

**PRÁCTICA PMDM05**  
**Interacciones entre clases y actividades**

29/02/2024

CÉSAR BOUZAS SOTO

## Contenido

---

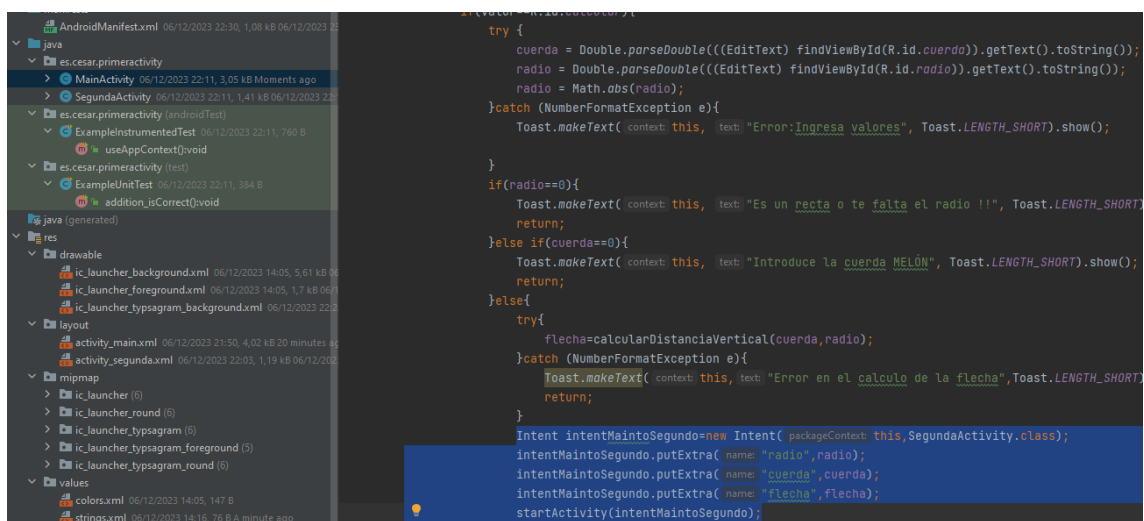
1	Intents: Explica los tipos y pon un ejemplo de cada uno.....	2
2	Manejo de eventos de clic: pon dos ejemplos.....	3
2.1	Manejo de clic mediante XML .....	3
2.2	Manejo de clic mediante código.....	3
3	Menús contextuales, flotantes y emergentes: pon un ejemplo de cada uno .....	4
3.1	Menú contextual (Contextual menú). .....	4
3.2	Menú Flotante. ....	5
3.3	Menú emergente (Popup menú):.....	5

# 1 Intents: Explica los tipos y pon un ejemplo de cada uno.

Los "intents" en el desarrollo de aplicaciones Android son objetos fundamentales que permiten la comunicación entre diferentes componentes de una aplicación y entre aplicaciones dentro del sistema operativo Android. Estos objetos se utilizan para iniciar acciones o procesos, así como para transmitir datos entre los diversos elementos de una aplicación.

Existen varios tipos de intents en Android:

**Intent explícito:** Este tipo de intent se emplea para iniciar un componente específico dentro de la misma aplicación. Por ejemplo, al hacer clic en un botón, se puede utilizar un intent explícito para abrir una nueva pantalla dentro de la aplicación.



**Intent implícito:** Se utiliza para iniciar un componente que puede ser manejado por cualquier aplicación que coincida con el criterio especificado en el intent. Un ejemplo común es abrir un enlace web en un navegador externo.

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.ejemplo.com"));
startActivity(intent);
```

**Intent de difusión (Broadcast Intent):** Este tipo de intent se utiliza para enviar mensajes a múltiples componentes o aplicaciones interesadas en eventos del sistema o de la propia aplicación.

