equals(accion[3])) {
```

```
    leer opciones
```

```
if (opciones.equals(accion[0])) {
```

3.1.1) Mostrar Contenido

```
    try {
        contenido = "";
        FileReader fr = new FileReader(archivo);
        BufferedReader br = new BufferedReader(fr);

        Escribir ("Ver las características en la línea de comandos");
        while ((contenido = br.readLine()) != null){
            Escribir (contenido);
        }
        br.close();
    } catch (IOException e) {
        Escribir ("Error al leer el archivo");
    }
}
```

3.1.2) Agregar datos

```
    } else if (opciones.equals(accion[1])) {
        a) Evaluar nombre (en blanco o vacía )
        do {
            Leer caracteristica;
```

```
} while (variable.isBlank() || variable.isEmpty());
```

Leer descripción;

b) Agregar datos a archivo de texto

```
try {  
    FileWriter fw = new FileWriter(archivo, true);  
    BufferedWriter bw = new BufferedWriter(fw);  
    bw.write(variable + " | " + descripcion + "\n");  
    bw.close();  
  
} catch (IOException ex) {  
    Escribir ("Error al escribir en el archivo");  
}
```

```
} else if (opciones.equals(accion[2])) {
```

3.1.3) Eliminar datos ingresados

a) Pedir característica a eliminar

Leer eliminar;

```
if (eliminar != null && !eliminar.isBlank()){
```

```
    try {
```

b) Leer el archivo principal y colocar los datos de este en una copia

```
    FileReader fr = new FileReader(archivo);
```

```
    BufferedReader br = new BufferedReader(fr);
```

```
    FileWriter fw = new FileWriter(archivoCopia);
```

```
    BufferedWriter bw = new BufferedWriter(fw);
```

c) Condiciones para que se elimine la característica

```
        while ((contenido = br.readLine()) != null) {
```

```
            String[] datos = contenido.split("\\|");
```

```
            if ((datos[0].trim().equalsIgnoreCase(eliminar.trim())) {
```

```
dato = true;
```

```
    }else{
```

```
        bw.write(contenido + "\n");
```

```
    }
```

```
}
```

```
br.close();
```

```
bw.close();
```

```
if (dato){
```

```
    archivo.delete();
```

```
    archivoCopia.renameTo(archivo);
```

```
    Escribir ("La característica se eliminará");
```

```
    }else{
```

```
        archivoCopia.delete();
```

```
        Escribir "No se encontró la característica";
```

```
    }
```

```
    } catch (IOException ex) {
```

```
        Escribir ("Error al leer el archivo");
```

```
    }
```

```
}
```

```
} else {
```

```
    Escribir ( "Regresará al inicio");
```

```
}
```

```
}
```

```
}
```

TECNOLOGÍAS UTILIZADAS:

Lenguaje de programación Java para desarrollo de aplicaciones de escritorio y Swing para la creación de la interfaz gráfica para el usuario.

CONTACTO:

Sol Arely Martínez García. / Carnet: 7590-24-4829

smartinezg17@miumg.edu.gt

Tel: 42303854