



Escola Politècnica Superior  
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL DE FI DE CARRERA

**TÍTOL:** Disseny de sistema operatiu Windows CE per un dispositiu empotrat

**AUTOR:** Adrián Couñago Silva

**DIRECTOR:** Marco Antonio Peña Basurto

**DATA:** 3 de Octubre de 2006

**Títol:** Disseny de sistema operatiu Windows CE per un dispositiu empotrat

**Autor:** Adrián Couñago Silva

**Director:** Marco Antonio Peña Basurto

**Data:** 3 de Octubre de 2006

## **Resum**

Este es un proyecto que tiene como objetivo desarrollar un sistema operativo Windows CE mediante la herramienta de desarrollo Platform Builder.

El proyecto pretende integrar con tres proyectos más del mismo departamento para implementar un dispositivo empotrado muy particular. Podemos, por tanto decir que el proyecto común está formado por los siguientes:

- Aplicación
- Protocolo de red
- Hardware
- Plataforma (Sistema Operativo)

El dispositivo empotrado tiene la funcionalidad principal de entrar en comunicación con un PC de nuestra red, ya sea cableada o wireless, y poder reproducir diversos formatos de audio y vídeo.

En pocas palabras, un reproductor multimedia conectado permanentemente en red.

El dispositivo incorpora una placa Mini-ITX VIA EPIA MII con un procesador EBGA C3/Eden. Sobre este sistema debe ejecutarse un sistema operativo basado en Windows CE que es el objetivo de este proyecto en particular.

Asimismo, como integración debe soportar la aplicación que será la encargada de interactuar con el usuario y de atender sus peticiones con un entorno gráfico y amigable. La aplicación se comunicará con el PC mediante un protocolo de red establecido. Por otro lado, el control físico del dispositivo se llevará a cabo por una botonera y la visualización en una pantalla LCD.

**Title:** Design of WCE operating system to a embedded device

**Author:** Adrián Couñago Silva

**Director:** Marco Antonio Peña Basurto

**Date:** October, 3rd 2006

## **Overview**

This is a Project that has got as target to develop a Windows CE operating system using the development tool called Platform Builder.

Furthermore, it has got an integration target with three projects too. These are in the same department and joined they will be an embedded device. For this, we can say that the global project is composed by the next ones:

- Application
- Network protocol
- Hardware
- Platform (Operating System)

The embedded device that we want to develop has got one basic function, it can establish a communication between a PC using our network, wired or wireless, and then it can play too many multimedia formats, audio and video.

In few words, a multimedia player with permanent connection to network.

The device incorporates a Mini-ITX VIA EPIA MII motherboard with a processor EPGA C3/Eden. Over this system will be executed a Windows CE operating system and it is the primary target in this project.

Also, like integration it must support the application that will be the one in interact with the user and to take care of his requests with graphical and friendly environment. The application will communicate with the PC using a network protocol established. On the other hand, the physical control of the device will be carried out in a button commandment, adding multimedia player function on LCD.

**Dedicatoria:**

Deseo dar mi más sincero agradecimiento a cada una de las personas que han hecho posible la consecución no sólo de este proyecto sino de toda una serie de objetivos en mi vida.

Mis padres, José y Ana, que me han apoyado en todo momento y me incitan a la superación, la honradez y el esfuerzo.

Mi hermano José Luís que me da toda la alegría que puede ofrecer un hermano pequeño y que aporta energía para continuar.

Mi novia Sara que aporta la comprensión y la paciencia que sólo una persona de su valía y constancia puede ofrecer.

Mis compañeros de piso, Kirian y Pol, que han sabido como apoyarme y como distensionarme durante tantas horas de trabajo

Y por último a todo el equipo de trabajo que me rodeó. El director de proyecto, Marco, que me invitó a esta buena experiencia y su colaborador, Juan, que me han enseñado a mejorar en muchos aspectos. A los compañeros de batalla, Gemma, Pau y Jordi que han demostrado una excelente predisposición y que han superado mis expectativas a todos los niveles.

Gracias.

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. ESTUDIO INICIAL Y ANÁLISIS</b> .....	<b>3</b>
<b>1.1. Escenario, visión y tecnología</b> .....	<b>3</b>
1.1.1. Escenario .....	3
1.1.2. Visión .....	6
1.1.3. Tecnología .....	8
<b>1.2. Motivación</b> .....	<b>14</b>
<b>1.3. Adecuación a los estudios</b> .....	<b>15</b>
<b>1.4. Objetivos individuales</b> .....	<b>16</b>
<b>1.5. Objetivos globales</b> .....	<b>17</b>
<b>1.6. Planificación y estimación de costes</b> .....	<b>17</b>
1.6.1. Recursos necesarios .....	17
1.6.2. Planificación prevista de tareas .....	18
1.6.3. Estimación de costes .....	21
<b>CAPÍTULO 2. WINDOWS CE Y PLATFORM BUILDER</b> .....	<b>23</b>
<b>2.1. Windows CE</b> .....	<b>23</b>
2.1.1. Orígenes y evolución de Windows CE .....	23
2.1.2. Objetivos de Windows CE .....	25
2.1.3. Características de Windows CE .....	27
2.1.4. Arquitectura de Windows CE .....	28
<b>2.2. Platform Builder</b> .....	<b>32</b>
2.2.1. Descripción de Platform Builder .....	32
2.2.2. Descripción del catálogo .....	33
2.2.3. Platform Builder IDE .....	34
2.2.4. Estructura de directorios de PB .....	37
2.2.5. Compiladores para microprocesadores .....	38
<b>2.3. Evaluación inicial de Platform Builder</b> .....	<b>39</b>
2.3.1. Experiencias preliminares .....	39
2.3.2. Primeros pasos con Platform Builder .....	40
<b>CAPÍTULO 3. DISEÑO DEL SISTEMA OPERATIVO</b> .....	<b>43</b>
<b>3.1. Como crear un sistema operativo WCE</b> .....	<b>43</b>
3.1.1. Contexto .....	43
3.1.2. Prestaciones y limitaciones del SO .....	44
3.1.3. Proceso de creación .....	46
<b>3.2. Diseño primera imagen: Tiny Kernel</b> .....	<b>47</b>
3.2.1. Descripción .....	47
3.2.2. Elementos que componen la imagen .....	48
<b>3.3. Diseño segunda imagen: Tiny Kernel con interfaz gráfica</b> .....	<b>49</b>
3.3.1. Descripción .....	49
3.3.2. Elementos que componen la imagen .....	49

3.4.	<b>Diseño imagen final: BSP Via Epia MII</b> .....	<b>52</b>
3.4.1.	Descripción.....	52
3.4.2.	Elementos que componen la imagen.....	53
 <b>CAPÍTULO 4. PRUEBAS</b> .....		<b>61</b>
4.1.	Pruebas primera imagen: Tiny Kernel.....	61
4.2.	Pruebas primera imagen: Tiny Kernel con interfaz gráfica.....	64
4.3.	Pruebas imagen finales: BSP Via EPIA MII.....	66
 <b>CAPÍTULO 5. INTEGRACIÓN</b> .....		<b>69</b>
5.1.	Adición de la aplicación de .NET sobre WCE.....	69
5.2.	Funcionamiento de la aplicación sobre WCE.....	71
 <b>CAPÍTULO 6. ELECCIÓN DE PLACA</b> .....		<b>73</b>
6.1.	<b>Elección de la placa</b> .....	<b>73</b>
6.1.1.	Estudio previo de mercado. Visión general.....	73
6.1.2.	Propuesta de comparativa presentada.....	74
6.1.3.	Elección final.....	79
6.1.4.	Consecución de expectativas.....	80
 <b>CAPÍTULO 7. BALANCE Y CONCLUSIONES</b> .....		<b>81</b>
7.1.	<b>Revisión de objetivos individuales y de grupo</b> .....	<b>81</b>
7.1.1.	Objetivos individuales.....	81
7.1.2.	Objetivos de grupo.....	81
7.2.	Líneas de trabajo futuras.....	82
7.3.	<b>Revisión de planificación y costes</b> .....	<b>82</b>
7.3.1.	Planificación final.....	82
7.3.2.	Costes finales.....	85
7.4.	<b>Ambientalización</b> .....	<b>86</b>
7.5.	<b>Conclusiones</b> .....	<b>87</b>
 <b>CAPÍTULO 8. APÉNDICE</b> .....		<b>89</b>
8.1.	<b>Diseño paso a paso de las plataformas</b> .....	<b>89</b>
8.1.1.	Imagen basada en Tiny Kernel.....	89
8.1.2.	Imagen basada en Tiny Kernel con interfaz gráfica.....	100
8.1.3.	Imagen basada en BSP VIA EPIA MII.....	115
8.2.	<b>Montaje del dispositivo</b> .....	<b>135</b>
8.3.	<b>Errores de la aplicación sobre WCE</b> .....	<b>137</b>
 <b>CAPÍTULO 9. BIBLIOGRAFÍA</b> .....		<b>139</b>
9.1.	<b>Bibliografía y fuentes de información</b> .....	<b>139</b>
9.1.1.	Bibliografía.....	139
9.1.2.	Otras fuentes de información.....	139

## INTRODUCCIÓN

Una visión general de nuestro alrededor nos puede desvelar que vivimos en la sociedad de la información, donde es preciso disponer de ella en cualquier lugar y desde cualquier dispositivo.

La portabilidad es una baza de gran interés ya también que vivimos en una sociedad de permanente cambio geográfico. Invertimos más tiempo en desplazamientos y tenemos la necesidad de seguir comunicándonos o accediendo a información.

Podemos constatar como los diferentes dispositivos a los que ya nos hemos acostumbrado interactúan entre ellos, tienen conexión a Internet y ofrecen, en algunos casos, portabilidad.

Desde luego, la evolución natural de este entramado tecnológico es conseguir dispositivos que integren el mayor número de funcionalidades posibles en el tamaño más pequeño posible y con portabilidad.

El objetivo global de este proyecto la implementación de un dispositivo empotrado con la capacidad de ser un reproductor de medios unido a nuestra red.

Para realizar este proyecto se dividirá en cuatro proyectos individuales. Los proyectos son: Plataforma, Aplicación, Red y Hardware.

Este proyecto en particular es el referente a la parte de Plataforma. El proyecto de Plataforma pretende diseñar un sistema operativo encargado de ejecutarse bajo el dispositivo y brindar los recursos necesarios a los otros proyectos.

En particular se aborda la herramienta Platform Builder encargada de la creación de sistemas operativos basados en la plataforma Windows CE, una versión reducida de la versión Windows de PC.

Aparte de la integración con otros proyectos se darán servicios adicionales al fin de enriquecer el sistema creado. Es decir, se añaden aplicaciones que normalmente utiliza el usuario final en un entorno Windows.





# CAPÍTULO 1. ESTUDIO INICIAL Y ANÁLISIS

## 1.1. Escenario, visión y tecnología

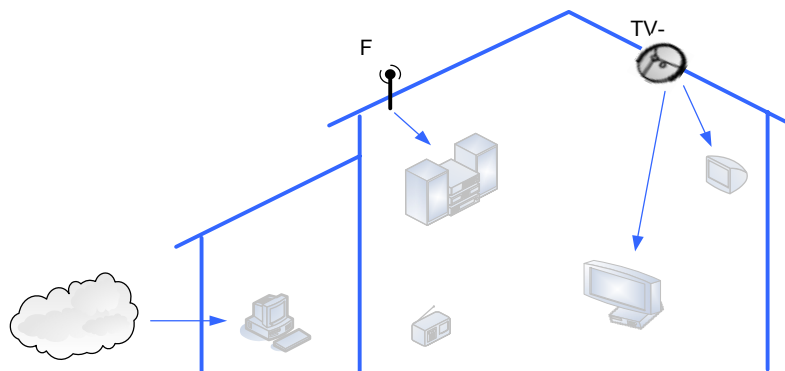
### 1.1.1. Escenario

El escenario en el que nos encontramos define un marco de entretenimiento en el hogar brindando diferentes servicios como la televisión, la radio, la televisión por satélite, Internet, etc. Estos canales de información llegan a nuestro hogar por medios distintos y se accede a ellos mediante dispositivos especialmente diseñados para el medio en cuestión.

La continua proliferación de Internet, además de que cada vez bajan más los precios y aumentan la velocidad de conexión, acerca nuevos servicios a los usuarios además de los servicios clásicos. Es decir, seguimos gozando de servicios como la televisión y la radio pero se unen nuevas tecnologías como la televisión y la radio digital combinando nuevos contenidos y ofreciendo un amplio abanico de archivos multimedia: imágenes, películas, música. Además ofrecen la versatilidad de soportar muchos formatos diferentes. Por este motivo, podemos decir que nuestro ordenador personal es el dispositivo centralizado que integra la posibilidad de trabajar con todas estas vías de información.

Todavía existe aún una limitación muy clara, y es que hasta el momento para acceder a la información de diferentes contenidos en un emplazamiento diferente a nuestro ordenador personal debemos grabar estos contenidos en un medio físico, ya sea un CD, DVD, USB, etc. Después debemos transportar dicho medio físico hasta otro dispositivo capaz de reproducir tales contenidos y que se encuentre en el emplazamiento deseado.

El objetivo del usuario es simplemente acceder a cualquier tipo de contenido en cualquier lugar del hogar de la forma más rápida y práctica posible.

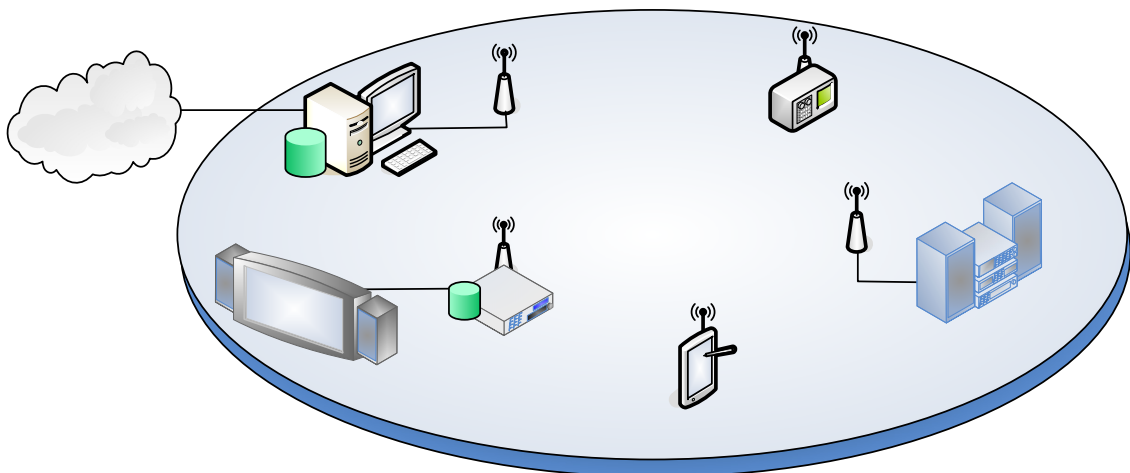


**Fig. 1.1** Posible entorno actual de entrada de medios

Evidentemente, para acceder a estos contenidos de forma fácil y desde cualquier lugar del hogar es preciso que exista una integración a nivel de conexión entre los dispositivos bien sea utilizando una red cableada (Ethernet) o con una red inalámbrica (Wifi). Asimismo deberá existir un protocolo de comunicación común para que los diferentes dispositivos puedan descubrirse en la red, identificar medios y formatos, indexar los medios, navegación, etc. De esta forma disfrutamos de imágenes, música o vídeos almacenados en el ordenador personal desde la televisión del salón o reproducir música directamente en la cadena musical situada en otro punto de la casa.

El escenario que se nos presenta será el comentado hasta ahora, un ordenador personal que se utiliza como centro de medios digitales y que actúa como servidor de contenidos multimedia. Las tareas básicas que debe cubrir son:

1. Almacenar los medios
2. Indexar los medios
3. Atender demandas
4. Distribución por la red de medios

**Fig. 2.1** Conexión de cada entrada de medios a la red

Los dispositivos conectados a la red serán los encargados de comunicarse con el servidor de contenidos y también de reproducir los medios recibidos. El ideal es que cualquier dispositivo pueda entrar en comunicación con el servidor de contenidos multimedia por muy simple que este fuera.

Debemos ser conscientes de que todos los dispositivos consumidores de contenidos multimedia tienen unas características comunes y otras particulares, lo cual requeriría un estudio específico.

Por esta razón, y para poder abordar este escenario, lo que nos planteamos es el diseño de un dispositivo que integre todos los elementos comunes y sea capaz de soportar los elementos diferenciadores. De esta forma, lo que se pretende es crear un dispositivo suficientemente modular para poder, a partir de este, construir cualquier otro dispositivo consumidor de medios.

El dispositivo que pretendemos diseñar está basado en una placa mini – ITX (17x17cm), concretamente el modelo llamado VIA EPIA-MII compatible con el sistema operativo Windows CE.

Debíamos determinar qué placa escoger y nos basamos en unas necesidades impuestas como son las siguientes:

- Salida VGA
- Salida de audio
- Salida de TV
- Puerto serie
- Posibilidad de tarjeta de red o llevarla integrada (preferible)
- Opcional: conexión wireless
- Conectividad: DVD, HDD, CF, PCMCIA...

Después de hacer un estudio de mercado de las diferentes opciones que se ofrecían, dispusimos adquirir el modelo antes mencionado porque considerábamos que era el que reunía mejor nuestras necesidades. Además de esto, el modelo VIA EPIA-MII tenía un coste muy atractivo por las características que dispone y en comparación con otras placas similares.

El dispositivo requiere de una aplicación que sirva al usuario de interacción con el dispositivo y que sea compatible con el sistema operativo Windows CE. El sistema debe proveer de servicios de reproducción (codecs, Windows Media Player...), navegador de Internet, etc. Y también incluirá un microcontrolador PIC para integrar una pantalla de LCD y una botonera externa que se comunicará con la placa mini-ITX mediante el puerto serie.

El protocolo de red pretende adaptar la tecnología de comunicación (Ethernet y Wifi) que dispongamos para enlazar el dispositivo y el generador de medios de consumo. El dispositivo a su vez deberá ser capaz de descubrir estos servidores de medios, anunciarse, indexar medios disponibles y acceder a ellos.

Debido a los numerosos frentes que se abren al abordar este proyecto, se optó por emprenderlo de manera eficiente dividiéndolo en cuatro partes:

- **Protocolo de red:** Implemente la capa de software que se precisa para almacenar, organizar y distribuir por la red los contenidos disponibles en los servidores de medios. La implementación del código del protocolo de red deberá poder ser ejecutada tanto desde un sistema Windows como un sistema UNIX/Linux.
- **Aplicación:** es el propio software de interacción con el usuario. Como en cualquier dispositivo de índole parecida dispondrá de una aplicación con un menú principal en el que se muestren las posibilidades del dispositivo. Podremos navegar por los contenidos multimedia de forma interactiva y amigable haciendo uso de un entorno gráfico.
- **Plataforma:** sistema operativo basado en Windows CE. Creado mediante la herramienta de diseño Platform Builder, añadirá los módulos y servicios necesarios para permitir el correcto funcionamiento tanto de la aplicación, como del protocolo de red y el soporte serie para la parte hardware. Incluirá el Board Support Package proporcionado por el fabricante VIA.
- **Hardware:** Elemento adicional al dispositivo en el que figura un microcontrolador, un LCD y una botonera encargada del control de la aplicación del dispositivo. Podremos navegar y seleccionar contenidos multimedia visualizando toda esta información desde la pantalla LCD.

### **1.1.2. Visión**

Podemos cerciorarnos de la cantidad de archivos multimedia que poseemos almacenadas en nuestro ordenador personal, además, con el uso de Internet tenemos al alcance muchos más medios que consumir. Pero igualmente que el servicio de información está centralizado en nuestro ordenador personal también el lugar de reproducción se encuentra centralizado en este mismo ordenador. Esto incomoda, hace ardua la tarea de acceder a cualquiera de nuestros medios almacenados y visualizarlos o escucharlos, si es música, en cualquier dependencia de nuestro hogar mediante nuestra red LAN.

En la nueva generación el escenario dispone que, los dispositivos del hogar, puedan acceder de manera simultánea a todos los contenidos multimedia disponibles en una red para disfrutarlos en cualquier lugar. Se pretende como objetivo la interacción de cada dispositivo con el entorno, es decir, con el usuario y con el resto de dispositivos.

En el hogar, podríamos clasificar los dispositivos como:

### **1. Dispositivos móviles**

Forman parte dispositivos como el teléfono móvil y similares (SPhone...) Agendas de bolsillo, PDA's, Pocket's y similares.

### **2. Dispositivos de escritorio con conexión a Internet**

Figura el ordenador personal de escritorio y sus periféricos que normalmente poseen conexión permanente a Internet.

### **3. Dispositivos de radiodifusión**

Aquí se agrupan elementos como la televisión o la cadena musical. Hoy en día se incluyen los decodificadores/receptores de satélite o cable. También se añaden los demás dispositivos electrónicos (radio,

En este sentido muchas compañías se encuentran involucradas en proporcionar una interacción entre las tres clases de dispositivos (móviles, de Internet y de radiodifusión). El objetivo que se pretende es llegar a lo que se conoce como "Digital Home and Digital Living".

Digital Home es el colofón a la idea de integrar estas tres clases de dispositivos y que trabajen juntos mediante la conexión por medio cableado o inalámbrico de forma que podemos compartir los contenidos multimedia.

Podemos decir que la interoperabilidad de dos dispositivos se da cuando ambos pueden dar operabilidad a un servicio de forma transparente al usuario. Por supuesto, esto incluye la propiedad de poder comunicarse entre ellos.

Sin embargo, debemos saber que la interoperabilidad de los dispositivos tal y como se entiende con "Digital Home" requiere el estudio de una serie de aspectos, como son:

- Colaboración industrial:

Los grandes fabricantes del sector de las comunicaciones digitales tienden a ofrecer un amplio abanico de productos nuevos pero siempre siguen unas mismas directrices.

Estos mismos fabricantes deben unir sus esfuerzos con el objetivo de desarrollar y materializar la idea de "Digital Home".

- Estructura de interoperabilidad:

En el momento de crear productos de nuevas categorías son los fabricantes los que deben consensuar la estructura de interoperabilidad que utilizarán estos productos. Definir unos estándares y/o pautas es lo más recomendable.

Estos estándares después pueden ir variando para asegurar una plena interoperabilidad entre dispositivos nuevos y antiguos formando lo que se llama “Digital Living Network”.

- **Productos competitivos:**

Se requieren de dispositivos versátiles e interoperantes, por ello pueden englobar una o más de una función como: DMS (servidor de recursos digitales) o DMP (reproductor de recursos digitales).

Para que la interoperabilidad salga adelante deben unir fuerzas tanto productores como distribuidores y tener en cuenta que los consumidores pagamos por un dispositivo digital que satisfaga realmente nuestras necesidades y no sólo algunas de ellas.

En corto o medio plazo, las casas serán digitales porque tendrán una o más plataformas digitales como es el caso de un ordenador personal o de “set-top-boxes” (STB). Estas plataformas serán las encargadas de la gestión, administración y distribución de los medios digitales hasta dispositivos como la televisión, la cadena musical, la cámara digital, etc.

### **1.1.3. Tecnología**

Hoy en día casi todo el mundo tiene un ordenador personal en su hogar. Este medio no sólo es cada vez más usual y asequible sino que aporta un abanico de posibilidades en cuanto a gestión de diferentes formatos de información. En cualquier ordenador familiar podemos encontrar desde archivos de imágenes, pasando por archivos de sonido hasta documentos del colegio o el trabajo.

Además, la proliferación masiva de nuevos accesos a Internet provoca que un ordenador conectado a la red sea fuente y destino de información. Aunque muchas veces no encontramos tiempo, espacio y ganas para acceder a esta información y simplemente se ignora como si se diese por perdido. Por ejemplo, cuando nos encontramos aseándonos y nos gustaría seguir escuchando nuestra cadena de radio favorita que podríamos estar escuchando de estar en el salón, encender nuestro equipo musical y sintonizar la emisora. Ahora, imaginemos que nos encontramos en el lavabo pero al entrar hacemos tres leves movimientos sobre una pantalla táctil, uno para encender el dispositivo, otro para elegir el modo radio y otro para elegir nuestra emisora entre nuestras favoritas. Automáticamente empezaría a reproducir esta sintonía mientras nosotros, por ejemplo, nos duchamos.

Pero, ¿podríamos rizar más el rizo? La respuesta es sí. Pensemos en elegir reproducir en este dispositivo archivos que se encuentran en nuestro PC familiar mediante su conexión a la red. De esta forma convertiremos nuestra casa en una extensión más de nuestro PC llevando aquí a cabo actividades como la navegación por la red, la reproducción de archivos multimedia o incluso la gestión y control de otros dispositivos conectados en el PC, como es el caso de una impresora.

De esta forma, podríamos conseguir un hogar donde una serie de nodos permiten interactuar de forma completa y muy activa sobre dispositivos de la casa. Ejemplo de estas ideas se derivan nuevas neveras conectadas a Internet las cuales nos permiten efectuar la compra derivando esta funcionalidad sobre el dispositivo.

Estas son algunas de las razones e ideas que hacen cobrar sentido a la existencia y desarrollo de dispositivos de estas características, y de una forma u otra, se irán introduciendo en nuestros hogares.

A continuación haremos un pequeño repaso por algunos dispositivos que encarnan el espíritu de nuestro dispositivo y alguna de sus características.

Ejemplos de dispositivos son:

- Ordenadores personales basados en placas ITX: El ejemplo más cercano a nosotros serían los nuevos ordenadores personales que comienzan a montarse. Podemos destacar los PC's que utilizan una placa ITX, como es nuestro caso, ya sea en su versión mini (17x17cm) o en su versión nano (12x12cm).

Estas placas tienen prácticamente todas las características que podemos demandar de un PC de escritorio común. Poseen interfaces de entrada y salida tales como audio, video o, incluso algunas más avanzadas, como puerto 1394 (firewire), ranuras PCMCIA y Compact Flash.

Todas estas características, y sus potentes microprocesadores que llegan a soportar velocidades hasta 3 GHz, hacen una apuesta segura como dispositivo receptor de medios teniendo en cuenta su reducido tamaño.

Por último destacar que los precios de estas placas oscilan entre los 150 y los 400 euros aproximadamente. Aunque en algún caso superan los 500 euros de coste.

- Otros ordenadores personales: Sin embargo, aparte de los ordenadores antes comentados, Apple ya comercializa equipos completos montados con las mismas prestaciones pero que ejecutan un Sistema Operativo Macintosh (MAC). Estos dispositivos se comercializan con el pseudónimo "MAC Mini". Ofrecen características muy similares también a nuestros ordenadores personales en un espacio menor y con un diseño moderno.

Apple monta un sistema basado en un procesador PowerPC G4 (1.42 Ghz) o bien, su modelo superior, utiliza un Intel Core Duo (1.83 GHz). Ambos sistemas han sido desarrollados para obtener un máximo rendimiento en la gestión y reproducción de archivos multimedia (vídeo, fotos....).

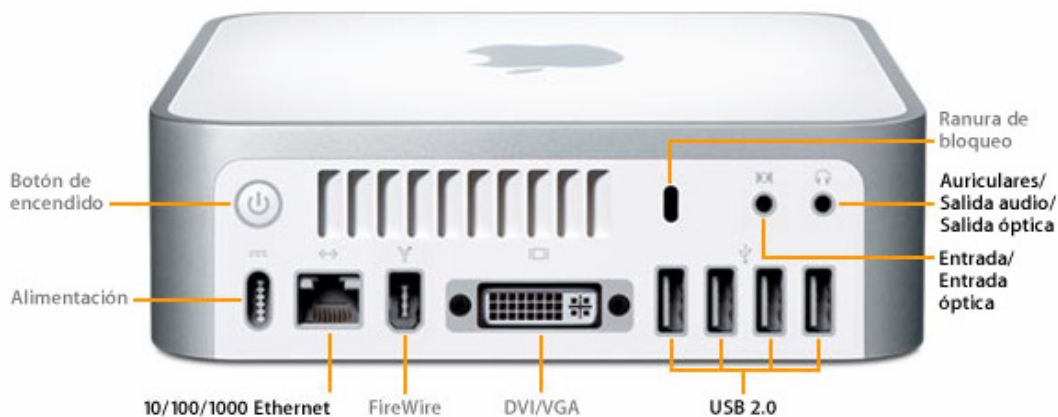


Las dimensiones del Mac Mini son realmente interesantes y podríamos afirmar que se convierte, de alguna forma, en un dispositivo portátil a efectos que sólo pesa 1.3 Kg. Aproximadamente.

**Fig. 3.1** Figura del Mini-MAC de Apple

Entre las interfaces disponibles se hallan las siguientes:

- Puerto Ethernet 10/100/1000
- Puerto Firewire IEEE 1394
- Salida DVI/VGA
- 4 puertos USB
- Entrada de audio (analógica y óptica)
- Salida de audio (analógica y óptica)
- Otros: bluetooth, 802.11g, módem...



**Fig. 4.1** Parte posterior del Mini-MAC de Apple

Otras características de serie son su unidad de almacenamiento masivo de 80 GBytes, opcionalmente 120 GBytes, y una memoria RAM que puede alcanzar hasta los 2 GBytes.

El sistema operativo que ejecuta es el Mac OS X versión 10.4 Tiger, desarrollado especialmente para exprimir al máximo sus características.

Este ordenador personal está disponible desde unos 640 euros.



- Nuevos dispositivos con conexión a red: Son aparatos a los que estábamos acostumbrados a tratar en nuestra vida cotidiana y ahora se le añade conexión a la red.

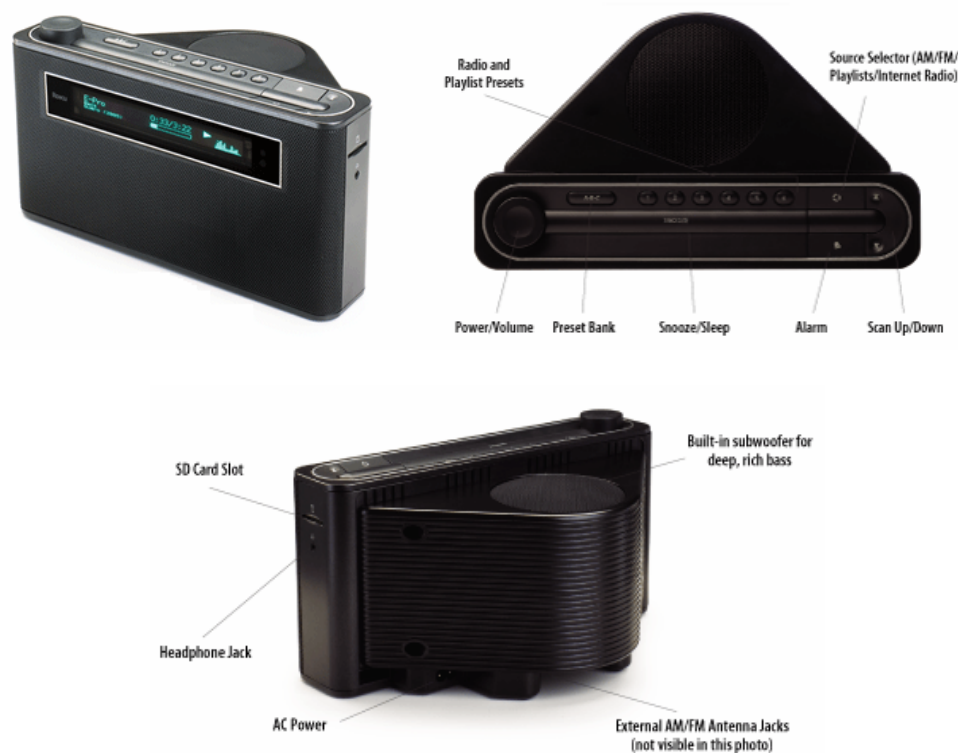
En este sentido, un ejemplo es la empresa Ruku que ha desarrollado diferentes dispositivos con esta característica básica, su conexión a la red permitiendo el streaming desde Internet.

Nos dedicaremos a tres diferentes productos de esta empresa: SoundBridge Radio+, SoundBridge Radio y PhotoBridge HD. Tengamos en cuenta que existen numerosas empresas que explotan estas mismas ideas incorporando otros productos.

### 1. SoundBridge Radio+

Este producto nace bajo la necesidad de disponer una radio que pueda sintonizar cualquier emisora tanto de radio convencional (AM/FM), radio sobre Internet y nuestros propios ficheros de audio desde cualquier lugar de la casa.

La radio estudiada se corresponde al modelo SoundBridge de Ruku. Se une como un elemento más de nuestra red doméstica mediante el uso de una conexión inalámbrica, lo que permite el acceso a Internet o a nuestros ficheros de audio compartidos en la red. A continuación podemos ver el dispositivo en cuestión:



**Fig. 5.1** Diferentes imágenes del modelo SouthBridge+ de Ruku

Tiene un diseño moderno y se ajusta a las posibilidades que ofrece. Desde selección de medio pasando por ranura SD/MMC o la salida de auriculares. Además incorpora un conector para antena externa de radio AM/FM.

Esta radio ofrece muy buenas características de sonido gracias a su sonido estéreo y la incorporación de subwoofer para enriquecer los sonidos graves de la reproducción.

Otras características a comentar son:

Conexión a estaciones de radio sobre Internet sin necesidad de PC sólo con la conexión Wifi. Reproducción de numerosos formatos de audio (MP3, WAV, WMA, AAC, AIFF) desde la red o bien desde la ranura SD/MMC. Alarma con selector de salida de cualquiera de sus fuentes. Reloj atómico actualizado directamente desde la hora oficial del observatorio naval por lo que no debemos preocuparnos de configurar la hora. Sensor luminoso que activa/desactiva la iluminación de pantalla. Acceso directo a favoritos y listas de reproducción. Mando a distancia. Posibilidad de actualización. El precio de este dispositivo es de 400 dólares.

## 2. SoundBridge Radio

Este modelo se define como el hermano pequeño del modelo anterior. Solamente actúa de servidor musical de nuestra red doméstica pero no posee la capacidad de reproducirla directamente en el dispositivo, sino que este debe conectarse a un equipo musical o bien a unos altavoces activos (con alimentación).

El diseño es mucho más compacto debido a que no incluye esta funcionalidad y esto repercute directamente en el precio final, mucho menor que el modelo superior, el cual oscila entre los 150 dólares y los 200 dependiendo la versión de modelo elegida. En la siguiente figura se observan las dos versiones de este modelo:



**Fig. 6.1** Diferentes imágenes del modelo SouthBridge de Roku

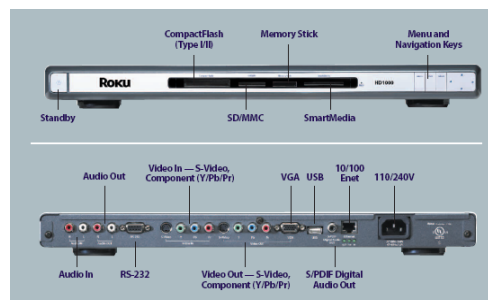
Las características principales de este modelo son muy parecidas al modelo anterior: Permite, aparte de conexión Wifi, conexión cableada a una red Ethernet gracias a su puerto RJ-45. Posee salida digital óptica de audio y también coaxial. Permite la reproducción de cualquiera de las fuentes mencionadas anteriormente en el otro modelo y no vale la pena repetirlas.

También existen algunas diferencias en las formas de acceder a la reproducción ya que la botonera cambia respecto el modelo superior, incorpora una función de búsqueda rápida de medios multimedia para facilitar la localización de nuestros medios.

### 3. PhotoBridge HD

La necesidad que pretende cubrir este dispositivo es la reproducción ya no sólo musical sino también de fotos y vídeo.

Gracias a este dispositivo podemos visualizar nuestras fotos y videos favoritos en la pantalla a la que se encuentre conectado, aunque el soporte final ideal sería una TV de alta definición. En la siguiente imagen vemos el diseño, su frontal y las interfaces disponibles en la parte trasera.



**Fig. 7.1** Imagen delantera, trasera y perspectiva de Photobridge HD

Posee conexión de red Ethernet con la posibilidad de acceder inalámbricamente mediante un adaptador USB inalámbrico puesto que viene provisto de dicho puerto.

Existen dos versiones diferentes, la diferencia principal entre ambos es que el modelo superior dispone del modo “galería de arte” lo que permite gozar de obras de arte, paisajes y demás en la pantalla.

Entre algunas de sus características más destacables se encuentran:

Reproducción de ficheros de audio (MP3,AIFF,ACC,WAV).  
 Reproducción de vídeo MPEG-2 y DVD's convencionales. Permite la reproducción de ficheros compartidos en redes Windows o MAC.  
 Permite funcionalidades sobre las fotos como opciones de zoom, rotación y otros efectos.

Tiene la ventaja de que su firmware está abierto y escrito sobre Linux, lo que permite desarrollar aplicaciones para este dispositivo. Este incluye un puerto RS-232 para estas tareas y otras de control como, por ejemplo, actualizar el firmware.

Tiene ranuras de expansión con soporte para los siguientes formatos: SD/MMC, CF, MemoryStick, MemoryStick PRO y SmartMedia.

Posee entrada y salida de RCA por lo que podemos conectar a él un dispositivo auxiliar o bien enviar la información, por ejemplo a una capturadora de TV de un PC con el objetivo de hacer grabaciones.

Este dispositivo ofrece una muy buena versatilidad a nuestro televisor pero resta en cierta medida la portabilidad de este producto. Su precio aún no está disponible en la Web oficial.

Podemos comprobar que en todos los productos vistos hasta ahora prima la posibilidad de reproducir muchos formatos de diferentes fuentes de información, mediante la conexión de red existente en el hogar.

Cada vez aumentan más posibilidades de integración y, aunque no lo hayamos tratado, comienzan a comercializarse productos para integrar órdenes domóticas en productos de ocio de esta índole.

## **1.2. Motivación**

Como ya comentamos anteriormente, los dispositivos empotrados cada vez son más comunes y cada vez interactúan más con otros dispositivos. En la actualidad estos dispositivos están en auge y se nos plantea un gran reto a la hora de implementar un dispositivo de esta clase.

A los integrantes de este grupo de trabajo nos mueve la posibilidad de a partir de unas directrices marcadas por el director de proyecto, desarrollar sendos proyectos y cuya integración conforman el dispositivo definido.

Asentar conocimientos adquiridos durante la carrera es uno de los motivos por el cuál decidí realizar este proyecto. Aparte de esto, hacer hincapié en la planificación de un proyecto serio y llevarlo a cabo dentro de unos márgenes establecidos supone un buen entrenamiento para el mundo laboral.

Este proyecto en particular pretende diseñar un sistema operativo basado en Windows CE que se ejecutará en el dispositivo. Windows CE es un sistema operativo propio de dispositivos empotrados y tiene limitaciones respecto al sistema Windows tradicional. Cada vez existen más dispositivos que ejecutan un sistema Windows CE por las necesidades que ya citamos en el apartado de Visión de este mismo capítulo.

La evolución de Windows CE hasta su versión actual (WCE 5.0) corrobora que es un sistema en crecimiento y expansión. Su versatilidad sobre diferentes

arquitecturas de computación (x86, ARM, CEPC...) aumentan cada vez más sus posibilidades de éxito entre los fabricantes de dispositivos empotrados para elegirlo como sistema para los mismos.

La herramienta de desarrollo utilizada para crear un sistema operativo basado en Windows CE es Platform Builder. Esta herramienta es muy desconocida y está muy poco documentada por lo que supone un esfuerzo añadido. Este esfuerzo se ve recompensado en el provecho sacado por parte de la consecución de este proyecto por lo que resulta a su vez una motivación añadida.

Debido a que nuestro proyecto general quiere implementar un dispositivo muy particular esto acrecentará el interés de crear un sistema operativo competente y llevar a cabo el proceso de aprendizaje de una herramienta tan puntera como Platform Builder.

### **1.3. Adecuación a los estudios**

Todo proyecto universitario pretende la consecución de adquirir los conocimientos prácticos necesarios que conlleva el desarrollo de un proyecto de cierta entidad.

Durante el transcurso de los estudios universitarios se adquieren una serie de conocimientos y de hábitos de trabajo que abastecen al estudiante de las capacidades necesarias para llevar a cabo un proyecto de esta importancia.

El Proyecto de Final de Carrera pretende potenciar las capacidades menos adquiridas o desarrolladas y consolidar aquellas que el estudiante ha adoptado naturalmente en el proceso de aprendizaje, seguimiento y responsabilidad que exige cualquier estudio universitario.

En particular, los estudios de Ingeniería de Telecomunicaciones ofrecen un amplio abanico de conocimientos en diversas materias. Debido a que el mercado de las Telecomunicaciones ofrece una salida profesional amplia, los estudiantes debemos ir especializándonos.

En mi caso personal, siempre me atrajo la programación y aunque en mis estudios figuran dos asignaturas de programación obligatorias decidí realizar una asignatura optativa de la plataforma .NET.

En esta línea he querido potenciar mis conocimientos en programación y relacionados desarrollando un sistema operativo para un dispositivo empotrado. Gracias a los conocimientos adquiridos sobre la plataforma .NET y el desarrollo de aplicaciones para dispositivos empotrados, puedo afirmar que estos se verán potenciados abriendo un nuevo frente en el desarrollo de sistemas operativos para dichos dispositivos.

## 1.4. Objetivos individuales

El objetivo principal de este proyecto es el diseño de un sistema operativo basado en Windows CE y ejecutarlo bajo un dispositivo con una placa mini – ITX del fabricante VIA Technologies, Inc. Por supuesto, como dije en el apartado anterior, prevalece el objetivo de consolidar los conocimientos adquiridos en la carrera.

Además del objetivo principal, existen objetivos secundarios propuestos para este proyecto. Estos objetivos son:

- Análisis y comprensión de la arquitectura de Windows CE

Este proyecto pretende introducir conocimientos sobre Windows CE. Si se pretende realizar un sistema operativo basado en Windows CE debemos conocer dicha plataforma. Tenemos como objetivo conocer el origen, la evolución, las prestaciones y los límites de este sistema.

- Análisis de la herramienta de desarrollo Platform Builder

Otro de los objetivos secundarios que pretende este proyecto es introducirnos en la herramienta de desarrollo de Microsoft para el desarrollo del sistema operativo basado en Windows CE.

Pretendemos diseñar un sistema mediante PB por lo que resulta imprescindible conocerlo básicamente. Platform Builder es una herramienta de desarrollo compleja, poco explotada y documentada. Por este motivo adquirir conocimientos acerca de esta herramienta es muy ventajoso puesto que no existen muchos desarrolladores al respecto.

- Evaluación de la dificultad del desarrollo del sistema operativo

Por último veremos el balance y las conclusiones que podemos extraer del proceso de desarrollo de la plataforma. Como afectó el proceso de desarrollo a la planificación y a los costes previstos en la etapa previa al diseño.

El cumplimiento de estos objetivos nos habrá hecho adquirir unos conocimientos sobre la implementación de plataformas para dispositivos empotrados mediante la herramienta Platform Builder.

La no consecución del objetivo obligará a la indagación sobre qué factores y circunstancias alteraron su consecución y qué soluciones podemos aportar para evitarlas.

## 1.5. Objetivos globales

Los objetivos globales relacionan este proyecto con el proyecto global, entre ellos figura la completa integración de la aplicación y el soporte hardware necesario. Es decir, el objetivo en este caso sería, aparte de crear una plataforma que funcione sobre la placa elegida, que sea capaz de brindar los servicios necesarios para que la aplicación instalada también funcione correctamente.

Si recordamos el resumen de este proyecto existen cuatro proyectos independientes: Plataforma, Aplicación, Protocolo de Red y Hardware. Cada uno de estos proyectos requiere algún servicio en particular para su correcta implementación. En la etapa de diseño veremos cuáles son los servicios que debe brindar el sistema operativo para cumplir los objetivos secundarios.

Los objetivos globales serán conseguidos si la plataforma resultante del trabajo técnico de este proyecto ofrece los servicios necesarios para cada uno de los proyectos antes mencionados.

Por ejemplo, en nuestro caso, instalaremos sobre el sistema operativo una aplicación que hace uso del paquete de Compact Framework 2.0, por lo que debemos añadir este módulo al diseño de nuestra plataforma.

## 1.6. Planificación y estimación de costes

### 1.6.1. Recursos necesarios

Recursos humanos necesarios:

- Analista
- Programador
- Documentador

Recursos materiales necesarios:

- Hardware: PC, disco duro, placa madre VIA EPIA MII, memoria RAM y fuente de alimentación
- Software: Platform Builder 5.0 , Board Support Package VIA EPIA MII
- Infraestructura

Se considera que cada uno de los recursos humanos es una persona física diferente con el objetivo de rarificar como si realmente se tratase de un proyecto del mundo laboral.

En el presupuesto no se contempla el coste en infraestructura porque es el propio Departamento de Arquitectura de Computadores quien brinda un espacio físico para el desarrollo del proyecto global. Tampoco tendremos en cuenta ni el PC ni el disco duro necesario para almacenar el sistema operativo.

