

# El Sistema Operativo: Una Historia

Nicolás Andrés Ortega Froysa

03/11/2021

# 1. Hoja De Control Del Documento

Cuadro 1: Documento/Archivo

|                                  |            |                         |        |
|----------------------------------|------------|-------------------------|--------|
| <b>Fecha Última Modificación</b> | 03/11/2021 | <b>Versión/Revisión</b> | v01r01 |
| <b>Fecha Creación</b>            | 13/10/2021 |                         |        |
| <b>Fecha Finalización</b>        | 03/11/2021 |                         |        |

Cuadro 2: Registro De Cambios

| <b>Versión/Revisión</b> | <b>Página</b> | <b>Descripción</b>        |
|-------------------------|---------------|---------------------------|
| v01r01                  | Todas         | Compleción del documento. |

Cuadro 3: Autores Del Documento

| <b>Apellidos, Nombre</b>   | <b>Curso</b> |
|----------------------------|--------------|
| Ortega Froyosa, Nicolás A. | 1            |

| <b>Preparado</b>           | <b>Revisado</b> | <b>Aprobado</b> |
|----------------------------|-----------------|-----------------|
| Ortega Froyosa, Nicolás A. |                 |                 |

## 2. Evolución del Sistema Operativo

Para poder comprender la historia de los sistemas operativos, es necesario comprender qué es y por qué se desarrolló en primer lugar.

Originalmente los ordenadores servían para resolver una única tarea, y por lo tanto su programación era muy específica a esa máquina. De este modo, no hacía falta que un software fuese generalizado para varios tipos de ordenadores, ya que para cada ordenador distinto se programaba su tarea. Mas esto limitaba mucho la usabilidad de las computadoras, ya que sólo se podían usar para una tarea. Si uno quisiera ejercer otra tarea con la misma máquina tendría que escribir el código desde cero, y el código tendría que ser específico a las propiedades (e.g. periféricos, tamaño de memoria, almacenamiento) de esa máquina. Es decir, con que dos máquinas no fuesen exactamente iguales en sus componentes y periféricos, el mismo código no podría correr en ambas ya que la forma de comunicar con estos componentes y periféricos cambiaría de una a otra. Por poner un ejemplo simple, imagine un monitor que recibe los datos por medio de un bloque de datos de 256 bytes, y otro que lo recibe por un *streaming* que se hace byte por byte. Si el código fuese creado de modo que sólo funciona con uno, sería absolutamente incompatible con el otro.

Para resolver este problema y por lo tanto aumentar la usabilidad de las computadoras, se inventaron los sistemas operativos para abstraer el hardware y así hacer que el software fuera más portable de un sistema a otro. De este modo, el software tan sólo tendría que usar el interfaz que provee el sistema operativo, que sería uniforme, y el sistema operativo, por su parte, se ocuparía de transmitir esta información al hardware correspondiente, siendo programado para poder tratar con una variedad más amplia de configuraciones. Usando nuestro ejemplo de antes, el software simplemente enviaría el texto a escribir a la pantalla al sistema operativo por el método que fuese, y el sistema operativo, sabiendo en el arranque qué tipo de monitor está conectado a la máquina, enviaría los datos de forma que lo entendería ese hardware en concreto.

A consecuencia de esto, los sistemas operativos también llegaron a implementar métodos para manejar los recursos de la máquina (e.g. memoria, CPU), en particular cuando se desarrollaron los sistemas multiprogramados. Al tener varios programas corriendo en la máquina a la vez, es necesario distribuir de manera eficaz y justa los recursos del sistema. De esta tarea también se encargaron los sistemas operativos.

Con esta evolución, el sistema operativo ha quedado como un interfaz entre el software y el hardware, además de arbitro entre los propios procesos del ordenador. En la actualidad se van designando más tareas todavía a

este programa para poder aprovechar del mejor rendimiento y control de la máquina.