```
Intent intent = new Intent("com.ejemplo.ACCION_CUSTOM");
intent.putExtra("mensaje", "¡Hola desde el broadcast!");
sendBroadcast(intent);
```

**Intent de servicio:** Se emplea para iniciar un servicio en segundo plano para realizar operaciones prolongadas sin necesidad de una interfaz de usuario. Por ejemplo, descargar un archivo mientras el usuario realiza otras tareas en la aplicación.

```
Intent intent = new Intent(this, MiServicio.class);
intent.putExtra("operacion", "descargar_archivo");
startService(intent);
```

Intent de notificación: Utilizado para mostrar notificaciones al usuario desde una aplicación. Por ejemplo, notificar al usuario sobre un nuevo mensaje recibido en una aplicación de mensajería.

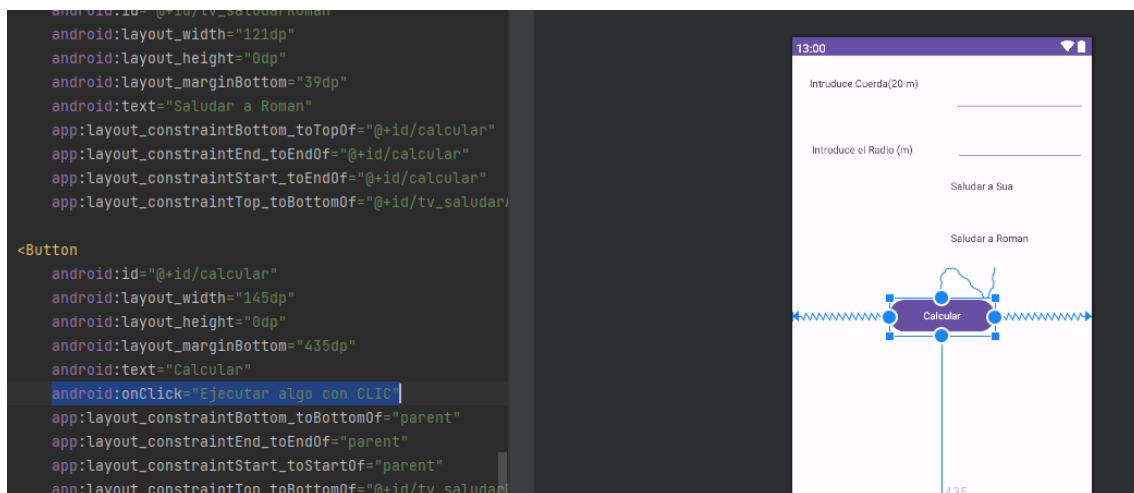
```
Intent intent = new Intent(this, DetalleNotificacionActivity.class);
intent.putExtra("notificacion_id", 456);
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setContentIntent(pendingIntent)
    .setContentTitle("Nueva notificación")
    .setContentText("¡Tienes una nueva notificación!")
    .setSmallIcon(R.drawable.ic_notification)
    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
notificationManager.notify(notificationId, builder.build());
```

## 2 Manejo de eventos de clic: pon dos ejemplos.

Son dos los enfoques comunes para manejar eventos de clic en Android, uno mediante XML y otro mediante código Java. Ambos logran el mismo resultado, pero es el usuario el que debe elegir el que mejor se adapte a sus preferencias y necesidades en el proyecto.

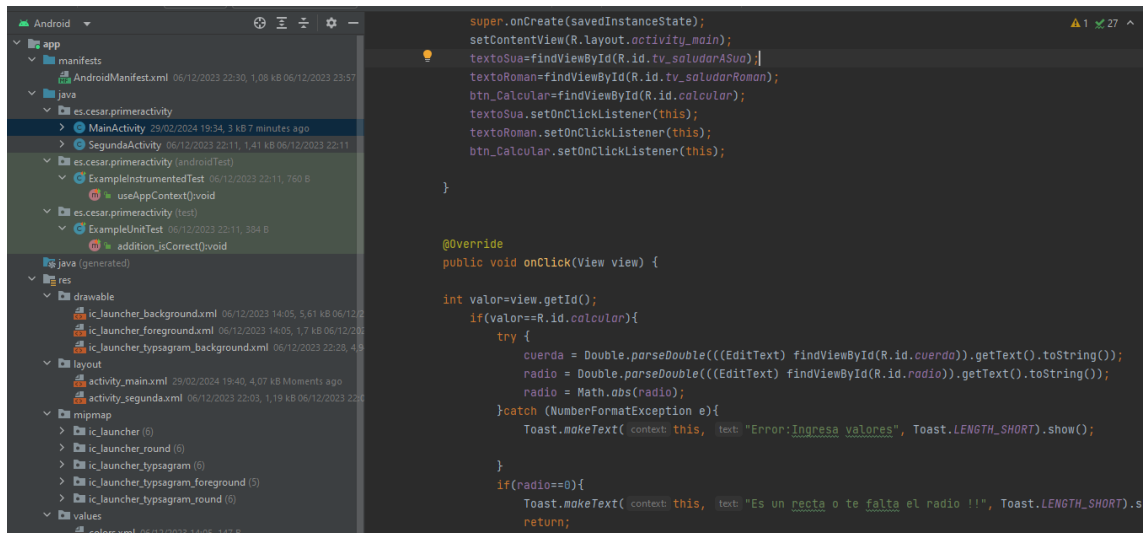
### 2.1 Manejo de clic mediante XML

En el archivo activity\_main.xml, agregarías un atributo android: onClick al botón para especificar el método que se llamará cuando se haga clic en él:



### 2.2 Manejo de clic mediante código.

Es en la actividad (MainActivity.java), donde se agregará un listener al botón para manejar el evento de clic:



### 3 Menús contextuales, flotantes y emergentes: pon un ejemplo de cada uno

#### 3.1 Menú contextual (Contextual menú).

Los menús contextuales en Android son menús que aparecen en respuesta a una interacción prolongada (presionar y mantener) sobre un elemento de la interfaz de usuario, como un elemento de lista o una vista específica. Estos menús proporcionan opciones relevantes para el elemento seleccionado.

Supongamos una lista de elementos en tu actividad y se quiere mostrar un menú contextual cuando el usuario mantenga presionado uno de los elementos:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.context_menu, menu);
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.edit_option:
            // Acción para editar el elemento seleccionado
            return true;
        case R.id.delete_option:
            // Acción para eliminar el elemento seleccionado
            return true;
        default:
            return super.onContextItemSelected(item);
    }
}
```

