

Universidad Nacional de Costa Rica

Escuela de Informática

I Ciclo, 2019

EIF-411 Diseño y programación de plataformas Móviles

Profesor: Dr. Juan de Dios Murillo Morera

Documento: Laboratorio #3 & #4

Estudiante

Alejandro Gamboa Barahona 10 a.m

Cristian Díaz Jiménez 8 a.m

Índice

Objetivo General	3
Objetivos Específicos	3
Parte A - Reporte.	4
Esquema de la evolución e historia de los dispositivos móviles y tablets.	4
Cuadro comparativo de ventajas y desventajas de las diferentes tecnologías móviles.	5
Principales elementos, herramientas configuraciones y perfiles del Android Studio.	7
Principales Elementos	7
Actividades	7
Ciclo de Vida de una Actividad.	7
Servicios	8
Proveedores de contenido	8
Receptores de mensajes	9
Widget	9
Intent	9
Herramientas	9
Instant Run	9
Editor de código inteligente	9
Emulador rápido y cargado de funciones	10
Configuraciones	10
Perfiles	10
Bibliografía	11

Objetivo General

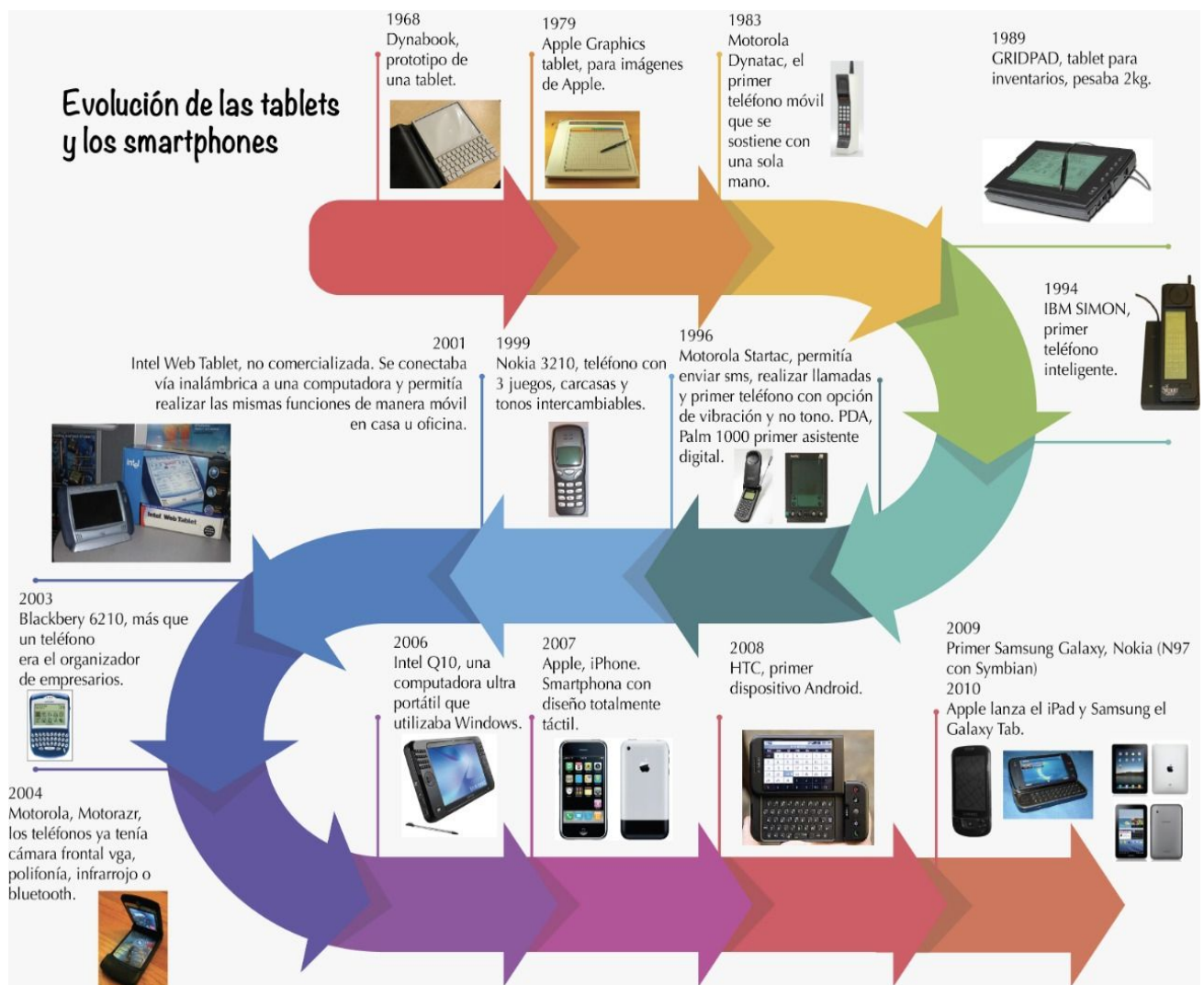
1. Identificar las diferencias entre arquitecturas y entender sus ventajas y desventajas.
2. Desarrollar una aplicación en el IDE Android Studio que sea capaz de navegar entre LAYOUTS a través del uso de ACTIVITY. La aplicación es un prototipo navegable (frontEnd) utilizando datos quemados (uso de intents y paso de objetos).

