|  |  |  |
| --- | --- | --- |
| **Nombre**: Miguel Alejandro Ruvalcaba Flores | | **Matrícula**: 2730919 |
| **Nombre del curso:**  Desarrollo de aplicaciones en plataforma Android | **Nombre del profesor**:  Rafael Nuño Fragoso | |
| **Módulo**:  Tema 1 | **Actividad**:  Actividad 1: Investigación de elementos Android | |
| **Fecha**: 18/08/19 | | |
| **Bibliografía**: Bibliografía Google Developers. (2019). *Actividades*. Obtenido de Developer.android: https://developer.android.com/guide/components/activities.html?hl=ES  Google Developers. (2019). *Clases*. Obtenido de Developers.Android: https://developer.android.com/reference/android/widget/Button | | |

**Objetivo:**

Investigar elementos utilizados en la sintaxis de Android para poderlos aplicar posteriormente en el desarrollo de una aplicación.

**Resultados:**

* Investiga en developer.android.com cada uno de los métodos definidos en el ciclo de vida de una actividad, describiendo cuando se usa.
  + Una actividad es un componente de la aplicación que contiene una pantalla con la que solo usuarios pueden interactuar para realizar una acción.
  + El ciclo de vida se compone de los métodos:
    - onCreate (): recibe una llamada cuando se crea la actividad por primera vez. En este punto se deben realizar todas las configuraciones estáticas normales, por ejemplo, crear vistas, enlazar datos con listas, etc. En este método se le pasa un objeto agrupado que contiene el estado anterior de la actividad, si se hubiera capturado ese estado. Siempre seguido del método onStart ();
    - onRestart (): recibe una llamada después de que se detiene la actividad, justo antes de que vuelva a iniciarse. Simpre seguido de onStart();
    - onStart (): recibe una llamada justo antes de que la actividad se vuelva visible para el usuario. Seguido por onResume () si la actividad pasa a primer plano, o por onStop () si se oculta.
    - onResume (): recibe una llamada justo antes de que la actividad comience a interactuar con el usuario. En este momento la actividad se encuentra en la parte superior de la pila de actividades, y recibe las entradas del usuario. Siempre seguido de onPause();
    - onPause (): Recibe una llamada cuando el sistema está a punto de reanudar otra actividad. Se usa generalmente se usa para confirmar los cambios sin guardar. Lo que sea que haga, debe hacerlo muy rápido porque la siguiente actividad no se reanudará hasta que el método regrese.

Seguido por onResume() si la actividad vuelve al primer plano, o por onStop() si se vuelve invisible para el usuario.

* + - onStop (): Recibe una llamada cuando la actividad ya no es visible para el usuario. Esto puede ocurrir porque se la destruyó o porque se reanudó otra actividad y la está cubriendo.

Seguido por onRestart() si la actividad vuelve a interactuar con el usuario, o por onDestroy() si la actividad desaparece.

* + - onDestroy (): Recibe una llamada antes de que se termine la actividad. Esta es la última llamada que recibirá la actividad. Se lo puede llamar porque la actividad está finalizando, o porque el sistema termina temporalmente esa instancia de la actividad para ahorrar espacio. Se diferencian estos dos escenarios con el método isFinishing().

(Google Developers, 2019)

* En el mismo sitio deberás investigar el nombre de la clase y el nombre y descripción de al menos tres métodos para los siguientes elementos de interfaz gráfica:
  + Layout
  + Texto Editable
    - Public class EditText
    - Extends TextView
      * Es un elemento de interfaz de usuario para ingresar y modificar texto. Cuando se define un widget de edición de texto, debe especificar el atributo R.styleable.TextView\_inputType
      * La elección del tipo de entrada configura el tipo de teclado que se muestra, por ejemplo, si es de tipo numérico, solo aparecerá el teclado numérico, pero su acepta letras, aparecerá el teclado completo.
  + Botón
    - Public Cass Button
    - Extends TextView
      * Es un elemento de interfaz de usuario que el usuario puede tocar o hacer clic para realizar una acción
      * Para mostrar un botón en una actividad se debe agregar al archivo XML el siguiente código:

<Botón

android: id = "@ + id / button\_id"

android: layout\_height = "wrap\_content"

android: layout\_width = "wrap\_content"

android: text = "@ string / self\_destruct" />

* + Etiqueta
    - Public class TextView
    - Extends View implements ViewTreeObserver.OnPreDrawListener
      * Es un elemento de usuario que muestra texto al usuario
  + Imagen
    - Public abstract class Image
    - Extends Object implements AutoCloseable
      * Es un único búfer de imagen completo para usar con una fuente de medios como a MediaCodec o a CameraDevice.
      * Esta clase permite un acceso eficiente y directo de la aplicación a los datos de pixeles de la imagen a través de uno o más byteBuffers. Cada búfer se encapsula en una Plane que describe el diseño de los datos de pixeles en ese plano. Debido a este acceso directo, y a diferencia de la clase Bitmap, las imágenes no se pueden usar directamente como recursos de la interfaz de usuario.
* (Google Developers, 2019)

Conclusión:

Durante el desarrollo de esta actividad se aprendieron los conceptos básicos de como desarrollas actividades en Android studio y los elementos básicos de interfaz gráfica, las cuales se usarán más adelante en la creación de apps moviles.