

**Instituto**

**Politécnico**

**Nacional**

**Escuela Superior de Cómputo**

**Investigación Estructura SO Windows**

**Materia:**

Sistemas Operativos

**Grupo:**

2CM8

**Alumno:**

Castro Cruces Jorge Eduardo

**Fecha:**

Lunes, febrero 10, 2020

**Introducción**

Con el paso de los años se ha producido una [evolución](https://www.monografias.com/trabajos16/teoria-sintetica-darwin/teoria-sintetica-darwin.shtml) gradual de la [estructura](https://www.monografias.com/trabajos15/todorov/todorov.shtml#INTRO) y capacidades de los [Sistemas](https://www.monografias.com/trabajos11/teosis/teosis.shtml) Operativos. Sin embargo, recientemente se ha introducido un cierto número de nuevos elementos de [diseño](https://www.monografias.com/trabajos13/diseprod/diseprod.shtml) en los nuevos Sistemas Operativos y en las nuevas versiones de los Sistemas Operativos existentes. Estos [Sistemas Operativos](https://www.monografias.com/Computacion/Sistemas_Operativos/) modernos responden a nuevos desarrollos del [hardware](https://www.monografias.com/Computacion/Hardware/) y nuevas aplicaciones. Entre estos dispositivos de hardware están las [máquinas](https://www.monografias.com/trabajos6/auti/auti.shtml) multiprocesador, incrementos enormes de la [velocidad](https://www.monografias.com/trabajos13/cinemat/cinemat2.shtml#TEORICO) de la máquina, alta velocidad en los enlaces de las [redes](https://www.monografias.com/Computacion/Redes/) de [comunicación](https://www.monografias.com/trabajos12/fundteo/fundteo.shtml) e incremento en el tamaño y variedad de los dispositivos de [almacenamiento](https://www.monografias.com/trabajos12/dispalm/dispalm.shtml) de [memoria](https://www.monografias.com/trabajos13/memor/memor.shtml). En los campos de aplicación que han influido en el diseño de los [Sistema](https://www.monografias.com/trabajos11/teosis/teosis.shtml) Operativos están las aplicaciones [multimedia](https://www.monografias.com/trabajos10/mmedia/mmedia.shtml), el acceso a [Internet](https://www.monografias.com/Computacion/Internet/) y páginas [Web](https://www.monografias.com/trabajos5/laweb/laweb.shtml) y la ejecución [cliente](https://www.monografias.com/trabajos11/sercli/sercli.shtml)/[servidor](https://www.monografias.com/trabajos12/rete/rete.shtml).

El porcentaje de cambios en las demandas de los Sistemas Operativos, requiere no solamente las modificaciones y mejoras en las arquitecturas ya existentes, sino nuevas formas de [organización](https://www.monografias.com/trabajos6/napro/napro.shtml) del [Sistema Operativo](https://www.monografias.com/Computacion/Sistemas_Operativos/). Muchos de los diferentes enfoques y elementos de diseño se han probado tanto en [Sistemas Operativos](https://www.monografias.com/Computacion/Sistemas_Operativos/) experimentales como comerciales, y muchos de ellos encajan dentro de las siguientes categorías:

* Arquitectura Micronúcleo.
* Multihilos.
* Multiproceso Simétrico.
* Sistemas Operativos Distribuidos.
* Diseño Orientado a Objeto.

La mayor parte de los Sistemas Operativos hasta hace poco [tiempo](https://www.monografias.com/trabajos901/evolucion-historica-concepciones-tiempo/evolucion-historica-concepciones-tiempo.shtml) se caracterizaban por un gran núcleo monolítico. Gran parte de la funcionalidad que se pensaba debía tener un Sistema Operativo la proporcionaba este gran núcleo, incluyendo [planificación](https://www.monografias.com/trabajos34/planificacion/planificacion.shtml), sistema de [archivos](https://www.monografias.com/trabajos7/arch/arch.shtml), redes, controladores de dispositivos, [gestión](https://www.monografias.com/trabajos15/sistemas-control/sistemas-control.shtml) de memoria y muchas cosas más. Normalmente un núcleo monolítico está implementado como un único [proceso](https://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE), con todos sus componentes compartiendo el mismo espacio de direcciones.

La [arquitectura](https://www.monografias.com/trabajos6/arma/arma.shtml) micronúcleo asigna solamente unas pocas [funciones](https://www.monografias.com/trabajos7/mafu/mafu.shtml) esenciales al núcleo, incluyendo espacios de direcciones, comunicación entre [procesos](https://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE) (IPC) y planificación básica. Otros [servicios](https://www.monografias.com/trabajos14/verific-servicios/verific-servicios.shtml) del Sistema Operativo los proporciona procesos, algunas veces llamados [servidores](https://www.monografias.com/trabajos12/rete/rete.shtml), que se ejecutan en modo usuario y que el micronúcleo trata como a cualquier otra aplicación. Este enfoque desconecta el núcleo y el [desarrollo](https://www.monografias.com/trabajos12/desorgan/desorgan.shtml) de servidores. Los servidores pueden estar diseñados para aplicaciones específicas o necesidades del entorno. El enfoque del micronúcleo simplifica la implementación, proporciona flexibilidad y se adapta bien para entornos distribuidos. En esencia, un micronúcleo interactúa de la misma forma con procesos servidores locales y remotos, facilitando la [construcción](https://www.monografias.com/trabajos35/materiales-construccion/materiales-construccion.shtml) de sistemas distribuidos.