### 1.6.2. Planificación prevista de tareas

Para poder planificar el proyecto debemos dividirlo en las tareas que se deben desarrollar para cumplir los objetivos propuestos por el proyecto. De esta forma nos resultará más cómodo asignar los recursos necesarios a cada tarea y estimar su duración y coste.

La siguiente ilustración recoge la planificación de las tareas previstas a desempeñar y su duración estimada. Debemos tener en cuenta que se incluye el proceso de aprendizaje de Windows CE y de herramienta de desarrollo Platform Builder. La duración estimada total en horas está calculada a partir de la duración estimada total en días considerando que la jornada diaria es de 4 horas.

**Tabla 1.1.** Planificación de tareas y duración estimada

<b>TAREA</b>	<b>RECURSO</b>	<b>DURACIÓN ESTIMADA (días)</b>
Situación y descripción de objetivos	Analista	5
Análisis del mercado y estudio de productos	Analista	7
Búsqueda de placas y materiales	Analista	7
Elección de placa	Analista	7
Pedido placa	Analista	30
Documentación Situación	Documentador	5
Documentación de análisis	Documentador	7
Documentación de búsqueda de placas	Documentador	5
Documentación elección de placa	Documentador	7
Documentación de Windows CE y Platform Builder	Documentador	9
Documentación de creación de imagen	Documentador	7
Documentación de creación de 2 imagen	Documentador	7
Documentación de imagen final	Documentador	7
Documentación de integración	Documentador	2
Elaboración de memoria	Documentador	14
Búsqueda software Platform Builder e instalación	Programador	7
Formación acerca de Windows CE	Programador	4
Formación acerca de Platform Builder	Programador	7
Creación 1era plataforma	Programador	7
Creación de 2a plataforma	Programador	7
Testeo de 1era imagen	Programador	5
Creación imagen final	Programador	9
Testeo de 2a imagen	Programador	5
Testeo imagen final	Programador	7
Integración	Programador	21
<b>RECURSO</b>	<b>DURACIÓN TOTAL (días)</b>	<b>DURACIÓN TOTAL (horas)</b>
Analista	26*	105
Documentador	70	280
Programador	79	316

\*A la suma de días de trabajo le restamos 30 días del pedido de la placa y añadimos una hora en concepto de esta labor.



Para estimar el tiempo estimado que requerido este proyecto tarea, se descompuso en tareas que permitan ver con más claridad qué trabajo requiere cada una de ellas y qué aporta al proyecto.

Una vez realizada esta división, analizamos la planificación completa del proyecto con la ayuda de una herramienta de gestión de proyectos. En nuestro caso trabajamos con la aplicación de Microsoft MS Project 2003 que nos permite seguir la evolución de las tareas así como anticipar y/o reaccionar ante los imprevistos que puedan surgir hasta la finalización del proyecto.

En la siguiente figura se nos muestran todos los detalles que proporciona dicha herramienta una vez introducida la planificación de proyecto prevista.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Situación y descripción de objetivos	5 days	Thu 23/02/06	Wed 01/03/06		Analista
2	Análisis del mercado y estudio de productos	7 days	Thu 02/03/06	Fri 10/03/06	1	Analista
3	Documentación Situación	5 days	Thu 02/03/06	Wed 08/03/06	1	Documentador
4	Búsqueda de placas y materiales	7 days	Mon 13/03/06	Tue 21/03/06	2	Analista
5	Documentacion de analisis	7 days	Mon 13/03/06	Tue 21/03/06	2	Documentador
6	Elección de placa	7 days	Wed 22/03/06	Thu 30/03/06	4	Analista
7	Documentacion de busqueda de placas	5 days	Wed 22/03/06	Tue 28/03/06	4	Documentador
8	Pedido placa	30 days	Fri 31/03/06	Thu 11/05/06	6	Analista
9	Busqueda software Platform Builder e instalcion	7 days	Thu 02/03/06	Fri 10/03/06	1	Programador
10	Documentacion eleccion de placa	7 days	Fri 31/03/06	Mon 10/04/06	6;7	Documentador
11	Formación acerca de Windows CE	4 days	Mon 13/03/06	Thu 16/03/06	9	Programador
12	Formación acerca de Platform Builder	7 days	Fri 17/03/06	Mon 27/03/06	11	Programador
13	Documentación de Windows CE y Platform Builder	9 days	Tue 28/03/06	Fri 07/04/06	11;12	Documentador
14	Creación 1era plataforma	7 days	Tue 28/03/06	Wed 05/04/06	12	Programador
15	Creación de 2a plataforma	7 days	Thu 13/04/06	Fri 21/04/06	16	Programador
16	Testeo de 1era imagen	5 days	Thu 06/04/06	Wed 12/04/06	14	Programador
17	Documentacion de creacion de imagen	7 days	Thu 13/04/06	Fri 21/04/06	16	Documentador
18	Creacion imagen final	9 days	Mon 01/05/06	Thu 11/05/06	19	Programador
19	Testeo de 2a imagen	5 days	Mon 24/04/06	Fri 28/04/06	15	Programador
20	Documentacion de creacion de 2 imagen	7 days	Mon 01/05/06	Tue 09/05/06	19	Documentador
21	Testeo imagen final	7 days	Fri 12/05/06	Mon 22/05/06	18;8	Programador
22	Documentacion de imagen final	7 days	Tue 23/05/06	Wed 31/05/06	21	Documentador
23	Integración	21 days	Tue 23/05/06	Tue 20/06/06	21	Programador
24	Documentación de integración	2 days	Wed 21/06/06	Thu 22/06/06	23	Documentador
25	Elaboracion de memoria	14 days	Fri 23/06/06	Wed 12/07/06	24	Documentador

**Fig. 8.1** Captura de MS Project. Tareas, duración, predecesores y recursos

MS Project ofrece muchas posibilidades y entre algunas de ellas la de generar unas fabulosas gráficas y tablas que ayudan mejor a entender el proceso de seguimiento y control de la planificación de un proyecto.

Por ejemplo, podemos extraer tablas sobre los costes previstos, la carga de trabajo por recurso humano, etc. Una de las gráficas más interesantes es el Diagrama de Gantt, donde se muestra el transcurso en el tiempo de las tareas y como preceden unas de otras. El camino crítico señalado en rojo determina que tareas si se retrasan retardan la consecución del proyecto.

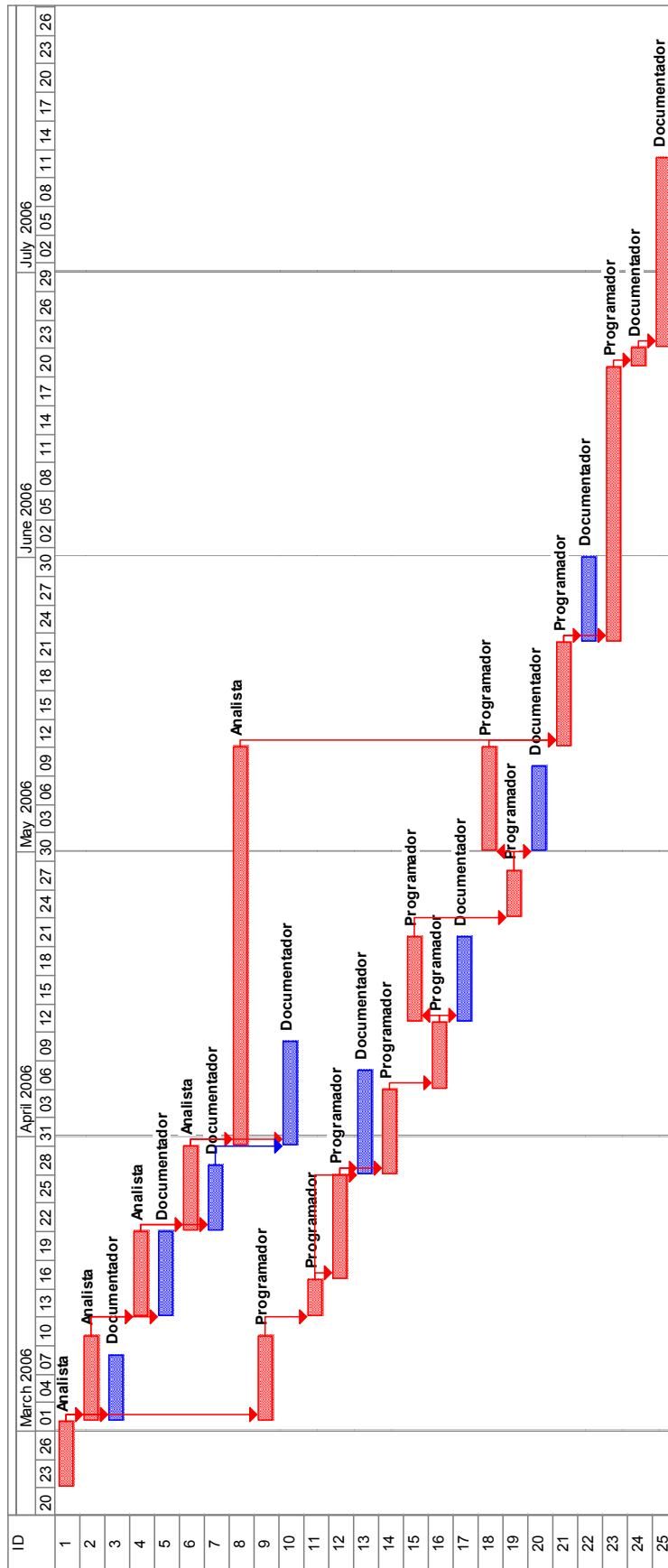


Fig. 8.1 Diagrama de Gantt y camino crítico

El Diagrama de Gantt resulta muy intuitivo y permite ver fácilmente qué tareas requieren especial atención por el bien del cumplimiento del plazo de finalización estimado. Según los datos introducidos y como se ve en la figura la finalización del proyecto está prevista para el día 12 de Julio de 2006. Posteriormente analizaremos la desviación entre la planificación prevista y la real.

### 1.6.3. Estimación de costes

A continuación podemos ver dos tablas en las que se recogen los costes de los recursos humanos y materiales respectivamente.

**Tabla 2.1.** Coste de recursos humanos

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
Analista	25 €/h
Documentador	25 €/h
Programador	30 €/h

El coste de los recursos humanos es el obtenido después de haber revisado otros proyectos anteriores y haber hecho un sondeo en el mundo laboral actual.

**Tabla 3.1.** Coste de recursos materiales

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
Platform Builder 5.0	1.099 €
Placa Mini - ITX EPIA MII	308,10 €

Los costes de los recursos materiales son extraídos de la página oficial de Microsoft en el caso del Platform Builder, y de un distribuidor oficial de productos de VIA por lo que se refiere a la placa. Veremos posteriormente que el precio de la placa incluye la memoria RAM y la fuente de alimentación.

A continuación veremos una tabla donde se recoge el coste estimado total por recurso humano:

**Tabla 4.1.** Coste estimado total de los recursos humanos

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
Analista	2.625 €
Documentador	7.000 €
Programador	9.480 €

Según los datos obtenidos podemos calcular el coste estimado total del proyecto sumando los costes de los recursos materiales y el coste estimado total de los recursos humanos.

**Tabla 51.** Coste estimado total del proyecto

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
<b>Recursos humanos</b>	<b>19.105 €</b>
Analista	2.625 €
Documentador	7.000 €
Programador	9.480 €
<b>Recursos materiales</b>	<b>1.407,1 €</b>
Platform Builder 5.0	1.099 €
Placa Mini - ITX EPIA MII	308,1 €
<b>TOTAL</b>	<b>20.512,1 €</b>

No se debe olvidar que este sería el coste estimado total del proyecto porque contamos con unos costes estimados de recursos humanos. Los recursos humanos tienen el mayor peso en el presupuesto por lo que estudiar su desviación final será interesante para corregir el coste total previsto.

Este presupuesto está previsto en base de que se disponga de tres recursos humanos diferentes: Analista, Documentador y Programador.

En el desarrollo de este proyecto los tres recursos son la misma persona por lo que no se ajustara a la realidad, no obstante el coste estimado total resulta más realista porque tiene en cuenta tres personas físicas diferentes como resulta en el mundo laboral.

En la etapa de planificación se optó por tener en cuenta el proceso de aprendizaje para elaborar este proyecto. En el mundo laboral, si un cliente encarga un proyecto que exige de un análisis y estudio previo este se incluirá en el presupuesto.

Los conocimientos adquiridos serán de gran utilidad puesto que nos podemos encontrar nuevos proyectos que exijan de ellos.

## CAPÍTULO 2. WINDOWS CE Y PLATFORM BUILDER

### 2.1. Windows CE

#### 2.1.1. Orígenes y evolución de Windows CE

Microsoft Windows CE es un sistema operativo de 32 bits diseñado para dispositivos que poseen limitaciones respecto un PC convencional.

Ejemplos de dispositivos para los que fue creado Windows CE son:

- Agendas personales
- Sistemas empotrados
- Sistemas dedicados
- WebPads
- TPV
- Cámaras de fotos
- Teléfonos inteligentes, etc.

El objetivo principal de Windows CE es dar soporte mediante un sistema operativo personalizado al dispositivo sobre el que se ejecutará. Esto hace que Windows CE utilice los mínimos recursos posibles y es por este motivo que encaje en dispositivos con limitaciones técnicas.

Como hemos visto en el capítulo de estudio y análisis, cada vez existen más dispositivos con funciones muy específicas y que tienen limitaciones inherentes, como por ejemplo el uso de baterías. Microsoft pensó crear Windows CE como un sistema capaz de adaptarse a cualquier dispositivo sea cual sea su limitación.

Windows CE comenzó en los laboratorios de Microsoft bajo el proyecto Pulsar. Este proyecto contó con estrictas limitaciones de hardware con el objetivo de producir un sistema operativo que verdaderamente tuviera éxito comercialmente.

A continuación veremos la evolución del sistema Windows CE:

- **Windows CE 1.0**

Nació con el objetivo de ejecutarse en una plataforma que no fuese un PC y su código fuente fue escrito desde cero.

Es la primera versión del sistema operativo Windows CE y empezó a comercializarse a principios del año 1996.

Enseguida tuvo acogida entre fabricantes y usuarios de dispositivos como los comentados anteriormente debido a sus características entre las que figuran:

- Compatibilidad con sistemas tradicionales Windows porque soporta la API Win32, aunque no es compatible en su totalidad ofrece facilidad en portar las aplicaciones a este sistema fácilmente.
  - Tiene un consumo limitado de RAM puesto que todos sus componentes pueden ser ejecutados en una ROM. Esto interesa a los fabricantes para reducir costes en el producto.
  - Es independiente al hardware del sistema porque es compatible con las más comunes arquitecturas de CPUs de 32 bits.
- **Windows CE 2.0**

La siguiente versión de Windows CE tuvo su primera aparición a mediados de 1997. Aunque mantiene la misma estructura que su predecesor aporta significativas mejoras:

- Soporte para procesadores Intel y AMD
  - Serie 82x de procesadores PowerPC
  - Pantalla con resolución de hasta 24 bits
  - Conexiones LAN utilizando protocolo NDIS
  - ActiveX
  - Java Virtual Machina
  - VB Script
  - Subconjunto MFC
  - Soporte FAT
  - Soporte USB
  - Soporte IRDA
  - Soporte de impresión
- **Windows CE 3.0**

La versión 3.0 estuvo disponible en la primavera de 2000 y pretende competir con el sistema conocido como Palm-OS dirigido a PDAs. Microsoft creó el concepto de Pocket PC o PC de bolsillo, mejorando el sistema anterior.

En la nueva entrega de Windows CE se mejoran los siguientes aspectos:

- Nuevos servicios del kernel con interrupciones jerarquizadas y mejora en la respuesta de threads, lo que hace que el sistema responda mejor a tales eventos
- Mayor capacidad de almacenamiento

- Mejora de las comunicaciones entre procesos
  - Mejora en la interoperabilidad con otros sistemas Windows
- **Windows CE 4.0**

Una de las versiones con más éxito fue la versión 4.0 de Windows CE que tuvo su aparición en marzo de 2002. Aunque su implantación no quedó patente hasta mediados de 2003 con el sistema Windows Mobile 2003 para PocketPC.

Esta vez Windows Ce se presenta como un sistema muy robusto y multifuncional capaz de trabajar con diversas arquitecturas. Microsoft incorporó una versión reducida de la herramienta Platform Builder en sus versiones 4.1 y 4.2. Pero esta herramienta tuvo la problemática de no poder adaptarse a cualquier dispositivo empotrado lo que dificultaba la acogida entre usuarios finales.

- **Windows CE 5.0**

Es la última versión que existe en el mercado de Windows CE. Su lanzamiento fue en mayo de 2005 y provocó un salto cuantitativo y cualitativo a este sistema. Tuvo su apogeo con el lanzamiento de Windows Mobile 5.0 (WM 2005).

Esta nueva versión aportaba numerosos avances, entre ellos:

- Mejora en los visores ofimáticos
- Mejora en Windows Media Placer
- Ampliación del paquete multimedia
- Mejora en red de área personal de Bluetooth
- Simplificación del sistema de Informe de errores.

Esta será la versión bajo la que desarrollamos nuestro sistema operativo.

### **2.1.2. Objetivos de Windows CE**

Windows CE tiene como objetivo trabajar sobre sistemas empotrados y ordenadores de bolsillo. Los sistemas empotrados interactúan entre Windows CE y otros sistemas, como por ejemplo con un sistema de control de un proceso industrial.

Estos sistemas requieren una fácil integración en cualquier entorno por muy diverso que este sea. Por ejemplo, debe existir la facilidad de desarrollar un sistema destinado para gestión industrial como otro enfocado para un entorno multimedia, es la versatilidad que se le exige a Windows CE.

A su vez, los ordenadores de bolsillo están en auge ya que integran muchas funcionalidades con son: agenda, calendario, gestión de correo, etc. Además

también permite la visualización de documentos, imágenes e incluso reproducción de medios multimedia.

A continuación se describen dos de los dispositivos más comunes que ejecutan Windows CE:

- **Handheld PC**

Dispositivo electrónico que suele contar con una pantalla de 480x240 o 640x240 en escala de grises. Normalmente tiene soporte para un teclado reducido y pantalla táctil a modo de sustituir al ratón.

En general poseen un puerto serie y otro de infrarrojos de 115 Kbps. Estos dispositivos cuentan además con slots de memoria como tarjetero PCMCIA o Compact Flash.

Son sistemas muy limitados en cuanto a capacidad RAM y de almacenamiento, por ello la configuración de la memoria física es de vital importancia y por este motivo se crean aplicaciones reducidas especiales.

Debido a los problemas de memoria, en estos sistemas se comparte la memoria ROM y RAM del sistema para optimizar la carga de aplicaciones y la velocidad de respuesta del sistema.

Normalmente estos dispositivos no cuentan con elementos de almacenamiento masivo como un disco duro, aunque generalmente se opta como hemos dicho antes por tarjetas de memoria externas.

- **PocketPC**

Son dispositivos muy reducidos en cuanto a su tamaño y tienen como objetivo ser una extensión de nuestro PC de escritorio pero en versión reducida. Tienen la capacidad de ser un organizador personal así como de tener aplicaciones de ofimática, reproducción de medios, navegación por Internet, etc. La entrada de datos suele ser táctil gracias a la utilización de un lápiz

Su memoria también se compone de RAM y ROM. La memoria ROM puede ser de dos tipos: ROM NOR más pequeña, de vida corta, lenta pero permite ejecución totalmente en tiempo real; ROM NAND más rápida, barata y de mayor vida pero necesita de RAM para poder escribir y leer.

Resumiendo, Windows CE tiene que ser capaz de trabajar con multitud de procesadores de 32 bits y debe ser adaptable en su configuración para amoldarse al dispositivo objetivo. A su vez, debe ser capaz de tener conectividad con otros sistemas puesto que el mercado actual así lo requiere.



### 2.1.3. Características de Windows CE

La nueva versión de Windows CE (WCE 5.0) debe proporcionar un entorno de desarrollo integrado donde el desarrollador disponga de una interfaz gráfica y un terminal de comandos en la misma herramienta.

Se potencia la seguridad y fiabilidad del sistema migrando automáticamente algunos componentes de seguridad. El reporte de informes de error permite obtener información acerca de los problemas ocurridos.

En esta nueva versión se potencia mucho la vertiente multimedia del sistema por lo que Microsoft añade soporte para Direct3D, un modelo de programación de gráficos propio de la librería DirectX. Esto está destinado a potenciar los equipos multimedia basados en Windows CE.

Windows CE soporta aplicaciones creadas con diferentes lenguajes de programación. Se pueden distinguir como:

- Aplicaciones de la plataforma .NET

Soporta varios lenguajes de la plataforma .NET como el C#.NET y VB#.NET. Además es compatible con el paquete Compact Framework .NET.

- Aplicaciones de Embedded Visual C++

Plataforma especialmente diseñada para el desarrollo de aplicaciones para dispositivos empotrados. Aporta flexibilidad, rendimiento y control.

Una de los avance en el diseño por módulos de Windows CE es que permite la implementación de drivers de cualquier hardware específico sin necesidad de soporte por parte de Microsoft. Windows CE permite el desarrollo de los drivers siguientes:

- Drivers de almacenamiento
- Drivers multimedia
- Drivers de red (NDIS, 10/100 Ethernet, Bluetooth, Wifi, etc.)
- Drivers de puerto serie y paralelo
- Drivers para Smart Cards, PCMCIA, Compact Flash, USB y IEEE1394

Los drivers en Windows CE son parte de librerías dinámicas conocidas con el acrónimo de DLLs. Estas están basadas en el estándar API de Windows Win32 para procesadores de 32 bits.

Con el objetivo de dar soporte a numerosas arquitecturas de computación, Microsoft idea los BSPs o Board Support Packages. Estos paquetes se instalan en nuestra estación de desarrollo e implementa los drivers necesarios para desarrollar un sistema WCE para ese dispositivo en particular.

Podemos encontrar BSPs para las siguientes arquitecturas de computación:

- ARM
- MIPS
- CEPC
- X86
- SuperH

Windows CE ofrece soporte para ejecución en tiempo real sobre todas las arquitecturas disponibles. Esta característica permite al sistema operativo interactuar con eventos externos en un tiempo corto, asegurando la prioridad necesaria y sin bloquearse.

Los requisitos para trabajar con Windows CE 5.0 sobre una estación de desarrollo son:

- Procesador Pentium II de 266-MHz o posteriores; se recomienda un procesador de 500-MHz
- Microsoft Windows 2000 Professional con Service Pack 4 o Windows XP Professional con SP1
- Microsoft .NET Framework, versión 1.1
- 128 MB de RAM; 192 MB de RAM son requeridos para poder ejecutar el emulador
- 1.7 GB de espacio en disco duro para una típica instalación con un solo microprocesador; aproximadamente 17 GB para instalar el producto entero, incluyendo el Platform Builder.
- CD-ROM o DVD-ROM drive
- Puerto paralelo bidireccional para cargar una imagen del sistema operativo o bien conexión Ethernet.

#### **2.1.4. Arquitectura de Windows CE**

Windows CE es un sistema operativo modular compuesto de varios elementos software. Estos elementos son interfaces compatibles con Win32 y estas interfaces permiten que cada uno de los elementos software pueda integrarse entre sí. Cada elemento comprende varios componentes de nivel más bajo y los diseñadores del sistema tienen la posibilidad de incluir o excluir estos componentes de nivel más bajo según sus necesidades.

Los principales elementos que conforman el sistema operativo son: la capa de abstracción OEM denominada OAL (OEM Abstraction Layer), que incluye la administración de poder, los drivers de dispositivos y los servicios de las tarjeta de PC (PC Card); el Kernel de Windows CE, USER, GDI, sistemas de archivos y bases de datos; los protocolos de comunicaciones IRDA y TCP/IP; las distintas APIs; la conectividad remota, Internet Explorer para Windows CE y el shell. Un diseñador puede incluir o excluir algunos de estos elementos (por ejemplo, el Internet Explorer) según sus necesidades.

La siguiente figura muestra los elementos de la arquitectura de Windows CE:

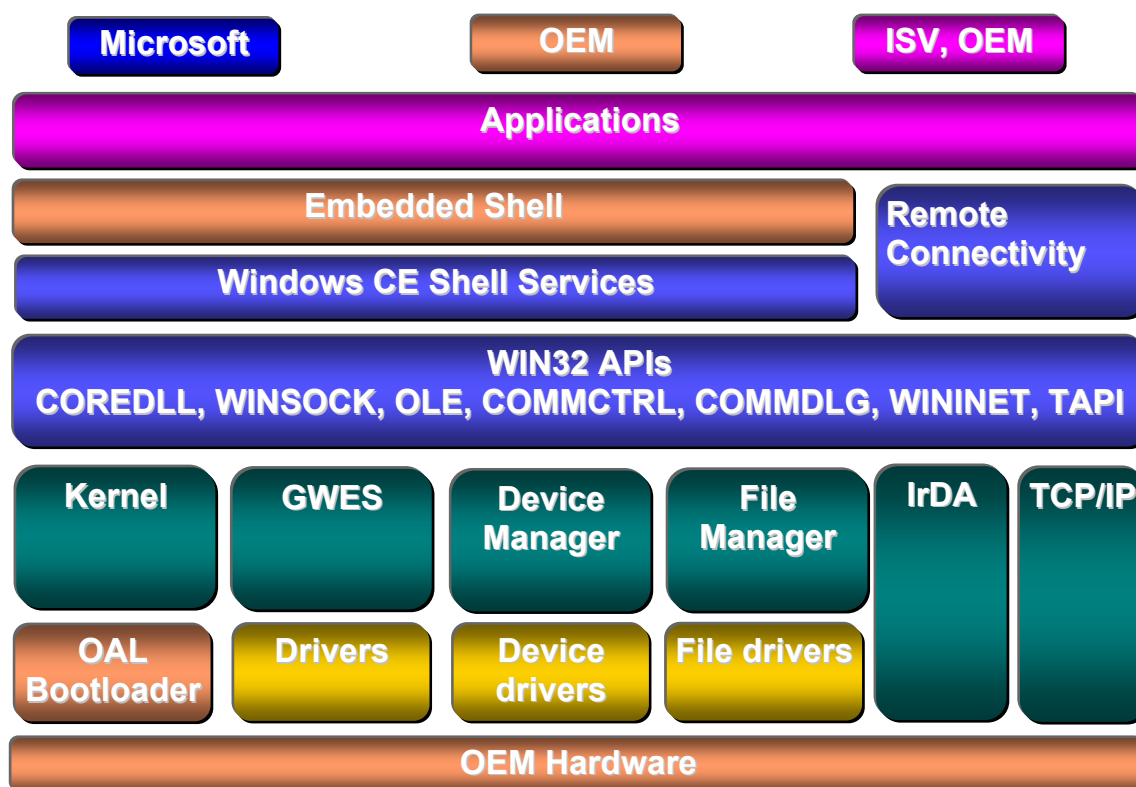


Fig. 1.2 Elementos que componen la arquitectura de WCE

#### 2.1.4.1. OAL (OEM Adaption Layer)

El OAL (OEM Adaptation Layer, es decir, la interfaz entre Windows CE y el aparato físico) permite que los fabricantes programen sus propias funciones de control de hardware (drivers) adaptándose a la interfaz que se les proporciona, que incluye rutinas de servicio a interrupciones e interfaces para funcionalidades específicas del hardware como puede ser la alimentación, permitiendo que el sistema operativo se ejecute en cualquier configuración de hardware o características como el "encendido instantáneo" que permiten que los dispositivos Windows CE sean apagados y luego encendidos instantáneamente si el dispositivo tiene RAM no volátil (respaldada con OEM baterías).

Se ha mejorado la manipulación de componentes mediante bibliotecas de código, estructuras de directorios para reutilizar código, archivos centralizados de configuración, y una arquitectura consistente a través de distintas familias de procesadores y plataformas de hardware.

En la figura anterior se observa como Windows CE ofrece una capa (OAL) para la adaptación de los dispositivos. El fabricante es el encargado de implementar las funciones correspondientes al nivel OAL.

### 2.1.4.2. Controladores (drivers) de dispositivos

Windows CE es compatible con dos tipos de drivers:

- Drivers nativos: Son los drivers propios del sistema operativo para el hardware que está siempre presente en el dispositivo, como por ejemplo el teclado.
- Drivers instalables: Son aquellos drivers suministrados por los fabricantes e instalables en tiempo de ejecución para periféricos que pueden agregarse. Esto permite que los fabricantes creen sus controladores de una forma bastante directa a través de las interfaces disponibles.

Windows CE 5.0 proporciona todas las facilidades posibles para que los fabricantes puedan desarrollar drivers para muchos tipos de dispositivos, esto se debe a la política de marketing comercial de Microsoft. Los tipos de dispositivos a los que Microsoft proporciona soporte son:

- Dispositivos multimedia
- Dispositivos de almacenamiento
- Dispositivos de red
- Dispositivos de entrada
- Dispositivos serie y paralelo
- Periféricos Smart Card
- Periféricos de expansión (PCMCIA, Compact Flash, USB y IEEE 1394)

### 2.1.4.3. APIs

Windows CE resulta compatible con la interfaz de programación de aplicaciones del sistema Windows convencional, llamada Win32. Más particularmente, soporta un subconjunto de este conjunto de funciones. Este hecho motiva la creación y/migración de aplicaciones para WCE porque los desarrolladores ya conocían esta API.

Además del protocolo TCP/IP, Windows CE provee varias de las APIs para comunicaciones de Windows, incluyendo sockets de Windows, comunicación por puerto serie, telefonía, WinINet (FTP y HTTP), LAN (NDIS), etc.

Windows CE aporta una herramienta de sincronización entre un PC y el dispositivo llamada ActiveSync y su arquitectura es abierta para que cualquiera pueda crear su propio sistema de sincronización personalizado.

### 2.1.4.4. Shell

La shell provee a un usuario de la interfaz de objetos necesaria para utilizar las aplicaciones del sistema operativo Windows CE en un dispositivo portátil. Los objetos pueden ser reales, como ficheros y carpetas; virtuales, como la

papelera de reciclaje; o pueden ser objetos remotos a los que se accede a través de una red.

Windows CE proporciona el código de fuente para la shell por lo que el usuario tiene acceso a este código para crear interfaces personalizadas.

#### **2.1.4.5. Kernel**

El kernel del Windows CE es el núcleo del sistema y está encargado principalmente de las siguientes funciones:

- Planificación y sincronización de tareas
- Procesado de interrupciones y excepciones del sistema
- Ejecución de aplicaciones
- Gestión de la memoria virtual del sistema

De la misma forma que en los sistemas Windows tradicionales el kernel es un sistema multitarea con desalojo basado en prioridad. Además incorpora numerosas funciones de sincronización, señalización y eventos.

El kernel de Windows soporta la ejecución de aplicaciones en la ROM o en la RAM. El kernel está provisto de una rutina de bajo nivel de servicios de interrupción y baja latencia de hilo. Este es el motivo por el cual Windows CE es un sistema apto para trabajar en tiempo real.

#### **2.1.4.6. Threads y procesos**

Un proceso es la representación en el sistema de que una aplicación se está ejecutando y es único por aplicación ejecutada. Los threads son los diferentes flujos (o tareas) que se ejecutan dentro de un proceso para atender las peticiones de la aplicación.

Windows CE es un sistema operativo multithread, esto significa que puede ejecutar varios threads de un mismo proceso simultáneamente.

Con Windows CE si un proceso tiene varios threads independientes entre sí y tienen la misma prioridad, se ejecutan simultáneamente repartiendo el tiempo del procesador.

Cuando se ejecuta una aplicación se crea un proceso que a su vez dividirá sus funciones en tareas (como mínimo una, que se crea a la vez que el proceso), representadas para el sistema operativo en forma de threads. El proceso gestiona los threads que de él dependen y la interacción entre ellos.

## **2.2. Platform Builder**

### **2.2.1. Descripción de Platform Builder**

Podemos definir Platform Builder como una herramienta de diseño de sistema operativo basado en Windows CE.

Microsoft ha dispuesto su sistema Windows, el más utilizado, y concretamente su versión reducida, Windows CE, sea un sistema que pueda ser desarrollado con este software adquiriendo la licencia pertinente valorada en unos 1000 euros.

Actualmente, la versión disponible y la utilizada para desarrollar la plataforma de nuestro dispositivo es 5.0. Platform Builder 5.0 soporta el desarrollo de sistemas basados en Windows CE 5.0 por lo que podremos añadir servicios de desarrollo como por ejemplo el paquete Compact Framework 2.0.

El Platform Builder de Microsoft genera un Ambiente Integrado de Desarrollo (IDE) para realizar diseños empotrados a medida, basados en el sistema operativo Windows CE.

Platform Builder para Windows CE 5.0 viene con todas las herramientas de desarrollo necesarias para diseñar, crear, construir y para probar una plataforma basada en Windows CE. El IDE proporciona un solo espacio de trabajo integrado donde se puede trabajar en diseños y proyectos.

Con esta herramienta de desarrollo seremos capaces de crear un sistema operativo a medida de nuestra aplicación. Para poder realizar esto necesitamos implementar un paquete de soporte de la placa (BSP) y un cargador de arranque concreto de nuestro dispositivo. Platform Builder permite elegir una plantilla de diseño para crear un diseño de sistema operativo, construir la imagen de arranque y descargar esta imagen a nuestro dispositivo.

El funcionamiento en tiempo real es esencial para las respuestas críticas que se requieren en sistemas empotrados de alto rendimiento. Hay muchas definiciones del trabajo en tiempo real. En el contexto de la familia de sistemas operativos empotrados de Windows, utilizaremos la definición que fue adoptada por el Control de Arquitectura Abierto y Modular (OMAC): un sistema en tiempo real "duro" es un sistema que fallaría si sus requisitos de sincronización no son resueltos; un sistema en tiempo real "suave" puede tolerar variaciones significativas en la entrega de servicios del sistema operativo como interrupciones o contadores de tiempo. Por este motivo, Windows CE es un sistema operativo en tiempo real duro.

Windows CE provee de tecnologías para el desarrollo en tiempo real de las aplicaciones, que incluyen:

- 256 niveles de prioridad de threads: De esta forma proporciona mayor flexibilidad en controlar la programación de threads de un sistema empujado.
- Interrupciones jerarquizadas: Permite que las interrupciones de alto nivel de prioridad se ejecuten antes para no tener que esperar la finalización de una interrupción de prioridad baja. El núcleo puede jerarquizar tantas interrupciones según la capacidad de la CPU.
- Inversión de la prioridad: Consiste en retrasar la ejecución de un thread de prioridad de alto nivel, porque otra aplicación está utilizando los mismos recursos que este thread. Para corregir esta situación y liberar el thread de alta prioridad, Windows CE permite que el thread de baja prioridad herede el thread más crítico y funcione hasta que libere su uso del recurso.

### 2.2.2. Descripción del catálogo

Después de crear un diseño de un sistema operativo con Platform Builder podemos ver la vista de diseño de sistema operativo y la ventana del catálogo a la vez en la pantalla principal y desde aquí podemos modificar nuestro diseño.

La vista de diseño operativo muestra todos los elementos del catálogo que se encuentran en nuestro diseño. El catálogo muestra los elementos que se pueden añadir al diseño actual, incluyendo BSPs, funcionalidades del núcleo del sistema operativo, capas de transporte y drivers de dispositivos.

Además, es posible mostrar elementos adicionales propios o creados por tercera parte. Se le llaman elementos de tercera parte a aquellos elementos creados por un fabricante para dar soporte a su dispositivo. El catálogo permite importar nuevos elementos aportando los archivos de información del elemento y que poseen extensión “\*.cec”.

El catálogo y la vista de diseño del sistema operativo también soportan la visualización de carpetas para agrupar diferentes elementos del catálogo.

La vista de diseño del sistema aparece una vez creada el diseño del sistema y describe únicamente las funcionalidades de nuestro sistema, mientras que el catálogo muestra todas las funcionalidades disponibles en la instalación de la herramienta Platform Builder.

Un grupo de elementos del catálogo describe la interfaz y funcionalidad que ofrecen cada uno de los elementos que pertenecen al catálogo. No poseen la capacidad de ejecutarse por sí mismos sino que necesitan ejecutarse en el sistema y sobre el dispositivo o el emulador.

Platform Builder permite que haya más de una implementación para un grupo del catálogo. Por ejemplo, “Display” es un grupo de elementos del catálogo

para los drivers de pantalla. Si se añade este grupo al proyecto se reserva el espacio para el driver de pantalla sin especificar de qué driver se trata.

Existen numerosas implementaciones de drivers estándares para pantalla, cada uno tiene la función de implementar un único tipo de driver.

Muchos de los elementos del catálogo se incluyen dependiendo la plantilla de diseño o del asistente que siguió el desarrollador en la creación del proyecto. Un elemento se incluye en el sistema durante el proceso de construcción de la imagen resolviendo a su vez las dependencias incluidas.

Cada vez que se añade o se elimina un elemento del catálogo o se retoca cualquier otro parámetro que requiera generar el sistema de nuevo, la ventana de los elementos del catálogo y la vista de diseño del sistema se refresca. En caso de que se añadiesen elementos del catálogo debido a dependencias, estas serán mostradas en la pestaña de "Build" en la ventana de salida.

Platform Builder hace sencillo modificar un diseño de sistema operativo. El desarrollador tan sólo debe eliminar y/o añadir aquellos elementos del catálogo que crea necesarios. Debemos tener en cuenta, que sólo se puede eliminar un elemento cuando fue añadido como dependencia después de eliminar el elemento que tenía dicha dependencia.

Para asegurar que un elemento del catálogo no sea eliminado si su elemento padre se elimina se puede añadir como elementos del catálogo especificados por el usuario. Esta opción se visualiza en el mismo momento de añadir el elemento. Asimismo es posible desactivarla escogiendo la opción de eliminar elemento especificado por el usuario.

Para determinar cuáles son los drivers soportados entre el diseño de sistema operativo y el BSP, Platform Builder realiza una comparación entre la lista de los módulos del núcleo del diseño del sistema con la lista de variables disponibles en el BSP.

Después la herramienta IDE automáticamente inserta los drivers necesarios para conocer los requerimientos del sistema. Estos drivers se añaden de forma fija al sistema.

Cada driver de un grupo de elementos del catálogo tiene un driver por defecto, el cual es seleccionado por las variables que contienen el BSP y por el driver por defecto del elemento. Para sobrescribir los drivers por defecto, se debe añadir al sistema el driver específico que se necesite.

### **2.2.3. Platform Builder IDE**

El entorno integrado de desarrollo (IDE) está compuesto por unas herramientas basadas en interfaz de usuario MS Windows para construir, testear y redefinir un diseño de sistema operativo y los elementos del catálogo asociados a él.



La interfaz de usuario está creada para simplificar el entorno de desarrollo y ofrece las siguientes características:

- Barra de menú sensible al contexto:

Los menús dependen del proceso que esté realizando el desarrollador. Por ejemplo, algunos comandos son específicos de un editor y sólo están presentes cuando estas usando el editor. Los comandos no son aplicables si estás trabajando en otro contexto.

- Menús de acceso directo:

En la mayoría de las ventanas y paneles de la interfaz de usuario se puede visualizar un menú de acceso directo haciendo clic con el botón derecho sobre un elemento de la lista, o sobre el fondo de la ventana o panel.

Usar los accesos directos permite ejecutar tareas en pocos pasos. Esto se debe a que muchas tareas de la barra de menú están disponibles como acceso directo.

- Páginas con propiedades editables:

Es posible mostrar las propiedades de un elemento seleccionándolo con el botón derecho y escogiendo "Propiedades" del menú de acceso directo. Las propiedades del elemento se muestran en una hoja de propiedades.

- Barra de herramientas sensible al contexto:

En el momento que se está editando o empezando una nueva actividad automáticamente aparecen las barras de herramientas apropiadas.

Para mostrar otras barras de herramientas disponibles debemos acceder al menú "Herramientas", elegir "Personalizado" y seleccionar aquellas barras que deseemos que estén a la vista.

Muchas de las barras de herramientas son equivalentes a utilizar los comandos disponibles en la barra de menú.

- Teclas de acceso directo con sensibilidad al contexto:

Las teclas de acceso directo cambian dinámicamente dependiendo de la tarea actual que se desempeñe. Es posible editar la configuración de teclas de acceso directo y personalizarlas como más convenga.

- Documentación de ayuda sensible al contexto:

Muchas de las ventanas de diálogo tienen ayuda disponible que describe cómo usar las ventanas de diálogo y otros objetos de la interfaz de usuario.

Se puede acceder a la ayuda contextual de un objeto pulsando la tecla F1 cuando este objeto está seleccionado. Si el objeto tiene ayuda disponible, PB abre la ayuda y muestra la página que describe el objeto seleccionado.

- Barras de herramientas y menús personalizables:

Platform Builder permite cambiar las barras de herramientas disponibles, crear nuevas personalizables, añadir menús a la barra de herramientas, mover o copiar botones y comandos de las barras de herramientas, reordenar menús y cambiar la apariencia de botones y comandos.

En el momento de la creación de un nuevo diseño de sistema operativo, se crea paralelamente un espacio de trabajo o workspace. Un workspace puede contener varios proyectos pero sólo un diseño de sistema operativo.

Un espacio de trabajo contiene los siguientes elementos:

- Incluye un fichero de extensión \*.pbxml, que tiene como propósito definir el diseño del sistema operativo. Este fichero contiene información acerca del espacio de trabajo, como por ejemplo la lista de proyectos. Este archivo es utilizado para abrir un workspace.
- Un fichero de proyecto, con toda la información necesaria para crear el catálogo de elementos definidos por el usuario. Este fichero tiene como extensión \*.pbpxml.
- Un diseño de sistema operativo, que describe cuáles son los elementos del catálogo que definen las características del propio sistema operativo.

Después de crear un workspace se pueden añadir proyectos y configuraciones al proyecto existente o al diseño del sistema. Se puede crear un espacio de trabajo de dos formas:

- Utilizando el asistente de nueva plataforma.
- Copiando los ficheros de un workspace a otra estación de desarrollo.

Si procedemos a crear un espacio de trabajo de la segunda forma citada, puede ocurrir que el diseño esté pensado para un microprocesador cuyo BSP no esté instalado en la nueva estación de desarrollo. Esto provoca que la parte de diseño correspondiente a este microprocesador sea obviada y excluida de la configuración activa de construcción del sistema.

Platform Builder permite ver y cambiar los contenidos de un proyecto. En la ventana del espacio de trabajo encontramos la vista de ficheros, la vista de clases y la vista de recursos donde se pueden ver los detalles del proyecto. La vista de ficheros muestra las relaciones de los proyectos y ficheros que están incluidos en el diseño del sistema. Esta relación se refleja en la organización de los ficheros en el disco duro.

Cada ventana es jerárquica y se pueden expandir sus carpetas y otros elementos de la ventana para ver sus contenidos.

Platform Builder provee las siguientes opciones para la gestión de proyectos:

- Añadir archivos al proyecto.
- Insertar proyectos al espacio de trabajo y borrar proyectos del mismo.
- Configurar las opciones del proyecto.

#### 2.2.4. Estructura de directorios de PB

Podemos clasificar los directorios disponibles en Platform Builder como:

- Imagen de tiempo real y directorios del sistema

Describen la imagen de tiempo real y los directorios incluidos cuando se instala Platform Builder. Cuando se instala WCE en el equipo de desarrollo se crea la jerarquía de directorios. Si esta instalación se lleva a cabo en el directorio por defecto de Windows CE, el directorio raíz se llamará "WINCE500".

Platform Builder crea algunos directorios durante la instalación y crea otros en la construcción de la imagen de tiempo real. La estructura de los directorios de cada diseño varía respecto otro en función de los elementos del catálogo que incluya el sistema.

El directorio "WINCE500" contiene los siguientes directorios:

**Tabla 1.2.** Directorios del sistema y de la imagen de tiempo real

<b>CRC</b>	<b>Un fichero llamado Crc.ini que provee información sobre ficheros de la instalación de PB</b>
<b>Others</b>	<b>Ficheros de tiempo real, archivos binarios para la construcción de archivos de imagen, archivos de registro y todos los ficheros usados para añadir elementos a nuestra imagen.</b>
<b>PBWorkspaces</b>	<b>Contiene los espacios de trabajo creados por el usuario</b>
<b>Platform</b>	<b>Ficheros y directorios de un dispositivo específico</b>
<b>Private</b>	<b>Código fuente para el sistema WCE</b>

<b>Public</b>	<b>Código fuente para el sistema WCE</b>
<b>SDK</b>	<b>Herramientas y librerías de soporte disponibles para PB</b>

La estructura de directorios de PB permite desarrollar múltiples proyectos usando un único árbol de directorios. Cada árbol de directorios de un sistema operativo puede soportar a su vez varias CPUs. Esto resulta muy útil porque PB utiliza los directorios “Public” y “Private” para almacenar el código fuente de cada dispositivo específico.

Cuando se instala Platform Builder, los archivos de código fuente de dispositivos específicos crean directorios por cada uno de los procesadores que se haya seleccionado. Por ejemplo, si instalamos soporte para un hardware con un procesador CEPC y otro ARM la estructura creada sería:

- %\_WINCEROOT%\Platform\ARM
- %\_WINCEROOT%\Platform\CEPC

Estos archivos fuente permiten a la imagen creada de nuestro sistema el soporte de los dispositivos utilizados.

