



Universidad Nacional Autónoma de México
Facultad de Ingeniería
División de Ingeniería Eléctrica



Alumnos:
López Tavera Alexa Fernanda
Ugalde Santos Atzin

Profesor:
Ing. Gunnar Eyal Wolf Iszaevich

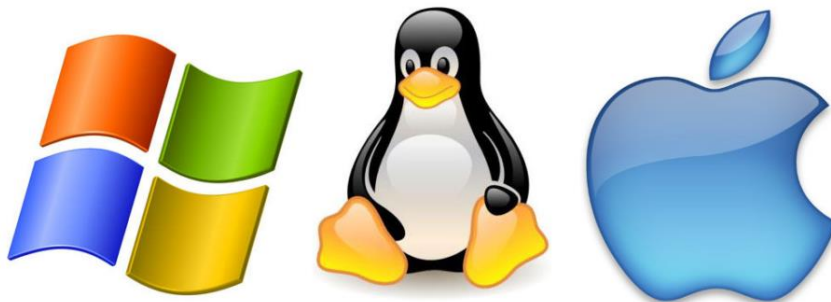
Asignatura:
Sistemas Operativos

Grupo:
06

Tema:
Exposición "Sistemas Operativos Móviles, ¿Android ó iOS?"

Semestre:
2024-2

Fecha de Entrega:
04/Abril/2024



1.(Sistemas Operativos, 2021)

Índice

I.	¿Qué son los sistemas operativos móviles?.....	3
	I.I Arquitectura: Android y iOS.....	3
	I.I.I. Interfaz de usuario: Capas.....	3
II.	Gestión de Energía.....	6
	III.I Administración de la batería en Android y iOS.....	6
	III.II Optimización de la batería en Android y iOS.....	7
III.	Frameworks.....	8
	III.I Android: Android SDK.....	8
	III.II iOS: Cocoa Touch.....	9
IV.	Entornos de Desarrollo Integrados.....	10
	IV.I Android Studio: Características.....	11
	IV.II iOS Xcode: Características.....	12
V.	Referencias consultadas.....	13

I. ¿Qué son los Sistemas Operativos móviles?

Un sistema operativo móvil ó para móvil, según la página *Nobbot* es “Un conjunto de programas y órdenes capaces de ejecutarse sobre el hardware del dispositivo para satisfacer las necesidades del usuario. En los teléfonos móviles, estas suelen ser de conexión a la red, por lo que el SO móvil está muy orientado a la conectividad inalámbrica”. (1)

Lo anterior se consigue con la correcta gestión de la memoria y cargando nuevos programas, lo que permite interactuar y darle ordenes al celular.

I.I Arquitectura: Android y iOS

I.I.I Interfaz de usuario: Capas

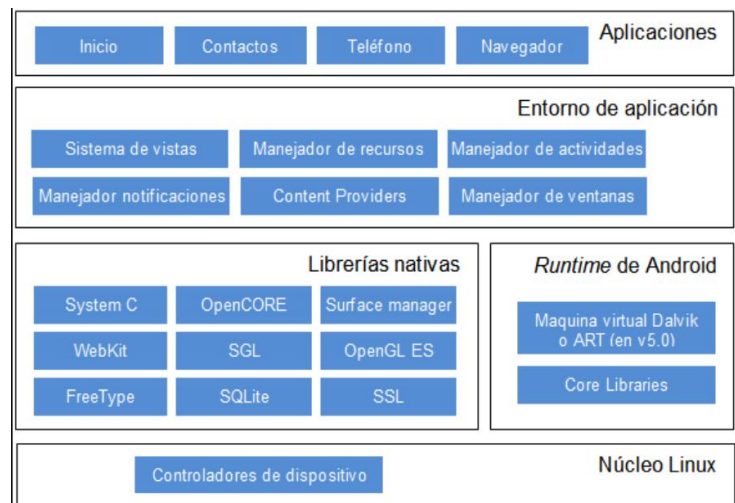
La arquitectura de un sistema operativo es el conjunto de componentes y servicios que permiten el funcionamiento básico de un sistema informático.

A continuación, veremos estos componentes para los dos SO de los que hablaremos a lo largo de toda la investigación: Android y iOS. (2)

Android

Está constituida por 4 capas o niveles.

