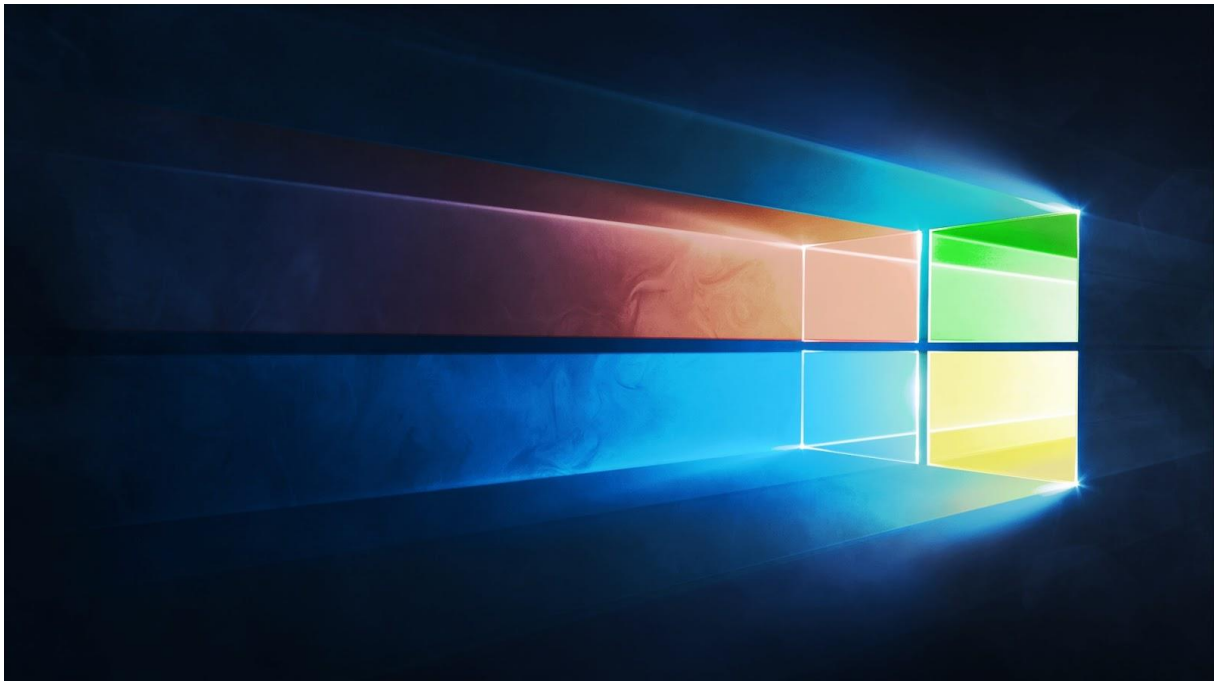


Windows



Grupo: 22
Autores: Aarón Orozco Fernández
Jonathan Arias Busto
Chen Xin Pan Wang
Eduardo Blanco Bielsa
Fecha: 15/04/2022
Asignatura: Sistemas Operativos
Profesor: Miguel Riesco Albizu

Tabla de Contenidos

1.	Introducción	3
2.	Historia	3
3.	Características generales	4
4.	Versiones de Windows	5
5.	Estructura interna del sistema	7
5.1.	Core	7
5.2.	Drivers	10
5.3.	KTM	10
5.4.	Manejo de errores	10
5.5.	Interrupciones	10
6.	Subsistemas de ambiente protegido y APIs	11
6.1.	API	11
6.2.	Win32	11
6.3.	WinRT	12
6.4.	.NET	13
7.	Interfaz gráfica	14
8.	Conclusiones	15
9.	Bibliografía	16

1. Introducción

Desde los inicios de la informática han ido surgiendo diversas tecnologías de una forma escalada, entre las que se encuentra Windows, uno de los sistemas operativos más utilizados hoy en día, creado por Paull Allen y Bill Gates.

En este trabajo se tratarán los distintos aspectos que conforman este sistema operativo, así como su historia y evolución a lo largo de los años. También se estudia qué características generales tiene este sistema operativo y como está formado internamente, hablando de los componentes más representativos como por ejemplo del kernel, de las API o de la interfaz gráfica del sistema operativo.

Para realizar el trabajo hemos obtenido la información de diversas páginas web (mirar bibliografía) como por ejemplo de la documentación oficial de Microsoft. En cuanto a la carga y estructura de trabajo de los componentes del grupo se decidió que lo óptimo era establecer una serie de días en los que todos pudiesen hacer una llamada, vía Discord, para ir avanzando y buscando información de forma conjunta.

2. Historia

En 1975 se fundó Microsoft en Albuquerque, en esa época muy poca gente había oído hablar de ordenadores. Sus desarrolladores fueron Paul Allen y Bill Gates.

