Unidad 1. Introducción a la Programación de Dispositivos Móviles

OBJETIVO

Conocer los orígenes y las bases para el desarrollo de aplicaciones para dispositivos móviles, como también diferentes arquitecturas y el ciclo de vida de las aplicaciones de diferentes sistemas operativos.

CONTENIDOS

1. Sistemas operativos para dispositivos móviles, historia y evolución.
2. Arquitecturas de aplicaciones para dispositivos móviles.
3. Entorno de desarrollo para la creación de aplicaciones para dispositivos móviles.
4. Componentes de una aplicación para dispositivos móviles.
5. Ciclo de vida de las aplicaciones para para dispositivos móviles.

INTRODUCCIÓN

1. Sistemas operativos para dispostivos móviles.

Bienvenido. En esta ocasión, hablaremos sobre la introducción a las tecnologías móviles y sus antecedentes. La funcionalidad que ofrecen actualmente los dispositivos móviles supera exponencialmente a la que tenían en el pasado. En general, ahora no solo tenemos acceso a telefonía convencional, sino que nuestros dispositivos, como teléfonos, tabletas, gafas, relojes y computadoras en vehículos, entre otros, nos proporcionan acceso a datos a través de internet, cámaras, GPS y aplicaciones que nos mantienen en constante interacción y comunicación con el mundo.

El primer antecedente operativo y funcional conocido como servicio de telefonía móvil se remonta a 1947, cuando la empresa AT&T implementó una red de transmisores de baja potencia. Cada uno de estos transmisores brindaba servicio a un área específica, denominada "célula", de donde proviene el término de "telefonía celular" para referirse a la telefonía móvil. A finales de los años 50 del siglo XX, el científico soviético Leonid Ivanovic Kupriyanovich desarrolló un sistema de comunicación móvil que culminó en el modelo K L 1. Este sistema utilizaba ondas de radio y tenía la capacidad de alcanzar una distancia de 30 kilómetros, brindando servicio a varios clientes. Este modelo sentó las bases para la investigación que Kupriyanovich llevó a cabo al año siguiente en el Instituto de Investigación Científica

Voronesh. De esta investigación surgió el Altai, que se distribuyó comercialmente en 1963 y llegó a estar presente en más de 114 ciudades de la Unión Soviética, brindando servicio a hospitales y médicos.

Sistemas operativos para dispositivos móviles: Los sistemas operativos móviles han experimentado avances significativos en los últimos 23 años. Desde pantallas básicas que solo presentaban datos sin interacción ni conexión a internet, hemos evolucionado hacia dispositivos con capacidades muy altas y funciones similares a las de una computadora de escritorio. Estos cambios han sido exponenciales. A continuación, se enumeran por orden de aparición diferentes sistemas operativos utilizados en dispositivos móviles.

En 1981, Symbian fue un sistema operativo propiedad de Nokia, que en el pasado resultó de la colaboración entre varias empresas de telefonía móvil, incluyendo a Nokia, Sony Mobile Communications, Samsung, Arima, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, entre otras. Sus orígenes se remontan a su predecesor, Época 32, utilizado en dispositivos portátiles y handheld. Symbian estuvo vigente entre 1997 y 2013, siendo durante algunos años el sistema operativo estándar para smartphones, ya que más del 85% de los fabricantes de estos dispositivos tenían licencia para usarlo. Symbian estaba diseñado para satisfacer los

requisitos específicos de los teléfonos móviles 2G y 3G.

1996 palm Os fue un sistema operativo móvil desarrollado inicialmente por tal cual o ese fue diseñado para la facilidad de uso con una interfaz gráfica de usuario basada en pantallas táctiles el sistema proporciona un conjunto de aplicaciones con retraso para gestión de información personal versiones más recientes del sistema operativo han extendido su soporte a smartphones

En 1999, BlackBerry OS fue un sistema operativo móvil de código cerrado desarrollado por BlackBerry. Este sistema permitía la multitarea y ofrecía soporte para diferentes métodos de entrada. Su desarrollo se remonta a la introducción de los primeros dispositivos BlackBerry en 1999. Estos dispositivos permitían el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes, además de realizar las funciones habituales de un teléfono móvil.

BlackBerry OS fue descontinuado después del lanzamiento de BlackBerry 10 en enero de 2013, continuando el soporte para las versiones más antiguas hasta finales de 2013.

En el año 2000, Windows Mobile fue un sistema operativo móvil compacto desarrollado por Microsoft y diseñado para su uso en teléfonos inteligentes y otros dispositivos móviles. Está basado en el núcleo del sistema operativo Windows y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Fue diseñado para ser similar a las versiones de escritorio de Windows en términos estéticos. Inicialmente, apareció bajo el nombre de "Pocket PC" como una rama de desarrollo de Windows para dispositivos móviles con capacidades limitadas.

En la actualidad, la mayoría de los teléfonos con Windows Mobile cuentan con un estilo digital que se utiliza para introducir comandos mediante la pulsación en la pantalla.

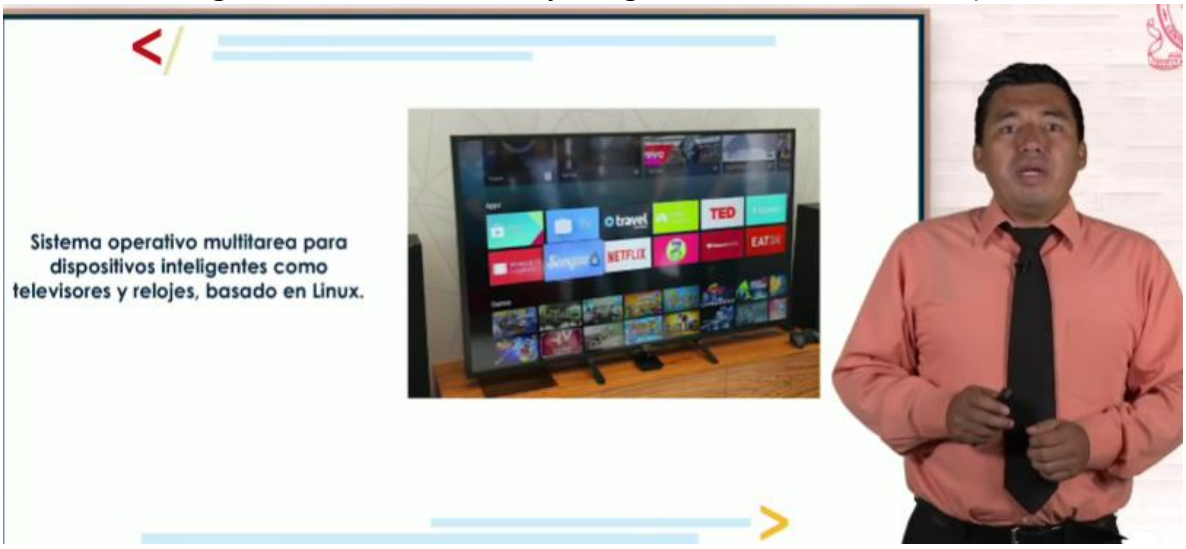
En 2007, iOS se estableció como un sistema operativo móvil desarrollado por la multinacional Apple, originalmente diseñado para el iPhone y posteriormente utilizado en dispositivos como el iPod Touch y el iPad. iOS no permite la instalación de terceros en hardware de otros fabricantes. La última versión del sistema operativo es iOS 13, lanzada en 2019.

Los elementos de control en iOS incluyen deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y proporciona una interfaz fluida. La interacción con el sistema operativo implica gestos como deslizamientos, toques y pellizcos, cada uno con definiciones diferentes dependiendo del contexto de la interfaz. iOS se deriva de macOS, que a su vez está basado en Darwin OS, siendo este último un sistema operativo de tipo Unix.

En 2008, Android se estableció como un sistema operativo móvil desarrollado por Google. Este sistema operativo es basado en el kernel de Linux y es de código abierto. Android fue diseñado para dispositivos móviles con pantallas táctiles, tales como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores. El código fuente principal de Android se conoce como Android Open Source Project (AOSP) y se licencia principalmente bajo la Licencia Apache.

En 2009, WebOS, también conocido como LG OS, Open WebOS y HP WebOS, fue lanzado como un sistema operativo multitarea para dispositivos inteligentes como

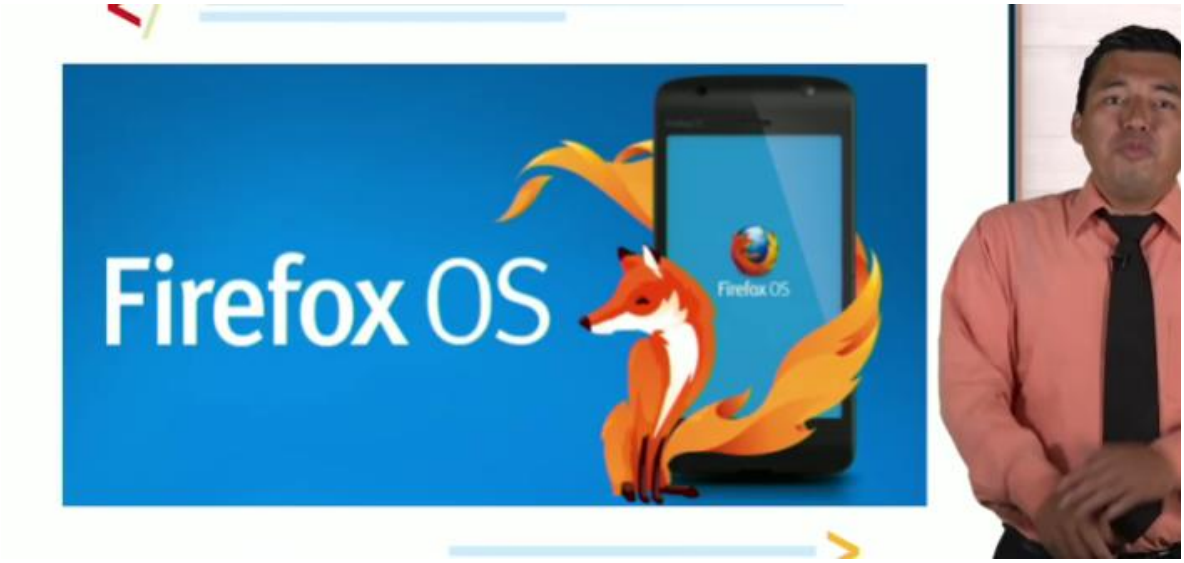
televisores y relojes, basado en Linux. Originalmente desarrollado por Palm como un sistema operativo para móviles y tabletas, la empresa fue posteriormente adquirida por Hewlett Packard (HP). Sin embargo, tras un cambio de estrategia, HP decidió liberar el código fuente de WebOS y luego vendió el sistema operativo a LG.



