

**Nombre Completo: Univ. Ricardo David La Madrid Alarcón**

**Asignatura: PROGRAMACIÓN III**

**Carrera: INGENIERÍA DE SISTEMAS**

**Paralelo:**  **PROG (1)**

**Docente: Lic. William R. Barra Paredes**

**fecha: 17/06/2020**

**Tarea Manejo de SWING y SPRING BOOT**

**Manejo de Spring y SWING**

Unifranz - Cbba

Programación III

**Visión general**

Se tiene como visión general a la capacidad de poder implementar y generar un aplicación de escritorio utilizando SWING y SPRING.

**Objetivos**

1. Manejo de SWING.
2. Diseño y manejo de Componentes(JFRAME, JLABEL, JBUTTON, etc).
3. Diseño y manejo de ventanas(FLOWLAYOUT, GRIDLAYOUT, BORDERLAYOUT).
4. Manejo eventos, listeners, etc

**Especificaciones**

Debe de generar un documento (.**pdf**), este documento también  debe estar disponible en **github**, en la carpeta **Hito4** debe crear un nuevo directorio de nombre **Tareas** y ahi debe estar el documento pdf adicionalmente todo el código debe estar disponible aquí.

La generación de este documento debe tener una **carátula**, **preguntas teóricas** y **preguntas prácticas**. Cada pregunta debe tener como respuesta una imagen(**capturas del EDITOR y la app corriendo**) del correcto funcionamiento y el **código** que resuelve el problema, adicionalmente una breve explicación de la solución de la pregunta.

Formato Carátula:

|  |
| --- |
| **UNIVERSIDAD PRIVADA FRANZ TAMAYO**  **DEFENSA HITO 4 - TAREA FINAL**  Nombre Completo: **Univ. Nombres Apellidos**  Asignatura: **PROGRAMACIÓN III**  Carrera: **INGENIERÍA DE SISTEMAS**  Paralelo: **PROG (1)**  Docente: **Lic. William R. Barra Paredes**  fecha: **xx/xx/2020**  github: **LINK de la carpeta Hito4 de GIT** |

**Parte Teórica.**

* Preguntas.

Responda de manera breve, clara y concisa posible.

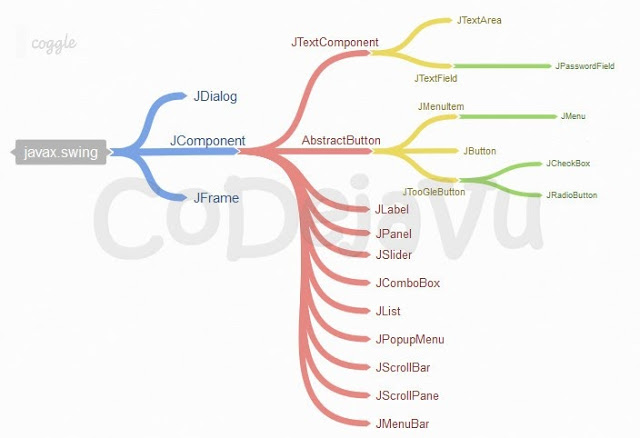
* **Defina que es SWING, a que se refiere cuando se habla de AWT.**

**SWING**. -es un paquete que hace parte de la Java Foundation Classes o más conocida como JFC, la cual provee herramientas o facilidades para la construcción de**GUI's** o interfaces Graficas de Usuario

**AWT**(Abstract Window Toolkit) al igual que **Swing**es un conjunto de librerias enfocadas a la construcción de interfaces

* **Qué son los componentes SWING, mencionar algunos de ellos.**

Los componentes son las partes visuales con funciones, como botones, tablas, marcos, etc.



* **Importancia de JFRAME.**

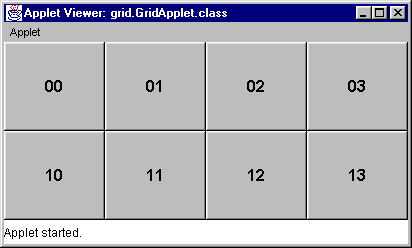
Tiene una importancia mayor porque es la ventana principal en la cual podremos ir añadiendo los demás componentes.