El motivo de su creación fue que estos dos jóvenes creían que la informática personal era el futuro y con la creación de Microsoft querían cambiar la forma en la que trabajábamos y proveer a cada hogar con un equipo informático.

En 1980 Steve Ballmer se incorporó a Microsoft para ayudar a dirigir Microsoft.

En 1981 IBM comienza a vender equipos con el sistema operativo, MS-DOS 1.0 (*Microsoft Disk Operating System*), de Microsoft. Este software, con interfaz gráfica de usuario, es la base sobre la que se pueden ejecutar los programas de usuario y además con él se presenta el lenguaje C, con el que se escribirán la mayor parte de los programas.

Esta primera versión nunca se hizo muy popular ya que no era muy eficiente y estuvo limitada a causa de los recursos legales de Apple, que no permitió imitaciones de sus interfaces gráficas de usuario.

Años después, en 1990, surgió la primera versión popular de Windows, Windows 3.0. Esta versión era claramente mejor, puesto que mejoraba las capacidades gráficas de los PCs de la época, gracias al microprocesador 80386 de Intel, pudiendo ejecutar gracias a éste aplicaciones del MS-DOS en modo multitarea gracias a la introducción de la memoria virtual.

3. Características generales

Windows es el nombre que recibe el sistema operativo desarrollado por Microsoft para PC de escritorio, servidores, sistemas empujados, etc. Consiste en un conjunto de programas que permiten la ejecución de los recursos que tiene un ordenador. El término “Windows” (*ventanas*) hace referencia a su interfaz gráfica que representa un modelo basado en ventanas independientes que se encargan de la interacción del usuario con el ordenador.

- **Windows es una distribución privada:** pertenece a Microsoft y no es de código libre, por lo que no puede ser modificado por los usuarios y es de pago. Además, Windows es considerado como un subsistema operativo ya que para su funcionamiento precisa del sistema operativo MS-DOS.
- **Administrador de Windows:** un administrador es una persona que puede realizar cambios en un equipo que afecten a otros usuarios del equipo. El administrador puede cambiar la configuración de seguridad, instalar hardware y software, obtener acceso a todos los archivos del equipo y realizar cambios en otras cuentas de usuario.
- **Ventanas (GUI):** Windows está representado por ventanas que son diferentes cuadros que presentan la información de forma gráfica al usuario. De aquí deriva su nombre, ya que es totalmente distinto al MS-DOS, que emulaba los discos. Cuentan con características a saber, como la barra del título, la barra de menús, la barra de herramientas y la barra de direcciones.
- **Sistema multitarea:** cuenta con la capacidad de ejecutar varias tareas al mismo tiempo.
- **Escritorio:** es la principal característica de este sistema operativo ya que en él se pueden agregar carpetas o accesos directos para tener acceso de forma rápida a la información que el usuario necesita.
- **Menú:** Windows cuenta con un menú de inicio donde se puede encontrar una lista con las aplicaciones instaladas y las subcarpetas creadas por los usuarios. Este menú se despliega mediante el botón de inicio de Windows ubicado en la parte inferior izquierda de la pantalla.
- **Barra de tareas:** conjunto de aplicaciones en ejecución y accesos directos a aplicaciones situado en la parte inferior de la pantalla que incluyen las funcionalidades principales del ordenador, así como aplicaciones instaladas por el usuario.
- **Acceso directo:** se trata de un icono vinculado a un archivo o directorio. Se reconoce por tener una flecha curva hacia arriba ubicada en el lado izquierdo del icono.