En 2010, Bada fue un sistema operativo para teléfonos móviles desarrollado por Samsung. Actualmente, ha sido reemplazado por Tizen. Fue diseñado para cubrir tanto los teléfonos inteligentes de gama alta como los de gama baja. Se basa en el sistema operativo propiedad de Samsung, SHP, y es utilizado en muchos de esos teléfonos, como el Samsung Wave.

En 2010, Firefox OS fue un sistema operativo basado en HTML5 con un núcleo Linux de código abierto para varias plataformas. Fue desarrollado por Mozilla Corporation con el respaldo de otras empresas y una gran comunidad de voluntarios de todo el mundo. Este sistema operativo, diseñado para permitir que las aplicaciones HTML5 se comuniquen directamente con el hardware del dispositivo mediante JavaScript u Open Web APIs, se enfocó inicialmente en teléfonos móviles, smartphones y tabletas, especialmente en el sector de gama baja.

A finales de 2015, Mozilla Corporation anunció la conclusión del desarrollo de Firefox OS para dispositivos móviles.



En 2017, AOSP (Kai) se convierte en un sistema operativo para dispositivos móviles basado en Linux, derivado de B2G, el sucesor de código abierto de la comunidad de Firefox OS, que fue interrumpido por Mozilla en 2016.

Las principales características de KaiOS incluyen soporte para 4G LTE, GPS y Wi-Fi con aplicaciones basadas en HTML5, mayor duración de la batería en dispositivos táctiles, interfaz de usuario optimizada, menor consumo de memoria y energía, actualizaciones inalámbricas, ligereza y la capacidad de ejecutarse en dispositivos con tan solo 256 MB de memoria.

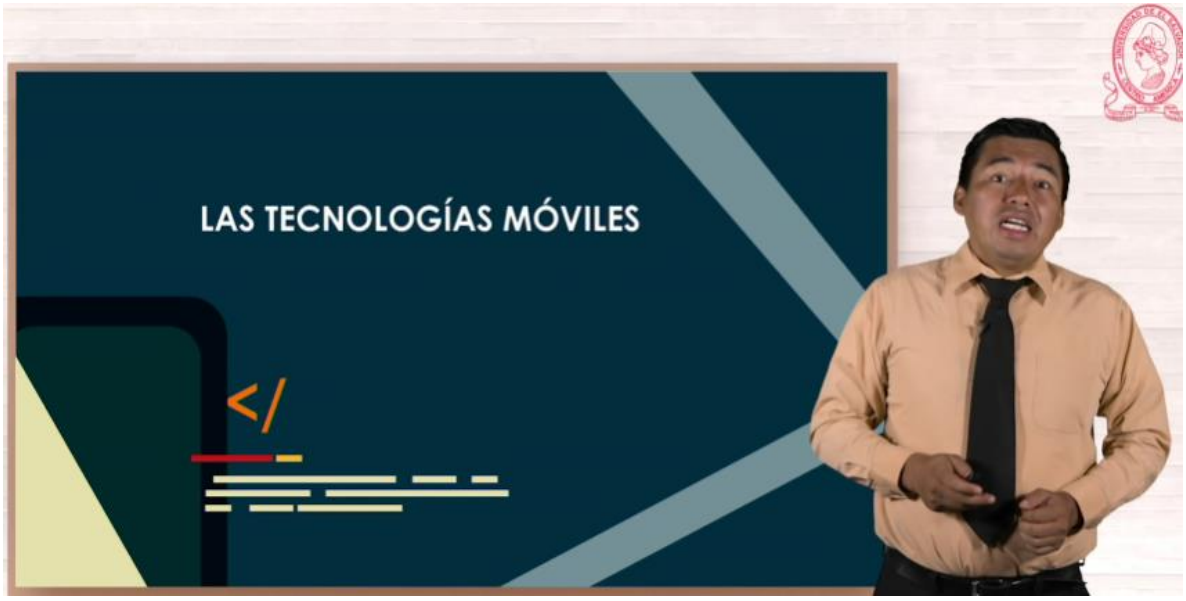
HISTORIA Y EVOLUCIÓN

2. Generaciones de la Tecnologías Móviles

Bienvenidos. Hoy hablaremos sobre las tecnologías para móviles. Las tecnologías móviles han estado con nosotros durante mucho tiempo, simplificando nuestras actividades cotidianas y facilitando nuestras labores, estudios o vida normal, gracias a sus innumerables aplicaciones disponibles que han ido incrementándose con el tiempo.

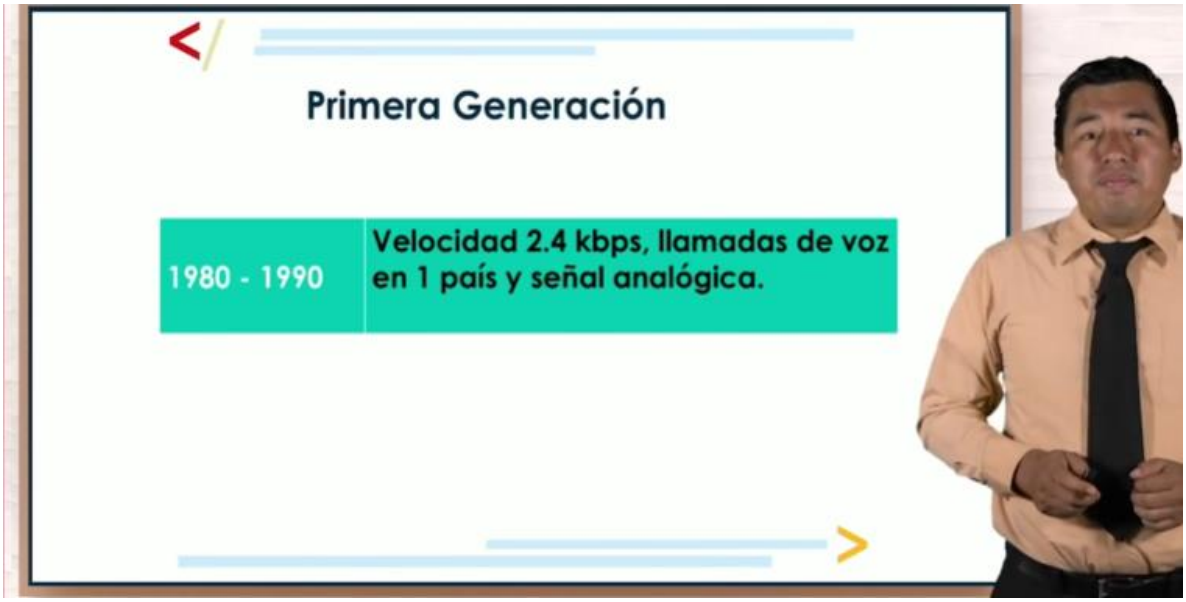
A lo largo de los años, la tecnología móvil se ha vuelto más popular, representando un mayor apoyo en las actividades diarias. En la actualidad, se está implementando como estándar la quinta generación del servicio de tecnología de transmisión, compatible con nuevas bandas de frecuencia.

Desde los inicios de la telefonía móvil comercial, las comunicaciones se han innovado constantemente, resultando en una mayor adopción desde edades tempranas.



Veamos las generaciones de la tecnología móvil:

Primera generación: Iniciada en la década de 1980 hasta 1990, los primeros teléfonos móviles comerciales poseían funcionalidades básicas de servicio de voz, utilizando el sistema avanzado de telefonía móvil (AMPS). Las características incluyen una velocidad de 2 puntos KB por segundo, llamadas de voz en un país, señal analógica, baja duración de la batería y tamaño grande del teléfono.

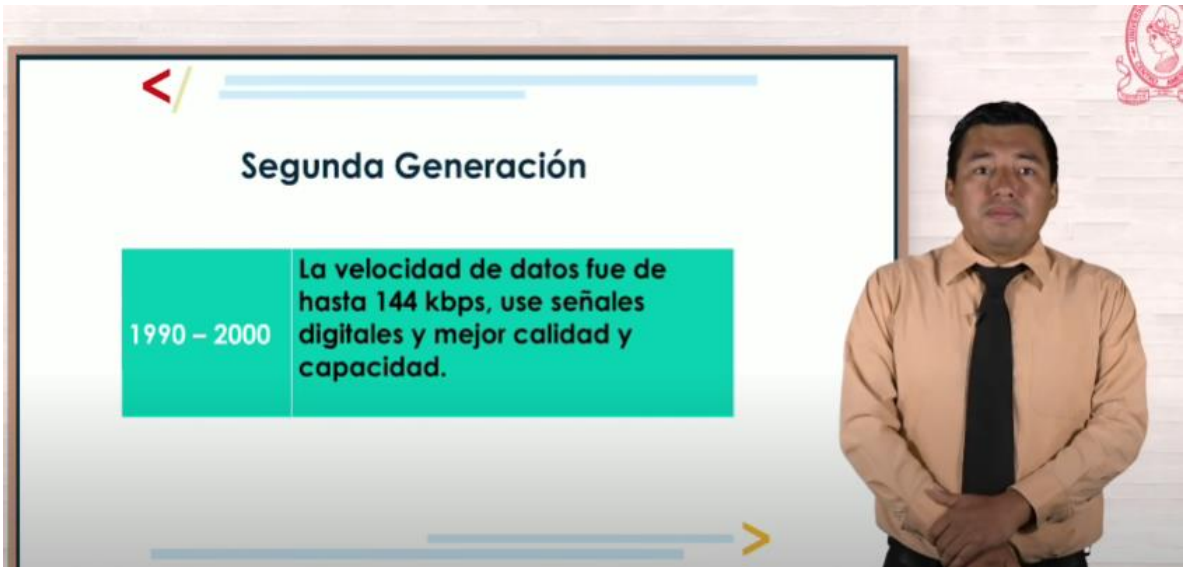


Segunda generación: Iniciada en 1990, introduce el internet y se basa en el Sistema Global para las Comunicaciones Móviles (GSM). Utiliza la señal digital para transmitir voz, con una velocidad de transmisión de 30 a 200 kilobits por segundo. Características principales incluyen la digitalización de las comunicaciones,

actualización a 2.5G, transmisión de datos mejorada de 64 hasta 144 KB por segundo utilizando GPRS, CDMA y LG.

Características

Posee uso de señales digitales, habilita servicios como la mensajería de texto, mensajes con imágenes y mensajes multimedia, proporciona mejor calidad y capacidad, no se pueden manejar datos complejos como videos.



