



TEMA 062

CONCEPTOS BÁSICOS DE OTROS SISTEMAS OPERATIVOS: OS X, IOS, ANDROID, Z/OS. SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES

Versión

30.1

Fecha de actualización

06/09/2024



Contexto normativo asociado a los dispositivos móviles

- **Real Decreto 1112/2018, de 7 de septiembre**, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público → Las disposiciones relativas a dispositivos móviles son de aplicación desde el 23 de junio de 2021.
- **App Factory**, medida incluida en España Digital 2026 (medida 18 del eje 5) y en el Plan de Digitalización de las AAPP 2021-2025 (medida 1 del eje 1).

1. Introducción

En el presente tema se estudiarán las características principales de los siguientes sistemas operativos:

- macOS
- iOS
- Android
- z/OS

2. Mac OS

macOS (inicialmente **Mac OS X** y más tarde **OS X**) es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple Inc. desde 2001.

Es el sistema operativo principal para la familia de computadoras Mac de Apple, y dentro del mercado de ordenadores de escritorio, portátiles y familiares, es uno de los sistemas operativos de escritorio más utilizado junto con Microsoft Windows.

Algunas de las características que ofrece son las siguientes:

- Escrito en: **C, C++, Objective-C y Swift**
- Tipo de núcleo: **Híbrido (XNU)**
- Interfaz gráfica predeterminada: **Gráficos (Aqua)**
- Plataformas admitidas: **x86-x64** (10.4.7 - presente), **ARM64** (11.0 - presente)
- **Licencia Software comercial, software propietario**

macOS está basado en el núcleo creado por Mach, un proyecto de diseño de sistemas operativos iniciado en la Universidad Carnegie Mellon con el objetivo de desarrollar un micronúcleo. Ciertas partes de las implementaciones de UNIX por parte de FreeBSD y NetBSD fueron incorporadas, posteriormente, surgiendo Mac OS X, que representa la décima versión del sistema operativo de Apple para computadoras Macintosh.

Las versiones previas a Mac OS X (Mac OS 9 y anteriores) eran la base del Mac OS clásico, desarrollado íntegramente por Apple, cuya primera versión vio la luz en 1985. Su desarrollo se extendería hasta la versión 9 del sistema, lanzada en 1999. A partir de entonces se comenzó con el desarrollo de la versión 10 (Mac OS X). El sistema cambió su arquitectura totalmente y pasó a basarse en Unix, aunque su interfaz gráfica mantiene muchos elementos de las versiones anteriores. El núcleo de Mac OS X es un sistema operativo compatible con POSIX construido sobre el kernel XNU, con funciones estándar de Unix disponibles desde la interfaz de línea de



comandos. Este software fue liberado por Apple como un sistema operativo libre y de código abierto llamado Darwin, que es el núcleo de los componentes en los que se basan macOS, iOS, watchOS, tvOS, e iPadOS.

Con excepción del Mac OS X Server 1.0 y la beta pública original, las versiones del Mac OS X tienen nombres de grandes felinos y, desde la versión Mac OS X v10.9 (Mavericks), tienen nombres de diferentes lugares y montañas de California.

A continuación, se muestra una tabla con los nombres de las diferentes versiones de macOS.

Versión	Nombre en código	Fecha de Presentación
Mac OS X Server 1.0	Hera	16 de marzo de 1999
Beta pública	Kodiak	13 de septiembre de 2000
10.0	Cheetah	24 de marzo de 2001
10.1	Puma	25 de septiembre de 2001
10.2	Jaguar	24 de agosto de 2002
10.3	Panther	24 de octubre de 2003
10.4	Tiger	29 de abril de 2005
10.5	Leopard	26 de octubre de 2007
10.6	Snow Leopard	28 de agosto de 2009
10.7	Lion	20 de julio de 2011
10.8	Mountain Lion	25 de julio de 2012
10.9	Mavericks	22 de octubre del 2013
10.10	Yosemite	16 de octubre de 2014
10.11	El Capitán	30 de septiembre de 2015
10.12	Sierra	20 de septiembre de 2016
10.13	High Sierra	23 de octubre de 2017
10.14	Mojave	24 de septiembre de 2018
10.15	Catalina	7 de octubre de 2019
11	Big Sur	12 de noviembre de 2020
12	Monterey	25 de octubre de 2021
13	Ventura	24 de octubre de 2022
14	Sonoma	26 de septiembre de 2023
15	Sequoia	TBA