La estructura de Android se encuentra construida sobre el Kernel de Linux. Luego hay una capa de librerías relacionadas con una estructura administradora en Tiempo de ejecución. En el siguiente nivel se encuentra un Framework de apoyo para construcción de aplicaciones y posteriormente la capa de Aplicaciones. (3)



2. (Arquitectura Android, s.f.)

A continuación, se describe cada capa:

1. **Kernel de Linux:** Android está construido sobre el núcleo de Linux debido a la excelente portabilidad, flexibilidad y seguridad que Linux presenta; ya que Linux está bajo la licencia de GPL, Android también.

2. **Capa de Librerías:** En esta capa se encuentran partes como la HAL (componente que permite la independencia del hardware), librerías nativas (interfaces nativas como OpenGL, SQLite para bases de datos, etc), demonios

o daemons(códigos que se ejecutan para ayudar al servicio del sistema) y la máquina virtual (Android tiene su propia máquina virtual interpretadora de bytecodes llamada Dalvik).

3. **Framework para aplicaciones**: Es la capa que le interesa a los desarrolladores ya que en ella se encuentran todas las librerías Java que se ocupan para programar las aplicaciones. Los paquetes con más preponderancia son los android.*, en ellos se alojan todas las características necesarias para construir una aplicación Android.

4. **Capa de Aplicaciones**: Es la capa que se centra en la ejecución, comunicación y estabilidad de las aplicaciones preinstaladas por el fabricante o las construidas por el desarrollador. A ella acceden todos los usuarios de Android debido a su fácil nivel de comprensión.

iOS

Está constituida por 4 capas o niveles. (4)

A continuación, se describe cada capa:

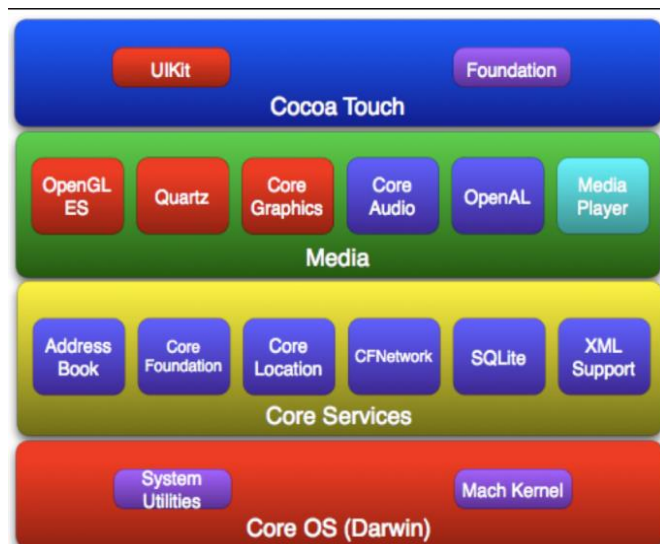
1. **Core OS**: Es la capa del núcleo del SO, basado en Darwin BSD. Es la capa más baja de la pila iOS y se sitúa directamente sobre el dispositivo hardware.

El kernel está basado en "MATCH" y es responsable de todos los aspectos del sistema operativo. Se encarga de:

- Gestión de la memoria virtual.
- Gestión de procesos, hilos y manejo de la memoria.
- Gestión del acceso a red de bajo nivel.

2. **Capa de Core Services**: En esta capa se le permite al usuario acceder a todos los servicios básicos y contiene los servicios fundamentales del SO que pueden ser usados por todos los desarrolladores de aplicaciones, por ejemplo:

- Almacenamiento de iCloud: Permite que las aplicaciones escriban documentos y los datos a una ubicación central en la nube para acceder desde otros dispositivos del usuario.



- Grand Central Dispatch(GCD): Ofrece alternativa para tareas de bajo nivel como leer y escribir, medidores de tiempo de ejecución y monitoreo de señales de control de procesos y eventos.

3. **Capa Media**: Conocida también como “capa de medios de comunicaciones”. Es una interfaz en el lenguaje C que permite la ejecución de tareas o ficheros multimedia. Contiene las tecnologías de gráficos, audio y video orientadas a crear la mejor experiencia de multimedia disponible en el dispositivo móvil.

4. **Capa Cocoa Touch**: Es la interfaz de los dispositivos para con el usuario. Es una exclusiva interfaz con un diseño único que está constituida con aplicaciones probadas que comparten patrones que se encuentran en la PC Mac pero se reconstruyeron con un enfoque especial en el tacto y las interfaces basadas en la optimización.