Tercera generación: Iniciada en el año 2000, se basa en el Sistema Universal de Telecomunicaciones Móviles (UMTS). Busca ofrecer mayor velocidad de transmisión, llegando hasta 15 megabytes por segundo mediante la conmutación de paquetes. Características principales son una mayor velocidad, compatibilidad con teléfonos inteligentes, aumento del ancho de banda, acceso a servicios como televisión y vídeos, y operación en un rango de 2100 megahercios, con un ancho de bando de entre 12 a 20 mgHrz.

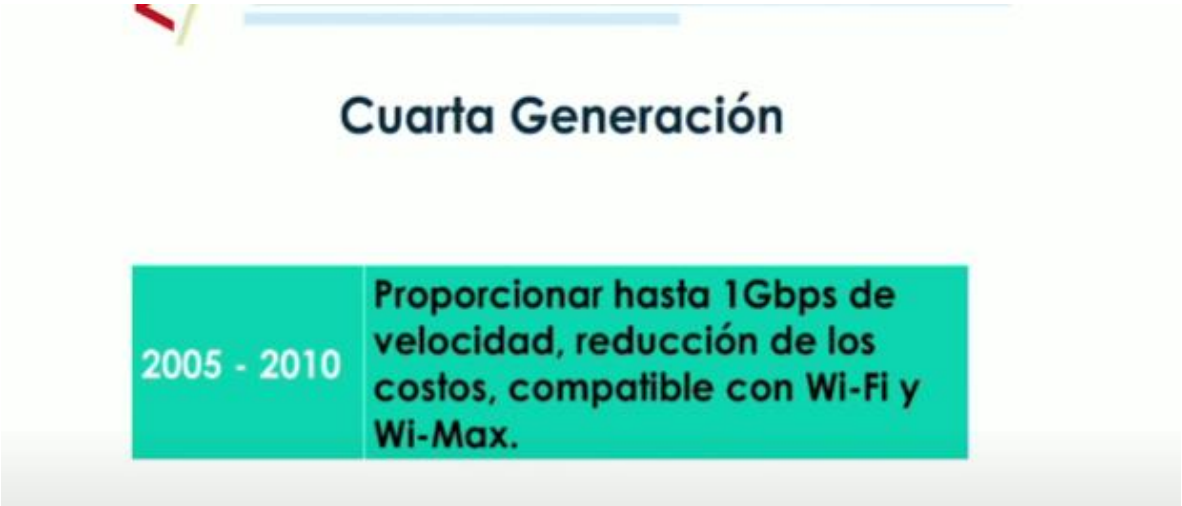
Características

Velocidad de 2mg por segundo, compatibles con los teléfonos inteligentes, aumento del ancho de bando y las tasas de transferencia de datos para acomodar aplicaciones y audio basados en la web y archivos de video, enviar y recibe mensajes de correo electrónico grandes, web de alta velocidad, más seguridad, video conferencia, juegos en 3D, transmisión de tv, tv móvil y llamadas telefónicas. Tasas telefónicas por servicios 3G.

Cuarta generación: Aumenta la velocidad de descarga a 100 megabytes por segundo, desarrollando el estándar LTE. Características incluyen acceso inalámbrico de ancho de banda, videochat, TV móvil, HDTV, servicios de voz y datos, y otros servicios que utilizan ancho de banda.

Características de 4G

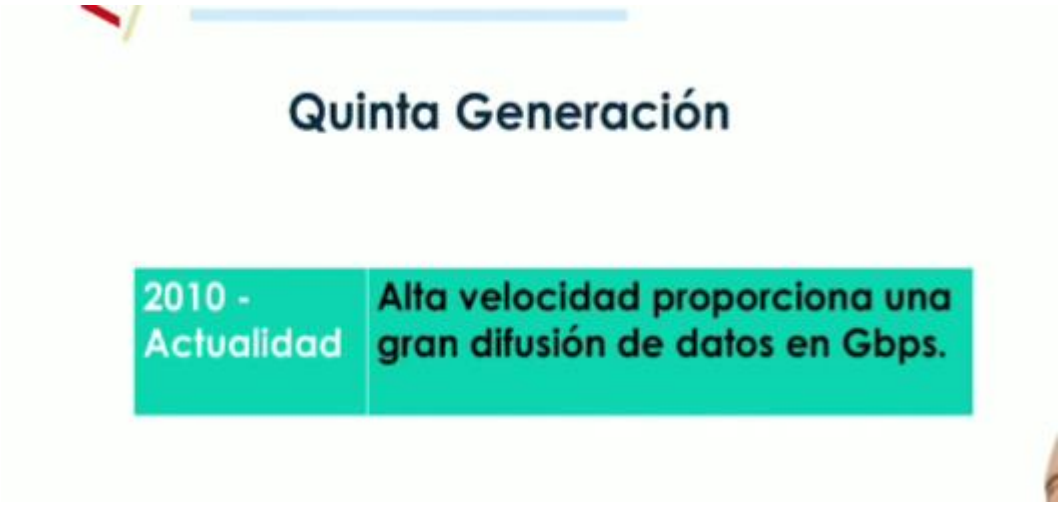
Proporcionar 100 mg por segundo a 1gb por segundo de velocidad, combinación de wi-fi y Wi-Max alta seguridad, bajo costo por bit



Quinta generación: Iniciada a fines de 2010, posee mejores niveles de conectividad y cobertura, basada en la Wireless World Wide Web (WWWW) y comunicación inalámbrica completa sin limitaciones. Características principales incluyen alta velocidad, alta capacidad, gran difusión de datos en gigabyte por segundo, mayor capacidad de memoria del teléfono, velocidad de marcación, claridad en audio y vídeo, y soporte para multimedia interactiva, voz, transmisión de vídeo e internet.

ARQUITECTURA

3. Arquitectura de aplicaciones para Dispositivos Móviles



ARQUITECTURA

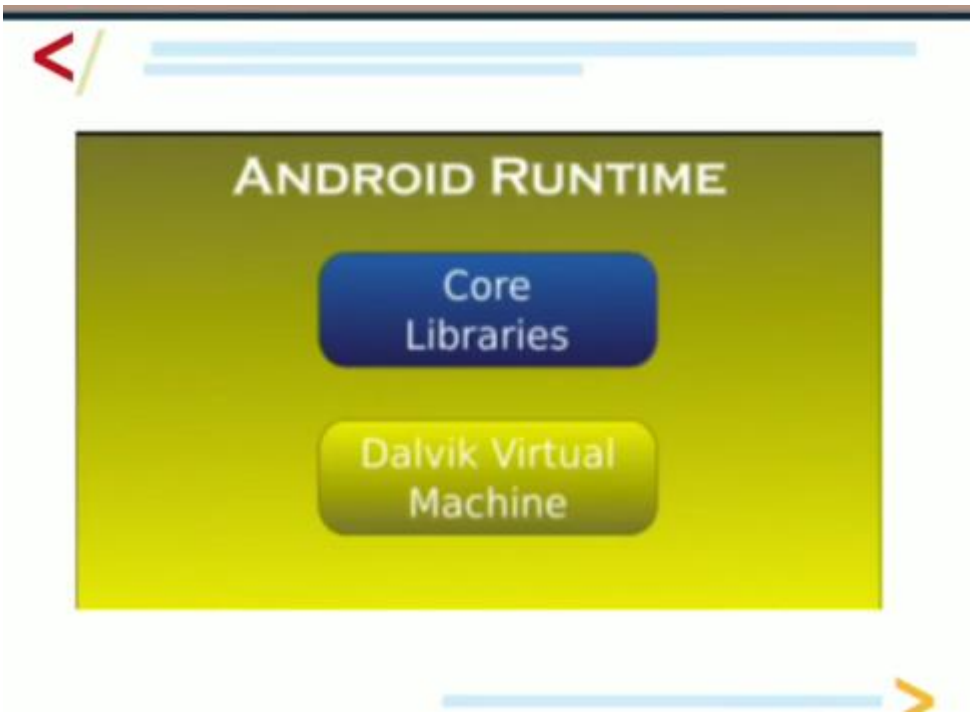
3. Arquitectura de aplicaciones para Dispositivos Móviles

Bienvenido a esta video clase. En este video, vamos a ver la arquitectura de aplicaciones para dispositivos móviles, específicamente la arquitectura de aplicaciones para dispositivos Android. Android es una pila de software de código abierto basada en Linux, creada para una amplia variedad de dispositivos y factores de forma.

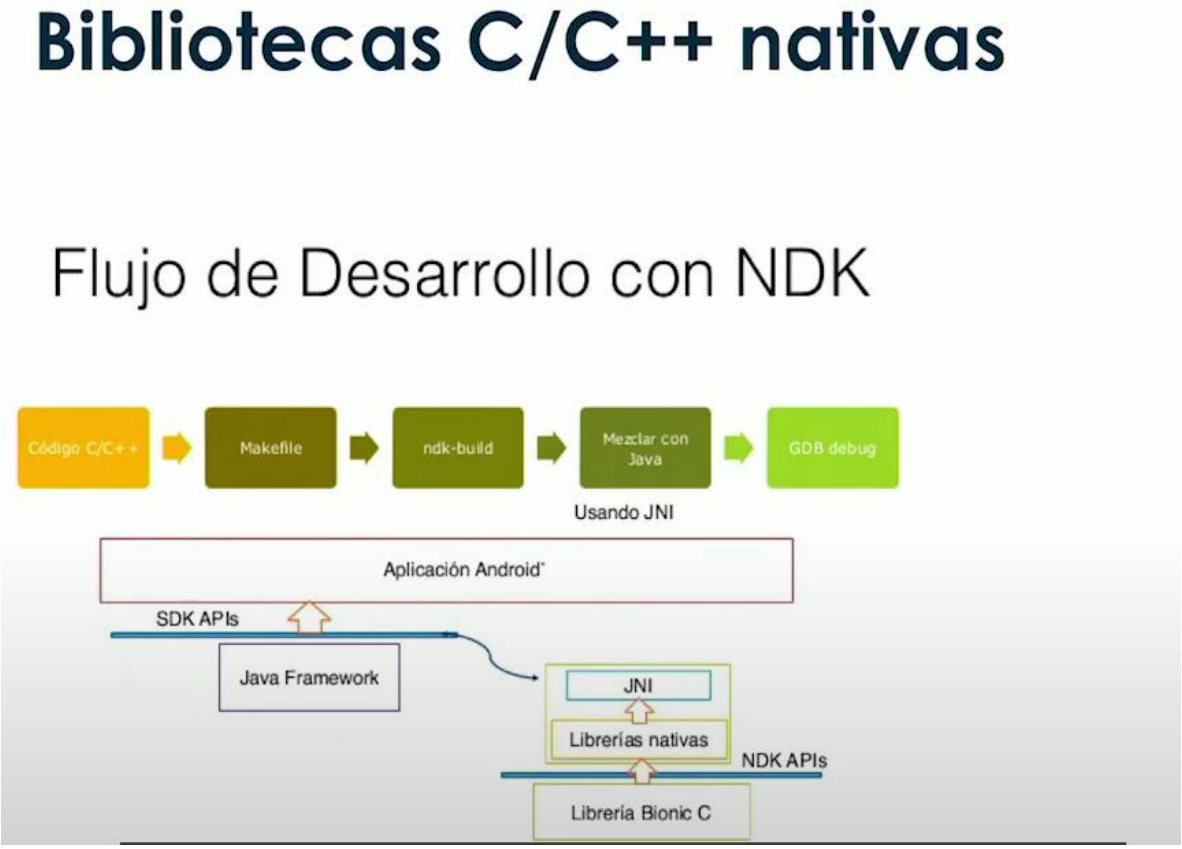
En la diapositiva se muestran los componentes principales de la plataforma Android. La base de la plataforma Android es el kernel de Linux. El uso del kernel de Linux permite que Android aproveche funciones de seguridad clave y al mismo tiempo permite a los fabricantes de dispositivos desarrollar controladores para un kernel conocido.

