

**TEMA:**

Kernels de los sistemas operativos

**ASIGNATURA:**

Sistemas operativos

**NRC 19364**

**DOCENTE:**

Juan Carlos Cantillo Bello

**PRESENTADO POR:**

Laura Carolina Zamora Guzmán - 661081

**INSTITUCION**

Corporación Universitaria Minuto de Dios

**Zipaquirá, Colombia                     22 de abril de 2021**

**Introducción**

El presente trabajo está orientado a realizar una investigación e informe sobre los diferentes kernels de los Sistemas Operativos (Windows, [Linux](https://202110.aulasuniminuto.edu.co/mod/resource/view.php?id=504061), Mac OS, UNIX, IBM, Android, etc), explicar cada uno de estos y mostrar sus diferentes diferencias, teniendo en cuenta lo explicado en clase y teniendo en cuenta los lineamientos establecidos por el docente. Este informe también tiene como fin fortalecer mis conocimientos, habilidades y formación educativa como futura ingeniera, en el aprendizaje de los diferentes kernels, ya que es de vital importancia que tengamos conocimientos de estos, además de generarnos más experiencia a la hora de poner en práctica nuestros conocimientos un espacio laboral.

**Kernels**

El Kern el o núcleo del sistema operativo está alojado dentro del SO, no forma parte del hardware y es el mediador o intermediario entre el software y el hardware, es una parte fundamental del sistema operativo que se encarga de conceder el acceso al hardware de forma segura para todo el software que lo solicita, el sistema operativo envía una serie de órdenes que son recibidas por el kernel quien a su vez las envía al componente de hardware que corresponda con el fin que sean ejecutadas. El kernel es una optimización de las funcionalidades básicas del sistema operativo y se encarga de darle su espacio y su momento a cada uno de los diferentes tipos de interrupciones del sistema.

El kernel tiene varias funciones dentro del sistema operativo , en primer lugar permite el acceso a los diferentes periféricos o accesorios que se conectan al equipo para que sean utilizados según lo requiera el usuario, también se encarga de administrar la memoria para que se utilice de manera eficiente por los distintos programas, además es el responsable de gestionar el tiempo de procesador que utilizan los programas y procesos ejecutados dentro del equipo, un kernel mal diseñado puede filtrar información sobre el usuario y sus aplicaciones o sobrecalentar en exceso un equipo.

**Kernel del sistema operativo Windows**

En la década de los noventa Microsoft estaba basando sus sistemas operativos en los kernel Windows 9x, donde el código básico tenía muchas similitudes con MS-DOS. De hecho necesitaba recurrir a él para poder operar. Paralelamente, Microsoft también estaba desarrollando otra versión de su sistema dirigido a los servidores llamada Windows NT, que nació [el 26 de julio de 1993](http://www.theregister.co.uk/2013/08/01/windows_nt_anniversary/). Ambas versiones de Windows fueron desarrollándose por separado. Windows NT era más bien una jugada a largo plazo, una tecnología que ir desarrollando para los Windows del mañana, y en el año 2000 dieron un nuevo paso en esa dirección. A la versión 5.0 de NT la llamaron Windows 2000, y se convirtió en un interesante participante en el sector empresarial. Tras ver la buena acogida que tuvo, Microsoft decidió llevar NT al resto de usuarios para que ambas ramificaciones convergieran. Lo hicieron en octubre del 2001 con la versión 5.1 de Windows NT, que llegó al mercado con el nombre de Windows XP. Por lo tanto, esta versión marcó un antes y un después no sólo por su gran impacto en el mercado, sino porque era el principio de la aventura del Kernel Windows NT en el mundo de los usuarios de a pie. Desde ese día, todas las versiones de Windows han estado basadas en este Kernel con más de 20 añós de edad. La versión 5.1.2600 fue Windows XP, la 6.0.6002 fue Windows Vista, y la 6.1.7601 Windows 7. Antes hubo otros Windows Server 2008 y 2003, y después llegaron las versiones de NT 6.2.9200 llamada Windows 8, la 6.3.9600 o Windows 8, y finalmente la NT 10.0, también conocida como Windows 10. *Yubal. (11 enero 2017). Cómo es el kernel de Windows y cuales son sus diferencias con el de Linux. GENBETA . https://www.genbeta.com/a-fondo/como-es-el-kernel-de-windows-y-cuales-son-sus-diferencias-con-el-de-linux*

La principal característica del Kernel de Windows NT es que es bastante modular, y está basada en dos capas principales, la de usuario y la de kernel. El sistema utiliza cada una para diferentes tipos de programa. Por ejemplo, las aplicaciones se ejecutan en el modo usuario, y los componentes principales del sistema operativo en el modo kernel. Mientras, la mayoría de los drivers suelen usar el modo kernel, aunque con excepciones. Es por eso que se refieren a él como Kernel híbrido, pero sobre todo también porque permite tener subsistemas en el espacio del usuario que se comunicaban con el kernel a través de un mecanismo de IPC. Cuando ejecutas una aplicación, esta accede al modo usuario, donde Windows crea un proceso específico para la aplicación. Cada aplicación tiene su dirección virtual privada, ninguna puede alterar los datos que pertenecen a otra y tampoco acceder al espacio virtual del propio sistema operativo. Es por lo tanto el modo que menos privilegios otorga, incluso el acceso al hardware está limitado, y para pedir los servicios del sistema las aplicaciones tienen que recurrir a la API de Windows.