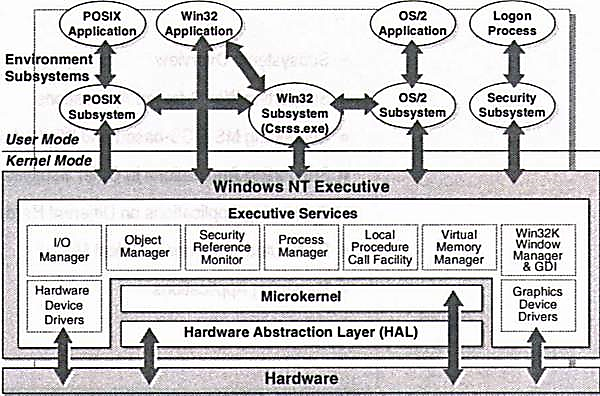
Este [trabajo](https://www.monografias.com/trabajos34/el-trabajo/el-trabajo.shtml) intenta abordar la arquitectura del Sistema Operativo [Windows](https://www.monografias.com/trabajos15/ms-windows/ms-windows.shtml) y los servicios que cada uno de sus componentes brinda para llevar a cabo cada una de las categorías antes expuestas.

**Visión General de la Arquitectura de Windows.**

Un Sistema Operativo serio, capaz de competir en el [mercado](https://www.monografias.com/trabajos13/mercado/mercado.shtml) con otros como [Unix](https://www.monografias.com/trabajos36/sistema-unix/sistema-unix.shtml) que ya tienen una posición privilegiada, en cuanto a resultados, debe tener una serie de características que le permitan ganarse ese lugar. Algunas de estas son:

* Que corra sobre múltiples arquitecturas de hardware y plataformas.
* Que sea compatible con aplicaciones hechas en plataformas anteriores, es decir que corrieran la mayoría de las aplicaciones existentes hechas sobre versiones anteriores a la actual, nos referimos en este caso particular a las de 16-bit de [MS-DOS](https://www.monografias.com/trabajos/manualdos/manualdos.shtml) y [Microsoft](https://www.monografias.com/trabajos13/quienbill/quienbill.shtml) Windows 3.1.
* Reúna los requisitos gubernamentales para POSIX (Portable Operating System Interface for Unix).
* Reúna los requisitos de la [industria](https://www.monografias.com/trabajos16/industria-ingenieria/industria-ingenieria.shtml) y del [gobierno](https://www.monografias.com/trabajos4/derpub/derpub.shtml) para la [seguridad](https://www.monografias.com/trabajos/seguinfo/seguinfo.shtml) del Sistema Operativo.
* Sea fácilmente adaptable al mercado global soportando [código](https://www.monografias.com/trabajos12/eticaplic/eticaplic.shtml) Unicode.
* Sea un sistema que corra y balancee los procesos de forma paralela en varios [procesadores](https://www.monografias.com/trabajos5/sisope/sisope.shtml) a la vez.
* Sea un Sistema Operativo de memoria virtual.

Uno de los pasos más importantes que revolucionó los Sistemas Operativos de la Microsoft fue el diseño y creación de un Sistema Operativo extensible, portable, fiable, adaptable, robusto, [seguro](https://www.monografias.com/trabajos5/segu/segu.shtml) y compatible con sus versiones anteriores (Windows NT).

Y para ello crearon la siguiente arquitectura modular:

La cual está compuesta por una serie de componentes separados donde cada cual es responsable de sus funciones y brindan servicios a otros componentes. Esta arquitectura es del tipo cliente – servidor ya que los [programas](https://www.monografias.com/Computacion/Programacion/) de aplicación son contemplados por el sistema operativo como si fueran [clientes](https://www.monografias.com/trabajos11/sercli/sercli.shtml) a los que hay que servir, y para lo cual viene equipado con distintas entidades servidoras.

Ya creado este diseño las demás versiones que le sucedieron a [Windows NT](https://www.monografias.com/trabajos11/introwin/introwin.shtml) fueron tomando esta arquitectura como base y le fueron adicionando nuevos componentes.

Uno de las características que Windows comparte con el resto de los Sistemas Operativos avanzados es la división de tareas del Sistema Operativo en múltiples categorías, las cuales están asociadas a los modos actuales soportados por los [microprocesadores](https://www.monografias.com/trabajos11/micro/micro.shtml). Estos modos proporcionan a los programas que corren dentro de ellos diferentes niveles de privilegios para acceder al hardware o a otros programas que están corriendo en el sistema. Windows usa un modo privilegiado (Kernel) y un modo no privilegiado (Usuario).

Uno de los [objetivos](https://www.monografias.com/trabajos16/objetivos-educacion/objetivos-educacion.shtml) fundamentales del diseño fue el tener un núcleo tan pequeño como fuera posible, en el que estuvieran integrados módulos que dieran respuesta a aquellas llamadas al sistema que necesariamente se tuvieran que ejecutar en modo privilegiado (modo kernel). El resto de las llamadas se expulsarían del núcleo hacia otras entidades que se ejecutarían en modo no privilegiado (modo usuario), y de esta manera el núcleo resultaría una base compacta, robusta y estable.