Esta interfaz provee la infraestructura básica clave para desarrollar las aplicaciones iOS, ya que contiene los recursos principales para ejecutar aplicaciones iOS, entre los recursos que se distinguen en esta capa tenemos las High Level Features como:

- Multitarea: Hace posible la transición requerida para la ejecución de la aplicación desarrollada y ejecutada, gracias al uso de UIKit.
- Protección de datos: Permite a las aplicaciones trabajar con datos sensibles del usuario, aprovechando la encriptación implícita (Cuando se bloquea el dispositivo, el contenido del fichero es inaccesible, cuando se desbloquea, se genera una clave de descryptación que permite a la aplicación nuevamente acceder al fichero).

II. Gestión de Energía

II.I Administración de la batería en Android y iOS

En Android:

En el SO de Android, se gestiona la batería por medio de mecanismos integrados en el mismo sistema operativo; estos mecanismos están diseñados para optimizar el consumo de energía y maximizar la duración de la batería del dispositivo. (5)

Algunos de los principales métodos de administración de batería son:

1. **Modo Ahorro de Batería:** Es un método que se activa automáticamente cuando la carga de la batería alcanza un nivel preestablecido. Con este método, se limitan las actualizaciones en segundo plano, se restringen las vibraciones y se reduce el rendimiento del dispositivo para conservar la energía el mayor tiempo posible.

2. **Administrador de Batería:** Android tiene incluido un administrador de batería que proporciona información detallada del consumo de energía de cada aplicación; los usuarios pueden acceder a esta información de manera sencilla y limitar el uso de energía por aplicación.



3. **Limitar tareas en segundo plano:** Las aplicaciones que se ejecuten en segundo plano, reciben restricciones en cuanto a la frecuencia con la que pueden realizar tareas, acceder a la red, utilizar sensores o cualquier otra actividad que consuma energía.

4. (Prolongar la duración de tu batería Android, 2011)

En iOS:

Por defecto, iOS administra algunas estrategias para la correcta administración de la batería: (6)

1. **Optimización del Sistema Operativo:** Apple trabaja en la optimización del sistema operativo reduciendo el uso del CPU y de la memoria que a su vez ocasiona un mayor consumo de energía.

2. **Modo de Bajo Consumo (Low Power Mode):** iOS incluye este modo que se activa de forma manual por los usuarios cuando la batería esta baja; con este modo, se reduce el rendimiento del dispositivo y desactiva ciertas funciones ejecutadas en segundo plano.



5. (Maximizar la duración de la batería en iOS 11, s.f.)

3. **Gestión de Redes y Conectividad:** Los dispositivos con iOS administran la conectividad de manera inteligente para ahorrar batería; un ejemplo de ello es que son capaces de desconectarse automáticamente del Wi-Fi cuando no está en uso.

4. **Actualizaciones y Parches de Software:** Con las constantes actualizaciones del software, se incluyen mejoras en el rendimiento de la batería y optimizaciones de energía.

II.II Optimización de la batería en Android y iOS

Para optimizar la batería en Android:

Algunas de las acciones que puede seguir el usuario para optimizar la batería son: (7)

1. Reducir el brillo de la pantalla.
2. Configurar el brillo para que se ajuste automáticamente.
3. Desactivar la vibración o el sonido del teclado.
4. Restringir las aplicaciones que consumen mucha batería.
5. Activar el modo oscuro.
6. Evita que el dispositivo se sobrecaliente.
7. Utilizar el adaptador de corriente incluido con el dispositivo.

Para optimizar la batería en iOS:

Algunas de las acciones que puede seguir el usuario para optimizar la batería en iPhone, iPad y iPod son:

(8)

1. Tener siempre el software actualizado.
2. Optimiza los ajustes de iluminación y brillo automático.
3. Intentar siempre conectarse a Wi-Fi ya que consume menos energía que la conexión a datos móviles.
4. Activar el Modo de Bajo Consumo.
5. Enchufar y encender el ordenador para cargar el dispositivo.

III. Frameworks

Un framework o “marco de trabajo” es, según la página Ebac, “una plantilla para desarrollar softwares de manera más rápida y eficiente. Solo necesitas escribir el código que se encargue de la lógica de tu app”.(9)

Un framework define la estructura de un proyecto futuro y proporciona las herramientas necesarias como “bloques de construcción” para el código.