* **Qué es un LISTENER en SWING.**

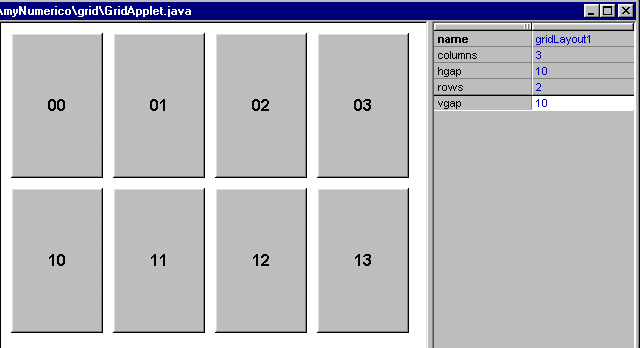
se encargan de controlar los eventos, esperan a **que** el evento se produzca y realiza una serie de acciones. Según el evento, necesitaremos un **Listener que** lo controle.

* **Describa cómo funciona paso a paso el componente GRIDLAYOUT.**

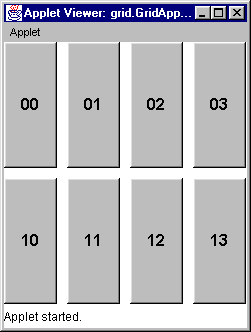
Este gestor dispone los controles en forma de un matriz tal como puede verse en la figura. Tenemos ocho botones dispuestos en dos filas y en cuatro columnas.



Para disponer los controles de esta manera, hemos de seleccionar el objeto *gridLayout1* en el panel de estructura y cambiar las popiedades **columns** y **rows** tal como se ve en la figura inferior. Opcionalmente podemos establecer un espaciado vertical y horizontal entre los controles, introduciendo nuevos valores en los editores asociados a las propiedades **hgap** y **vgap**.



Los controles ocupan todo el panel, de modo que cuando se cambian las dimensiones del applet los controles cambian de tamaño para ajustarse a sus nuevas dimensiones, tal como se ve en la figura inferior.



* **Describa cómo funciona paso a paso el componente FLOWLAYOUT.**

1. Crear los botones (objetos de la clase Button) y el gestor de diseño (objeto de la clase FlowLayout)

Button btn1 = new Button();

FlowLayout flowLayout1 = new FlowLayout();

1. Establecer sus propiedades en init

btn1.setFont(new Font("Dialog", 1, 16));

btn1.setLabel("1");

flowLayout1.setHgap(20);

1. Establecer el gestor de diseño del applet (o de un panel) mediante setLayout

this.setLayout(flowLayout1);

1. Añadir los controles al applet (o a un panel) mediante add

this.add(btn1, null);

Lo que se ha dicho para un applet vale para cualquier panel, ya que un applet no es otra cosa que un panel especializado.

* **Describa cómo funciona paso a paso el componente BORDERLAYOUT.**

Los pasos para establecer el gestor *BorderLayout* son distintos a los empleados para el gestor *FlowLayout*.

1. Crear los botones (objetos de la clase *Button*) y el gestor de diseño (objeto de la clase *BorderLayout*)

Button btnOeste = new Button();

BorderLayout borderLayout1 = new BorderLayout();

1. Establecer sus propiedades en *init*

btnOeste.setFont(new Font("Dialog", 1, 16));

btn1.setLabel("Oeste");

1. Añadir los controles al applet (o a un panel) mediante *add,*indicando en el segundo argumento la posición que ocupará cada control en el panel mediante miembros estáticos de la clase *BorderLayout*.

this.add(btnOeste, BorderLayout.WEST);

**Parte Práctica.**

Debe de trabajarlos en un nuevo proyecto mismo que debe estar en la carpeta de github del hito actual (HITO 4).

* GRIDLAYOUT

Mostrar un ejemplo en donde se use un GRIDLAYOUT.

* Como se usa.
* Cómo funciona
* En el componente principal JFRAME debe mostrarse el uso.
* FLOWLAYOUT

Mostrar un ejemplo en donde se use un FLOWLAYOUT.

* + Como se usa.
  + Cómo funciona
  + En el componente principal JFRAME debe mostrarse el uso.
* BORDERLAYOUT

Mostrar un ejemplo en donde se use un BORDERLAYOUT.

* + Como se usa.
  + Cómo funciona
  + En el componente principal JFRAME debe mostrarse el uso.