El Modo Usuario es un modo menos privilegiado de funcionamiento, sin el acceso directo al hardware. El código que corre en este modo sólo actúa en su propio espacio de [dirección](https://www.monografias.com/trabajos15/direccion/direccion.shtml). Este usa las APIs (System Application Program Interfaces) para pedir los servicios del sistema.

El Modo Kernel es un modo muy privilegiado de funcionamiento, donde el código tiene el acceso directo a todo el hardware y toda [la memoria](https://www.monografias.com/trabajos16/memorias/memorias.shtml), incluso a los espacios de dirección de todos los procesos del modo usuario. La parte de WINDOWS que corre en el modo Kernel se llama Ejecutor de Windows, que no es más que un conjunto de servicios disponibles a todos los componentes del Sistema Operativo, donde cada [grupo](https://www.monografias.com/trabajos14/dinamica-grupos/dinamica-grupos.shtml) de servicios es manipulado por componentes que son totalmente independientes (entre ellos el Núcleo) entre sí y se comunican a través de interfaces bien definidas.

Todos los programas que no corren en Modo Kernel corren en Modo Usuario. La mayoría del código del Sistema Operativo corre en Modo Usuario, así como los subsistemas de [ambiente](https://www.monografias.com/trabajos15/medio-ambiente-venezuela/medio-ambiente-venezuela.shtml) (Win32 y POSIX que serán explicados en capítulos posteriores) y aplicaciones de usuario. Estos programas solamente acceden a su propio espacio de direcciones e interactúan con el resto del sistema a través de mensajes Cliente/Servidor.

**Capítulo 1**

**Modo Kernel**

**Capa de Abstracción de Hardware (HAL).**

Conocido por sus siglas en [inglés](https://www.monografias.com/trabajos16/manual-ingles/manual-ingles.shtml) HAL (Hardware Abstraction Layer) es una interfaz entre el hardware y el resto del Sistema Operativo, está implementada como una [biblioteca](https://www.monografias.com/trabajos10/ponency/ponency.shtml) de enlace dinámico (dll) y es responsable de proteger el resto del sistema de las especificaciones del hardware, tales como controladores de interrupción e interfaces de entrada/salida. Esta abstracción hace al sistema más portable ya que el resto del sistema no tiene que preocuparse de que plataforma está corriendo. Cada plataforma en que el sistema corre necesita un HAL específico. El diseño intenta que cuando Windows sea portado a una nueva arquitectura de [procesador](https://www.monografias.com/trabajos5/sisope/sisope.shtml), el HAL sea reescrito para el nuevo procesador, pero el resto del sistema simplemente debe ser recompilado.

Este también suministra la interfaz para el multiprocesamiento simétrico (conocido por sus siglas en inglés SMP). Las versiones Server contienen dos HALs para arquitectura de procesador (Intel, MIPS, PowerPC y and Alpha), el primero es usado para soportar un solo procesador, mientras que el segundo soporta hasta cuatro procesadores.

Para cada procesador físico que existe en [la computadora](https://www.monografias.com/trabajos15/computadoras/computadoras.shtml) el HAL representa un procesador virtualizado al microkernel. La idea es que el procesador virtualizado esconda las características especiales del propio procesador al sistema operativo, quiere esto decir que si por ejemplo se tiene dos sistemas multiprocesadores, uno corriendo sobre un procesador Intel y otro corriendo con un Alpha, los HALs en cada sistema serían diferentes, pero los procesadores virtualizados que este presenta al microkernel en ambos casos pudieran ser idénticos. Sobre un sistema SMP (Multiprocesamiento Simétrico) para cada procesador físico en el sistema el HAL representa un procesador virtualizado al microkernel.

A este componente solo pueden acceder componentes del Ejecutor de Windows y nunca se llama por los programas del Modo Usuario. El HAL también intenta ser la única pieza de [software](https://www.monografias.com/Computacion/Software/) dentro del sistema que se comunique con el hardware, la ventaja de esto es que otros programas no pueden escribir [información](https://www.monografias.com/trabajos7/sisinf/sisinf.shtml) en el hardware ni accidentalmente, ni intencionalmente y causar una caída del sistema, también impidiendo que programas lean información directamente del hardware.

Aunque [la meta](https://www.monografias.com/trabajos7/plane/plane.shtml) de Windows es que todas las llamadas relacionas con el hardware sean a través del HAL, la realidad es que un número pequeño de llamadas de los drivers y del Kernel bordean al HAL e interactúan directamente con el hardware.

La capa de Abstracción de Hardware conocida por sus siglas en inglés (HAL) es una biblioteca de manipulación de hardware con rutinas suministradas por Microsoft o por el fabricante del hardware. Esta capa queda en el nivel más bajo del Ejecutor de Windows (entre el hardware y el resto del Sistema Operativo), esta esconde las características de la plataforma para que todas las plataformas y arquitecturas parezcan igual al Sistema Operativo, esto permite al SO correr sobre diferentes plataformas con uno o varios procesadores, facilitando además a los drivers de dispositivos adaptarse a distintas arquitecturas de E/S sin tener que ser modificados en gran medida.

**MicroKernel**

Es el responsable de todas las [acciones](https://www.monografias.com/trabajos4/acciones/acciones.shtml) que se realizan sobre le sistema y casi todas las funciones del sistema pasan a través de él.