El modo núcleo en cambio es ese en el que el código que se ejecuta en él tiene acceso directo a todo el hardware y toda la memoria del equipo. Aquí todo el código comparte un mismo espacio virtual, y puede incluso acceder a los espacios de dirección de todos los procesos del modo usuario. Esto es peligroso, ya que si un driver en el modo kernel toca lo que no debe podría afectar al funcionamiento de todo el sistema operativo. Este modo núcleo está formado por servicios executive, como el controlador de caché, el gestor de comunicación, gestor de E/S, las llamadas de procedimientos locales, o los gestores de energía y memoria [entre otros](https://es.wikipedia.org/wiki/Arquitectura_de_Windows_NT). Estos a su vez están formados por varios módulos que realizan tareas específicas, controladores de núcleo, un núcleo y una Capa de Abstracción del Hardware o HAL. Como dije anteriormente Windows controla el software del sistema de bajo nivel y las interacciones con el hardware del ordenador a través de la capa de abstracción de hardware (HAL). El HAL es un elemento del sistema que funciona como interfaz entre software y hardware, y como las API, permite que las aplicaciones sean independientes del hardware.

* El kernel de Windows es un kernel comercial desarrollado por Microsoft.
* Al ser un software comercial, no hay acceso al código fuente de Windows.
* Tiene una arquitectura híbrida.
* Usa una Lista de control de acceso para los protocolos de control de acceso a archivos.
* Incluye una pila de GUI en el kernel.
* Puede ofrecer soporte para múltiples usuarios y múltiples sesiones, pero depende de la versión y edición de su sistema operativo Windows.
* Mantiene un registro para almacenar sus configuraciones.
* Dependiendo de qué dispositivo esté ejecutando el kernel de Windows, se comportará de manera diferente debido a un mecanismo variable.
* Controla el software del sistema de bajo nivel y las interacciones con el hardware del ordenador a través de la capa de abstracción de hardware (HAL).
* El HAL es un elemento del sistema que funciona como interfaz entre software y hardware, y como las API, permite que las aplicaciones sean independientes del hardware.

**Kernel del sistema operativo Linux**

Linux se refiere al núcleo del sistema operativo GNU/Linux. En términos más generales, se refiere a la familia de distribuciones derivadas. Linux es solo el Kernel, solo las distribuciones Linux se consideran como sistemas operativos. Si lo comparamos con UNIX, este si es un sistema operativo completo pues todo proviene del mismo desarrollador. En resumen, una distribución Linux es un sistema operativo creado sobre el kernel de Linux al que se adjuntan varias aplicaciones entre las que se encuentra el sistema de administración de paquetes. Una distribución estándar Linux consta de un kernel Linux, un sistema GNU, utilidades GNU, bibliotecas, compiladores, software adicional, documentación, un sistema de ventanas, un administrador de ventanas y un entorno de escritorio. Normalmente todo es gratuito y de [código abierto](https://www.sololinux.es/alternativas-a-photoshop-de-codigo-abierto/).

El lanzamiento del primer kernel de linux fue en 1991 creado por linus torvalds y a partir de ahí ha continuado su evolución con la ayuda de miles de desarrolladores de todo el mundo ya que es uno de los proyectos más importantes de código abierto, las compañías de Google y IBM ayudan en su desarrollo con aportaciones económicas.

* El kernel de Linux es de código abierto basado en Unix.
* Todos tienen acceso al código fuente de Linux, cualquiera puede trabajar en su desarrollo y es de uso gratuito, algo que le sirve para estar presente en múltiples sistemas operativos o distros GNU/Linux.
* En Linux tomaron los principios de modularidad de Unix y decidieron abrir el código y las discusiones técnicas. Gracias a ello, Linux ha creado [una comunidad meritocrática](https://gallir.wordpress.com/2013/05/11/por-que-el-nucleo-windows-nt-es-peor-que-linux-problemas-sociales-y-de-incentivos/) de desarrolladores, una en la que todos pueden colaborar y en la que cada cambio que se sugiere se debate con dureza para desechar las peores ideas y quedarse con las mejores.
* Tiene una arquitectura monolítica.
* Usa los permisos tradicionales de Unix y POSIX ACL para el control de acceso a archivos.
* Mantiene la pila de la GUI en el espacio del usuario.
* Ofrece entornos 100% multiusuario.
* Elige mantener sus configuraciones en archivos.
* Se comportará igual en todos los dispositivos.
* HAL está descontinuado en sistemas GNU/Linux, y su funcionalidad se ha incorporado a udev.
* El kernel de macOS (XNU) ha existido más tiempo que Linux y se basó en una combinación de dos bases de código aún más antiguas. Por otro lado, Linux es más nuevo, escrito desde cero, y se utiliza en muchos más dispositivos.