Cuando creamos un nuevo diseño utilizando el asistente de nueva plataforma se crea un directorio a razón de nuestra elección. Por ejemplo, si hemos creado un diseño de sistema operativo llamado “Myimage”, el directorio creado sería:

- %\_WINCEROOT%\PBWorkspaces\Myimage
- Directorios de código fuente compartido

Es un listado de directorios que contienen código fuente compartido y sólo están presentes si se escoge instalar el código fuente compartidos durante la instalación de PB en el equipo de desarrollo.

En la instalación típica de WCE, el código fuente se encuentra en los directorios “%\_WINCEROOT%\Public” y “%\_WINCEROOT%\Platform” que se encuentran en el directorio que se instaló Platform Builder.

Cuando se instala el código fuente compartido se instala también este código dentro de los directorios “%\_WINCEROOT%\Private”.

### 2.2.5. Compiladores para microprocesadores

MS Windows CE soporta varias familias de microprocesadores y provee herramientas de compilación, assemblers y funciones intrínsecas.

Las herramientas proporcionadas permiten una completa optimización dependiendo de la arquitectura del microprocesador.

En muchos casos debemos instalar un BSP (Board Support Package) para tener compatibilidad con algún dispositivo basado en alguna de las arquitecturas soportadas por WCE. En nuestro caso, debemos proceder a instalar el BSP de VIA EPIA MII con el fin de disponer de los drivers específicos del dispositivo.

Los procesadores soportados por WCE son los siguientes:

- X86
- ARM para arquitecturas v4, v4T, Thumb, v5TE e Intel XScale
- Renesas SuperH SH4
- MIPS

## **2.3. Evaluación inicial de Platform Builder**

### **2.3.1. Experiencias preliminares**

La impresión que causa la herramienta Platform Builder en su iniciación se podría resumir como un entorno de desarrollo complejo. No es un entorno intuitivo, los catálogos de servicios y aplicaciones son muy extensos y está clasificado de forma demasiado modular que invita a la pérdida de un orden en el desarrollo.

Resulta muy difícil buscar directamente algún módulo en particular. A veces, un módulo requiere dependencias de otros. Muchos de los módulos y servicios disponibles para nuestra plataforma tienen dependencias de otros. En la mayoría de ocasiones, estas dependencias se resuelven transparentemente al usuario pero en otras ocasiones las dependencias son relevantes para el proceso de compilación y es el usuario quien debe añadirlas. Esto puede provocar equivocaciones, olvidos de algún módulo o no encontrarlo hasta pasado un buen tiempo.

La ayuda es útil y muy similar a otros entornos de desarrollo y aplicaciones de Microsoft por lo que resulta intuitiva. La ayuda se ve limitada en cuanto alguna prestación que se hecha en falta, como por ejemplo, las dependencias de un módulo, detección, definición y reporte de errores/warnings. Son de gran interés los tutoriales básicos que figuran en la ayuda disponible en Platform Builder y ayudan al usuario a iniciarse.

La tarea más ardua es la compilación y construcción de la imagen puesto que el tiempo de espera es muy elevado para realizar ambas tareas. Si realizamos un pequeño cambio se exige de nuevo la compilación y construcción de la imagen del sistema aunque se reduce el tiempo de espera considerablemente.

Además el tiempo de espera depende directamente del tamaño de la imagen del sistema, en nuestro caso la imagen es de unos 20 MBytes aproximadamente en su versión definitiva. Mis experiencias personales demuestran que las esperas superan en numerosas ocasiones los veinte minutos dependiendo del equipo y su carga, por lo que me imagino que la compilación de imágenes de sistema muy extensas deben provocar esperas desesperantes en según que equipo.

La compilación provoca muchos archivos temporales propios de la compilación y construcción del sistema operativo. La totalidad de estos archivos suponen una carga considerable de tamaño efectivo en disco duro. Esto se ve incrementado si se está iniciando con esta herramienta y pretende tener un avance paulatino en las imágenes creadas, por lo que finalmente se dispone de numerosas imágenes que requieren un gran espacio de disco.

Por último, se puede apreciar que con el dominio básico de pocos parámetros podemos crear una imagen de forma rápida gracias al asistente.

### 2.3.2. Primeros pasos con Platform Builder

Para definir esta plataforma lo primero que debemos tener en cuenta es el interfaz principal donde podemos encontrar todos los menús de acceso a las funciones principales que nos ofrece Platform Builder.

Cuando abrimos el Platform Builder nos encontraremos con la siguiente pantalla:

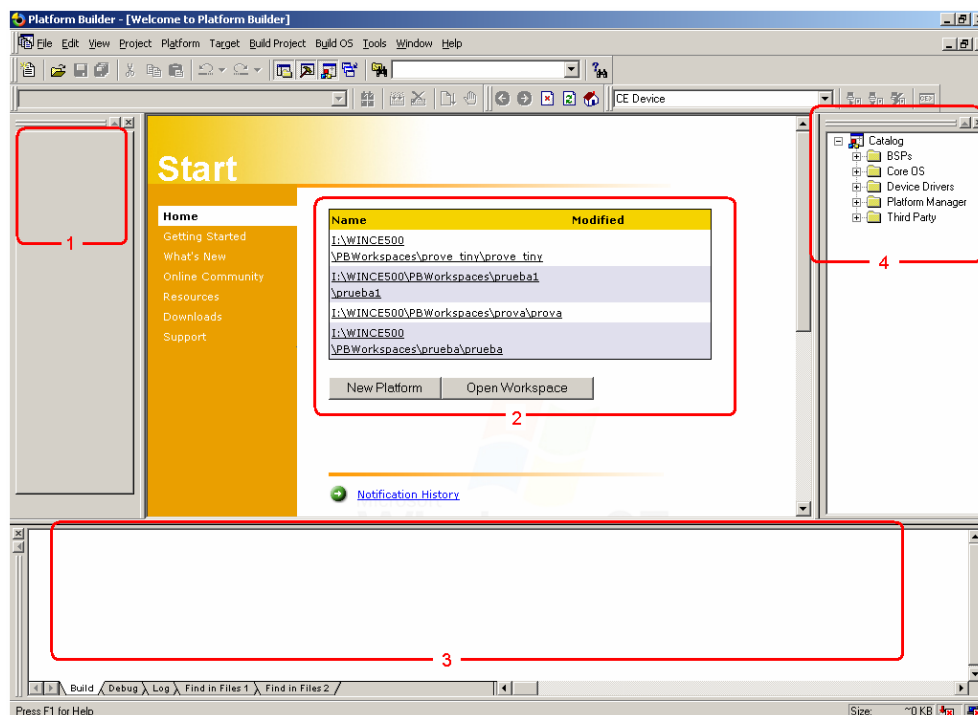


Fig. 1.2 Captura Platform Builder. Entorno principal de trabajo

Como vemos en la imagen anterior, destacan cuatro zonas en la pantalla principal de Platform Builder:

1. **Módulos/servicios añadidos a nuestra imagen:** Corresponde a los módulos que ya estarán disponibles en nuestro sistema operativo.
2. **“Workspaces” recientes:** Muestra un listado de los proyectos que utilizamos en antiguas entradas a la herramienta PB.
3. **Ventana de seguimiento:** Posee varias ventanas y en ellas se pueden observar el proceso de compilación y debug del sistema. Nos muestra información sobre cada uno de los servicios añadidos y compilados, además de informarnos sobre los “warnings” y errores derivados de la compilación del sistema, etc.
4. **Módulos/servicios disponibles:** Muestra un listado completo de todos los módulos y servicios que ofrece Platform Builder.

A partir de aquí, el primer paso que debemos llevar a cabo es crear un nuevo proyecto. Como en cualquier entorno Windows el menú principal que esta situado en la parte superior y nos permite acceder a todas las funciones que realiza el Platform Builder.

Por este motivo Platform Builder se reafirma como una herramienta de desarrollo que se asemeja mucho a otros entornos de programación anteriores de la marca Microsoft. La distribución de los elementos en el entorno es muy intuitiva. Sin embargo, los primeros contactos prevén que es un entorno complejo porque posee muchos submenús que a priori no conocemos pero que después no se utilizan o bien, si se da uso, es de una forma superficial y no determinante a la hora de crear una imagen propia.

Para crear una plataforma nueva accedemos al menú superior “File” -> “New Platform” y automáticamente nos cargará el asistente de creación de una nueva plataforma donde nos permite elegir entre una serie de procesadores genéricos y unos módulos básicos que dependen de nuestra primera elección.

El asistente facilita enormemente la creación de una nueva imagen, de no utilizarlo debemos añadir todos los módulos mínimos necesarios para que funcione la plataforma creada. Esto lleva a confusión por los numerosos servicios y módulos a añadir.

Por este motivo se clasifica la evolución de la plataforma diseñada para nuestro dispositivo en tres niveles. El primer nivel corresponderá a la parte básica del sistema para que arranque, el siguiente escalón será añadir el soporte gráfico al anterior sistema y por último conseguir un sistema basado en el BSP de la placa y con todos los servicios requeridos.





## CAPÍTULO 3. DISEÑO DEL SISTEMA OPERATIVO

### 3.1. Como crear un sistema operativo WCE

#### 3.1.1. Contexto

Una vez realizada la descripción y análisis sobre Windows CE y la herramienta de desarrollo Platform Builder podemos llevar a cabo el proceso de diseño de nuestro sistema operativo.

El proceso de diseño se ha dividido en tres partes con el fin de definir un orden lógico en el proceso de aprendizaje. Estas partes serán:

- Sistema basado en el núcleo del sistema
- Sistema basado en el núcleo del sistema con interfaz gráfica
- Sistema basado en el Board Support Package de VIA EPIA MII

El sistema basado únicamente en el núcleo de Windows CE es muy limitado pero ocupa pocos KBytes. Nos ayudará a familiarizarnos con el entorno de trabajo.

El sistema que incorpora interfaz gráfica está también muy limitado pero en cambio tiene entorno gráfico. Veremos por primera vez el entorno gráfico de un sistema Windows CE creado por nosotros mismos. Aprenderemos como añadir módulos y servicios al núcleo del sistema para que tenga interfaz gráfica.

En la última etapa de diseño instalaremos el paquete de soporte de la placa o Board Support Package (BSP) del modelo VIA EPIA MII. Este paquete proporciona a Platform Builder los drivers necesarios para crear un sistema Windows CE para este hardware.

Aprenderemos a añadir nuevas aplicaciones como MS Messenger y personalizaremos nuestra plataforma.

Finalmente, se añadirá la aplicación al sistema para poder ejecutarla en el dispositivo final.

En caso de obtener la imagen final del sistema operativo se conseguirá el objetivo principal de este proyecto. De la misma forma, si se consigue la correcta integración con la aplicación habremos cumplido los objetivos globales del proyecto común.

### 3.1.2. Prestaciones y limitaciones del SO

El primer paso para llevar a cabo el diseño del sistema operativo que funcionará sobre un dispositivo es analizar qué prestaciones debemos brindar y, en este sentido, podremos definir que servicios son imprescindibles que figuren en el sistema operativo.

Por este motivo, es de vital importancia definir cuáles son estos módulos y servicios necesarios. Para ello se consultó los demás integrantes del proyecto común por lo que la información extraída es la siguiente:

- Proyecto de Aplicación: Se precisa Compact Framework 2.0, codecs de audio y video y DirectX. Se añade también compatibilidad XML.
- Proyecto de Protocolo de Red: Se necesita la pila TCP/IP y .NET Framework.
- Proyecto de Hardware: Puerto serie.

Una vez se definen finalmente las necesidades de cada una de las partes integrantes del proyecto se valoran las limitaciones impuestas por el sistema operativo Windows CE.

La primera limitación de Windows CE es que no dispone de la librería multimedia DirectX. Esta librería es muy potente y ocupa demasiados recursos como para instalarla y darle uso sobre un dispositivo empujado.

Esta limitación de WCE parecía insalvable en cuanto a la parte de la aplicación, puesto que la responsable de esta parte había enfocado todo su trabajo mediante la librería DirectX.

Sin embargo, investigando sobre el tema se encuentra información acerca de la alternativa brindada en esta versión reducida de Windows. La librería que toma relevancia en WCE a DirectX es DirectShow. Además, según información oficial de Microsoft, es compatible con DirectX.

Pese a esta vía de escape, DirectShow no promete unos resultados tan óptimos como si la reproducción hubiera sido mediante DirectX y seguro presenta limitaciones propias de la librería y de los dispositivos para que fuera diseñada.

Otro elemento de discordia, era la necesidad de tener toda la plataforma .NET Framework sobre Windows CE, en lo relacionado con la parte de protocolo. Después de un tiempo de razonamiento, era lógica la imposibilidad de que el paquete completo Framework .NET estuviese disponible sobre WCE puesto que, en primer lugar, su desarrollo no va dirigido a este sistema operativo, sólo a Windows XP y similares. A esto se une también, que la disponibilidad de espacio de almacenamiento se veía saturada si pretendiésemos instalar un paquete de desarrollo de esa envergadura.

Evidentemente, las funciones principales y necesarias para desarrollar una aplicación sobre C# .NET se encuentran disponibles para su correcto funcionamiento en un dispositivo empotrado con Windows CE. Sin embargo, no podemos creer que será compatible en su totalidad y que su eficacia de ejecución será menor debido a las propias limitaciones hardware.

En cuanto a la petición, en respecto la parte hardware, de dar soporte a comunicación mediante el uso del puerto serie no hubo ningún problema.

Hasta aquí encontramos las limitaciones que nos impone Windows CE sobre las necesidades que demanda nuestro dispositivo del sistema operativo que corra sobre él. Pero además de estas limitaciones, no olvidemos otras más evidentes propias más bien del dispositivo en sí, como son:

- De hardware:

Como dije anteriormente, son las limitaciones propias de los elementos que forman el dispositivo. Este no dispone de los mismos recursos y estos no tienen la misma eficiencia que un PC doméstico. Por ejemplo, no disponemos de un almacenamiento masivo de las características de un disco duro común aunque la placa nos permite su uso. La CPU tiene una frecuencia de CLOCK muy inferior a las frecuencias actuales de PC doméstico. También la memoria dispone de una frecuencia de trabajo inferior a las que ya existen disponibles y la cantidad total con la que se puede trabajar en la placa es reducida.

El procesador VIA Eden™ EPGA no supera los 600 MHz por lo que limita nuestro sistema y la memoria RAM máxima es de 1 GB. Aunque estas características sean suficientes para cualquier reproducción de medios debemos ser conscientes de las limitaciones que suponen.

- Limitaciones físicas:

Aquí podemos destacar aquellas que tienen que ver con la disponibilidad de espacio en el dispositivo para albergar periféricos y otros. En los ejemplos antes citados; aunque la placa permite el uso de disco duro convencional nosotros decidimos no incorporarlo debido al peso que esto supone en el dispositivo y por lo tanto pierde movilidad como dispositivo móvil-empotrado. No disponer de un teclado limita el sistema operativo y la aplicación en sí, la introducción de datos debe ser mínima y sencilla. Pese a esto, debemos recordar que nuestra placa dispone de puertos PS-2 para teclado y ratón.

Una gran ventaja de la placa es que podemos disponer tanto de un disco duro convencional, lector DVD, etc. como también la posibilidad de insertar dispositivos externos como un lápiz USB, una tarjeta PCMCIA o una tarjeta Compact Flash por lo que ofrece un amplio abanico de posibilidades en relación a la expansión de almacenamiento masivo en el dispositivo.

Debemos tener en cuenta que el sistema operativo determina muchas de las prestaciones que estarán disponibles. Por este motivo, cualquier servicio o prestación que deseemos en nuestro sistema deberá contar con el debido soporte por parte del sistema operativo y es que algunas de las prestaciones de la placa elegida se verán limitadas por este mismo factor. Por ello, lo recomendable es dar soporte exclusiva y estrictamente a los servicios que requieren las otras partes de este proyecto. Sin embargo, la versión final del sistema operativo incluirá funcionalidades extra.

Como funcionalidades extra consideramos las que no afectan a la parte de integración de este proyecto y, por tanto, no entorpecen la consecución del objetivo global de este proyecto.

Como prestaciones extras podemos citar las siguientes:

- Microsoft Messenger
- Microsoft Windows Media Player
- Explorador de Windows
- Microsoft Internet Explorer
- Microsoft WordPad
- Word Viewer
- PDF Viewer
- Image Viewer
- Excel Viewer
- PowerPoint Viewer
- Panel de control
- Opciones de conectividad: red local y soporte bluetooth

Un dispositivo es comercial cuando cubre las necesidades para las que fue diseñado. Estas prestaciones extra mejoran la calidad final del sistema operativo generado porque cubre más necesidades del usuario final.

En resumen, las características inherentes al dispositivo final se verán enriquecidas con una plataforma más completa y funcional.

### **3.1.3. Proceso de creación**

El proceso de creación de una plataforma es común independientemente del diseño que se escoja.

En nuestro caso, procedemos a elaborar tres diseños diferentes: Tiny Kernel, Tiny Kernel con interfaz gráfica y la imagen definitiva que se ejecutará en el procesador.

Para los tres diseños el proceso de creación es común por lo que a continuación podemos ver un resumen de cómo podemos elaborar una plataforma siguiendo unos pasos. No obstante, el detalle pormenorizado de los pasos a seguir para elaborar las tres plataformas figura en el Apéndice.

Los pasos para la creación de una plataforma son:

- Arrancar el asistente de creación de imagen
- Elegir un nombre y una ruta para el proyecto
- Escoger BSPs que deseemos que estén disponibles en el workspace
- Escoger el diseño que más nos convenga
- Escoger los elementos que deseemos en caso de que el asistente nos ofrezca hacerlo
- Finalizar el asistente
- Editar las opciones de construcción si fuese necesario
- Generar el sistema haciendo clic en el menú superior “Build OS” -> “SysGen”
- Crear la imagen a partir del sistema generado haciendo clic en el menú superior “Build OS” -> “Make Run-time Image”

Una vez realizados estos pasos deberíamos poder encontrar el fichero, por defecto, llamado nk.bin que corresponde a la imagen del sistema operativo diseñado. Esta imagen se encuentra en el directorio del diseño activo y que sigue la ruta que elegimos en el asistente. Por ejemplo, si la ruta era “J:\WINXP\WINCE500\PBWorkspaces\miimagen” y el diseño activo era “Emulator:x86 Release”, el fichero se encontrará en “J:\WINXP\WINCE500\PBWorkspaces\miimagen\ Emulator:x86 Release”. Podemos copiar la imagen nk.bin al disco duro del dispositivo para arrancar el sistema creado o bien podemos enlazar la estación de desarrollo con el dispositivo mediante conexión Ethernet.

## 3.2. Diseño primera imagen: Tiny Kernel

### 3.2.1. Descripción

El sistema operativo que designado como “Tiny Kernel” se describe como aquel sistema operativo que el objetivo de su diseño es probar simplemente la compatibilidad y la correcta ejecución del sistema operativo Windows CE. Por este motivo, no se añade ningún servicio o módulo extra.

Este sistema operativo correrá bajo el emulador y no sobre el dispositivo. Esto se debe a que este sólo pretende ser un paso pedagógico para completar el aprendizaje de la creación del sistema para el dispositivo.

Además, como el Tiny Kernel sólo posee el kernel o núcleo del sistema, en este caso de Windows CE 5.0, y ya conocíamos de la compatibilidad de este sistema con nuestro dispositivo gracias a la información proporcionada por el fabricante. Esto revalida aún más si cabe la justificación de ejecutar este sistema sobre el emulador de Windows CE.

Por el contrario, se puede destacar que realizar una imagen basada en el Tiny Kernel es interesante desde dos vertientes:

- La educativa, ya que nos ha servido para entender el manejo de PB y el desarrollo de una imagen de SO
- Como modo de prueba, porque debemos familiarizarnos con el entorno de trabajo. Además de servir como prueba a la hora de un posible testeo de Windows CE (Tiny Kernel) en el dispositivo que trataremos la placa VIA EPIA MII.

### 3.2.2. Elementos que componen la imagen

A continuación se ofrece un listado con todos los elementos que componen la imagen basada en el Tiny Kernel, sus servicios y módulos.

Ahora bien, muchos vienen añadidos gracias al asistente de creación de imagen y seguramente carecen de interés aunque no esta de más citarlos. Los elementos son los siguientes:

- Elementos generales del catálogo
  - Device Drivers -> Storage Devices -> Storage Devices
- Elementos específicos de Tiny Kernel (Windows CE devices)
  - Applications and Services Development -> C Libraries and Runtimes
    - Full C Runtime
    - Standard I/O
    - String Safe Utility Functions
  - Core OS Services -> Debugging Tools -> Toolhelp API
  - Core OS Services -> Kernel Features
    - Memory Mapped Files
    - Message Queue – Point – to – Point
    - Target Control Support
  - Core OS Services -> Serial Port Support
  - File Systems and Data Store
    - Compression
    - Storage Manager

### **3.3. Diseño segunda imagen: Tiny Kernel con interfaz gráfica**

#### **3.3.1. Descripción**

En este segundo sistema operativo intentaremos confeccionar una plataforma muy básica pero que posea interfaz gráfica de ahí su nombre Tiny Kernel con IG (Interfaz Gráfica).

Este sistema operativo tiene el objetivo principal de subir un escalón más en el aprendizaje de la creación de sistemas personalizables. Por este motivo podemos catalogarlo como un sistema de ensayo al igual que en el primer caso, la plataforma Tiny Kernel, nos ayudó a familiarizarnos con el proceso.

No obstante, cabe destacar la particularidad de que posea interfaz gráfica puesto que tendemos a buscar nuestro sistema operativo final el cual, sin duda, dispondrá de esta característica ya que la aplicación que correrá sobre el dispositivo en de entorno gráfico en su totalidad. Por ello se puede afirmar que la evolución de un sistema operativo sin interfaz gráfica a uno que sí la tiene es muy notoria ya que podemos encontrar sistemas empujados que encajen en un entorno sin interfaz visual en control y gestión de procesos industriales, mientras que los dispositivos con tintes multimedia dispondrán de interfaz visual, de periféricos de interacción con el usuario como teclado, ratón, altavoces, etc.

Como es de suponer las pruebas realizadas de este sistema operativo también serán sobre el emulador puesto que es una versión de prueba y de avance a la imagen definitiva. Además recordemos que hasta ahora aún no hemos empleado el Board Support Package proporcionado para nuestra placa, lo que significa que nuestro sistema operativo será construido exclusivamente para el emulador como en el caso anterior.

#### **3.3.2. Elementos que componen la imagen**

Igual que en el caso anterior citare todos los elementos que componen esta imagen. Podemos diferenciar tres apartados: el primero corresponde a los drivers generales de nuestro sistema operativo, el segundo son todos los elementos que componen el sistema operativo que nosotros diseñamos y el último es el que proporciona los drivers específicos del dispositivo que vamos a utilizar.

Así bien, los elementos son los siguientes:

- Elementos generales del catálogo
  - Device Drivers -> Display -> Multiple-Display Support

- Device Drivers -> Serial -> Com16550
- Device Drivers -> Storage Devices -> Storage Devices
- Elementos específicos de Custom Device (Windows CE devices)
- Applications End User
  - CAB File Installer/Uninstaller
  - Terminal Emulator
- Applications and Services Development -> C Libraries and Runtimes
  - Full C Runtime
  - Standard I/O
  - Standard I/O ASCII
  - Standard String Features - ASCII
  - String Safe Utility Functions
- Applications and Services Development -> Component Services
  - Component Object Model (COM)
- Communication Services and Networking -> Networking - WAN
  - Telephony API
- Communication Services and Networking -> Networking Features
  - Networking Driver Architecture
  - TCP/IP
  - Winsock Support
- Communication Services and Networking -> Servers
  - Core Server Support
  - SNTP Client with DST
- Core OS Services -> Debugging Tools -> Toolhelp API
- Core OS Services -> Kernel Features
  - Fiber API
  - Format Message API
  - Memory Mapped Files
  - Message Queue – Point – to – Point
  - Target Control Support
- Core OS Services
  - Notification LED Support
  - PNP Notifications
  - Power Management (Full)
  - Serial Port Support
  - UI based Notification
- Device Management -> Device Management Client



- File Systems and Data Store
  - Bit-based
  - Compression
  - Database Support
  - Hive-based Registry
  - RAM and ROM File system
  - Storage Manager
  - System Password
- Fonts
  - Arial
  - Courier New
  - Tahoma
  - Times New Roman
- Graphics and Multimedia Technologies -> Audio
  - Waveform Audio
- Graphics and Multimedia Technologies -> Graphics
  - Alpha blend API (GDI version)
  - Gradient Fill Support
  - Multiple Monitor Support
- International
  - Input Method Manager (IMM)
  - Locale Specific Support
  - Multilingual user Interface (MUI)
  - National Language Support (NLS)
- Internet Client Services -> Internet Explorer 6.0 for Windows CE Components
  - Internet Explorer Multiple-Language Base API
  - URL Moniker Services
- Security
  - Authentication Services (SSPI) -> Schannel (SSL/TLS)
  - Credential Manger
  - Cryptography Services -> Certificates (CryptoAPI 2.0)
- Shell and User Interface -> Graphics, Windowing and Events
  - Minimal GDI Configuration
  - Minimal GWES Configuration
  - Minimal Input Configuration
  - Minimal Windows Manager Configuration
- Shell and User Interface -> Shell
  - AYGShell API Set
  - Command Shell -> Command Processor
  - Command Shell -> Console Windows
  - Standard Shell

- Shell and User Interface -> User Interface
  - Common Controls -> Animation Control
  - Common Controls -> Common Control
  - Common Dialog Support
  - Control Panel Applets
  - Customizable UI -> Windows XP-like Sample Skin
  - Mouse
- Elementos específicos de Emulator: x86 (Windows CE devices)
- Device Drivers
  - Audio -> Emulator Waveform Audio Device
  - Display -> Emulator Display
  - DMA trans
  - Input Devices -> Emulator Keyboard/Mouse English
  - Networking -> Local Area Networking (LAN) devices -> Emulator DC21x40 Ethernet Controller
  - Storage Devices -> Drive Mapping

### **3.4. Diseño imagen final: BSP Via Epia MII**

#### **3.4.1. Descripción**

Finalmente nos disponemos a crear la imagen del sistema operativo que verdaderamente se ejecutará bajo el dispositivo físico real. Por este motivo, esta es la parte del Capítulo de Diseño más importante debido que sujeto a su éxito se dará por conseguido el objetivo principal de este proyecto.

Este sistema operativo tendrá aparte de la interfaz gráfica que ya introducimos en el diseño anterior otras muchas funcionalidades que hará al sistema operativo compatible con la aplicación que correrá sobre este así como de los servicios que esta necesita como el protocolo TCP/IP o el puerto serie.

Además de todo esto también dispondrá de otras funciones que quise añadir al fin de obtener un sistema operativo más completo. Todos los visores de archivos están añadidos, como por ejemplo los documentos de Word, las imágenes, hojas de Excel, presentaciones de PowerPoint o documentos PDF.

Este sistema operativo al mismo tiempo de cumplir el objetivo principal del proyecto alcanza el más alto escalafón de la guía de aprendizaje y familiarización con este software de desarrollo. En este sentido he intentado introducir, aparte de los nuevos elementos imprescindibles para el desarrollo del proyecto común, brindar nuevos servicios como por ejemplo disponer de soporte de audio e Internet mediante los drivers del BSP de la placa, añadir la aplicación como parte de la imagen del sistema operativo, dar soporte a almacenamiento persistente masivo, etc. Todo ello con el fin de potenciar la vertiente pedagógica que reviste a mi proyecto en particular.

En resumen, este sistema operativo pretende ser una plataforma multimedia que interaccione con el usuario de varias formas ya sea por teclado, ratón, botonera por puerto serie o bien pantalla táctil ya que disponemos del soporte a pantalla de este tipo. Es decir, una amplia versatilidad en cuanto a la relación del usuario con el dispositivo. Además de lo anterior deberá dar soporte de codecs de audio y video más comunes como el formato MP3, WMA, AVI y Mpeg4.

En esta ocasión sí que diseñaremos este sistema para trabajar sobre el dispositivo final puesto que es el objetivo de este proyecto. Por esta razón las pruebas y la creación de imagen se verán alteradas en este sentido ya que el dispositivo objetivo ya no será el emulador.

Será difícil encontrar el sistema operativo perfecto puesto siempre pretendemos tener todas las funcionalidades posibles para cumplir así con creces las necesidades del consumidor final o incluso de nosotros mismos. Pero la verdad es que el tiempo limitado para este y mis propias limitaciones en un campo que acabo de empezar harán que adopte la solución más favorable para conseguir una plataforma que funcione y reúna las características más óptimas posibles.

Se debe añadir que para este último diseño hemos procedido a tres actualizaciones de la herramienta Platform Builder. Una es la respectiva a la actualización del Platform Builder que incluye nuevos servicios, aunque la experiencia que me ha dado no ha sido satisfactoria porque he visto que la tarea de compilación daba warnings que antes no daba. Aunque esto no me sorprende debido a que no es la primera vez que una actualización de Microsoft a más problemas que versiones más obsoletas. La segunda se refiere al paquete de actualización de Compact Framework para el uso de su versión 2.0 ya que Platform Builder tiene la versión 1.0. La última sería la correspondiente al Board Support Package (BSP) de la placa elegida.

### **3.4.2. Elementos que componen la imagen**

Así bien, los elementos son los siguientes:

- Elementos generales del catálogo
  - Device Drivers -> Audio -> VIA 3059 Audio (Unified)
  - Device Drivers -> Display -> Multiple-Display Support
  - Device Drivers -> Display -> VIA CLE 266/CN400/CN700
  - Device Drivers -> IEEE1394
  - Device Drivers -> Input Devices -> 8042 Keyboard/Mouse English

- Device Drivers -> Networking -> Local Area Networking (LAN) devices -> VIA Rhine Family Fast Ethernet
- Device Drivers -> Networking -> Personal Area Networking (PAN) devices -> Serial infrared (SIR)
- Device Drivers -> Networking -> Wide Area Networking (WAN) devices -> Asyncmac NDIS Driver
- Device Drivers -> PCI Bus
- Device Drivers -> Power Saver
- Device Drivers -> Printer Devices
- Device Drivers -> Serial -> Co16550
- Device Drivers -> Serial -> Com\_Card
- Device Drivers -> Storage Devices
- Device Drivers -> USB Function -> USB Function Clients -> Mass Storage
- Device Drivers -> USB Host -> EHCI
- Device Drivers -> USB Host -> UHCI
- Elementos específicos de mobile Handheld (Windows CE devices)
- Applications End User
  - ActiveSync
  - CAB File Installer/Uninstaller
  - File Viewers: Excel, Image, PDF, PowerPoint and Word viewer.
  - Help
  - Windows Messenger
  - WordPad
- Applications and Services Development
  - Active Template Library (ATL)
  - Lightweight Directory Access Protocol (LDAP) Client
  - Message Queuing (MSMQ)
  - Standard SDK for Windows CE
- Applications and Services Development -> .NET Compact Framework
  - OS Dependencies for .NET Compact Framework 1.0

- Applications and Services Development -> .NET Compact Framework 2.0
  - .NET Compact Framework 2.0
  - OS Dependencies for .NET Compact Framework 2.0
- Applications and Services Development -> Component Services (COM and DCOM) -> Component Object Model
  - COM
- Applications and Services Development -> .Object Exchange Protocol (OBEX)
  - OBEX Client
  - OBEX Server
- Applications and Services Development -> SOAP Toolkit
  - Client
- Applications and Services Development -> SQL Mobile
  - SQL Mobile
  - SQL Mobile .Net 2.0 Data Provider
  - SQL Server .Net 2.0 Data Provider
- Applications and Services Development -> XML -> MSXML 3.0
  - XML Core Services and Document Object Model (DOM)
- Applications and Services Development -> C Libraries and Runtimes
  - C++ runtime Support for Exception Handling and Runtime Type Information
  - Full C Runtime
  - Standard I/O
  - Standard I/O ASCII
  - Standard String Features - ASCII
  - String Safe Utility Functions
- Applications and Services Development -> Component Services
  - Component Object Model (COM)
- Communication Services and Networking -> Networking - LAN
  - Wired Local Area Network (802.3, 802.5)
  - Wireless LAN (802.11) STA – Automatic Config. & 802.1x
- Communication Services and Networking -> Networking - PAN
  - IrDA
  - Bluetooth HS/HF and Audio Gateway Service
  - Bluetooth Stack with Universal Loadable Driver
- Communication Services and Networking -> Networking - WAN
  - Telephony API
  - Dial Up Networking (RAS/PPP)
  - Virtual Private Networking -> PPTP

- Communication Services and Networking -> Networking Features
  - Extensible Authentication Protocol
  - NDIS User-mode I/O Driver
  - Networking Driver Architecture
  - TCP/IP
  - TCP/IPv6 Support
  - Windows Networking API/Redirector (SMB/CIFS)
  - Winsock Support
- Communication Services and Networking -> Servers
  - Core Server Support
  - Simple Network Time Protocol (SNTP) -> SNTP Automatic Updates and Server Synchronization
  - Simple Network Time Protocol (SNTP) -> SNTP Client with DST
  - Telnet Server
- Core OS Services
  - Battery Driver
  - Device Manager
  - Display Support
  - Internet Appliance (IABASE) Support
  - Notification LED Support
  - PNP Notification
  - Power Management (Full)
  - Serial Port Support
  - UI based Notification
  - USB Host Support
- Core OS Services -> Kernel Features
  - Fiber API
  - Format Message API
  - Memory Mapped Files
  - Message Queue – Point – to – Point
- Device Management -> Device Management Client
- File Systems and Data Store
  - Bit-based
  - Compression
  - Database Support
  - Hive-based Registry
  - RAM and ROM File system
  - Storage Manager
  - System Password
- Fonts
  - Courier New
  - Symbol
  - Tahoma

- Times New Roman
  - Wingding
- Graphics and Multimedia Technologies -> Audio
  - Waveform Audio
- Graphics and Multimedia Technologies -> Graphics
  - Direct3D Mobile
  - DirectDraw
  - Gradient Fill Support
  - Multiple Monitor Support
- Graphics and Multimedia Technologies -> Graphics -> Imaging
  - Still Image Codec Support (encode and decode)
  - BMP, JPG, GIF & PNG decoders
  - BMP, JPG, GIF & PNG encoders
- Graphics and Multimedia Technologies -> Media
  - Streaming Media Playback
  - WMA and MP3 Local Playback
  - WMA and MP3 Streaming
- Graphics and Multimedia Technologies -> Media -> Audio Codecs and Renderers
  - MP3 Codec
  - MPEG-1 Layer 1 and 2 audio Codec
  - MS ADPCM Audio Codec
  - Wave/AIFF/au/snd File Parser
  - WMA Codec
  - WMA Voice Codec
- Graphics and Multimedia Technologies -> Media -> DirectShow
  - ACM Wrapper Filter
  - DirectShow Core
  - DirectShow Display
  - DMO Wrapper Filter
- Graphics and Multimedia Technologies -> Media -> DVD - Video
- Graphics and Multimedia Technologies -> Media -> Media Formats
  - AVI Filter
  - MPEG-1 Parser/Splitter
- Graphics and Multimedia Technologies -> Media -> Video Codecs and Renderers
  - DirectShow Video Renderer
  - MPEG-1 Video Codec
  - MS RLE Video Codec
  - Video/Image Compression Manager

- WMV/MPEG-4 Video Codec
- Graphics and Multimedia Technologies -> Media -> Windows Media Player
  - Windows Media Player
  - Windows Media Player OCX
  - Windows Media Technologies
- International
  - Input Method Manager (IMM)
  - National Language Support (NLS)
- Internet Client Services -> Browser Application
  - Internet Explorer 6.0 for Windows CE – Standard Components
- Internet Client Services -> Internet Explorer 6.0 for Windows CE Components
  - Internet Explorer Browser Control Host
  - Internet Explorer HTML/DHTML API
  - Internet Explorer Multiple-Language Base API
  - URL Moniker Services
  - Windows Internet Services
  - XML MIME Viewer
- Internet Client Services -> Scripting
  - Jscript 5.6
  - VBScript 5.6
- Internet Client Services -> Internet Options Control Panel
- Security
  - Authentication Services (SSPI) -> Schannel (SSL/TLS)
  - Credential Manger
  - Cryptography Services -> Certificates (CryptoAPI 2.0)
- Shell and User Interface -> Graphics, Windowing and Events
  - Minimal GDI Configuration
  - Minimal GWES Configuration
  - Minimal Input Configuration
  - Minimal Windows Manager Configuration
- Shell and User Interface -> Shell
  - AYGShell API Set
  - Command Shell -> Command Processor
  - Command Shell -> Console Windows
  - Standard Shell
- Shell and User Interface -> User Interface
  - Common Controls -> Animation Control
  - Common Controls -> Common Control
  - Common Dialog Support



- Control Panel Applets
  - Customizable UI -> Windows XP-like Sample Skin
  - Mouse
  - Network user interface
  - Overlapping Menus
  - Software Input Panel -> SIP for Large Screens
  - Software Input Panel -> SIP for Small Screens
  - Software Input Panel -> Software-based Input Panel Driver
  - Touch Screen (Stylus)
- Voice over IP Phone Services -> Real-time Communications (RTC) Client API
- Elementos específicos de VIA CLE266/CN400/CN700 CEPC: x86
- HDD Registry Save



## CAPÍTULO 4. PRUEBAS

### 4.1. Pruebas primera imagen: Tiny Kernel

En esta versión del sistema operativo basado únicamente en el núcleo de Windows CE procederemos a realizar pocas pruebas de interés ya que no tenemos interfaz gráfica y tampoco el sistema corre bajo el dispositivo final.

Como podremos ver en cada una de las versiones del sistema operativo creado las pruebas se remiten principalmente a realizar un test de conectividad con el dispositivo objetivo. En nuestro caso la imagen basada en Tiny Kernel correrá bajo el emulador por lo que será nuestro dispositivo objetivo.

Si consultamos el tutorial de diseño de la plataforma que figura en el apéndice, nos quedamos enlazando con nuestro dispositivo objetivo. Este sería el paso de debemos retomar para realizar nuestro testado.

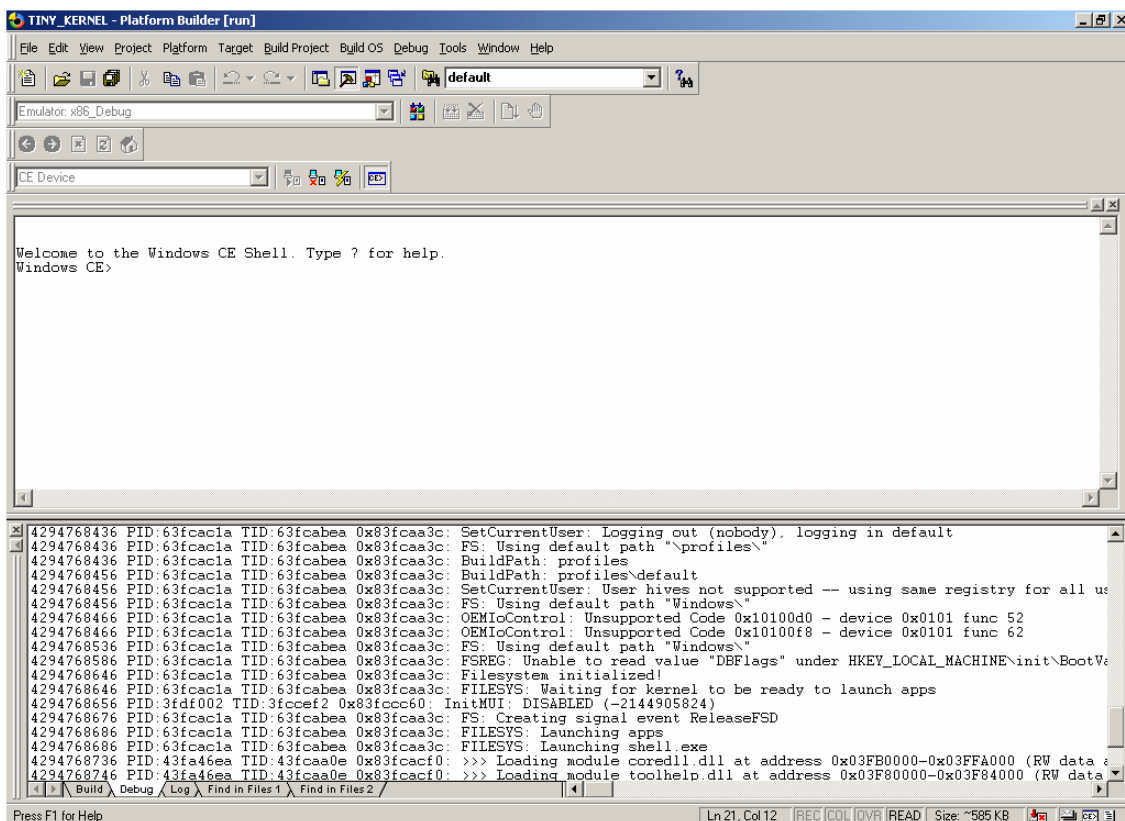
Si abrimos la herramienta Platform Builder, podemos acceder al menú superior "File" -> "Recent Workspaces". En este menú vemos los últimos proyectos con los que hemos trabajado y si hacemos clic sobre uno de ellos se abre automáticamente. En caso de que hubiéramos trabajado con otros proyectos y no figurara en este menú el proyecto que deseamos abrir siempre podremos abrirlo accediendo a "File" -> "Open Workspace" y encontrando la ruta del proyecto que se pretende abrir.

Después de aclarar todo lo anterior procedemos a abrir nuestro primer diseño, el sistema operativo basado en Tiny Kernel.

En cuanto abramos este proyecto nos aseguramos que en el menú superior "Target" -> "Connectivity Options" figura el dispositivo emulador tanto en el apartado de "Download" como el de "Transport". A partir de aquí ya podemos probar la imagen.

Para probar la imagen solamente debemos hacer clic en el menú superior "Target" -> "Attach Device" con lo que provocamos en enlace entre el sistema creado y listo para probar con el dispositivo objetivo antes elegido en la pantalla de "Connectivity Options".

En cuanto enlacemos con el dispositivo objetivo podemos advertir dos cosas interesantes. La primera es que a pesar de que se abrió la ventana del emulador de Windows CE este aparece en negro. Esto se debe a que estamos probando una versión del sistema sin interfaz gráfica por lo que no vemos nada. La segunda es que en la pantalla principal del Platform Builder nos figurará una ventana en la que podemos ver una interfaz de comandos. Este es el único shell disponible de nuestro dispositivo tiny kernel, en él podemos ejecutar comandos más o menos interesantes.



**Fig. 1.4** Terminal de Windows CE (Shell)

Una vez nos encontramos con el terminal de Windows CE podemos plantear algún tipo de ejemplo con el objetivo de comprobar el correcto funcionamiento de este sistema sin interfaz gráfica.

Para conocer los comandos disponibles en el Terminal de Windows CE debemos ejecutar el siguiente comando:

Windows CE>?

Introduciendo el carácter “?” y pulsando Intro veremos todos los comandos disponibles en este Terminal.

La primera apreciación que podemos distinguir es que los comandos se clasifican en tres tipos:

- GI (Get Information) -> Extraer información
- MI (Memory Information) -> Información de memoria
- ZO (Zone Ops) -> Zona de operaciones

Según esta clasificación podemos construir los comandos que deseamos ejecutar. En el primer ejemplo propuesto ejecutamos el siguiente comando:

Windows CE>gi thrd

Como podemos ver, el comando se compone de dos partes, la primera que distingue qué tipo de comando se va a ejecutar y la segunda que es la propia acción que deseamos realizar. En este caso, utilizamos un comando para extraer información general del sistema como son los threads o hilos de ejecución.