La capa de extracción de hardware brinda interfaces estándares que exponen las capacidades del dispositivo al API de Java de nivel más alto. Este API consiste en varios módulos de bibliotecas, cada uno implementando una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de Bluetooth.

Android 5.0 o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android. ART (Android Runtime) está diseñado para ejecutar varias máquinas virtuales en dispositivos de memoria baja, ejecutando archivos de texto en un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo, crea cadenas de herramienta con Jacks y compila fuentes de java en código de bytes decks que se puede ejecutar en la plataforma Android, además se incluye un conjunto de bibliotecas de entorno de ejecución centrales que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java, se incluyen algunas funciones del lenguaje java 8 que usan el framework trabajo de Java



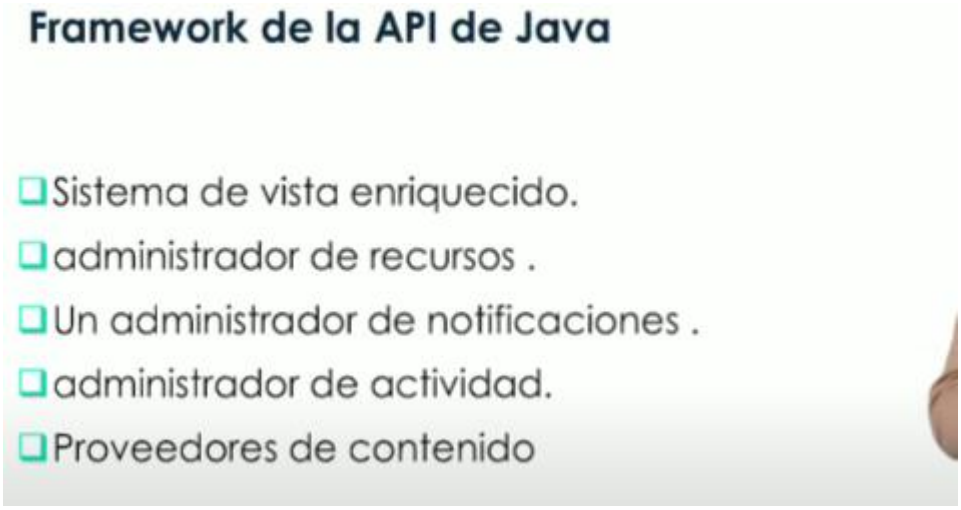
Bibliotecas C / C++ nativas, muchos componentes y servicios centrales del sistema android como el ART y la HALD se basan en código nativo que requieren bibliotecas nativas escritas en C y en C++. Bibliotecas nativas a las apps. Por ejemplo, puedes acceder a OpenGL ES a través de la API de OpenGL ES de Java del framework de trabajo de Android para agregar a tu app compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D. Si desarrollas una app que requiere C o C++, puedes usar el NDK de Android para acceder a algunas de estas bibliotecas de plataformas nativas directamente desde tu código nativo.



El framework de lápiz deja todo el conjunto de funciones del sistema operativo Android disponible mediante APIs y escritas en lenguaje Java. Estas APIs son los cimientos que necesitas para crear apps de Android, simplificando la reutilización de componentes del sistema y servicios centrales y modulares que son un sistema de vista enriquecido y extensible que puedes usar para compilar la interfaz de una app.

Se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable. También hay un administrador de recursos que te brinda acceso a recursos sin código, como cadenas localizadas, gráficos y archivos de diseño. Un administrador de notificaciones permite que todas las apps muestren alertas personalizadas en la barra de estado. Un administrador de actividad administra el ciclo de vida de las apps y proporciona una pila de retrocesos de navegación común. Los proveedores de contenido permiten que las apps accedan a datos desde otras apps, como la app de contactos, o compartan sus propios datos.

Los desarrolladores tienen acceso total a las mismas APIs del framework de trabajo que usan las apps del sistema Android.



En Android, las apps del sistema incluyen un conjunto de aplicaciones centrales para correo electrónico, mensajería, SMS, calendarios, navegación en internet y contacto, entre otros elementos. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elige instalar. Por ello, una app externa se puede convertir en el navegador web.

Ahora, pasemos a la arquitectura de aplicaciones para dispositivos iOS. La arquitectura de iOS es en capas. En el nivel superior, iOS funciona como un intermediario entre el hardware subyacente y las aplicaciones. Las aplicaciones no se

comunican directamente con el hardware subyacente. En su lugar, hablan con el hardware a través de una colección de interfaces de sistemas bien definidas.

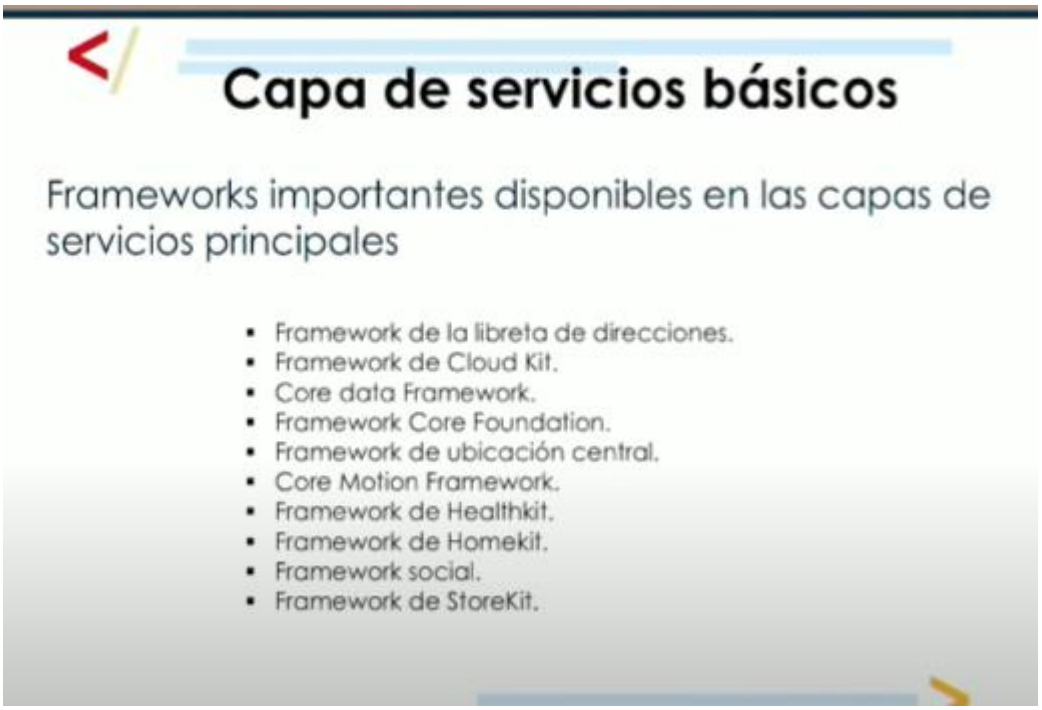
Estas interfaces facilitan la escritura de aplicaciones que funcionan constantemente en dispositivos que tienen varias capacidades de hardware. Las capas inferiores brindan los servicios básicos en los que se basa toda la aplicación, y la capa del nivel superior brinda gráficos sofisticados y servicios relacionados con la interfaz. Apple proporciona la mayoría de sus interfaces de sistemas en paquetes especiales llamados frameworks. Un framework es un directorio que contiene una biblioteca compartida dinámica, archivos y recursos relacionados, como archivos encabezados e imágenes, necesarios para admitir esa biblioteca.

Cada capa tiene un conjunto de frameworks que el desarrollador utiliza para construir las aplicaciones.

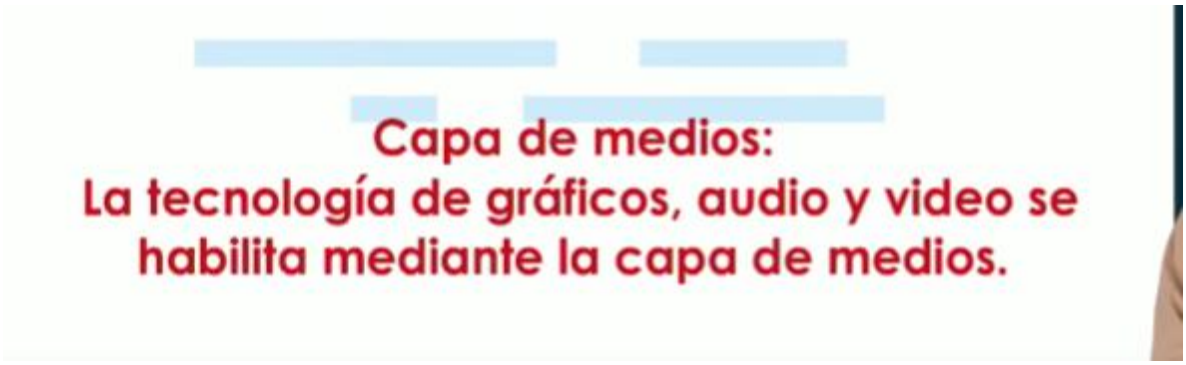
La capa principal del sistema operativo, la capa "Core OS," posee características de bajo nivel sobre las que se basan la mayoría de las otras tecnologías. Algunas son pregón central de Bluetooth, framework de acelerador, freno de accesorio externos, framework de servicios de seguridad, framework de autenticación local. El soporte de 64 bits de iOS admite el desarrollo de aplicaciones de 64 bits y permite que la aplicación se ejecute más rápido.