**Kernel del sistema operativo Mac OS**

El kernel de macOS se conoce oficialmente como XNU, XNU es un [núcleo](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) o [kernel](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) desarrollado originalmente por [NeXT](https://es.wikipedia.org/wiki/NeXT" \o "NeXT) e implementado por [Apple Inc.](https://es.wikipedia.org/wiki/Apple) en 1996 en su [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) [macOS](https://es.wikipedia.org/wiki/MacOS). El acrónimo significa "XNU no es Unix". Según la página oficial de Github de Apple, XNU es "un kernel híbrido que combina el kernel Mach desarrollado en la Universidad Carnegie Mellon con componentes FreeBSD y C++ para los controladores". Fue liberado como [software libre y de código abierto](https://es.wikipedia.org/wiki/Software_libre_y_de_c%C3%B3digo_abierto) como parte del sistema operativo [Darwin](https://es.wikipedia.org/wiki/Darwin_(sistema_operativo)). Es un núcleo híbrido, lo que significa que su arquitectura de diseño es una combinación de arquitecturas basadas en [núcleo monolítico](https://es.wikipedia.org/wiki/N%C3%BAcleo_monol%C3%ADtico) y [micronúcleo](https://es.wikipedia.org/wiki/Micron%C3%BAcleo). Está basado en [Mach](https://es.wikipedia.org/wiki/Mach_(n%C3%BAcleo)) 3.0 (desarrollado en la [Universidad Carnegie Mellon](https://es.wikipedia.org/wiki/Universidad_Carnegie_Mellon) en 1980) y el código [BSD](https://es.wikipedia.org/wiki/Berkeley_Software_Distribution), en especial de [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD) 5.x y usa [I/O Kit](https://es.wikipedia.org/w/index.php?title=I/O_Kit&action=edit&redlink=1), un [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) de controlador de dispositivos orientado a objetos que hace de XNU un núcleo diferente al de los sistemas tradicionales.

**Mach**

El corazón de XNU, [Mach](https://es.wikipedia.org/wiki/Mach_(n%C3%BAcleo)), fue originalmente concebido como un micronúcleo. Como tal, es capaz de hacer funcionar la base de un [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) como procesos separados, lo que permite una mayor flexibilidad (se podrían ejecutar varios sistemas operativos en paralelo sobre la base de Mach), pero a menudo reduce el rendimiento debido al tiempo consumido para cambiar del contexto núcleo al contexto usuario, y al trabajo que se deriva de mapear o copiar mensajes entre el espacio de direcciones del micronúcleo y el de los [demonios](https://es.wikipedia.org/wiki/Daemon_(computer_software)) de servicios. Con Mac OS X, los diseñadores han intentado hacer más sencillas algunas tareas, y las funcionalidades de BSD se han implementado en espacio privilegiado junto con Mach. El resultado es una combinación de Mach y un clásico núcleo BSD, con algunas ventajas y desventajas de ambos. [GNU Hurd](https://es.wikipedia.org/wiki/GNU_Hurd) también hace uso de Mach, sin embargo toda la funcionalidad aparte de Mach que XNU implementa en espacio privilegiado, Hurd lo hace en espacio de usuario (los servidores o [daemon|demonios]) junto con la biblioteca de [tiempo de ejecución](https://es.wikipedia.org/wiki/Tiempo_de_ejecuci%C3%B3n) ([la biblioteca del lenguaje C](https://es.wikipedia.org/w/index.php?title=Biblioteca_de_C&action=edit&redlink=1)). GNU Hurd es fiel a un diseño de micronúcleo. *XNU. Wikinzer. https://www.wikizero. com/es/XNU*

La parte del subsistema BSD del código "normalmente se implementa como servidores de espacio de usuario en sistemas microkernel". La parte Mach es responsable de trabajos de bajo nivel como multitarea, memoria protegida, administración de memoria virtual, soporte de depuración de kernel e E/S de consola. El corazón de todo se llama Darwin; y dentro de él, tenemos utilidades de sistema separadas y el kernel XNU, que está compuesto en partes por el kernel Mach y por el kernel BSD y a diferencia de Linux, este kernel se divide en lo que llaman el kernel híbrido, lo que permite que una parte de ella se detenga para el mantenimiento, mientras que otra continúa funcionando. En varios debates esto también abrió la cuestión del hecho de que un núcleo híbrido es más estable; si una de sus partes se detiene, la otra puede volver a iniciarla.