La salida obtenida será algo del estilo de la siguiente figura, donde podemos visualizar cada una de las aplicaciones que en ese momento están ejecutándose en el dispositivo objetivo así como los hilos de ejecución que dependen de éstas. Pese que no es una información demasiado relevante para un usuario común, puede ser bastante útil para un desarrollador que pretende tener consciencia de las aplicaciones y threads que corren bajo su sistema.

```

PROC: Name      hProcess: CurAKY :dwWMBase:CurZone
THRDR: State :hCurThrd:hCurProc: CurAKY :Cp :Bp :Kernel Time  User Time
P00: NK.EXE     03fdf002 00000001 c2000000 00000100
T   Blockd 83fcfb2e 03fdf002 00000001 1 1 00:00:00.000 00:00:00.000
T   Blockd 83fd09ae 03fdf002 00000001 130 130 00:00:00.170 00:00:00.000
T   Sleepg 23fdbdce 03fdf002 00000001 247 247 00:00:00.010 00:00:00.000
T   Sl/Blk 03fdf1de 03fdf002 ffffffff 255 255 00:00:00.500 00:00:00.000
P01: filesys.exe c3fcac1a 00000002 04000000 00000020
T   Blockd 23faab1a c3fcac1a 00000003 251 251 00:00:00.000 00:00:00.010
T   Blockd c3fcabea c3fcac1a 00000003 251 251 00:00:00.000 00:00:00.720
P02: shell.exe 63fa4a52 00000004 06000000 00000000
T   Runing 63fcaa0e 63fa4a52 00000005 130 130 00:00:00.000 00:00:00.360

```

**Fig. 2.4** Salida al comando “gi thrd”

En el segundo ejemplo introducimos el comando siguiente:

```
Windows CE>mi kernel
```

Vemos que ahora el comando es del tipo “Memory Information”. Con este comando ejecutamos la aplicación conocida como “Windows CE Kernel Memory Usage Tool 0.2”. Esta aplicación lanzada por el comando citado muestra la ocupación de memoria actual del sistema.

```

Windows CE Kernel Memory Usage Tool 0.2
Page size=4096, 7635 total pages, 7558 free pages, 7556 MinFree pages (323584 MaxUsed bytes)
20 pages used by kernel, 0 pages held by kernel, 77 pages consumed.
Inx Size Used Max Extra Entries Name
0: 428 2996 4280 1284 7( 10) Thrd
1: 228 1824 2052 228 8( 9) Mod
2: 48 4464 4656 192 93( 97) API/CStk/ClnEvt/StbEvt/Prxy/HData/KMod
3: 168 14280 16296 2016 85( 97) Crit/Evt/Sem/Mut/ThrdDbg
4: 64 512 512 0 8( 8) FullRef/FSMap/ThrdTm
5: 16 8208 8224 16 513(514) MemBlock
6: 524 0 1572 1572 0( 3) Name
7: 2048 12288 14336 2048 6( 7) HlprStk
Total Used = 44572 Total Extra = 7356 Waste = 56

```

**Fig. 3.4** Salida al comando “mi kernel”

Podemos comprobar que tenemos acceso a todo tipo de información del estado de la memoria como por ejemplo el número de entradas de threads en el sistema y la ocupación de memoria que hay en el sistema.

Pongamos por caso que el programador de una aplicación que utiliza mucho las API de Windows. Podríamos consultar cuanta memoria se ocupa debido al lance de esta aplicación en el sistema.

Por último me gustaría destacar que podemos probar los demás comandos a modo de prueba y ensayo del sistema, aunque más bien es para un objetivo de depurador de aplicaciones o del propio sistema.

Recordemos que para finalizar la simulación debemos hacer clic en el menú superior "Target"->"Detach Device". Esta acción desenlazará el dispositivo objetivo de nuestra estación de desarrollo y, en este caso, finaliza el enlace con el emulador.

## **4.2. Pruebas primera imagen: Tiny Kernel con interfaz gráfica**

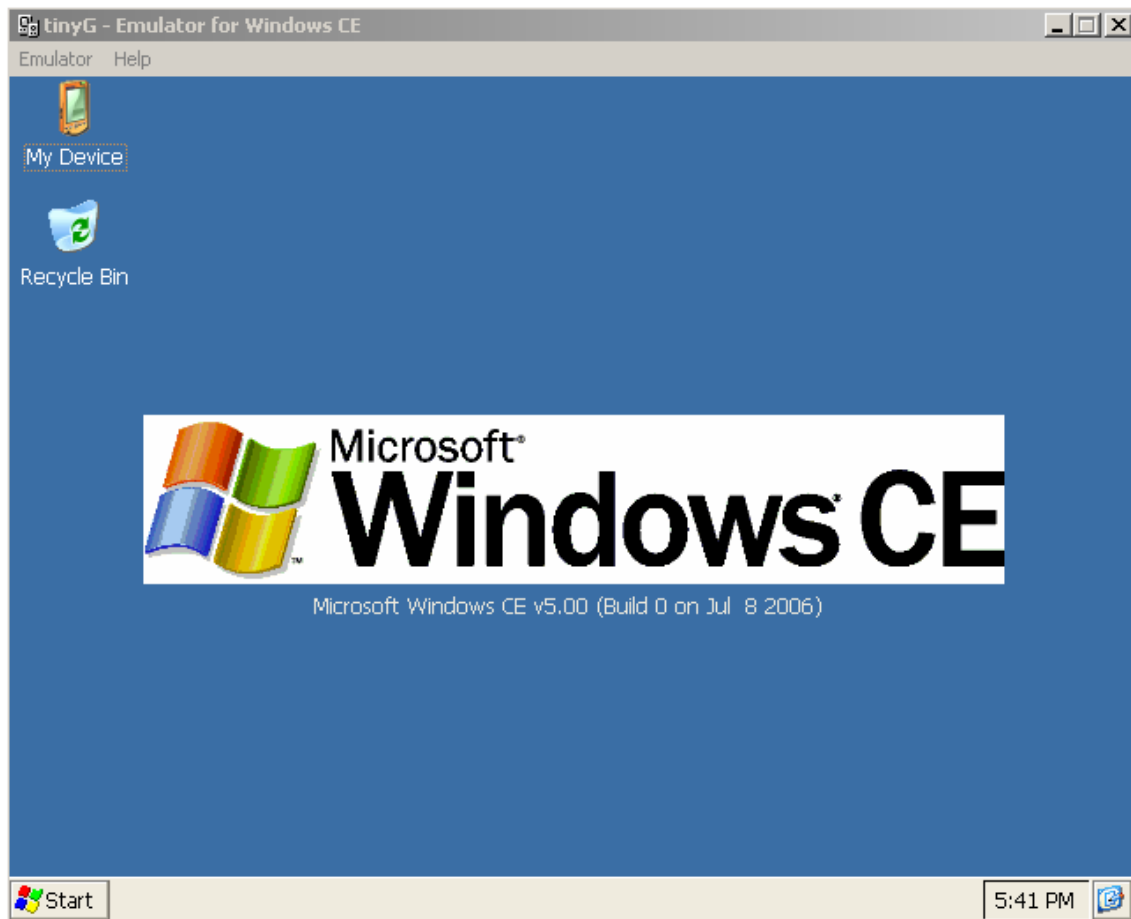
En esta segunda imagen nos limitaremos a probarla sobre el emulador. Es decir, sabemos que esta imagen es como la anterior aunque un poco más completa y con interfaz gráfica. Podemos consultar los módulos añadidos en la segunda imagen para recordarlos.

Pues bien, la prueba de esta imagen tan sólo consiste en arrancar el emulador corriendo la imagen visualizando el interfaz de Windows CE. Por supuesto, posteriormente podemos hacer pruebas como arrancar alguna aplicación añadida, etc.

Lo primero será situarnos en el mismo punto que figura en los pasos descritos de diseño en el Apéndice de este proyecto.

Abrimos el proyecto de la imagen deseada. Después comprobamos el dispositivo objetivo como anteriormente y nos aseguramos que esté seleccionado el emulador.

Por último enlazamos nuestra estación de trabajo con el dispositivo objetivo y deberá lanzarse el emulador, después de cargar la imagen podemos visualizar nuestro sistema operativo en la ventana del emulador.



**Fig. 4.4** Tiny Kernel con interfaz gráfica

En cuanto sea cargada la imagen podemos cerciorarnos de que el look & feel es de un Windows de PC doméstico común. Esto motiva la facilidad de interacción que tendremos con el sistema Windows CE.

Acostumbrados a la plataforma Windows nos resultará fácil movernos por él y se propone como prueba del sistema esto mismo, interactuar con él y probar los menús y opciones que nos permite.

Como prueba alternativa podemos arrancar algún servicio disponible como por ejemplo la terminal de comandos o Command Prompt que equivale a la existente en Windows. De la misma forma que en el caso de Tiny Kernel, disponemos de un terminal de depuración en la estación de trabajo donde tengamos Platform Builder. Podríamos realizar algunas pruebas como ejecutar el explorador de Windows y certificar que se ejecuta un nuevo thread, o bien saber qué carga en memoria ocupa, etc.

No obstante, debemos recordar que este sistema tiene limitaciones porque es un sistema muy básico basado en tiny Kernel pero con interfaz gráfica. Aunque posee algún servicio más que el diseño predecesor es un sistema muy elemental. Podemos recordar los elementos que componen la imagen en la etapa de diseño.

### 4.3. Pruebas imagen finales: BSP Via EPIA MII

En este apartado veremos como cargamos la imagen en el dispositivo, una vez cargada en el dispositivo la arrancaremos. Con este proceso llevaremos a cabo la propia prueba del sistema.

Como podemos advertir la imagen de nuestro sistema ya no se ejecutará en el emulador como dispositivo objetivo sino que directamente en el dispositivo final. Esto se debe a que se trata ya de la imagen preparada mediante el BSP de VIA para ejecutarse en la placa EPIA MII.

Nuestro dispositivo dispone de un disco duro para poder almacenar la imagen del sistema operativo. El primer paso entonces sería formatear este disco duro con el formato FAT, compatible con WCE, mediante una herramienta como el comando "fdisk" de MS-DOS.

Una vez preparado el formato debemos cargar un sistema de arranque para que el dispositivo sea capaz de arrancar la imagen. Este sistema de arranque será capaz de cargar la imagen del sistema de forma local en nuestro disco duro, de forma remota vía Ethernet o puerto serie por si deseamos enlazar con nuestra estación de desarrollo.

Nosotros arrancaremos normalmente de forma local, esto quiere decir que existe una imagen de sistema volcada en el disco duro del dispositivo. Si quisiéramos hacer alguna prueba del sistema mientras lo desarrollamos podríamos cargar el sistema de forma remota mediante Ethernet (más rápido).

El proceso para instalar el sistema de arranque en el dispositivo es el siguiente:

- Formateamos un disquete de 3½
- Dejando insertado el disquete en nuestra unidad, vamos a Inicio -> Ejecutar, escribimos cmd y pulsamos Intro.
- Nos situamos en el directorio "%ProgramFiles%\Windows CE Platform Builder\5.00\CEPB\Utilities" de donde hayamos instalado el paquete PB.
- Ejecutamos el comando "CEPCBoot.144"
- Inmediatamente comenzará a escribir en el disquete.
- Una vez finalizado debemos copiar el contenido del disquete a la partición del disco duro del dispositivo.

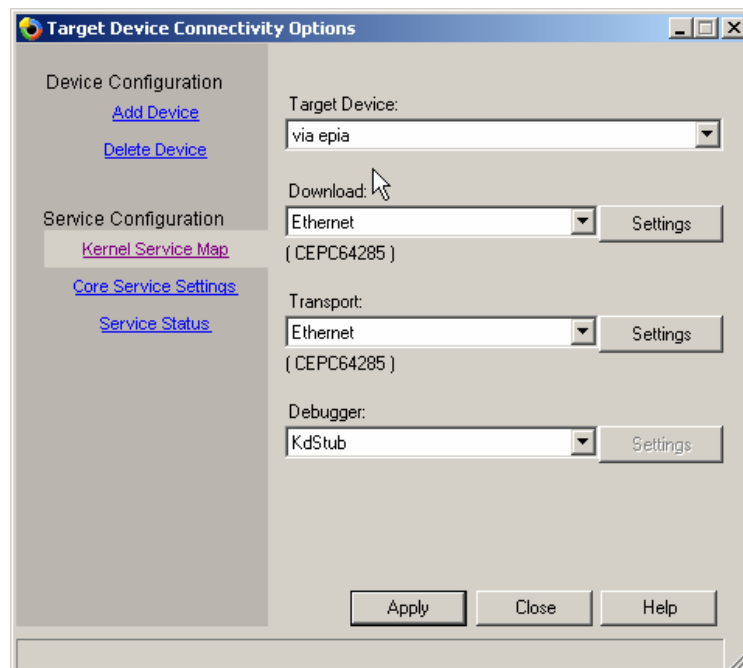
Una vez finalizado este proceso sólo tendremos que volcar la imagen a la partición del disco duro que contiene los archivos anteriormente copiados.

La imagen de nuestro sistema fue generada y según la especificación de diseño del capítulo del apéndice se encuentra, en mi caso, en "J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\ReIDir\VIACN\_x86\_Release". La imagen corresponde al fichero "nk.bin".

Si deseamos cargar alguna imagen vía Ethernet sólo deberíamos escoger la opción 4 llamada "ether via eboot" en el menú de arranque del dispositivo y desde la estación de desarrollo configurar el dispositivo objetivo. Para esto,



accedemos a “Target” -> “Connectivity Options” y debemos añadir un nuevo dispositivo “Add device”. El nombre del dispositivo no es relevante pero sí la configuración que podemos ver a continuación:



**Fig. 5.4** Captura de Platform Builder. Opciones de conectividad

Haciendo esto, cuando llevemos a cabo el enlace entre el dispositivo y la estación de desarrollo haciendo clic en “Target” -> “Attach Device” se descargará la imagen vía Ethernet.

Pero si volvemos al caso que nos ocupa, nuestra imagen será cargada de forma local porque copiamos el archivo de imagen de sistema a la partición donde se encuentran los archivos de arranque que antes mencionamos.

Para arrancar de forma local sólo debemos elegir la opción adecuada del menú de arranque, en este caso la opción 1 llamada “local nk.bin”.

Automáticamente arrancará nuestro sistema tal y como muestra la siguiente figura:



**Fig. 6.4** Captura del dispositivo. Sistema operativo final

Podríamos ejecutar algún programa añadido a modo de prueba, pero este proceso ya deja de tener importancia para ser incluido en el proyecto.

Además después de la valoración general del sistema vemos que es estable en todas sus aplicaciones y servicios, por ejemplo conexión a Internet, Messenger, etc.

## CAPÍTULO 5. INTEGRACIÓN

### 5.1. Adición de la aplicación de .NET sobre WCE

Para probar la aplicación sobre el dispositivo se debe añadir los ficheros a la imagen del sistema operativo. Con ello conseguiremos que la aplicación forme parte del sistema operativo y se encuentre disponible en la ruta que deseemos.

Para llevar a cabo esto tenemos que hacer uso de una aplicación externa a Platform Builder llamada "CE File Wizard". Esta aplicación la podemos encontrar en la web de Mike Hall desarrollador de Microsoft y máximo exponente en lo que a WCE se refiere. Una vez descargada esta aplicación, la ejecutamos y podemos desde esta añadir los archivos que deseemos en la ruta que elijamos y para la plataforma deseada.

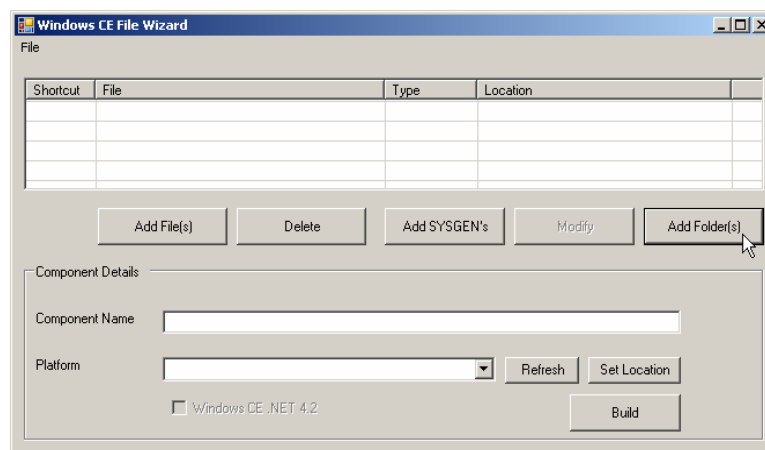


Fig. 6.4 Captura de CE File Wizard. Pantalla principal

Ahora procedemos a añadir los ficheros de la aplicación deseados:

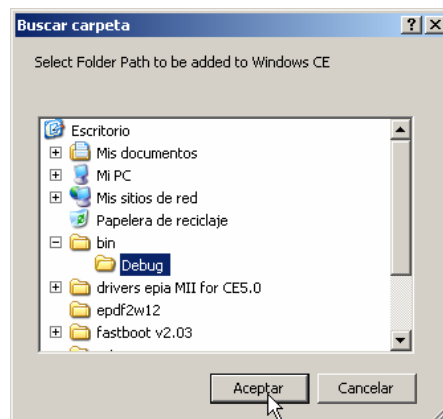
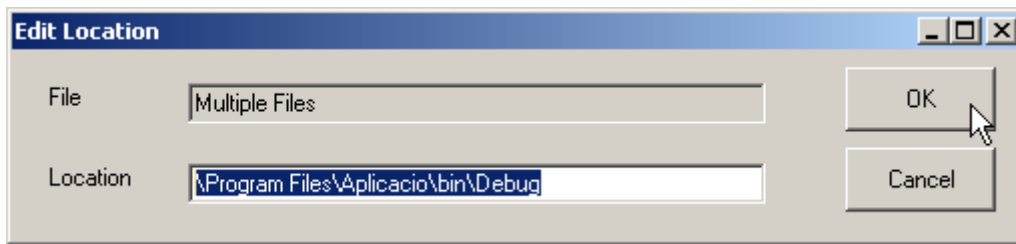


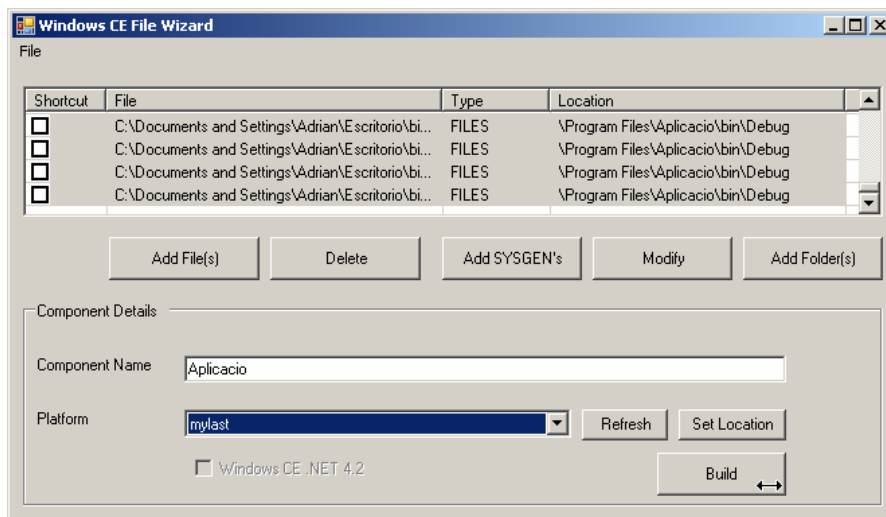
Fig. 7.4 Captura de CE File Wizard. Añadiendo una carpeta

Cambiamos la ruta donde se alijarán en el dispositivo seleccionando los archivos y haciendo clic en “Modify”:



**Fig. 8.4** Captura de CE File Wizard. Cambiamos la ruta

Y por último damos un nombre y elegimos para qué plataforma se añadirán los archivos, después presionamos “Build”.



**Fig. 9.4** Captura de CE File Wizard. Cambiamos la ruta

Después de este proceso con la aplicación “CE File Wizard” podemos arrancar Platform Builder y abrimos nuestro workspace de la imagen final.

Posteriormente, accedemos al menú “File” -> “Manage Catalog Items” donde podemos añadir elementos al catálogo. Inmediatamente se nos abrirá una ventana donde podemos importar el catálogo creado en la ruta del actual diseño con el nombre que hayamos especificado, en este caso “Aplicacio”.

Una vez importado se añadirá a la lista de la venta “Manage Catalog Items” y pulsamos el botón “OK”.

Con ello provocamos que se actualice la ventana del catálogo de Platform Builder y en la carpeta de “Third Party” se hallará un componente llamado “Aplicacio Files Component” que corresponde a los archivos añadidos. Solamente deberemos hacer clic con botón derecho sobre el elemento y añadirlo a nuestro sistema.

Por último tenemos que generar el sistema haciendo clic en “Build”->”SysGen” y crear la imagen del sistema haciendo clic en “Build”->”Make Run-time Image”.

Cuando acabemos este proceso ya podemos volver a copiar el archivo de imagen del sistema al disco duro del dispositivo igual que en capítulo anterior.

En la ruta que elegimos debe encontrarse la aplicación.

## 5.2. Funcionamiento de la aplicación sobre WCE

Durante el proceso de integración de la aplicación con el sistema operativo hemos encontrado los siguientes problemas:

- Cambio del nombre de las librerías por “GAC+[ nombre de la librería ] + versión”
- Problemas con la lectura XML
- Reproducción de vídeo con ausencia de vídeo pero no de audio.

El primer problema ha sido solventado renombrando las librerías con el nombre que pide la aplicación. Aunque posteriormente esperamos encontrar algún sistema para hacerlo más cómodamente.

El segundo de los problemas era debido a que el sistema no tiene disponible la librería System.Xml.dll y por tanto tiene que ser agregada como una librería más.

Desafortunadamente, aún no conseguimos reproducir vídeo. Creemos que puede deberse a que la reproducción de vídeo sólo soporta vídeos con unas características particulares en relación a resolución, frames, etc. Aunque hemos conseguido escuchar el audio del vídeo intentaremos solucionar este problema para poder hacer la demostración con esta funcionalidad.

No obstante, la valoración de la aplicación su funcionamiento es el adecuado y el exigido pese a los anteriores problemas antes mencionados.

En el apéndice podemos ver alguna captura de los errores que produjo la aplicación ejecutándose sobre el emulador.



## CAPÍTULO 6. ELECCIÓN DE PLACA

### 6.1. Elección de la placa

#### 6.1.1. Estudio previo de mercado. Visión general.

El estudio inicial del mercado estuvo enfocado principalmente sobre tres plataformas: VIA, Intel y PC-104.

Optamos por hacer un estudio sobre estos fabricantes puesto que son las placas dominantes del mercado además de poseer un gran abanico de modelos de placas con el objetivo de satisfacer en mayor grado las necesidades específicas de un sistema empotrado.

Al resultar este un proyecto universitario debemos reducir los costes en la medida de lo posible para poder adquirir todos los elementos y poder realizar el montaje físico de nuestro dispositivo y, por lo tanto, concluir el objetivo de este proyecto en común.

Como no es de extrañar la primera idea que surgió es acudir al fabricante más prestigioso, Intel. Sabemos que Intel se caracteriza por un producto de calidad pero normalmente sus elevados precios hacen que no estén al alcance de todos los bolsillos. Esta es una de las principales desventajas que encontramos en las placas Intel para declinar la opción de desarrollar nuestro dispositivo sobre esta plataforma. No obstante, debemos afirmar que las placas Micro – ITX que oferta Intel tienen una gran potencia ya que soportan microprocesadores de ordenador de sobremesas, hasta 3.06 GHz. Por este motivo están muy bien consideradas a pesar de que pierden algo de versatilidad a favor de su competencia.

A ello se une que el microprocesador para placas Intel Micro – ITX debe adquirirse por separado, es decir, no viene integrado en la placa por lo que encarece más su adquisición.

Nuestra segunda opción contemplada era la plataforma conocida como PC-104. Es una plataforma que está muy extendida y consolidada. Esta plataforma se caracteriza por una gran fiabilidad de sus productos pero está más enfocada a desarrollo de equipos empotrados industriales.

Aunque ya existen varios modelos multimedia aún no tienen las características y la potencia ofrecida por, por ejemplo, Intel o VIA.

Además de esto, muchos de los modelos sólo soportan Linux como sistema operativo por lo que estos modelos nunca podrán ser una propuesta viable ya que el sistema operativo diseñado es Windows.

Por último, estas placas son demasiado moduladas, con esto quiero decir que muchos de las necesidades extras que podamos desear implican la adquisición de algún módulo adicional con el fin de satisfacerla. Por estos motivos entre otro, podemos afirmar que tampoco es una plataforma que nos acabara de satisfacer para desarrollar nuestro proyecto.

En último lugar, el fabricante VIA ha sido el que más interés nos suscitó por su amplio repertorio de modelos que satisfacen todas las necesidades incluso de un PC de sobremesa.

La primera ventaja a destacar es la versatilidad de sus productos. Esta versatilidad se muestra en varios aspectos. Poseen numerosos modelos incluso de formato Nano – ITX, estos modelos suministran diferentes opciones a nuestra placa desde tarjeta de sonido integrada con salida y entrada de audio como puertos USB, serie, lector PCMCIA, Compact Flash, etc.

Otro aspecto muy destacable es la versatilidad que ofertan sus placas con el procesador integrado y con la opción además de optar por una refrigeración pasiva o activa. En pocas palabras, podemos optar por un modelo cuyo procesador sea menos potente pero refrigerado por disipador por lo que se convierte en un dispositivo muy silencioso.

El precio final de venta al público hizo de estas placas la opción más interesante. Mientras los modelos más caros del fabricante VIA no sobrepasan en el peor de los casos los 400 €, Intel posee algún modelo por más de 500 €.

En resumen, el fabricante decidido por los integrantes de este proyecto fue VIA, por lo que a continuación se presentaría una propuesta de los tres modelos de este fabricante que cubren nuestras necesidades.

### **6.1.2. Propuesta de comparativa presentada**

Según lo descrito en el apartado anterior, hemos dispuesto hacer tres propuestas diferentes entre modelos del fabricante VIA. Los tres modelos que parecen más interesantes y cumplen una buena relación calidad – precio son los que veremos a continuación. Además reúnen unas características similares y nos parece que reúnen las características necesarias.

#### **PROPUESTA 1: VIA EPIA SP**

Integran la nueva generación de chipset CN400/VT8237. Con funciones multimedia. Más rápidas y con soporte discos SATA.

Procesador: EDEN (SP8000) sin ventilador.

Memoria: Un módulo de memoria DDR400 hasta 1Gb.

Onboard: 1 slot PCI. 2 ATA 133. 2 SATA. 1 Lan 10/100. Audio. VGA Unichrome con aceleradora MPEG-2/MPEG-4.



Conexiones traseras: 1 Lan 10/100 - 2 USB 2.0 - PS2 ratón - PS2 teclado - 1 Serie - 1 VGA - 1 RCA (sonido SPDIF o TV out) - 1 S-Video - Audio (line-out, line-in, mic-in que se pueden configurar como salida de 6 canales).

Conectores en placa para salidas adicionales: 6 USB adicionales - 1 FireWire 1394 - Audio Frontal- Segundo puerto serie - Puerto Paralelo - FIR (InfraRojos) - VIP (Video Interface).

### Back Panel EPIA-SP

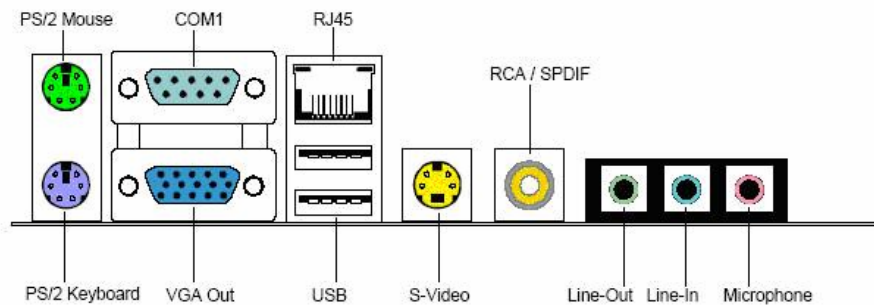


Fig. 1.6 Panel trasero EPIA SP

### MAINBOARD LAYOUT

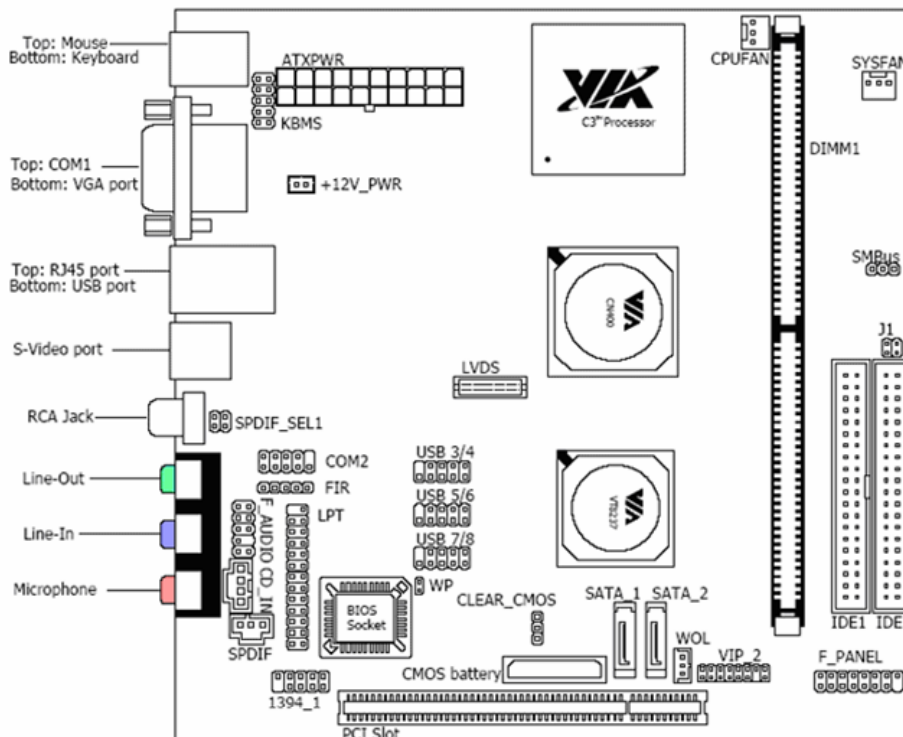


Fig. 2.6 Foto y layout de EPIA SP

## **PROPUESTA 2: VIA EPIA N (Nano-ITX)**

Integran la nueva generación de chipset CN400/VT8237R. Con funciones multimedia. Rápidas y con soporte discos SATA.

Procesador: LUKE CoreFusion (N8000E) sin ventilador.

Memoria: Un módulo de memoria SODIMM DDR400 (empleado por los portátiles) hasta 1Gb.

Onboard: 1 slot Mini-PCI. 2 ATA 133. 1 SATA. 1 Lan 10/100. Audio. VGA Unichrome PRO con aceleradora MPEG-2/MPEG-4.

Conexiones traseras: 1 Lan 10/100 - 2 USB 2.0 - 1 VGA - 1 RCA (sonido SPDIF o TV out) - 1 S-Video - Audio (line-out, line-in, mic-in que se pueden configurar como salida de 6 canales).

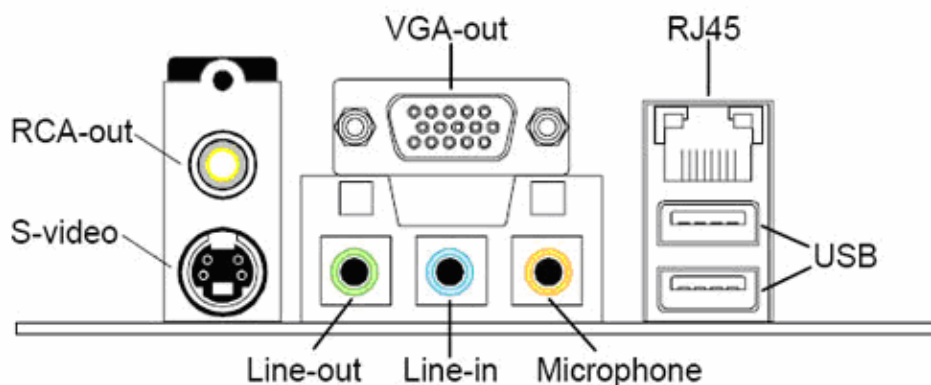
Conectores en placa para salidas adicionales: 4 USB adicionales - LVDS/DVI (requiere tarjetita adicional) - VIP (Video Interface) - PS2 para teclado - YPbPr (salida TV por componentes) - conector para dispositivos LPC/SIR.

Consumo: En idle: 11,087W (N8000E) / 12,278W (N10000). En full: 14,085W (N8000E) / 15,677W (N10000).

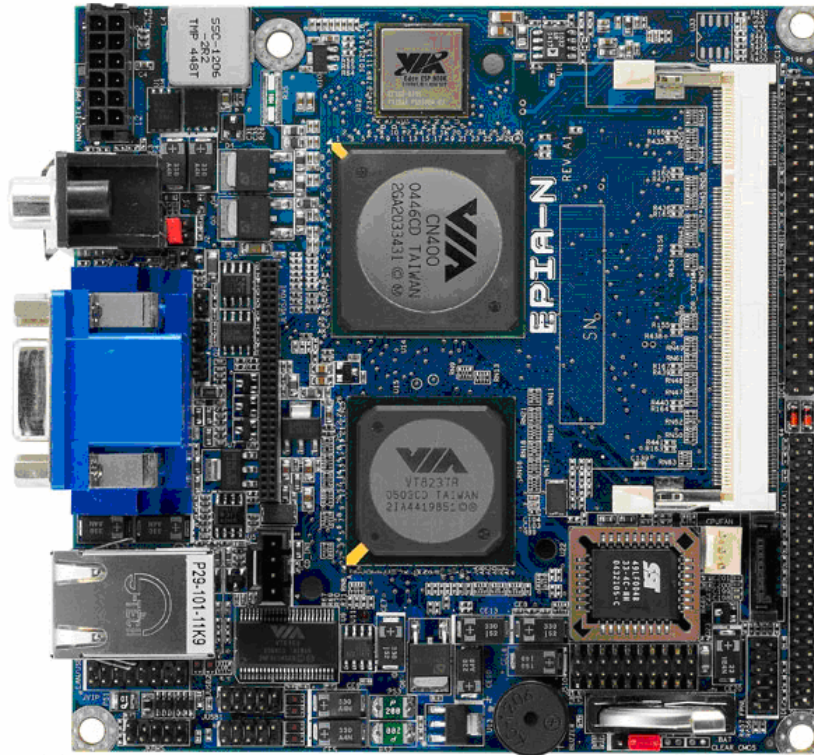
ATENCIÓN: La conexión a la alimentación no es ATX-Estándar. Requiere un adaptador.

ATENCIÓN II: No tiene conexiones PS2 a ratón ni teclado externo. Solo cuenta con conexiones USB. Requiere adaptador.

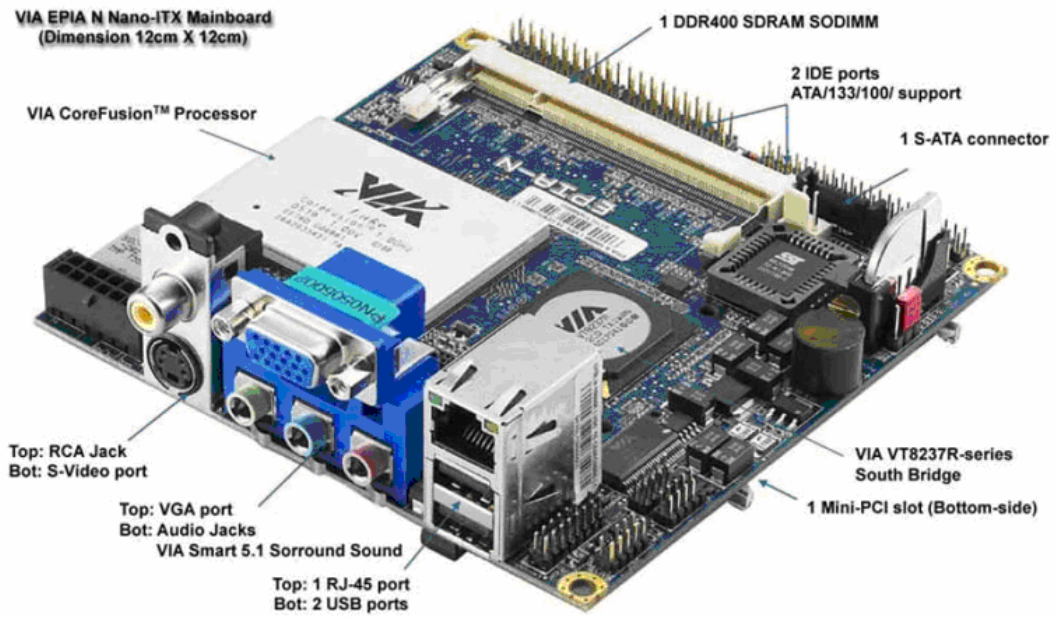
## **VIA EPIA N-Series I/O Back Panel Layout**



**Fig. 3.6** Panel trasero EPIA N-Series



**VIA EPIA N-Series Layout**



**Fig. 4.6 Foto y layout EPIA N-Series**

### **PROPUESTA 3: VIA EPIA MII**

Con funciones multimedia. Añade puerto PCMCIA y lector Compact Flash. Increíble flexibilidad.

Procesador: EDEN (MII6000E) sin ventilador.

Memoria: Un módulo de memoria DDR266 hasta 1Gb.

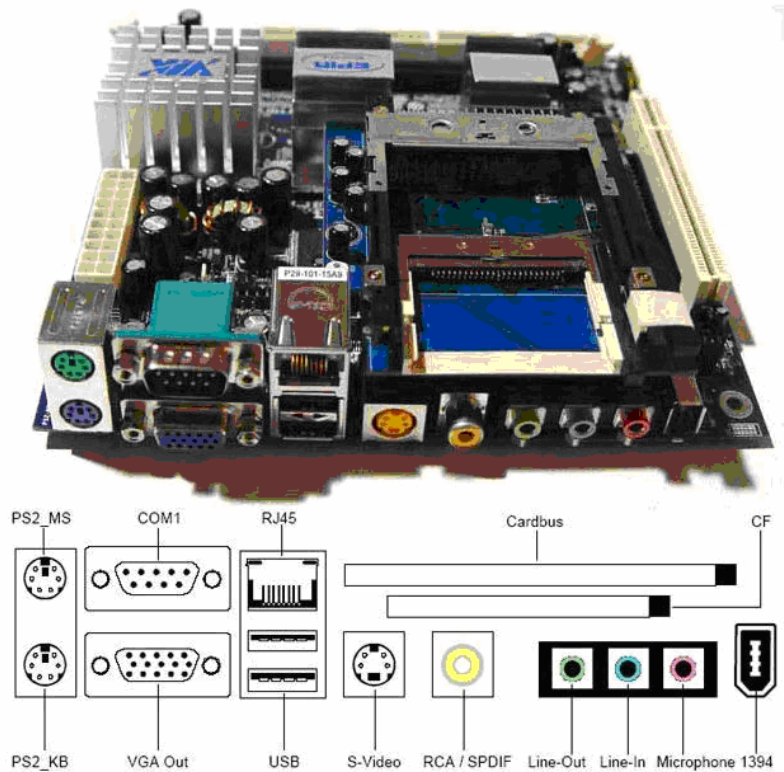
Onboard: 1 slot PCI. 2xATA 133. 1 Floppy. 1 Lan 10/100. Audio. VGA Unichrome con aceleradora MPEG-2.

Conexiones traseras: 1 Lan 10/100 - 2 USB 2.0 - PS2 ratón - PS2 teclado - 1 Serie - 1 VGA - 1 RCA (sonido SPDIF o TV out) - 1 S-Video - Audio (line-out, line-in, mic-in Via Vinyl Six-Track) - 1 FireWire 1394 - 1 CardBus Tipo I y Tipo II - 1 Compact Flash.

Conectores en placa para salidas adicionales: 2 USB mas - 1 Paralelo - Audio Frontal- Segundo puerto serie.

Consumo: En idle: 11,847W (MII6000E) / 15,655W (MII10000E). En full: 16,011W (MII6000E) / 25,708W (MII10000E).

**VIA EPIA MII-Series I/O Back Panel Layout**



**Fig. 5.6** Foto y panel trasero EPIA N-Series

### 6.1.3. Elección final

Después de haber visto las diferentes propuestas presentadas, podemos ver una tabla comparativa donde encontramos un presupuesto aproximado de ellas.

**Tabla 1.6** Presupuesto EPIA SP, EPIA N y EPIA MII

Concepto	Descripción	Precio
<b>PROPUESTA 1</b>		
<b>Placa madre</b>	EPIA SP	225
<b>Memoria</b>	Kingston KVR400X64C3A/256 256MB 400MHz DDR Non-ECC CL3 (3-3-3) DIMM	25,52
<b>Alimentación</b>	Fuente de alimentación Mini-ITX de 120W	95
	<b>TOTAL</b>	345,5 €
<b>PROPUESTA 2</b>		
<b>Placa madre</b>	EPIA N	348,8
<b>Memoria</b>	Kingston KVR266X64SC25/256 256MB 266MHz DDR Non-ECC CL2.5 SODIMM	29
<b>Alimentación</b>	Fuente de alimentación Mini-ITX de 120W	95
	<b>TOTAL</b>	472,8 €
<b>PROPUESTA 3</b>		
<b>Placa madre</b>	EPIA MII	188,8
<b>Memoria</b>	Kingston KVR266X64C25/256 256MB 266MHz DDR Non-ECC CL2.5 DIMM	24,36
<b>Alimentación</b>	Fuente de alimentación Mini-ITX de 120W	95
	<b>TOTAL</b>	308,1 €

Como podemos ver la opción más económica es el modelo EPIA MII. Este es un factor determinante ya que queremos reducir los costes al máximo. Además a esto se une que una vez expuestas las tres propuestas ante la junta llegamos a la decisión que este modelo era el más adecuado.

El primer motivo, como digo, el precio que es algo más reducido sobretodo comparándolo al modelo Nano (12x12 cm) mientras que el modelo EPIA MII es Micro (17x17 cm). A parte de esto, tampoco necesitamos un gran procesador puesto que básicamente hará tareas de reproducción de medios y cuando se requiere un procesador más potente necesitamos una refrigeración mayor lo que supondría más ruido en el dispositivo.

Por último creemos que el panel trasero de este modelo es muy completo y satisface perfectamente nuestras necesidades. Incluye además lector Compact Flash y PC – Card (PCMCIA) que es muy útil para tener almacenamiento

masivo. Soporta unidades IDE y posee tarjetas de expansión de bus PCI. Todas estas características hacen de este modelo el candidato elegido.

#### **6.1.4. Consecución de expectativas**

Como este apartado indica en su título, pretendo recoger la consecución de las expectativas generadas en el momento de elegir este modelo de placa para este proyecto. De esta forma podemos advertir en qué nos ha defraudado la compra o si satisfizo todas nuestras expectativas.

La valoración general que tenemos de esta placa es positiva porque ha cubierto las necesidades que exigíamos. Durante el transcurso de este proyecto no ha dado problema alguno por lo que podemos afirmar que es un hardware de calidad y un dispositivo fiable. Además, las pruebas ejecutadas sobre el dispositivo han sido satisfactorias. El sistema operativo cargado siempre funcionó adecuadamente y aún no pudimos apreciar que el sistema se colgase durante su ejecución, por este motivo, podemos decir que el sistema no conlleva a inestabilidades en el sistema.

El soporte de VIA es el adecuado puesto que en su web oficial disponemos del BSP de VIA EPIA MII. Con este paquete de soporte a la placa disponemos de los drivers necesarios para nuestro dispositivo específico, por ejemplo la tarjeta de audio.

Como puntos negativos podemos destacar dos:

- Creíamos que existía la posibilidad de cargar el sistema operativo desde una tarjeta Compact Flash insertada en el puerto del dispositivo. Esta opción tuvo fue rechazada porque el dispositivo no acepta este tipo de arranque.
- VIA Technologies debería dar soporte en su BSP para el almacenamiento persistente. Durante la elaboración del sistema operativo vimos que aún añadiendo los elementos relativos al soporte de almacenamiento persistente, como soporte USB, el dispositivo no reconoce estos dispositivos por lo que no disponemos de almacenamiento persistente.

Al primero de los puntos negativos no se le dio más importancia porque podíamos utilizar un disco duro para arrancar desde allí la imagen, por lo que no afectaba al proyecto. El segundo de los puntos negativos nos parece más preocupante porque contábamos con esta funcionalidad y la impresión que da sobre la herramienta Platform Builder y sobre el paquete de soporte de VIA es que resulta insuficiente y complejo. A mi parecer, brindar soporte USB debería ser muy trivial pero resultó ser todo lo contrario.

Suponemos que este problema se resolvería con más tiempo, pero esto es un proyecto y debemos cumplir unos plazos y determinar qué líneas de mejora proponemos. A pesar de ello la versatilidad del producto, su estabilidad, fiabilidad y su precio hacen que estemos satisfechos de este producto.

## **CAPÍTULO 7. BALANCE Y CONCLUSIONES**

### **7.1. Revisión de objetivos individuales y de grupo**

#### **7.1.1. Objetivos individuales**

Podemos decir que el objetivo principal de este proyecto se ha conseguido satisfactoriamente.

Recordemos que el objetivo era diseñar un sistema operativo que se ejecutase sobre una placa VIA EPIA MII.

Este objetivo se ha creado con creces puesto que se ha mejorado el sistema operativo con servicios y aplicaciones extras que comentamos en el capítulo de diseño.

#### **7.1.2. Objetivos de grupo**

En relación a los objetivos de grupo hemos tenido algún contratiempo. Debemos recordar que los objetivos de grupo eran dar soporte a las otras partes del proyecto: Aplicación, Protocolo de Red y Hardware. Esto lo vimos en el capítulo de diseño.

Por una parte el objetivo ha sido completado puesto que el sistema operativo da soporte para todas las peticiones de las otras partes. Por lo que podemos decir que el sistema está preparado para soportar todas las demandas de cada una de las partes del proyecto global.