4. Versiones de Windows

- **Windows 1.0:** se anunció en noviembre de 1985. Esta versión ofrecía poca funcionalidad y no se trataba de un sistema operativo completo ya que más bien era una extensión gráfica de MS-DOS que intentaba dejar atrás los comandos. Fue desarrollado en 16 bits y requería de 320 KBs de memoria RAM. Tenía un precio de 99\$. Contó con soporte hasta 2001.
- **Windows 2.0:** se anunció en noviembre de 1987. Esta versión incluía por primera vez ventanas que podían solaparse entre ellas. Además, incluía ciertas utilidades, consideradas el origen de las actuales herramientas de ofimática. Fue la primera versión en contener el procesador de texto Word y la hoja de cálculo Excel, teniendo una mejor acogida en el mercado. En él se encuentra el primer panel de control. Contó con soporte hasta 2001, igual que Windows 1.0.
- **Windows 3.0:** se anunció en mayo de 1990. Se modificó la interfaz gráfica con unas ventanas mucho más visuales y exitosas. Fue la primera versión que tuvo un éxito comercial comparable al de Apple. Con la actualización a **Windows 3.1** en abril de 1992, el sistema ofreció el famoso Buscaminas, entre otros juegos, así como el administrador de archivos y la compatibilidad con las tipografías TrueType desarrolladas por Apple. Dejó de tener soporte en 2001.
- **Windows 95:** se anunció en agosto de 1995. Ofrece una interfaz muy mejorada donde ya aparecen la barra de tareas y el menú de inicio, dos elementos característicos de Windows. Se incluía Internet Explorer, el navegador web de Microsoft. Se pasaba de una arquitectura multitarea de 16 bits a una arquitectura apropiativa de 32 bits. Fue también la primera edición que soportaba el modo multitarea. Dejó de tener soporte en 2001.
- **Windows NT Server:** se anunció en julio de 1993. Es un sistema operativo para servidores enfocado a estaciones de trabajo y servidores de red, que destaca por ser ampliable e independiente de la plataforma y hardware. Se podía ejecutar en procesadores Intel x86, RISC y DEC Alpha para que los usuarios pudiesen escoger sus sistemas informáticos. Necesitaba como mínimo 75 MB de disco duro libre y 12 MB de RAM. Tuvo soporte hasta 2004.
- **Windows 98:** se anunció en junio de 1998. A diferencia de las versiones anteriores esta versión se centraba más en el usuario que en el entorno empresarial e industrial. Incorporó el sistema de archivos FAT32 que trató de potenciar el acceso a la red. Además, incluyó la compatibilidad con lectores de DVDs y la incorporación de los primeros puertos USB. Era un sistema lento y poco fiable en comparación con su antecesor Windows 95. Con la versión 2.0 de este sistema operativo se incluyeron mejoras como Internet Explorer 5 y conexión compartida. Contó con soporte hasta 2006.
- **Windows 2000:** se anunció en marzo de 1999. Es una versión profesional cuyo objetivo era ofrecer todo el rendimiento a todos aquellos usuarios avanzados y profesionales que ejecutaban programas de alto rendimiento. Hubo una filtración

de gran parte de su código fuente en Internet lo que hizo a la compañía desarrollar otra versión del sistema. También tenía versiones profesionales enfocadas a servidores, con una mayor protección de datos.

- **Windows XP:** se anunció en octubre de 2001. Fue un gran éxito debido a su gran calidad, se construyó a partir del kernel de Windows NT y tuvo varias versiones en función del usuario, se mejoró la interfaz de usuario permitiendo así a los usuarios profundizar y controlar mucho más el sistema. Con este sistema hubo una gran mejora en el rendimiento general del sistema e incorporó funciones como el filtrado de archivos, el soporte wifi, la asistencia remota y su versión de 64 bits, provocando una explosión el mercado de sistemas operativos.
- **Windows Vista:** se anunció en noviembre de 2007. Incorporó un shell rediseñado y una interfaz gráfica con grandes cambios y enfocada a mejorar la seguridad de los usuarios. Esta versión estaba dirigida a equipos personales y de oficina, portátiles y tablets. Sin embargo, no fue muy exitoso debido a su pésima seguridad.
- **Windows 7:** se anunció en octubre de 2009. Fue una de las versiones más importantes de Microsoft debido a su solidez. Añadía soporte multi touch, un shell rediseñado, una nueva barra de tareas, sistemas de red, mejoras en el rendimiento y una reducción en el consumo de recurso. Contó con soporte hasta enero de 2020.
- **Windows 8:** se lanzó en octubre de 2012. Añadía soporte para microprocesador ARM e incorporaba una interfaz más simple adecuada para pantallas táctiles. Fue un desastre para la compañía teniendo que lanzar Windows 8.1 para intentar solventar la baja aceptación de Windows 8. Tiene soporte hasta 2023.
- **Windows 10:** se lanzó en julio de 2015. Cuenta con una interfaz moderna, un gran rendimiento y además es multiplataforma. Destaca por implementar un asistente virtual (Cortana), un rediseño de la configuración del sistema y el Timeline. Es para muchos usuarios la mejor versión de la historia. Tiene soporte hasta 2029.
- **Windows 11:** fue lanzado en octubre de 2021. Su principal cambio fue la interfaz gráfica redondeando los bordes de las ventanas y mejorando la fluidez de las transiciones. También tiene un menú rediseñado y mejora el rendimiento. Solo admite sistemas de 64 bits.

5. Estructura interna del sistema

Desde hace décadas, Windows ha sido uno de los líderes dentro del mundo de la informática ya que su sistema operativo ha estado en casi todos los ordenadores del mundo. Este sistema operativo es privado y para poder utilizarlo es necesario tener una licencia.

Por otro lado, Linux presenta diferentes distribuciones siendo Ubuntu la más famosa. Su característica más notable es que es un software libre y de código abierto, por lo que la mayoría de sus implementaciones son gratuitas, así como su alta seguridad y su personalización.