Mientras que el kernel de macOS combina las capacidades de un microkernel con Mach y un kernel monolítico como BSD, Linux es sólo un kernel monolítico. Un kernel monolítico es responsable de administrar cpu, memoria, comunicación entre procesos, controladores de dispositivos, sistema de archivos y llamadas de servicio del sistema. Es decir, lo hace todo sin subdivisiones. Obviamente, esto ya ha generado mucha discusión incluso con el propio Linus y otros desarrolladores, que afirman que un núcleo monolítico es más susceptible a los errores además de ser más  
lento; pero Linux es lo opuesto a esto cada año, y se puede optimizar como un kernel híbrido. Además, con la ayuda de RedHat,el kernel ahora incluye un live patch que permite el mantenimiento en tiempo real sin necesidad de reiniciar.

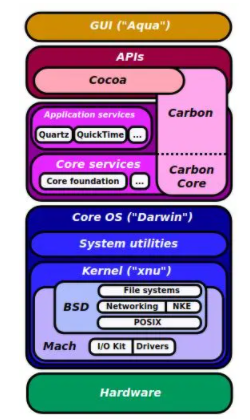


Ilustración 1. Mapa de MacOS

* MacOS tiene un núcleo híbrido.
* El kernel de macOS se conoce oficialmente como XNU
* El acrónimo significa "XNU no es Unix"
* Un kernel que fue liberado como software libre y de código abierto como parte del SO Darwin
* El kernel de macOS combina las capacidades de un microkernel con Mach y un kernel monolítico como BSD.
* El kernel de macOS (XNU) ha existido durante más tiempo que Linux y se basó en una combinación de dos bases de código aún más antiguas. Esto pesa a favor, para la estabilidad y la historia.
* El kernel de macOS (XNU) ha existido más tiempo que Linux y se basó en una combinación de dos bases de código aún más antiguas. Por otro lado, Linux es más nuevo, escrito desde cero, y se utiliza en muchos más dispositivos.

**Kernel del sistema operativo UNIX**

UNIX es un sistema operativo de tiempo compartido (el sistema UNIX divide el tiempo de la computadora en un numero de partes, repartiéndolas entre los diferentes procesos). El kernel del sistema es un programa que siempre esta residente en memoria y entre otros, brinda los siguientes servicios:

* Controla los recursos básicos.
* Controla los dispositivos periféricos (discos, terminales, impresoras, etc).
* Permite a distintos usuarios compartir recursos y ejecutar sus programas.
* Proporcionar un sistema de archivos que administra el almacenamiento de información (programas, datos, documentos, etc.)
* Es un sistema interactivo, permite el redireccionamiento de la E/S, tuberías y ejecución de procesos en background.

El kernel del sistema, esta escrito en C en su mayor parte, aunque coexiste con código ensamblador. El kernel del sistema operativo es un programa que esta siempre en memoria y brinda servicios para controlar los recursos del sistema. UNIX ofrece al usuario dos entidades: proceso y archivo. Los dos conceptos centrales sobre los que se basa la arquitectura de UNIX son los procesos y los archivos. El kernel esta pensado para facilitar servicios relacionados con el sistema de archivos y con el control de procesos.

El [Kernel de Unix](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) tiene el control total sobre el sistema. Tiene otros subsistemas que manejan los servicios como el sistema de archivos, los recursos, la administración de memoria, aplicaciones de inicio y otras tareas básicas de bajo nivel. El núcleo es el corazón del sistema operativo, y ejerce como si fuera una interfaz entre el usuario y el [hardware](https://www.sololinux.es/ver-las-temperaturas-del-hardware-consola/). Cada subsistema del kernel tiene sus propias características como concurrencia, memoria virtual, paginación y su sistema de archivos virtual. En las capas externas de la arquitectura Unix, nos encontramos con la shell, sus comandos y herramientas. La [shell](https://en.wikipedia.org/wiki/Unix_shell) (mal llamada terminal o consola) es la interfaz entre el usuario y el núcleo; cuando el usuario introduce comandos, estos son interpretados y llaman a las herramientas o aplicaciones que corresponda.

