# Instituto Tecnológico Superior de Jerez



**Jerez de García Salinas a 03 de mayo del 2019**

**Ricardo Benjamín Viramontes Juárez**

benja120599@gmail.com

**S17070162**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

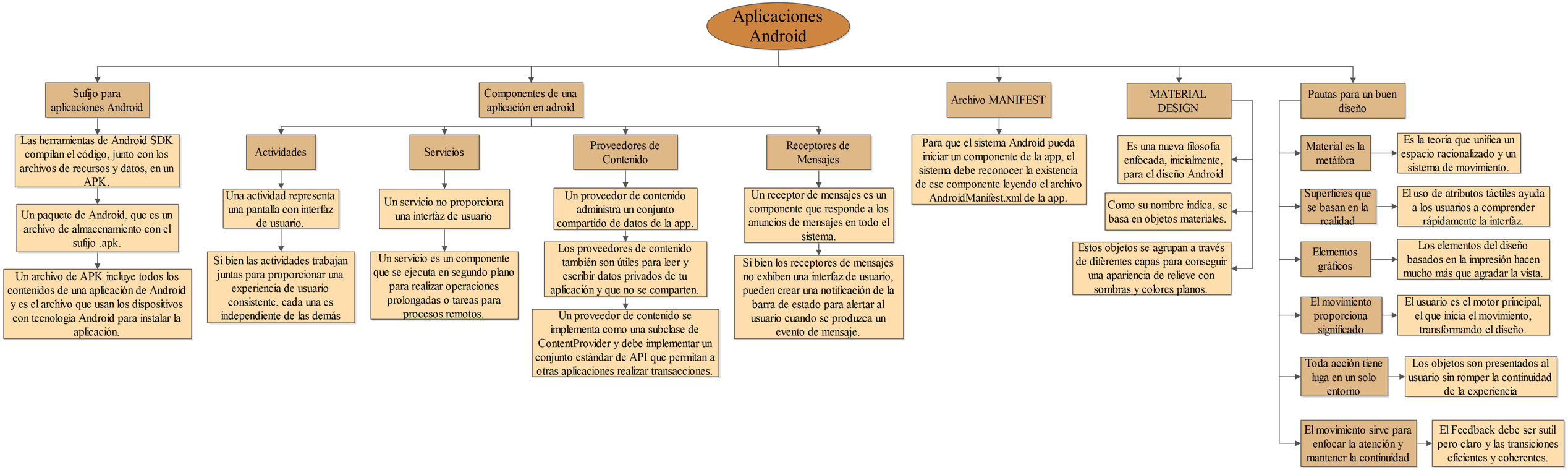
**Tópicos Avanzados de Programación.**

**4to. SEMESTRE.**

**Tema 5**

**Mapa Conceptual Android.**

**I.S.C. Salvador Acevedo Sandoval.**


Aplicaciones Android
Sufijo para aplicaciones Android
Las herramientas de Android SDK compilan el código, junto con los archivos de recursos y datos, en un APK.
Un paquete de Android, que es un archivo de almacenamiento con el sufijo .apk.
Un archivo de APK incluye todos los contenidos de una aplicación de Android y es el archivo que usan los dispositivos con tecnología Android para instalar la aplicación.
Componentes de una aplicación en adroid
Actividades
Una actividad representa una pantalla con interfaz de usuario.
Si bien las actividades trabajan juntas para proporcionar una experiencia de usuario consistente, cada una es independiente de las demás
Servicios
Un servicio no proporciona una interfaz de usuario
Un servicio es un componente que se ejecuta en segundo plano para realizar operaciones prolongadas o tareas para procesos remotos.
Proveedores de Contenido
Un proveedor de contenido administra un conjunto compartido de datos de la app.
Los proveedores de contenido también son útiles para leer y escribir datos privados de tu aplicación y que no se comparten.
Un proveedor de contenido se implementa como una subclase de ContentProvider y debe implementar un conjunto estándar de API que permitan a otras aplicaciones realizar transacciones.
Receptores de Mensajes
Un receptor de mensajes es un componente que responde a los anuncios de mensajes en todo el sistema.
Si bien los receptores de mensajes no exhiben una interfaz de usuario, pueden crear una notificación de la barra de estado para alertar al usuario cuando se produzca un evento de mensaje.
Archivo MANIFEST
Para que el sistema Android pueda iniciar un componente de la app, el sistema debe reconocer la existencia de ese componente leyendo el archivo AndroidManifest.xml de la app.
MATERIAL DESIGN
Es una nueva filosofía enfocada, inicialmente, para el diseño Android
Como su nombre indica, se basa en objetos materiales.
Estos objetos se agrupan a través de diferentes capas para conseguir una apariencia de relieve con sombras y colores planos.
Pautas para un buen diseño
Material es la metáfora
Es la teoría que unifica un espacio racionalizado y un sistema de movimiento.
Superficies que se basan en la realidad
El uso de atributos táctiles ayuda a los usuarios a comprender rápidamente la interfaz.
Elementos gráficos
Los elementos del diseño basados en la impresión hacen mucho más que agradar la vista.
El movimiento proporciona significado
El usuario es el motor principal, el que inicia el movimiento, transformando el diseño.
Toda acción tiene luga en un solo entorno
Los objetos son presentados al usuario sin romper la continuidad de la experiencia
El movimiento sirve para enfocar la atención y mantener la continuidad
El Feedback debe ser sutil pero claro y las transiciones eficientes y coherentes.