5.1. Core

Está situado en el corazón de Windows. Posee una arquitectura cliente-servidor ya que los programas de aplicación son contemplados por el sistema operativo como si fueran clientes a los que hay que servir. Trabaja estrechamente con el **HAL** (*Nivel de Abstracción del Hardware*), que planifica la ejecución de hilos y manipula las interrupciones y excepciones de procesos. Se encarga de mantener a los procesadores lo más ocupados posible.

La mayoría del kernel está escrito en C, aunque poco a poco se va refactorizando en código a C#, JavaScript, C++, etc.

Se organiza o divide en dos modos de ejecución:

- **Modo Usuario:** modo menos privilegiado sin acceso directo al hardware. El código que corre solo actúa en su propio espacio de dirección y utiliza las APIs para pedir los servicios del sistema.
- **Modo Kernel (Administrador):** es un modo muy privilegiado de funcionamiento donde el código tiene acceso directo a todo el hardware y a toda la memoria, incluso a los espacios de dirección de todos los procesos del modo usuario. La parte de Windows que corre en este modo se llama ejecutor de Windows, que es un conjunto de servicios disponibles a todos los componentes del Sistema Operativo, donde cada grupo de servicios es manipulado por componentes que son totalmente independientes entre sí y se comunican a través de interfaces bien definidas.

La mayoría del código de Windows se ejecuta en modo Usuario, así como el subsistema de ambiente (*Win32*) y las aplicaciones de usuario. Estos programas solamente pueden acceder a su propio espacio de direcciones e interactúan con el resto del sistema a través de mensajes Cliente/Servidor.

El **Modo Kernel** (*modo núcleo o Administrador*) de Windows está compuesto por:

- **Capa de Abstracción Hardware (HAL):** es una interfaz entre el Hardware y el resto del Sistema Operativo que está implementada con una biblioteca 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. Cada plataforma que corre en el sistema necesita un HAL específico. De esta forma, se intenta que cuando Windows sea portado a una nueva arquitectura del procesador, el HAL sea reescrito para el nuevo procesador.

El HAL también suministra la interfaz para el multiprocesamiento simétrico (*SMP*). Para cada procesador físico que existe en el ordenador el HAL representa un procesador virtualizado al microkernel, que aun teniendo diferentes características, al ser virtualizados pasan a ser idénticos.

El HAL intenta ser el único componente del sistema que se comunica con el hardware, lo que impide que otros programas escriban y lean información en el hardware accidental o intencionadamente y causar una caída del sistema.
- **MicroKernel:** es el responsable de todas las acciones que se realizan sobre el sistema y casi todas las funciones del sistema pasan a través de él. La principal tarea de este componente es la planificación de ejecución de hilos en el entorno de un proceso. Este proceso hace posible la multitarea. También es el responsable de la manipulación de interrupciones del sistema desde dispositivos físicos. Cuando el sistema es interrumpido el microkernel se apropia del hilo que esté corriendo en ese momento para procesar la interrupción. También es el encargado de manipular las excepciones del procesador. El uso final del microkernel es suministrar un soporte para la recuperación del sistema en caso de una caída de energía, permitiendo así al sistema restaurarse consecuentemente.
- **El Ejecutor de Windows:** se encarga de las tareas más importantes como el acceso a los directorios y la ejecución de aplicaciones.
- **El Administrador de Objetos:** se encarga de crear, modificar y eliminar objetos (*tipos de datos abstractos usados para representar recursos del sistema operativo*) utilizados por todos los sistemas que conforman el Ejecutor de Windows. También proporciona información sobre el estado de los objetos a todo el Sistema Operativo y se asegura de que éstos no consuman muchos recursos. Además, se encarga de limpiar objetos huérfanos, lo que se conoce como **recolección de basura**.
- **El Administrador de procesos (Process Manager):** es el responsable de crear, quitar y modificar los estados de los procesos e hilos. También proporciona información sobre el estado de los procesos e hilos al resto del sistema. Un **proceso** incluye un espacio de dirección virtual, mientras que un **hilo** incluye 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**, y un conjunto de **valores del registro**.

