



Ciclo 2

Semana 7

Pruebas de código e interfaz gráfica de usuario

Lectura 2 - Interfaz gráfica de usuario (GUI)



| Interfaz gráfica de usuario (GUI)



La interfaz de usuario reúne recursos, sistemas y metodologías para facilitar la interacción de los usuarios con ordenadores o máquinas, donde la interacción sea lo más sencilla e intuitiva posible, en donde la funcionalidad, facilidad de uso y la apariencia tienen un papel importante. Debe tener una compatibilidad entre el usuario final, la aplicación, las tareas y procesos que desarrolle.

Existen diferentes modelos de interface de usuario como por ejemplo CLI (Interfaz de línea de comandos) donde se introducen comandos específicos directamente pero que normalmente requiere conocimientos adicionales funcionando como líneas de instrucciones donde se introduce información o nuevos comandos. También están los TUI (Interfaz de usuario de texto) donde se introducen textos para ejecutar órdenes y transmitir o recibir información y es muy usado en la instalación de sistemas operativos. Encontramos también los VUI (Interfaz de usuario de voz), NUI (Interfaz de usuario natural), y las GUI (Interfaz gráfica de usuario) sobre las cuales centraremos nuestra atención en este documento.

La **GUI (Graphical User Interface)** es el medio de interacción entre el usuario final de la aplicación y el modelo del mundo.



Es la interfaz más usada en la actualidad por su facilidad de uso y sencillez en general, donde el usuario interactúa a través de periféricos como el teclado o el ratón, y se basa en gráficos generalmente simples que describen acciones a través de botones, imágenes, iconos, menús y muchos tipos de componentes, que podrán realizar diferentes tipos de funciones como el encendido y apagado de dispositivos, control funciones, manejo de archivos, comunicación con otros sistemas, envío o recepción de información, temas de configuración, sistemas de ayuda, etc.

En general se puede decir que las interfaces gráficas de usuario con programas que se encargan de interactuar entre el sistema y el usuario a través de un lenguaje gráfico interactivo o entorno visual de imágenes, objetos gráficos o animaciones que serán las que representen la información del sistema y sus posibles acciones, dando un entorno amigable, de fácil de uso, intuitiva, clara y agradable.

Estas han ido evolucionando a través de los años en donde inicialmente fueron creadas como una simulación de un escritorio de oficina pues las primeras computadoras fueron diseñadas exclusivamente como herramientas de trabajo. Luego con la llegada de nuevos productos fueron cambiando tanto que hoy en día no se concibe un sistema informático sin el uso de herramientas virtuales.

Existen muchas tecnologías y lenguajes para detallar la GUI de una aplicación. Ésta puede ser:



Semana 7

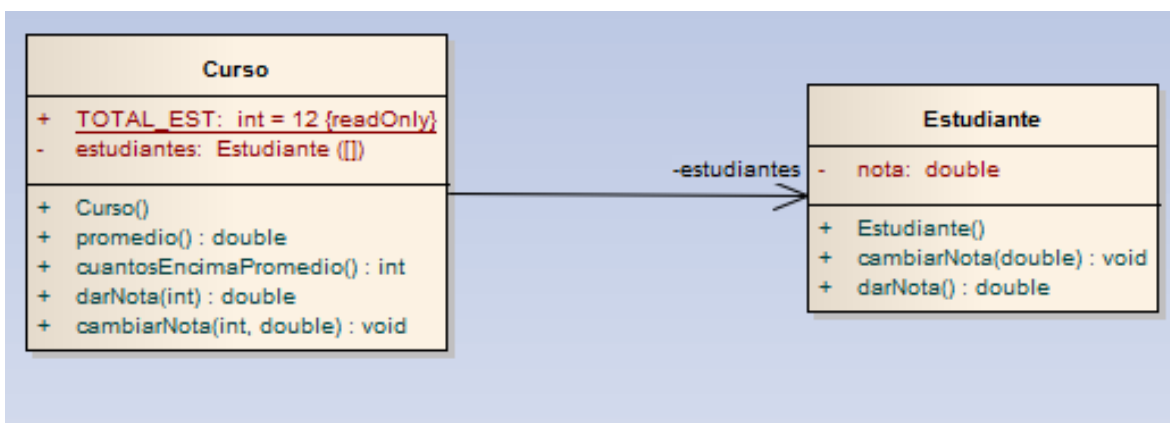
Pruebas de código e interfaz
gráfica de usuario

- Standalone = Escritorio (Java swing).
- Web = A través de un navegador (HTML5, CSS3, Javascript).
- Móvil = A través de dispositivos móviles (IOS, Android, Windows Phone...).

