

# Microsoft Windows y Microsoft .NET



Ingeniería de Software

Méndez Martínez Araceli Nancy  
Rios Alvarado Ana Bertha



# Contenido

- Historia de Windows
- .NET
- Estandarización ECMA

# Historia

- Antecedentes .- MSDOS

- 1985 .- Windows 1.0

Interfaz Grafica de Usuario para MS-DOS imitando el MacOS Apple

- 1987.- Windows 2.0

Inclusión de aplicaciones graficas como Excel y Word

# Historia

- 1990 .- Windows 3.0

Primera versión popular

Mejoras en las capacidades multitareas

Se beneficio de las capacidades graficas de la PC

- 90's .- Windows 3.1

Soporte multimedia

# Historia

- 90's .- Windows 3.11  
Windows para trabajo en grupo, mejoras en los protocolos de red (Microsoft desarrollaba Windows NT)
- 1992 .- Windows NT 3.1 (versión beta)



# Historia

- Agosto 1995 .- Windows 95 (Chicago)
- **Windows 95**
  - 1) Instalación integrada entre MS-DOS y Windows
  - 2) Introducía un subsistema en modo protegido

API Win32



# Historia

- 1998 .- Windows 98
- **Windows 98 :**
  - Sistema de ficheros FAT 32
  - Soportaba particiones mayores a los 2GB
- 1999 .- Windows 98 Second Edition
  - Capacidad de compartir entre varios equipos una conexión a Internet
- 2000 .- Windows ME
  - Copia de Windows 98 con más aplicaciones añadidas

# Historia

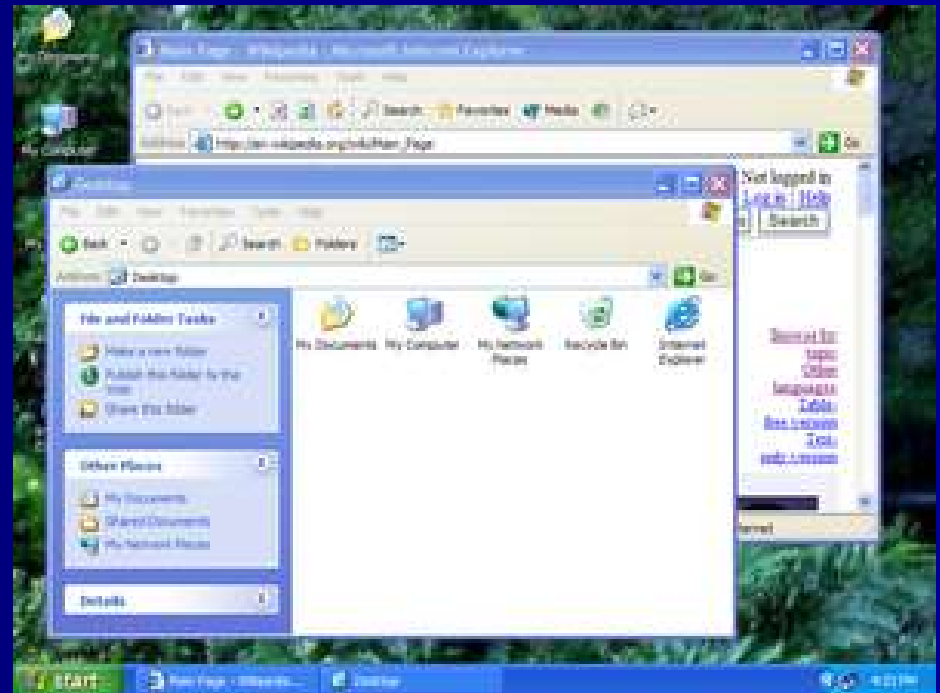
- 2000
- Windows 2000  
Nueva versión de Windows NT  
Gran cantidad de servicios de red





# Historia

- 2001 .- Windows XP  
Unión de Windows NT/2000 y Windows 3.1/95/98/SE  
Usa el nucleo de Windows NT  
Nueva Interfaz,  
Mayores capacidades multimedia



# Historia

## Sistemas Operativos de Microsoft

### 16 bits

- MS-DOS (última liberada oficialmente fue la 6.22)
- Windows 1.0
- Windows 2.0
- Windows 3.0
- Windows 3.1
- Windows 3.11
- Windows 95
- Windows 98
- Windows 98 SE
- Windows ME

### 32 bits

- OS/2 1.3
- OS/2 2.0
- Windows NT 3.1
- Windows NT 3.5
- OS/2 3.0
- OS/2 4.0
- Windows NT 3.51
- Windows NT 4.0
- Windows 2000
- Windows XP

# El infierno de las DLLs

- Termino dado a cualquier problema originado por la dificultad de administrar las instalaciones de DLLs en un sistema operativo.