En la capa de servicios básicos, algunos de los frameworks importantes disponibles son el framework de la libreta de direcciones, que proporciona acceso mediante programación a una base de datos de contactos del usuario; el framework de iCloud, que proporciona un medio para mover datos entre su aplicación e iCloud; Core Data framework, tecnología para administrar el modelo de datos de una aplicación; y el framework Model-View-Controller (MVC) que ofrece interfaces que cubren muchas de las características encontradas en el framework Foundation.

Otros frameworks incluyen el framework de ubicación central, que proporciona información de ubicación y rumbo a las aplicaciones; Chrome 8 framework, que brinda acceso a todos los dispositivos de datos basados en movimiento disponibles en un dispositivo; framework Foundation, que cubre muchas características encontradas en el framework Foundation; Freiburn de Gel kit, un nuevo framework para manejar la información del usuario relacionada con la salud; y el framework de Jong Kuyt, un nuevo framework para hablar y controlar dispositivos conectados en el hogar de un usuario premium.



La capa de medios gráficos, audio y vídeo



incluye frameworks como el framework de gráficos, que describe el soporte de alto nivel para diseñar imágenes y también se utiliza para animar el contenido de las vistas. El framework Core Graphics es el motor de dibujo nativo para aplicaciones iOS y brinda soporte para renderizado 2D basado en imágenes y vectores. Core Animation es una tecnología inicial que optimiza la experiencia de animación de las aplicaciones. Core Imagine, se brinda soporte avanzado para controlar videos e imágenes inmóviles de una manera no destructiva, OpenGI ES y GLKit gestiona la representación avanzada en 2D y 3D mediante interfaces aceleradas hardware metal permite un rendimiento muy alto para sus sofisticados trabajos de procesamientos y cálculos gráficos

Framework de audio regular del reproductor multimedia, es un framework de alto nivel que proporciona un uso simple a la biblioteca de itunes de un usuario y soporte para el pluralista de reproducciones, AV Foundation es una interfaz para manejar la grabación y reproducción de audio y video, OpenAL es una tecnología estándar de la industria para proporcionar audio.

Capa de toque de cacao

Framework Eventkit: proporciona controladores de visualización para las interfaces estándar del sistema para ver y modificar eventos relacionados con el calendario.

Gamekit Framework: Implementar soporte para game center que permite al usuario compartir su información relacionadas con el juego en línea

iAd Framework: Le permite entregar anuncios basados en el banner desde su aplicación

MapKit Framework: Proporciona un mapa desplazable que se debe incluir en la interfaz de usuario

PushKitFramework: proporciona soporte de registro para aplicaciones voip

Framework de Twitter: Admite una interfaz de usuario para generar tweets y para crear una url para acceder a los servicios de Twitter

Franmework UIKit: Proporciona una infraestructura vital para aplicaciones gráficas basadas en eventos en iOS algunas de las funciones importantes del framework de uikits

COMPONENTES

4. Componentes de una aplicación para dispositivos móviles

¡Bienvenido! Hoy vamos a ver los entornos de desarrollo para dispositivos móviles. Los entornos y los lenguajes varían según el sistema operativo. Para programar aplicaciones en el sistema operativo Android, se debe conocer el lenguaje Java o Kotlin, mientras que para programar aplicaciones en el sistema operativo iOS, se debe conocer el lenguaje Swift y Objective-C. Ambos tienen similitudes ya que pueden programar con un lenguaje de programación de propósito general como Java y Objective-C, y con uno específico como Kotlin y Switf.

El entorno de desarrollo para Android.

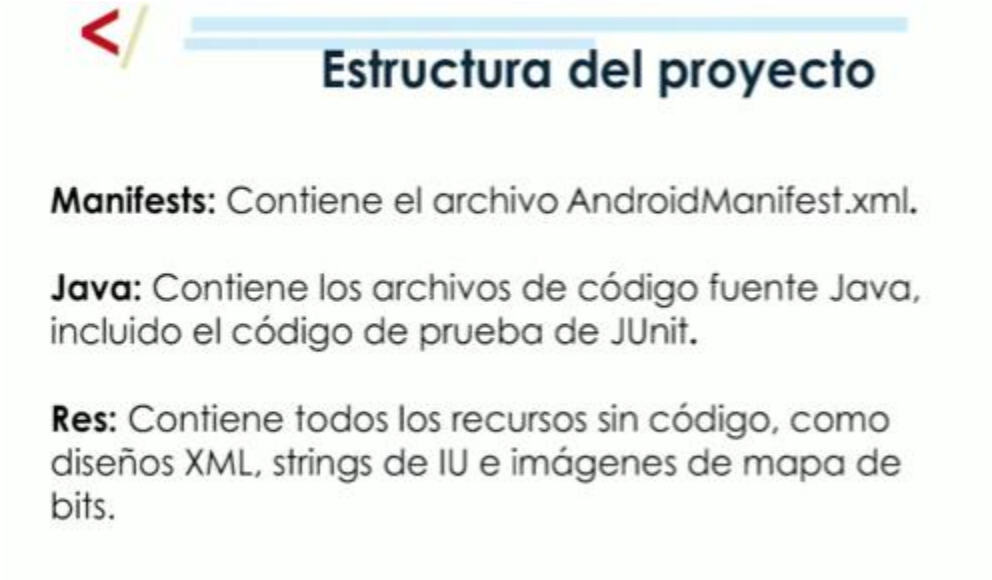
Entorno de desarrollo para Android

Android Studio, es el entorno de desarrollo integrado oficial basado en IntelliJ IDEA. Posee características como un sistema de compilación flexible basado en Gradle, un emulador rápido y cargado de funciones, un entorno unificado para desarrollar en todos los dispositivos Android, aplicación de cambios para insertar cambios de códigos y recursos sin reiniciar la aplicación, integración con Git y plantillas de código, variedad de marcos de trabajo y herramientas de prueba, entre otras, usabilidad y compatibilidad de la versión, compatibilidad con C++ y NDK, compatibilidad integrada para Google Chrome, classroom familiarizando la integración de google cloud, message y app engine.

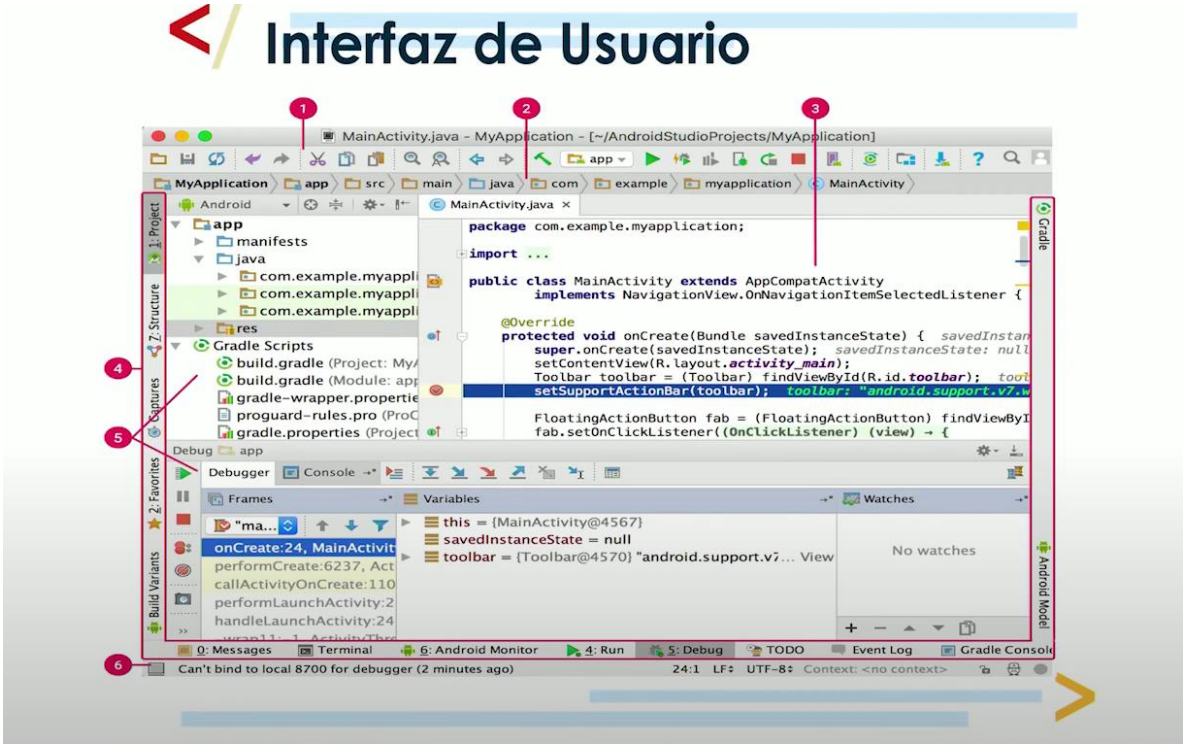
La Estructura del proyecto en Android Studio incluye módulos de aplicaciones, bibliotecas, y para Google App Engine. Cada módulo contiene carpetas como 'manifest' para el archivo AndroidManifest.xml, 'java' para los archivos de código fuente, y 'res' para recursos como diseños XML, strings, e imágenes de mapa de bit.

La ventana principal de Android Studio tiene una barra de herramientas, barra de navegación, ventana del editor y una barra de herramientas fuera del IDE. La barra de estado muestra el estado del proyecto y advertencias.

