

CUESTIONARIO EXAMEN ELECTIVA IV

Para crear un menu de opciones, escoger lo correcto para que el código sea el correcto.

```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater =  ;  
    inflater. (R.menu.menu_principal,  );  
    return  ;  
}
```

Para crear un menú emergente, escoger lo correcto para que el código sea el correcto.

```
public void showPopup(View v) {  
     popup = new PopupMenu(this, v);  
    MenuInflater inflater = popup. ;  
    inflater. (R.menu.menu_emergente,  );  
      
}
```

```
public void showPopup(View v) {  
    PopupMenu popup = new PopupMenu(context: this, v);  
    MenuInflater inflater = popup.getMenuInflater();  
    inflater.inflate(R.menu.menu_emergente, popup.getMenu());  
    popup.show();  
}
```

Se tiene un radio button para que el usuario escoja si quiere ir al activity de nuevo usuario o nuevo proveedor.

- 1.- Si escoge ir a nuevo usuario entonces deberá ir a una actividad que tiene nombre de *new_usuario.class*
- 2.- Si escoge ir a nuevo proveedor entonces deberá ir a una actividad que tiene nombre de *new_proveedor.class*

Escoja lo correcto:

```
public void onRadioButtonClicked(View view) {  
     checked = ((RadioButton) view).isChecked();  
    switch (  ) {  
        case R.id.nuevo_usuario:  
            if (  )  
                Intent intent = new Intent(  ,  );  
                 (intent, 0);  
                break;  
        case R.id.nuevo_proveedor:  
            if (  )  
                Intent intent = new Intent(  ,  );
```

Escoja lo correcto:

```
public void onRadioButtonClicked(View view) {  
    boolean checked = ((RadioButton) view).isChecked();  
    switch (view.getId()) {  
        case R.id.nuevo_usuario:  
            if (checked) {  
                Intent intent = new Intent(v.getContext(), new_usuario.class);  
                startActivityForResult(intent, 0);  
                break;  
            }  
        case R.id.nuevo_proveedor:  
            if (checked) {  
                Intent intent = new Intent(v.getContext(), new_proveedor.class);  
                startActivityForResult(intent, 0);  
                break;  
            }  
    }  
}
```

```
public void onRadioButtonClicked(View view) {  
    boolean checked = ((RadioButton) view).isChecked();  
    // Check which radio button was clicked  
    switch (view.getId()) {  
        case R.id.rdb_masculino:  
            if (checked) {  
                // Masculino  
                Toast.makeText(getApplicationContext(), text: "Seleccionó Masculino", Toast.LENGTH_SHORT).show();  
                break;  
            }  
        case R.id.rdb_femenino:  
            if (checked) {  
                // Femenino  
                Toast.makeText(getApplicationContext(), text: "Seleccionó Femenino", Toast.LENGTH_SHORT).show();  
                break;  
            }  
    }  
}
```

Para llamar a otro activity desde el activity actual, complete el siguiente código según corresponda.

```
btn_aceptar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent( , );  
        intent. ( "name_usuario",txt_nombre.getText().toString());  
        startActivityForResult( , 0);  
    }  
});
```

Para capturar la información que fue enviada desde el actividad de origen,

```
Bundle bundle = getIntent(). ;  
 nombre_jugador=bundle.getString("name_usuario");
```

Para llamar a otro activity desde el activity actual, complete el siguiente código según corresponda.

```
btn_aceptar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent( v.getContext() , MainActivity.class );  
        intent. putExtra ( "name_usuario",txt_nombre.getText().toString());  
        startActivityForResult( intent , 0);  
    }  
});
```

Para capturar la información que fue enviada desde el actividad de origen,

```
Bundle bundle = getIntent(). getExtras() ;  
 String nombre_jugador=bundle.getString("name_usuario");
```

```
btn_aceptar.setOnClickListener((view) -> {  
    Intent intent = new Intent(view.getContext(),MainActivity.class);  
    if(txt_nombre.getText().toString().equals(nombre) && txt_contra.getText().toString().equals(contrasena)) {  
        intent.putExtra( name: "name_usuario",txt_nombre.getText().toString());  
        startActivityForResult(intent, requestCode: 0);  
    }  
});
```

```
Bundle bundle = getIntent().getExtras();  
String nombre_jugador = bundle.getString( key: "name_usuario");  
txtjugador.setText(" " + nombre_jugador);
```

TEORIA SON LAS LECCIONES

Respecto a los métodos de una actividad:

- Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.