- **El Administrador de memoria virtual (VMM):** proporciona la gestión de memoria virtual del sistema, asignando espacios de direcciones de 32 bits a cada proceso sin preocuparse de la cantidad de memoria física del sistema, utilizando los recursos del disco en lugar de la memoria física del sistema. 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. Se encarga de todo lo relacionado con la política de **gestión de memoria**, determina los **conjuntos de trabajo** de cada proceso, mantiene un **conjunto de páginas libre**, elige páginas que se van a pasar a la **memoria real** y sube y baja páginas entre la memoria **RAM** y el archivo de **intercambio en disco**.
- **Servicio de llamadas a procedimientos locales (LPC):** es la interfaz entre todos los procesos clientes y servidores que corren localmente en el sistema (*diseño cliente-servidor de Windows*). La estructura del LPC 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. Permite a dos hilos en procesos diferentes intercambiar información.
- **El Monitor de seguridad (SRM):** es el responsable de la seguridad de Windows, haciendo cumplir todas las políticas de seguridad en el sistema. Este componente trabaja juntamente con los subsistemas de tiempo de ejecución, proceso de conexión al sistema (*logon process*) y control de la seguridad local (LSA). Cuando un usuario intenta conectarse con el sistema su identidad se verifica, el subsistema de proceso de conexión pide una fecha de acceso de seguridad (SAT). El SAT contiene una lista de los privilegios de usuarios y grupos y se usa como llave para ese usuario durante la sesión de conexión.
- **Administrados de Entrada Salida (I/O Manager):** es el responsable de gestionar la comunicación de los distintos drivers de dispositivo, para lo cual implementa una interfaz que permite el tratamiento de todos los drivers de una forma homogénea, sin conocer los detalles específicos de cada driver. Tiene una serie de subcomponentes:
 - **Driver del sistema de archivos:** establece la comunicación con los drivers de los sistemas de ficheros.
 - **El servidor y redirector de red.**
 - **Los drivers de dispositivo del sistema.**
 - **El administrador de cachés (cache manager):** mejora el rendimiento de los sistemas de archivos almacenando los ficheros más frecuentados en una caché de memoria.

5.2. Drivers

Los drivers o controladores son programas que sirven de intermediarios entre Windows y los dispositivos o componentes de hardware del PC. Son necesarios para que funcionen impresoras, escáneres, la placa madre, tarjetas de red, ..., es decir, para que Windows se “entienda” con ellos. Cada controlador está creado para un sistema operativo y un dispositivo concretos. Estos se suelen ir actualizando para corregir defectos o incluir mejoras, no son algo fijo.

5.3. KTM

El **administrador de transacciones del kernel (KTM)** es un servicio de administración de transacciones que permite al kernel crear un sistema de procesamiento de transacciones (*TPS*) en modo usuario y en modo kernel. Hace que las transacciones estén disponibles como objetos del kernel. Una **transacción** es un grupo de operaciones que poseen las propiedades **ACID** (*atomicidad, coherencia, aislamiento y durabilidad*). Estas propiedades garantizan que el software pueda controlar errores inesperados, ya que simplemente se puede anular una transacción cuando una situación inesperada impide una finalización correcta. La infraestructura de transacción garantiza que todos los efectos de la transacción anulada se revienten devolviendo los datos a un estado coherente. Por lo tanto, un sistema transaccional permite una recuperación correcta de los errores del sistema.

5.4. Manejo de errores

Cuando se produce un error en Windows, el código ampliado y el resultado se registran como un evento informativo en el registro de la aplicación mediante Informe de errores de Windows como evento. Cada vez que se produce un error se genera un código de error y se guarda en los archivos de registro (*el .log del sistema*).

5.5. Interrupciones

Las interrupciones del sistema aparecen como un proceso en el administrador de tareas, pero no son un proceso como tal, sino que son una representación del uso de la CPU que ocurren a bajo nivel en el sistema operativo. Las interrupciones pueden venir provocadas tanto por el software como por el hardware, lo que significa que cualquier aplicación es capaz de generar interrupciones en nuestro sistema. Por otro lado, los componentes y periféricos que tenemos conectados a nuestro ordenador también son capaces de generar interrupciones.

Una interrupción avisa al procesador de que tiene una tarea de máxima prioridad requiriendo que se interrumpa el código que se esté procesando en ese instante. Entonces el procesador suspende dicha actividad, guarda su estado, y ejecuta una función llamada gestor de interrupciones para gestionar el caso.

Después de esto el procesador maneja la nueva solicitud, y cuando el gestor de interrupciones ha terminado su trabajo el procesador continúa desde el punto en el que se quedó con el proceso que estaba en ejecución.

6. Subsistemas de ambiente protegido y APIs

Dos de los objetivos de Windows son la personalidad y compatibilidad. Esto ha sido logrado gracias a los subsistemas de ambiente protegido. La personalidad expone múltiples conjuntos de interfaces de programa de aplicación (*API*) y puede actuar eficazmente como si fuese un sistema operativo diferente. Windows viene con una personalidad POSIX y OS/2 además de sus personalidades Win32, Win16 y DOS.

Los subsistemas de ambiente protegido actúan como los mediadores entre las aplicaciones de modo usuario y el ejecutor de Windows. Cada subsistema trabaja independientemente a los otros subsistemas.

6.1. API

Una API es una abreviatura de Application Programming Interfaces, o lo que es lo mismo, interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software.

Las API otorgan flexibilidad, simplifican el diseño, la administración y el uso de aplicaciones y ofrecen oportunidades de innovación. Las API suelen ser consideradas como contratos, con documentación que representa un acuerdo entre las partes.