Ejemplo:

Por ejemplo, se tiene un sistema de manejo de notas:

Éste es nuestro MODELO, en dónde especificamos la organización de clases y objetos que estructuran nuestra aplicación.

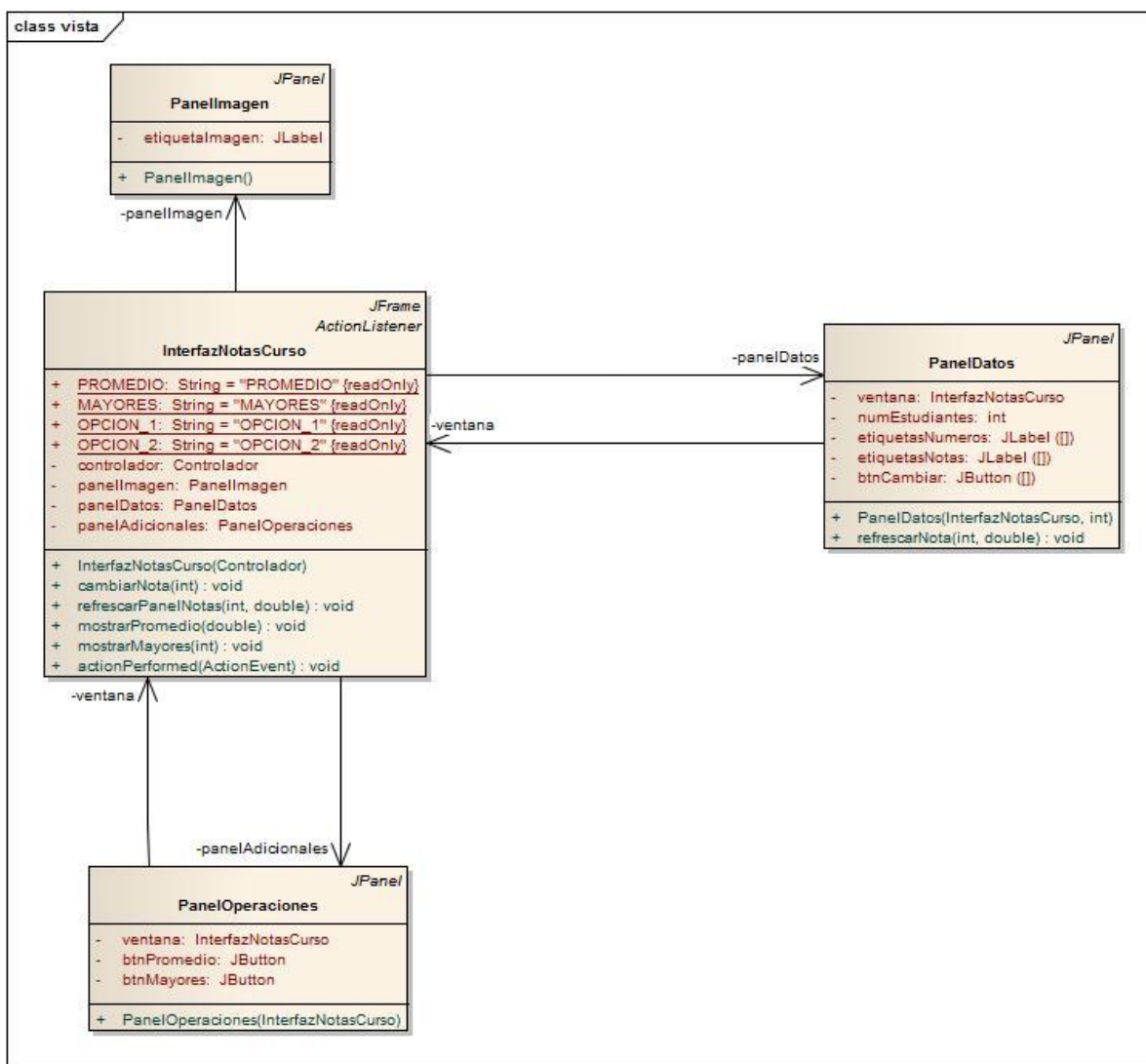


Ésta es nuestra **VISTA**, en dónde especificamos la organización de clases y objetos gráficos que proveen la visualización de información y la interacción con el usuario.



