



Introducción a .NET compact framework

COMPUTACION EN LA RED
Daniel Díaz Sánchez

SISTEMAS OPERATIVOS DE DISPOSITIVOS MÓVILES

Introducción



Dispositivo móvil

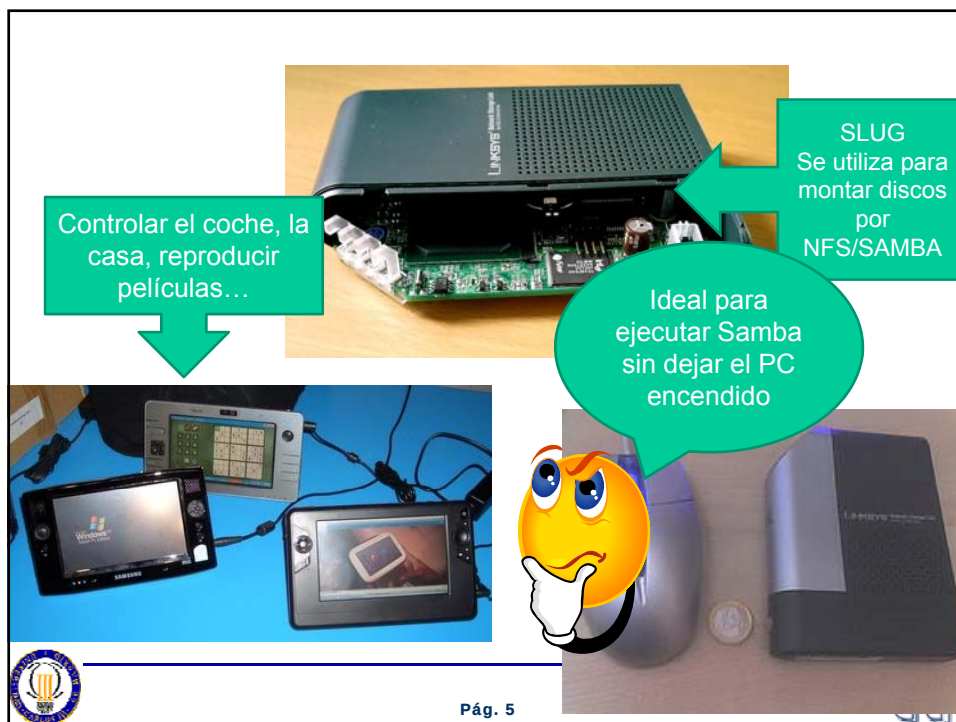
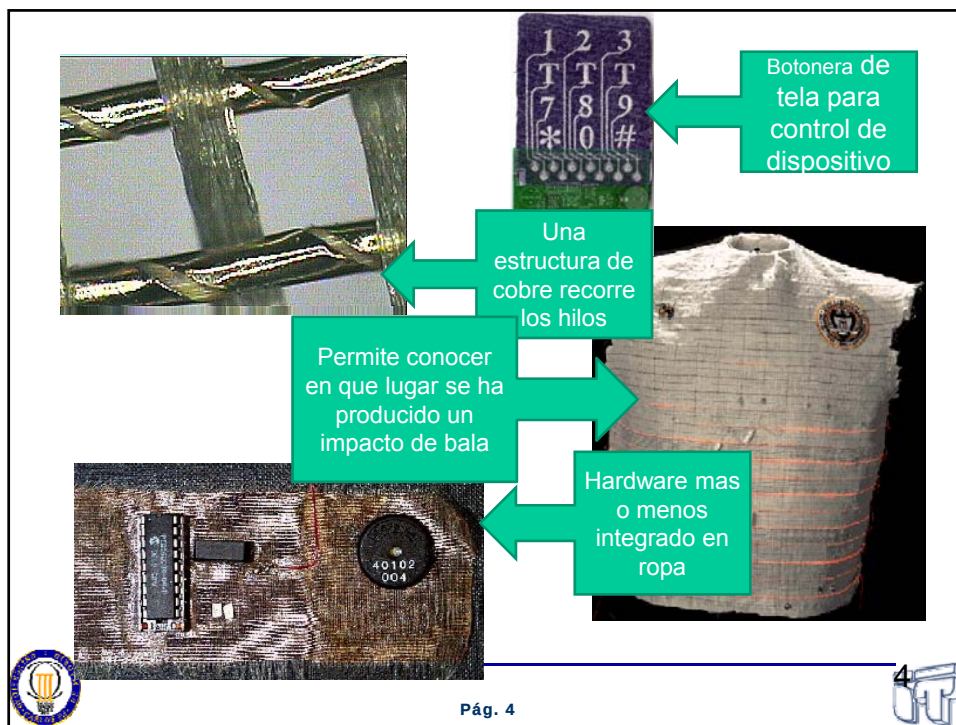
- En número de dispositivos personales prolifera
- Se conjugan:
 - Mayor capacidad de proceso, almacenamiento, capacidad de comunicación
 - Menor consumo (Mayor autonomía), menor peso, menor coste



En un futuro...

- Podríamos llegar a disponer de:
 - Ropa inteligente
 - Terminales móviles con la capacidad de un PC
 - Comunicación con dispositivos de entrada y salida inalámbrica
 - Pantallas
 - Teclados, Ratones
 - Redes de pares
 - Servicios de banda ancha inalámbrica...





Dispositivos móviles avanzados

- Basados en Symbian
 - Nokia, Siemens
- Basados en Windows Mobile (CE)
 - Motorola, Samsung, Qtek, HP, Dell
- Basados en Linux
 - Zaurus, Nokia
- Basados en Palm
 - Pierden protagonismo



FAMILIA WINDOWS MOBILE

Familia Windows Mobile



Microsoft RoadMap

- Dispositivos conocidos
 - Reproductores MP3 (Apple iPod)
 - Móviles (Symbian, Smartphones)
 - Personal Digital Assistants (Palm, MS, Zaurus)
 - Consolas (xBox, Pstation...)
 - Tablet PCs, Portátiles, PC, Pantallas y proyectores
- Un gran número de ellos son portátiles otros móviles, otros sirven de pasarelas:
 - Pasarelas de servicios: decodificadores, servicios en el coche, autobuses, aviones...