1. ¿Cuál es el sufijo para las aplicaciones que se instalan en Android?

   Las herramientas de Android SDK compilan el código, junto con los archivos de recursos y datos, en un APK: un paquete de Android, que es un archivo de almacenamiento con el sufijo .apk. Un archivo de APK incluye todos los contenidos de una aplicación de Android y es el archivo que usan los dispositivos con tecnología Android para instalar la aplicación.

2. ¿Cuáles son los 4 componentes que forman a una aplicación Android?

   Actividades
   Una actividad representa una pantalla con interfaz de usuario. Por ejemplo, una aplicación de correo electrónico tiene una actividad que muestra una lista de los correos electrónicos nuevos, otra actividad para redactar el correo electrónico y otra actividad para leer correos electrónicos. Si bien las actividades trabajan juntas para proporcionar una experiencia de usuario consistente en la aplicación de correo electrónico, cada una es independiente de las demás. De esta manera, una aplicación diferente puede iniciar cualquiera de estas actividades (si la aplicación de correo electrónico lo permite). Por ejemplo, una aplicación de cámara puede iniciar la actividad en la aplicación de correo electrónico que redacta el nuevo mensaje para que el usuario comparta una imagen.
   Una actividad se implementa como una subclase de Activity y puedes obtener más información acerca de este tema en la guía para desarrolladores Actividades.

   Servicios
   Un servicio es un componente que se ejecuta en segundo plano para realizar operaciones prolongadas o tareas para procesos remotos. Un servicio no proporciona una interfaz de usuario. Por ejemplo, un servicio podría reproducir música en segundo plano mientras el usuario se encuentra en otra aplicación, o podría capturar datos en la red sin bloquear la interacción del usuario con una actividad. Otro componente, como una actividad, puede iniciar el servicio y permitir que se ejecute o enlazarse a él para interactuar.

   Proveedores de contenido
   Un proveedor de contenido administra un conjunto compartido de datos de la app. Puedes almacenar los datos en el sistema de archivos, en una base de datos SQLite, en la Web o en cualquier otra ubicación de almacenamiento persistente a la que tu aplicación pueda acceder. A través del proveedor de contenido, otras aplicaciones pueden consultar o incluso modificar los datos (si el proveedor de contenido lo permite). Por ejemplo, el sistema Android proporciona un proveedor de contenido que administra la información de contacto del usuario. De esta manera, cualquier app

con los permisos correspondientes puede consultar parte del proveedor de contenido (como ContactsContractData) para la lectura y escritura de información sobre una persona específica.
Los proveedores de contenido también son útiles para leer y escribir datos privados de tu aplicación y que no se comparten. Por ejemplo, la aplicación de ejemplo Bloc de notas usa un proveedor de contenido para guardar notas.

Un proveedor de contenido se implementa como una subclase de ContentProvider y debe implementar un conjunto estándar de API que permitan a otras aplicaciones realizar transacciones.