Mac OS X posee una **arquitectura modular** que se construye a partir de cuatro componentes básicos:

- La interfaz de usuario Aqua
- Sistemas de gráficos basados en estándares
- Un conjunto de estructuras de aplicaciones
- El sistema operativo central Darwin

Aqua

Es la interfaz de usuario creada por Apple para el Mac OS X; utiliza colores, transparencias y animaciones para aumentar la funcionalidad y la coherencia del sistema y de las aplicaciones.

Gráficos

El sistema de gráficos del Mac OS X se compone de tres potentes tecnologías basadas en estándares que están totalmente integradas en el sistema operativo y proporcionan servicios a nivel de sistema. **Quartz 2D** es una librería de generación de gráficos de alto rendimiento basada en el estándar multiplataforma **PDF** (Portable



Document Format) de Adobe. Permite visualizar e imprimir gráficos y textos suavizados de alta calidad y proporciona soporte de última generación para las tipografías OpenType, PostScript y TrueType. **OpenGL** es el estándar del mercado para visualizar formas y texturas **tridimensionales**. El Mac OS X incorpora una implementación totalmente estandarizada, optimizada e integrada. El software para medios digitales QuickTime provee un entorno totalmente estandarizado para crear, reproducir y generar vídeo (MPEG-4), audio (AAC o Codificación de Audio Avanzada) e imágenes (JPEG 2000 y otros formatos).

Estructuras

El Mac OS X incorpora un conjunto de estructuras de aplicaciones que permite trabajar con desarrolladores de diferentes comunidades.

Cocoa es un framework orientado a objetos diseñado para agilizar el desarrollo de aplicaciones nativas para Mac OS X, ya que permite añadir potentes interfaces de usuario gráficas de Aqua al software UNIX existente y crear aplicaciones completamente nuevas desde cero. En el caso de **iOS** se utiliza el framework "**Cocoa Touch**".

Carbon está concebido para simplificar la migración a aquellos desarrolladores que desean realizar una transición de sus aplicaciones del Mac OS 9 al Mac OS X.

Java permite el desarrollo y la ejecución de programas multiplataforma Java 2 Edición Estándar en Mac OS X, incluidos los que están programados con Java Developer Kit (JDK) 1.4.1.

Darwin

Bajo la interfaz de fácil uso y los gráficos del Mac OS X subyace Darwin, una **base UNIX** sólida de **código abierto** construida a partir de la aplicación de tecnologías ampliamente contrastadas como FreeBSD, Mach, Apache y gcc. Darwin es un sistema operativo completo, comparable a Linux o FreeBSD, en el que puede utilizarse el entorno de líneas de comando, red, librerías y kernel con el que están familiarizados los usuarios de UNIX.

3.iOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), posteriormente se ha usado en dispositivos como el iPod touch y el iPad.

iOS se deriva de **MacOS**, que a su vez está basado en **Darwin BSD** y, por lo tanto, es un **sistema operativo** tipo **Unix**. La última versión del sistema operativo es **iOS 17.6.1** publicada el **19 de agosto de 2024** (la versión **iOS 18** está previsto que se publique en septiembre de 2024).

