

Módulo 2: Diseño de la interfaz de usuario: Vistas y Layouts

Infórmate

Contenidos

Creación de una interfaz de usuario por código

Creación de una interfaz de usuario usando XML

Test - Creación de una interfaz por código y XML

Test - El fichero R.java

Edición visual de las vistas

Test - Edición visual de vistas

Atributos de la clase View

Atributos de la clase TextView

Test - Atributos de la clase View y TextView

Los Layouts en Android

Test - Tipos de Layouts

Una aplicación de ejemplo: Asteroides

Test - Atributos de los Layouts

Recursos alternativos

Test - Recursos alternativos

Tipos de recursos y recursos del sistema

Acceso a los recursos

Estilos y temas en Android

Los estilos

Heredar de un estilo propio

Los temas

Test - Estilos y temas

Uso de Vistas y Layouts

Acceder y modificar las propiedades de las vistas por código

Test - Uso de Vistas y Layouts

Examen . Módulo 2

Material complementario: Referencia clase View y sus descendientes

Encuesta módulo 2 (creada por 2ª vez)

Examen . Módulo 2

Has de acertar al menos un 60% con un número de intentos ilimitado.

Examen no temporizado.

Prueba realizada

Tu resultado en el test ha sido: 72%

Has superado el test.

Tus respuestas

¿Cuál es una ventaja de diseñar el interfaz de usuario en XML en lugar de en código?

☒ Permite realizar el diseño de forma independiente a la programación

☐ Se consigue una ejecución más rápida

☐ Los programas obtenidos son más seguros

iBIEN!

¿Qué afirmación es cierta sobre la clase R.java?

☐ Asigna un valor entero a cada recurso

☐ Permite identificar los recursos desde código Java

☒ Se genera automáticamente y nunca has de modificarlo

☐ Todas las respuestas anteriores son ciertas

Inténtalo otra vez.

A partir del API 8 aparece match_parent. Su significado es totalmente equivalente a:

☐ wrap_content

☐ layout_width

☐ layout_height

☒ fill_parent

iBIEN!

Cuál de los siguientes atributos NO modificará la forma en la que se visualizará la vista.

☐ padding

☒ id

☐ visibility

☐ background

☐ style

Vídeo

¡BIEN!

Todos los atributos siguientes se aplican a descendientes de TextView, pero hay uno que además se aplica a todo descendiente de View. ¿Cual es?

- ☐ gravity
- ☒ visibility
- ☐ width
- ☐ typeface
- ☐ text_style

¡BIEN!

¿Cuál de los siguientes Layouts permite posicionar las vistas en coordenadas x, y?

- ☒ AbsoluteLayout
- ☐ RelativeLayout
- ☐ FrameLayout
- ☐ LinearLayout
- ☐ TableLayout

¡BIEN!

¿El atributo orientation a qué tipo de Layout se aplica?

- ☐ AbsoluteLayout
- ☐ RelativeLayout
- ☐ FrameLayout
- ☒ LinearLayout
- ☐ TableLayout

¿Cuál de los siguientes NO es un sufijo válido para definir recursos alternativos?

- ☐ layout
- ☒ port
- ☐ land
- ☐ xlarge
- ☐ hdpi

¿Qué afirmación es INCORRECTA sobre los recursos alternativos?

- ☒ Se pueden combinar varias opciones para definir un recurso (ejemplo: idioma + densidad gráfica)
- ☐ Hay que definir un identificador de recurso diferente por cada recurso alternativo
- ☐ El sistema elige el recurso según el sufijo de la carpeta donde se almacena

¿Qué diferencia hay entre un estilo y un tema?

- ☐ Un tema hereda de un estilo
- ☐ Un estilo se aplica a una actividad mientras que un tema se aplica a una vista

- ☒ Un estilo se aplica a una vista mientras que un tema se aplica a una actividad
- ☐ No existen diferencias, es solo una distinción semántica

El siguiente método es ejecutado al pulsar sobre un botón, donde entrada es un TextView.

```
public void sePulsa0(View view){  
    view.setText( (String) entrada.getTag());  
}
```

¿Qué realiza?

- ☐ Añade al texto de la vista entrada un texto definido en el atributo tag de la vista pulsada
- ☒ El texto del botón cambiará a un texto definido en el atributo tag del TextView
- ☐ Permite que cualquier botón pueda añadir el carácter "0" al texto de la vista entrada
- ☐ Añade al texto de la vista entrada un "0"
- ☐ Añade al texto de la vista pulsada un "0"

Enhorabuena. Terminaste este módulo.

[anterior](#)

[siguiente](#)



Telefónica Learning Services



2012 MiriadaX | [Aviso legal](#)