Seleccione una:

- ☐ a. onStart()
- ☒ b. onResume()
- ☐ c. onCreate()
- ☐ d. onRestart()

[Quitar mi selección](#)



El componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema es:

Seleccione una:

- ☐ a. Service
- ☐ b. Content provider
- ☐ c. Broadcast Receiver
- ☐ d. Intent

- **Broadcast Receiver**

Los controles básicos, como cuadros de texto, botones, listas desplegables o imágenes son:

Seleccione una:

- ☐ a. Activity
- ☐ b. Widget
- ☒ c. View
- ☐ d. Content provider

Respecto al estado de una actividad:

Cuando la actividad no es visible. El programador debe guardar el estado de la interfaz de usuario, preferencias

Seleccione una:

- ☒ a. Parada
- ☐ b. Destruída
- ☐ c. Activa
- ☐ d. Visible

[Quitar mi selección](#)



¿Las plataformas móviles posibilitan el acceso ubicua e instantáneamente a múltiples recursos disponibles a través de Internet?

Seleccione una:

- ☒ Verdadero
☐ Falso



Escoja dos retos de las plataformas móviles

Seleccione una o más de una:

- ☒ a. Estándares de seguridad
☐ b. Velocidad de respuesta
☐ c. Circuitos de alta velocidad
☒ d. Interferencias

En el opción de match_parent, la view será tan grande como su contenido

Seleccione una:

- ☒ Verdadero
☐ Falso

¿Donde se puede adquirir el contexto?

Seleccione una:

- ☒ a. Contexto social
☐ b. Contexto global
☐ c. Contexto fisico
☐ d. Contexto en linea

[Quitar mi selección](#)

Todos los proyectos incluyen el archivo res/values/ string.xml

Seleccione una:

- ☐ Verdadero
☐ Falso

```
<resources>
  <string name="app_name">GeoQuiz</string>
  <string name="pregunta_text">Quito es la capital del Ecuador</string>
  <string name="verdadero_button">Verdadero</string>
  <string name="falso_button">Falso</string>
</resources>
```

- Todos los proyectos incluyen el archivo **res/values/ string.xml**
- Para obtener el valor de un recurso string desde cualquier xml, se lo referencia como **@string/nombre_recurso**

Entre mayor sea el nivel del API usada, a menos dispositivos podrá llegar nuestra aplicación.

Seleccione una:

☐ Verdadero

☐ Falso

ACTIVIDAD

Una actividad en Android puede estar en uno de estos cuatro estados:

- **Activa (Running):** La actividad está encima de la pila, lo que quiere decir que **es visible y tiene el foco**.
- **Visible (Paused):** La actividad **es visible pero no tiene el foco**. Se alcanza este estado cuando **pasa a activa otra actividad con alguna parte transparente o que no ocupa toda la pantalla**. Cuando una actividad está tapada por completo, pasa a estar parada.
- **Parada (Stopped):** Cuando la actividad **no es visible**. El programador debe guardar el estado de la interfaz de usuario, preferencias, etc.
- **Destruída (Destroyed):** Cuando la **actividad termina al invocarse el método `finish()`**, o es matada por el sistema.

Cada vez que una actividad cambia de estado se van a generar eventos que podrán ser capturados por ciertos métodos de la actividad. A continuación se muestra un esquema que ilustra los métodos que capturan estos eventos.

Escoja lo correcto:

```
public void onRadioButtonClicked(View view) {  
    boolean checked = ((RadioButton) view).isChecked();  
    switch (view.getId()) {  
        case R.id.nuevo_usuario:  
            if (checked) {  
                Intent intent = new Intent(v.getContext(), new_usuario.class);  
                startActivityForResult(intent, 0);  
                break;  
            }  
        case R.id.nuevo_proveedor:  
            if (checked) {  
                Intent intent = new Intent(v.getContext(), new_proveedor.class);  
                startActivityForResult(intent, 0);  
                break;  
            }  
    }  
}
```

La siguiente línea de código es correcta:

Button aceptar = (Button) findViewById(R.id.btn_aceptar)

Seleccione una:

☐ Verdadero

☒ Falso

```
btnVerdadero = findViewById(R.id.button_verdadero);  
btnFalso = findViewById(R.id.button_falso);  
btnSiguiente = findViewById(R.id.button_siguiente);  
textpregunta = findViewById(R.id.text_pregunta);  
btnAtras = findViewById(R.id.button_atras);
```

Entre mayor sea el nivel del API usada, a menos dispositivos podrá llegar nuestra aplicación.