Algunas de las características técnicas que ofrece son las siguientes:

- Escrito en: **C, C++, Objective-C y Swift**
- Tipo de núcleo: **Híbrido (XNU)**
- Interfaz gráfica predeterminada: **Cocoa Touch**
- Plataformas admitidas: **ARM** (iPhone y iPod Touch)
- Se puede actualizar mediante iTunes

Como características generales, se pueden destacar las siguientes:



- Interfaz gráfica diseñada para pantalla táctil, con capacidad para gestos "multitouch".
- Interfaz basada en deslizadores, interruptores y botones.
- Soporta acelerómetros del dispositivo.
- Soporta multitarea (con limitaciones).
- No tiene soporte para Adobe Flash ni Java.

iOS cuenta con una **arquitectura** formada por **cuatro capas de abstracción**

- La capa de "Cocoa Touch"
- La capa de "Medios" (Media Services)
- La capa de "Servicios Principales" (Core Services)
- La capa del núcleo del sistema operativo (Core OS)

Capa Cocoa Touch

Es la capa superior, la que los usuarios utilizan para interactuar con las aplicaciones, es decir, la capa visible. Es la zona donde se encuentran los componentes visuales, tratándose de una capa de abstracción.

Capa Media Services

Se trata de una capa basada en la mezcla de lenguaje C y Objective C que contiene las tecnologías que dan acceso a ficheros multimedia relacionados con audio, gráficos, vídeos, etc.

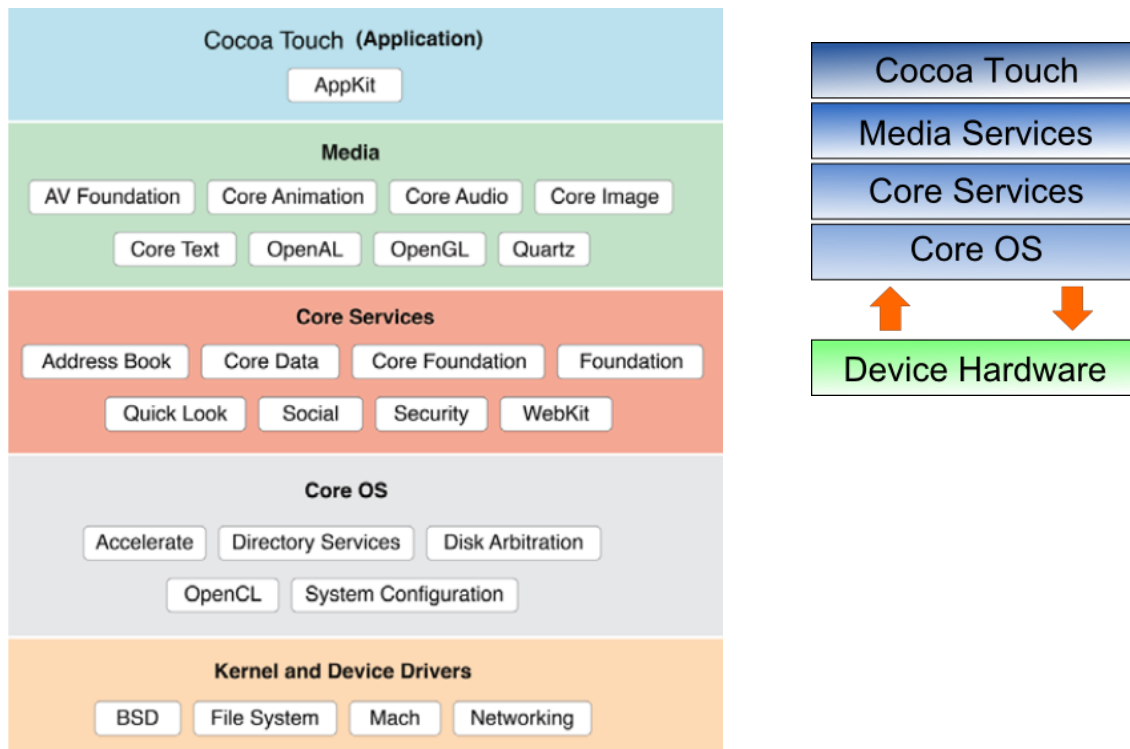
Capa Core Services

Se trata de la capa de servicios principales disponibles en el dispositivo y que pueden ser utilizados por todas las aplicaciones como pueden ser la base de datos SQLite, el acceso a la red o el soporte para XML.