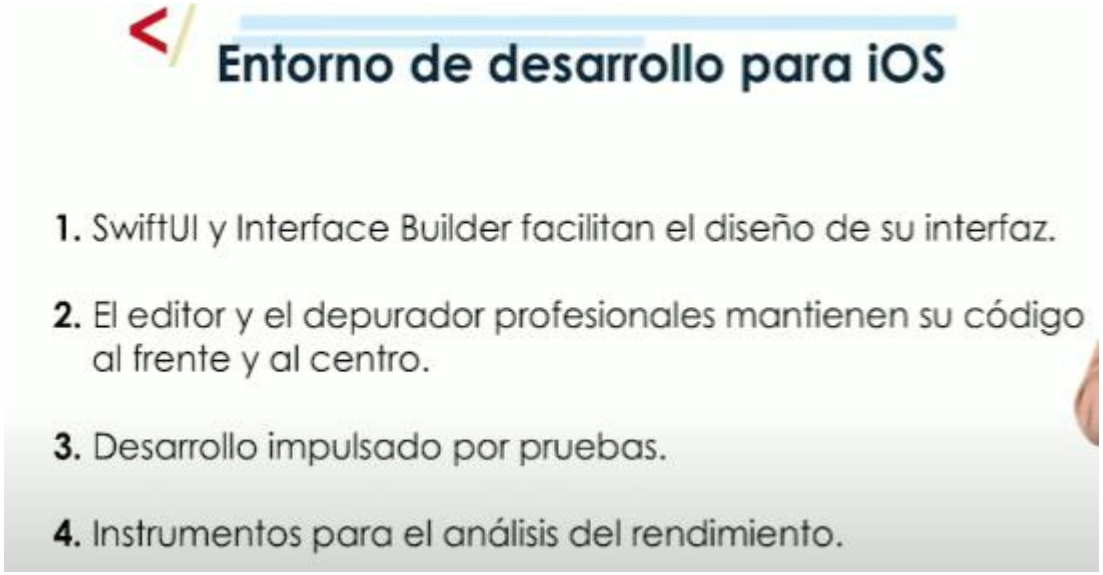
El entorno de desarrollo para iOS, Xcode, permite desarrollar aplicaciones para iPhone, iPad, Apple TV, y Apple Watch. Incluye compiladores Swift y Objective-C, herramientas de análisis, instrumentos, simuladores y el último SDK. Facilita el diseño de interfaces, mantiene el código al frente y centro, y ofrece desarrollo impulsado por pruebas.



La interfaz de usuario de Xcode tiene una barra de herramientas, vista de actividades y esquemas del menú, panel de botones de la ventana, área de navegación, área de edición, área del inspector, y área de depuración. Estas áreas proporcionan acceso rápido a diferentes partes del proyecto y permiten editar código, ver información y depurar.

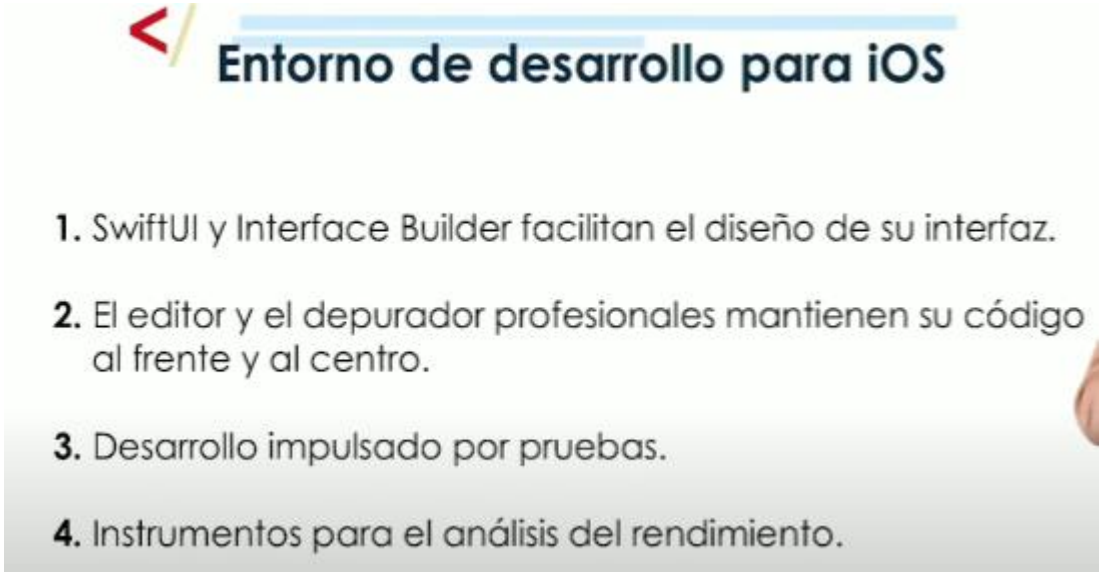


1. La barra de herramientas te permite realizar una gran variedad de acciones como ejecutar tu app e iniciar las herramientas de android
2. La barra de navegación te ayuda a explorar tu proyecto y abrir archivos para editar proporciona una vista más compacta de la estructura visible de la ventana project
3. La ventana del editor es el área en la que puedes crear y modificar código según el tipo de actividad actual el editor puede cambiar por ejemplo cuando ves un archivo de diseño el editor muestra el editor de diseño
4. La barra de la ventana de herramientas se encuentra afuera de la ventana del ide y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales
5. Las ventanas de herramientas te brindan acceso a tareas específicas como la administración de proyectos la búsqueda, el control de versiones entre otras pueden expandir las y contraer las
6. En la barra de estado se muestra el estado de tu proyecto y el ide además de advertencias o mensajes



El entorno de desarrollo para iOS, Xcode, es esencial para la creación de aplicaciones destinadas a iPhone, iPad, Apple TV y Apple Watch. Xcode ofrece a los desarrolladores un flujo de trabajo integrado que abarca desde el diseño hasta la codificación, la prueba y la depuración de la interfaz de usuario.

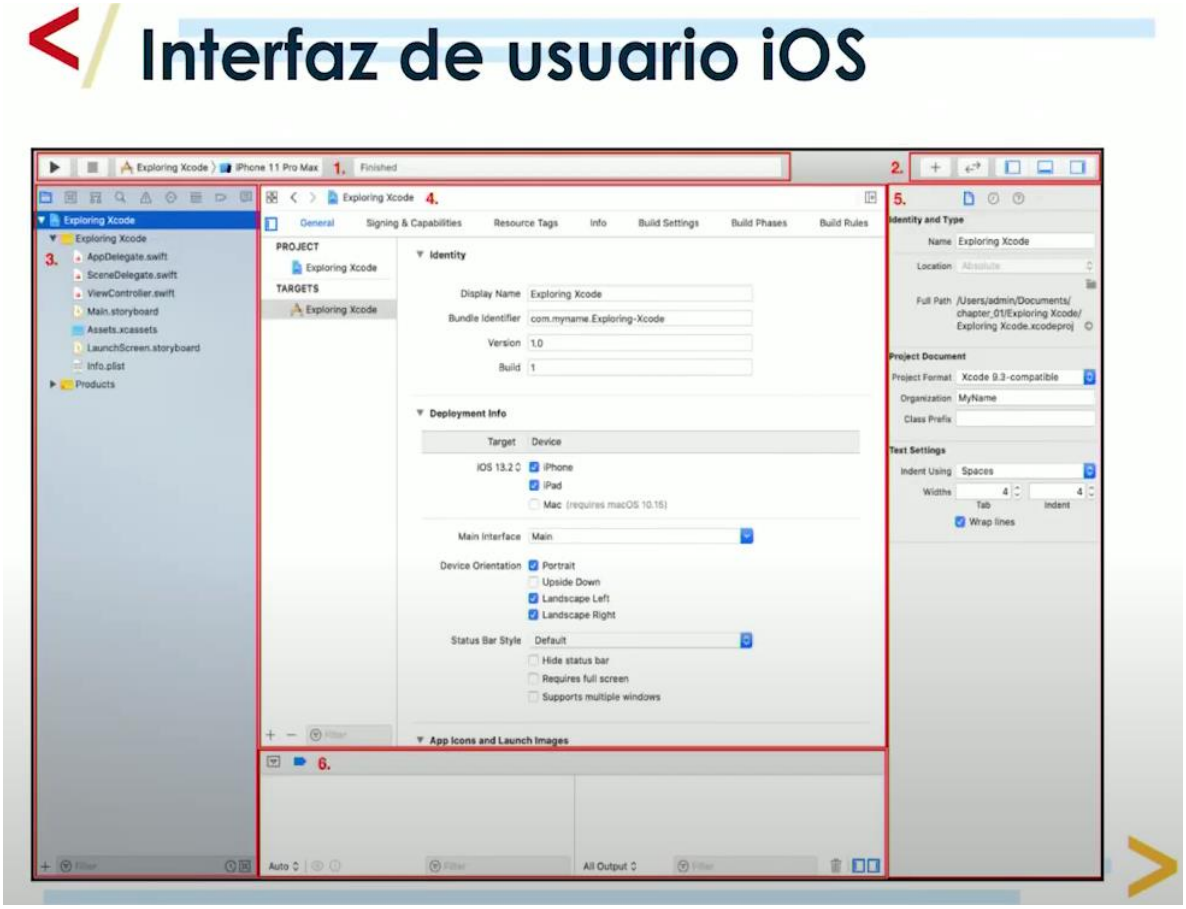
Xcode excluye los compiladores Xcode y de la suite C, C++ o Jetix, y en su lugar, proporciona herramientas de análisis, instrumentos y los últimos SDK. Una característica destacada de Xcode es su suite UI (Interfaz de Usuario), que simplifica el diseño de la interfaz, y cuenta con un editor y depurador profesionales que mantienen el código en el centro de la atención. Además, Xcode respalda el desarrollo impulsado por pruebas y ofrece instrumentos para el análisis del rendimiento.



La interfaz de usuario cuenta con los siguientes elementos: una barra de herramientas utilizada para construir y ejecutar aplicaciones, mostrando el proceso de la vista y las tareas en ejecución. Incluye botones de reproducción y detención, así como la vista de actividades y esquemas del menú.

El panel de botones de la ventana se emplea para configurar el entorno de trabajo e incluye la librería de objetos, la versión del editor, el navegador, el inspector y la depuración. Además, el área de navegación facilita un acceso rápido a diversas partes del proyecto, mientras que el área de edición permite la modificación del código fuente, la interfaz de usuario y otros recursos.

La zona del inspector posibilita ver, seleccionar y editar información de los elementos elegidos en el área de navegación o en el área de edición. En cuanto al área de debut o depuración, presenta un espacio dinámico para la depuración, mostrando variables de la vista y la consola.