Seleccione una:

- ☐ Verdadero
- ☐ Falso

¿Donde se puede adquirir el contexto?

Seleccione una:

- ☒ a. Contexto social
- ☐ b. Contexto global
- ☐ c. Contexto fisico
- ☐ d. Contexto en linea

[Quitar mi selección](#)

En el opción de match_parent, la view será tan grande como su contenido

Seleccione una:

- ☒ Verdadero
- ☐ Falso

Escoja dos retos de las plataformas móviles

Seleccione una o más de una:

- ☒ a. Estándares de seguridad
- ☐ b. Velocidad de respuesta
- ☐ c. Circuitos de alta velocidad
- ☒ d. Interferencias

¿Las plataformas móviles posibilitan el acceso ubicua e instantáneamente a múltiples recursos disponibles a través de Internet?

Seleccione una:

- ☒ Verdadero
- ☐ Falso



La versión de la API escogida al momento de crear un proyecto en android indica:

Seleccione una:

- ☐ a. La versión mínima con la cual funcionará la APP creada.
- ☐ b. La versión con la cual no funcionará la APP creada
- ☐ c. La versión con la cual funcionará la APP creada
- ☐ d. La versión máxima con la cual funcionará la APP creada

El comercio móvil son actividades que se realizan en entornos inalámbricos.

Seleccione una:

- ☐ Verdadero
- ☐ Falso

Entre mayor sea el nivel del API usada, a menos dispositivos podrá llegar nuestra aplicación.

Seleccione una:

- ☐ Verdadero
- ☐ Falso

Los controles básicos, como cuadros de texto, botones, listas desplegables o imágenes son:

Seleccione una:

- ☐ a. View
- ☐ b. Content provider
- ☐ c. Widget
- ☐ d. Activity

Desde la vista del proyecto:

Contiene la definición en XML de muchos de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, icono, ...), sus componentes (pantallas, servicios, ...), o los permisos necesarios para su ejecución

Seleccione una:

- ☐ a. /app/AndroidManifest.xml
- ☐ b. /app/src/main/Manifest.xml
- ☐ c. /app/src/main/Android.xml
- ☐ d. /app/src/main/AndroidManifest.xml

Usando la vista de proyecto.

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, layouts, cadenas de texto, etc.

Seleccione una:

- ☐ a. /app/src/main/res/
- ☐ b. /app/src/res/main
- ☐ c. /app/res/java
- ☐ d. /app/src/main/java

CODIGOS

Conectando widgets con el código



```
package com.curso.miprimeraaplicacion

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
```

```
class MainActivity : AppCompatActivity() {

    var btnVerdadero: Button? = null
    var btnFalso: Button? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

```
        btnVerdadero = findViewById(R.id.btn_verdadero) as Button
        btnFalso = findViewById(R.id.btn_falso)
```

Inflar Variables

Eventos en Android

```
ass MainActivity : AppCompatActivity(), View.OnClickListener
```

```
    override fun onClick(v: View?) {
    }
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        btnVerdadero?.setOnClickListener {
        }
```

```
        btnVerdadero?.setOnClickListener(View.OnClickListener {
```

```
        })
```

```
        btnVerdadero?.setOnClickListener(this)
```

```
    }
```

Identificar y describir
cada uno de los
métodos eventos

Método agrupa un conjunto de instrucciones
por ejemplo una función

Evento es una actividad por la que pasa un
componente. Ejemplo evento clicked,
eliminar, etc

25

Eventos en Android

```
package com.curso.miprimeraaplicacion
```

```
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
```

```
class MainActivity : AppCompatActivity() {
```

```
    var btnVerdadero: Button? = null
    var btnFalso: Button? = null
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
```

```
        btnVerdadero = findViewById(R.id.btn_verdadero) as Button
        btnFalso = findViewById(R.id.btn_falso)
```