Capa Core OS

Representa el núcleo del sistema. Está basado en el OS X de Apple, que fue desarrollado a partir de una base Unix. En esta capa se encuentran los elementos de seguridad, la memoria, los procesos o el manejo de ficheros, por ejemplo.

A continuación, se presenta el framework del SO iOS desglosado en los distintos componentes que lo conforman.



Para realizar desarrollos, se utiliza el entorno de desarrollo integrado (IDE, en sus siglas en inglés) **Xcode**, escribiendo las aplicaciones en **Objective-C o Swift**. El formato de archivo para las aplicaciones de Apple en los dispositivos iPhone, iPod Touch e iPad es “**ipa**”.

iOS no permite la instalación en hardware de terceros y para poder saltarse dicha limitación es necesario realizar el llamado “**jailbreak**”.

El jailbreak, en iOS, es el proceso de saltarse las limitaciones impuestas por Apple en dispositivos que usen el sistema operativo a través del uso de kernels modificados. El jailbreak permite a los usuarios acceder al sistema de archivos del sistema operativo, permitiéndoles instalar aplicaciones adicionales, extensiones y temas que no están disponibles en la App Store oficial. Un dispositivo con jailbreak puede seguir usando la App Store, iTunes y las demás funciones normales como, por ejemplo, realizar llamadas.

4.Android

Android es un sistema operativo móvil desarrollado por Google, basado en el **Kernel de Linux**, de código abierto, que se ejecuta sobre **hardware con arquitectura ARM**. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores.

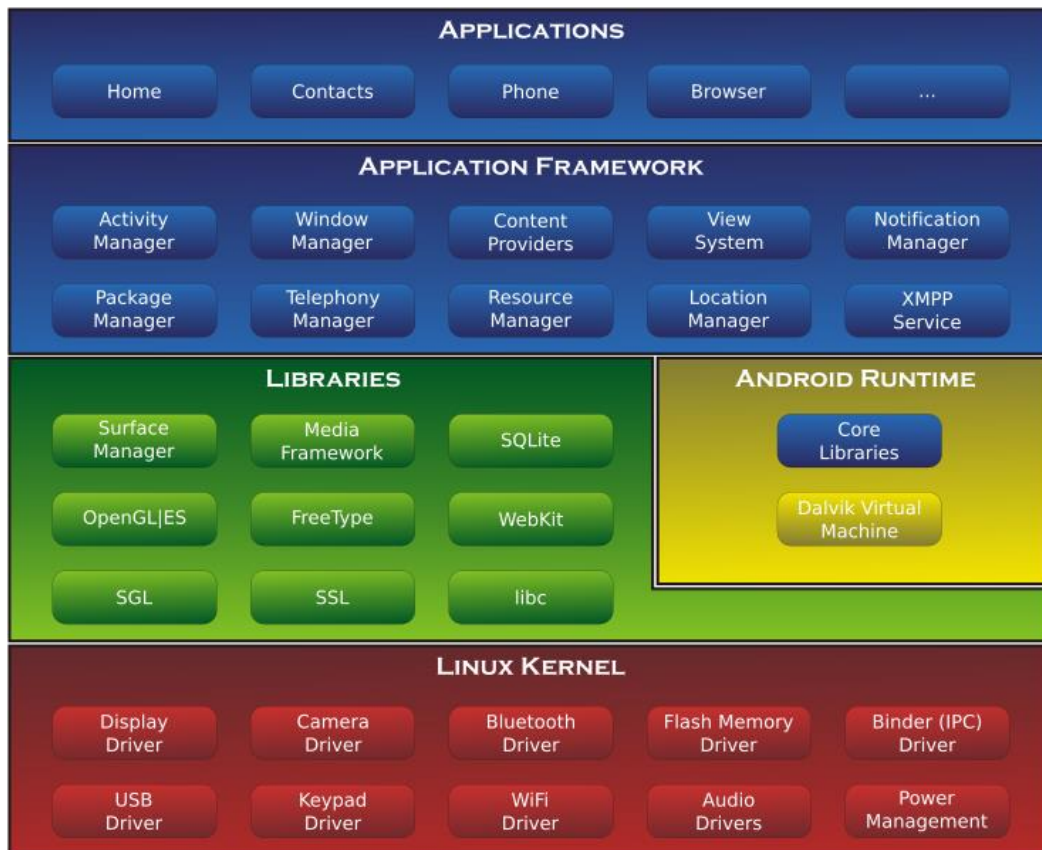
El código fuente de Android es liberado bajo las licencias de código abierto Apache 2.0 y GNU GPL 2.