Objetivos Específicos

1. Identificar las principales tecnologías móviles (iOS/Android/Window Phone).
2. Estudiar la evolución e historia de los smartphones y tablets.
3. Instalar y configurar diferentes emuladores (al menos 2) para programar las aplicaciones.
4. Estudiar diferentes configuraciones y perfiles.
5. Estudiar emuladores existentes.
6. Generar un APK y probarlo en el teléfono.

Parte A - Reporte.

Esquema de la evolución e historia de los dispositivos móviles y tablets.



Evolución de Tablets y Dispositivos Móviles. Desarrollo Propio.

Cuadro comparativo de ventajas y desventajas de las diferentes tecnologías móviles.

Dada la investigación, se realiza un cuadro comparativo de las tres tecnologías móviles más utilizadas en el mercado.

	iOS	Android	Windows Phone
Ventajas	<ul style="list-style-type: none"> - Elevada seguridad: eficiencia frente a las ciberamenazas externas. - Menos fragmentación: cuenta con la tasa de fragmentación más baja en dispositivos móviles. - Mayor exclusividad en el mercado: Todas las apps atraviesan un proceso de revisión manual por parte del equipo de Apple, antes de ser publicadas en la tienda. - Mayor duración de la batería: claramente es un punto muy en alto en los dispositivos Apple. Esto redundará en una envidiable autonomía. - Sincronización entre dispositivos: es una de los pilares de Apple. Brindarle al usuario accesibilidad a toda su información desde cualquier dispositivo Apple de forma sencilla. - Siri: el asistente de voz de iOS resulta verdaderamente cómodo, y se encuentra muy por delante de Google Now. 	<ul style="list-style-type: none"> - Numerosas Aplicaciones: Android nos brinda la posibilidad de instalar aplicaciones de terceros o ajenas a la Tienda. - Excelentes notificaciones: el sistema de notificaciones de Android es líder. Los usuarios les resulta cómodo organizarse con él. - Versatilidad: Android soporta cargadores estándar microUSB o USB Type-C, permite las baterías extraíbles, las tarjetas de memoria y además de la conexión de periféricos USB - Sistema de multitarea inteligente: es capaz de gestionar varias aplicaciones abiertas a la vez dejando en suspensión aquellas que no se utilicen y cerrarlas en caso de resultar ya inútiles para evitar un 	<ul style="list-style-type: none"> - Interfaz sencilla. - Integración total con Windows: Por lo cual al tener Windows en PC, sincronizar el servicio de correo, contactos, OneDrive, y programas de Microsoft. - Copia de Seguridad: Con Windows 10 Mobile es sin duda uno de los elementos que más por encima está con respecto a Android. - Copia de Seguridad: Con Windows 10 Mobile es sin duda uno - Hardware: Es casi imposible que exista un dispositivo móvil en el planeta capaz de soportar lo que un Nokia. - Menor riesgo de adquirir malware: Por el motivo que la mayor parte de los virus están