```
        btnVerdadero?.setOnClickListener {
        }
```

```
        btnFalso?.setOnClickListener {
```

```
        }
```

```
    }
```

Identificar y
describir cada uno
de los métodos
eventos

```

Spinner spinner = (Spinner) findViewById(R.id.spn_tipo_telefono);

// Create ArrayAdapter using the string array and default spinner layout.
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource( context: this, R.array.labels_array,
    android.R.layout.simple_spinner_item);

// Especifique el diseño que se utilizará cuando aparezca la lista de opciones.
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

//Aplicar el adaptador al spinner
spinner.setAdapter(adapter);

spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        mSpinnerLabel = parent.getItemAtPosition(position).toString();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        .
    }
}

```

CALENDARIO

- ▶ Implementar los métodos haciendo uso de los @override
 - ▶ onDateSet()
 - ▶ public void onDateSet(DatePicker view, int year, int month, int day)
- ▶ Quitar la constructor vacío DatePickerFragment.
- ▶ Reemplazar **onCreateView()** con **onCreateDialog()** que retorna **Dialog**, and anota **onCreateDialog()** with **@NonNull** para indicar que no puede retornar nulo.

```

public class DatePickerFragment extends DialogFragment implements DatePickerDialog.OnDateSetListener {

    private DatePickerDialog.OnDateSetListener listener;

    public static DatePickerFragment newInstance(DatePickerDialog.OnDateSetListener listener) {
        DatePickerFragment fragment = new DatePickerFragment();
        fragment.setListener(listener);
        return fragment;
    }

    public void setListener(DatePickerDialog.OnDateSetListener listener) {
        this.listener = listener;
    }

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the current date as the default date in the picker.
        final Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);
        // Create a new instance of DatePickerDialog and return it.
        return new DatePickerDialog(getActivity(), listener, year, month, day);
    }

    @Override
    public void onDateSet(DatePicker view, int year, int month, int day) {

```

En el activity principal

```

public void showDatePickerDialog(View v) {
    final TextView etPlannedDate = (TextView) findViewById(R.id.txt_fecha);

    DatePickerFragment newFragment = DatePickerFragment.newInstance(new DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker datePicker, int year, int month, int day) {
            // +1 because January is zero
            final String selectedDate = day + "/" + (month+1) + "/" + year;
            etPlannedDate.setText(selectedDate);
        }
    });

    newFragment.show(getSupportFragmentManager(), "Calendario");
}

```

```

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.usuario:
            Toast.makeText(context, this, text: "Usuario", Toast.LENGTH_LONG).show();
            return true;
        case R.id.proveedor:
            Toast.makeText(context, this, text: "Proveedor", Toast.LENGTH_LONG).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

public void showPopup(View v) {
    PopupMenu popup = new PopupMenu(context, this, v);
    MenuInflater inflater = popup.getMenuInflater();
    inflater.inflate(R.menu.menu_emergente, popup.getMenu());
    popup.show();
}

```

ESTILOS

Sin estilos:

```
<textView android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="#00FF00"
    android:typeface="monospace"
    android:text="@string/hello" />
```

Con estilos:

```
<textView style="@style/CodeFont"
    android:text="@string/hello" />
```

```
<style name="CodeFont" parent="@android:style/TextAppearance.Medium">
    <item name="android:layout_width">fill_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">#00FF00</item>
    <item name="android:typeface">monospace</item>
</style>
```

Herencia

- El atributo parent sirve para heredar propiedades de otros estilos, podemos heredar tanto de estilos del sistema como de los nuestros propios.

Del sistema:

```
<style name="GreenText" parent="@android:style/TextAppearance">
    <item name="android:textColor">#00FF00</item>
</style>
```

De nuestros propios estilos:

```
<style name="CodeFont.Red">
    <item name="android:textColor">#FF0000</item>
</style>
<style name="CodeFont.Red.Big">
    <item name="android:textSize">30sp</item>
</style>
```

Aplicar estilos y temas a la interfaz gráfica

Para aplicar un tema a una actividad o aplicación usaremos:

```
<application android:theme="@style/CustomTheme">
</application>
```

Para aplicarlos sobre actividades, usamos:

```
<activity android:theme="@android:style/Theme.Dialog"></activity>
<activity android:theme="@android:style/Theme.Translucent"></activity>
```