El Kernel es el componente central de un sistema operativo (OS) Unix que puede controlar todo dentro del sistema operativo. Es el primer programa que se carga al iniciar un sistema operativo Unix. Una vez cargado, gestionará el resto del proceso de inicio, gestiona la memoria, la programación, así como la comunicación con periféricos como impresoras, teclados, etc. Pero Kernel no interactúa directamente con un usuario. Para una nueva tarea, Kernel generará un shell y el usuario funcionará en un shell. Los términos Kernel y Shell tienden a relacionarse equivocadamente tan solo con los sistemas operativos Linux. El Shell, o intérprete de órdenes, son aplicaciones capaces de interpretar las órdenes del usuario a través de comandos escritos, como por ejemplo el sistema MS-DOS o los terminales de consola de los sistemas operativos Linux. Estas aplicaciones permiten al usuario interactuar con el ordenador, normalmente a través de una sencilla interfaz de texto plano, y suponen la forma más básica de interacción de un usuario con su ordenador, escribiendo las órdenes en este Shell a través de comandos y recogiendo las respuestas de la máquina. *Una visión general de los kernels de Unix. Oreilly. https://www.oreilly.com/library/view/understanding-the-linux/0596002130/ch01s06.html*

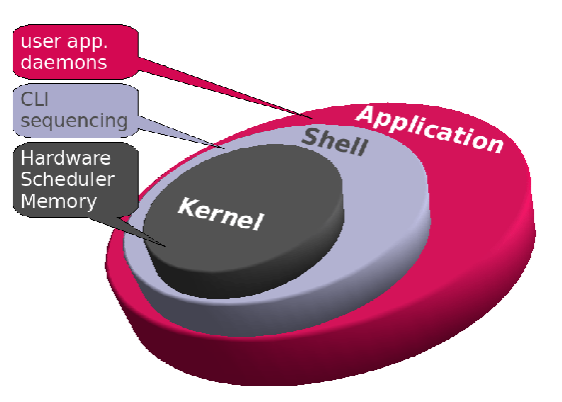


Ilustración 2. Shell

* El núcleo de Unix puede ser monolítico, microkernel o híbrido, depende del sistema elegido.
* El kernel Unix necesita la vinculación estática de los nuevos sistemas que se agregan.
* En Unix se incluye el subsistema E/S de secuencias como interfaz para escribir controladores de dispositivo, controladores de terminal, etc.
* Solo algunos son preventivos, como por ejemplo el de Solaris.
* En Unix, el LWP se basa en hilos del kernel.
* Los hilos se utilizan para cambiar los procesos.
* Es un kernel que está escrito en C en su mayor parte
* Es un kernel pensado para facilitar servicios relacionados con el sistema de archivos y con el control de procesos
* Brinda una serie de servicios como Controlar los recursos básicos, controlar los dispositivos periféricos (discos, terminales, impresoras, etc.), permitir a distintos usuarios compartir recursos y ejecutar sus programas, proporcionar un sistema de archivos que administra el almacenamiento de información (programas, datos, documentos, etc.), permite el redireccionamiento de la E/S, tuberías y ejecución de procesos en background.

**IBM**

El kernel de Linux todavía era una opción valiente cuando IBM se unió a la comunidad Linux a finales de la década de 1990. IBM comenzó su trabajo de linux-kernel con un puerto skunkworks de Linux al mainframe de IBM y un compromiso corporativo que resultó en la inversión de IBM [$ 1B](https://news.slashdot.org/story/13/09/17/1513227/ibm-promises-1b-investment-in-linux-development) en Linux en 2001. Linux fue portado a todos los servidores de IBM, e incluso al [reloj de pulsera impulsado por Linux de](https://researcher.watson.ibm.com/researcher/view_group.php?id=6101)IBM Research. Linux pronto disfrutó de un uso generalizado dentro del hardware, software y servicios de IBM. Por supuesto, IBM no fue el único jugador que apostó por Linux. Por ejemplo, un equipo de ventas de IBM pasó mucho tiempo preparándose para convencer a un cliente de larga data, técnicamente conservador, de empezar a moverse hacia Linux. Cuando el equipo fue a dar su lanzamiento, el cliente abrió la discusión con: "Hemos decidido que vamos con Linux. *McKenney. (29 julio 2019). IBM y el kernel de Linux. Developer IBM. https://developer.ibm.com/technologies/linux/blogs/ibm-and-the-linux- kernel*

El punto de inflexión fue el lanzamiento del primer kernel de Linux, creado por Linus Torvalds en 1991. A partir de ahí ha continuado su evolución con la ayuda de miles de desarrolladores de todo el mundo, ya que es uno de los proyectos más importantes de código abierto. Tanto es así que compañías de la talla de Google o IBM ayudan en su desarrollo con aportaciones económicas. El kernel de Linux todavía era una opción valiente cuando IBM se unió a la comunidad Linux a finales de la década de 1990. Linux fue portado a todos los servidores de IBM, e incluso al [reloj de pulsera impulsado por Linux de](https://researcher.watson.ibm.com/researcher/view_group.php?id=6101) IBM Research. Linux pronto disfrutó de un uso generalizado dentro del hardware, software y servicios de IBM. Por supuesto, IBM no fue el único jugador que apostó por Linux. Por ejemplo, un equipo de ventas de IBM pasó mucho tiempo preparándose para convencer a un cliente de larga data, técnicamente conservador, de empezar a moverse hacia Linux. Cuando el equipo fue a dar su lanzamiento, el cliente abrió la discusión con: "Hemos decidido que vamos con Linux. ¿Vendrás con nosotros?" Aunque esto destruyó horas incalculables de preparación, produjo un resultado más allá de la imaginación más salvaje del equipo de ventas.