Su función principal es ahorrar tiempo y esfuerzo a los desarrolladores, evitando escribir código desde 0 y aplicando éstos, en múltiples proyectos.

III.I Android: Android SDK

El Android SDK es un conjunto de herramientas y recursos que los desarrolladores utilizan para crear aplicaciones y software para dispositivos Android. El SDK proporciona una variedad de componentes esenciales para el desarrollo de aplicaciones, incluyendo: (10)



6. (Instalar Android SDK, 2015)

1. Entorno de desarrollo: Incluye un entorno de desarrollo integrado llamado Android Studio, que es la principal herramienta para escribir, depurar y compilar aplicaciones Android.

2. Emuladores y Simuladores: Proporciona emuladores y simuladores que permiten a los desarrolladores probar sus aplicaciones en distintas versiones de Android.

3. Herramientas de Depuración: Incluye herramientas de depuración que permiten a los desarrolladores rastrear errores, inspeccionar variables y realizar pruebas exhaustivas de sus aplicaciones.

4. Documentación: Ofrece documentación detallada y ejemplos de código para ayudar a los desarrolladores a comprender cómo utilizar las APIs y construir aplicaciones Android efectivas.

5. SDK de Google Play Services: Además del Android SDK estándar, los desarrolladores también pueden utilizar el SDK de Google Play Services para acceder a servicios de Google como Google Maps, Google Drive o la autenticación de cuentas de Google.

III.II iOS: Cocoa Touch

Es una API que se utiliza para la creación de programas en el iPad, iPhone y iPod Touch de la compañía Apple Inc. (11)

Está escrito en Objective-C; ofrece un modelo de desarrollo basado en eventos en el que los componentes de la interfaz de usuario generan eventos que desencadenan acciones en el código. También puede utilizar los gestos para desencadenar acciones, y Cocoa Touch incluye soporte incorporado para los gestos comunes, tales como toques, deslizamientos y pellizcos.

La API se divide en dos categorías principales:

1. El Foundation Framework, que proporciona tipos de datos y servicios básicos, como el manejo de redes y archivos.
2. El UIKit Framework, que proporciona los componentes de la interfaz de usuario y el manejo de eventos necesarios para construir aplicaciones iOS.

Algunas de las características que posee Cocoa Touch son: (12)

1. Gestión de datos.
2. Selector de documentos.
3. Kit de texto.
4. Guiones gráficos.
5. Reconocedores de gestos
6. Diseño automático.



7. (WHAT'S NEW IN COCOA TOUCH, s.f.)

IV. Entornos de desarrollo integrados

IV.I Android Studio: Características

Es el IDE oficial de Android para el desarrollo de aplicaciones, esta basado en IntelliJ IDEA; un entorno o ambiente de desarrollo para programas, que posee potentes herramientas de edición de código; por ejemplo, cuando el programador tiene errores, es capaz de marcarlos de forma inmediata para así dar una solución rápida a ellos.(13)

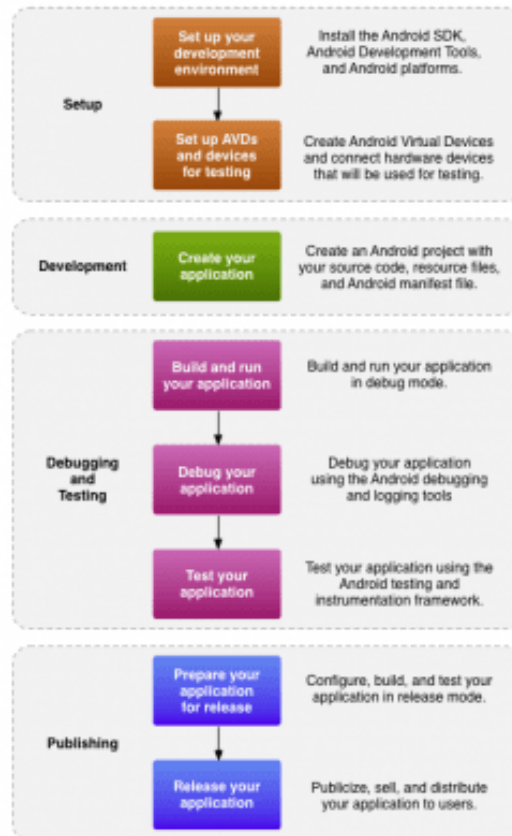
Android Studio posee distintas características para el desarrollo de aplicaciones, por ejemplo:

1. Tiene un sistema de construcción basado en Gradle (Sistema de compilación).
2. Posibilidad de trabajo con Logcat (registro de mensajes).
3. Soporte integrado para Google Cloud Platform.
4. IntelliJ IDEA soporta variados lenguajes que se basan en JVM; Java (por ello la “J” de IntelliJ), Clojure, Groovy, Kotlin y Scala.
5. Un soporte para Maven y Gradle.