La versión básica de Android es conocida como Android Open Source Project (AOSP), y es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado del 72,15 %, en junio de 2024, frente al 27,19% de iOS, según los datos de StatCounter publicados por Statista. Además, gracias a los datos de la consultora de tecnología IDC, podemos saber que los dispositivos Android representaron algo más del 84% de las unidades en 2020, y los iOS de Apple casi el 16% restante. Las versiones de Android más utilizadas en enero del 2022 son



Android 10 con un 26,5% y Android 11 con un 24,3%.

Android presenta la siguiente **arquitectura**, formada por 4 capas principales divididas en 5 secciones:



Los componentes principales del sistema operativo de Android son los siguientes:

- **Aplicaciones (Applications):** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en los lenguajes de programación **Java o Kotlin**.
- **Marco de trabajo de aplicaciones (Application Framework):** los desarrolladores tienen acceso completo a las mismas API del entorno de trabajo usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas (Libraries):** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android. Algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android (Android Runtime):** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Desde la versión 5.0 utiliza el Android Runtime (ART).



- **Núcleo Linux (Linux Kernel):** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Las aplicaciones Android están formadas por los siguientes componentes:

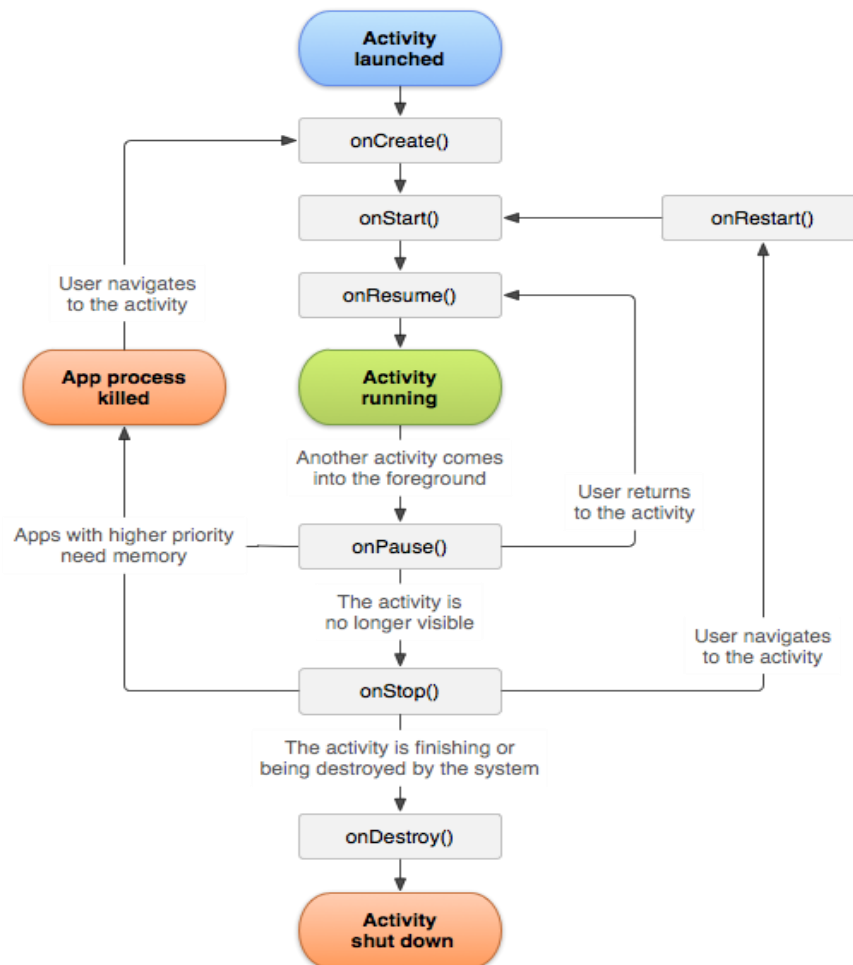
- **Actividades.** Similar a una pantalla o interfaz de usuario con la que el usuario puede interactuar, es el principal componente de las aplicaciones Android.
- **Servicios.** Permiten realizar ejecución en segundo plano y a diferencia de las Actividades, no proporcionan una interfaz de usuario.
- **Receptores de emisiones.** Se utilizan para recibir eventos por parte del sistema, por ejemplo, un Aviso Push, y al igual que las Actividades y los Servicios son iniciados a partir de un objeto Intent.
- **Proveedores de contenido.** Para compartir datos entre aplicaciones. Por ejemplo, información de contactos.
- **AndroidManifest.xml.** Fichero utilizado para declarar el nombre e icono de la aplicación, los componentes que la forman, las APIs o Bibliotecas utilizadas, los permisos del sistema utilizados, las versiones de Android con las que es compatible y las características hardware del dispositivo a las que se accede.

