Teoría Desarrollo de Interfaces

Índice

A considerar

Texto

Tamaño de caracteres >= 12.
• El espaciado debe de ser proporcional.
• Fuentes fácilmente legibles.
• Evitar en la medida de posible las mayúsculas.

Color

• Tratar siempre elegir las combinaciones de los colores que sean compatibles, ejemplos de colores que hay que evitar elegir (rojo-verde, azul-amarillo, rojo-azul, verde-azul).
• No usar elementos desproporcionadamente grandes brillantes existentes en la pantalla.
• Para personas que tienen daltonismo se recomienda el uso de códigos redundantes.

Inspección de estándares

Observación de campo
Trata de visitar el puesto de trabajo donde estén las actividades y los usuarios representativos de lo que se pretenda estudiar.

Grupos de discusión
Un grupo de discusión dirigido también denominado Focus Group es una de las diversas técnicas las cuales sirven para recolectar datos donde normalmente en un grupo en torno 6 a 9 usuarios discuten para criticar aspectos clave del sistema.

Grabación
En este método se trabaja directamente sobre el sistema donde se analizará directamente cómo los usuarios interaccionan con el sistema, de este modo se tiene mucho mejor en cuenta la frecuencia con la cual un usuario usa una característica concreta o la asiduidad de un mensaje de fallos concreto.

Entrevistas
Tener entrevistas es una manera súper óptima de recoger diversas experiencias que hayan podido tener los usuarios con un sistema interactivo.

Cuestionarios
Resultan ser menos flexibles que las entrevistas, pero tienen una diferencia fundamental que son muchísimo más fácilmente analizables.

Tipos de Cuestionarios

Tipo de Pregunta	Descripción	Ejemplos de Uso
Pre-test	Se recopila información y el perfil de los participantes antes de realizar tareas o pruebas.	Recopilar datos demográficos y preparar a los participantes para la actividad.
Post-tarea	Se recogen conclusiones de los participantes después de que completan cada tarea o actividad.	Obtener retroalimentación inmediata sobre la experiencia de los usuarios.
Post-test	Similar al pre-test, se recopila información y el perfil de los participantes después de las tareas.	Evaluar si hubo cambios en el perfil o la experiencia después de las actividades.
Preguntas Generales	Se usan para crear perfiles basados en la población y el propósito del estudio.	Recopilar información demográfica y contextual como edad, sexo, ocupación laboral, residencia, etc.
Abierta	Permiten respuestas detalladas y subjetivas para obtener sugerencias y detectar problemas no anticipados.	¿Tiene alguna sugerencia o comentario adicional?
Escalar	Los encuestados indican respuestas en una escala numérica para medir satisfacción, opinión, etc.	Indique su nivel de satisfacción: Poca (1) - (2) - (3) - (4) - Mucha (5).
Opción Múltiple	Permiten seleccionar varias respuestas de una lista de opciones para obtener múltiples respuestas.	Marque cuáles de las siguientes videoconsolas tiene en su hogar: PS4 [ ], PS5 [ ], Nintendo Switch [ ], XBOX One [ ], XBOX Series X [].
Ordenadas	Requieren que los encuestados ordenen respuestas en un orden específico, como preferencias o uso.	Ordene de mayor a menor uso de vehículos: Patines (0), Bicicleta (1), Patinete eléctrico (2),...

Principios Mandel (1997)

Colocar a los Usuarios en Control de la Interfaz	Reducir la Carga de Memoria de los Usuarios
- Permitir el uso del teclado y el ratón	- Proporcionar pistas visuales
- Permitir a los usuarios cambiar la atención	- Proporcionar opciones por defecto
- Mostrar mensajes y textos descriptivos	- Proporcionar atajos
- Proporcionar acciones inmediatas, reversibles y realimentación	- Emplear metáforas del mundo real
- Permitir personalizar la interfaz	- Emplear la revelación progresiva para evitar abrumar al usuario. Mostrar la información poco a poco.
- Permitir manipular los objetos de la interfaz	- Promover la claridad visual
- Acomodar a los usuarios con diferentes niveles de habilidad	

Tipos de Layouts en Windows

FlowLayout
Coloca los componentes en una sola fila o columna, agregándoles uno al lado del otro, con la opción de alinearlos en la izquierda, centro o derecha del contenedor.

BorderLayout
Divide el contenedor en cinco regiones: norte, sur, este, oeste y centro. Cada región puede contener un único componente que ocupa todo el espacio de esa región.

GridLayout
Organiza los componentes en una cuadrícula con un número fijo de filas y columnas, de manera que cada celda contiene un único componente de tamaño uniforme.

GridBagLayout
Es una versión más flexible del GridLayout que permite a los componentes ocupar múltiples celdas y ajustar su tamaño y posición de manera más precisa.

BoxLayout
Permite organizar los componentes en una sola fila o columna, con la opción de cambiar entre orientación horizontal y vertical. No se adapta automáticamente a las dimensiones del contenedor.

GroupLayout
Diseñado para herramientas de diseño de interfaces gráficas. Divide el diseño en grupos horizontales y verticales, permitiendo un control detallado sobre la colocación de los componentes.

SpringLayout
Similar a GroupLayout, permite especificar la distancia entre componentes y su posición en el contenedor. Útil en herramientas de diseño de interfaces.

CardLayout
Permite gestionar varios paneles como una pila de cartas, mostrando solo un panel a la vez. Se usa para crear interfaces con pestañas o pantallas de bienvenida.

Absolute Layout
Permite poner los componentes donde el usuario quiera. Y no es Responsive.