El diseño de este componente asigna muchas de las funciones normalmente asignadas al Kernel en los Sistemas Operativos tradicionales a un grupo de programas llamado Ejecutor de Windows, del cual el microkernel es parte, corre en el modo privilegiado y ambos (el ejecutor y el microkernel) se comunican a través de primitivas del sistema operativo a bajo nivel.

La principal tarea de este componente es la planificación de ejecución de hilos (segmento de código perteneciente a un proceso particular). A cada hilo es asignada una prioridad de 0 a 31, este entonces envía hilos a correr en dependencia de su número de prioridad y los permite ejecutarse un tiempo determinado antes de apropiarse de ellos y permitir que otro proceso corra.

Aquí es importante aclarar que el microkernel no planifica la ejecución de procesos, sino que planifica la ejecución de hilos en el entorno de un proceso, este [procedimiento](https://www.monografias.com/trabajos13/mapro/mapro.shtml) es el que hace posible la multitarea con preferencia al ser el microkernel el que planifica la ejecución de todo el código que corre en el sistema.

En un sistema multiprocesador, una copia del microkernel corre en cada procesador. Estos segmentos del microkernel son usados para mantener la coherencia de los [recursos](https://www.monografias.com/trabajos4/refrec/refrec.shtml) del sistema que son compartidos ya que son accedidos por los hilos que corren en todos los procesadores.

Este también es responsable de la manipulación de interrupciones del sistema desde dispositivos físicos. Normalmente cuando el sistema es interrumpido, el microkernel se apropia del hilo que este corriendo en ese momento para procesar la interrupción.

El microkernel también manipula las excepciones del procesador, donde estas excepciones ocurren cuando el procesador intenta hacer alguna operación que no se le está permitida, como el intento de escribir en una porción de memoria a la cual no tiene acceso o cuando se divide por cero.

El uso final del microkernel es suministrar un soporte para la recuperación del sistema de una caída de energía. Si el sistema está equipado con un suministrador de energía ininterrumpible (más conocido por sus siglas inglés UPS) el microkernel es advertido cuando la caída de energía es detectada, entonces este coordina un cierre ordenado del sistema, el cual incluye la advertencia a los dispositivos de Entrada/Salida de la caída de la energía y permitir entonces restaurarse consecuentemente.

Puesto que el Microkernel está involucrado en la mayoría de las acciones asumidas por el Sistema Operativo, las porciones críticas de este son escritas en [lenguaje](https://www.monografias.com/trabajos35/concepto-de-lenguaje/concepto-de-lenguaje.shtml) [ensamblador](https://www.monografias.com/trabajos14/lenguaje-ensamblador/lenguaje-ensamblador.shtml) para garantizar que este pueda correr lo más rápido y eficientemente posible, lo que trae consigo que su optimización sea un factor crítico de funcionamiento cuando el sistema es portado a diferentes arquitecturas.

El microkernel está situado en el [corazón](https://www.monografias.com/trabajos5/ancar/ancar.shtml) de Windows, trabaja muy estrechamente con el HAL (Nivel de Abstracción de Hardware), este planifica la ejecución de hilos y manipula las interrupciones y excepciones de procesos. El papel de este es mantener a los procesadores lo mas ocupado posible. En sentido general este se encarga de las funciones más básicas de todo el SO, como son:

* Ejecución de subprocesos.
* Sincronización multiprocesador.
* Manejo de las interrupciones de hardware.

**El Ejecutor de Windows.**

El Ejecutor de Windows se encarga de las tareas importantes, las que son de vital importancia para el sistema completo, ya que el microkernel está casi siempre demasiado ocupado para dirigirse directamente.

Una definición clara es que el Ejecutor de Windows provee los fundamentos del sistema operativo que serán suministradas a todas las aplicaciones que corren sobre el sistema. Este incluye servicios como la [Administración](https://www.monografias.com/trabajos36/administracion-y-gerencia/administracion-y-gerencia.shtml) de Objetos, de [Memoria virtual](https://www.monografias.com/trabajos10/gesmem/gesmem.shtml#VIRTU), de Entrada-Salida y de Procesos.

El Ejecutor de Windows corre exclusivamente en Modo Kernel y es llamado por los subsistemas de ambiente protegido cuando estos necesitan de sus servicios. Debido a la jerarquía de Windows las aplicaciones que corren en Modo Usuario no pueden llamar segmentos del Ejecutor de Windows directamente, sino servicios de [demanda](https://www.monografias.com/trabajos/ofertaydemanda/ofertaydemanda.shtml) de los subsistemas de ambiente (explicado en capítulos posteriores), como Win32 y POSIX los que a su vez se encargan de llamar los componentes del Ejecutor de Windows.

**El**[**Administrador**](https://www.monografias.com/trabajos10/habi/habi.shtml)**de Objetos.**

El Administrador de Objetos (Object Manager) es usado para crear, modificar y eliminar objetos ([tipos de datos](https://www.monografias.com/trabajos11/basdat/basdat.shtml) abstractos que son usados para representar recursos del Sistema Operativo) usados por todos los sistemas que conforman el Ejecutor de Windows. Este también proporciona información sobre [el estado](https://www.monografias.com/trabajos12/elorigest/elorigest.shtml) de los objetos a todo el Sistema Operativo.