Para realizar un uso eficiente de los recursos del sistema operativo, las Actividades hacen uso de un ciclo de vida descrito a continuación. Los Servicios también disponen de un ciclo de vida similar, aunque diferente al carecer de Interfaz de Usuario.

Los eventos del ciclo de vida de una aplicación Android son los siguientes:

- **onCreate():** dónde el sistema de Android crea y lanza la aplicación.
- **onStart():** cuando el sistema ya está listo para comenzar a trabajar. Automáticamente se pasa a onResume().
- **onResume():** la aplicación ya es visible en primer plano. El usuario ya puede interactuar con la aplicación.
- **onPause():** se ejecuta cuando se produce cualquier interrupción. De onPause() se puede regresar a onResume(). La otra opción si está en pausa es “matar” la aplicación. Si esto pasa, la aplicación vuelve a crearse en el onCreate()).
- **onStop():** se ejecutará cuando la aplicación ya no se considere visible. Por otro lado, también se puede reanudar la actividad, pero lo hará desde el onStart(). Con onStop() se puede liberar recursos de una forma eficiente.
- **onRestart():** el flujo de la aplicación vuelve a reanudarse sin que existiese un problema grave.
- **onDestroy():** se ejecutará cuando el usuario abandone la aplicación de forma consciente.

En el siguiente diagrama se muestra el ciclo de vida de una aplicación Android.



Algunas **definiciones** que conviene tener claras son las siguientes:

- **La Máquina Virtual Dalvik (DVM):** permite ejecutar aplicaciones programadas en Java. La DVM no se definía como una máquina virtual de Java (JVM) por posibles problemas de licenciamiento, aunque cumplía ese propósito. Desde la versión Android 2.2 – Froyo, Dalvik utilizaba como compilador el compilador JIT (Just in time) cada vez que se iniciaba una aplicación.
- **Android Runtime (ART):** es el entorno de ejecución de aplicaciones utilizado por el sistema operativo móvil Android que reemplaza a Dalvik (la máquina virtual utilizada originalmente por Android), y lleva a cabo la transformación de la aplicación en instrucciones de máquina, que luego son ejecutadas por el entorno de ejecución nativo del dispositivo. ART introduce el uso del AOT (Ahead of time), que crea un archivo de compilación posterior a la instalación de la aplicación, evitando que ésta se compile continuamente cada vez que se ejecuta.