Microsoft RoadMap

- Microsoft tiene en cuenta esta diversidad
 - Soluciones adaptadas a las limitaciones
- Mantener un modelo de S.O. común (cambia con Windows Phone)
 - Curva de aprendizaje rápido
 - Manejar un PC implica manejar una PDA, reproductor de MP3, tablet-pc, consola...
- Mantener un modelo de desarrollo común
 - Portabilidad de programadores
 - En algunos casos portabilidad de aplicaciones



Microsoft RoadMap



Fuente Microsoft
Pag. 10



Microsoft RoadMap

- Sistemas operativos embebidos de MS
 - Windows CE (Compact Edition)
 - Tiempo real
 - Soporta PAN, LAN, WAN
 - Alta seguridad
 - Windows XP embedded
 - Basado en WXP
 - Soporta cerca de 10.000 modulos configurables
 - Windows XP embedded for point of service
 - TPVs
 - Smarter Retailing (Smarter shopping, smarter selling, smarter operations)



Pág. 11

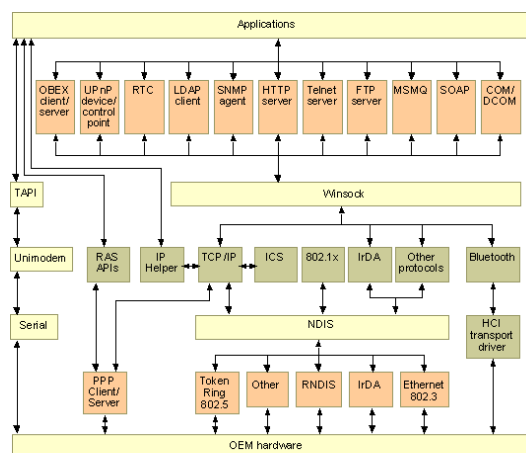


¿Qué es Windows CE y Windows Mobile?

- Windows CE es un sistema operativo enorme
- Solo el sistema de comunicaciones incluye:
 - Sistemas TAPI (Telephony API)
 - Soporte de sockets completo
 - Soporte NDIS
 - Servidores http, telnet, ftp, SOAP, COM/DCOM



¿Qué es Windows CE y Windows Mobile?



¿Qué es Windows CE y Windows Mobile?

- WCE se particulariza para cada dispositivo
- Digital Media Receiver, reproductor de música y video
- Enterprise terminal: TPV
- Enterprise Web Pad
- Gateway: Plataforma para casa
- Industrial controller: Interfaces humanos de control
- Internet appliance
- IP phone
- Mobile Handheld
- Mobile phone
- Set top box
- Tiny kernel
- Windows Thin Client



¿Qué es Windows CE y Windows Mobile?

- Windows Mobile es un perfil de particularización (Mobile Handheld)
- Es un perfil para PDAs
- En el caso de los móviles (Smartphones) se utiliza perfil de Mobile Phone



Fotos de terminales de HP, T-Mobile, BSQUARE y SAMSUNG



¿Qué es Windows CE y Windows Mobile?

- Windows CE .NET (4.0/4.2/4.22)
 - .NET compact framework
 - Gestión avanzada de la batería
 - Windows Mobile 2003 (4.22)
 - Pocket PC 2003 second edition (4.22)
 - Pocket PC 2003 (4.0)
 - Smartphone (4.0/4.2/4.22)
- Windows CE 5.0 / CE 6.0
 - Apagado completo, almacenamiento persistente
 - En fase de despliegue
 - Windows Mobile 2005



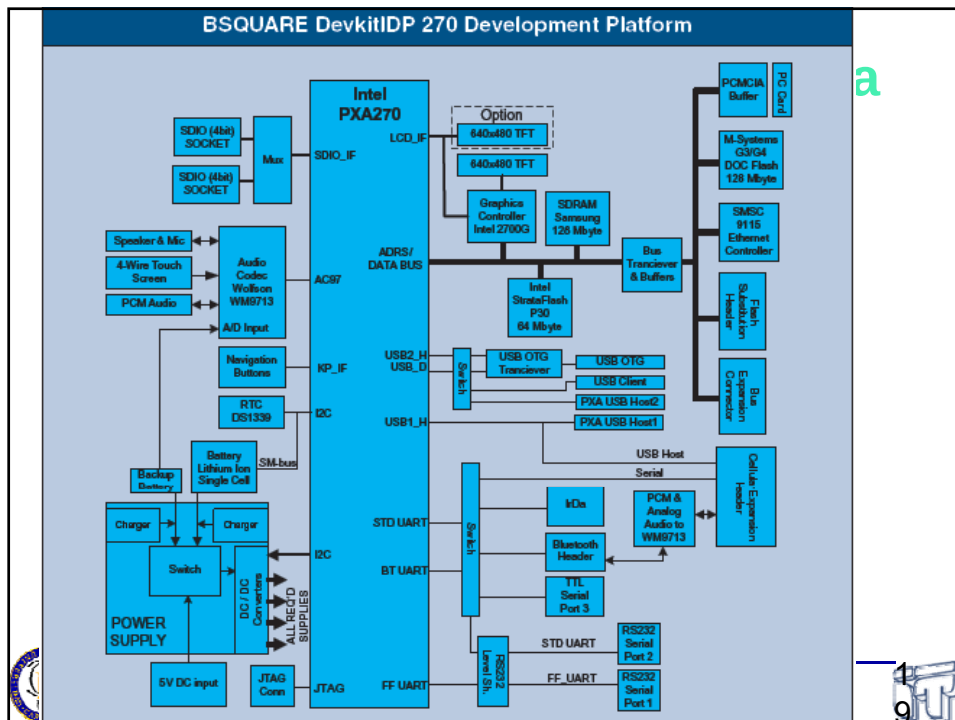
¿CÓMO CREAR UN SISTEMA OPERATIVO?