Por otra parte, la aplicación no ha podido reproducir vídeo cuando existe soporte desde el sistema operativo. En cambio la reproducción de audio o bien el audio de un vídeo si que se lleva a cabo correctamente. Esta es una de las mejoras que debemos abordar inmediatamente al fin de solucionarlo lo antes posible.

Que ocurra esto puede hacernos pensar que exista algún tipo de incompatibilidad con alguna librería y/o función de la aplicación pero no se ha demostrado que sea así. Seguiremos trabajando para hallar la respuesta. Aunque podemos descartar que no se presten los servicios por parte del sistema porque de ser así no podríamos reproducir el audio, si puede ser responsable de alguna limitación que desconocemos y deberíamos tener en cuenta porque interfiere en la reproducción de vídeo.



## **7.2. Líneas de trabajo futuras**

La primera mejora que debemos revisar es poder solucionar la reproducción de vídeo que aún no funciona sobre el sistema. Aunque esta medida sea a nivel del proyecto global es primordial solventar este problema.

Otra de las mejoras propuestas y que vemos de gran interés es el soporte de almacenamiento masivo: soporte para disco duro, USB y Compact Flash. Aunque el sistema soporta dichas tecnologías según especificaciones de MSDN no podemos brindar estos servicios. Esta mejora es la única que pensamos aportar de momento al sistema operativo y únicamente afecta a este proyecto en particular.

Creemos que es cuestión de tiempo llevar a cabo estas mejoras pero este proyecto tiene un límite que debemos cumplir por lo que se definen como mejoras y cambios a corto medio plazo.

Por último, comentar que la parte de Hardware se integrará el siguiente cuatrimestre por lo que hay una línea de trabajo abierta en el proyecto global.

## **7.3. Revisión de planificación y costes**

### **7.3.1. Planificación final**

En este apartado veremos que la planificación final real es diferente a la estimada al comenzar este proyecto, esto es lo que sucede normalmente en cualquier proyecto. Este proyecto tiene como objetivo potenciar la capacidad de planificación por lo que se tratará en futuros proyectos de minimizar la diferencia entre la planificación real y la estimada.

Esta planificación tiene en cuenta todos los factores que a priori no se tienen en cuenta.

En mi caso personal, la planificación estimada contaba con una dedicación de cuatro horas cuando posteriormente esta cantidad se vio afectada debida a mi situación laboral. Aunque la mayor extensión de algunas de las tareas no alteran esta media de trabajo por lo que seguimos considerando una media de 4 horas diarias.

En la planificación se tienen en cuenta el tiempo de investigación para lograr funciones como el soporte de almacenamiento masivo y la adición de ficheros a la imagen.

También las dificultades encontradas en la parte de integración con la aplicación han retrasado en transcurso previsto de este proyecto.



A continuación podemos ver la tabla con la planificación real de acuerdo con el calendario finalmente establecido:

**Tabla 1.7.** Planificación de tareas y duración

<b>TAREA</b>	<b>RECURSO</b>	<b>DURACIÓN ESTIMADA (días)</b>
Situación y descripción de objetivos	Analista	5
Análisis del mercado y estudio de productos	Analista	7
Búsqueda de placas y materiales	Analista	7
Elección de placa	Analista	7
Pedido placa	Analista	30
Documentación Situación	Documentador	5
Documentación de análisis	Documentador	7
Documentación de búsqueda de placas	Documentador	5
Documentación elección de placa	Documentador	7
Documentación de Windows CE y Platform Builder	Documentador	6
Documentación de creación de imagen	Documentador	10
Documentación de creación de 2 imagen	Documentador	9
Documentación de imagen final	Documentador	11
Documentación de integración	Documentador	1
Elaboración de memoria	Documentador	30
búsqueda software Platform Builder e instalación	Programador	7
Formación acerca de Windows CE	Programador	4
Formación acerca de Platform Builder	Programador	9
Creación 1era plataforma	Programador	7
Creación de 2a plataforma	Programador	7
Testeo de 1era imagen	Programador	5
creación imagen final	Programador	21
Testeo de 2a imagen	Programador	9
Testeo imagen final	Programador	16
Integración	Programador	35
<b>RECURSO</b>	<b>DURACIÓN TOTAL (días)</b>	<b>DURACIÓN TOTAL (horas)</b>
Analista	26	105
Documentador	91	364
Programador	120	480

\*A la suma de días de trabajo le restamos 30 días del pedido de la placa y añadimos una hora en concepto de esta labor.

Como podemos comprobar se han incrementado considerablemente las horas realizadas por el documentador y el programador. Esto conllevará un incremento en el presupuesto del proyecto.

Como vimos en el capítulo de estudio y análisis, a continuación se muestra el diagrama de Gantt de la planificación real del proyecto. El proyecto finaliza el día 28/09/2006.

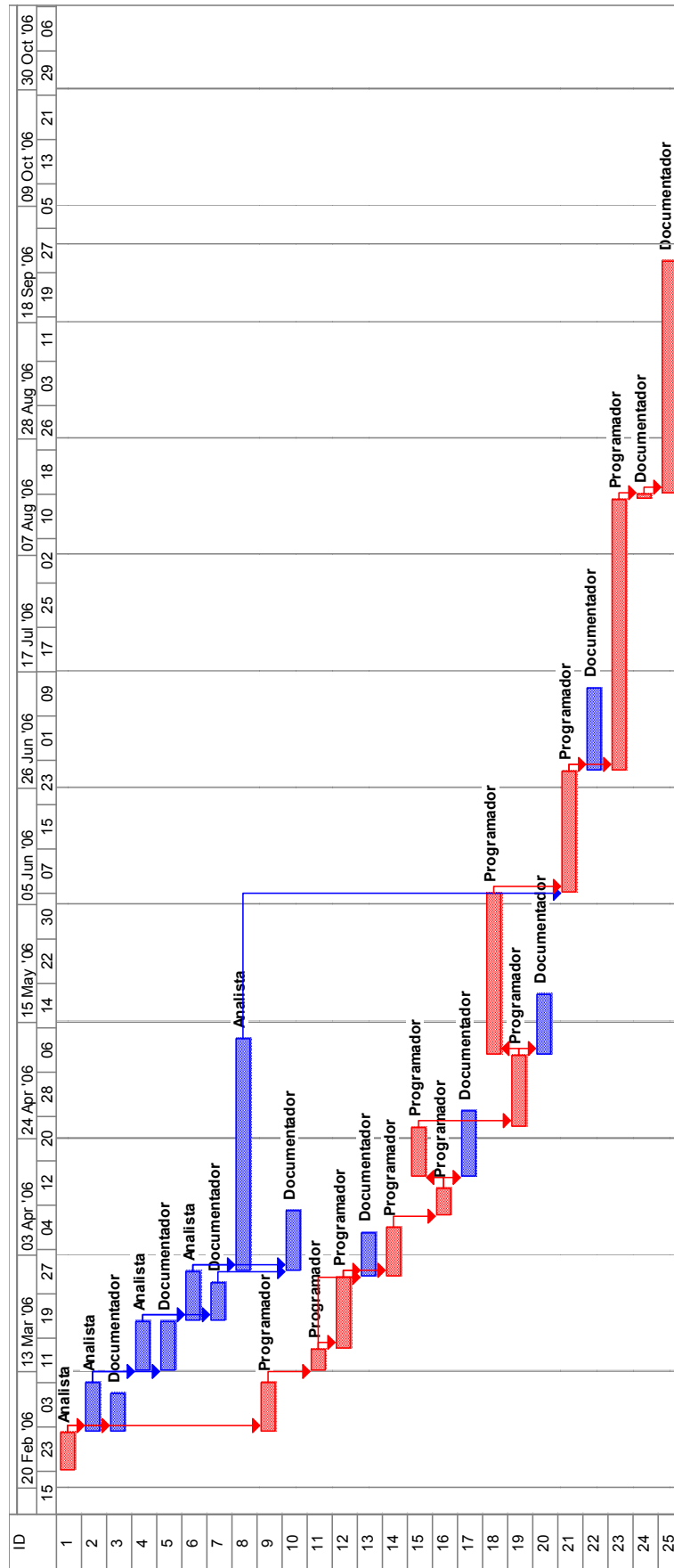


Fig. 1.7 Diagrama de Gantt y camino crítico

### 7.3.2. Costes finales

Los costes finales se verán modificados a razón de la descripción de la planificación anterior.

Las horas de programador y documentador incrementarán el precio total del proyecto.

Las horas por cada trabajador son:

**Tabla 2.7.** Coste total de los recursos humanos

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
Analista	2.625 €
Documentador	9.100 €
Programador	14.400 €

Por otro lado los costes materiales no tienen variación alguna por lo que el presupuesto total resulta:

**Tabla 3.7.** Coste total del proyecto

<b>DESCRIPCIÓN</b>	<b>COSTE</b>
<b>Recursos humanos</b>	<b>26.125 €</b>
Analista	2.625 €
Documentador	9.100 €
Programador	14.400 €
<b>Recursos materiales</b>	<b>1.407,1 €</b>
Platform Builder 5.0	1.099 €
Placa Mini - ITX EPIA MII	308,1 €
<b>TOTAL</b>	<b>27.532,1 €</b>

Por lo que podemos considerar la siguiente tabla comparativa para establecer el desvío respecto el coste de proyecto estimado como:

**Tabla 4.7.** Comparativa coste estimado – coste real

<b>COSTE TOTAL ESTIMADO</b>	<b>COSTE TOTAL REAL</b>	<b>DIFERENCIA</b>
20.512,1 €	27.532,1 €	7.020 €

En resumen, el coste del proyecto resultó ser casi un 35 % más caro que lo estimado inicialmente.

## 7.4. Ambientalización

A todos los efectos, nuestro dispositivo tiene el mismo impacto medioambiental que un ordenador personal por lo que consideraremos el dispositivo como si de un PC se tratara.

A pesar de que a lo largo de esta década se han venido realizando varios estudios sobre el impacto medioambiental de los ordenadores personales, la mayoría presentan información poco exhaustiva y sin actualizar. No obstante, a modo indicativo se pueden extraer algunas conclusiones de los mismos.

El PC de una oficina consume durante su uso 3 o 4 veces más energía primaria que la empleada para fabricar y producir los componentes del equipo, mientras que los costes y créditos energéticos de gestión de residuos y reciclado representan un porcentaje mínimo (menos del 15% de la energía empleada en la producción).

No hay que olvidar que estas cifras corresponden a un PC de oficina típico, utilizado unas 8 horas al día (incluyendo el modo preparado) durante unos 260 días. No obstante, cualquier PC de un hogar al que se le dé un uso moderado y eficiente consume en general a lo largo de sus 6 años de vida útil (media de la UE) más del doble de la energía que se empleó para su producción.

Un ordenador portátil, equivalente a nuestro dispositivo, suele ser de un 50 a un 80% más eficiente en el ámbito energético que uno de escritorio y además es mucho más ligero (de 1.1 a 2 Kg. frente al peso de 8 Kg. o más de uno de escritorio). Por lo tanto, a lo largo de la vida útil de un ordenador, el consumo energético resulta uno de los factores dominantes.

La vida media de un PC en los países de la Comunidad Europea se sitúa entorno a los 6 años y en el mercado de los PC de segunda mano, la vida media de los modelos que se ofrecen es de unos 3 años. Las nuevas CPUs (Unidad Central de Procesamiento) suelen consumir más energía que las CPUs antiguas (por ejemplo un procesador 486 de 60 W con un Pentium-4 de 120 W en modo encendido), por ello se recomienda, desde una perspectiva ecológica, alargar al máximo la vida del producto y, en la medida de lo posible, no utilizarlo como servidor.

Además del ahorro energético, gracias a la legislación actual se ha avanzado en otros aspectos medioambientales como la eliminación de determinados retardadores de llama, la introducción de soldaduras que no contienen plomo, etc. Asimismo, se está debatiendo sobre el uso de PVC en los cables, del mercurio en las pantallas, sobre las consecuencias de las radiaciones electromagnéticas y el ruido, entre otros. Resulta difícil medir el impacto que tienen todas estas cuestiones en la energía y en las emisiones de CO<sub>2</sub> pero lo que es evidente es que la mayoría de estos aspectos se encuentran fuera del alcance de los compradores o consumidores y los resultados suelen venir a través de la legislación (Directiva sobre residuos de aparatos eléctricos y electrónicos - RAEE, Restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos, etc.).

En general, el PC es un producto en el que no se consideran predominantes estos aspectos medioambientales. Por ello, el reciente informe sobre la etiqueta ecológica en los equipos informáticos establece como ámbitos prioritarios de actuación el ahorro en el consumo energético y el ruido.

## **7.5. Conclusiones**

Las conclusiones extraídas después de la elaboración de este proyecto son muy positivas.

El proyecto ha sido muy interesante y fructificante para completar mis estudios universitarios.

He adquirido conocimientos en una materia que se puede considerar privilegiada puesto que hay pocos desarrolladores de Platform Builder.

Además, he mejorado mi capacidad de trabajar en equipo, capacidad que se exige en el mundo laboral, por lo que he potenciado una cualidad fundamental.

En algunas ocasiones el transcurso del proyecto resultó frustrante debido a dos factores principales: la complejidad de la herramienta PB que en algunas ocasiones desespera y la duración del proyecto dificultada por mi compromiso laboral.

No obstante, la experiencia ha servido para curtirme en dichas experiencias, aceptar las limitaciones que existen en el momento de realizar un proyecto y valorar más positivamente el trabajo realizado.

Por todos estos motivos debo afirmar que estoy muy satisfecho del trabajo realizado y creo que el proyecto global, pese a las dificultades de integración, es un éxito debido al duro trabajo de todos los integrantes.

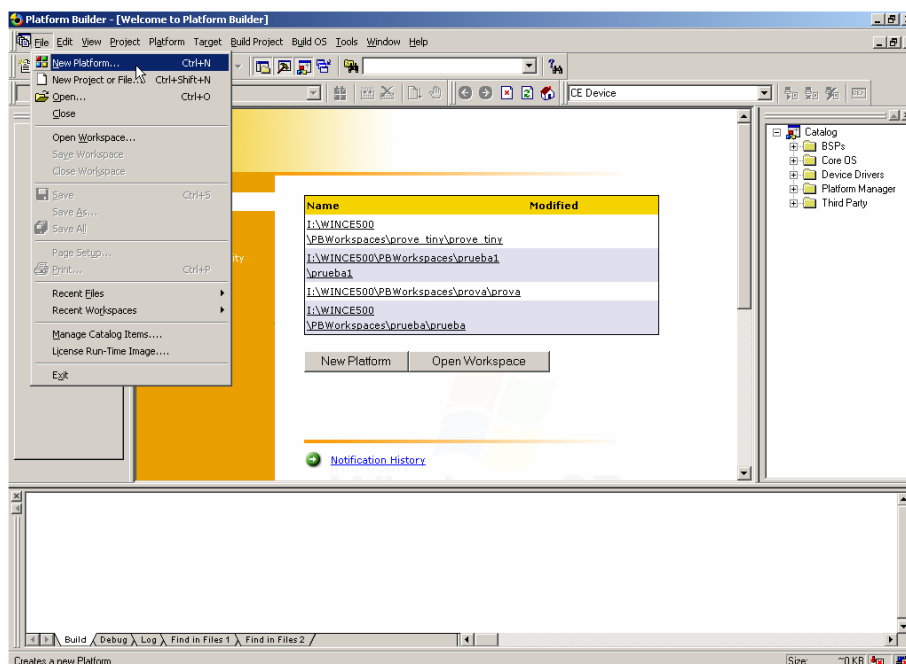


## CAPÍTULO 8. APÉNDICE

### 8.1. Diseño paso a paso de las plataformas

#### 8.1.1. Imagen basada en Tiny Kernel

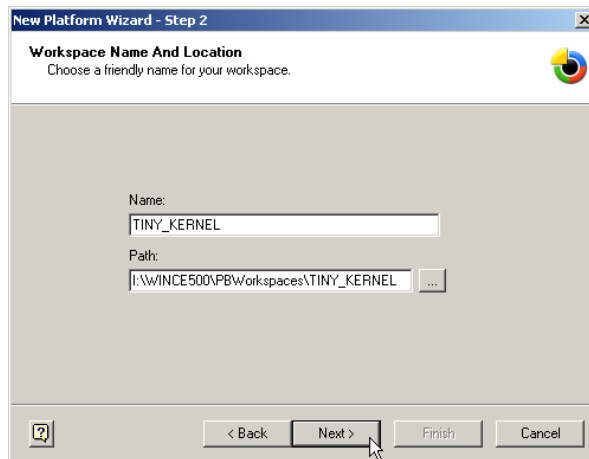
El primer paso será hacer clic en el menú “File”→”New Platform”, para crear una nueva plataforma:



**Fig. 1.8** Captura de Platform Builder. Iniciando asistente

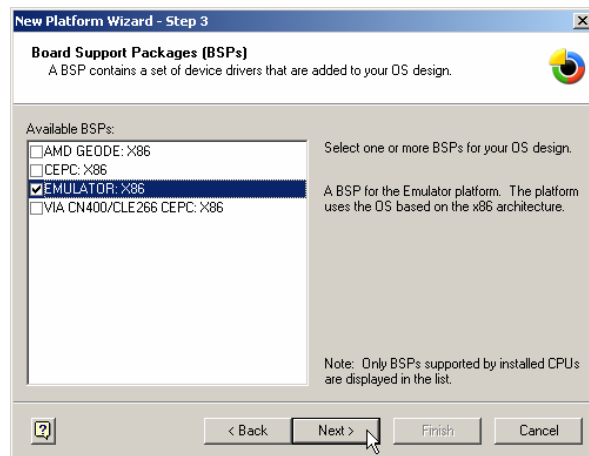
Inmediatamente se abrirá una pantalla de asistente de creación de la plataforma, la primera página es de información por lo que haremos clic en Siguiente (“Next”).

Seguidamente deberemos escoger un nombre para nuestra plataforma y el directorio donde se guardará el proyecto de dicha plataforma y hacemos clic en Siguiente.



**Fig. 2.8** Captura de Platform Builder. Primer paso, definir nombre al proyecto

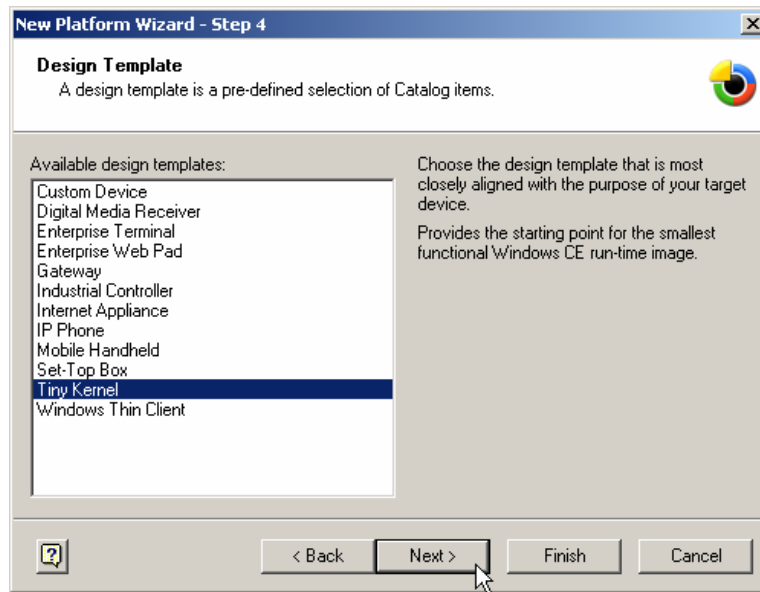
Ahora debemos elegir el dispositivo sobre el que correrá nuestra plataforma. En nuestro caso, deberíamos elegir el BSP apropiado a nuestra placa, pero como aún se debe de la imagen inicial optaremos de momento a simularlo con el emulador de Windows CE. Por este motivo elegiremos el emulador y haremos clic en Siguiente.



**Fig. 3.8** Captura de Platform Builder. Elegir BSP deseado

En el siguiente paso elegimos la opción “Tiny Kernel” ya que es la deseada en este caso y hacemos clic sobre Siguiente.

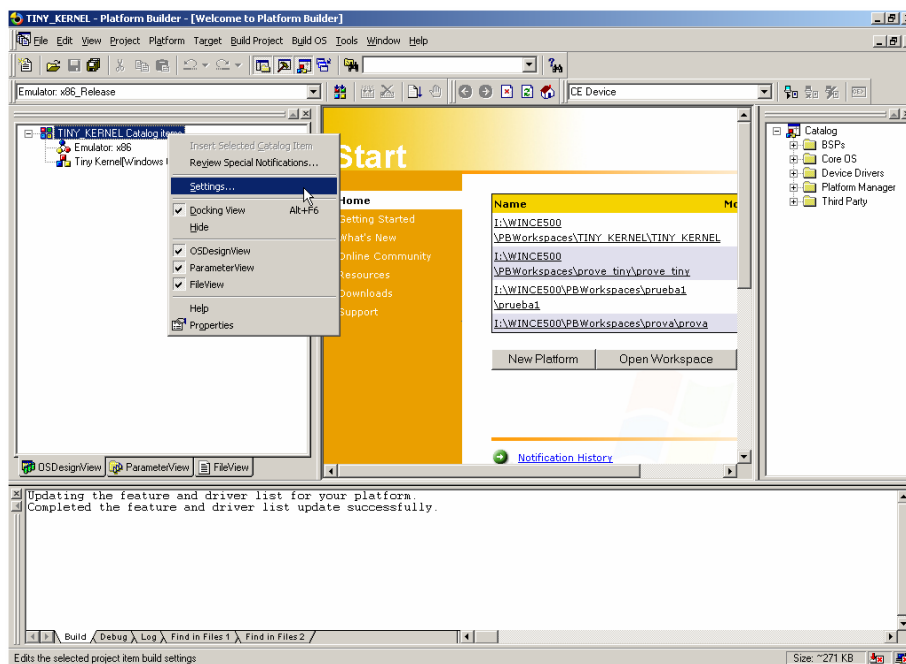




**Fig. 4.8** Captura de Platform Builder. Elegimos una plantilla de diseño

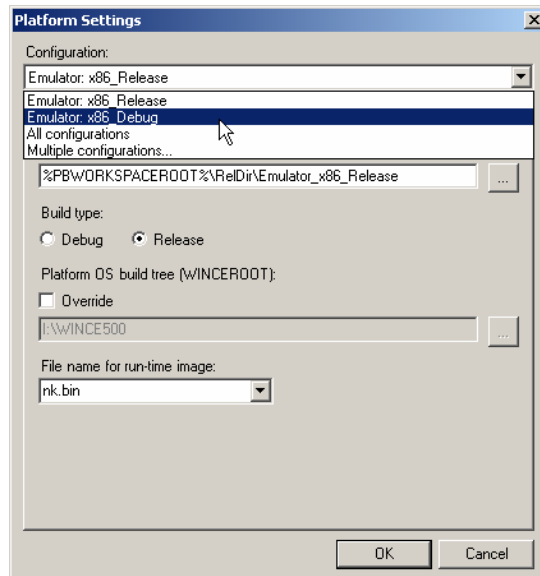
Una vez hecho esto, haremos clic sobre Finalizar (“Finish”) y esperaremos a que se abra el proyecto elegido.

Además podremos ver como en la parte izquierda del Platform Builder se cargó nuestro proyecto llamado “TYNY KERNEL” y los dispositivos que soporta. Nuestro siguiente paso es seleccionar con el botón derecho nuestro proyecto en el visor de diseño de SO (“OS Design Viewer”) y abrir las Opciones (“Settings”).



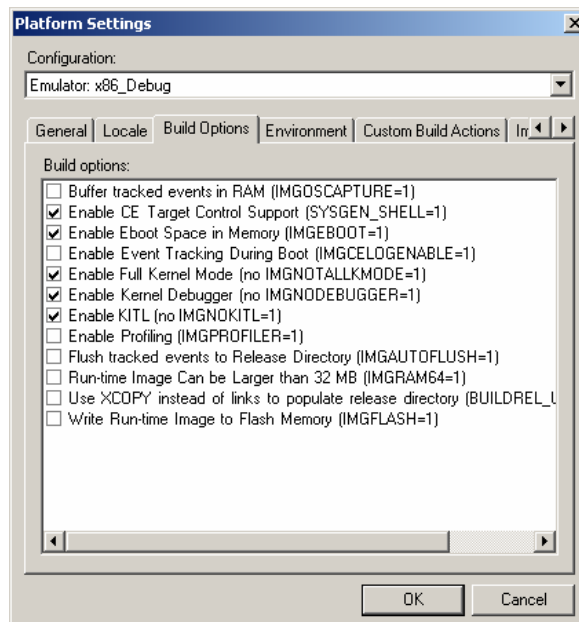
**Fig. 5.8** Captura de Platform Builder. Opciones de la plataforma

Ahora en el menú de Configuración (“Configuration”) elegimos la opción “Emulador: x86\_Debug” en lugar de “Emulador: x86\_Release”.



**Fig. 6.8** Captura de Platform Builder. Elegimos versión “Debug”

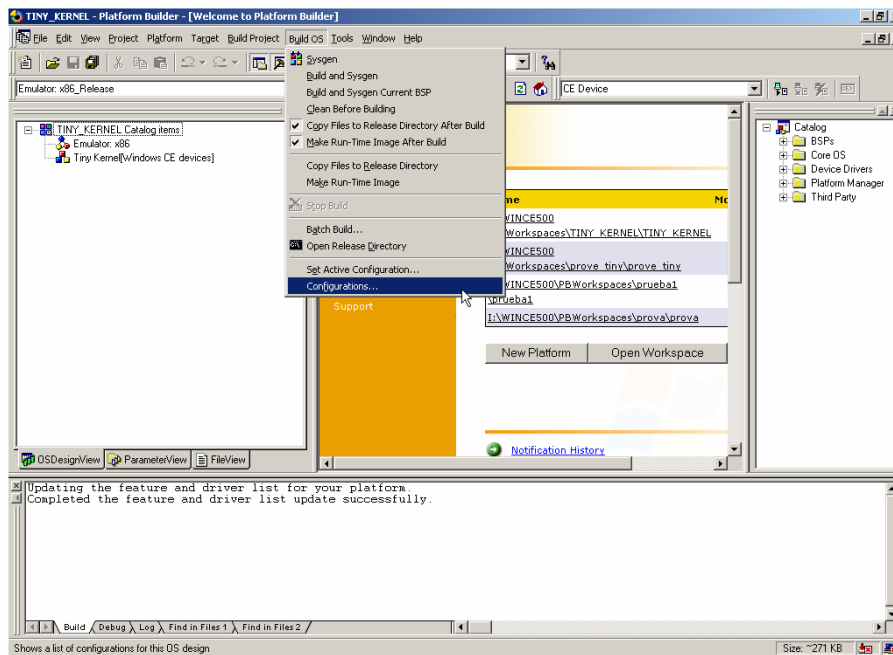
En la pestaña de “Build Options” debemos marcar las siguientes opciones que más adelante explicaremos en qué consisten. Después pulsamos “OK”.



**Fig. 7.8** Captura de Platform Builder. Seleccionamos las opciones expuestas

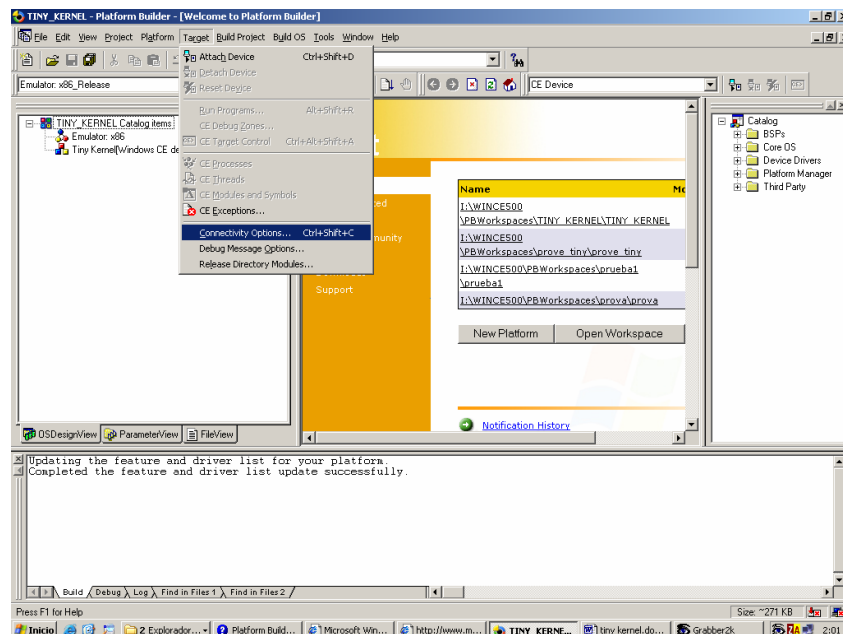
Después debemos activar que las opciones “Copy Files to Release Directory After Build” y “Make Run-Time Image After Build” en el menú “Build OS”.

Además en “Build OS”→”Configurations” deben estar añadidas tanto la opción “Debug” como la “Release”.



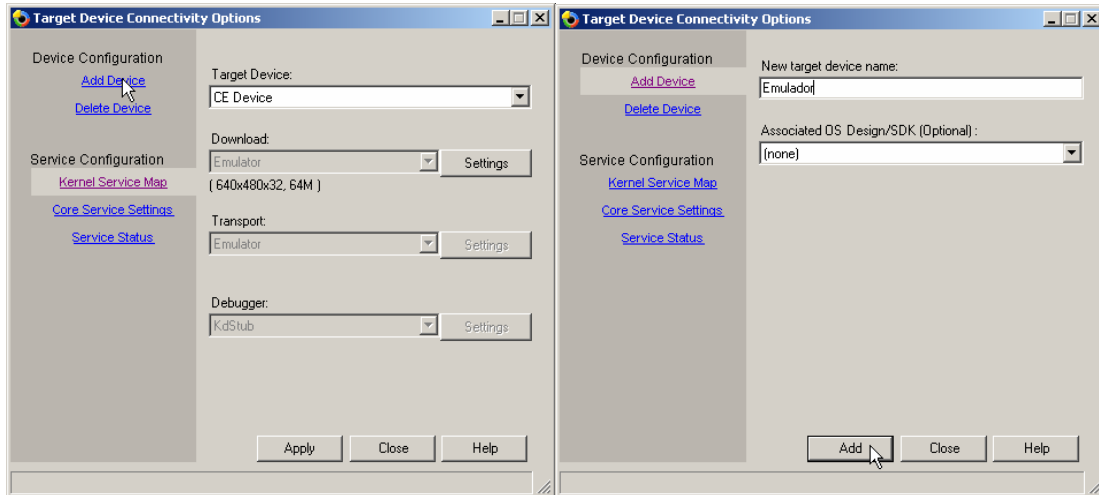
**Fig. 8.8** Captura de Platform Builder. Comprobamos “Configurations”

Por último debemos enlazar el dispositivo con la plataforma que después de compilarla y debutarla correrá en él. Para esto nos situaremos en el menú “Target”→”Connectivity Options”.



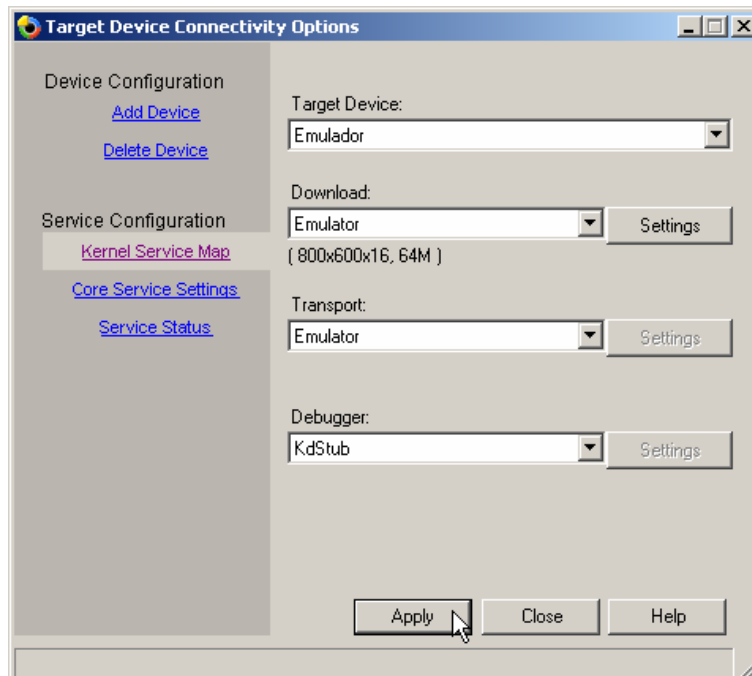
**Fig. 9.8** Captura de Platform Builder. Accedemos a opciones de conectividad

Se abrirá una nueva pantalla. En la parte izquierda añadiremos un nuevo dispositivo haciendo clic en “Add Device”. En este caso, lo nombraré “Emulador” ya que es el dispositivo donde finalmente correrá. Finalmente haremos clic en “Add”.



**Fig. 10.8** Captura de Platform Builder. Creando un nuevo dispositivo

Debemos comprobar que en “Kernel Service Map” se cumple lo siguiente:



**Fig. 11.8** Captura de Platform Builder. Configuración principal de opciones de conectividad

Asimismo debemos comprobar que en “Core Service Settings” la configuración es:

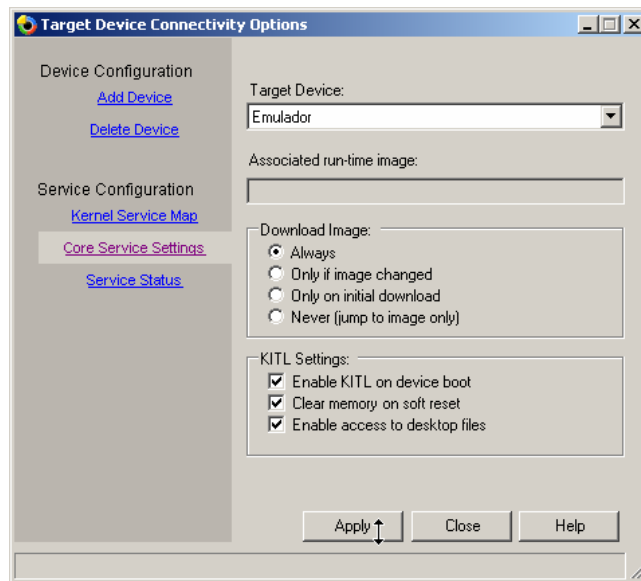


Fig. 12.8 Captura de Platform Builder. Opciones de servicios del núcleo

Finalmente haremos clic sobre “Apply” y después “Close”. Y ya tendremos configurado nuestro emulador como el dispositivo que ejecutará la imagen de nuestra plataforma.

Recordemos marcar la opción Emulador: x86\_Debug, en “Set Active Configuration”.

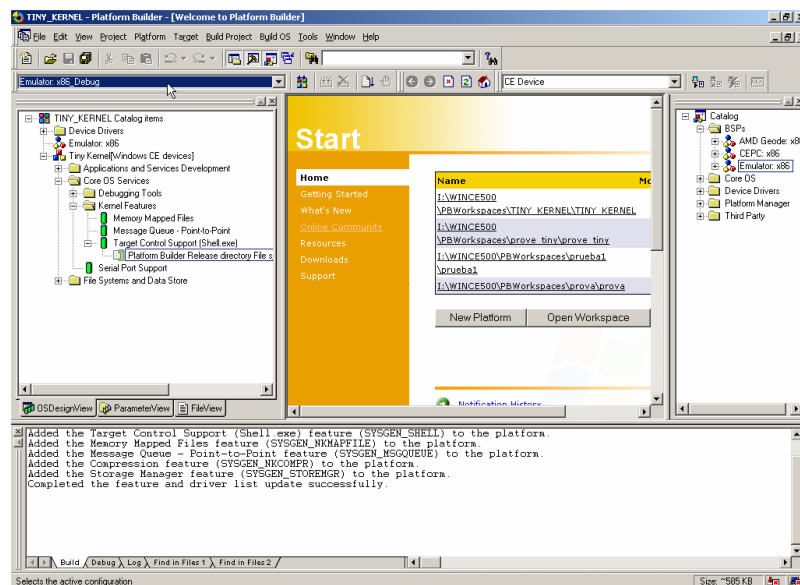
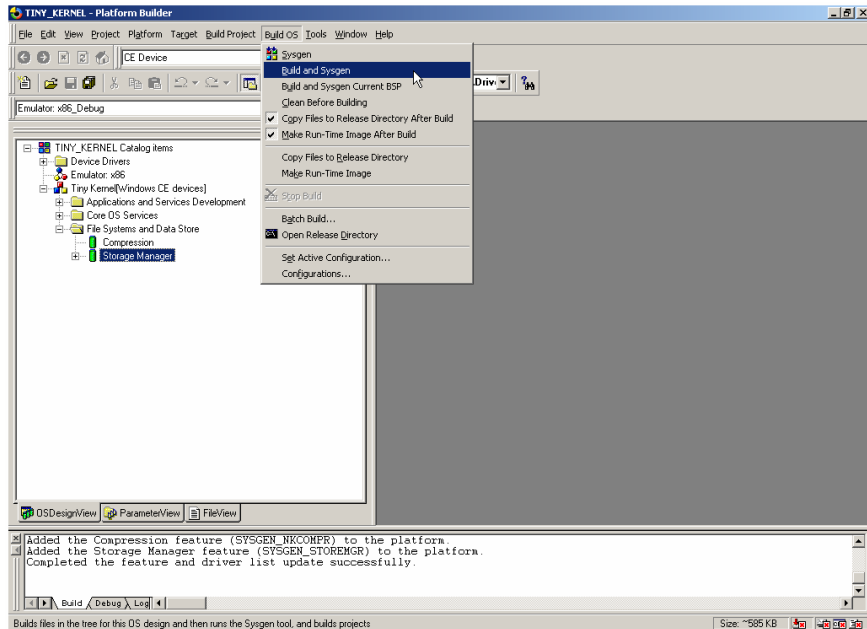


Fig. 13.8 Captura de Platform Builder. Elección de plataforma a compilar

Como vemos sale el árbol de dispositivos necesarios y librerías añadidas al núcleo de nuestro sistema operativo.

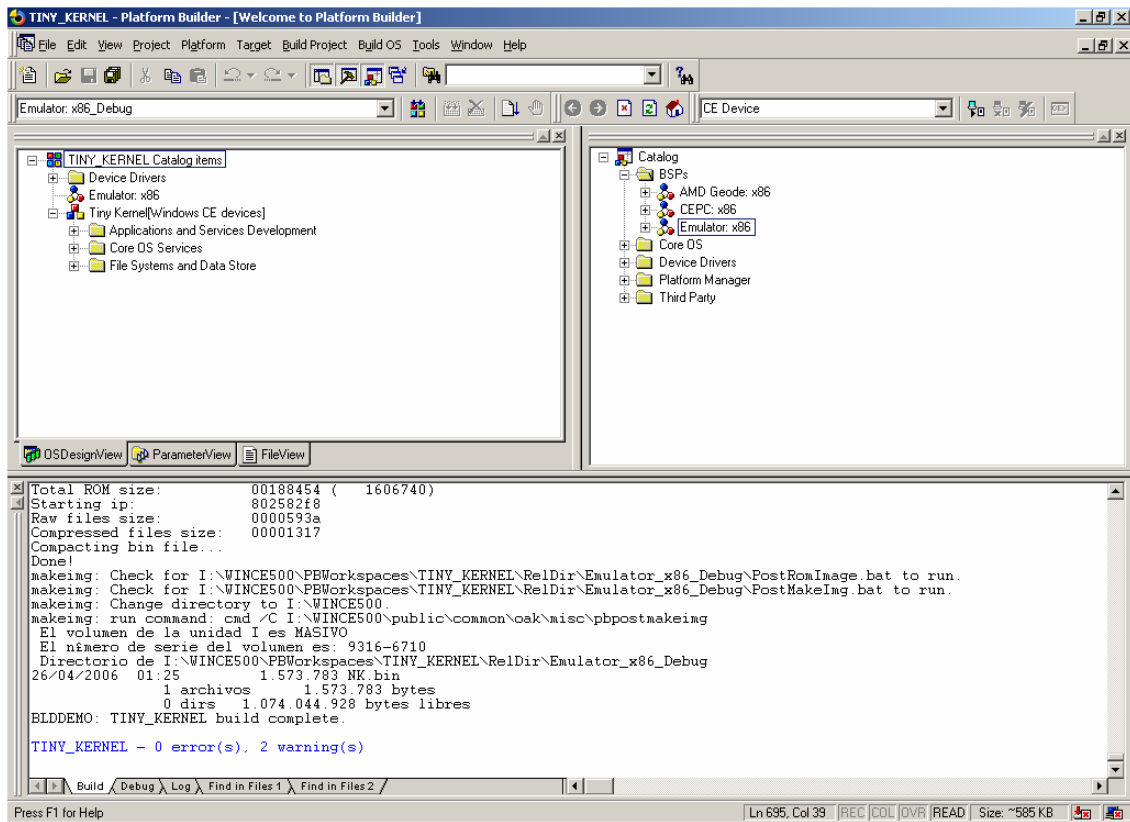
Por último sólo nos queda compilar y generar la imagen de nuestra plataforma accediendo al menú “Build OS” → “Build and Sysgen”.



**Fig. 14.8** Captura de Platform Builder. Compilamos nuestro proyecto

Un marcador en la parte derecha inferior de la pantalla indicará que la aplicación de Platform Builder está ahora compilando. Además podemos observar como la ventana de compilación comienza a brindarnos información acerca del proceso que le ocupa.

Al finalizar el proceso nos dará el resultado de errores y warnings en la compilación y creación de la imagen.



**Fig. 15.8** Captura de Platform Builder. Finalizada la compilación

En este caso nos ofrece un resultado de 2 warnings, los tres corresponden a warnings de bien de localización como de creación de directorios de sistema. Muy posiblemente estos sean debidos a que estamos realizando la imagen sobre el emulador y seguramente no es capaz de crear los directorios adecuados. Los warnings son los siguientes y aparecen en la ventana de “Build” en color azul:

BUILD: [00:0000000008:WARNS ] Directory: I:\WINCE500\PRIVATE\OSINTERNAL\TEST\HAL\ does not exist.

BUILD: [00:0000000026:WARNS ] Directory: I:\WINCE500\private\osinternal\test\hal\ does not exist.

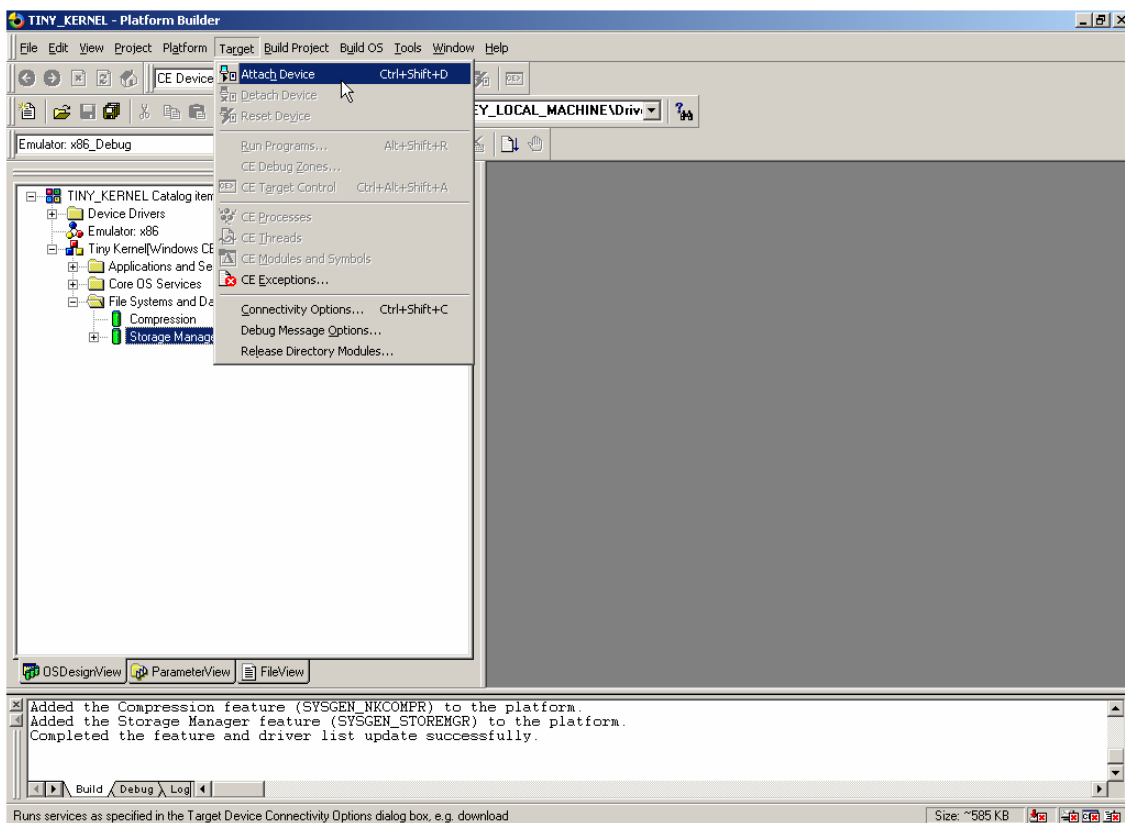
Después de compilar la imagen podemos ver los módulos añadidos a esta.

