**Multi302 marzo 20 Interfaz Gráfica JFrame - Final proceso Evaluación**

1. Qué es GUI – FrontEnd
2. Características de JFrame y Controles.
3. Ejercicio GUI sin y con MVC.

**Desarrollo**

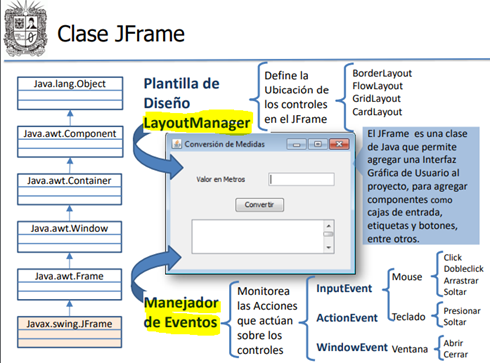
1. **Qué es GUI – FrontEnd**

GUI ( Graphical User Interface) - Interfaz Gráfica de Usuario, está compuesta por los controles y contenedores que permiten una presentación más amigable al usuario, el componente principal casi siempre es una ventana o formulario, en este caso se hará uso de un JFrame para diseñar la Interfaz. Comúnmente el diseño de interfaz en ambientes web está asociado al término FrontEnd, que corresponde al ambiente visual de una aplicación y se encuentra relacionado con HTML, CSS, JavaScript. Pero en el caso de las aplicaciones de escritorio sobre las cuales se está desarrollando la práctica de esta asignatura, este tipo de tecnologías no se utilizarán.

El FrontEnd que se trabajará está compuesto por los componentes gráficos de Java que se encuentran en el paquete swing y especialmente se trabajará con el JFrame.

1. **Características de JFrame y Controles.**

El Jframe contiene dos elementos asociados, el primero corresponde a la **plantilla de diseño o Layout** que corresponde al formato de distribución de los controles dentro del contenedor de la Ventana. El segundo corresponde al **manejador de eventos** que permite identificar las acciones que el usuario realiza (usando mouse o teclado) y a las cuales la aplicación da respuesta.

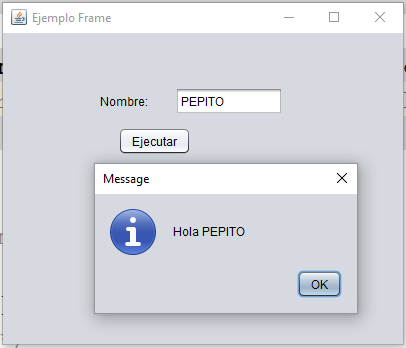


1. Ejercicio GUI sin y con MVC.

* **Ejemplo de GUI sin MVC**

Para usar el GUI en los proyectos se usará la clase JFrame, debe crearse dentro del paquete Vista.

Ejemplo programando desde la ventana



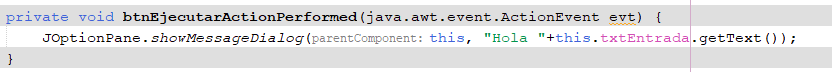
Código generado por Netbeans al agregar los controles

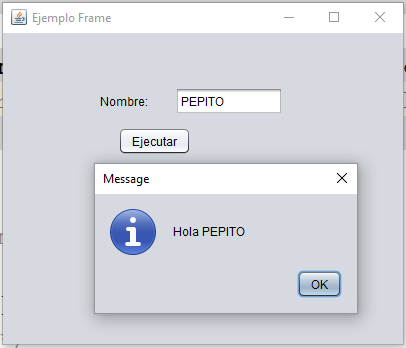


**Nota:**  
El JFrame puede codificarse dentro de su misma clase, tal como aparece en el método siguiente, donde se observa el **ActionPerformed** directamente sobre el botón **btnEjecutar** y el cúal recibe un objeto tipo evento de acción **ActionEvent** para que se realicen las instrucciones que se ejecutan cada vez que el botón sea activado.

Procedimiento actionPerformed de Botòn programado

Código del botón

Resultado de ejecución



* **Ejemplo de GUI con MVC**

Para usar el JFrame con MVC en los proyectos se sugiere realizar los siguientes pasos: Implementar el modelo , Diseñar la Vista e Implementar el controlador. Para este ejercicio solo se trabajará la vista y el controlador.

* **Diseñar la Vista**

**P**ara aplicar el MVC, inicialmente se debe preparar la vista, en ese caso debe agregarse una clase JFrame y realizar los siguientes procesos

1. Agregar los controles (Cajas de texto - JTextField, Etiquetas - JLabel, Botones -JButton y otros)
2. Asignar nombres a controles (Se pueden usar tres formas explicadas en clase)
3. Generar get y set de los controles
4. Eliminar el método main (Para convertir la clase JFrame en una clase de lógica del negocio)
5. **NO PROGRAMAR SOBRE EL JFRAME**

**Clase Ventana**



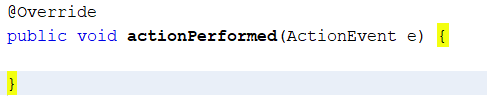
* **Implementar el controlador:** Siga los siguientes pasos

1. Agregar el **implements** al Controlador

Al usar el MVC, es necesario controlar el manejador de eventos desde el controlador, por lo tanto se hará uso de la interfaz de java **ActionListener**, la cual contiene el método **actionPerformed**(), en el cual se debe validar cual es el control que recibe el evento y que instrucciones deberán ejecutarse.

Insertando imagen...

Al realizar esta acción el sistema solicitará la generación de los métodos abstractos, en este caso **actionPerformed**()



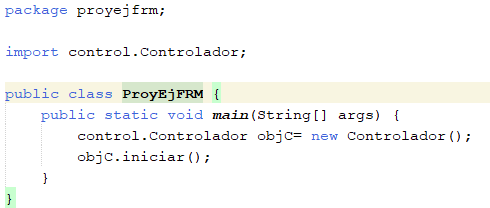
1. Genere constructores y método iniciar, agregue el método .addActionListener(this) a los botones o controles que vaya a programar.

Insertando imagen...

**Clase Controlador**



**Clase Main**



**Próxima clase**

Implementar Ejercicio ejemplo Interfaz gráfica material complementario

Implementar Ejercicio Vehiculos con Interfaz gráfica