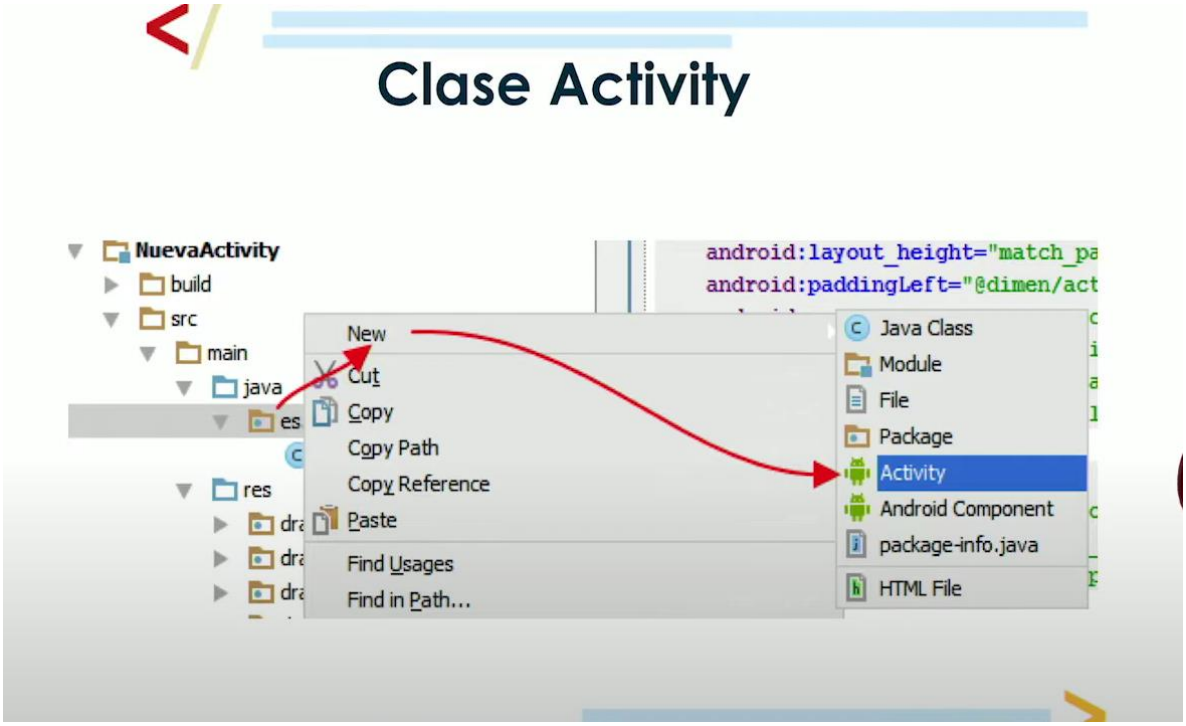


CICLO

5. Ciclo de vida de las aplicaciones para para dispositivos móviles.

Ciclo de vida de las aplicaciones para dispositivos móviles en el ciclo de vida de las aplicaciones se describe la creación y proceso de los pasos de una aplicación desde que se inicia hasta que se concluye

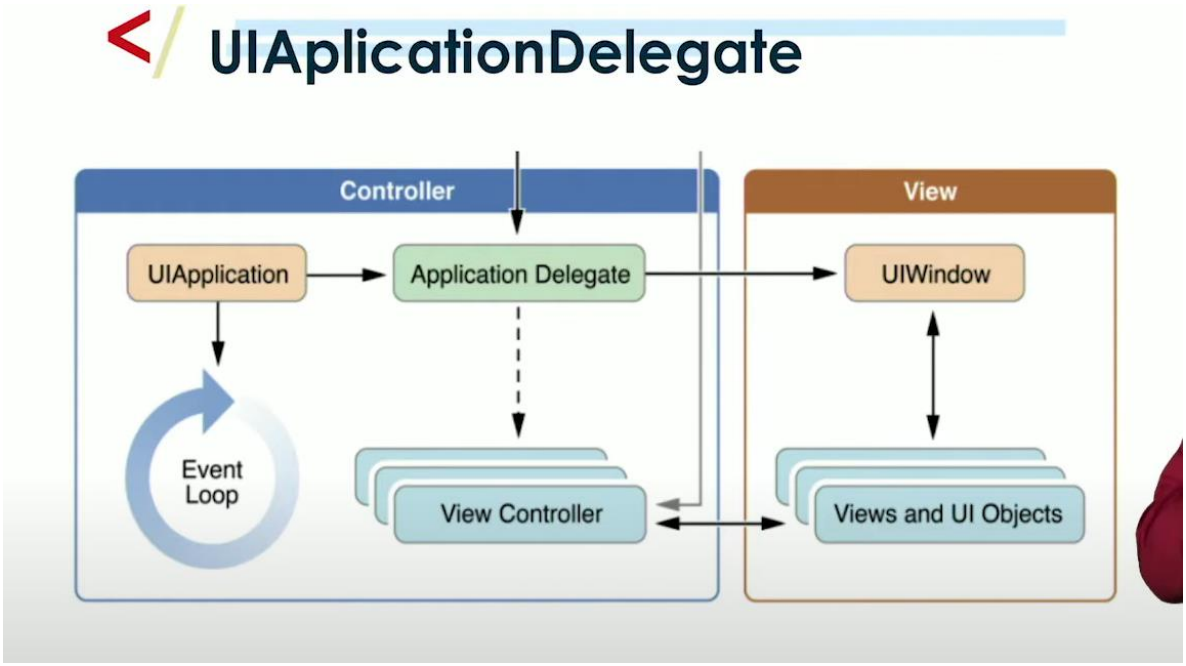
ciclo de vida en android al iniciar una aplicación se pasa por varios estados a esto le conocemos como el ciclo de vida de la aplicación en android la clase activity proporciona una serie de llamadas que permiten cambiar de estado para una aplicación ya sea este para crear una nueva o finalizar el proceso en el que se encuentra dentro de los métodos de la devolución de llamada de un ciclo de vida se puede declarar el comportamiento que tendrá la activity cuando el usuario la abandone y la reanude por ejemplo si creas un reproductor de vídeo puedes pausar el vídeo y cancelar la conexión de red cuando el usuario cambia a otra app cuando el usuario regrese podrá volver a establecer la conexión con la red y permitir que reanude el vídeo desde el mismo punto permite hacer el trabajo preciso en el momento adecuado y administrar las transiciones correctamente



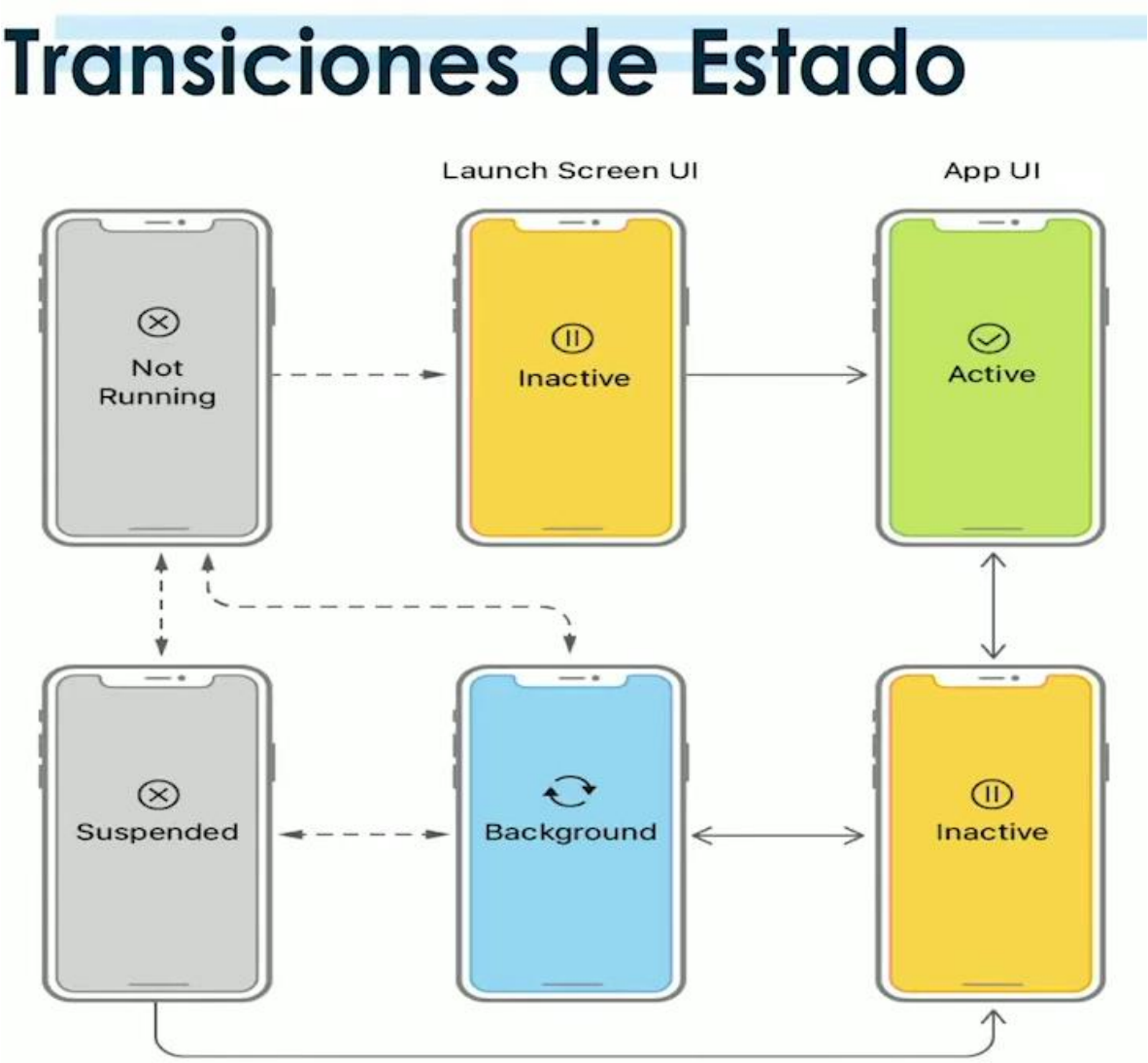
para navegar por las transiciones entre las etapas del ciclo de vida de una actividad la clase activity proporciona un conjunto básico de 6 devoluciones de llamadas con crear con start en resumen el sistema invoca cada una de estas devoluciones de llamada cuando una operación entra en un nuevo estado cuando el usuario comienza a abandonar la actividad el sistema llama a métodos para desmantelarla en algunos casos este desmantelamiento es solo parcial la actividad todavía reside en la memoria y aún puede volver al primer plano si el usuario regresa a esa actividad se reanuda desde donde el usuario la dejó con algunas excepciones se restringe a las apps para que no inicien actividades cuando se ejecutan en segundo plano la probabilidad de que el sistema finalicé un proceso determinado junto con las actividades que contiene depende del estado de la actividad en ese momento según la complejidad de tu actividad probablemente no necesite implementar todos los métodos del ciclo de vida sin embargo es importante que comprendas cada uno de ellos y que implemente es aquellos que garanticen que tu app se comporte como esperan los usuarios

ciclo de vida en hijos míos cuando el estado de una aplicación cambia ui kit lo notifica llamando a los métodos del objeto delegado apropiado ios 13 y versiones posteriores se usa el objeto ui ex en delegar para responder a eventos del ciclo de vida en una aplicación basada en escena en ios 12 y versiones anteriores se usa el objeto ui application delegue para responder a eventos del ciclo de vida

