



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

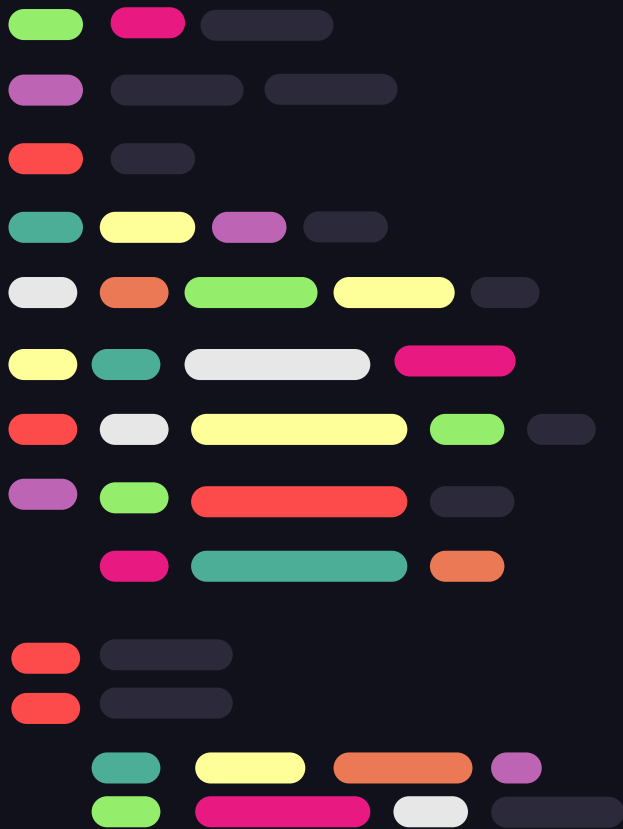
Viernes, 16/08/2024

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [B]

Josué Rodolfo Morales Castillo



{ ..



Clase 4- Agenda

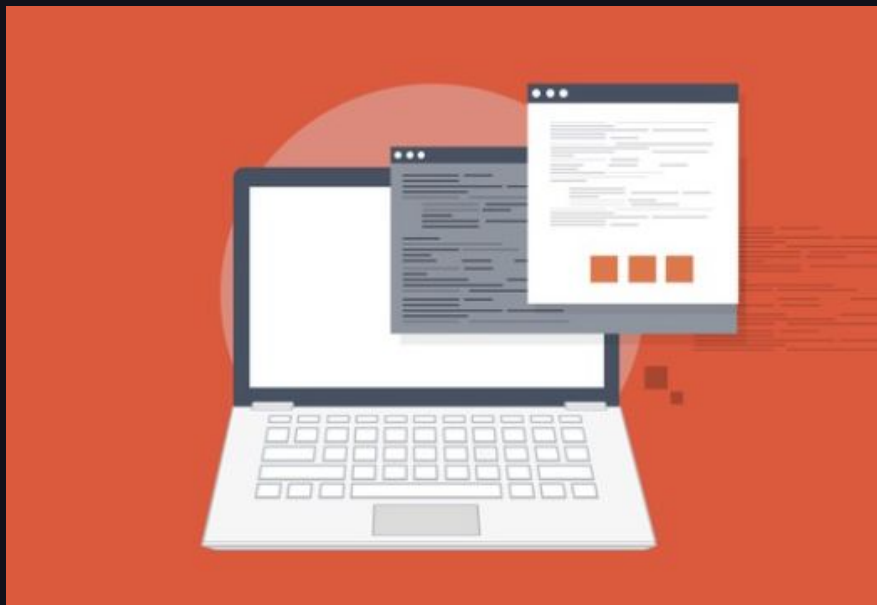
- Foro No. 4
- Recordatorio Commits semanales
- Preguntas proyecto 1
- Interfaces Gráficas en JAVA (AWT y SWING) y Manejo de Archivos

} ..

{



Interfaces Gráficas en Java



}

¿Qué son?