Herramientas



Desde el Hardware al Sistema Operativo: Platform Builder

- Windows CE es personalizable
 - Se adapta al hardware concreto
 - Se adapta al perfil o rol del dispositivo
- Partiendo del hardware, por ejemplo el caso de BSquare DevKit PXA270
 - Seleccionar un perfil de sistema operativo
 - Agregar todos los drivers necesarios
 - Personalizar todo lo que se desee
- Veamos un ejemplo



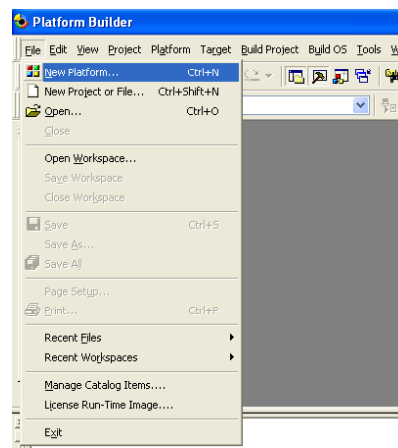
Desde el hardware al sistema operativo

- Platform Builder © es una herramienta que permite construir un S.O.
- Dado el hardware un fabricante puede:
 - Compilar un sistema operativo completo
 - Seleccionar las aplicaciones
 - Seleccionar las características
 - Desarrollar drivers a terceros
 - Desarrollar aplicaciones C/C++ (sin ALT/MFC)

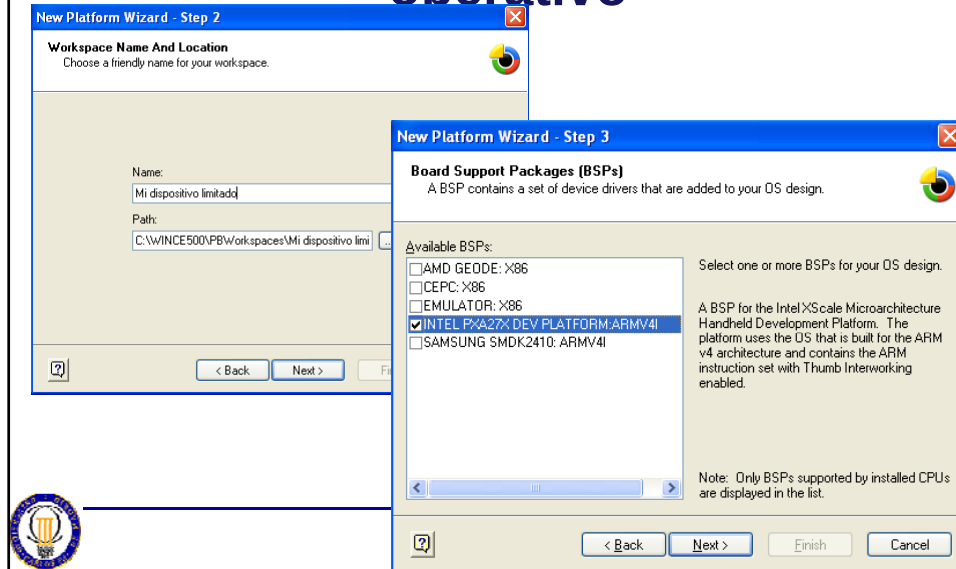


Desde el hardware al sistema operativo

- Abrimos Platform Builder
- Nueva plataforma
- Damos un nombre



Desde el hardware al sistema operativo



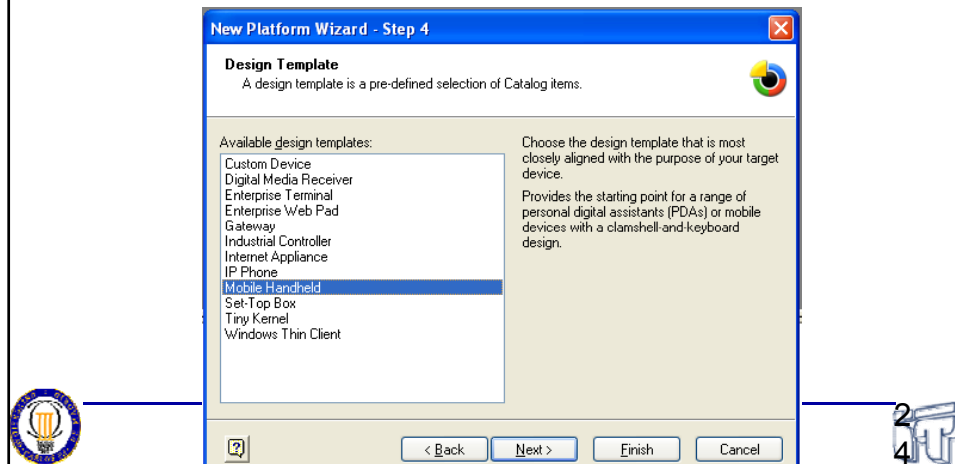
Desde el hardware al sistema operativo

- BSP: Board Support Package
 - Contiene una descripción del hardware
- Este archivo de descripción de hardware
 - Lo proporciona el fabricante
 - Otros son modelos de referencia
 - Se puede utilizar plataformas parecidas entre si para desarrollar drivers



Desde el hardware al sistema operativo

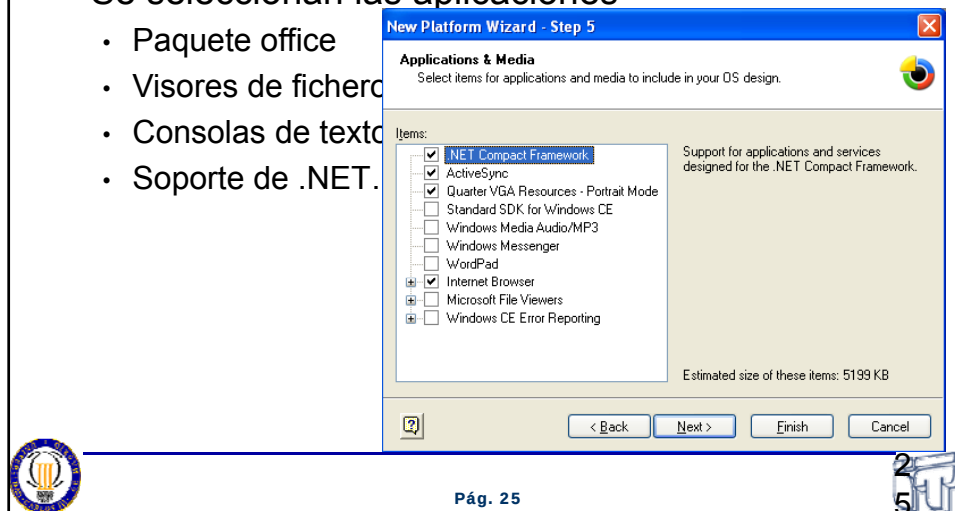
- Seleccionamos el perfil para ese hardware