Receptores de mensajes
Un receptor de mensajes es un componente que responde a los anuncios de mensajes en todo el sistema. Muchos mensajes son originados por el sistema; por ejemplo, un mensaje que anuncie que se apagó la pantalla, que la batería tiene poca carga o que se tomó una foto. Las aplicaciones también pueden iniciar mensajes; por ejemplo, para permitir que otras aplicaciones sepan que se descargaron datos al dispositivo y están disponibles para usarlos. Si bien los receptores de mensajes no exhiben una interfaz de usuario, pueden crear una notificación de la barra de estado para alertar al usuario cuando se produzca un evento de mensaje. Aunque, comúnmente, un receptor de mensajes es simplemente una "puerta de enlace" a otros componentes y está destinado a realizar una cantidad mínima de trabajo. Por ejemplo, podría iniciar un servicio para que realice algunas tareas en función del evento.
Un receptor de mensajes se implementa como una subclase de BroadcastReceiver y cada receptor de mensajes se proporciona como un objeto Intent.

3. ¿Como se "activan" dichos componentes?

Tres de los cuatro tipos de componentes (actividades, servicios y receptores de mensajes) se activan mediante un mensaje asincrónico llamado intent. Las intents enlazan componentes individuales en tiempo de ejecución (son como mensajeros que solicitan una acción de otros componentes), ya sea que el componente le pertenezca a tu aplicación o a otra.

Una intent se crea con un objeto Intent, que define un mensaje para activar un componente específico o un tipo específico de componente; una intent puede ser explícita o implícita, respectivamente.

Para actividades y servicios, una intent define la acción a realizar (por ejemplo, "ver" o "enviar" algo) y puede especificar el URI de los datos en los que debe actuar (entre otras cosas que el componente que se está iniciando podría necesitar saber). Por

ejemplo, una intent podría transmitir una solicitud para que una actividad muestre una imagen o abra una página web. En algunos casos, puedes iniciar una actividad para recibir un resultado; en cuyo caso, la actividad también devuelve el resultado en una Intent (por ejemplo, puedes emitir una intent para que el usuario elija un contacto personal y te lo devuelva; la intent de devolución incluye un URI que apunta al contacto seleccionado).

Para los receptores de mensajes, la intent simplemente define el anuncio que se está transmitiendo (por ejemplo, un mensaje para indicar que la batería del dispositivo tiene poca carga incluye solo una string de acción conocida que indica "batería baja").

El otro tipo de componente, proveedor de contenido, no se activa mediante intents, sino a través de solicitudes de un ContentResolver. El solucionador de contenido aborda todas las transacciones directas con el proveedor de contenido, de modo que el componente que realiza las transacciones con el proveedor no deba hacerlo y, en su lugar, llame a los métodos del objeto ContentResolver. Esto deja una capa de abstracción entre el proveedor de contenido y el componente que solicita información (por motivos de seguridad).

4. ¿Qué es el archivo MANIFEST y para qué sirve?

   Para que el sistema Android pueda iniciar un componente de la app, el sistema debe reconocer la existencia de ese componente leyendo el archivo AndroidManifest.xml de la app (el archivo de "manifiesto"). Tu aplicación debe declarar todos sus componentes en este archivo, que debe encontrarse en la raíz del directorio de proyectos de la aplicación.

5. ¿Cuáles son los estados en los que se puede encontrar una app?

   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.
   - Destruida (Destroyed): Cuando la actividad termina al invocarse el método finish(), o es matada por el sistema.

6. ¿Cuáles son los métodos que permiten manipular dichos estados?

    - onCreate(Bundle): Se llama en la creación de la actividad. Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Puede recibir información de estado de la actividad (en una instancia de la clase Bundle), por si se reanuda desde una actividad que ha sido destruida y vuelta a crear.
    - onStart(): Nos indica que la actividad está a punto de ser mostrada al usuario.
    - onResume(): Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.
    - onPause(): Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra actividad es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.
    - onStop(): La actividad ya no va a ser visible para el usuario
    - onRestart(): Indica que la actividad va a volver a ser representada después de haber pasado por onStop().
    - onDestroy(): Se llama antes de que la actividad sea totalmente destruida. Por ejemplo, cuando el usuario pulsa el botón de volver o cuando se llama al método finish().

7. ¿Qué es y para qué sirve MATERIAL DESIGN?