# Abarca

- Conflictos entre versiones de DLLs.
- Dificultad en obtener un gran número de DLLs.
- Tener muchas copias innecesarias de diferentes versiones de DLLs.

# Problemas

- Varias aplicaciones pueden compartir la misma DLL.
- Las aplicaciones pueden introducir una versión diferente de una DLL en particular presente en el sistema (dos posibilidades).
- Aumento de tiempo de ejecución de programas.

# Problemas

- No hay manera de distinguir las DLLs necesarias de las innecesarias.

# Suceso

- El infierno de las DLLs fue un fenómeno muy común tan pronto se liberaab una nueva versión de windows.
- Microsoft tuvo limitados servicios para administración de archivos de sitema y manejo de versiones.

# Medidas para evitar el problema

- Tener una autoridad central para distribuir la DLL.
- Usar ligas estáticas o crear un nuevo paquete con un nombre diferente para la librería.
- Diseño apropiado de software (ligado estático).



# Motivo para .NET

2001 Microsoft introduce el framework .NET para introducir su propio paquete de desarrollo llamado assemblies.

# Características de la solución

- Los usos deben ser auto-descritos, quita dependencia en el registro.
- La información de la version debe ser registrada.
- Se debe recordar cuando un uso funciona con éxito, se debe recordar el sistema de componentes incluyendo las versiones que trabajaron juntas.

# Microsoft .NET



- Antecedentes
- ¿Qué es .NET?
- Infraestructura
- Lenguajes
- C# (c sharp)

# Antecedentes

- El producto existente de Microsoft J++ fue transformado en los principios de .NET

# ¿Qué es .NET?

- .NET es un proyecto de Microsoft para crear nuevas plataformas de desarrollo de software.
- Incluye muchas tecnologías que están diseñadas para facilitar el rápido desarrollo de aplicaciones de Internet e Intranet.
- Fue liberado en 2002

# Infraestructura

- .NET es una plataforma independiente que integra:
- Transparencia en la red
- Desarrollo rápido de aplicaciones
- Seguridad

# Estructura

- Infraestructura del Lenguaje Común (CLI) : maquina virtual y librería de clases
- Lenguaje Común Intermedio(CIL): Lenguaje orientado a objetos assembly

# Lenguajes .NET

- La CLI esta diseñada para proveer soporte para cualquier lenguaje de programación orientado a objetos
- Microsoft y otros vendedores proveen versiones de .NET en muchos lenguajes



# Lenguajes de .NET

- Lenguajes de construcción
  - C#
  - Visual Basic.NET
  - Managed C++
  - J#, Java y J++
  - Jscip.NET

.NET soporta alrededor de 40 lenguajes

# C # (C sharp)

- Es un lenguaje de programación orientado a objetos desarrollado por Microsoft como parte de la iniciativa .NET
- Esta basado en C++ y Java

# C #

- Las características de este lenguaje permiten interoperabilidad con subrutinas escritas en otros lenguajes de programación.

# C #

- El código es organizado en un conjunto de namespaces
- Los namespaces son grupos de clases con funciones similares
- Ejemplos:
  - System.Drawing para gráficos
  - System.Collections para estructuras de datos

# C #

- Un nivel de organización dado por este lenguaje es el concepto de assembly.
- Un assembly puede ser un solo archivo o un conjunto de archivos ligados a través de un .exe, los cuales contienen namespaces y objetos.

# C #

## Hello world example

```
public class ExampleClass {  
    public static void Main() {  
        System.Console.WriteLine("Hello  
world!");  
    }  
}
```

# .NET vs J2EE

- .NET esta disponible para plataformas Windows y Java esta disponible en muchas plataformas

# .NET vs COM

- COM fue la versión previa de tecnologías de software de componentes.
- .NET puede envolver y desenvolver los objetos COM



# Estandarización

- Microsoft ha sometido una parte de las especificaciones de .NET a ECMA para crear una estandarización.

# Plataforma de desarrollo mono

- Proyecto open source.
- Conjunto de herramientas compatibles .NET, compilador C#, CLR...para cobrir sobre GNU/Linux, Mac OS X y Windows.
- Microsoft hizo versión de .NET para FreeBSD(Rotor).

# Plataforma de desarrollo mono

- MV de mono contiene maquinas JIT para procesadores x86, SAPARC, Power PC(32 bits) y x86-64, SAPARC (64 bits). Para otros interprete.
- Versiones: 1.0.6(estable), 1.1.4 (desarrollo), 1.1.x-series.

# Plataforma de desarrollo mono

- MV de mono puede compilar justo a tiempo o precompilar el código a código nativo.
- .NET portable (DotGNU).