MODULES Section	Section	Start	Length	psize	vsize	Filler
nk.exe	FileName	803a7ff8	7			FILLER
kd.dll	FileName	802faaf4	7			FILLER
hd.dll	FileName	80389fec	7			FILLER
osaxst0.dll	FileName	80389ff4	12			FILLER
osaxst1.dll	FileName	80346fe4	12			FILLER
coredll.dll	FileName	80346ff0	12			FILLER
fileSYS.exe	FileName	802bbff0	12			FILLER
relfsd.dll	FileName	80390d94	11			FILLER
shell.exe	FileName	80390da0	10			FILLER
shellcelog.dll	FileName	80390dac	15			FILLER
toolhelp.dll	FileName	80390dbc	13			FILLER
fsdmgr.dll	FileName	80390dcc	11			FILLER
ceconfig.h	FileName	80390dd8	11			FILLER
initobj.dat	FileName	80390de4	12			FILLER
default.fdf	FileName	80390df0	12			FILLER

**Fig. 16.8** Captura de Platform Builder. Algunos de los elementos añadidos a la plataforma

Ahora ya podemos probar nuestra imagen enlazando el dispositivo basado en emulador antes creado con la imagen de Windows CE compilada. Para realizar este paso sólo debemos acceder al menú “Target”→”Attach Device”.

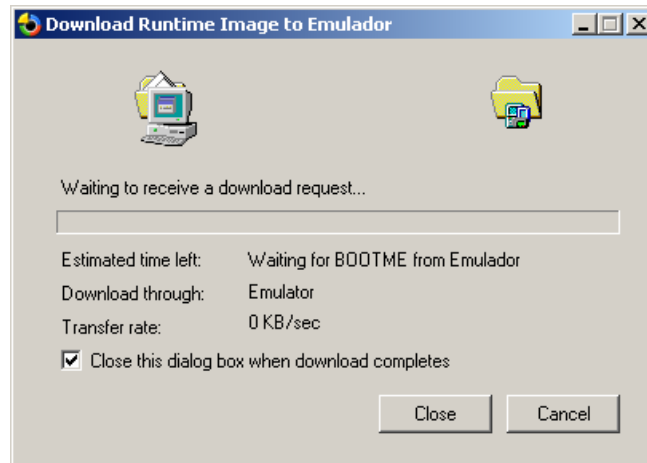
Recordemos que antes habíamos seleccionado en “Target”→”Connectivity Options” el dispositivo emulador como lugar de arranque de la imagen.



**Fig. 17.8** Captura de Platform Builder. Enlazamos con nuestro dispositivo

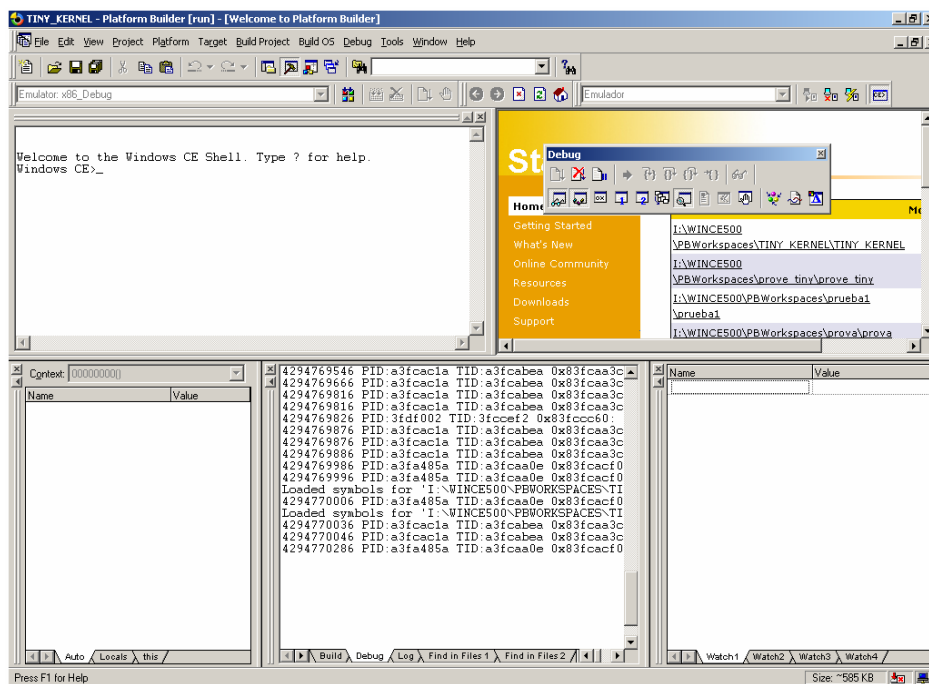


Una ventana como la siguiente nos informa que la transferencia de la imagen sobre el dispositivo se está realizando y en qué momento se encuentra de la descarga hacia el dispositivo. En nuestro caso la imagen sólo ocupa unos 585 KBytes y la descarga será bastante rápida aunque debemos esperar que cargue el emulador. La información del tamaño actual de nuestra imagen la podemos ver en la esquina inferior derecha.



**Fig. 18.8** Captura de Platform Builder. Descargando imagen sobre el dispositivo

Esperados estos instantes, la pantalla principal de Platform Builder quedará de la siguiente forma:

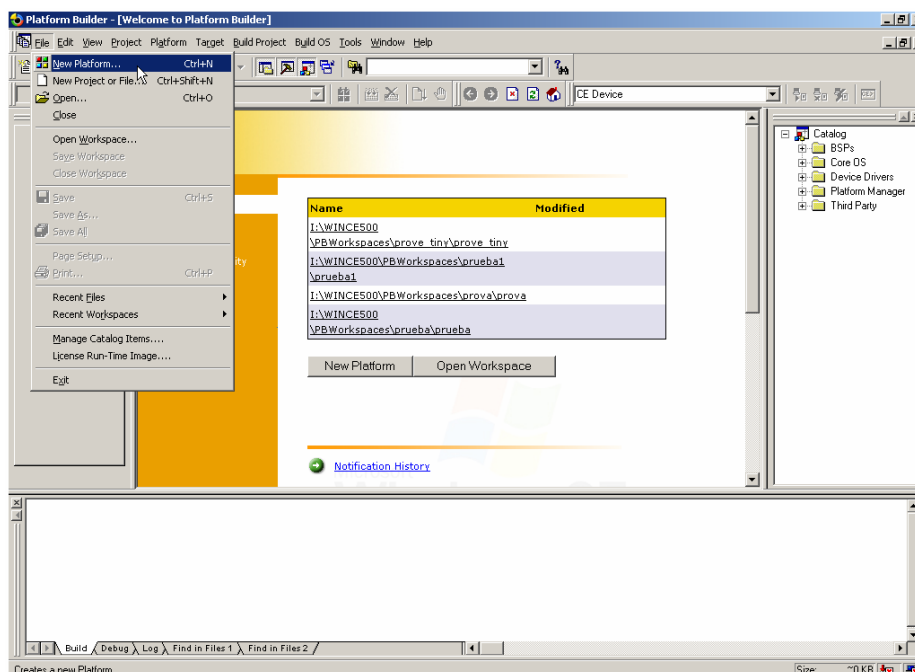


**Fig. 19.8** Captura de Platform Builder. Estado final de Platform Builder

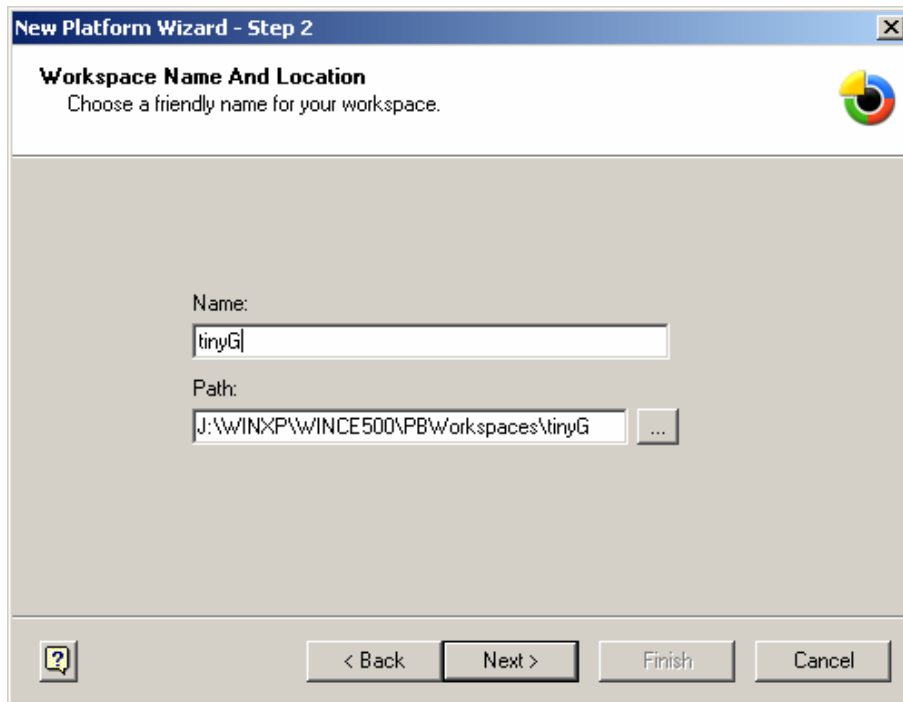
Donde, como podemos ver existe una terminal donde podemos ejecutar comandos sobre nuestro sistema operativo. Cabe destacar que Tiny Kernel prescinde de las librerías gráficas por ello tenemos un interfaz de consola. Además la funcionalidad del núcleo es limitada, puesto que al tratarse del Tiny Kernel no existen la mayoría de las funcionalidades propias de Windows CE.

### 8.1.2. Imagen basada en Tiny Kernel con interfaz gráfica

Accedemos al asistente de creación de imagen en el menú superior “File” -> “New Platform”. Después sólo tendremos que fijar la ruta y el nombre del proyecto que vamos a crear.

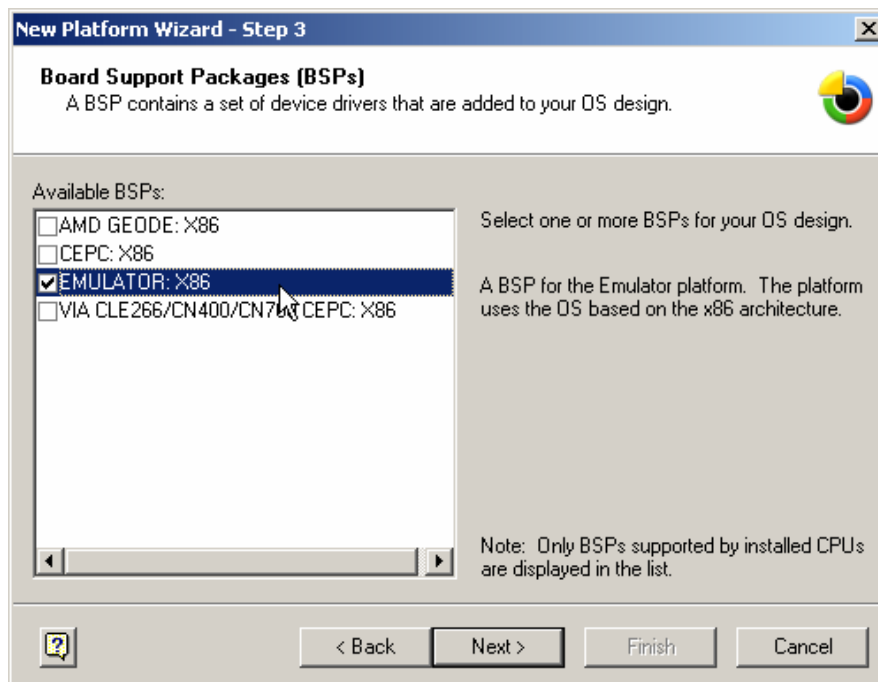


**Fig. 20.8** Captura de Platform Builder. Iniciando el asistente

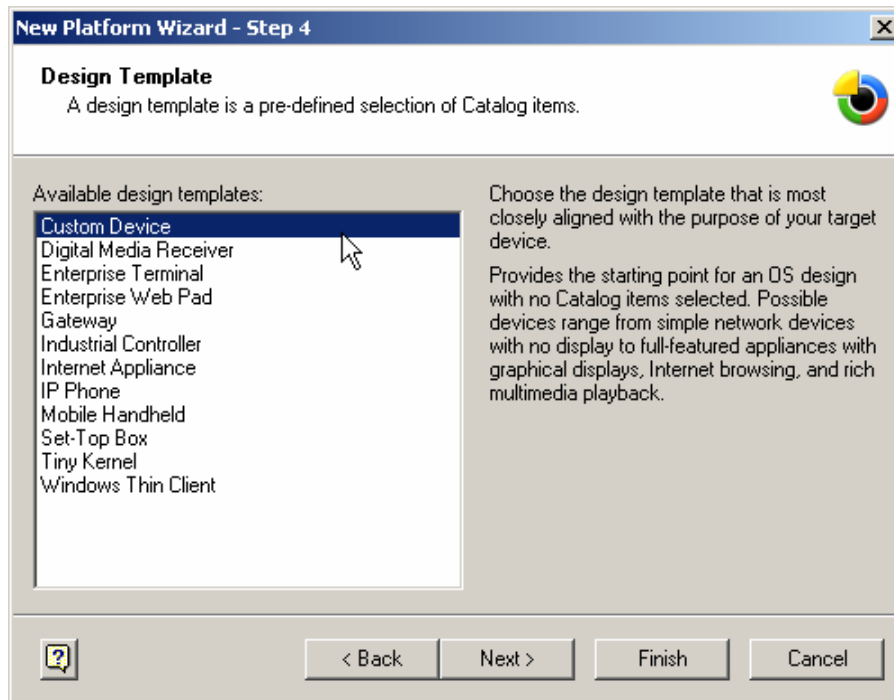


**Fig. 21.8** Captura de Platform Builder. Nombre y ruta del proyecto

Elegimos el BSP para el cuál será diseñado el sistema operativo, en este caso el emulador. En la siguiente pantalla elegimos el tipo de dispositivo, en este caso elegimos “Custom Device” que aunque su base es la misma que el “Tiny Kernel” nos permitirá escoger todos los elementos disponibles.

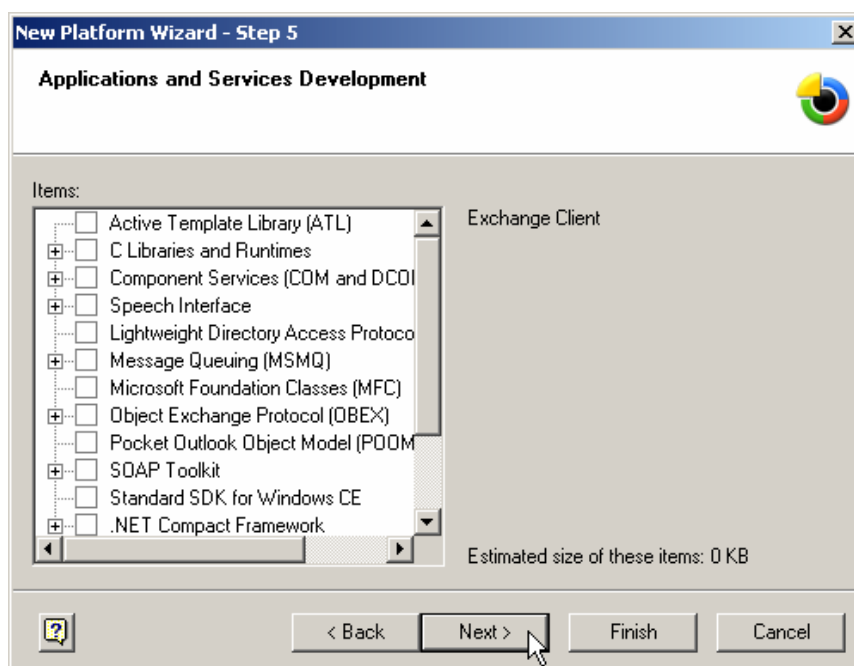


**Fig. 22.8** Captura de Platform Builder. Elegimos BSP del dispositivo

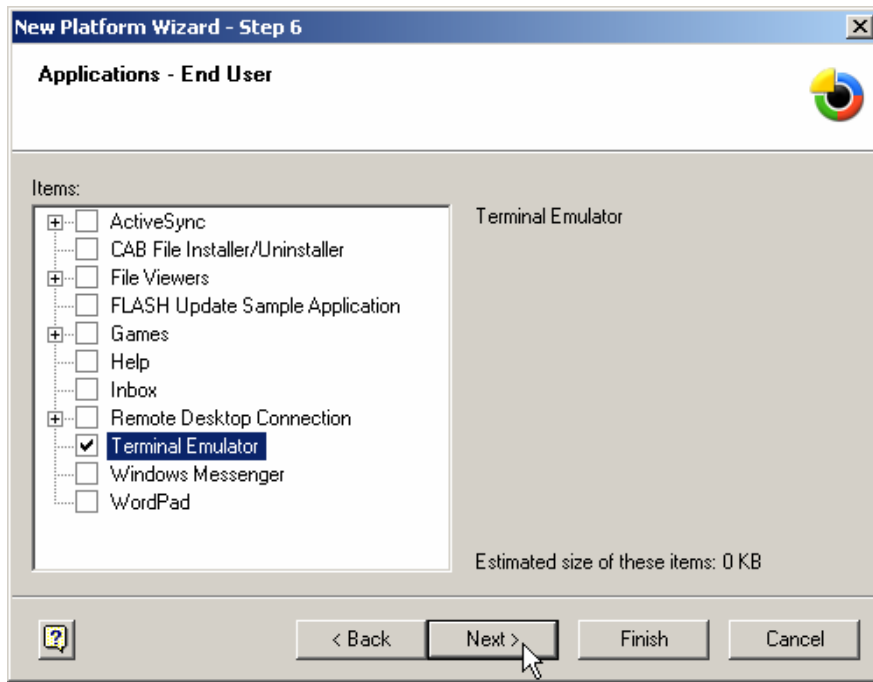


**Fig. 23.8** Captura de Platform Builder. Elección tipo de dispositivo

A continuación veremos una consecución de ventanas donde cada una será una agrupación de elementos (módulos y servicios) que guardan relación entre sí. La siguiente figura nos muestra los elementos disponibles del grupo “Applications and Services Development”, pero como queremos hacer una plataforma lo más sencilla pero con interfaz gráfica no elegiremos ninguno de este grupo. En cambio, del siguiente elegimos “Terminal Emulator”.

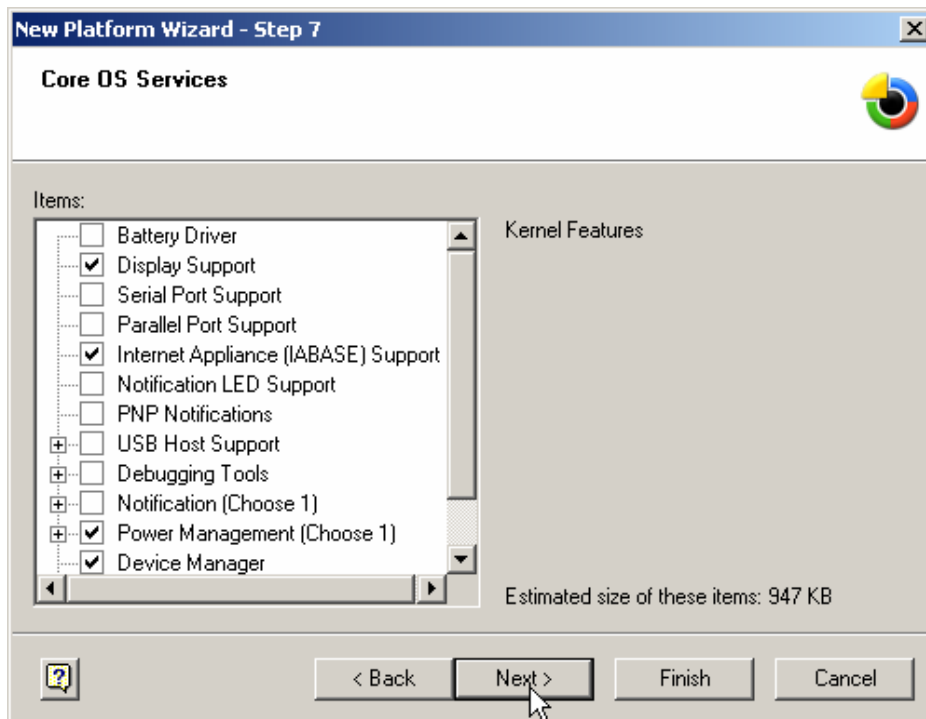


**Fig. 24.8** Captura de Platform Builder. Aplicaciones y Servicios de Desarrollo

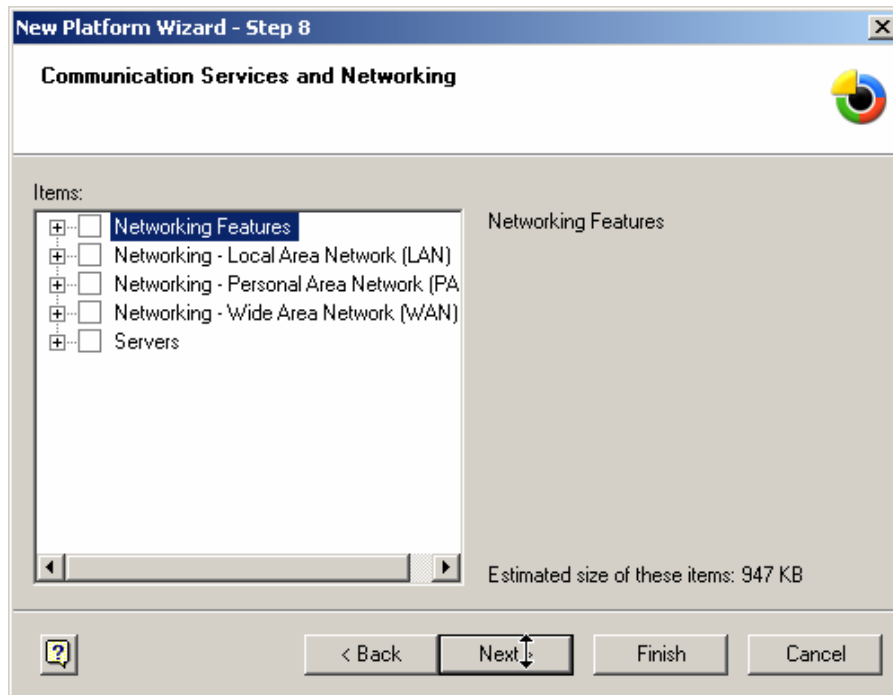


**Fig. 25.8** Captura de Platform Builder. Aplicaciones – Usuario final

Debemos elegir “Display Support”, “Internet Appliance”, “Power Management Full” y “Device Manager” y “Kernel Features” del grupo “Core OS Services”. En cambio, del grupo “Communication Services and Networking” no seleccionamos ninguno puesto que en esta versión no necesitamos conexión.

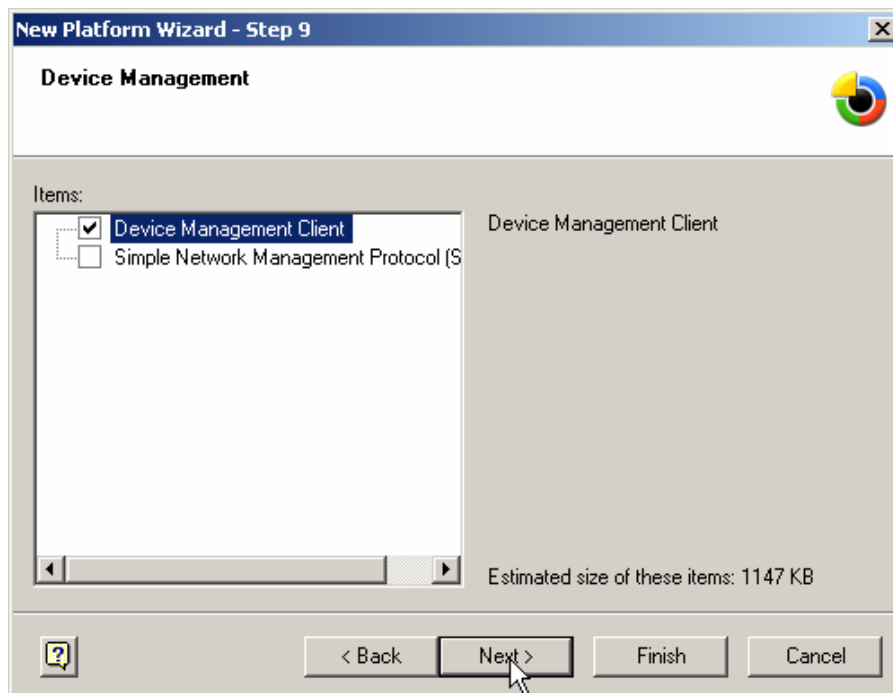


**Fig. 26.8** Captura de Platform Builder. Servicios del núcleo de SO

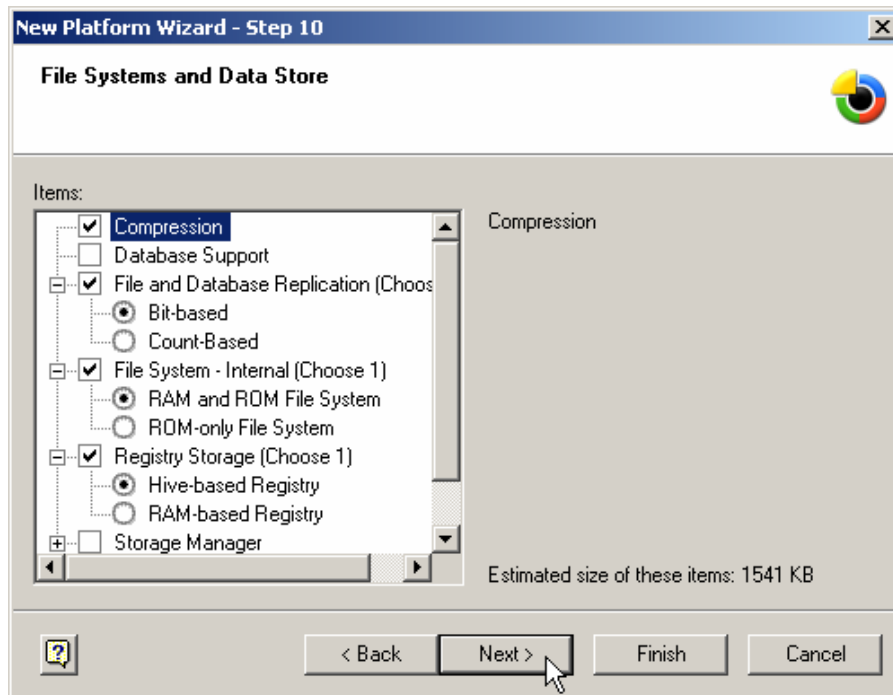


**Fig. 27.8** Captura de Platform Builder. Servicios de comunicación y redes

En el apartado de “Device Management” seleccionamos el componente “Device Management Client”. En la siguiente ventana elegimos “Compression”, “File and Database Replication” -> “Bit-based” y “Registry Storage” -> “Hive-based Registry”.

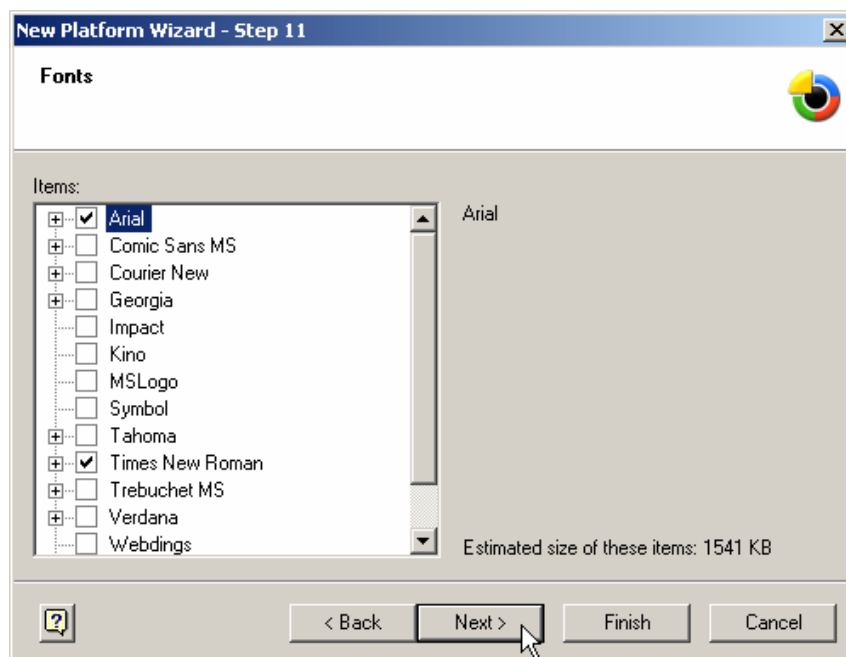


**Fig. 28.8** Captura de Platform Builder. Gestión de dispositivos

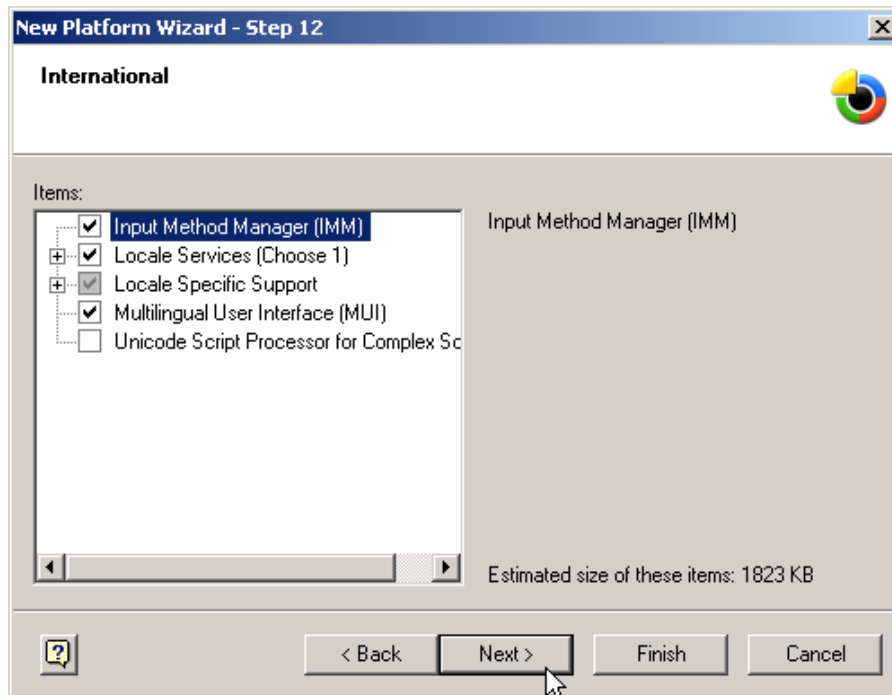


**Fig. 29.8** Captura de Platform Builder. Sistemas de archivos y almacenamiento

En las siguientes ventanas encontramos las fuentes disponibles del sistema, donde elegimos alguna a nuestro gusto y la ventana de “Internacional” donde se da soporte para multilinguaje. Aunque es totalmente prescindible por ahora he seleccionado los elementos que se ven en la figura 31 a modo de probar la posibilidad de utilizar el teclado español.

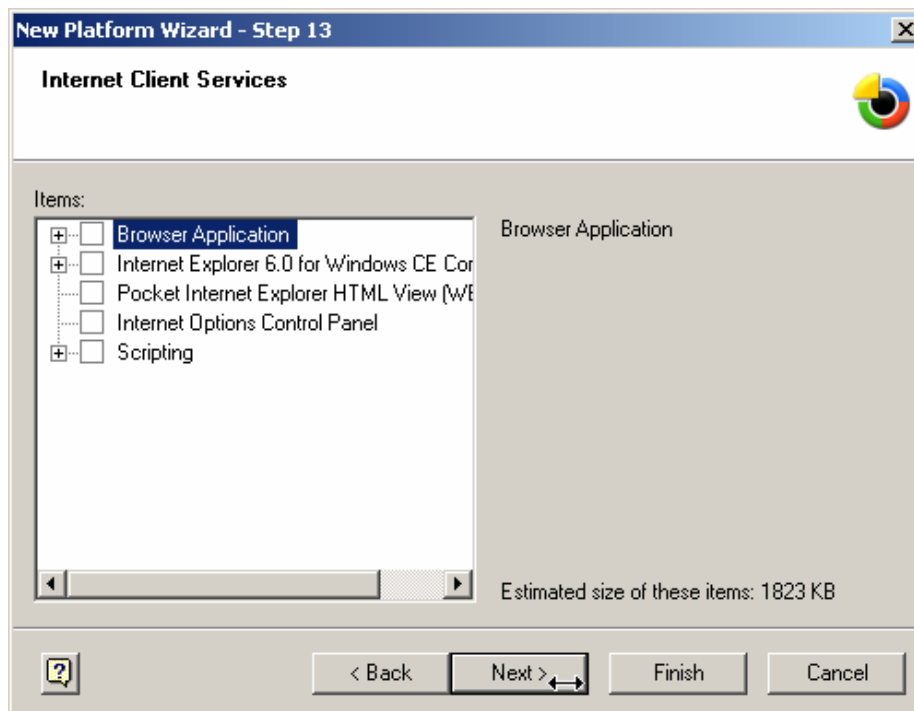


**Fig. 30.8** Captura de Platform Builder. Fuentes del sistema operativo



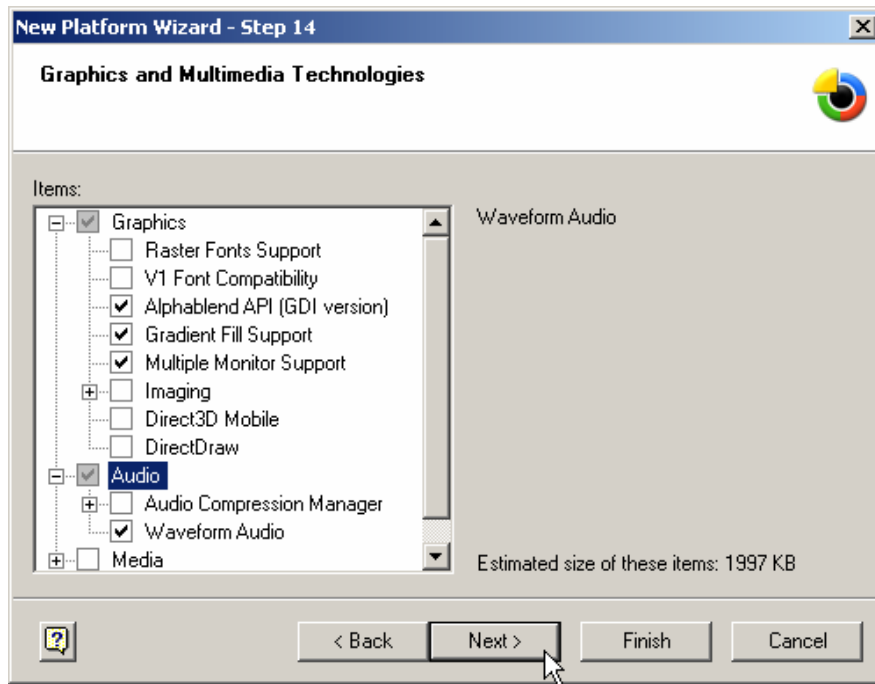
**Fig. 31.8** Captura de Platform Builder. Soporte internacional

En lo que refiere a los “Internet Client Services” dejaremos sin marcar todos los elementos ya que no queremos disponer de navegador. Por el contrario, en la ventana de “Graphics and Multimedia Technologies” marcaremos los elementos básicos de los grupos “Graphics” y “Audio”.



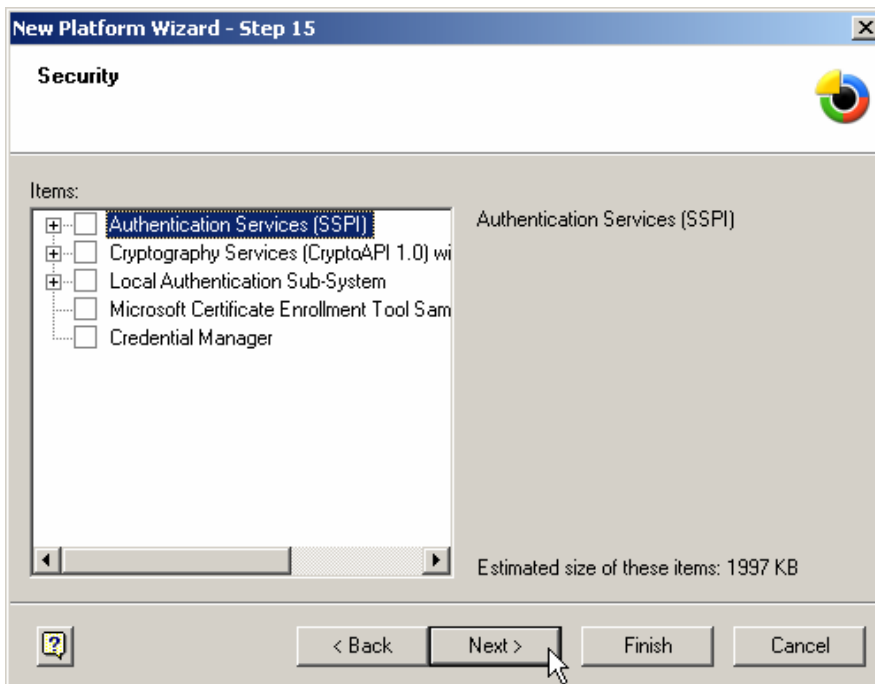
**Fig. 32.8** Captura de Platform Builder. Servicios Cliente de Internet



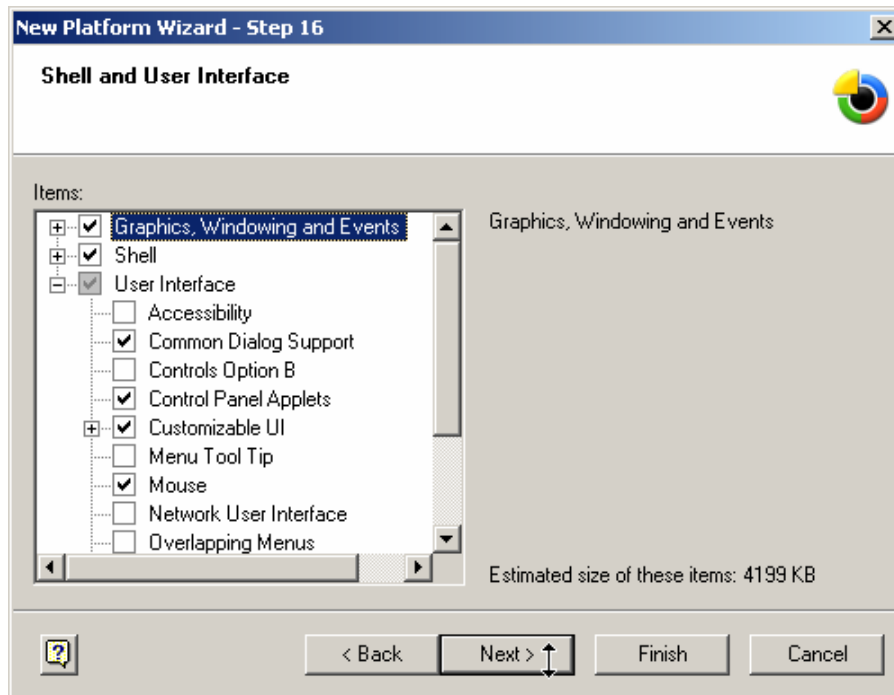


**Fig. 33.8** Captura de Platform Builder. Gráficos y Tecnologías Multimedia

Para la consecución de esta plataforma también prescindiremos de capacidades de este sistema en brindar seguridad. Del grupo “Shell and User Interface” elegiremos “Graphics, Windowing and Events”, “Shell” y del subgrupo “User Interface” marcamos “Common Dialog Support”, “Control Panel Applets”, “Customizable UI”, “Mouse” y “Common Controls”.

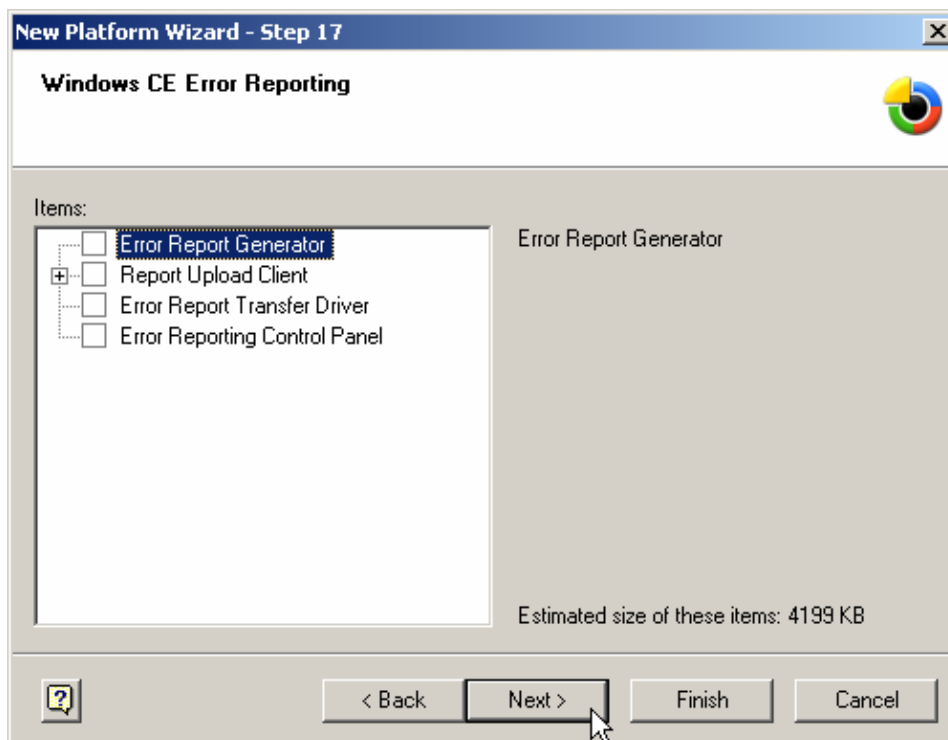


**Fig. 34.8** Captura de Platform Builder. Seguridad

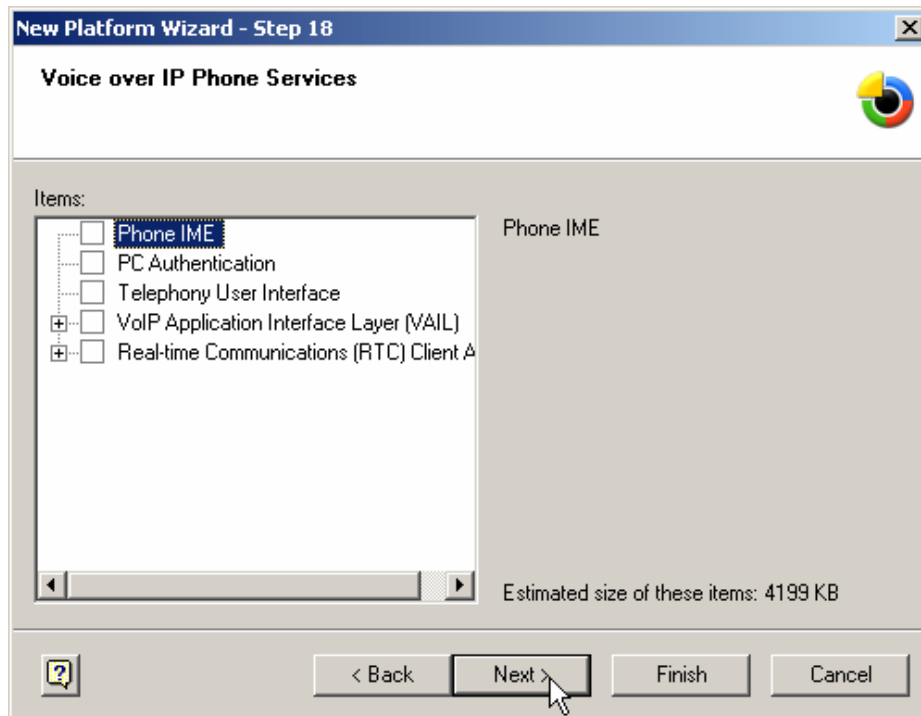


**Fig. 35.8** Captura de Platform Builder. Terminal e Interfaz de Usuario

Tampoco marcamos ningún elemento de “Windows CE Error Reporting” porque no necesitamos informe de errores para nuestro fin. Mucho menos del siguiente grupo llamado “Voice over IP Phone Services”.