Los objetos pueden ser cosas concretas, tales como puertos de dispositivos, o pueden ser más abstractos como hilos. Cuando un objeto es creado a este se le da un nombre por el cual otros programas pueden accederle. Cuando un proceso necesita acceder al objeto este solicita un tratamiento de objeto al administrador de objetos. El manipulador de objetos suministra un puntero que es usado para localizar al objeto, así como una información de [control](https://www.monografias.com/trabajos14/control/control.shtml)de acceso que dice como se puede acceder a él. Esta información de control de acceso es suministrada por el subsistema de seguridad (tema que se abordará en próximos temas).

Este también se asegura que los objetos no consuman muchos recursos (por lo regular la memoria), manteniendo cuotas para los diferentes tipos de objetos.

Además el Administrador de Objetos se encarga de limpiar objetos huérfanos (objetos que parecen no tener dueño), esto es conocido como recolección de [basura](https://www.monografias.com/trabajos11/recibas/recibas.shtml). La carencia de esta facilidad en Windows 3.x era la causa de muchos [problemas](https://www.monografias.com/trabajos15/calidad-serv/calidad-serv.shtml#PLANT), ya que cuando un [programa](https://www.monografias.com/Computacion/Programacion/) colapsaba o manipulaba incorrectamente los recursos del sistema, los recursos consumidos por este no eran devueltos al sistema para que volvieran a estar disponibles produciendo un error por falta de recursos del sistema. De hecho, esto era un escape de memoria.

A modo de resumen el Administrador de Objetos se encarga de crear, destruir y gestionar todos los objetos del Ejecutor de Windows.

**El Administrador de Procesos.**

El Administrador de Procesos (Process Manager) es el responsable de crear, quitar y modificar los estados de todos los procesos e hilos. Este también proporciona información sobre el [estado](https://www.monografias.com/trabajos12/elorigest/elorigest.shtml) de procesos e hilos al resto del sistema.

Un proceso, por la definición, incluye un espacio de dirección virtual, uno o más hilos, un segmento de código del programa ejecutable, y un conjunto de recursos del sistema. Un hilo es un objeto ejecutable que pertenece a un solo proceso y contiene a un contador del programa que apunta a su posición actual en el segmento de código ejecutable del proceso, dos [pilas](https://www.monografias.com/trabajos11/pila/pila.shtml), y un conjunto de [valores](https://www.monografias.com/trabajos14/nuevmicro/nuevmicro.shtml) del [registro](https://www.monografias.com/trabajos7/regi/regi.shtml).

El Administrador de Procesos, como todos los miembros del Ejecutor de Windows, juega un papel vital en el funcionamiento del sistema entero. Cuando una aplicación comienza su ejecución, se crea como un proceso lo que requiere una llamada al Administrador de Procesos. Como todo proceso debe tener por lo menos un hilo, el Administrador de Procesos es invocado de nuevo para crear el hilo.

El Administrador de Procesos se usa para manejar los hilos, pero no tiene su propio conjunto de [políticas](https://www.monografias.com/trabajos10/poli/poli.shtml) sobre cómo planificar la ejecución de procesos e hilos. Estas políticas son determinadas por el propio microkernel.

El administrador de Procesos (Process Manager) es el responsable de crear, quitar y modificar los estados de todos los procesos e hilos, así como de proporcionar información sobre el estado de procesos e hilos al resto del sistema.

**El Administrador de Memoria Virtual.**

El Administrador de Memoria Virtual (Virtual Memory Manager o VMM) proporciona la gestión de memoria virtual del sistema. La memoria virtual es un esquema que permite usar los recursos del disco en lugar de la memoria [física](https://www.monografias.com/Fisica/index.shtml) del sistema moviendo las páginas al disco cuando estas no están siendo usadas y recuperándolas cuando se les necesitan. Este es un segmento integral de Windows el cual asigna espacios de direcciones de 32 bit a cada proceso sin preocuparse de la cantidad de memoria física del sistema.

A cada proceso se asigna un espacio de memoria virtual de 4GB. De este espacio, los dos giga bites superiores son reservados para el uso del sistema, mientras que los otros dos giga bites restantes son para el uso del proceso. El Administrador de Memoria Virtual es el responsable de traducir las direcciones de memoria del proceso a las direcciones de memoria reales del sistema. Si la dirección de memoria del proceso hace referencia a un segmento de memoria que ha sido paginada hacia el disco, el Administrador de Memoria Virtual recupera la página del disco.

El Administrador de Memoria Virtual se encarga de todo lo relacionado con la [política](https://www.monografias.com/Politica/index.shtml) de gestión de la memoria, determina los [conjuntos](https://www.monografias.com/trabajos10/historix/historix.shtml) de trabajo de cada proceso, mantiene un conjunto de páginas libres, elige páginas que se van a pasar a la memoria real, sube y baja páginas entre la [memoria RAM](https://www.monografias.com/trabajos11/memoram/memoram.shtml) y el [archivo](https://www.monografias.com/trabajos7/arch/arch.shtml) de intercambio en disco.

**Servicios de Llamadas a**[**Procedimientos**](https://www.monografias.com/trabajos13/mapro/mapro.shtml)**Locales.**

El [Servicio](https://www.monografias.com/trabajos14/verific-servicios/verific-servicios.shtml) de Llamadas a Procedimientos Locales (Local Procedure Call Facility o LPC) se integran al diseño cliente/servidor de Windows. Este es la interfaz entre todos los procesos clientes y servidores que corren localmente en el sistema.