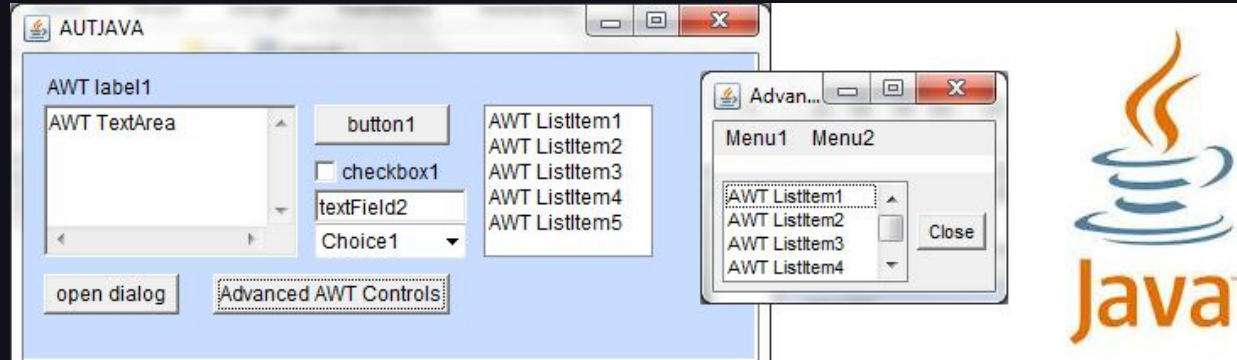
Una interfaz gráfica de usuario (GUI) en Java es un conjunto de componentes gráficos que permiten a los usuarios interactuar con programas mediante elementos visuales, como ventanas, botones, cajas de texto, menús y otros elementos.

Java proporciona varias bibliotecas para crear interfaces gráficas, siendo Swing y AWT dos de las más utilizadas.



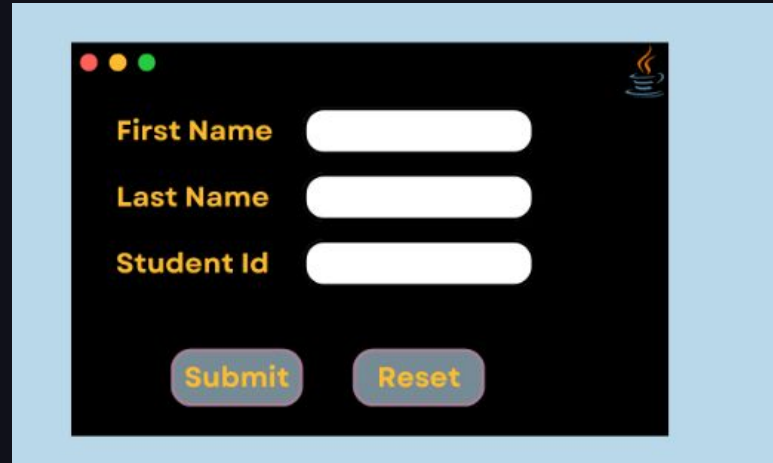
AWT (Abstract Window Toolkit)

Es una biblioteca gráfica para el desarrollo de interfaces de usuario en Java. Es parte del paquete `java.awt` y fue una de las primeras bibliotecas de este tipo en Java. AWT proporciona un conjunto de clases y métodos que permiten la creación de ventanas, cuadros de diálogo, botones, campos de texto y otros elementos de la interfaz gráfica.



Swing

Swing es una extensión de AWT para proporcionar una mayor flexibilidad y una interfaz gráfica más rica y consistente. Swing es más ligero, personalizable y ofrece componentes más avanzados que AWT. Sin embargo, AWT aún se utiliza en ciertos casos y puede ser relevante en entornos más específicos.





Componentes de la Interfaz Gráfica

Los componentes de la interfaz gráfica son elementos visuales que se utilizan para construir la interfaz de usuario de una aplicación. Algunos de los componentes más comunes en Java son:

JFrame La ventana principal de la aplicación.

Un contenedor que se puede usar para organizar y agrupar otros componentes

JPanel

JButton Un botón que el usuario puede hacer clic

Un componente que muestra texto o una imagen

JLabel

TextField Un campo de texto que permite al usuario escribir texto.

Le muestra al usuario varias opciones que puede seleccionar.

**JCheckBox y
JRadioButton**

Disparadores de Eventos

Los eventos en una interfaz gráfica son acciones o situaciones específicas que ocurren durante la interacción del usuario con la aplicación. Los disparadores de eventos son objetos que representan estas acciones.

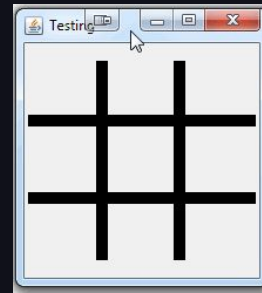
ActionEvent

Se dispara cuando ocurre una acción, como hacer clic en un botón.



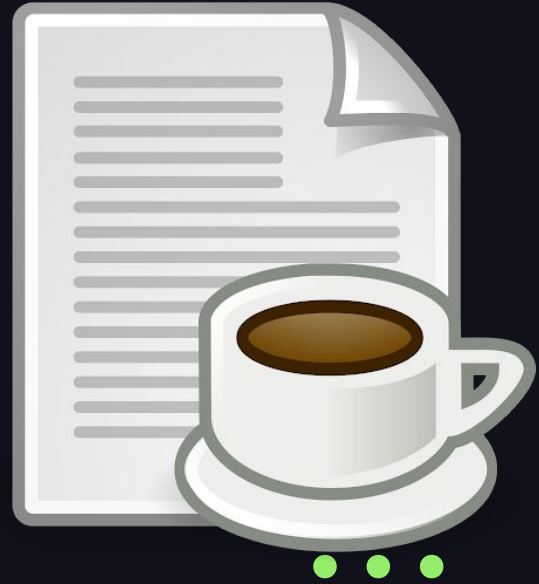
MouseListener

Se utiliza para manejar eventos relacionados con el ratón, como clics y movimientos.