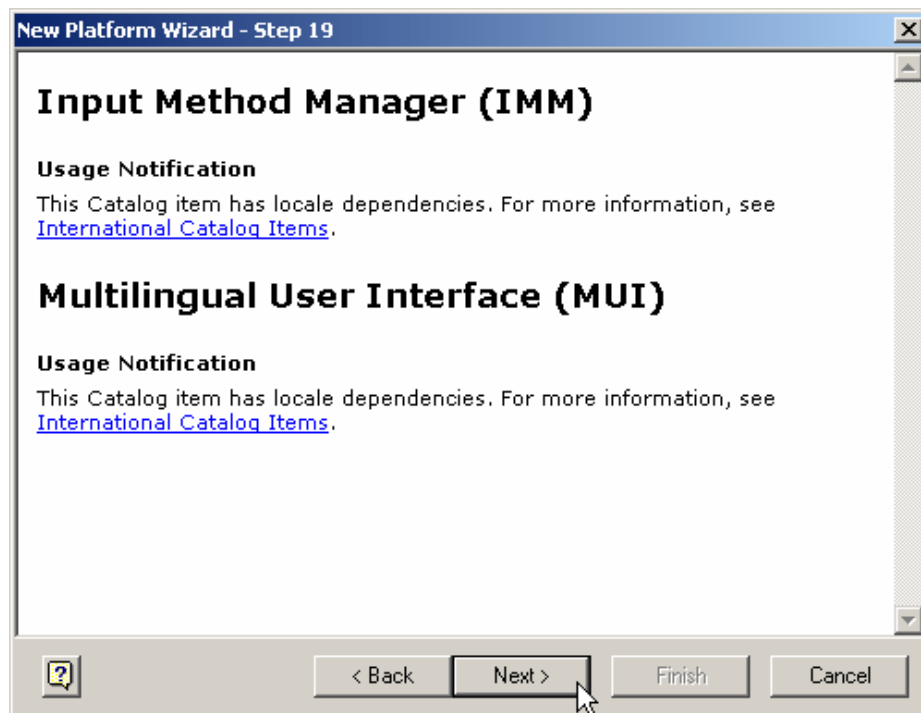


**Fig. 36.8** Captura de Platform Builder. Informe de errores de WCE



**Fig. 37.8** Captura de Platform Builder. Servicios de Telefonía sobre Voz IP

Por último, nos informa sobre algunas dependencias que existen con algún elemento elegido, ambos del soporte multilinguaje como podemos comprobar. No le damos más importancia y finalizamos el asistente.

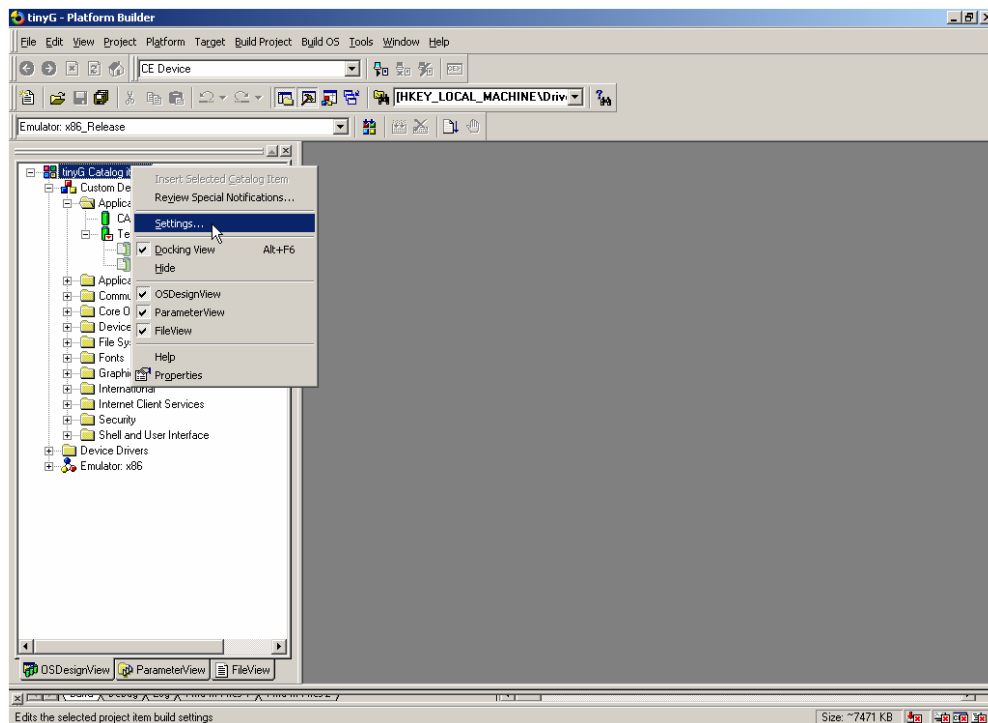


**Fig. 38.8** Captura de Platform Builder. Finalizando el asistente.

Una vez finalizado el asistente esperaremos como en el caso anterior a que se muestre la pantalla principal de Platform Builder y podremos ver como se creó un nuevo proyecto con el nombre que nosotros fijamos y con todos los elementos antes seleccionados en el asistente.

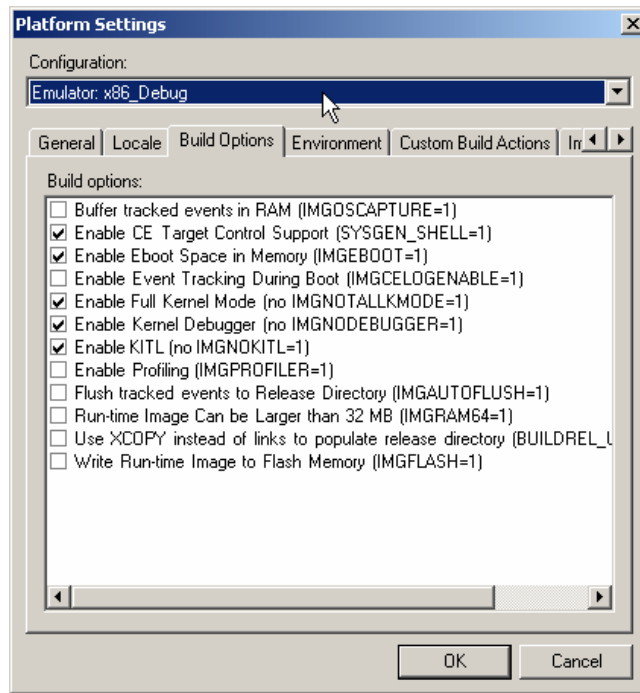
Además vemos que el dispositivo del que depende nuestra plataforma es un “Custom Device” justamente el tipo de dispositivo que elegimos para diseñar esta plataforma.

Como en cualquiera de las tres plataformas que vamos a crear en este proyecto accedemos al menú de “Settings”. Para acceder a las opciones de la plataforma solamente hacemos clic con el botón derecho sobre la raíz de la plataforma y en el menú desplegable hacemos clic sobre “Settings”. Esto lo podemos ver en la figura 39.



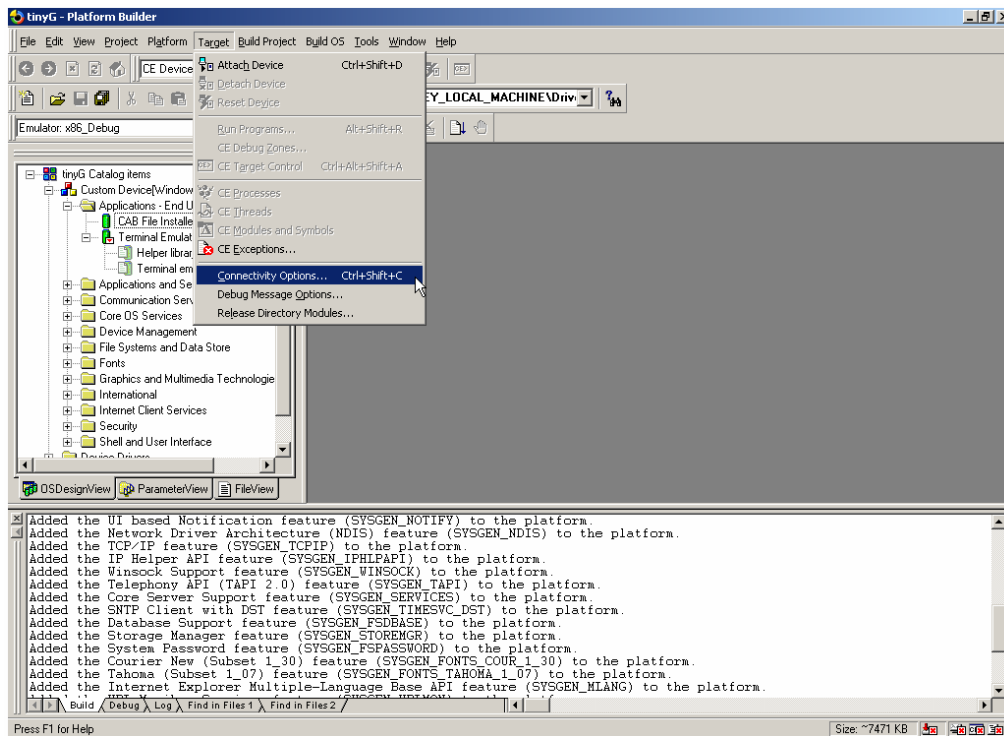
**Fig. 39.8** Captura de Platform Builder. Opciones de la plataforma

En la pestaña de “Build Options” las opciones que podemos ver en la siguiente figura, después presionamos “OK” y nos percatamos que la plataforma tarda un poco en cargar la pagina inicial de nuevo ya que con este cambio está modificando parámetros internos de compilación de la imagen.

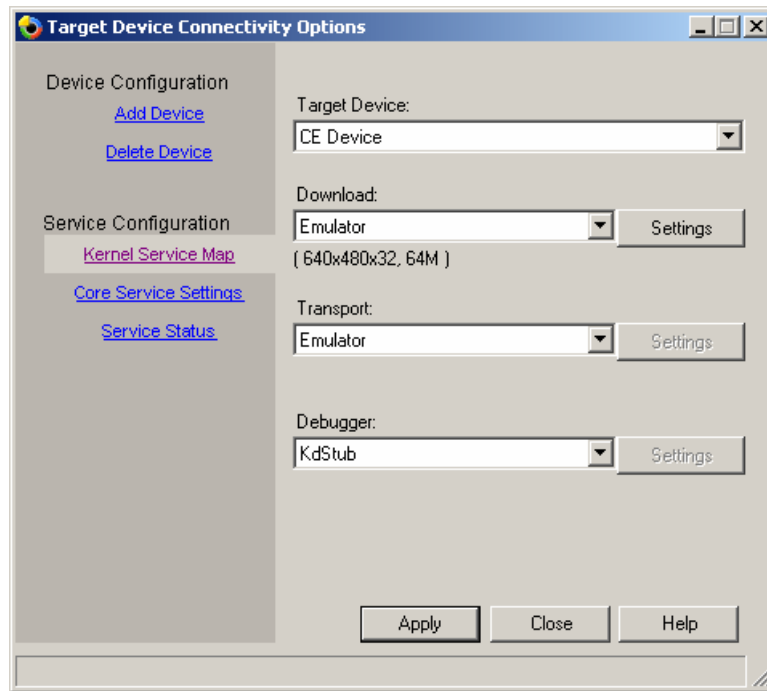


**Fig. 40.8** Captura de Platform Builder. Opciones de compilación de la imagen

Ahora debemos comprobar la configuración del dispositivo destino. En este caso será el emulador por lo que accedemos al menú superior “Target” -> “Connectivity Options”. En este menú debemos dejar las opciones desplegadas según la figura número 42.

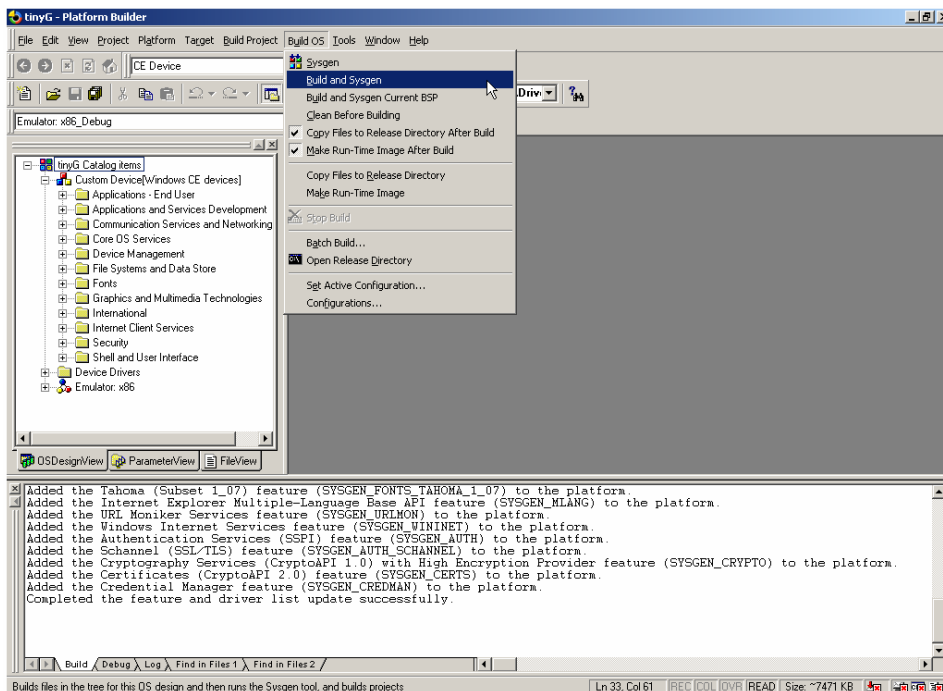


**Fig. 41.8** Captura de Platform Builder. Accedemos a opciones de conectividad

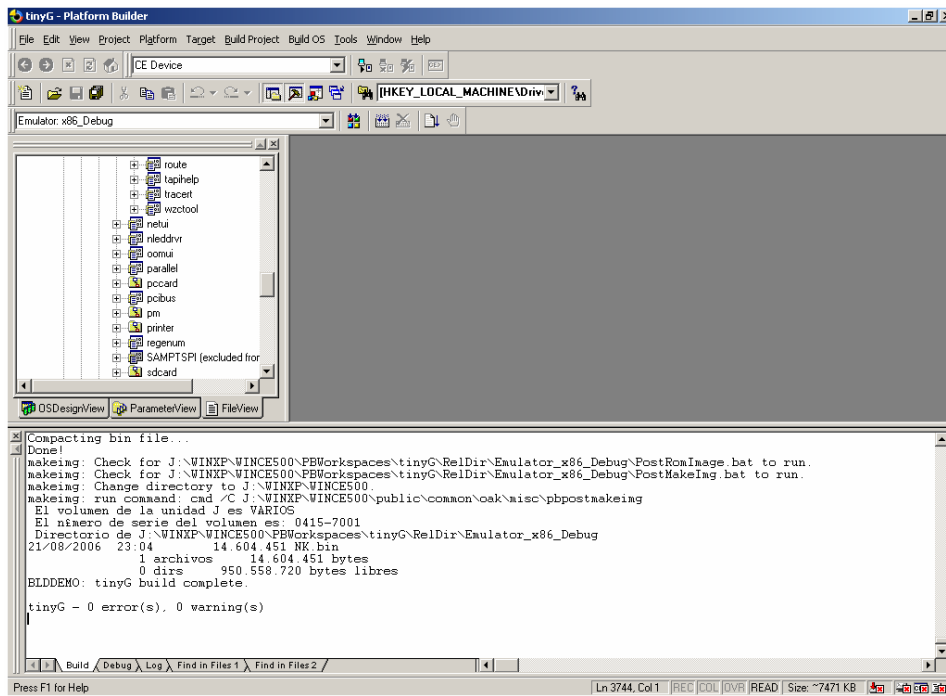


**Fig. 42.8** Captura de Platform Builder. Opciones de conectividad del dispositivo

En este momento ya podemos compilar la imagen que hemos creado y generar la imagen pertinente. Para esto nos vamos a “Build OS” -> “Build and Sysgen”. Al final de este proceso veremos los errores y/o warnings recibidos, en este caso ninguno.

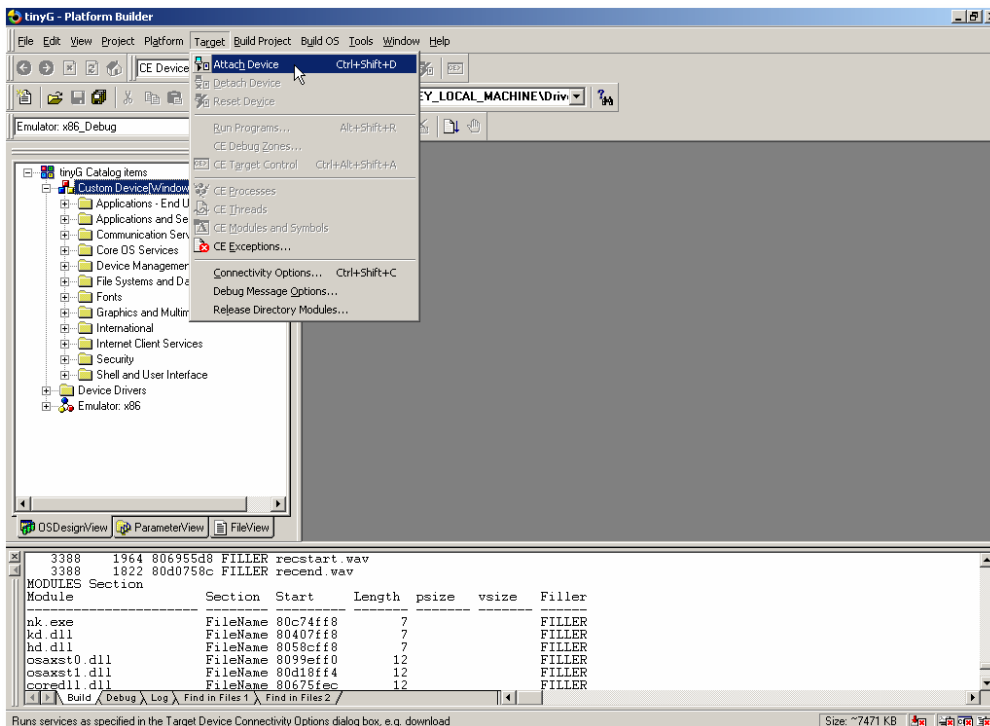


**Fig. 43.8** Captura de Platform Builder. Compilamos el proyecto y generamos la imagen

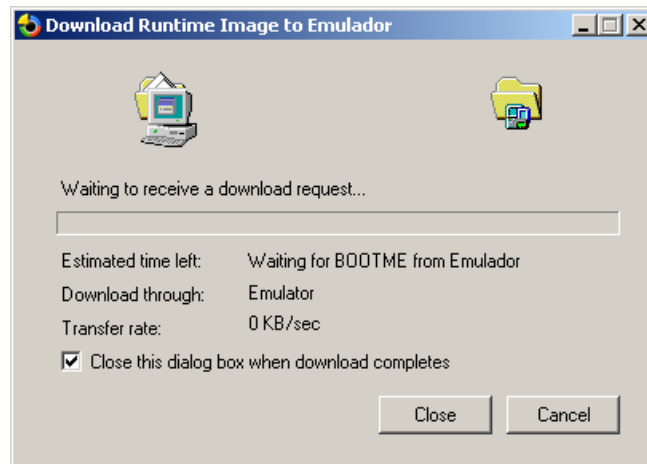


**Fig. 44.8** Captura de Platform Builder. Estado final de Platform Builder

El paso final sería descargar la imagen generada al dispositivo objetivo, en este caso el emulador que previamente habíamos configurado en el menú de opciones de conectividad. Para descargarla accedemos a “Target” -> “Attach Device”.

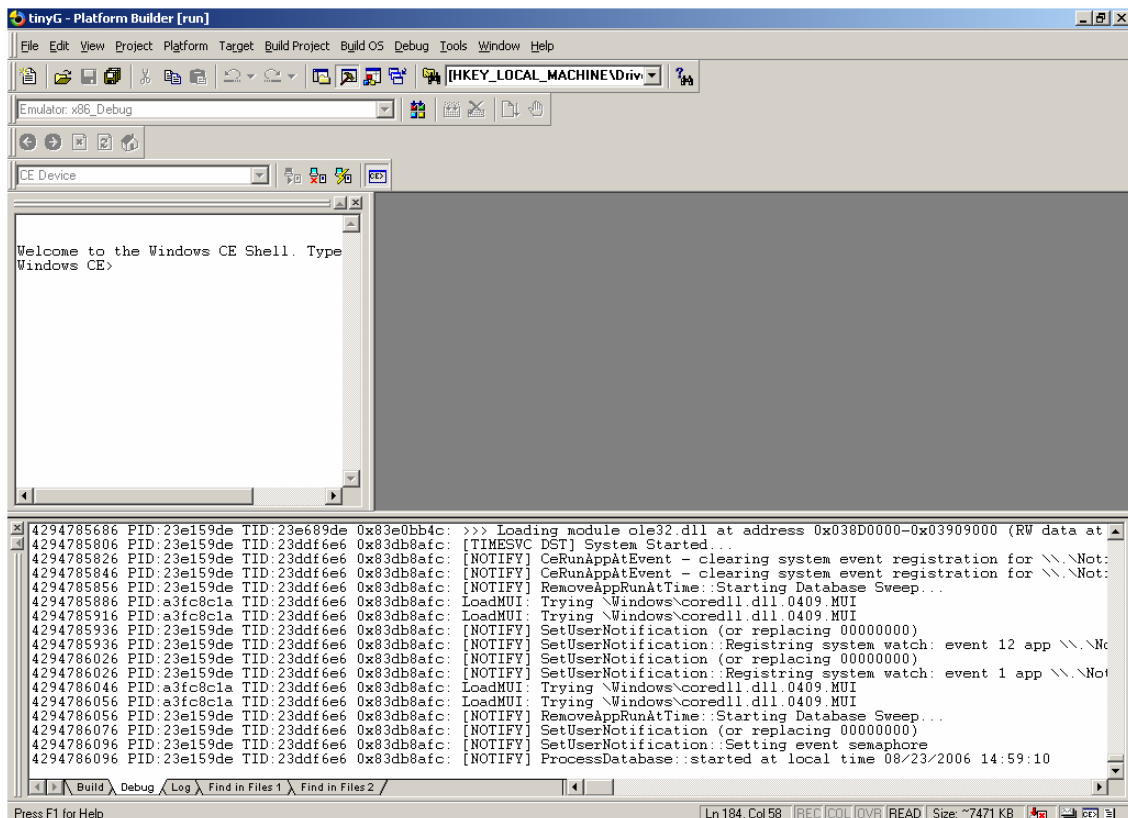


**Fig. 45.8** Captura de Platform Builder. Enlazando con el dispositivo



**Fig. 46.8** Captura de Platform Builder. Descargando la imagen

En la siguiente figura podemos observar el estado final del Platform Builder, que como en el caso de “Tiny Kernel” nos permite introducir comandos para visualizar procesos, threads, zonas de memoria ocupadas y otras características más particulares del sistema operativo.



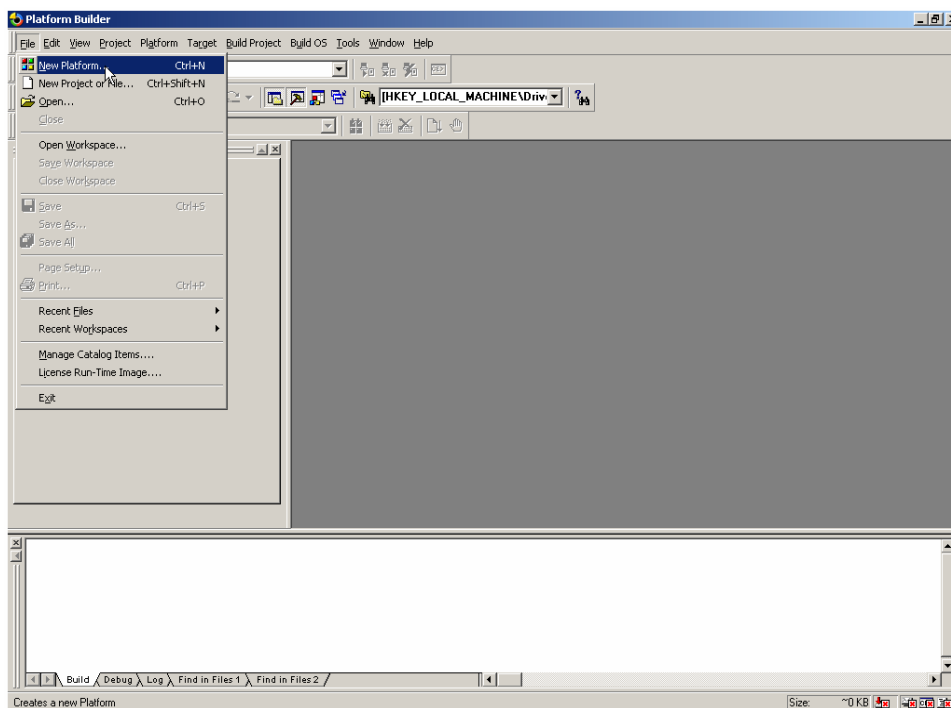
**Fig. 47.8** Captura de Platform Builder. Estado final de Platform Builder



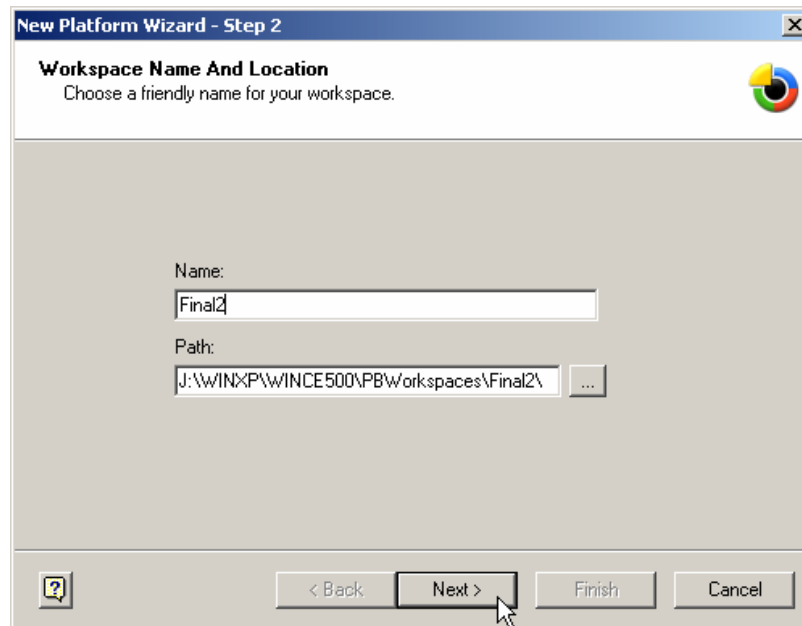
Aparte de esto, tendremos una ventana abierta del emulador de Windows CE ejecutando la imagen actual siempre y cuando lo tengamos instalado. No nos ocuparemos de visualizar la imagen sobre el emulador puesto que lo haremos en el Capítulo 4, donde podremos ver las pruebas de todas las imágenes del capítulo de Diseño.

### 8.1.3. Imagen basada en BSP VIA EPIA MII

Como en los casos anteriores procedemos a crear una nueva plataforma accediendo al menú superior “File” -> “New Platform”. Posteriormente introducimos el nombre del proyecto y la ruta donde deseamos que se encuentre.

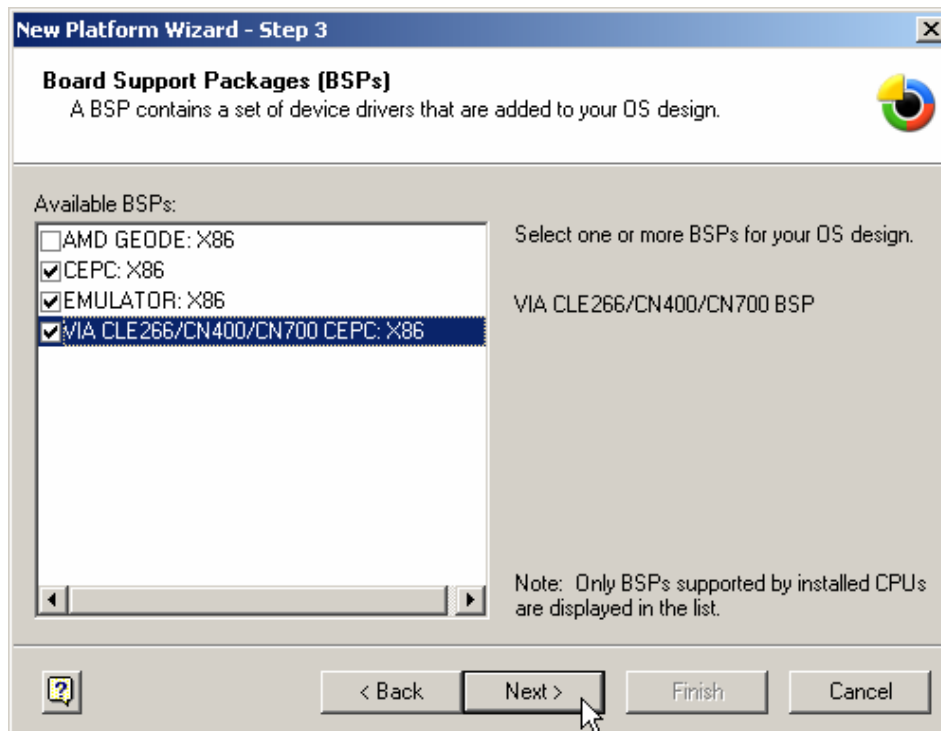


**Fig. 48.8** Captura de Platform Builder. Iniciando el asistente



**Fig. 49.8** Captura de Platform Builder. Nombre y ruta del proyecto

Ahora debemos seleccionar qué BSP estarán disponibles para dar compatibilidad a la plataforma que vamos a generar. VIA CLE266/CN400/CN700 CEPC es la propia de nuestra placa, CEPC la incluyo para dar conectividad entre la placa y el Platform Builder y por último el emulador por si queremos hacer alguna prueba sobre él.

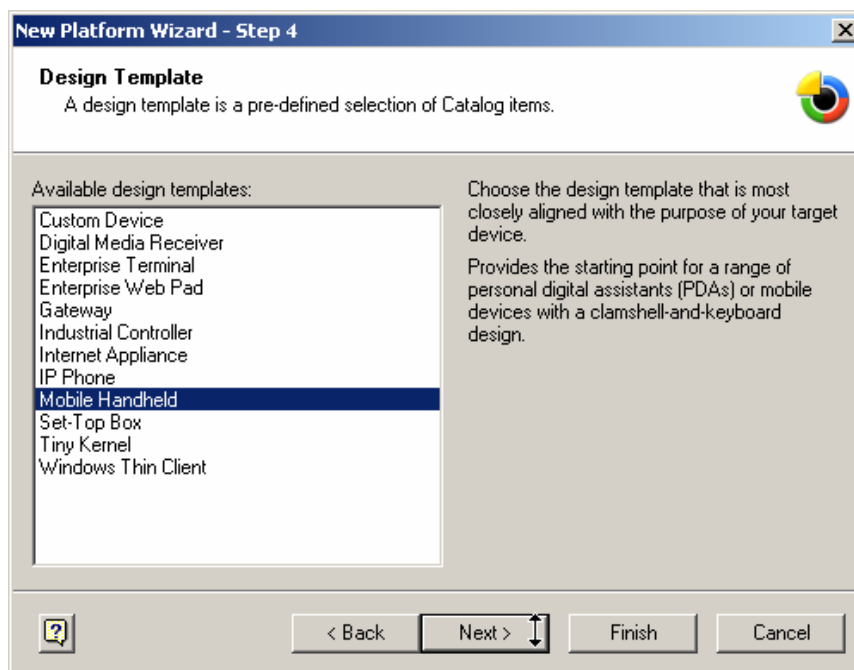


**Fig. 50.8** Captura de Platform Builder. BSP disponibles

Llegado a este punto debemos decidir para qué tipo de dispositivo estamos realizando esta plataforma.

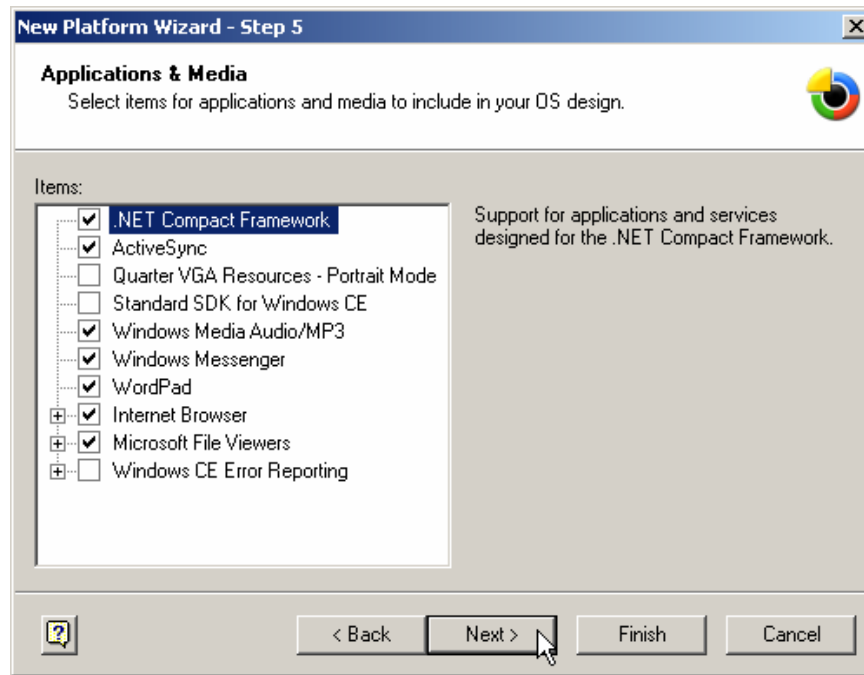
Después de ver los módulos y servicios de cada uno de los tipos que nos ofrece el asistente llegué a una conclusión. Debido al gran número de elementos que debía añadir para componer una imagen y que existía un tipo que reunía casi todas las características deseadas opté por él.

Este dispositivo es el “Mobile Handheld” o un teléfono inteligente (SmartPhone) pero algo más avanzado ya que el sistema operativo es mucho más completo en funcionalidades. Gracias a esta elección también se unía la característica de disponer de teclado en pantalla para facilitar la introducción de datos alfanuméricos.



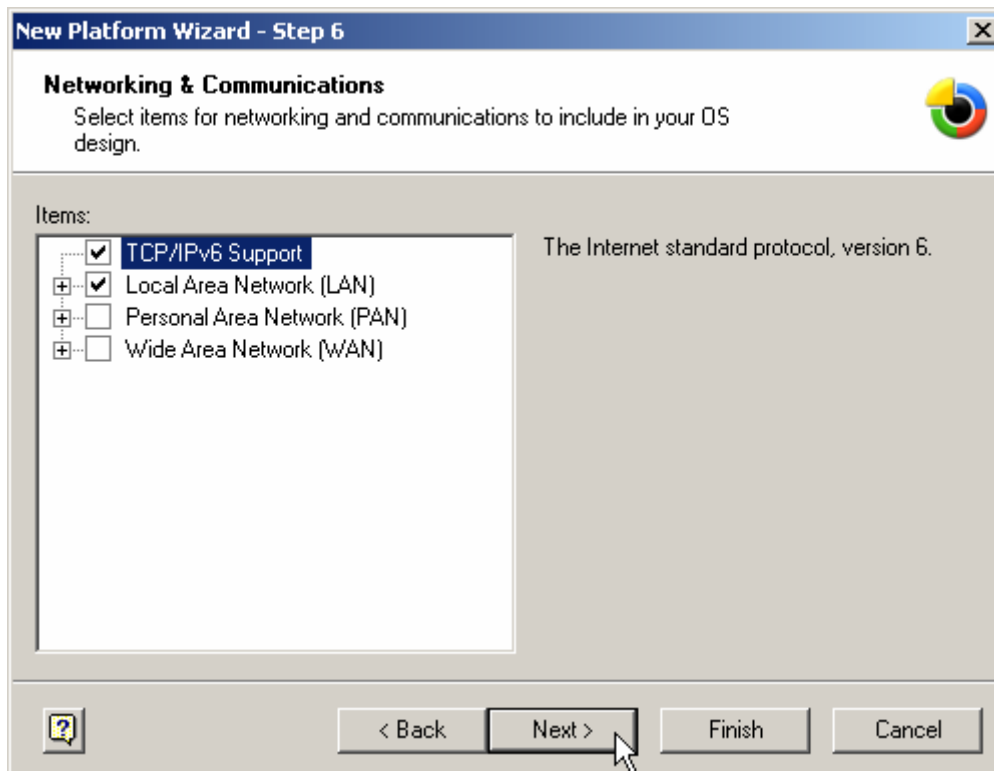
**Fig. 51.8** Captura de Platform Builder. Elegimos tipo de dispositivo

Después nos saldrán diversas ventanas con agrupaciones de elementos que podemos añadir a nuestra plataforma, exactamente igual que en el caso anterior con la imagen de “Tiny Graphics”. En la ventana de “Applications & Media” elegimos “.NET Compact Framework”, “ActiveSync”, “Windows Media Audio/MP3”, “Windows Messenger”, “WordPad”, “Internet Browser” y “Microsoft File Viewers”.



**Fig. 52.8** Captura de Platform Builder. Aplicaciones y Medios

En la siguiente agrupación llamada “Networking & Communications” marcamos “TCP/IPv6” y “Local Area Network”.



**Fig. 53.8** Captura de Platform Builder. Redes y comunicaciones

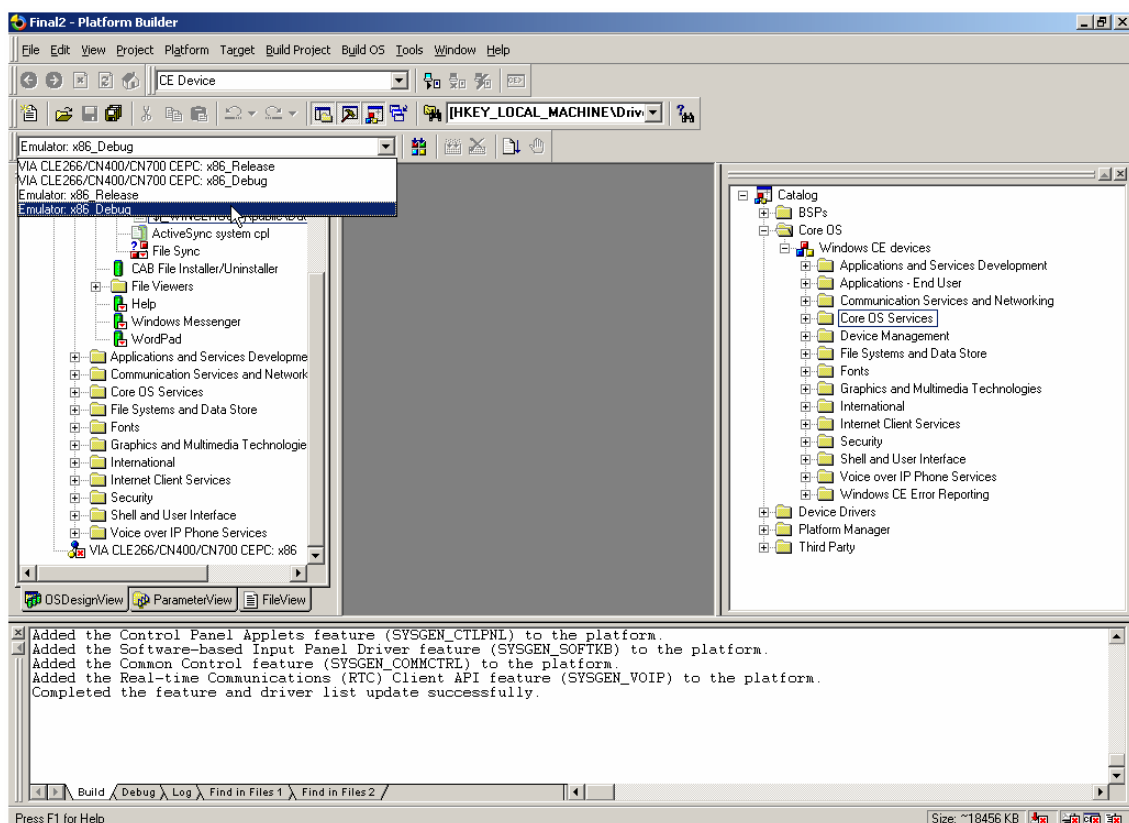
Con esto ya acabaría el asistente, sólo deberíamos finalizar y se crea el catálogo con todos los elementos seleccionados.

Posteriormente, haremos los pasos típicos para acabar de configurar nuestra plataforma. Por un lado debemos modificar las características de compilación de la imagen del sistema operativo. Por otro lado debemos introducir aquellos elementos extras que necesitamos que estén disponibles en nuestro sistema operativo, por ejemplo, Compact Framework 2.0 o soporte para red de área personal, etc.

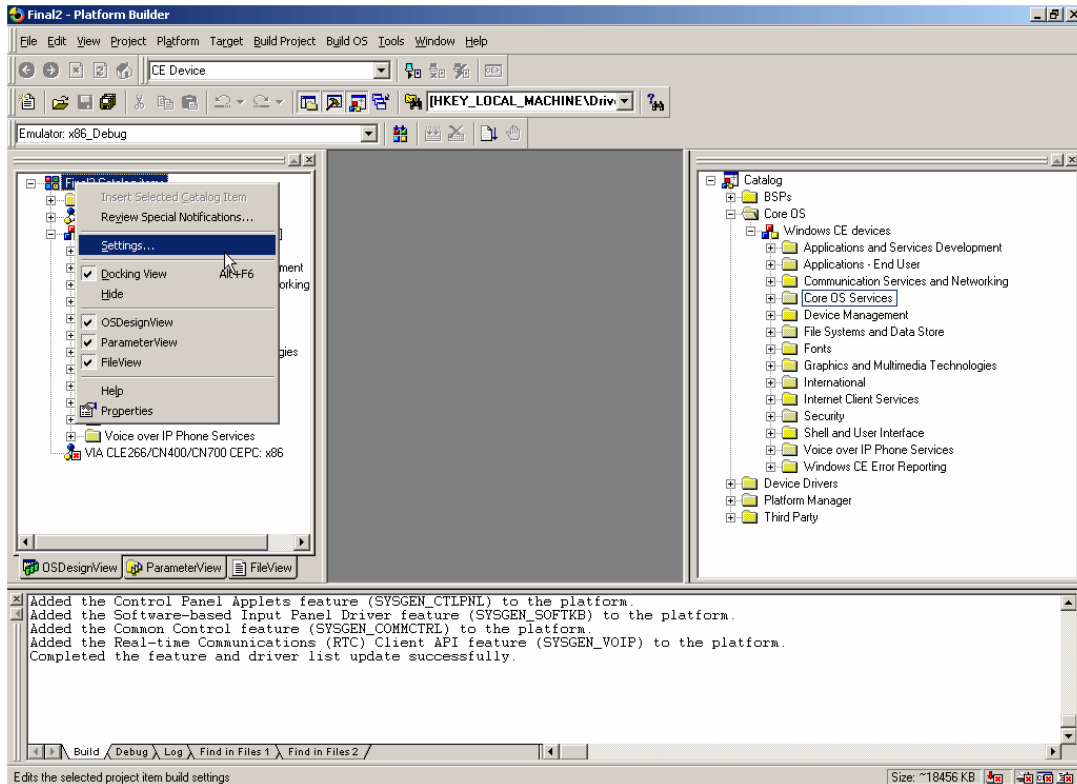
Como decimos anteriormente este tipo de dispositivo ya creaba una imagen bastante ajustada a nuestras necesidades pero como es comprensible no estrictamente la prevista. Por este motivo incluiremos estos elementos.

A continuación podemos ver en primera instancia las modificaciones pertinentes en las opciones de compilación de la imagen del sistema. Como ya sabemos de los anteriores ejemplos primero seleccionamos el BSP con el que queremos compilar el proyecto, en este caso comenzaremos por el emulador.

Después hacemos clic con botón derecho sobre la raíz del catálogo y clic sobre "Settings", esto lo podemos observar en la figura 55.