La estructura del Servicio de Llamadas a Procedimientos Locales es muy similar a la de las llamadas a Procedimientos Remotos (RPC), excepto que esta está optimizada y solamente soporta comunicación entre procesos clientes y servidores localmente. Más específicamente, el LPC es un mecanismo que permite a dos hilos en procesos diferentes intercambiar información.

Recuerde que nosotros dijimos que el subsistema de Win32 es una aplicación que corre en el Modo Usuario y correrá en su propio espacio de memoria. Cuando un programa se quiere comunicar con el subsistema Win32 para solicitar servicios, llama una [función](https://www.monografias.com/trabajos7/mafu/mafu.shtml) desde la DLL apropiada, esta función entonces usa la LPC para pasar la petición al subsistema de procesos Win32, la que procesa la demanda y realiza la [acción](https://www.monografias.com/trabajos35/categoria-accion/categoria-accion.shtml) pedida y devuelve un mensaje de realización a través de la LPC.

El Servicio de Llamadas a Procedimientos Locales es el módulo que se encarga de recibir y enviar las llamadas de procedimiento locales entre las aplicaciones cliente y los subsistemas servidores.

**El**[**Monitor**](https://www.monografias.com/trabajos5/losperif/losperif2.shtml#moni)**de Seguridad.**

El Monitor de Seguridad (Security Reference Monitor o SRM) es el lecho de toda la seguridad dentro del sistema WINDOWS y es el responsable de hacer cumplir todas las políticas de seguridad en la [computadora](https://www.monografias.com/trabajos15/computadoras/computadoras.shtml) local.

Este componente trabaja en conjunto con los subsistemas de tiempo de corrida, proceso de conexión al sistema (conocido como logon process) y control de la seguridad local (local security authority). Cuando un usuario intenta conectarse al sistema su [identidad](https://www.monografias.com/trabajos14/cambcult/cambcult.shtml) es verificada, el subsistema de proceso de conexión pide una ficha de acceso de seguridad (conocido por sus siglas en inglés SAT o security [access](https://www.monografias.com/trabajos5/basede/basede.shtml) token) del usuario. El SAT contiene una lista de los privilegios de usuarios y [grupos](https://www.monografias.com/trabajos11/grupo/grupo.shtml). Este se usa como llave para ese usuario durante la sesión de conexión. Siempre que el usuario quiera hacer algo, el SAT es presentado y usado para determinar si el usuario puede realizar las acciones.

Este componente trabaja estrechamente con el Administrador de Objetos. Cada vez que un usuario intenta acceder a un objeto el Administrador de Objetos crea un manipulador para acceder a este y llama al SRM para determinar el nivel de acceso concedido por el manipulador. El SRM usa información contenida en la ficha de acceso del usuario y lo compara con la lista de control de accesos sobre el objeto para ver si al usuario debe concederse el nivel de acceso pedido. De esta forma el SRM tiene el control de la seguridad de acceso de todos los objetos en el sistema.

**El Administrador de Entrada-Salida.**

El Administrador de Entrada-Salida (I/O Manager) es responsable de gestionar la comunicación entre los distintos drivers de dispositivo, para lo cual implementa una interfaz bien definida que permite el tratamiento de todos los drivers de una manera homogénea, sin que intervenga el cómo funciona específicamente cada uno. Tiene una serie de subcomponentes que son:

* Driver del Sistema de Archivos: este se encarga de establecer [la comunicación](https://www.monografias.com/trabajos/lacomunica/lacomunica.shtml) con los drivers de los Sistemas de Ficheros, ya que el sistema permite la coexistencia de múltiples Sistemas de Archivos en diferentes particiones lógicas de la misma unidad física.
* El servidor y el redirector de [red](https://www.monografias.com/Computacion/Redes/).
* Los drivers de dispositivo del sistema.
* El administrador de caches (Cache Manager): este se encarga de manipular la cache para todo el Sistema de Entrada y Salida. Este es un [método](https://www.monografias.com/trabajos11/metods/metods.shtml)que utilizan los sistemas de archivos para mejorar su rendimiento, donde en lugar de leer y escribir en disco un fichero usado frecuentemente este se almacena en una cache de memoria y [la lectura](https://www.monografias.com/trabajos16/metodo-lecto-escritura/metodo-lecto-escritura.shtml) y [escritura](https://www.monografias.com/trabajos16/metodo-lecto-escritura/metodo-lecto-escritura.shtml) de estos ficheros se realiza desde memoria. Este componente se encarga de la magia negra que es a menudo necesaria para hacer que varios dispositivos se comuniquen entre si y convivan juntos en un segmento. El Administrador de Entrada-Salida (I/O Manager) es responsable de gestionar la comunicación entre los distintos drivers de dispositivo.

**Capítulo 2**

**Modo Usuario**

**Subsistemas de Ambiente Protegido**

Dos de los objetivos de WINDOWS son [personalidad](https://www.monografias.com/trabajos14/personalidad/personalidad.shtml) y compatibilidad. Esto ha sido logrado a través de los subsistemas de ambiente protegido.