### 3.2 Menú Flotante.

Los menús flotantes en Android son menús que aparecen como una lista de opciones sobre la interfaz de usuario, generalmente en forma de botones flotantes. Estos menús se muestran en respuesta a una acción específica del usuario, como tocar un botón de acción flotante.

Un ejemplo de cómo crear y mostrar un menú flotante utilizando un FloatingActionButton:

```
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        PopupMenu popup = new PopupMenu(MainActivity.this, view);
        popup.getMenuInflater().inflate(R.menu.floating_menu, popup.getMenu());
        popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.action_add:
                        // Acción para agregar un nuevo elemento
                        return true;
                    case R.id.action_search:
                        // Acción para buscar elementos
                        return true;
                    default:
                        return false;
                }
            }
        });
        popup.show();
    }
});
```

### 3.3 Menú emergente (Popup menú):

Los menús emergentes en Android son similares a los menús flotantes, pero se pueden mostrar en cualquier lugar de la pantalla en respuesta a una acción del usuario, como tocar y mantener presionado un área de la pantalla.

Un ejemplo de cómo crear y mostrar un menú emergente en respuesta a un evento de largo clic en un botón:

```
Button button = findViewById(R.id.button);
button.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {
        PopupMenu popup = new PopupMenu(MainActivity.this, view);
        popup.getMenuInflater().inflate(R.menu.popup_menu, popup.getMenu());
        popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.action_copy:
                        // Acción para copiar el contenido
                        return true;
                    case R.id.action_paste:
                        // Acción para pegar el contenido
                        return true;
                    default:
                        return false;
                }
            }
        });
        popup.show();
        return true;
    }
});
```