6.2. Win32

Es el subsistema nativo y primario de Windows. Las bases para este sistema es el conjunto de APIs de Win32. Muchas de estas APIs son extensiones directas de sus homólogos 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.

Este subsistema es el encargado de toda la entrada y salida. Posee el control de la pantalla, el teclado y el ratón. Cuando otros subsistemas necesitan utilizar estos dispositivos piden los servicios a Win32.

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

- Permite que los programas hechos con DOS funcionen sin modificación.
- Permite ejecutar aplicaciones de Windows de 16 bits sin modificación.
- Protege al sistema y otras aplicaciones de 32 bits de las interferencias con las aplicaciones de 16 bits y DOS.
- Permite a las plataformas RISC (*Reduce Instruction Set Computer*) ejecutar aplicaciones de Win16 y DOS.
- Suministrar un mecanismo para compartir datos entre aplicaciones de Win32 y Win16.

El subsistema Win32 es el más importante, ya que atiende no solo a las aplicaciones nativas de Windows, sino que para aquellos programas ajenos a este subsistema reconoce su tipo y los lanza hacia el subsistema correspondiente. Sin embargo, en el caso de que la aplicación sea MS-DOS o Win16, lo que hace es crear un nuevo subsistema protegido. De esta forma la aplicación se ejecutaría en el contexto de un proceso llamado VDM (*Virtual DOS Machine*), que no es más que un simulador de un ordenador funcionando bajo MS-DOS.

Las **aplicaciones de Win32** son aquellas que ofrecen el soporte para una arquitectura x86, que es el método en el que se programan todas las aplicaciones. La extensión de estas aplicaciones viene dada por un .exe o .msi. Son aplicaciones que en su gran mayoría podemos abrir varias instancias en el mismo equipo. Estas aplicaciones solo funcionan con procesadores Intel o AMD. La gran mayoría de estas aplicaciones tienen permisos de administrador.

6.3. WinRT

WinRT es la abreviación de *Windows Runtime*, un modelo de programación implementado por Microsoft que facilita el desarrollo de aplicaciones software con estilo Metro para Windows 8. Este modelo soporta tanto C++, como lenguajes de código como JavaScript o VB.NET y C# (*lenguajes de .NET*). Lo bueno que presenta WinRT es que permite escribir aplicaciones tanto para procesadores Intel, como para procesadores ARM.

WinRT es una API nativa no administrada basada en **COM** (*Component Object Model*), que podrá ser definida como la plataforma encargada de servir aplicaciones Metro. De hecho, WinRT ha sido diseñada para trabajar únicamente en sistemas operativos como Windows 8.

WinRT puede beneficiarse del diseño de aplicaciones Metro manejando estas desde un punto de vista gestual, lo que nos facilita utilizar dispositivos externos usando gestos.

Otro aspecto a tener en cuenta respecto al desarrollo de aplicaciones **Metro** basadas en WinRT es que todas las aplicaciones son asíncronas, es decir, que el tiempo de respuesta entre la aplicación y el usuario debe ser muy pequeño.

La ejecución de una aplicación Metro se hace por su parte en lo que se conoce como **SandBox** (*cada aplicación es responsable de sus recursos*). Cada aplicación sólo puede acceder a su parcela de memoria, asignada para que esa aplicación se encargue de gestionarla de forma independiente al resto de la memoria del sistema. De esta manera, cada aplicación se aísla y no interfiere en el resto de las aplicaciones, de modo que si ocurriese un error inesperado, éste sólo afectaría a la aplicación y no al sistema ni a ninguna otra aplicación que se estuviera ejecutando en el sistema.

6.4. .NET

.NET es una plataforma de desarrollo de código abierto para compilar muchos tipos de aplicaciones tales como aplicaciones web, aplicaciones nativas en la nube, juegos, etc. Comparte la funcionalidad entre diferentes aplicaciones mediante el uso de bibliotecas de clases. Con .NET, el código y los archivos del proyecto presentan el mismo aspecto con independencia del tipo de aplicación que se compile.

.NET es multiplataforma, es decir, se pueden crear aplicaciones .NET para muchos sistemas operativos, no solo Windows, y es compatible con todo tipo de procesadores.

Lo que .NET pretende es poder desarrollar aplicaciones y sistemas que sean independientes de la arquitectura física y del sistema operativo sobre los que se ejecuta.

Desde un punto de vista comercial, .NET podría considerarse como la alternativa de Microsoft para competir con Java.

La construcción de aplicaciones usando .NET trae consigo una serie de **beneficios**:

- Disminuir el tiempo de desarrollo de los proyectos.
- Poder usar las funcionalidades ya implementadas (*adaptación*).
- Simplificar el mantenimiento de las aplicaciones.
- Reducir los costes.