La personalidad esencialmente significa que WINDOWS expone múltiples conjuntos de interfaces de programas de aplicación (APIs) y puede actuar eficazmente como si fuera un sistema operativo diferente. WINDOWS viene con una personalidad POSIX y OS/2 además de sus personalidades Win32, Win16 y DOS.

En WINDOWS, hay tres subsistemas de ambiente protegido:

* El subsistema de Win32
* El subsistema de POSIX
* El subsistema de OS/2

Aunque algunas veces se muestran las personalidades Win16 y DOS incluidas en una lista de subsistemas de ambiente protegido, ellas realmente son parte del subsistema Win32.

Los subsistemas de ambiente protegido actúan como los mediadores entre las aplicaciones del Modo Usuario y el Ejecutor de Windows.

Recuerde que el Ejecutor de Windows y todos sus componentes viven en el Modo Privilegiado o Modo Kernel, mientras que todos los demás viven en el Modo Usuario, esto incluye todos los subsistemas de ambiente. Cuando una aplicación hace una llamada a un subsistema de ambiente, este es pasado a través de una capa de servicios del Ejecutor de Windows.

Cada subsistema de ambiente guarda huella de sus propios procesos y trabaja independientemente de los otros subsistemas. Cada aplicación sólo puede correr en el subsistema para el cual fue diseñado. Cuando usted inicia una aplicación en WINDOWS, mira el encabezamiento representado por el archivo y determina en cuál subsistema ejecutar la aplicación.

**El Subsistema Win32**

Win32 es el subsistema nativo y primario de WINDOWS. Las bases para este subsistema es el conjunto de APIs de Win32. Muchos de estas API son extensiones directas de sus homólogas Win16.

Este subsistema actúa como un servidor para todos los otros subsistemas de ambiente soportados en WINDOWS, los que actúan como clientes y traducen sus llamadas API hacia las API apropiadas de Win32.

El subsistema Win32 es responsable de toda la entrada y salida. Este posee el control de la pantalla, el [teclado](https://www.monografias.com/trabajos5/sisope/sisope2.shtml#tecla), y el ratón. Cuando otros subsistemas, como OS/2 o POSIX, necesitan beneficiarse de estos dispositivos, ellos piden los servicios al subsistema de Win32.

Algunos de los objetivos que se trazaron para mantener la compatibilidad con las aplicaciones hechas en versiones anteriores fueron:

* Permitir que los programas hechos sobre DOS pudieran correr sin modificación.
* Suministrar la capacidad para ejecutar la mayoría de las aplicaciones Windows de 16 bits sin modificación
* Proteger al sistema y otras aplicaciones de 32 bits de la interferencia de las aplicaciones de 16 bits y DOS.
* Permitir a las plataformas RISC (Reduced Instruction set Computer, [microprocesador](https://www.monografias.com/trabajos12/microco/microco.shtml) cuyo número de instrucciones es reducido para lograr una frecuencia más alta de trabajo) ejecutar aplicaciones Windows de 16 bits y DOS.
* Suministrar un mecanismo para compartir [datos](https://www.monografias.com/trabajos11/basda/basda.shtml) entre aplicaciones Windows de 32 y 16 bits.

Muchas personas piensan en Windows 3.x como un Sistema Operativo. Técnicamente, no es un verdadero Sistema Operativo, sino una interfaz de usuario que es miembro del DOS, el verdadero Sistema Operativo.

Así que, el primer paso en proporcionar compatibilidad fue crear un ambiente de DOS. El ambiente de DOS en WINDOWS se llama la máquina virtual de DOS (Machine DOS Virtual o VDM). El VDM es una aplicación de modo usuario de 32 bits el cual solicita los servicios del subsistema de Win32 y en ocasiones directamente a la capa de servicios del sistema. Es basado en DOS 5.0.

WINDOWS permite ejecutar tantas aplicaciones de DOS como uno desee, donde cada aplicación corre en su propio VDM. Puesto que los VDMs son nada más que procesos normales bajo WINDOWS, ellos también son multitarea preventiva al igual que otros procesos en el sistema. Por consiguiente, puede decirse que WINDOWS permite la multitarea preventiva de programas de DOS.

Uno de los rasgos adicionales del VDM es que le da 620 KB de memoria "convencional" libre al usuario. Lo milagroso sobre esto es que también da a las aplicaciones de DOS soporte de ratón, red, y [CD-ROM](https://www.monografias.com/trabajos/multimediaycd/multimediaycd.shtml).

El Subsistema Win32 es el más importante, ya que atiende no sólo a las aplicaciones nativas de Windows, sino que para aquellos programas no Win32, reconoce su tipo y los lanza hacia el subsistema correspondiente. En el caso de que la aplicación sea MS-DOS o Windows de 16 bits (Windows 3.11 e inferiores), lo que hace es crear un nuevo subsistema protegido. Así, la aplicación DOS o Win16 se ejecutaría en el contexto de un proceso llamado VDM (Virtual DOS Machine, máquina virtual DOS), que no es más que un simulador de un ordenador funcionando bajo MS-DOS. El subsistema soporta una buena parte del API Win32. Así, se encarga de todo lo relacionado con la interfaz gráfica con el usuario (GUI), controlando las entradas del usuario y salidas de la aplicación.