		<p>consumo de memoria.</p> <ul style="list-style-type: none"> - Comunidad: la facilidad de poseer un dispositivo con Android, hace que la comunidad sea bastante extensa. - Sistema operativo creado por Google: esto facilita el uso de los principales servicios de Internet, como Gmail, Gtalk, Google Maps y YouTube. 	<p>diseñados para Android.</p> <ul style="list-style-type: none"> -Menor riesgo de adquirir malware: Por el motivo que la mayor parte de los virus están diseñados para Android.
Desventajas	<ul style="list-style-type: none"> - Menor flexibilidad: A diferencia que en Android, Apple no brinda diversos launchers para sustituir la apariencia por defecto. - Muy Costoso: Poseer un dispositivo Apple, es costoso. - Sistema Operativo Cerrado: no deja la instalación de aplicaciones de terceros. - Comunidad: Por razones de la accesibilidad a dispositivos con iOS, la comunidad es algo pequeña comparada con la de Android. - Páginas web que contengan flash: Apple no ha dado soporte y aunque el flash se utiliza cada vez menos, en las webs que tengan se nota el defecto. 	<ul style="list-style-type: none"> - Actualizaciones: las actualizaciones para Android representan ser muy tediosas, en muchos modelos tardan meses en llegar, y a veces se quedan en el camino. - Rendimiento: Android necesita más potencia que iOS para ofrecer el mismo rendimiento, lo que supone aumentar los costes. - Seguridad: se cuelan a veces aplicaciones maliciosas en la tienda oficial, y resulta difícil parchear las vulnerabilidades. - Es poco intuitivo, lo que dificulta su manejo. 	<ul style="list-style-type: none"> - Baja posibilidad de personalización: Aunque se dice que la nueva versión de Windows 10 se existe una mejora en este sentido. - Número de aplicaciones: no alcanza al número de aplicaciones que se pueden buscar para Android o iOS, aunque cada día se incorporan nuevas. - Se incorporan tarde al Mercado: Llegan tarde a competir con Android y iOS. - No soporte a flash: Al igual que iOS , Windows Phone no ha dado soporte a flash.

Principales elementos, herramientas configuraciones y perfiles del Android Studio.

Principales Elementos

Actividades

Su implementación es con una subclase de **Activity**. Representa una pantalla con interfaz de usuario, siendo este el componente principal de la interfaz gráfica donde el usuario puede realizar una acción. Cada una de las actividades pese a estar separadas, trabajan juntas para proporcionar una experiencia de usuario. Una aplicación generalmente consiste en múltiples actividades vinculadas de forma flexible entre sí.

Ciclo de Vida de una Actividad.

Se da mediante la implementación de métodos callbacks. El ciclo de vida de una actividad se ve directamente afectado por su asociación con otras actividades, con sus tareas y con la pila de actividades.

Una actividad puede existir en tres estados:

Reanudada	Pausada	Detenida
Es cuando la actividad se encuentra en primer plano de la pantalla y tiene la atención del usuario.	Es cuando otra actividad está visible por encima (total o parcial) de una actividad que antes fue utilizada por el usuario. Una actividad pausada se conserva en la memoria, mantiene toda la información de estado y miembro y continúa anexado al administrador de ventanas. Aunque se sabe que el el sistema puede eliminarla en situaciones en que la memoria sea extremadamente baja.	La actividad está completamente opacada por otra actividad. Una actividad detenida se conserva en memoria, mantiene toda la información de estado y miembro, pero no está anexado al administrador de ventanas. Sin embargo, ya no está visible para el usuario y el sistema puede eliminarla cuando necesite memoria en alguna otra parte.

Servicios

Es un componente de una aplicación que puede realizar operaciones de larga ejecución en segundo plano y que no proporciona una interfaz de usuario. Al igual que una actividad, puede existir otro componente que puede iniciar un servicio y continuará ejecutándose en segundo plano aunque el usuario cambie a otra aplicación. Además, un componente puede enlazarse con un servicio para interactuar con él e incluso realizar una comunicación entre procesos (IPC).

Un servicio puede adoptar dos formas:

Servicio iniciado	Servicio de enlace
<p>Es cuando un componente de aplicación lo inicia llamando a startService(). Una vez iniciado, un servicio puede ejecutarse en segundo plano de manera indefinida, incluso si se destruye el componente que lo inició.</p> <p>Por lo general, un servicio iniciado realiza una sola operación y no devuelve un resultado al emisor.</p> <p>Cuando la operación está terminada, el servicio debe detenerse por sí mismo.</p>	<p>Es cuando un componente de la aplicación se vincula a él llamando a bindService(). Un servicio de enlace ofrece una interfaz cliente-servidor que permite que los componentes interactúen con el servicio, envíe solicitudes, obtengan resultados e incluso lo hagan en distintos procesos con la comunicación entre procesos.</p> <p>Un servicio de enlace se ejecuta solamente mientras otro componente de aplicación está enlazado con él.</p> <p>Se pueden enlazar varios componentes con el servicio a la vez, pero cuando todos ellos se desenlazan, el servicio se destruye.</p>

Proveedores de contenido

Se implementa como una subclase de ContentProvider. Administra un conjunto compartido de datos de la aplicación. Es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. 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.