Semana 7

Pruebas de código e interfaz gráfica de usuario

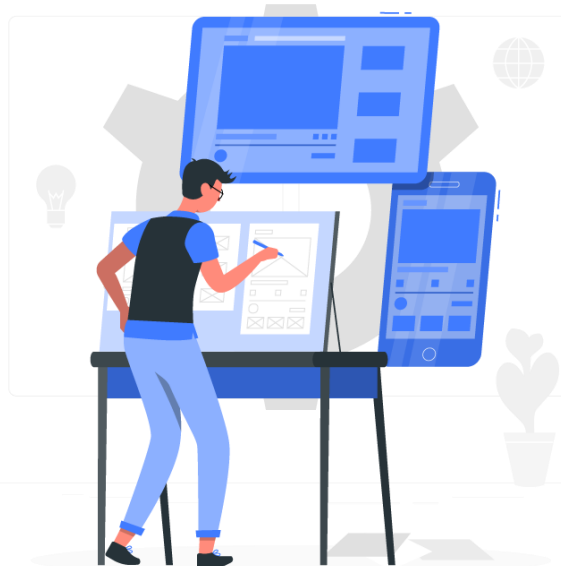


Nuestra **VISTA** resulta siendo la Interfaz Gráfica de Usuario (GUI)



La interface gráfica contiene elementos como lo son la contenedora principal de todos los elementos gráficos de nuestra aplicación, divisiones dentro de la ventana principal que se encargan de agrupar los elementos gráficos por contenido y uso, facilitándole al usuario localizarlos y usarlos. Botones, Etiquetas, Imágenes, Etc.

Diseño de pantalla



El diseño de pantalla muestra el diseño de la interfaz gráficas de usuario, a través de la disposición de cada uno de los elementos que se mostraran en pantalla, en donde cobra gran



importancia la ubicación de cada uno de los elementos como lo son la navegación, textos, encabezados, imágenes, iconos, botones, en fin, todo lo que hace parte de la interfaz de usuario.

Es importante que al momento de diseñar la interfaz se tenga en cuenta para qué tipo de aplicación se está haciendo, si es para un sitio web, webs móviles o aplicaciones de Smartphone o de equipos de escritorio, en fin, cualquier aplicación donde el elemento central es la interacción con el usuario, y estos pueden ser presentados en diferentes tamaños de pantalla, por tanto el formato de salida se puede convertir en un reto, y debe mantener los requisitos estéticos que correspondan sin perder de vista la usabilidad y la experiencia del usuario.

Las guías de estilo deben tener en cuenta factores como la arquitectura de información, su disposición general, asignación de funciones a menús, navegación, contenido, y a estilos de interacción como el control con ratón, teclado, etc. También debe tener en cuenta conceptos tipográficos, esquemas de color, estilos de ventanas, pop-ups, iconos, botones y animaciones, formatos de números y fechas, entre otras.

Algunos componentes de la GUI



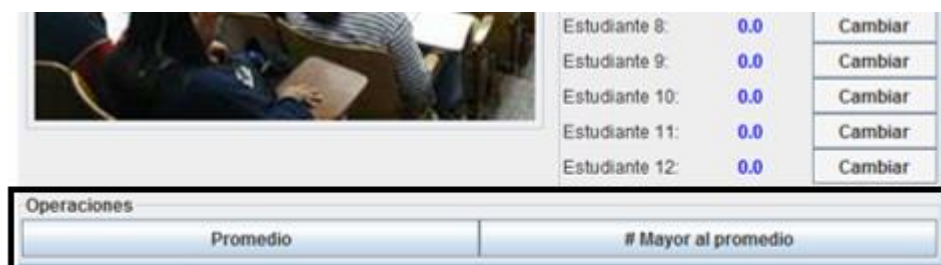


Ventanas

A las ventanas se les puede adicionar componentes para minimizarse, cerrarse, moverse o maximizarse.



Y botones con funcionalidades.



En java se tienen algunos métodos para ventanas como lo son:

- **setSize(ancho, alto):** permite cambiar el alto y ancho de la ventana.
- **setResizable(booleano):** Indica si el usuario puede o no cambiar el tamaño de la ventana.
- **setDefaultCloseOperation(EXIT_ON_CLOSE):** Indica que la aplicación debe terminar su ejecución (destruir el proceso en memoria) en el momento en el que se cierre la ventana. EXIT_ON_CLOSE es una constante de la clase.
- **setVisible(booleano):** Hace aparecer o desaparecer la ventana de la pantalla, dependiendo del valor booleano ingresado.



Semana 7

Pruebas de código e interfaz
gráfica de usuario

- **add(componente):** Agrega un componente gráfico a la ventana.

Distribuidores gráficos

Los Paneles se implementan generalmente en una clase aparte en la **VISTA**.

Todo contenedor requiere que se le especifique cómo se distribuirán sus elementos.