**El Subsistema POSIX.**

Microsoft prestó mucha [atención](https://www.monografias.com/trabajos14/deficitsuperavit/deficitsuperavit.shtml) a los diferentes estándares de sistemas abiertos cuando Windows NT estaba en vía de desarrollo. Ellos reconocieron el [valor](https://www.monografias.com/trabajos14/nuevmicro/nuevmicro.shtml) de soportar sistemas abiertos como un método para ganar aceptación de su nuevo sistema operativo avanzado dentro del mercado.

Uno de los estándares más frecuentemente citados soportados por Windows es el POSIX (Interfaz de Sistema operativo Portable Basado en Unix), el cual representa la interfaz del Sistema Operativo portable y fue desarrollado por el IEEE (Instituto de Ingenieros en [Electricidad](https://www.monografias.com/trabajos10/nofu/nofu.shtml) y [Electrónica](https://www.monografias.com/trabajos5/electro/electro.shtml)) como un método de proporcionar portabilidad a las aplicaciones hechas sobre plataformas UNIX. No obstante, POSIX se ha integrado en muchos sistemas no UNIX.

Existen muchos niveles de obediencia con POSIX. Estos niveles representan un conjunto de evoluciones de propuestas, aunque no todas han sido aprobadas como estándares.

El subsistema de POSIX requiere un mínimo de servicios que son proporcionados por WINDOWS. Cuando una aplicación de POSIX corre en WINDOWS, el subsistema es cargado y traduce las llamadas API del [lenguaje C](https://www.monografias.com/trabajos4/lenguajec/lenguajec.shtml), requeridas para soportarlo en llamadas a APIs de Win32 las que son servidas por el subsistema Win32.

Debido a la [naturaleza](https://www.monografias.com/trabajos36/naturaleza/naturaleza.shtml) limitada, el subsistema de POSIX en WINDOWS no suministra soporte para gestión de redes o sistema de seguridad.

El Subsistema POSIX interacciona con el Ejecutor de Windows. Se encarga de definir aspectos específicos del Sistema Operativo UNIX, como pueden ser las relaciones jerárquicas entre procesos padres e hijos (las cuales no existen en el subsistema Win32, por ejemplo, y que por consiguiente no aparecen implementadas directamente en el Ejecutor de Windows).

**El Subsistema OS/2.**

El subsistema de OS/2 está implementado como un subsistema de ambiente protegido, parecido al subsistema POSIX. Este traduce las llamadas API de OS/2 en llamadas a APIs de Win32 que son servidas por el subsistema de Win32.

El subsistema y sus aplicaciones corren en su propio espacio de memoria protegido de 32 bits y constituyen multitarea preventiva unas respecto a otras y respecto a otras aplicaciones que corren en el sistema.

Además de un conjunto de [motores](https://www.monografias.com/trabajos10/motore/motore.shtml) APIs de OS/2, el subsistema implementa muchos APIs gestores de [LAN](https://www.monografias.com/trabajos15/redes-clasif/redes-clasif.shtml) (Red de Área Local), incluyendo tuberías, NETBIOS y mailslots. De esta manera difiere del subsistema POSIX ya que este no posee soporte para gestión de redes.

El Subsistema OS/2 igual que el subsistema POSIX proporciona un entorno para aplicaciones UNIX, este subsistema da soporte a las aplicaciones OS/2. Proporciona la

interfaz gráfica y las llamadas al sistema; las llamadas son servidas con ayuda del Ejecutor de Windows.

**Conclusiones.**

Windows es un sistema que aprovecha la [potencia](https://www.monografias.com/trabajos14/trmnpot/trmnpot.shtml) de los procesadores, ha sido diseñado para adaptarse a las nuevas tecnologías, ofrece compatibilidad con varias plataformas (OS/2, Unix y versiones anteriores a el mismo), soporta el multiprocesamiento simétrico, buen rendimiento y conectividad, seguridad y al no estar encasillado en ningún [modelo](https://www.monografias.com/trabajos/adolmodin/adolmodin.shtml) estandar de Sistema Operativo tiene la capacidad de combinar las ventajas del modelo cliente/servidor, puede correr además sobre múltiples arquitecturas con un mínimo de cambios, permite que varios procesos sean ejecutados simultáneamente en varios procesadores y estos no se apropien de recursos del sistema por tiempo indefinido, sino por tratamiento del sistema.

**Referencias.**

[Solo00] Solomon, David A.y Russinovich Mark "Inside Microsoft [Windows 2000](https://www.monografias.com/trabajos5/windows/windows.shtml)". 3ra Edi. Microsoft Press. Washington. 2000.

[Stal98] Stallings, William. "Operating Systems". 3ra Edi. Prentice-Hall, Inc. New Jersey. 1998.

[Stal01] Stallings, William. "Systemas Operativos". 4ta Edi. Pearson Edicación, S.A. [Madrid](https://www.monografias.com/trabajos901/historia-madrid/historia-madrid.shtml). 2001.

URL: [http](https://www.monografias.com/trabajos11/wind/wind2.shtml)://www.[monografias](https://www.monografias.com/trabajos7/mono/mono.shtml).com/trabajos7/arso/arso2.shtml

URL: <http://www.windowstimag.com/>

URL: <http://usuarios.lycos.es/betzweb/>

URL: <https://www.monografias.com/trabajos26/arquitectura-windows/arquitectura-windows.shtml>