Desde el hardware al sistema operativo

- Se seleccionan las aplicaciones

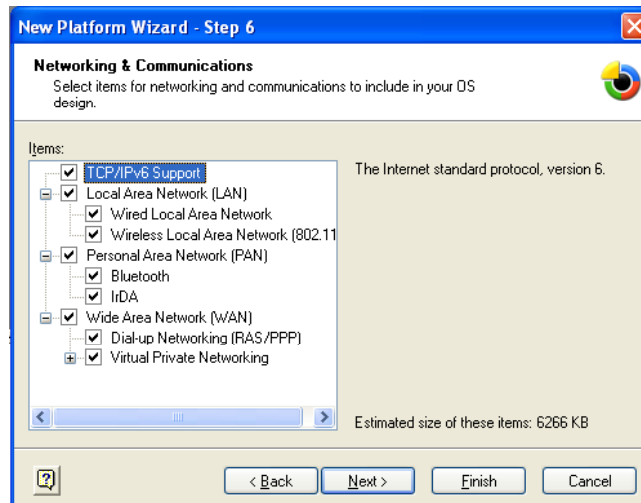
- Paquete office
- Visores de ficheros
- Consolas de texto
- Soporte de .NET.



Desde el hardware al sistema operativo

Soporte de red

- IPv4/IPv6
- LAN, WAN, PAN



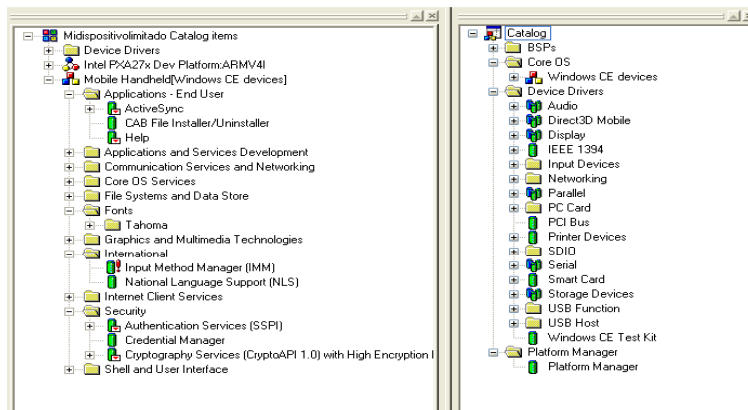
Desde el hardware al sistema operativo

- Una vez finalizados esos pasos ya tenemos descrito a grandes rasgos el sistema operativo
- Es posible personalizarlo aun más y modificar el código de alguno de los módulos



Desde el hardware al sistema operativo

Se añaden controladores y aplicaciones



Desde el hardware al sistema operativo

- Finalmente el proceso de pruebas
 - Se utilizan PDAs y móviles de ingeniería



Intel PXA270 DevKitIDP



¿CÓMO PROGRAMAR EN DISPOSITIVOS EMPOTRADOS?

Herramientas



Pág. 30



Herramientas de desarrollo

- Herramientas de desarrollo
 - Embedded Visual C++
 - Visual Studio .NET y Visual Studio .NET 2003
 - Visual Studio 2005
 - Visual Studio 2008
 - Visual Studio 2010
- Seleccionando la herramienta adecuada

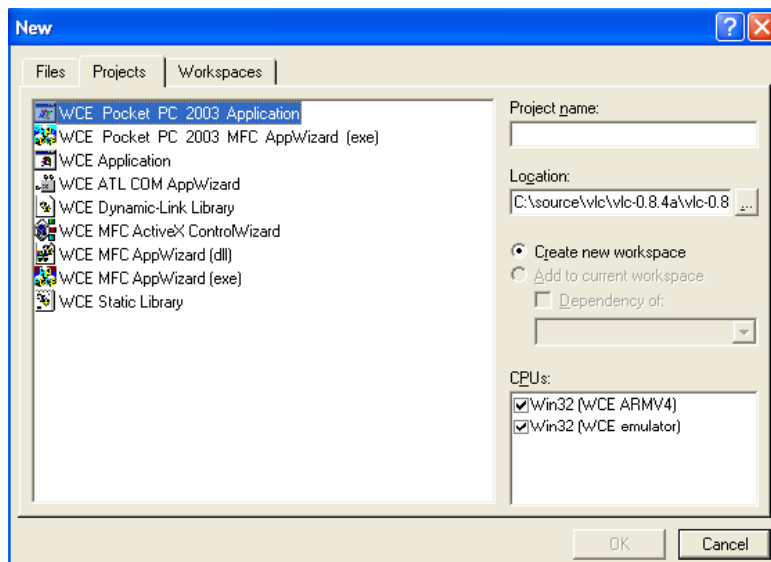


29/09/2010
Pág. 31



Embedded Visual C++ 4.0

- IDE gratuito de Microsoft
- Permite desarrollar aplicaciones en C/C++ para Windows CE
- Incluye soporte para ATL y MFC
- No permite programar en .NET
- Compatible con todas las versiones WCE hasta la versión 4.22



Programando en C/C++

- Los posibles tipos de proyectos:
 - WCE Pocket PC Application
 - Aplicaciones C/C++ win32
 - WCE MFC Appwizard
 - Asistente que genera parte del código . Utiliza MFC.
 - WCE dynamic link library
 - Bibliotecas dinámicas. Permiten compartir recursos
 - WCE Static library
 - Bibliotecas estáticas



Embedded Visual C++ 4.0

- Para cada plataforma se requiere un SDK
- Los SDKs incluyen:
 - Binarios (librerías)
 - Ficheros de cabecera .h
 - Ejemplos
- Los SDKs se distinguen:
 - Por CPU (ARM, MIPS...)
 - Por Plataforma (Pocket PC 2003, Second Edition)