### 3. IBM y los Primeros Sistemas Operativos

Los primeros sistemas operativos se hicieron para las computadoras de IBM, que en esta época tan sólo se usaban en ambientes de industria, academia, o militar. El primero fue desarrollado por General Motors, que se llamaba GM-NAA I/O. Cada sistema operativo era muy específico a la máquina en sí, y muy diferente de los demás. Mientras que hoy en día estamos acostumbrados a que en todos los sistemas haya cierta similitud de comandos e interfaces, este no era el caso en los años 50. Los sistemas operativos eran muy diversos y sus comandos e interfaces variaban radicalmente debido a que se desarrollaron de forma completamente independiente, lo cual no había ningún estándar al que todos conformaban. Aún así servían para su propósito que era abstraer los dispositivos periféricos que se usaban en conjunto con la máquina.

Esto continuó de esta manera hasta los años 60, en que IBM tomó la decisión técnica y de negocios de facilitar lo que ya estaban haciendo sus clientes. En 1964 anunciaron la creación de la serie *System/360*; una serie de máquinas caracterizadas por usar el mismo juego de instrucciones y comandos de entrada/salida. Todo esto fue con la intención de desarrollar un único sistema operativo para toda la serie: *OS/360*. Aunque esto fue un fracaso tan estrepitoso debido a la diferencia de rendimiento tuvieron que desarrollar también una serie de variaciones del sistema operativo en sí.

### 4. Minicomputadoras y la Computación en Tiempo-Real

En los años 60 se estaba viendo la potencial que tenían las computadoras, no sólo para resolver tareas programadas, sino también para resolver problemas con entradas en tiempo-real. Para esto, se empezaron a desarrollar sistemas operativos en tiempo-real (RTOS). El más famoso de éstos fue el sistema operativo UNIX, que fue desarrollado en Bell Laboratories de AT&T por Ken Thompson y Dennis Ritchie (entre otros). Este sistema operativo revolucionó el mundo de la informática, especialmente con su nuevo lenguaje de programación, C, que facilitaba la abstracción de las instrucciones de cada arquitectura en una lógica única. Esto fue muy necesario en esta época en que había tantas arquitecturas, y por lo tanto, tantos juegos de instrucciones

distintos. Esto significaba que (en teoría) se podría escribir el código en C una vez y se podría compilar para cualquier sistema.

Gracias a esta revolución, cada vez que C se portaba a otra arquitectura, el sistema operativo UNIX también se podía portear a la misma. Esto fue facilitado por el hecho de la filosofía de desarrollo que seguía, que se fundamentaba en modularidad y simplicidad, que se denomina de forma popular como la *filosofía UNIX*. Esto lo convirtió en modelo para la computación moderna y todos sus sucesores. Es así hasta el punto que es casi imposible encontrar algún sistema operativo moderno que no esté escrito en algún lenguaje de alto nivel (sea C, C++, Rust, u otros). Su filosofía también dio lugar a los movimientos de software libre y código abierto, además de inspirar la creación de proyectos como serían GNU is Not UNIX (GNU), Linux, las variaciones de BSD, y macOS.

## 5. Microcomputadores: Computación para las Masas

Ya para los años 70 empezaron a llegar los microcomputadores. Aunque al principio eran para los aficionados a la computación, seguirían desarrollándose hasta convertirse en lo que hoy en día reconoceríamos como un ordenador PC.

En esta nueva época de computación, los sistemas operativos se tenían que adaptar a las necesidades cotidianas, que solían precisar de un uso altamente compartido de los recursos entre muchos procesos a la vez. Ya el sistema operativo cobraba cada vez más importancia, aunque pasase completamente desapercibido por el usuario que se fijaba únicamente en las aplicaciones gráficas que veía en la pantalla.

Los sistemas operativos más populares para salir en esta época de la computación serían: Microsoft Windows, macOS, GNU/Linux, y BSD.

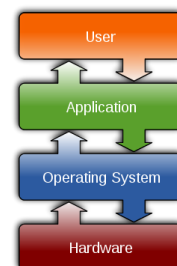


Figura 1: Jerarquía de un sistema.