Desde la versión 4.4 (Kit Kat), ART se incorporó como opción en los sistemas Android, siendo la única posibilidad a partir de la versión 5.0 (Lollipop), en la cual Dalvik fue sustituida por ART.

- **Compilador JIT (Just-In-Time):** componente del entorno de ejecución que mejora el rendimiento de aplicaciones Java compilando códigos de bytes en código de máquina nativo en tiempo de ejecución. JIT fue incorporado a Android desde su versión 2.2 (Froyo).



- **Ahead-Of-Time (AOT):** es el acto de compilar un lenguaje de programación, como Java bytecode, a un código de máquina nativo con la intención de ejecutar el archivo binario resultante nativamente de manera más rápida y eficiente. AOT produce código máquina optimizado.

Cada actualización del sistema operativo Android se desarrolla bajo el nombre de un dulce, siguiendo el orden de las letras del alfabeto. Sin embargo, esta tendencia se abandonó a partir de Android 10. En la actualidad (abril de 2023), la última versión de Android es **Android 15**, publicada en septiembre de 2024.

Por último, el formato instalable y ejecutable de archivos para las aplicaciones de Android es “**apk**” (Android Application Package), que es una variante del formato JAR de Java, además las aplicaciones también pueden publicarse en formato “**aab**” (Android App Bundle).

5. Otros SSOO Móviles

- WINDOWS MOBILE (DESCONTINUADO)
- WINDOWS PHONE (DESCONTINUADO)
- WINDOWS 10 MOBILE (DESCONTINUADO)
- WINDOWS CORE O WINDOWS 10X (CANCELADO)
- FIREFOX OS (DESCONTINUADO)
- BLACKBERRY OS Y BLACKBERRY 10 (DESCONTINUADOS)
- Symbian (DESCONTINUADO)

6. z/OS

z/OS es el sistema operativo actual de las computadoras centrales (mainframes) de IBM. Del sistema MVT de 1967, se pasó al MVS en 1974 añadiéndole múltiples espacios de memoria virtual. Se agregó a éste compatibilidad UNIX y, en 1995, se evolucionó al OS/390. En el año 2000, se añadió direccionamiento de 64 bits y se lanzó el z/OS.

En definitiva, **z/OS** es el sistema operativo de **64 bits** para **mainframe IBM** con núcleo de tipo monolítico. Su última versión estable es **V3R1** (v3.1), de septiembre de 2023.

Las **características** de este sistema operativo son:

- **Fiabilidad:** los componentes hardware cuentan con amplias capacidades de autocomprobación y autorecuperación.
- **Alta disponibilidad y tolerancia a fallos:** el sistema puede recuperarse del fallo de un componente sin afectar al resto del sistema.
- Uso de tecnologías de **clúster** (Parallel Sysplex) que permiten ejecutar múltiples copias de un S.O. como si fuera un único sistema, proporcionando alta disponibilidad (99.999%).
- **Seguridad:** el sistema cuenta con protecciones a nivel hardware y software que ofrecen aislamiento entre cargas de trabajo, protección del almacenamiento y comunicaciones seguras.