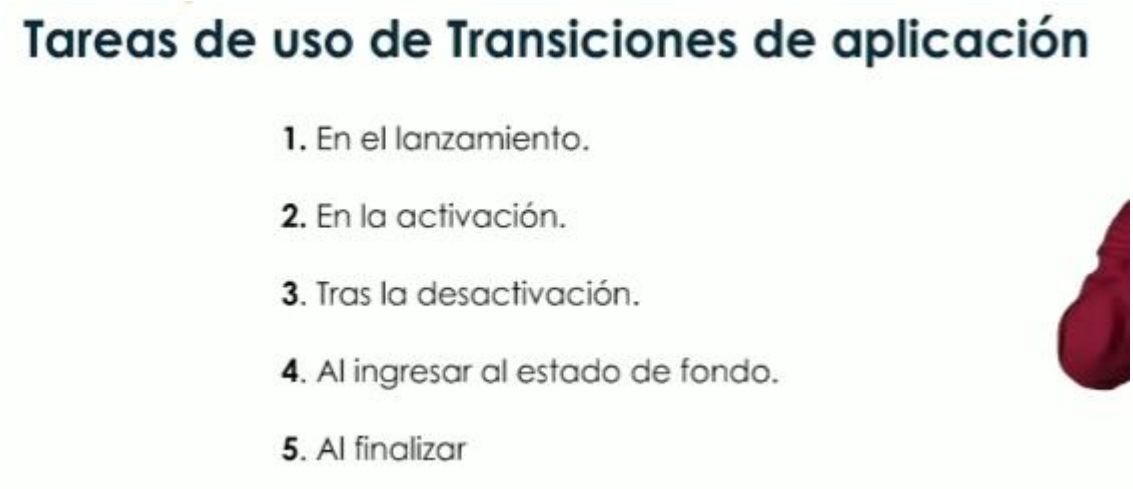
responder a eventos del ciclo de vida basado en aplicaciones en hijos 12 y versiones anteriores y en aplicaciones que no admiten escenas wiki de entrega todos los eventos del ciclo de vida al objeto we application delegada el delegado de la aplicación administra todas las ventanas de su aplicación incluidas las que se muestran en pantalla separadas como resultado las transiciones de estado de la aplicación afecta a la interfaz de usuario completa de la aplicación incluido el contenido en pantallas externas



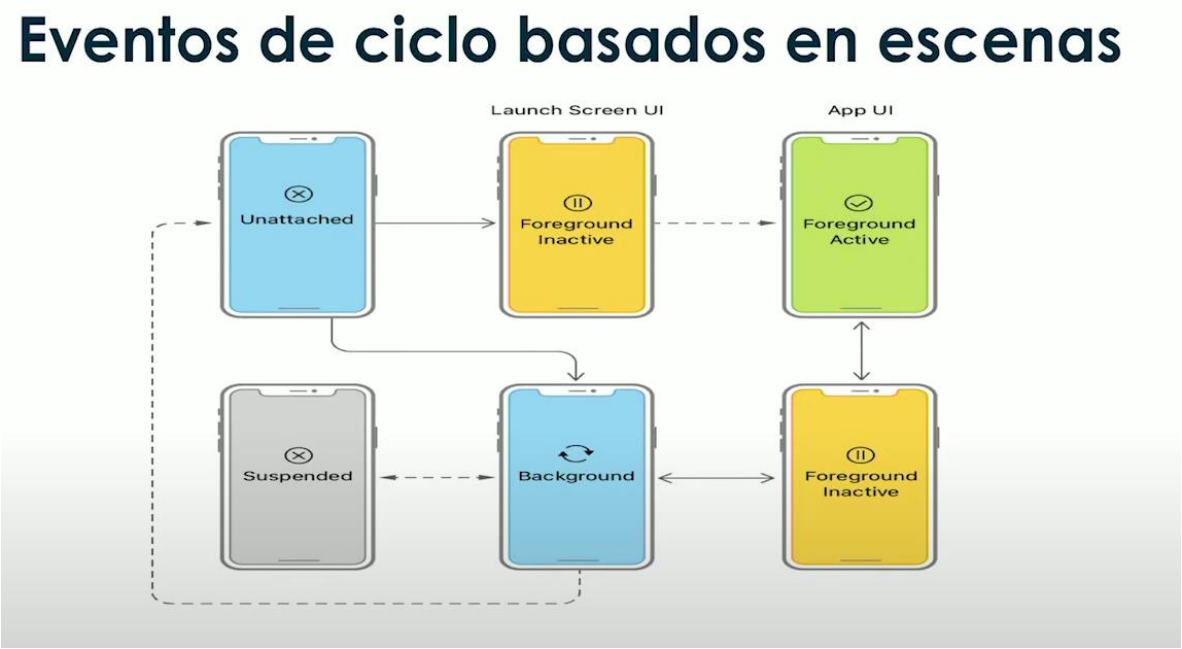
en la diapositiva se muestran las transiciones de estado que involucran el objeto delegado de la aplicación después del inicio el sistema coloca la aplicación en el estado inactivo o en segundo plano dependiendo si la iu está a punto de aparecer en la pantalla cuando se inicia en primer plano el sistema pasa a la aplicación al estado activo automáticamente después de eso el estado fluctúa entre activo y en segundo plano hasta que la aplicación finaliza



use las transiciones de la aplicación para realizar las siguientes tareas el lanzamiento inicializa las estructuras de datos y la iu de su aplicación en la activación termine de configurar su iu y prepárese para interactuar con el usuario tras la desactivación guarde datos y silencia el comportamiento de su aplicación al ingresar al estado de fondo finalicé tareas cruciales libere la mayor cantidad de memoria posible y prepárense para ver su aplicación al finalizar detenga todo el trabajo de inmediato y liberen los recursos compartidos

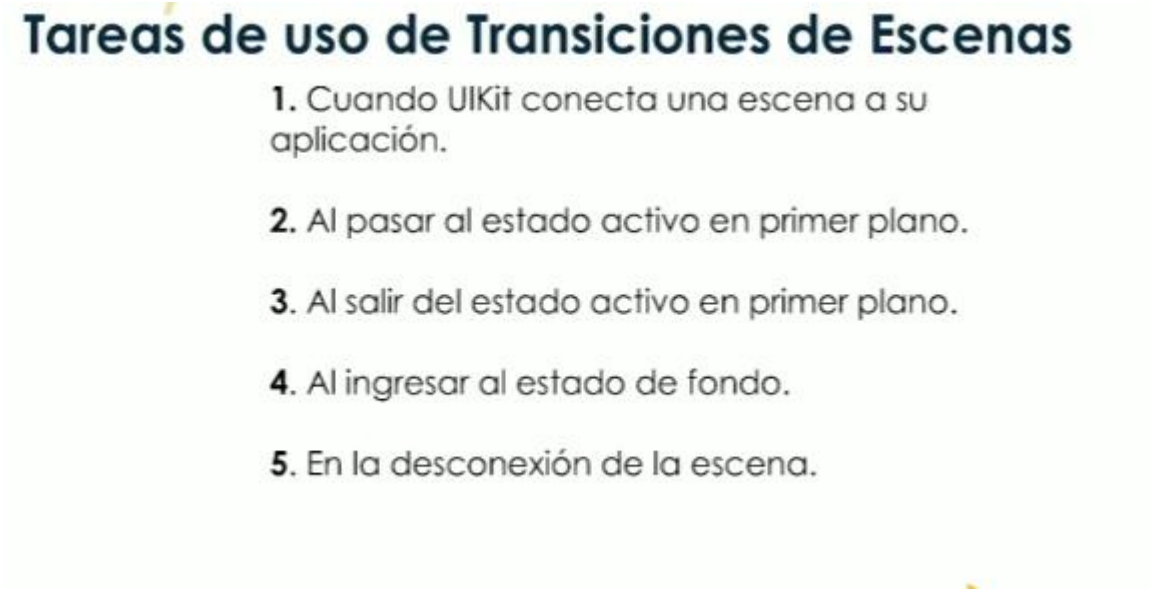


eventos del ciclo de vida basados en la escena si su aplicación admite escena wiki ofrece eventos de ciclo de vida separados para cada uno una escena representa una instancia de la interfaz de usuario de su aplicación que se ejecuta en un dispositivo el usuario puede crear múltiples escenas para cada aplicación y mostrarlas y ocultarlas por separado debido a que cada escena tiene su propio ciclo de vida cada una puede estar en un estado diferente de ejecución por ejemplo una escena puede estar en primer plano mientras que otras están en segundo plano o están suspendidas la diapositiva muestra la transición de estado para escena cuando el usuario o el sistema solicita una nueva escena para su aplicación wiki la crea y la coloca en el estado no conectado las escenas solicitadas por el usuario se mueven rápidamente al primer plano donde aparecen en la pantalla una escena solicitada por el sistema generalmente se mueve al fondo para que pueda procesar un nuevo evento por ejemplo el sistema puede iniciar la escena en segundo plano para progresar un evento de ubicación cuando el usuario descarta la interfaz de usuario de su aplicación iu kit mueve la escena asociada al estado de fondo y finalmente al estado suspendido ui kit puede desconectar un fondo o una escena suspendida en cualquier momento para recuperar sus recursos devolviendo esa escena al estado no conectado



utilicé las transiciones de escenas para realizar las siguientes tareas cuando wikis conecta a una escena a su aplicación configura el ui kit inicial de su escena y cargue los datos que necesita al pasar al estado activo en el primer plano configure su iui prepárese para interactuar con el usuario consulta preparación de su iui para ejecutar en primer plano al salir del estado activo en primer plano guarde los datos y silencia su comportamiento de su aplicación consulte preparación de su iui para ejecutar en segundo plano al ingresar al estado de fondo finalicé tareas cruciales libere la mayor cantidad de memoria posible y prepárense para ver su aplicación consulte la segundo plano en la desconexión de la escena limpie los recursos compartidos asociados con la escena además de los eventos relacionados con la escena también debe responder al inicio de su aplicación utilizando su objeto ui application deleget para detener obtener información sobre qué hacer al iniciar la aplicación consulte a

responder el lanzamiento de su application



Tutoriales

Instalación de Android Studio.

Tutoriales

Creación de Dispositivos virtuales.