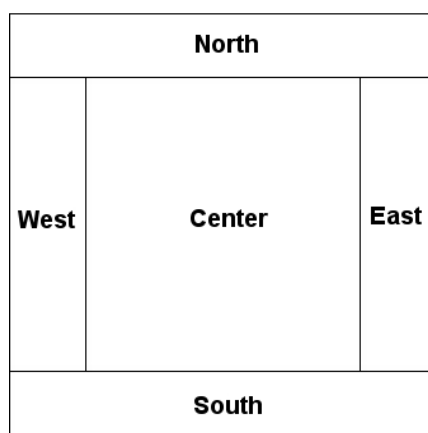
setLayout(....)

Se tienen dos tipos de distribuidores:

- **BorderLayout:** Haciendo uso de la rosa de los vientos, donde se define el distribuidor en el método constructor.

```
setLayout(new BorderLayout());
```

```
add(componente, BorderLayout.NORTH);
```



- **GridLayout:** Pensando en una Matriz.

```
setLayout(new GridLayout(filas, columnas));
```

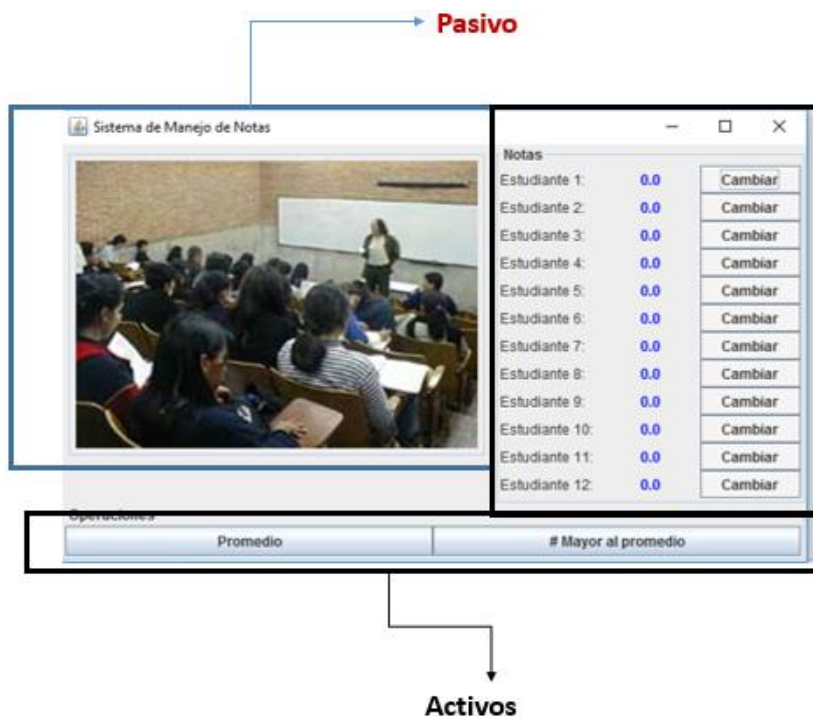
```
add(componente);
```



| | | | |
|--------|----|----|----|
| fila 1 | 1 | 2 | 3 |
| fila 2 | 4 | 5 | 6 |
| fila 3 | 7 | 8 | 9 |
| fila 4 | 10 | 11 | 12 |

Interacción

Un panel puede ser de tipo activo o pasivo.



Pero, ¿Cómo le indicamos a un Panel que va a ser Activo o no?

- Implementamos ActionListener en la Ventana Principal.



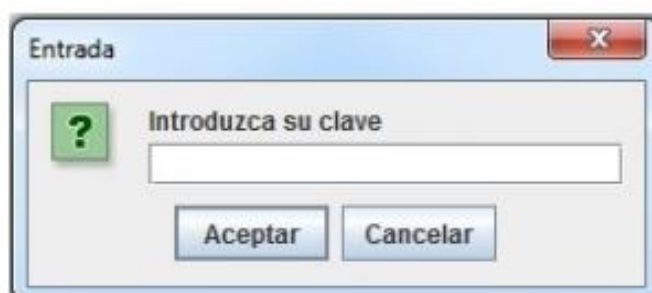
Semana 7

Pruebas de código e interfaz
gráfica de usuario

- Lo relacionamos con la Ventana Principal, pasándole una instancia de ésta por parámetro.
- Le asignamos al panel botones con la capacidad de recibir eventos del usuario.
- Conectamos el Modelo y la Vista a través de un Controlador.

Mensajes para capturar información

Hay ventanas con mensajes que no son solo informativos sino que pueden solicitar introducir información.



Mensajes en una ventana

Los mensajes que se muestran al usuario hace parte de la interfaz y puede ser para un mensaje de error, mensajes informativos, de advertencia, de confirmación, etc.