Este entusiasmo generalizado motivó a IBM no sólo a hacer contribuciones sustanciales a Linux, sino también a venir a su defensa. IBM también es un fuerte contribuyente técnico al kernel de Linux, ocupando el [top ten de colaboradores corporativos](https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/) y teniendo mantenedores para una amplia gama de subsistemas linux-kernel. Por supuesto, IBM contribuye en gran medida a apoyar sus propias ofertas, pero también es un fuerte contribuyente en las áreas de escalabilidad, robustez, seguridad y otras áreas que benefician al ecosistema Linux.Por supuesto, no todo lo que IBM intentó funcionó. El trabajo de escalabilidad de IBM en el programador nunca fue aceptado en el kernel de Linux. Aunque su sistema de archivos de registro en diario (JFS) fue aceptado y sigue siendo parte del kernel de Linux, parece seguro decir que JFS nunca alcanzó el nivel de popularidad que IBM esperaba. Sin embargo, parece probable que los esfuerzos de IBM ayudaran a inspirar el trabajo que condujo a la excelente escalabilidad, características y funcionalidad del kernel de Linux en sus sistemas de archivos y programador. *McKenney. (29 julio 2019). IBM y el kernel de Linux. Developer IBM. https://developer.ibm.com/technologies/linux/blogs/ibm-and-the-linux- kernel*

Además, estas experiencias enseñaron a IBM a trabajar más estrechamente con la comunidad, allanando el camino a contribuciones sustanciales posteriores. Un ejemplo es la característica de grupos de CPU del programador de la comunidad que ahora sustenta tecnologías de contenedores como Docker, junto con la característica virtio que se reproduce en la virtualización basada en hipervisores más tradicional. Otro ejemplo son numerosas mejoras que conducen al sistema de archivos EXT4 de la comunidad. Un ejemplo final es la función de especificación de hardware del árbol de dispositivos, desarrollada originalmente para los servidores Power de IBM, pero ahora también utilizada por muchos sistemas Linux integrados.

**Kernel del sistema operativo Android**

Android es un sistema operativo móvil proporcionado por Google. Se basa en la versión modificada del kernel de Linux y otro software de código abierto. Está diseñado específicamente para dispositivos móviles con pantalla táctil como teléfonos inteligentes y tabletas. Fue desarrollado utilizando C, Java, C++ y otros idiomas. La primera versión de Android fue lanzada por Google en 2008.  Se proporciona totalmente gratuito. Es el sistema operativo más utilizado en general.

El kernel en el dispositivo es el mediador entre el apartado de software de tu teléfono y el apartado del hardware, por ejemplo cada vez que abrimos la cámara, whatsapp, la linterna, cada vez que ejecutamos cualquier aplicación el software le va a mandar una orden al kernel y el kernel al hardware para que ejecute cualquier proceso, cualquier acción que hayamos seleccionado, entonces el kernel en general es el mediador entre el apartado del software de nuestro dispositivo y el apartado del hardware.

Android utiliza un kernel de Linux modificado que proporciona a los desarrolladores toneladas de características pre-construidas y bien mantenidas. Esto, a su vez, ahorra tiempo y permite a los desarrolladores de Android centrarse en las características más importantes para los dispositivos móviles. El aspecto de Android que es Linux es su kernel monolítico, que es una versión modificada del kernel de Linux y viene con sus propias bibliotecas y API. Las modificaciones en el kernel de Linux son principalmente para los requisitos de potencia especializados de dispositivos portátiles con baterías algo pequeñas y las bibliotecas especializadas y API son para interactuar con la parte de comunicaciones celulares de un teléfono celular, así como para implementar una programación común e interfaz de usuario para el soporte de aplicaciones para que no tengan que ser específicos de la línea de modelo de teléfono de un fabricante en particular.

Otro detalle relevante es que cuando el sistema operativo Linux fue desarrollado en 1991 como un sistema operativo de código abierto para computadoras de escritorio por Linus Torvalds, fue desarrollado como el sistema operativo MINIX y no era compatible con características de 32 bits con máquinas Intel 80386. Una diferencia significativa entre los dos en ese momento es que la arquitectura de Android soportaba sólo x86 y ARM que emplea para dispositivos móviles de Internet (MIDs) y teléfonos móviles.