El proceso de desarrollo de aplicaciones en Android podría dividirse en cuatro etapas que se presentan a continuación:

1. **Configuración de entorno:** Durante esta fase se instala y configura el entorno de desarrollo; se realiza la conexión a los elementos en donde se puede realizar la instalación de la app y se crean dispositivos virtuales Android (AVDS).
2. **Configuración del Proyecto y Desarrollo:** Durante esta fase se realiza el proyecto y el desarrollo de éste. Aquí se crean los módulos que contengan recursos para la aplicación y archivos de código fuente.
3. **Pruebas, depuración y construcción de la aplicación:** Se construye el proyecto en un paquete (s) depurable .apk que se puede instalar y ejecutar en el emulador o en un dispositivo con Android. Se utiliza un sistema de construcción basado en Gradle. Con este se proporciona flexibilidad, variantes de construcción a la medida y la resolución de dependencias. En el caso de usar otro IDE se puede desarrollar el proyecto usando Gradle, y a su vez, instalarlo en un dispositivo que use ADB. Posteriormente se hace la depuración de la aplicación a través de los mensajes de supervisión de dispositivos, más un dispositivo de registro de Android (Logcat) junto con la idea de IntelliJ. Además, Se puede usar un depurador JDWP compatible, adicionando las herramientas de depuración y de registro que se proporcionan con el SDK de Android. Para finalizar, se usan las herramientas de prueba SDK de Android para las pruebas a la aplicación.

4. **Publicación de la aplicación:** Se realiza la configuración y se arma la solicitud para el uso y libre distribución de la aplicación a los usuarios. Durante la etapa de preparación se construye una versión de la aplicación, que los usuarios pueden descargar e instalar en sus dispositivos de modo que se pueda vender y distribuir esa versión de la aplicación.



8. (Características y cualidades de Android Studio, s.f.)

IV.II iOS Xcode: Características

Es el entorno de desarrollo integrado creado por Apple y es herramienta esencial para los desarrolladores que trabajan en este ecosistema y está diseñado para que los desarrolladores puedan construir aplicaciones para dispositivos iOS, macOS, watchOS, tvOS, iPadOS y ahora visionOS. (14)

Este entorno incluye herramientas como un editor de código, compiladores, depuradores que son esenciales para el desarrollo de software.

Algunas funciones destacables de Xcode son:

1. El programa ofrece un sistema guiado para los desarrolladores; desde el principio da la opción de elegir en un principio si se va a crear una aplicación, un juego e incluso una extensión de Safari.
2. Editor de código avanzado: Ofrece un editor de texto capaz de admitir múltiples lenguajes de programación incluidos Swift y Objective-C. Este editor proporciona características inteligentes como resaltado de sintaxis, completado automático y corrección de errores.
3. Simuladores y depuradores: Xcode proporciona simuladores de dispositivos iOS, macOS, watchOS y tvOS que permiten probar y depurar aplicaciones en entornos virtuales antes de lanzarlas. Es decir, se puede lanzar la aplicación que se esté creando en un iPhone virtual con la versión del software que quieras para probar su interfaz, su funcionamiento o si puede tener algún bug.
4. Con la opción de Instruments, los desarrolladores pueden medir el rendimiento de sus aplicaciones a la hora de buscar posibles cuellos de botella.
5. Integración con SDK y frameworks: Xcode ofrece una estrecha integración con los SDK (Software Development Kits) y frameworks de Apple, y esto va a permitir a los desarrolladores acceder a las API oficiales de Apple que han sido diseñadas especialmente para el software donde están trabajando.
6. Se permite probar la integración del proyecto en las nuevas versiones de software que se van lanzando por parte de Apple.