Embedded Visual C++ 4.0

■ Herramientas útiles incluidas

- **nmake.exe**: herramienta make similar al GNU make de linux
- **dumpbin.exe**: permite ver el contenido de ficheros binarios, puntos de entrada de las DLLs, información y recursos contenidos en
- **clarm.exe**: compilador



Visual Studio .NET 2003

- IDE de pago
- Permite desarrollar aplicaciones en C/C++ para Windows XP / Server 2003...
- Sólo permite desarrollar aplicaciones para Windows CE en .NET compact framework
- Compatible con todas las versiones WCE que soporten .NET

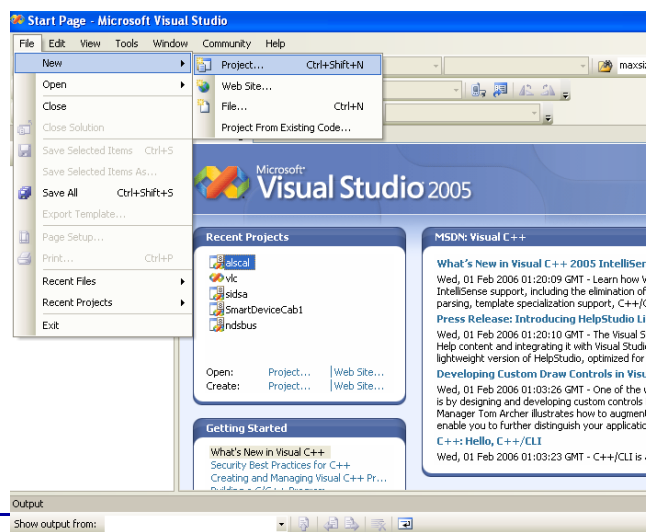


Visual Studio 2005

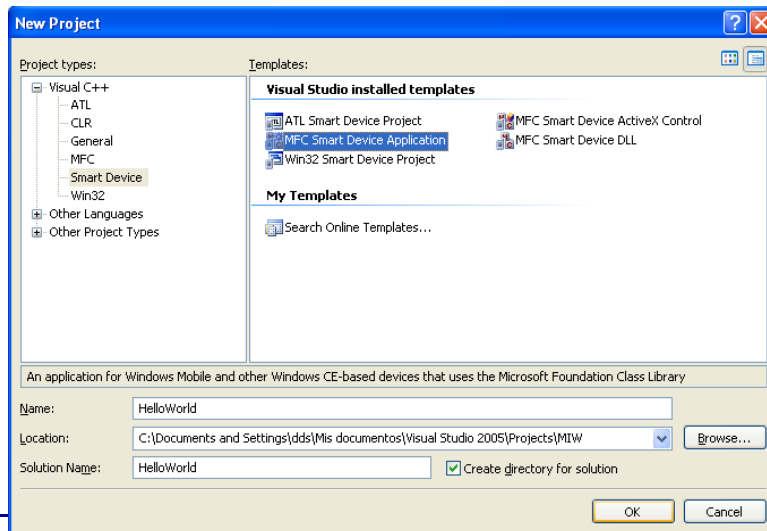
- IDE resultado de la sinergia de Embedded Visual C++ y Visual Studio
- Existen versiones “express” gratuitas
- Permite desarrollar para cualquier versión de Windows CE
 - En C/C++
 - En .NET compact framework
- Da soporte a Windows CE 5.0



Visual Studio 2005



Visual Studio 2005



Pág. 40

Visual Studio 2005

- Proyectos Visual C++ Smart Device permite desarrollar
 - Aplicaciones en C/C++ con soporte ATL, MFC
 - Aplicaciones en C/C++ para consola de texto
 - DLLs y librerías estáticas
- Proyectos Visual C# y Visual Basic para Smart Device permite desarrollar en .NET para cualquier versión de Windows CE

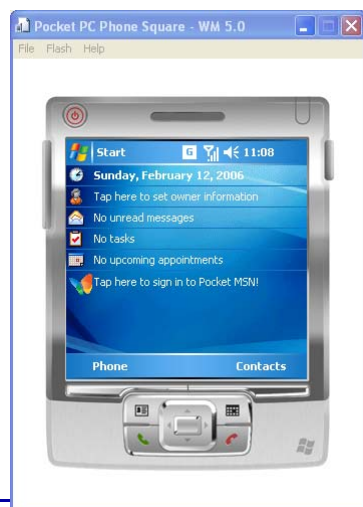
Pág. 41

Visual Studio 2005

- Emuladores
- Su número y tipo dependen de los SDKs instalados
- Ejecutan código en ARM directamente
- Las imágenes de emuladores incluyen:
 - Windows Mobile 5.0
 - PDA, PDA VGA, SmartPhone,
 - Windows Mobile 2003
 - PDA, PDA VGA, Smartphone
 - Emulador X86



Visual Studio 2005



Seleccionando la herramienta adecuada

- Aplicaciones y librerías C/C++
 - WCE 3.0 (Pocket PC 2002)
 - Embedded Visual C++ (Service Pack 4)
 - Visual Studio 2005 (Con los SDKs adecuados)
 - WCE 4.22 (Pocket Pc 2003, Windows Mobile 2003)
 - Embedded Visual C++ (Service Pack 4)
 - Visual Studio 2005 (Con los SDKs adecuados)
 - Platform Builder 4.0 (Sin soporte MFC/ALT)
 - WCE 5.0 (Windows Mobile 5.0)
 - Visual Studio 2005 (Con los SDKs adecuados)
 - Platform Builder 5.0 (Sin soporte MFC/ALT)



Seleccionando la herramienta adecuada

- Aplicaciones y librerías .NET
 - WCE 3.0 (Pocket PC 2002)
 - No soporta .NET
 - WCE 4.22 (Pocket Pc 2003, WMobile 2003)
 - Visual Studio .NET 2003
 - Visual Studio 2005
 - WCE 5.0 (Windows Mobile 5.0)
 - Visual Studio 2005



Seleccionando la herramienta adecuada

- Drivers

- WCE 3.0 (Pocket PC 2002)
 - Platform Builder
- WCE 4.22 (Pocket Pc 2003, Windows Mobile 2003)
 - Embedded Visual C++ (Service Pack 4): Algunos
 - Visual Studio 2005 (Con los SDKs adecuados): Algunos
 - Platform Builder 4.0 (Sin soporte MFC/ALT): Todos
- WCE 5.0 (Windows Mobile 5.0)
 - Visual Studio 2005 (Con los SDKs adecuados): Algunos
 - Platform Builder 5.0 (Sin soporte MFC/ALT): Todos



INTRODUCCIÓN A .NET

¿Qué es eso de .NET?. En que se diferencia de J2EE. Estructura de .NET.
¿Qué es C#?. En qué se diferencia de Java



¿Qué es .NET (dotNet)?