Android es un sistema operativo de código abierto que fue creado por primera vez por Android Inc. y ahora es propiedad de Google, mientras que Linux es el núcleo que fue creado por Linus Torvalds (y contribuido por muchos otros) bajo el proyecto GNU. Está desarrollado para teléfonos y dispositivos móviles de Internet, mientras que Linux está desarrollado para computadoras de escritorio / portátiles / servidores. Utiliza su propia biblioteca C, Bionic; mientras que las distros linux utilizan GNUC. No puede ejecutar ningún programa destinado a distros GNU/Linux. Implementa una versión modificada del kernel de Linux con bibliotecas especializadas y API para interactuar con redes celulares y proporcionar soporte para aplicaciones que pueden ser utilizadas por modelos de teléfono independientemente de sus proveedores.

A partir de Android 11, los cernes comunes de Android se utilizan para crear imágenes de kernel genéricas (GKI), que son imágenes de kernel Aarch64 que se pueden usar para ejecutar cualquier dispositivo con SoC y compatibilidad con controladores implementados en módulos de proveedores. Para obtener más información, consulte la [descripción general de GKI](https://source.android.com/devices/architecture/kernel/generic-kernel-image). GKI introduce el concepto de una interfaz de módulo de núcleo (KMI) estable que permite que el núcleo del núcleo se actualice de forma asincrónica desde los módulos del proveedor. Cuando la KMI está congelada, no se pueden realizar cambios que rompan la compatibilidad binaria con los módulos de proveedores existentes. Consulte la [descripción general de GKI](https://source.android.com/devices/architecture/kernel/generic-kernel-image) para obtener detalles sobre el KMI. *Núcleos comunes de Android. Source-android. https://source.android.com/devices/architecture/kernel/android-common*

Algunos de los kernels personalizados más populares que utiliza Android que no sólo ofrecen una mayor duración de la batería, rendimiento, sino que también son famosos por su estabilidad y seguridad entre los usuarios y ahora son una opción ideal para kernels personalizados son :

**ElementalX**

ElementalX es uno de los kernels personalizados más famosos desarrollado por XDA Recognized Developer [flar2.](https://forum.xda-developers.com/member.php?u=4684315) No sólo ofrece una mayor duración de la batería, sino que tampoco compensa el rendimiento excesivo. Como el propio flar2 afirma en una de las entrevistas dadas a XDA. Puede leer la entrevista completa [aquí.](https://www.xda-developers.com/flar2-elementalx-interview-part-1/) El objetivo de ElementalX siempre ha sido agregar características de hardware útiles, sin comprometer la estabilidad o el rendimiento, y liberar algo de lo que las personas puedan depender para simplemente trabajar. ElementalX está disponible para varios dispositivos Android. Dirígete a su [sitio oficial](https://elementalx.org/devices/) para comprobar si está disponible para tu dispositivo o no. Los hilos de soporte oficial también están disponibles en foros relacionados con dispositivos en XDA. También ofrece su propia aplicación para personalizar su kernel que se puede instalar desde Play Store.

**Núcleo franquista**

Franco Kernel desarrollado por XDA Recognized Developer [franciscofranco](https://forum.xda-developers.com/member.php?u=3292224) es uno de los núcleos más famosos por su estabilidad y eficiencia, también está disponible para una serie de dispositivos con una gran cantidad de optimizaciones, características mágicas y más. Franco Kernel no sólo ofrece una gran duración de la batería y mejoras de rendimiento, sino también el soporte activo del desarrollador, es decir, Francisco Franco en su dispositivo XDA thread también. Franco Kernel también ofrece su propia aplicación oficial para todos los dispositivos que tiene soporte para personalizar, parpadear y actualizar el kernel desde la propia aplicación.

**blu\_spark**

blu\_spark kernel desarrollado por XDA Senior Member [eng.stk](https://forum.xda-developers.com/member.php?u=3873953) no sólo ofrece una gran duración de la batería, sino también una plétora de personalizaciones para ajustar el rendimiento del dispositivo al máximo. Por supuesto, ya puede asumir que hay actualizaciones de seguridad, así como hilos de soporte activo en XDA. Algunas de las características que son comunes en varios dispositivos soportados por blu\_spark son el rendimiento mejorado, parches de batería, parches de seguridad CVE, correcciones generales ascendentes y CAF, mientras que el soporte para varios ajustes, hotplugs, gobernadores también.

**Kernel sultán**

El kernel personalizado del desarrollador reconocido de XDA [Sultanxda,](https://forum.xda-developers.com/member.php?u=4800121) es decir, Sultan Kernel, tiene como objetivo ser un simple reemplazo para almacenar kernel mientras que tiene características simples y útiles que un usuario diario necesitaría. Ya sea CPU Boost que un jugador necesitaría o SafetyNet para que usted pueda utilizar sus aplicaciones bancarias, Sultan's Kernel lo cubrió. También se basa en la teoría flash y olvidado y no se supone que sea personalizado por los usuarios como se indica en su [hilo XDA](https://forum.xda-developers.com/pixel-2/development/kernel-sultan-kernel-t3802158) para el Pixel 2 de Google.