- **Escalabilidad horizontal y vertical:** capacidad de añadir y eliminar recursos de computación dinámicamente y sin interrupción del servicio.
- **Compatibilidad total hacia atrás** con aplicaciones desarrolladas para versiones anteriores del sistema operativo.
- Ofrecen varios niveles de virtualización, permitiendo la ejecución de varios SO distintos:
 - **LPAR** (Logical Partition): subconjunto del hardware de procesamiento definido para soportar un sistema operativo.
 - **z/VM:** hypervisor que permite ejecutar diversos SO en máquinas virtuales.
- **Control centralizado** de los recursos.
- Se basan en un estilo de operación que involucra procesos detallados y estrictamente organizados para **tareas de backup**, recuperación de datos y recuperación ante desastres.
- Hardware y S.O. preparados para trabajar con **miles de operaciones** de I/O simultáneas

Modos de funcionamiento:

La mayoría de cargas de trabajo en un mainframe pertenece a una de las siguientes categorías:

- **Procesamiento batch:** los trabajos batch se procesan en el mainframe sin interacción con el usuario, procesando grandes cantidades de información y generando una salida con el resultado.
- **Procesamiento de transacciones en línea (OLTP):** el procesamiento de transacciones tiene lugar de forma interactiva con el usuario, siendo necesario un corto tiempo de respuesta por parte del sistema. Utilidades para trabajo interactivo en z/OS:
 - **TSO/E (Time Sharing Option/Extensions):** permite a los usuarios crear una sesión interactiva con el sistema z/OS, ofreciendo un interfaz básico de comandos. Es posible ejecutar una lista de comandos utilizando:
 - **CLIST (Command List):** contiene un conjunto de comandos que se ejecutarán en secuencia bajo TSO (similar a un script Unix).
 - **REXX:** lenguaje interpretado de comandos para TSO.
 - **ISPF (Interactive System Productivity Facility):** ofrece una colección de menús y paneles para ayudar a los usuarios en su interacción con el sistema

Z/OS y otros Sistemas Operativos del Mainframe IBM

Un mainframe IBM puede ejecutar múltiples sistemas operativos, siendo los más importantes:

- **z/VM (z/Virtual Machine):** se basa en dos componentes, un programa de control (CP) y un sistema operativo de usuario (CMS). z/VM es un hypervisor que ejecuta otros sistemas operativos en las máquinas virtuales que gestiona, como z/OS, Linux on System Z, z/VSE o z/TPF
- **z/VSE (z/Virtual Storage Extended):** proporciona un sistema más pequeño y sencillo para el procesamiento de transacciones y batch que el ofrecido por z/OS.
- **Linux on IBM System Z:** se refiere a una distribución de Linux que se ejecute sobre un mainframe IBM System Z.
- **z/TPF (z/Transaction Processing Facility):** sistema operativo de propósito específico para el procesamiento de un alto volumen de transacciones.

Data Sets

Definiciones:



- **Data set:** colección de registros de datos relacionados lógicamente. Corresponde al concepto de fichero.
- **Registro:** número fijo de bytes que contienen datos. Es la unidad básica de información usada por cualquier programa ejecutándose sobre z/OS.
- **Campo:** parte de un registro utilizada para almacenar una categoría particular de datos (ej: nombre, DNI).

z/OS incluye un data set denominado como **master catalog**, que permite acceder a cualquiera de los data sets en el sistema. La data sets en z/OS suelen codificarse en el juego de caracteres **EBCDIC**.

JCL (Job Control Language)

- **JCL:** lenguaje que informa a z/OS acerca de qué recursos son necesarios para procesar un trabajo batch o arrancar una tarea de sistema (started task).
- **PROC (Procedimiento JCL):** es un miembro de una librería de procedimientos que contiene parte de un JCL para una determinada tarea. Permite organizar JCLs que de otra manera tendrían un tamaño difícil de manejar.

En z/OS también es posible ejecutar una **versión de 64 bits de Java, C, C++** y soporta APIs de UNIX, y se comunica directamente con TCP/IP, incluyendo IPv6, e incluye servidores HTTP estándar (uno de Lotus, el otro Apache-derivado), junto con servicios tales como **FTP, NFS, CIFS/SMB**. El sistema operativo complementario **z/VM**, otorga la manipulación de varios sistemas operativos virtuales ("guests") en el mismo mainframe físico.