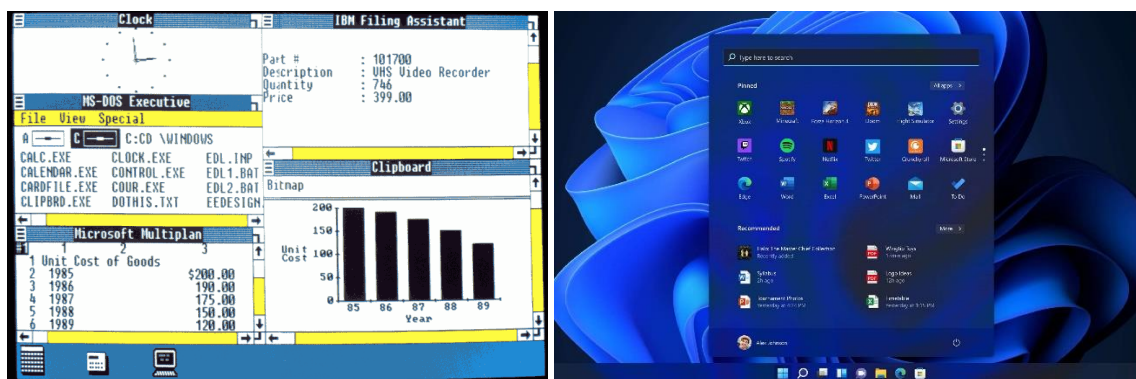
Al ser una plataforma de propósito general se han desarrollado varias implementaciones en función del dispositivo a usar, como son:

- **.NET Framework**: es compatible con páginas web, servicios y aplicaciones de escritorio de Windows, entre otros.
- **.NET Core**: utilizada para sitios web, servidores y aplicaciones de consola en múltiples sistemas operativos.
- **Xamarin/Mono**: permite ejecutar aplicaciones en los principales sistemas operativos móviles.

7. Interfaz gráfica

La **interfaz de usuario** es el conjunto de puntos de contacto del usuario y el ordenador, e incluye todo lo que el sistema emite o muestra (*salida o "output"*) y todo lo que el sistema recibe (*entrada o "input"*).

Con la primera versión de Windows se daba paso a las interfaces gráficas mediante ventanas (*de ahí su nombre*) que aceleraban el trabajo y mejoraban la experiencia del usuario, aunque no aportaban mucha funcionalidad.



Windows 1 (1985)

vs

Windows 11 (2022)

Las siguientes versiones introdujeron algunos cambios como:

- Iconos de escritorio
- Mejores gráficos
- La posibilidad de superponer ventanas
- Usar atajos de teclado
- Algunos juegos como el Solitario, Buscaminas, ...

Pero no fue hasta **Windows 95** que las interfaces gráficas de usuario comienzan a coger la forma que servirá de plantilla para las futuras versiones del sistema operativo. Esta versión estaba más orientada al usuario, con un rediseño completo de la interfaz y nuevas herramientas, como la barra de tareas o el botón de inicio que usamos en nuestro día a día, pero que en su momento constituyeron algo revolucionario.

Nuevas funcionalidades comenzaron a llegar con las siguientes versiones, que seguían ampliando la funcionalidad y la comodidad del consumidor. Un ejemplo de ello es el poder minimizar una ventana que no surgió hasta 1998 con **Windows 98**.

En 2001 con **Windows XP** sale a la luz una nueva interfaz gráfica con un manejo más sencillo, la capacidad de agrupar aplicaciones en la barra de tareas, usar estilos visuales para cambiar la interfaz y el clásico fondo de escritorio con colinas verdes y un cielo azul.

Pasando a la siguiente versión, aunque no fuera muy exitosa, a **Windows Vista** le caracteriza una cosa, y es su nueva interfaz gráfica. Permitía la transparencia de ventanas y con la combinación de Win + Tab, se mostraban en 3D las ventanas que seguían abiertas pudiendo moverte entre ellas.

Llega entonces **Windows 7**, que siguió con lo definido en las anteriores versiones mejorando nuevamente la barra de tareas y añadiendo la posibilidad de utilizarse en pantallas táctiles.

Con **Windows 8** se modificó mucho la interfaz de usuario con el fin de simplificar el manejo en pantallas táctiles, se cambiaron los iconos de escritorio por las Apps ... Y además de una cierta transformación en el explorador de archivos permitiendo tener funciones más a mano, algo que no gustó demasiado al público.

Llegando al final tenemos **Windows 10**, con una interfaz mucho más moderna que al igual que sus hermanas pequeñas esta versión se centra en gran parte en mejorar la experiencia del usuario, permitiendo personalizar la barra de tareas de múltiples formas (*color, tamaño, localización en la pantalla ...*), un nuevo menú que nos muestra las aplicaciones más utilizadas en el PC, acceso a herramientas y directorios. Una característica que hay que destacar son los escritorios virtuales, ofreciendo la posibilidad de tener varias áreas de trabajo o escritorios donde tener diferentes aplicaciones, carpetas o archivos abiertos.