Nota: Sultanxda ha descontinuado el soporte para dispositivos antiguos.

**Arter97 Kernel**

Arter97 kernel de XDA Recognized Developer [arter97](https://forum.xda-developers.com/member.php?u=4898097) está disponible para un montón de dispositivos que permiten que sea nuevo o viejo con una serie de optimizaciones y características. Al igual que el kernel de arter97 de todos los demás kernels, no solo se ha fusionado caf y subversiones más recientes, sino que también tiene una instalación sin sistema para dispositivos compatibles con optimizaciones tomadas de Pixel2/3. También tiene soporte para el último sistema de archivos f2fs que es utilizado por una gran cantidad de usuarios. Tiene una base de usuarios activa, así como el propio arter97 proporcionando soporte y explicaciones a la mayoría de sus dudas sobre sus hilos XDA.

Nota: arter97 ha descontinuado el soporte para dispositivos antiguos.

*Kernels personalizados más populares para Android. XDA. https://www.xda-developers.com/most-popular-custom-kernels-for-android/*



Ilustración 3. Arquitectura Android

* Android se basa en Linux para servicios principales del sistema, como seguridad, administración de memoria, administración de procesos, pila de red y modelo de controlador. El kernel también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.
* Es un kernel de Linux modificado que proporciona a los desarrolladores toneladas de características preconstruidas y bien mantenidas.
* Es kernel monolítico
* Android no es una distro GNU/Linux, pero su kernel (su potencia) es Linux y ya que es la convención que una colección de software reunido alrededor del kernel de Linux se conoce como Linux, Android es Linux.
* Puede pasar por la sección de información “Acerca de “en su configuración para comprobar la versión del kernel que su teléfono está ejecutando.
* A partir de Android 11, los kernels comunes de Android se utilizan para crear imágenes de kernel genéricas (GKI).
* Andriod utiliza algunos kernels personalizados y algunos de los más populares como : ElementalX, Franco, blue\_spark, Sultán y Arter97.

**Mapa conceptual- Kernels de los sistemas operativos**

[*https://lucid.app/lucidchart/invitations/accept/inv\_0a2091dd-a6bb-47f0-a0eb-e605ebc05e27?viewport\_loc=429%2C-2555%2C13199%2C6171%2C0\_0*](https://lucid.app/lucidchart/invitations/accept/inv_0a2091dd-a6bb-47f0-a0eb-e605ebc05e27?viewport_loc=429%2C-2555%2C13199%2C6171%2C0_0)

**Bibliografía**

* *Yubal. (11 enero 2017). Cómo es el kernel de Windows y cuales son sus diferencias con el de Linux. GENBETA* ***.*** [***https://www.genbeta.com/a-fondo/como-es-el-kernel-de-windows-y-cuales-son-sus-diferencias-con-el-de-linux***](https://www.genbeta.com/a-fondo/como-es-el-kernel-de-windows-y-cuales-son-sus-diferencias-con-el-de-linux)
* *XNU. Wikinzer.* ***https://www.wikizero. com/es/XNU***
* *McKenney. (29 julio 2019). IBM y el kernel de Linux. Developer IBM.* ***https://developer.ibm.com/technologies/linux/blogs/ibm-and-the-linux- kernel***
* *Kernels personalizados más populares para Android. XDA.* ***https://www.xda-developers.com/most-popular-custom-kernels-for-android/***

**Imágenes**

* *Kernerl de Linux vs Mac Kernel.**Linuxandubuntu.* [***https://www.linuxandubuntu.com/home/difference-between-linux-kernel-mac-kernel#:~:text=The%20macOS%20kernel%20is%20officially%20known%20as%20XNU.,with%20FreeBSD%20and%20C%2B%2B%20components%20for%20the%20drivers***](https://www.linuxandubuntu.com/home/difference-between-linux-kernel-mac-kernel#:~:text=The%20macOS%20kernel%20is%20officially%20known%20as%20XNU.,with%20FreeBSD%20and%20C%2B%2B%20components%20for%20the%20drivers)***.”***
* ¿Qué es Kernel en es sistema operativo Unix?. Quora. [***https://www.quora.com/What-is-Kernel-in-Unix-operating-system***](https://www.quora.com/What-is-Kernel-in-Unix-operating-system)
* *González. (21 julio 2014). ¿Qué significa que Android esté basado en Linux?. Blogthinkbig.*[***https://blogthinkbig.com/android-esta-basado-en-linux***](https://blogthinkbig.com/android-esta-basado-en-linux)