Referencias consultadas:

1. De las imágenes ilustrativas:

1. Redeweb, & Redeweb. (2023, 12 mayo). *Sistemas operativos*. Revista Española de Electrónica | Todas las Noticias de Electrónica Actualizadas A Diario. <https://www.redeweb.com/actualidad/sistemas-operativos/>
2. *Arquitectura de Android*. (s/f). Androidcurso.com. Recuperado el 10 de marzo de 2024, de <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>
3. *Documentation archive*. (s/f). Apple.com. Recuperado el 10 de marzo de 2024, de <https://developer.apple.com/library/archive/navigation/>
4. iRe, iRe, & iRe. (2013, 23 mayo). *Cómo prolongar la duración de batería en tu Android* | ADNFriki. ADNFriki.com |. https://adnfriki.com/prolongar-la-bateria-de-tu-android/#google_vignette
5. *Cómo maximizar la duración de la batería en iOS 11* - consejotecnologico.com. (2019, 23 marzo). consejotecnologico.com. <https://consejotecnologico.com/como-maximizar-la-duracion-de-la-bateria-en-ios-11/>
6. Rg, E. (2015, 29 septiembre). *Instalar Android SDK en Ubuntu*. Urban Tecno. <https://www.mundodeportivo.com/urbantecno/linux/android-sdk-ubuntu>
7. Kumar, V. (s. f.). *What's New in Cocoa Touch* Matrid Technologies. <https://www.matridtech.net/whats-new-in-cocoa-touch/>
8. Pedrini. (2016, 24 mayo). *Características y cualidades de Android Studio*. Desde Linux. <https://blog.desdelinux.net/caracteristicas-y-cualidades-de-android-studio/>

2. De la información recabada:

1. Martínez, M. (2021, 14 abril). *Sistema operativo del móvil: qué es, cómo funciona y cuál es más seguro*. Nobbot. <https://www.nobbot.com/sistema-operativo-movil/>
2. Webmazter. (2023, 20 junio). *Arquitectura de un sistema operativo*. Y Arquitectura. <https://www.yarquitectura.com/arquitectura-de-un-sistema-operativo/>
3. Revelo, J. (2020, 30 noviembre). *Aprendiendo sobre la arquitectura de Android*. Develou. <https://www.develou.com/aprendiendo-la-arquitectura-de-android/>
4. *Arquitectura de los sistemas operativos móviles* [1430WOxO6J4J]. (s. f.). <https://idoc.pub/documents/arquitectura-de-los-sistemas-operativos-moviles-1430woxo6j4j>

5. *Información acerca de cómo ahorrar batería.* (s. f.). Android Developers.
<https://developer.android.com/develop/connectivity/preserving-battery?hl=es-419>
6. Apple Inc. (s. f.). *Power down: Improve battery consumption - WWDC22 - Videos - Apple Developer.* Apple Developer.
<https://developer.apple.com/videos/play/wwdc2022/10083/>
7. *Optimizar la duración de la batería de un dispositivo Android - Ayuda de Android.* (s. f.).
<https://support.google.com/android/answer/7664692?hl=es#zippy=>
8. Apple. (s. f.). *Baterías - Optimizar el rendimiento.* Apple (España).
<https://www.apple.com/es/batteries/maximizing-performance/>
9. Sandoval, E. (2023, 21 septiembre). *Frameworks: Marcos de trabajo para programadores.* Ebac. <https://ebac.mx/blog/frameworks>
10. Centeno, A. (2023, 13 septiembre). *Android SDK: qué es, para qué sirve y cómo se instala.* <https://www.internetizado.com/android/sdk>
11. Noel. (2022, 26 octubre). *Cocoa Touch.* TechEdu.
<https://techlib.net/techedu/cocoa-touch/>
12. *What is cocoa?* (2013, 18 septiembre). Copyright 2018 Apple Inc. All Rights Reserved.
<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CocoaFundamentals/WhatIsCocoa/WhatIsCocoa.html>
13. Pedrini. (2016, 24 mayo). *Características y cualidades de Android Studio. Desde Linux.* <https://blog.desdelinux.net/caracteristicas-y-cualidades-de-android-studio/>
14. Lizana, J. A. (2024, 24 enero). *Xcode: qué es, para qué sirve y qué funciones para desarrolladores incluye este IDE de Apple.* Applesfera.
<https://www.applesfera.com/nuevo/xcode-que-sirve-que-funciones-para-desarrolladores-incluye-este-ide-apple>