Manejo de Archivos en Java



Archivos de texto plano

Los archivos de texto plano son una forma fundamental de almacenar y manipular datos en el desarrollo de software. En Java, trabajar con archivos de texto es una tarea común y esencial para una amplia gama de aplicaciones.





Creación de Archivos (FileWriter)

```
1 import java.io.FileWriter;
2 import java.io.IOException;
3
4 public class EscribirArchivo {
5     public static void main(String[] args) {
6         try {
7             FileWriter archivo = new FileWriter("archivo.txt");
8             archivo.write("¡Hola, mundo!");
9             archivo.close();
10            System.out.println("Texto escrito correctamente.");
11        } catch (IOException e) {
12            System.out.println("Error al escribir en el archivo.");
13            e.printStackTrace();
14        }
15    }
16 }
```

Lectura de Archivos(FileReader)

```
1  import java.io.FileReader;
2  import java.io.IOException;
3
4  public class LeerArchivo {
5      public static void main(String[] args) {
6          try {
7              FileReader archivo = new FileReader("archivo.txt");
8              int caracter;
9              while ((caracter = archivo.read()) != -1) {
10                 System.out.print((char) caracter);
11             }
12             archivo.close();
13         } catch (IOException e) {
14             System.out.println("Error al leer el archivo.");
15             e.printStackTrace();
16         }
17     }
18 }
```

Eliminación de Archivos(delete)

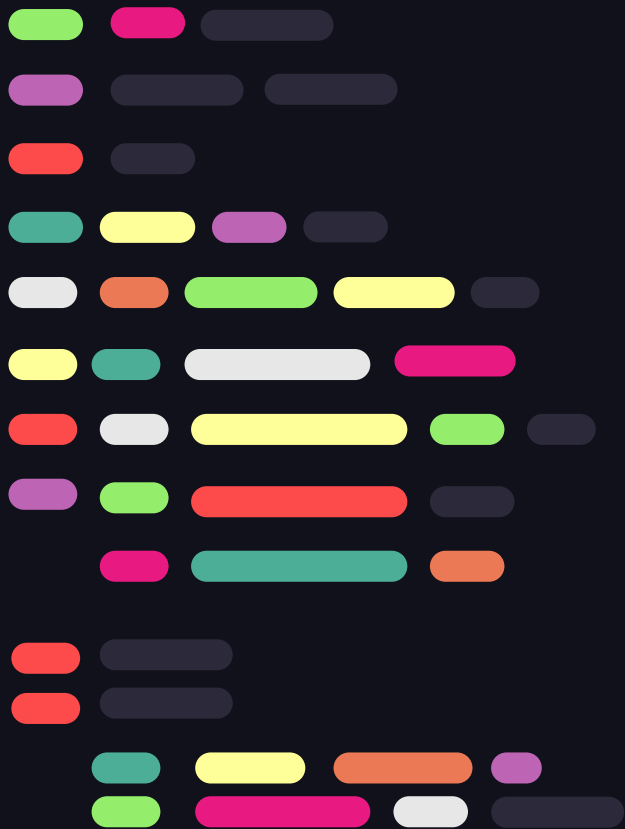
```
1  import java.io.File;
2
3  public class EliminarArchivo {
4      public static void main(String[] args) {
5          File archivo = new File("archivo.txt");
6          if (archivo.delete()) {
7              System.out.println("Archivo eliminado correctamente.");
8          } else {
9              System.out.println("Error al eliminar el archivo.");
10         }
11     }
12 }
13
```

Modificar archivo

```
1  String nombreArchivo = "archivo.txt";
2
3  try {
4      // Abrir el archivo en modo de lectura y escritura
5      FileWriter fileWriter = new FileWriter(nombreArchivo, true);
6      BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
7      PrintWriter printWriter = new PrintWriter(bufferedWriter);
8
9      // Modificar el contenido del archivo
10     printWriter.println("Nueva línea agregada al archivo.");
11
12     // Cerrar el archivo
13     printWriter.close();
14
15     System.out.println("El archivo se ha modificado correctamente.");
16 } catch (IOException e) {
17     System.out.println("Ocurrió un error al modificar el archivo: " + e.getMessage());
18 }
```

¿Dudas?





{ ..



Ejemplo

} ..