- Plataforma de desarrollo nueva:
 - Integración con redes, servicios web, XML...
 - Independencia con el hardware
 - **Rápido desarrollo de aplicaciones**
 - **Usable en Linux, Unix, Solaris, Palm...**
- Y para Microsoft
 - Plataforma horizontal para sus productos
 - S.O. (XP, W2003Server, PocketPC)
 - IIS (Internet Information Server)
 - Limpiar errores del API de Win32



29/09/2010
Pág. 48



¿.NET en Linux y otros?

- .NET se apoya en BCL
 - Base Class Library (ECMA-335, ISO/IEC 23271)
 - Define el conjunto de clases base para que .NET pueda ejecutarse en otros S.O.
- MS en Windows y MONO (Ximian) en:
 - MAEMO
 - Debian



29/09/2010
Pág. 49



Características

Interoperabilidad

- Permite la interacción con otras aplicaciones fuera de .NET.
Permite acceso a COM y librerías externas.

Fácil despliegue

- Evita conflictos de versiones
- Evita problemas de seguridad con nuevas instalaciones



Características (II)

Independencia del lenguaje

- Introduce el Common Type System (CTS)
 - **CTS** define todos los posible tipos y bloques de programación CIL (MSIL)
 - Por esta razón se pueden pasar instancias de objetos de un lenguaje a otro



Características (III)

Common Language Runtime

- Los programas se compilan (compilador cruzado) a lenguaje intermedio llamado *Common Intermediate Language (CIL)*. Microsoft lo llama **MSIL** (*Microsoft Intermediate Language*).
- Este lenguaje **nunca** se interpreta, sino que se compila en nativo antes de su uso (usando un compilador bajo demanda **JIT**).
- Todos estos conceptos juntos son el **Common Language Infrastructure**. Microsoft lo llama Common Language Runtime.



Características (IV)

Portabilidad

- Cualquier programa puede ejecutarse en otras implementaciones de .NET.
- Es posible por estandarización (ECMA/ISO):
 - Common Language Infrastructure, Common Type System, Common Intermediate Language, C#, C++/CLI

Seguridad

- Control automático para evitar buffer overflow, code injection...



Características (V)

Librería de clases básica

- Librería de clases con diferente funcionalidad, disponible para el CIL y por tanto para todos los lenguajes de .NET
- Es común, por lo que es más sencillo evitar errores (una única implementación para todos)
- Microsoft la llama Framework Class Library



Arquitectura de .NET

Tres bloques diferenciados

- Lenguajes (+20) de programación
 - Define CLI (Common Language Infrastructure)
 - C#, Visual Basic, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran, Cobol.NET y PowerBuilder. Java (IKVM).
- Biblioteca de Clases Base (BCL)
- Entorno Común de Ejecución: Common Language Runtime (CLR)



Common Language Runtime (I)

- Es una implementación de CLI
 - Common Language Infrastructure
- ¿Qué es CLI?
 - Entorno de ejecución virtual para que aplicaciones en diferentes lenguajes puedan ejecutarse en múltiples plataformas hardware y software.
 - El origen es COM (COM+): usar algo escrito en Visual Basic, C, C++ desde cualquiera de los otros lenguajes (COMPLICADO)



Common Language Runtime (II)

Historia de CLI:

- 2000: Microsoft presenta en ECMA (European Computer Manufacturers Association) la propuesta
- 2001: Microsoft+Intel+HP+IBM... finalizan el estándar
- 2003: ISO lo adopta.



Common Language Runtime (III)

¿Qué permite CLI?

- Escribir componentes interoperables
 - Independientemente de la plataforma subyacente
 - Independientemente del lenguaje
- Sistema unificado de tipos
- Entorno de ejecución con privilegios
- Código autocontenido y autodescriptivo
- Compilación bajo demanda (JIT)
- Proporcionar funcionalidad común (BCL)



29/09/2010
Pág. 58



Common Language Runtime (IV)

- Se compone de las siguientes partes:
 - Common Type System (CTS)
 - Define tipos comunes y permite integración e interacción de código escrito en varios lenguajes
 - Orientado a objetos
 - Common Language Specification
 - Define cómo debe ser el lenguaje intermedio y por tanto lo que deben cumplir los compiladores cruzados
 - Metadatos
 - Permiten que el código sea autocontenido
 - Virtual Execution Environment
 - Define qué es necesario para ejecutar el conjunto de instrucciones de Common Language

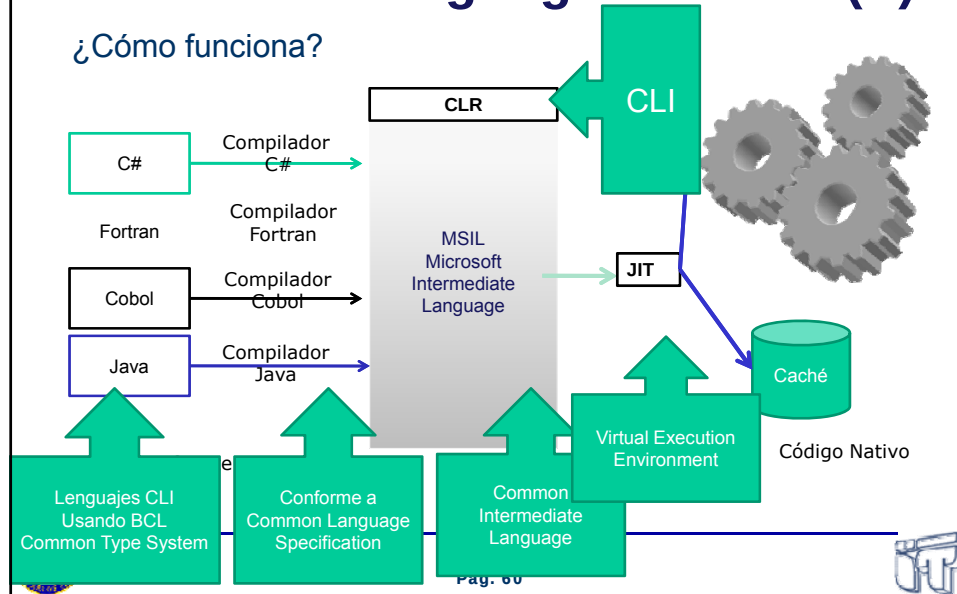


29/09/2010
Pág. 59



Common Language Runtime (V)

¿Cómo funciona?