Tipos de Contenedores

Existen muchos contenedores y algunos suelen usar como layout los populares que son más conocidos y se suelen centrar en los que son más simples de usar. Otros optan por usar **layout especializados** que fueron diseñados para ser usados para un contenedor muy concreto como es el caso del **JScrollPane** que usa un layout que no está pensado para construir una interfaz gráfica sino como base para ser usado como un **gestor de barra de desplazamiento** que debe tener en cuenta cuando el contenido de su contenedor excede sus límites para poder habilitar la correcta barra de desplazamiento y que se pueda ver el contenido de forma correcta sin modificar o ampliar el tamaño del contenedor.

También puede ocurrir que un contenedor tenga como **layout** por defecto un **valor null** lo que significa que tiene una **referencia nula** (dirección de memoria cuyo contenido es nulo es decir no apunta a ningún sitio y por tanto no contiene nada) y por tanto no usa ningún layout, siendo **responsabilidad del programador** realizar el trabajo que realizaría un layout para poder colocar los componentes de forma correcta en el contenedor o puede **asignarle un layout de su elección** aunque eso siempre es posible haya o no uno por defecto.

Contenedor	Layout por defecto	Tipo
JFrame	BorderLayout	Popular
JDialog	BorderLayout	Popular
JApplet	BorderLayout	Popular
JPanel	BorderLayout	Popular
JScrollPane	ScrollPaneLayout	Especializado
JSplitPane	BasicHorizontalLayoutManager	Especializado
JTabbedPane	TabbedPaneLayout	Especializado
JDesktop	Null	Nulo
JInternalFrame	Handler	Privado y Especializado
JToolBar	DefaultToolBarLayout	Privado y Especializado

Todos los layouts heredan de la **clase LayoutManager** de ahí que se conozca con el nombre de Layout manager comúnmente que significa gestor o administrador de orientación o diseño.
Cuando asignamos un layout a un contenedor éste se almacena como un objeto de la clase padre es decir como un **LayoutManager** por eso si queremos extraer el layout del contenedor usando el **método getLayout** deberemos convertirlo con un casting a nuestro layout deseado para poder acceder a los métodos propios de nuestro layout.

Contenedores Secundarios

Los contenedores que no actúan como principal o raíz deben ir dentro de uno ya sea de forma directa o indirecta y son los siguientes:

JPanel
Contenedor simple y versátil para organizar componentes dentro de un panel.
Puedes ocultar, mover o agrupar componentes en paneles.

JScrollPane
Contenedor que agrega barras de desplazamiento horizontal y vertical a un componente interno, como un panel o un área de texto.

JSplitPane
Divide el contenido en dos secciones que se pueden ajustar horizontal o verticalmente. Permite al usuario cambiar el tamaño relativo de las secciones.

JTabbedPane
Crea pestañas en una ventana, donde cada pestaña contiene un contenedor independiente con sus componentes. Útil para organizar información en pestañas.

JDesktop
Contenedor para ventanas internas (JInternalFrame), lo que permite crear ventanas dentro de una ventana principal. Las ventanas internas no pueden salir de los límites de la ventana principal.

JToolBar
Contenedor para crear una barra de herramientas en la interfaz de usuario. Puedes personalizar y reorganizar las herramientas en la barra.

Añadir un JPanel a un JFrame

```
6
7  public class Ventana{
8
9      public static void main(String[] args){
10         JFrame ventana = new JFrame("Primera Ventana");
11         ventana.setSize(400, 400);
12
13         JPanel panel = new JPanel();
14         JButton boton = new JButton("Bienvenido");
15
16         panel.add(boton);
17         ventana.getContentPane().add(panel);
18
19         ventana.setVisible(true);
20     }
21 }
```

ComponentUI

Lo que permite manipular la apariencia de un componente es el objeto **ComponentUI** que realiza todas las tareas de dibujo, manejo de eventos, control de tamaño, etc. Dicho objeto está contenido en la **clase abstracta JComponent** que es una clase de la que heredan todos los componentes salvo los contenedores principales que son especiales y es otra razón más por la cual el marco de una ventana no se ve afectado en su apariencia.

Crear un menú

JMenuBar, creamos la barra de menús.
• setJMenuBar establece la barra de menú en el contenedor.
• JMenuItem creamos el ítem.
• Con menu1.add(mi1) lo añadimos al menú en este caso clientes.

```
public class Ventana8Menu extends JFrame implements ActionListener {
private JMenuBar mb;
private JMenu menu1;
private JMenuItem mi1, mi2, mi3;


mb = new JMenuBar();
setJMenuBar(mb);
menu1 = new JMenu("Clientes");
mb.add(menu1);


mi1 = new JMenuItem("Añadir");
mi1.addActionListener(this);
menu1.add(mi1);


mi2 = new JMenuItem("Listado");
mi2.addActionListener(this);
menu1.add(mi2);


mi3 = new JMenuItem("Salir");
mi3.addActionListener(this);
menu1.add(mi3);
}


public static void main(String[] args) {
SwingUtilities.invokeLater(() -> {
Ventana8Menu formulario1 = new Ventana8Menu();
formulario1.setBounds(10, 20, 300, 200);
formulario1.setVisible(true);
});
}
}
```

Sintaxis importante

• En el caso del contenedor principal **JFrame** es necesario configurar que cuando se pulse en el botón de cierre de ventana desde su marco de ventana acabe matando al proceso de la aplicación:

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

Frame sería en este caso una variable de tipo Jframe.

• El método **pack()** se usa para ajustar el tamaño de la ventana al contenido de la misma. Por ejemplo si hemos usado un tamaño preferido con **setPreferredSize**.

• Podemos configurar las coordenadas de la ventana usando el método **setlocation(x,y)**, en donde las posiciones 0,0, corresponden a la esquina superior izquierda.