De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido. Los proveedores de contenido también son útiles para leer y escribir datos privados de tu aplicación y que no se comparten.

Receptores 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. Comúnmente, un receptor de mensajes es simplemente una "puerta de enlace" a otros componentes y está destinado a realizar una cantidad mínima de trabajo.

Un receptor de mensajes se implementa como una subclase de `BroadcastReceiver` y cada receptor de mensajes se proporciona como un objeto `Intent`.

Widget

Son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

Intent

Es el elemento básico de comunicación entre los distintos componentes. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etcétera.

Herramientas

Se incluyen herramientas completas de edición, depuración, pruebas y perfilamiento de códigos. Algunas de ellas son:

Instant Run

Cuando hagas clic en Run o Debug, aplicará los cambios en el código y los recursos en tu aplicación en ejecución. Esta interpreta de manera inteligente los cambios y a menudo los entrega sin reiniciar tu aplicación ni volver a compilar tu APK, para que puedas ver los efectos de inmediato.

Editor de código inteligente

Al ofrecer compleción avanzada de código, refactorización y análisis de código. A medida que escribes, Android Studio proporciona sugerencias en una lista desplegable. Simplemente presiona Tab para insertar el código.

Emulador rápido y cargado de funciones

El Android Emulator se instala e inicia tus apps más rápido que un dispositivo real. También te permite crear prototipos de tu app y probarlos en todas las configuraciones de dispositivos Android: teléfonos, tablets y dispositivos Android Wear y Android TV. También puedes simular varias funciones de hardware, como la localización de GPS, la latencia de red y las funciones multitáctiles.

Configuraciones

Perfiles

La nueva ventana del generador de perfiles de Android de Android Studio 3.0 reemplaza a las herramientas de Android Monitor. Estas nuevas herramientas de generación de perfiles proporcionan datos en tiempo real relacionados con la CPU, la memoria y la actividad de red de tu app.

Se considera que una aplicación tiene un rendimiento deficiente si responde lentamente, muestra animaciones entrecortadas, se congela, se bloquea o consume mucha energía. Para evitar estos problemas de rendimiento, Android Studio nos permite utilizar las herramientas de creación de perfiles para identificar dónde la aplicación está haciendo un uso ineficiente de los recursos, como por ejemplo en la CPU, la memoria, los gráficos, la red y la batería del dispositivo.

Bibliografía

¿iPhone o Android? Ventajas e inconvenientes de cada uno. (2019). Revisado desde <https://www.adslzone.net/2016/01/29/iphone-o-android-ventajas-e-inconvenientes-de-cada-uno/>

<http://goponygo.com/blog/analisis/ventajas-y-desventajas-de-windows-phone/>

<https://winphonemetro.com/2016/10/ventajas-comparativa-windows-mobile-android>

<https://www.xataka.com/moviles/como-es-usar-windows-phone-en-2018>

<https://visual.ly/community/infographic/computers/evolution-tablets-infographic>

<http://www.nicolasnova.net/pasta-and-vinegar/2015/6/28/some-smartphone-evolution-diagrams>

<https://developer.android.com/guide/components/fundamentals?hl=es-419>

<https://developer.android.com/guide/components/activities.html?hl=es-419>

<https://developer.android.com/guide/topics/providers/content-providers.html?hl=es-419>

<https://developer.android.com/guide/topics/providers/content-provider-basics.html?hl=es-419>

<https://developer.android.com/reference/android/content/BroadcastReceiver.html?hl=es-419>

<https://developer.android.com/studio/features?hl=es>

<https://es.slideshare.net/JaquelineLuna3/herramientas-de-desarrollo-de-android-72794458>

<https://okhosting.com/blog/android-studio-herramienta-crear-aplicaciones/>

<https://desarrollador-android.com/category/desarrollo/herramientas/>

<https://developer.android.com/studio/profile/android-profiler>

<https://developer.android.com/studio/releases/sdk-tools>