Por último, **Windows 11** llega con una interfaz completamente rediseñada y afable cara al público. Se dejan atrás los clásicos bordes cuadrados de Windows y se redondean, así como la centralización de la barra de tareas, muy similar al entorno de MacOS. Se han rediseñado los clásicos widgets de Windows 10 y se ha rediseñado la forma de redimensionar las ventanas abiertas.

8. Conclusiones

Mediante la redacción de este trabajo, se han abordado diversos temas tales como la organización interna de Windows, su visibilidad hacia el usuario y su evolución a lo largo del tiempo.

Con todos estos temas se ha podido ahondar en los conocimientos de esta herramienta que, sin darnos cuenta, utilizamos tanto en nuestro día a día y nos hemos dado cuenta de la amplia dificultad y el trabajo que este sistema posee.

Este trabajo pretende transmitir esos conocimientos al lector e inspirarle a ahondar más por su cuenta, proporcionando toda la información consultada para que, si le interesa, la revise, la confirme y la amplíe.

Por otra parte, los integrantes del grupo nos hemos interesado en gran medida en la estructura interna del sistema que, aunque parta de ciertas ideas originarias de UNIX, se diferencia de una forma radical de los sistemas Linux. Además, nos ha impactado la evolución que ha ido contemplando este sistema, que pese a haber introducido infinidad de mejoras, siempre ha mantenido su clásica estética.

Nosotros, desde nuestro punto de vista, recomendamos al lector inmiscuirse en el ámbito interno de Windows ya que averiguará una infinidad de información relevante que incluso podrá aplicar a su día a día en el manejo de este sistema operativo.

9. Bibliografía

2. Historia

<https://news.microsoft.com/es-es/2015/04/06/historia-microsoft-40-aniversario/>
https://es.wikipedia.org/wiki/Historia_de_Microsoft
<https://softwarelab.org/es/windows-historia/>

3. Características generales

https://www.ecured.cu/Microsoft_Windows
<https://softwarelab.org/es/windows-historia/>
<https://www.informaticogranada.com/caracteristicas-windows-linux-linux-versus-windows/>
<https://ejemplos.net/20-caracteristicas-de-windows/>
<https://10caracteristicas.com/windows/>

4. Versiones de Windows

<https://www.adslzone.net/esenciales/windows-10/versiones-windows/>
<https://www.campusmvp.es/recursos/post/La-evolucion-de-Windows-de-Windows-1-a-Windows-10.aspx>

5. Estructura interna del sistema

<https://es.slideshare.net/eyes666/kernel-de-windows>
<https://www.muycomputer.com/2019/04/16/microsoft-explica-como-es-windows-10/>
<https://www.about espanol.com/que-son-los-drivers-o-controladores-y-como-conseguirlos-3507888>
<https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/introduction-to-ktm>
<https://docs.microsoft.com/es-es/windows/win32/ktm/about-ktm>
<https://docs.microsoft.com/es-es/windows-hardware/drivers/kernel/when-to-use-kernel-mode-ktm>
<https://docs.microsoft.com/es-es/windows/deployment/upgrade/windows-error-reporting>
<https://www.ibm.com/docs/es/integration-bus/10.0?topic=extension-errors-exception-handling>
<https://www.vichauter.org/general/las-interrupciones-del-sistema>
<https://www.monografias.com/trabajos26/arquitectura-windows/arquitectura-windows>

6. Subsistemas de ambiente protegido y APIs

<https://es.slideshare.net/eyes666/kernel-de-windows> diapositiva 12
<https://docs.microsoft.com/es-es/windows/win32/apiindex/windows-api-list>

<https://www.xataka.com/basics/api-que-sirve>
<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
<https://www.monografias.com/trabajos26/arquitectura-windows/arquitectura-windows>
<https://www.elgrupoinformatico.com/noticias/que-win32-t81462.html>
<https://geeks.ms/jorge/2012/06/11/qu-es-winrt/>
<https://techlib.net/blog/que-es-winrtwindows-runtime/>
<https://docs.microsoft.com/es-es/dotnet/core/introduction>
<https://www.emagister.com/blog/que-es-el-net-para-que-sirve/>
<https://orientacion-laboral.infojobs.net/que-es-net-para-que-sirve>

7. Interfaz gráfica

<https://www.campusmvp.es/recursos/post/La-evolucion-de-Windows-de-Windows-1-a-Windows-10.aspx>
<https://www.uv.mx/apps/afbgcursos/Cursos%20anteriores/CBApoyoModalidadPresencial/internet/leccion1.html>
<https://internetpasoapaso.com/interfaz-windows-10/>