    Es una nueva filosofía enfocada, inicialmente, para el diseño Android; sin embargo, su implementación en el mundo del desarrollo ha sido tal, que muy pronto comenzó a extenderse a toda la web. Es así como la encontramos, actualmente, no solo en aplicaciones Android, sino en páginas web y demás plataformas de software.
    Como su nombre indica, se basa en objetos materiales. Estos objetos se agrupan a través de diferentes capas para conseguir una apariencia de relieve con sombras y colores planos. Otro de los pilares fundamentales es el uso de animaciones y transiciones lógicas, con el objetivo de dar la sensación de que los objetos se guían por las leyes de la física.

8. ¿Cuáles son las 6 grandes pautas que especifica MATERIAL DESIGN para un buen diseño de apps?

    - Material es la metáfora. Es la teoría que unifica un espacio racionalizado y un sistema de movimiento. El material está basado en la realidad táctil, inspirado en el estudio del papel y tinta, pero tecnológicamente avanzado y abierto a la imaginación y magia.

- Superficies que se basan en la realidad. El uso de atributos táctiles ayuda a los usuarios a comprender rápidamente la interfaz. La flexibilidad de los objetos crea nuevas posibilidades que reemplazan las del mundo físico, sin romper las reglas de la física.
- Elementos gráficos e intencionales. Los elementos fundamentales del diseño basados en la impresión (tipografía, espacio, escala) hacen mucho más que agradar la vista. Crean jerarquía, significado y enfoque. El uso adecuado de colores, imágenes y espacios sumergen al usuario en una mejor experiencia.
- El movimiento proporciona significado. El usuario es el motor principal, el que inicia el movimiento, transformando el diseño.
- Toda acción tiene lugar en un solo entorno. Los objetos son presentados al usuario sin romper la continuidad de la experiencia a medida que se transforma y reorganizan.
- El movimiento sirve para enfocar la atención y mantener la continuidad. El Feedback debe ser sutil pero claro y las transiciones eficientes y coherentes.

9. Menciona 5 "mejores prácticas" indicadas por Google para el "desempeño" (performance) de la aplicación

    - Uso de procesos u subprocesos: La entrada de manifiesto de cada tipo de elemento de componente (<activity>, <service>, <receiver> y <provider>) admite un atributo android:process que puede especificar un proceso en el cual el componente debe ejecutarse.
    - Optimizar para la duración de la batería.
    - Reducir el tamaño del apk
    - Administra la memoria de la aplicación
    - Verificación del comportamiento de la app en el tiempo de ejecución de Android (ART)

10. Menciona 5 "mejores prácticas" indicadas por Google para la "Crear apps para miles de usuarios"

    - Conectividad: Descubre cómo brindar una mejor experiencia a los usuarios de redes más lentas. Enfócate en optimizar imágenes, el uso de redes y la transferencia de datos.
    - Capacidad del dispositivo: Descubre cómo agregar compatibilidad para dispositivos con capacidades diferentes de aquellos para los que sueles desarrollar contenido. Ten en cuenta los diferentes tamaños de pantalla, la compatibilidad con versiones anteriores y el uso eficiente de la memoria.

- Costo de datos: Descubre cómo ayudar a los usuarios a minimizar sus costos de tráfico de red reduciendo el tamaño de las apps y ofreciendo ajustes de red configurables.
- Consumo de batería: Descubre cómo tu app puede ayudar a extender la duración de la batería. Sigue las recomendaciones indicadas para la administración y el control de la batería a fin de asegurarte de que tu app no agote la carga de forma innecesaria.
- Interfaz de usuario y contenido: Descubre cómo presentar el contenido para obtener la mejor experiencia del usuario posible. Entre las áreas principales, se incluyen la capacidad de respuesta de la interfaz, las recomendaciones para IU y la localización.

REFERENCIAS BIBLIOGRÁFICAS.

Developers Android. (2019). Aspectos fundamentales de la aplicación de Developers Android. Sitio web: https://developer.android.com/guide/components/fundamentals?hl=es-419

Delvis Echeverria. (enero 2019). Tips para Mejorar el Desempeño de tus Aplicaciones Web de SG. Sitio web: https://sg.com.mx/revista/58/tips-para-mejorar-el-desempeno-de-tus-aplicaciones-web

Sonia Ruiz Cayuela. (N.E.). Qué es Material Design y cómo se aplica a un sitio web de Web App Design. Sitio web: https://webappdesign.es/que-es-material-design/