### 5.1. Microsoft Windows

El sistema operativo más popular para los PCs es sin duda Microsoft Windows. Su primera versión, Windows 1.0, salió al mercado en 1985. Fue un intento de poner un interfaz gráfico por encima de MS-DOS, que lo precedía. Se popularizó sobre todo por ser un sistema que intentaba acomodar al usuario final en las tareas cotidianas, y por lo tanto también se encargó de soportar una gran variedad de hardware. Hasta hoy sigue siendo uno de los sistemas que más dispositivos y periféricos soporta.

### 5.2. macOS

Sistema desarrollado por Apple Inc., originalmente tuvo su propia base, mas después de un fracaso y de casi desaparecer del mundo de la computación decidieron adoptar el modelo UNIX, rompiendo el soporte con versiones anteriores para poder proveer un sistema más estable y sostenible. Para hacer esto tomaron bastante código del sistema operativo BSD, usando su propio kernel, y desarrollaron la versión Mac OSX en 2001. Se popularizó por proveer una computación más estable, seguro, y fácil de usar que Windows,

además de enfocarse en la estética de sus productos.

Originalmente macOS estaba hecho para correr en CPUs con la arquitectura de PowerPC, mas cuando esto empezó a caer en desuso, y había una falta de aplicaciones que lo soportaban, empezaron a cambiar también a la arquitectura x86, y sus ordenadores usaban CPUs de Intel. Recientemente han empezado a cambiar de nuevo su arquitectura de base, adoptando la arquitectura de ARM con los CPUs M1.

### 5.3. GNU/Linux

El sistema operativo de GNU/Linux (más conocido simplemente como Linux) es el conjunto de herramientas GNU y el kernel de Linux. Originalmente el proyecto GNU fue creado para desarrollar un sistema operativo que fuese completamente software libre, basándose en el modelo del sistema UNIX. Llegaron a implementar casi todo el sistema menos el kernel, que durante esa época solían usar Mach en su lugar mientras fueron desarrollando su propio kernel, GNU Hurd. Pero como GNU Hurd es un microkernel, su estructura y desarrollo era más complicado, y tardaban mucho más en sacar una versión que fuera utilizable. A la misma vez Linus Torvalds desarrolló su propio kernel monolítico llamado Linux, y a recomendación de sus compañeros lo publicó bajo la licencia GNU GPL versión 2. Al principio este kernel no cobró mucha importancia, ya que era un proyecto personal de Torvalds, y sólo lo usaban los aficionados. Pero se popularizó cuando se empezó a usar en conjunción con el resto de componentes de GNU.

La combinación GNU/Linux se dio a conocer cuando Ian Murdock creó su distribución, Debian, que hoy en día es una de las distribuciones de GNU/Linux que más se usan (contando también sus derivados, como sería Ubuntu). A partir de ahí, más proyectos surgieron de crear distribuciones juntando el *userland* de GNU y el kernel de Linux.

Ya hoy, GNU/Linux es el sistema operativo más usado, y se ve incluso en algunos de sus derivados como sería Android.

### 5.4. Berkley Software Distribution (BSD)

Este sistema operativo fue desarrollado en la universidad de Berkley California. Muchas veces se confunde por una distribución de GNU/Linux, cuando en realidad es un sistema operativo completamente diferente, aunque también se basa en UNIX.

Cuando se discontinuó el proyecto en 1995, surgieron también varias distribuciones, más notablemente FreeBSD, OpenBSD, y NetBSD; cada una con

su especialidad. Es conocida también por sus licencias permisivas, que generalmente sólo requieren atribución del autor/proyecto original. Fue debido a esto, y que fue uno de los primeros sistemas operativos a tener capacidades de internet integrados desde el primer momento, fue adoptado en parte, o incluso completamente, por otros sistemas operativos. Apple lo usó como base de la nueva versión de Mac OSX, Microsoft integró su código de redes en su propio sistema, y tanto el Playstation 4 como el Nintendo Switch hacen uso de código de BSD.



## 6. Derechos de Autor

Copyright © 2021 Nicolás A. Ortega Froya <nicolas@ortegas.org>  
Este documento se distribuye bajo las condiciones de la licencia *Creative Commons Attribution No Derivatives 4.0 International*. Creado con L<sup>A</sup>T<sub>E</sub>X.