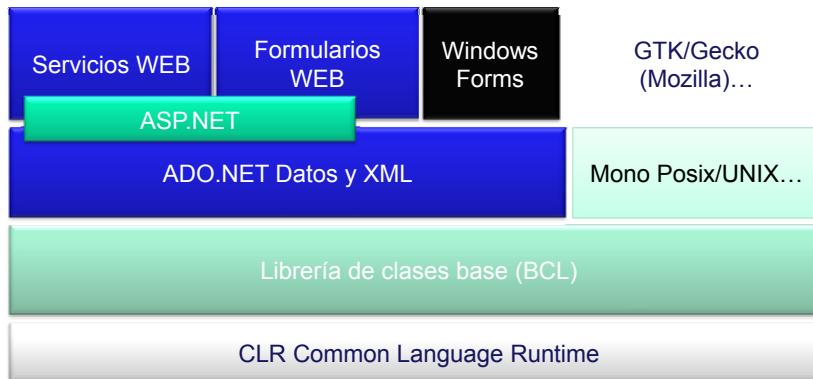


Common Language Runtime (V)



BCL: Base Class Library

- Es parte básica de la funcionalidad de .NET



29/09/2010
Pág. 62



BCL: Base Class Library (II)

System	System.Reflection
System.CodeDom	System.Resources
System.Collections	System.Runtime
System.ComponentModel	System.Security
System.Configuration	System.Text
System.Data	System.Threading
System.Diagnostics	System.Timers
System.Drawing	System.Web
System.Globalization	System.Windows.Forms
System.IO	System.Xml
System.Messaging	Microsoft.CSharp
System.Net	Microsoft.VisualBasic
	Microsoft.Win32



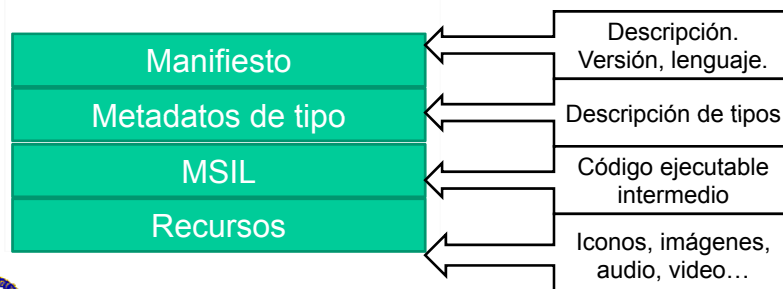
29/09/2010
Pág. 63



Ensamblados

- Son ficheros EXE o DLL que contienen toda la información de ejecución

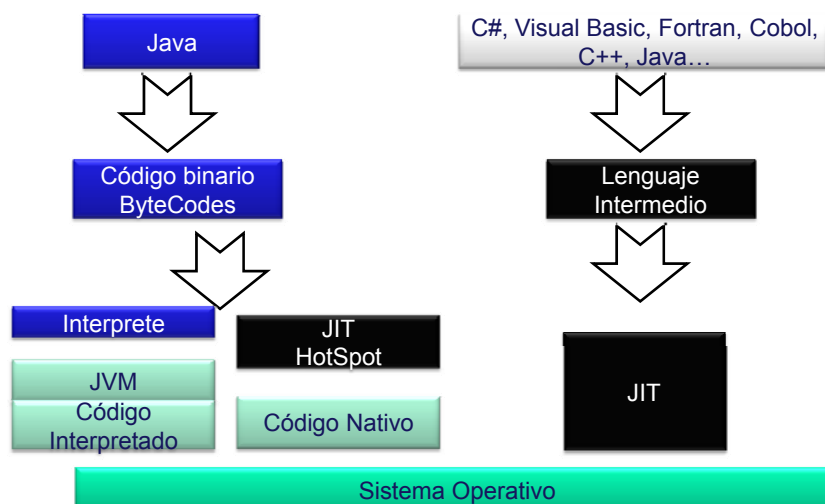
- EXE: aplicaciones
- DLL: clases (librerías)



29/09/2010
Pág. 64



Comparación .NET con Java (I)



29/09/2010
Pág. 65



Comparación .NET con Java (II)

- Estandarización
 - .NET está estandarizado. ECMA ISO/IEC estandariza CLR, CLI y CIL
 - J2EE no estandarizado. Sun tiene permiso para cambiarlo en cualquier momento.
 - Voluntariamente participa en Java Community Process para acordar los cambios con los usuarios.



Comparación .NET con Java (III)

- Licencia
 - Sun sólo permite que se use el nombre Java con su consentimiento.
 - Sólo lo permite para implementaciones compatibles 100%
- .NET es CLI, estándar.
 - Cualquiera puede implementarlo sin licencia alguna (Mono project, DotGNU...)



Comparación .NET con Java (III)

- Respecto a presencia
 - Java está muy presente
 - Gran cantidad de proyectos OpenSource
 - Promete implementaciones libres de Java
 - Está portado a gran cantidad de S.O.
 - .NET
 - Presente en Windows
 - Otros por el proyecto Mono
 - Gran penetración en el mercado móvil (pocket PC)
 - Gran penetración en Servidores



29/09/2010
Pág. 68



INTRODUCCIÓN A C#

¿Qué es C#?



29/09/2010
Pág. 69



Introducción a C#

- Comentarios

```
/* Esto es un comentario */  
int a =0;  
Int32 b= -128; // comentario int e Int32 son iguales  
// es decir, enteros de 32 bits  
UInt32 c= 1; // es entero sin signo
```

- Identificadores

```
/* Cualquier cadena de caracteres que no empiece por núm*/  
int a =0;  
Int32 variable = -128; // comentario int e Int32 son iguales  
// es decir, enteros de 32 bits  
UInt32 año = 1; // es entero sin signo
```



Introducción a C#

- Palabras reservadas

abstract, as, base, bool, break, byte, case, catch, char, checked, class, const, continue, decimal, default, delegate, do, double, else, enum, event, explicit, extern, false, finally, fixed, float, for, foreach, goto, if, implicit, in, int, interface, internal, lock, is, long, namespace, new, null, object, operator, out, override, params, private, protected, public, readonly, ref, return, sbyte, sealed, short, sizeof, stackalloc, static, string, struct, switch, this, throw, true, try, typeof, uint, ulong, unchecked, unsafe, ushort, using, virtual, void, while



Introducción a C#

- Literales

```
/* Enteros */  
int positivoDecimal = 0; // decimal  
int negativoDecimal = -128; // negativo  
int hexadecimal = 0x12;
```

```
/* Reales */  
Single real7digitos = 1,2; // Real 7 dígitos precisión (32bits)  
Double real16 = 1.8e4; // Real 16 dígitos precisión (64bits)  
Decimal real29 = 2.4E+3; // Real 29 dígitos precisión (128bits)
```

```
/* Lógicos */  
bool a = true; Boolean b = false;
```



Introducción a C#

- Literales

```
/* Caracter */  
Char caracter = 'a'; // entre comillas  
Char caracterUnicode = \u0041; // Unicode
```

Carácter Escape	Código Unicode	Código
Comilla simple	\u0027	\'
Comilla doble	\u0022	\"
Carácter nulo	\u0000	\0
Salto de página	\u000C	\f
Nueva línea	\u000A	\n
Retorno de carro	\u000D	\r
Tabulación horizontal	\u0009	\t
Tabulación vertical	\u000B	\v
Barra invertida	\u005C	\\



Introducción a C#

- Literales

```
/* Cadenas de texto */  
String cadena = "hola mundo"; // entre comillas  
String cadena2 = "hola\nQué tal?\nBien, y tú?";
```

```
/* Literal nulo*/  
String cadena = null;  
MiClase instanciaDeMiClase = null;
```



Introducción a C#

- Operadores

```
/* Aritméticos */  
int a = 3+2;  
int b = 7-15;  
b = a/5;  
a = 3*2;  
b = 5%2;  
a++;  
b--;
```

```
/* lógicos */  
if ((a && b) || (b && c))  
{  
    if(!a)  
        doSomething();  
}
```



Introducción a C#

- Operadores

```
/* Relacionales */  
if (a>=b)  
else if (b>c)  
else if (c<d)
```

```
/* Asignación */  
A = b;  
C = C + 3;  
C+=3;
```

```
/* concatenar cadenas */  
String hola = "hola";  
String mundo = "mundo";  
String holamundo = hola + " " + mundo;
```



Introducción a C#

- Operadores

```
/* Arrays */  
Int32 a[] = new Int32[12];  
a[1] = 0; a[2] = 23;
```

```
/* acceso a objetos */  
MiClase a = new MiClase();  
a.miAtributoPublico = 12;  
a.miMétodo();
```

```
/* Tipo y tamaño */  
System.Type miTipo = typeof(miVariable);  
if (miVariable is Int32)  
    mivariable++;  
Console.WriteLine("Tamaño de miVariable" + sizeof(miVariable));
```



Introducción a C#

- Operadores

```
/* Crear objetos */  
Figura fig = new Figura();  
Figura fig = new Figura("triangulo");
```

```
/* conversion o cast*/  
i = (int) 1;
```



Tipos de dato

Bytes con signo	8	-128 – 127
Bytes sin signo	8	0 – 255
Enteros cortos con signo	16	[-32.768, 32.767]
Enteros cortos sin signo	16	[0, 65.535]
Enteros normales	32	[-2.147.483.648, 2.147.483.647]
Enteros normales sin signo	32	[0, 4.294.967.295]
Enteros largos	64	[-9.223.372.036.854.775.808, 9.223.372.036.854.775.807]



Enteros largos sin signo	64	[0-18.446.744.073.709.551.615]
Reales con 7 dígitos de precisión	32	[1,5×10 ⁻⁴⁵ - 3,4×10 ³⁸]
Reales de 15-16 dígitos de precisión	64	[5,0×10 ⁻³²⁴ - 1,7×10 ³⁰⁸]
Reales de 28-29 dígitos de precisión	128	[1,0×10 ⁻²⁸ - 7,9×10 ²⁸]
Valores lógicos	32	true, false
Caracteres Unicode	16	['\u0000', '\uFFFF']
Cadenas de caracteres	Variable	El permitido por la memoria
Cualquier objeto	Variable	Cualquier objeto


29/09/2010
Pág. 80


Clases

```

class miClase
{
    public int ultimoNumero;
    private int numeroTotal;
    public miClase()
    {
        numeroActual = 1;
        numeroTotal = ultimoNumero;
    }
    public void incrementa(int cantidad)
    {
        numeroTotal += ultimoNumero;
        ultimoNumero = cantidad;
    }
    public int getnumeroTotal()
    {
        return numeroTotal + ultimoNumero;
    }
    static void Main(string[] args)
    {
        miClase mc = new miClase();
        mc.incrementa(12);
        int a = mc.getnumeroTotal();
    }
}

```

Algoritmia

Secuencias de control

```
if (b>=(2*4))
{
    a = 2;
}else
{
    a = 0;
}
```

```
switch(a)
{
    case 0:
        b=0;
        break;
    case 1:
    case 2:
        b=5
        break;
    default:
        trhow new
        Exception("error");
}
```



Algoritmia

```
int a = 20;
while(a<200)
{
    Console.WriteLine("Voy por el "+a);
    a++;
}
do
{
    Console.write(a);
    a++;
}while(a<210);

for(int i=1;i<200;i++)
{
    a++;
}

foreach(tipoElemento variable
in colección)
{
    dowhatever();
}
```



PRÁCTICAS

Programas sencillo en C#. Es recomendable hacer las prácticas demostrativas



29/09/2010
Pág. 84