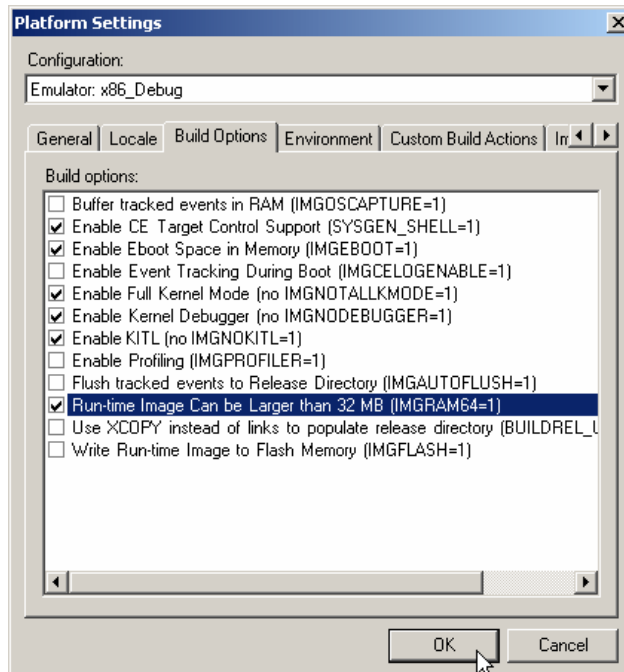


**Fig. 54.8** Captura de Platform Builder. Seleccionamos BSP

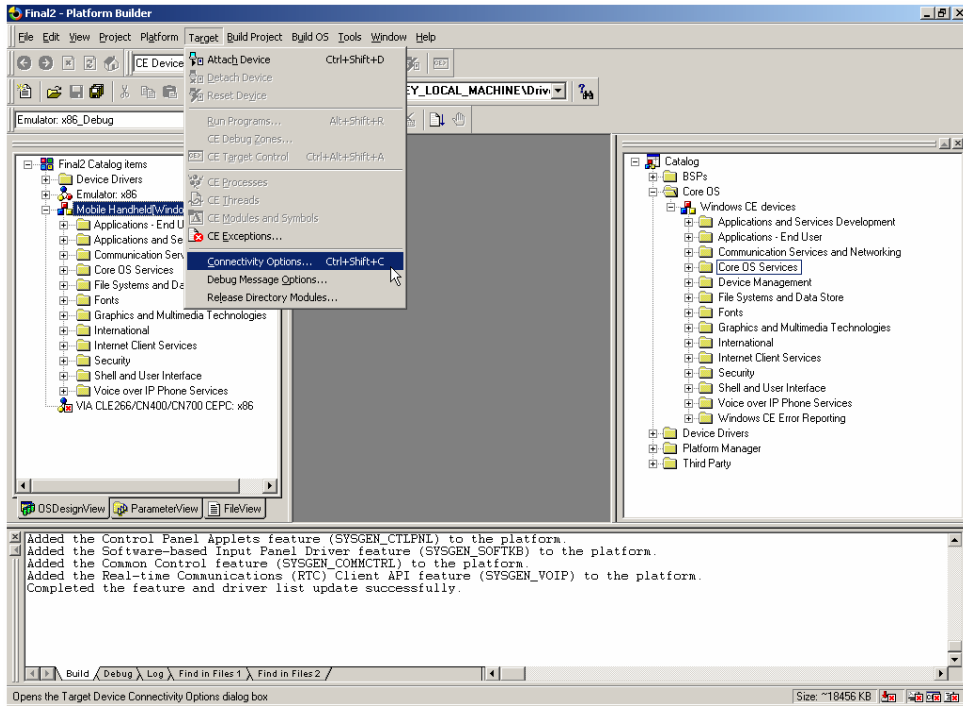


**Fig. 55.8** Captura de Platform Builder. Opciones de la plataforma

En la pestaña de “Build Options” marcaremos las siguientes que se pueden observar en la figura. Como paso posterior nos dirigimos a “Target” -> “Connectivity Options”.

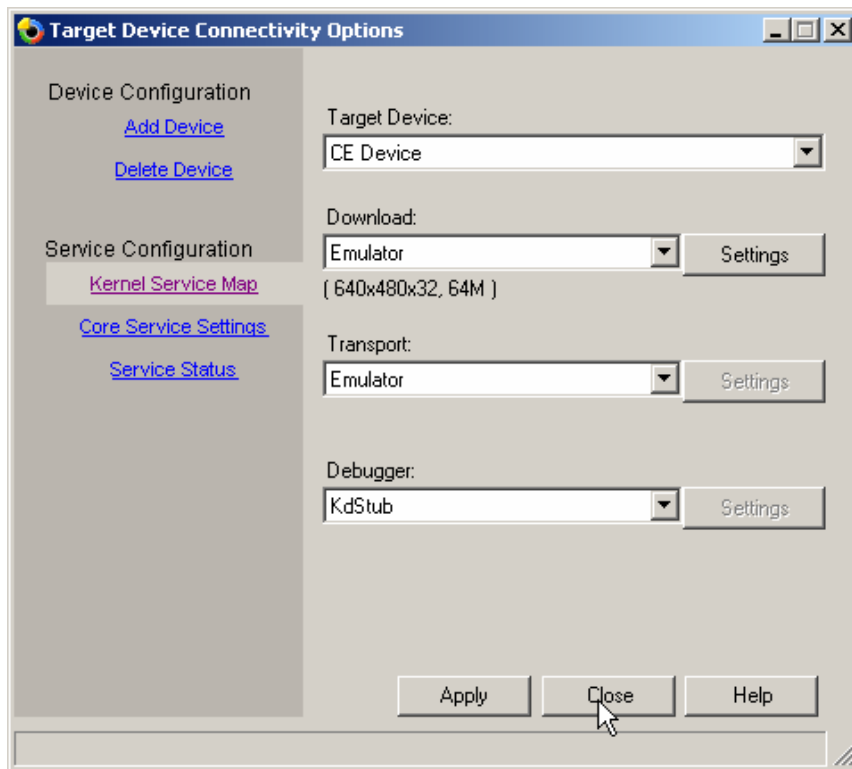


**Fig. 56.8** Captura de Platform Builder. Opciones de compilación



**Fig. 57.8** Captura de Platform Builder. Opciones de conectividad

Comprobamos que en las opciones de conectividad presentamos la siguiente configuración para trabajar sobre el emulador.



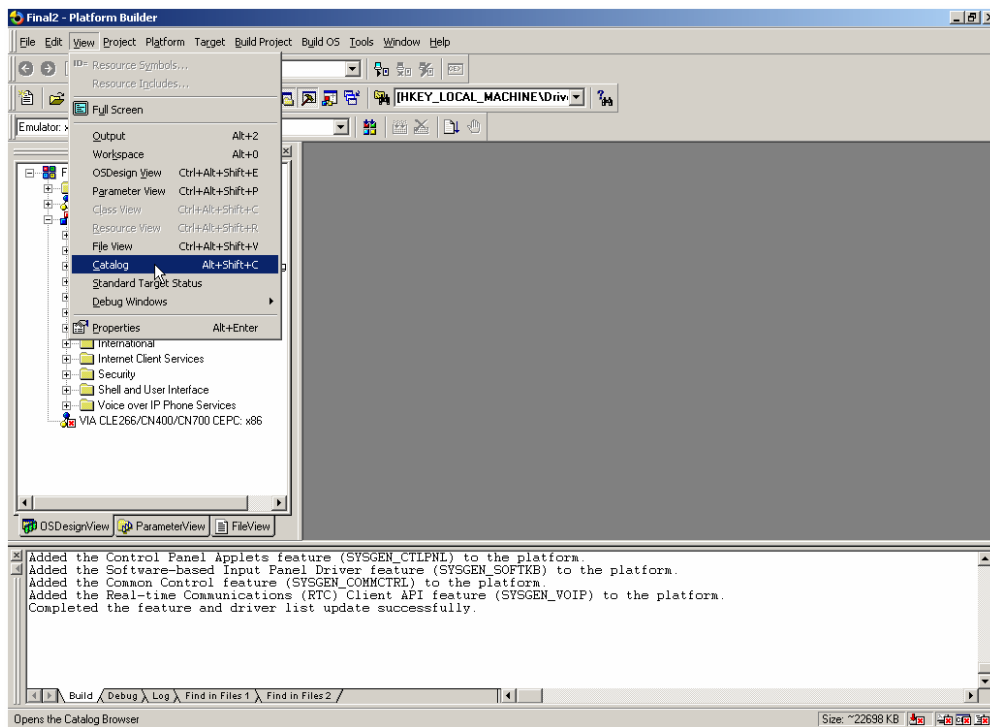
**Fig. 58.8** Captura de Platform Builder. Estado final de Platform Builder

En la siguiente ventana hacemos visible la ventana del catálogo disponible para añadir los elementos que necesitamos. En este caso, aseguraremos que poseemos los servicios necesarios para el proyecto común.

Recordemos que los servicios que debemos brindar eran:

- Protocolo TCP/IP
- Compact Framework 2.0
- Puerto serie
- Otros: Directshow, codecs...

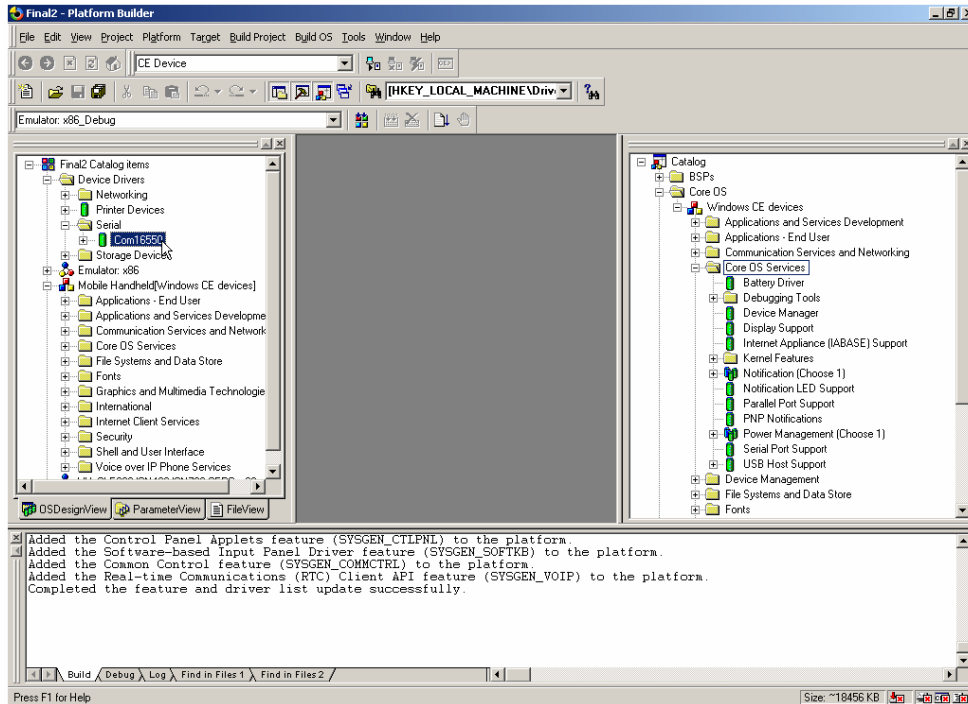
A continuación veremos como añadimos a nuestra plataforma estos elementos que precisamos. No obstante algunos podemos encontrarlos ya añadidos por las características del dispositivo añadido.



**Fig. 59.8** Captura de Platform Builder. Abrimos la ventana de “Catalog”

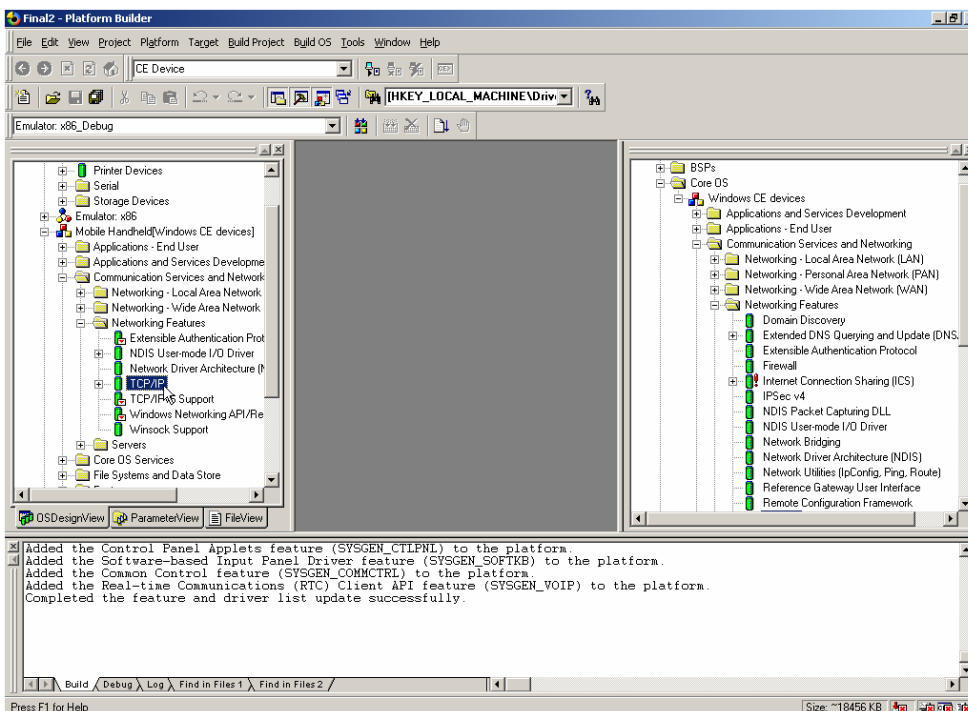
Podemos comprobar que el puerto serie es uno de los elementos que ya se encuentran disponibles en la actual plataforma por lo que podemos prescindir de incluirla puesto que sería inútil.





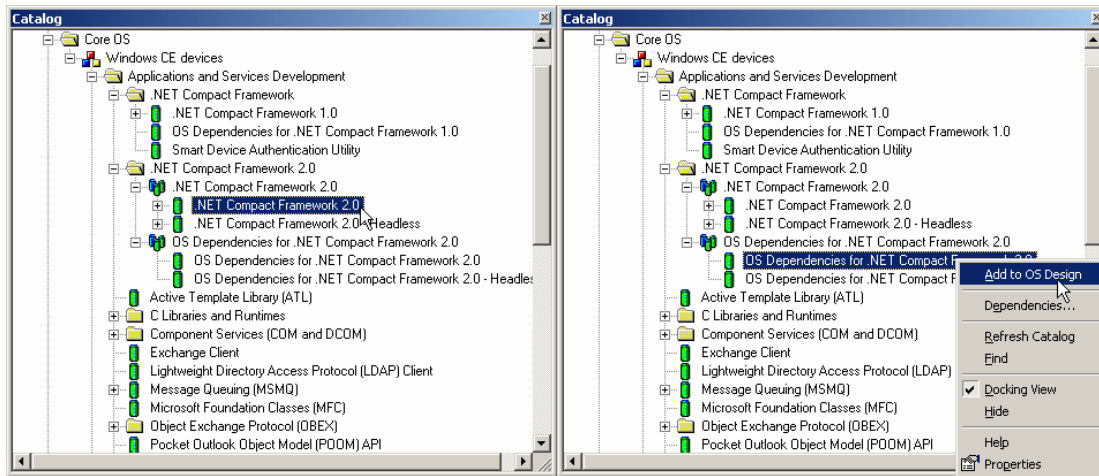
**Fig. 60.8** Captura de Platform Builder. Soporte para puerto serie

El protocolo TCP/IP es necesario para brindar conectividad al dispositivo, característica que se hace imprescindible para el uso del dispositivo. Pero en este caso ya estaba incluido gracias al asistente, y no sólo eso sino que también dando soporte a su nueva versión TCP/IP v6.



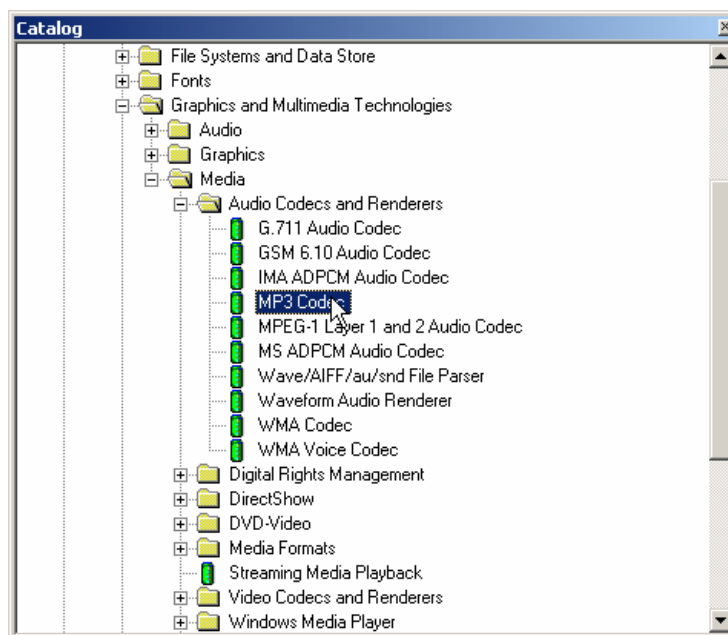
**Fig. 61.8** Captura de Platform Builder. Protocolo TCP/IP

En este momento añadimos los elementos pertenecientes a CF 2.0. Haciendo clic con el botón derecho y luego clic en “Add to OS Dessign” se incluye.



**Fig. 62.8** Captura de Platform Builder. Añadiendo Compact Framework 2.0

Debemos añadir los codecs de audio entre ellos figurarán “MP3 Codec”, “MPEG-1 Layer 1 and Layer 2 Audio Codec”, “MS ADPCM Audio Codec”, “Wave/AIFF/au/snd File Parser”, “WMA Codec” y “WMA Voice Codec”.



**Fig. 63.8** Captura de Platform Builder. Codecs de audio

En la siguiente ventana podemos ver como añadimos el núcleo de DirectShow, también incluiremos “ACM Wrapper Filter”, “DirectShow Display” y “DMO Wrapper filter”.

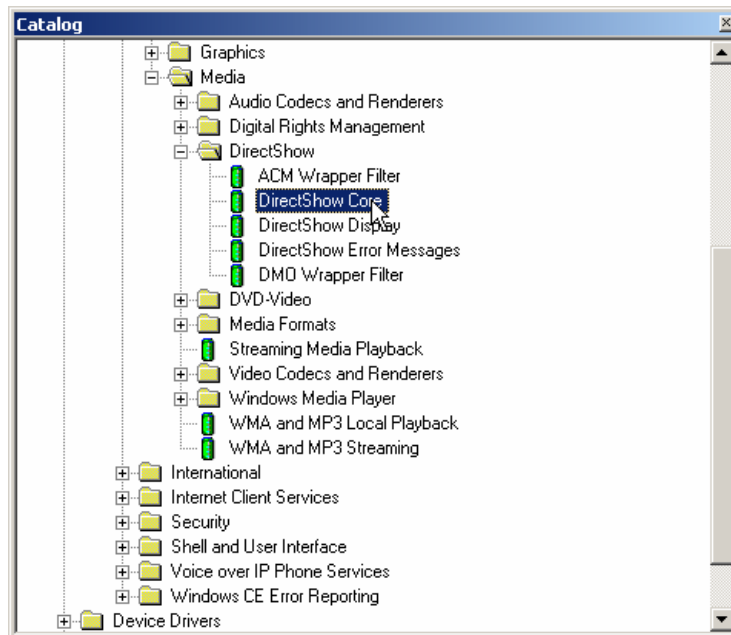


Fig. 64.8 Captura de Platform Builder. DirectShow

Añadimos a nuestra plataforma el soporte para reproducción de DVD.

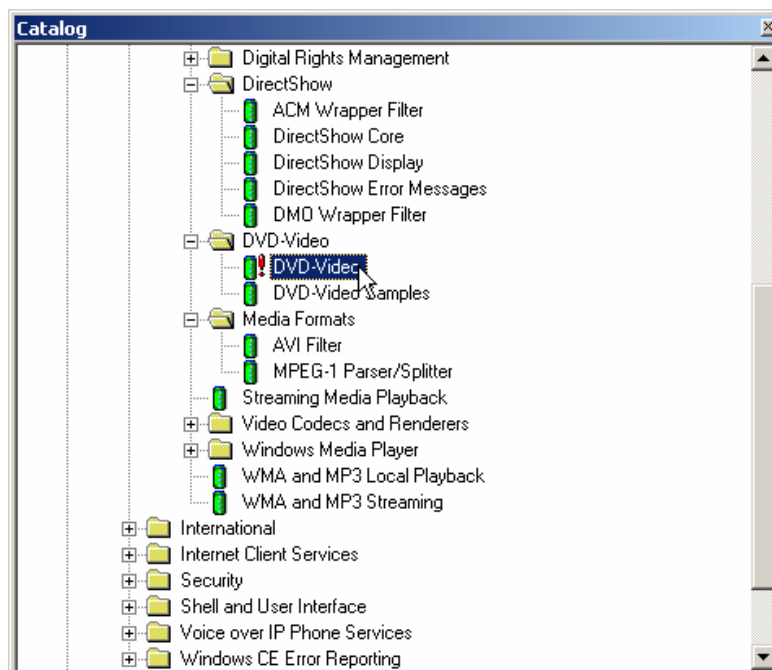
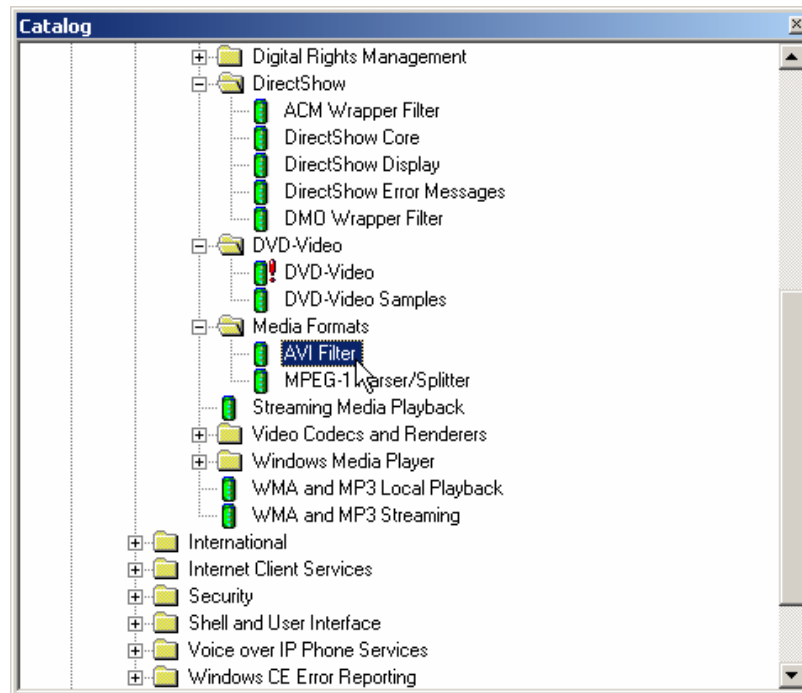


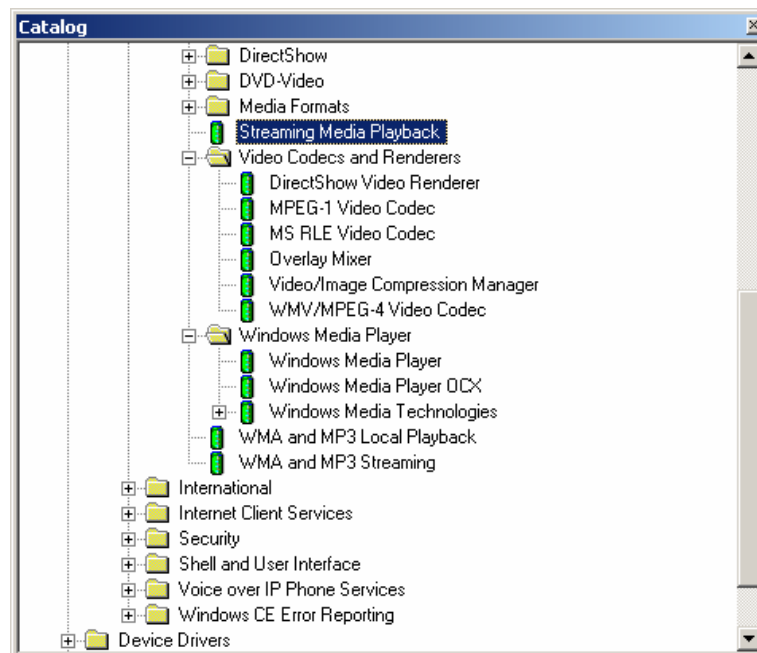
Fig. 65.8 Captura de Platform Builder. Reproducción DVD

Añadimos los elementos “AVI Filter” y “MPEG-1 Parser/Splitter”.



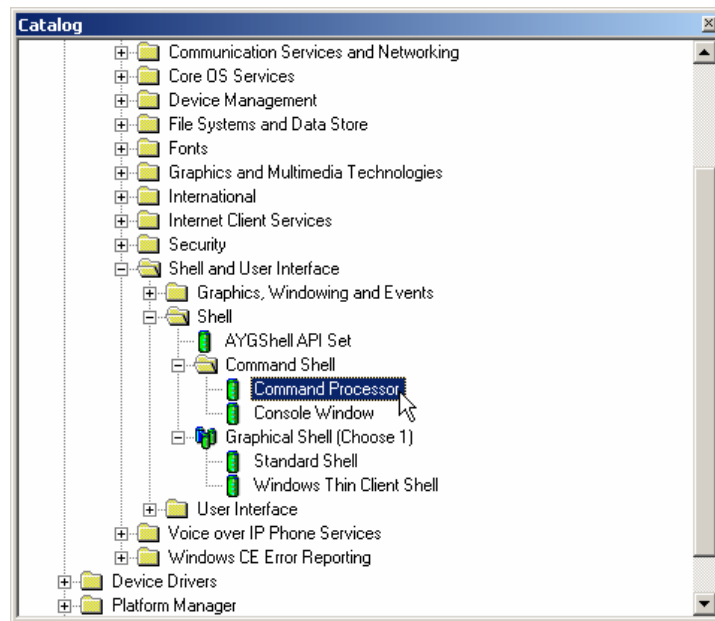
**Fig. 66.8** Captura de Platform Builder. Filtros de medios

Permitimos el soporte para la reproducción de medios mediante streaming. Asimismo los elementos “Directshow Video Renderer”, “MPEG-1 Video Codec”, “MS RLE Video Codec”, “Video/Image Compressor Manager”, “WMV/MPEG-4 Video Codec”, “WMA and MP3 Local Playback” y “WMA and MP3 Streaming”.



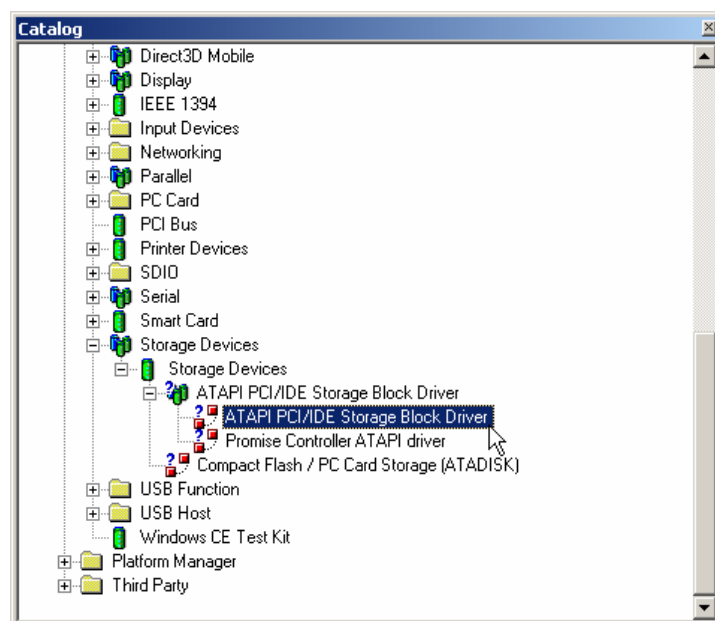
**Fig. 67.8** Captura de Platform Builder. Streaming y codecs de video

Añadimos a nuestro sistema operativo los siguientes elementos “Command Processor”, “Console Window”, “AYG Shell API Set” y “Standard Shell”.



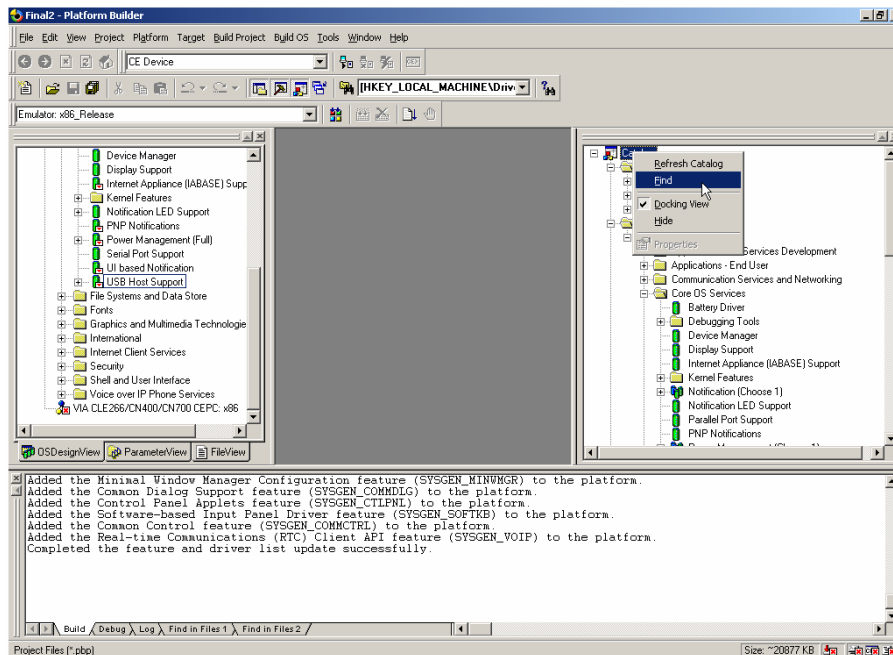
**Fig. 68.8** Captura de Platform Builder. Agregamos Terminal de comandos

Agregamos también compatibilidad con unidades IDE o bien con tarjetas PCI además del soporte a Compact Flash. Para esto agregamos “ATAPI PCI/IDE Storage Block” y “Compact Flash/PC Card Storage (ATADISK)”.

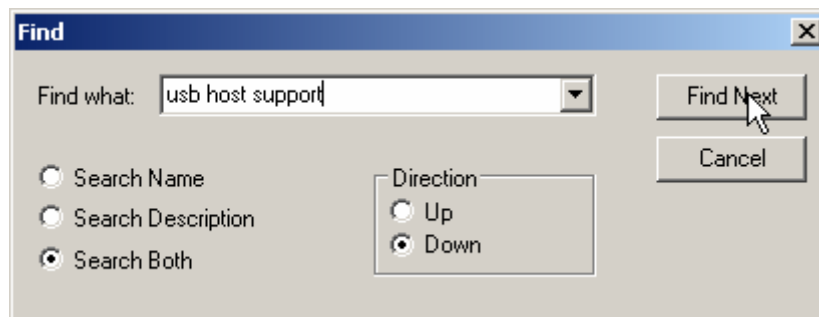


**Fig. 69.8** Captura de Platform Builder. Compatibilidad IDE/PCI y CF/PC Card

Si queremos encontrar algún elemento hacemos clic con el botón derecho sobre el catálogo y clic sobre “Find”. En este caso buscamos “USB Host Support”.

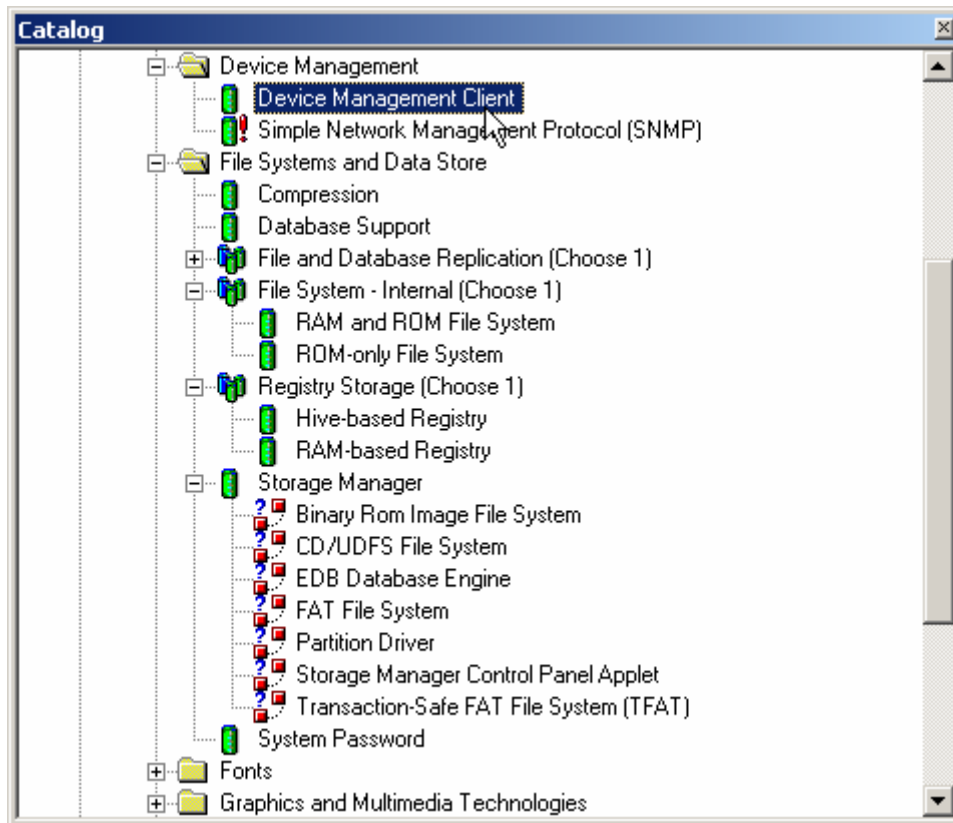


**Fig. 70.8** Captura de Platform Builder. Encontrando un servicio



**Fig. 71.8** Captura de Platform Builder. Ventana de búsqueda

Para finalizar incluimos el cliente de gestión de dispositivos, “Device Management Client”. También agregamos el elemento “Compression”, Hive-based Registry” y “RAM and ROM File System”, además de todos los elementos de la agrupación “Storage Manager”.



**Fig. 72.8** Captura de Platform Builder. Gestión de dispositivos, gestión de almacenamiento y sistema de archivos

A partir de este momento tendríamos que tener todos los elementos citados en el apartado 3.4.2. De no ser así, en este momento nos preocuparíamos de agregar los elementos que nos faltan suponiendo que queremos tener la misma plataforma que en este diseño o eliminar los elementos que no quisiéramos.

Hacemos clic en “Sysgen” provocando la compilación y generación del sistema. En la figura 75 podemos ver como después de este paso debemos crear la imagen que correrá en el dispositivo (“Build OS” -> “Make run-time image”).

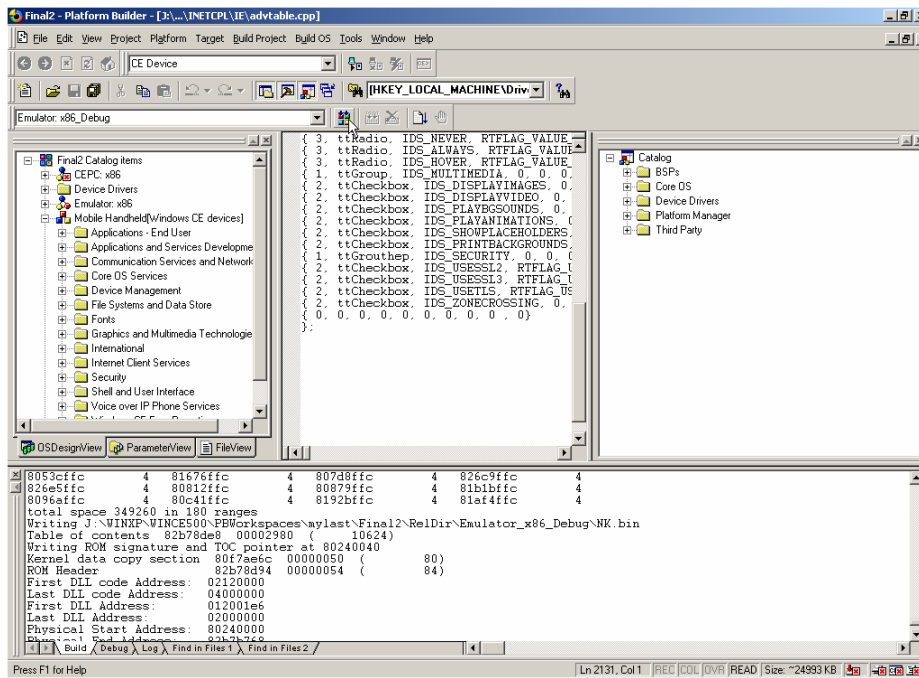


Fig. 73.8 Captura de Platform Builder. Generar sistema operativo

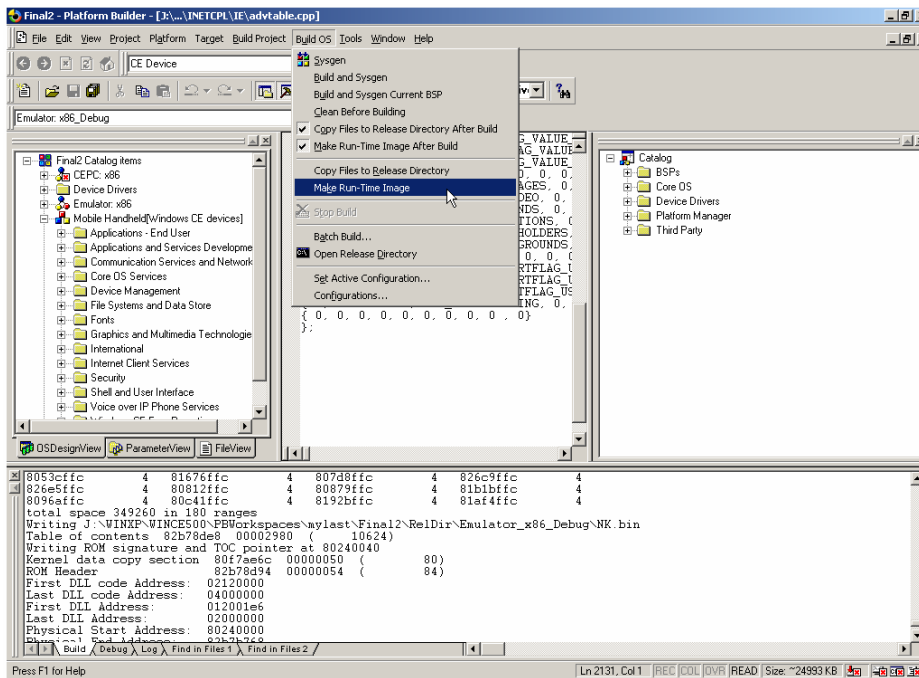
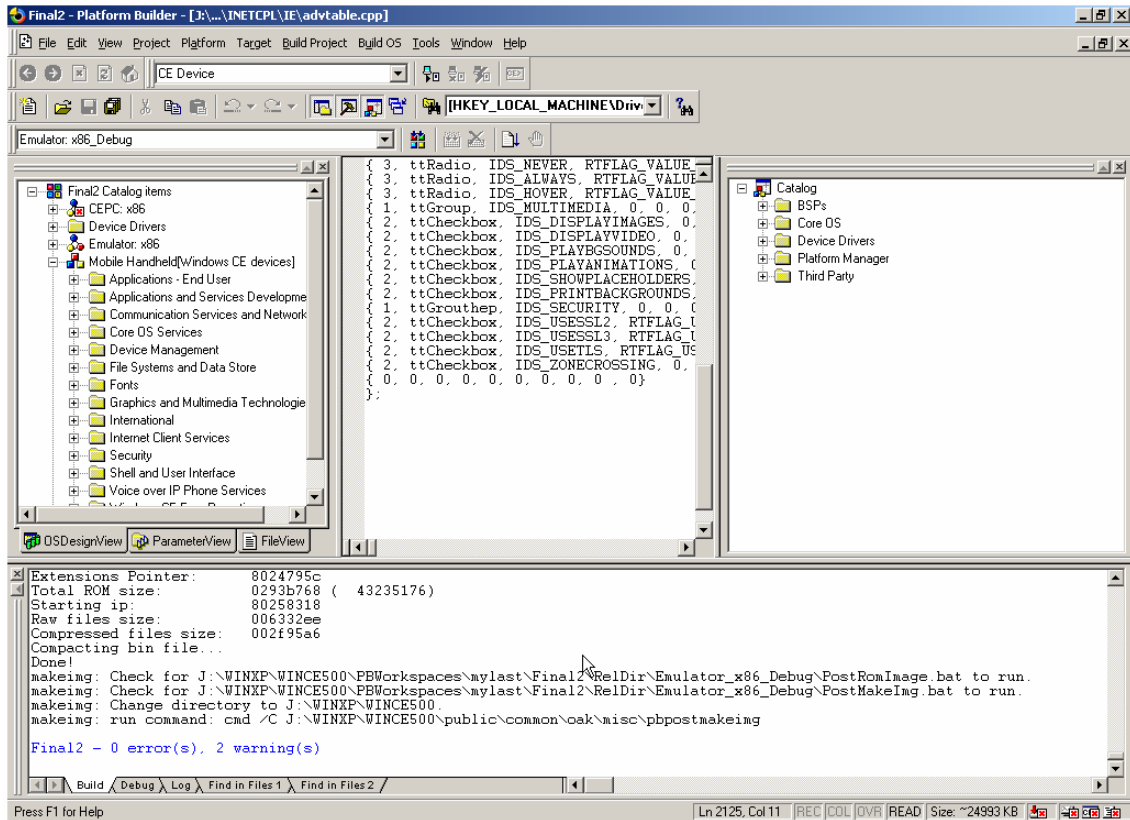


Fig. 74.8 Captura de Platform Builder. Crear imagen \*.bin

Finalmente nos quedamos en la ventana principal de Platform Builder donde podemos ver los errores y warnings producidos. En este caso se han producido 2 warnings pero debidos a ficheros \*.rel de versiones obsoletas. Normalmente, por lo que tengo entendido sucede cuando no hay versiones más actuales de estos ficheros acordes con otras actualizaciones hechas en PB.



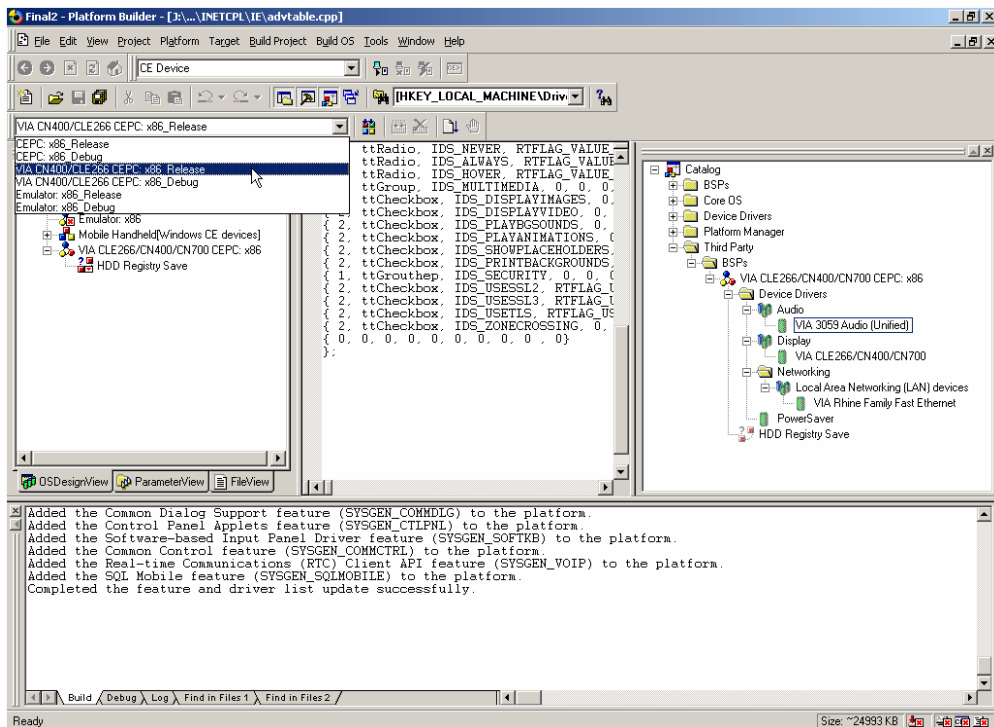


**Fig. 75.8** Captura de Platform Builder. Estado final de Platform Builder

Una vez llegado a este punto dispondríamos de una imagen \*.bin para poder arrancarla en el emulador puesto que el BSP utilizado para generar este sistema operativo era el del emulador. Recordemos que habíamos preparado las opciones de conectividad para que el dispositivo objetivo fuese el emulador.

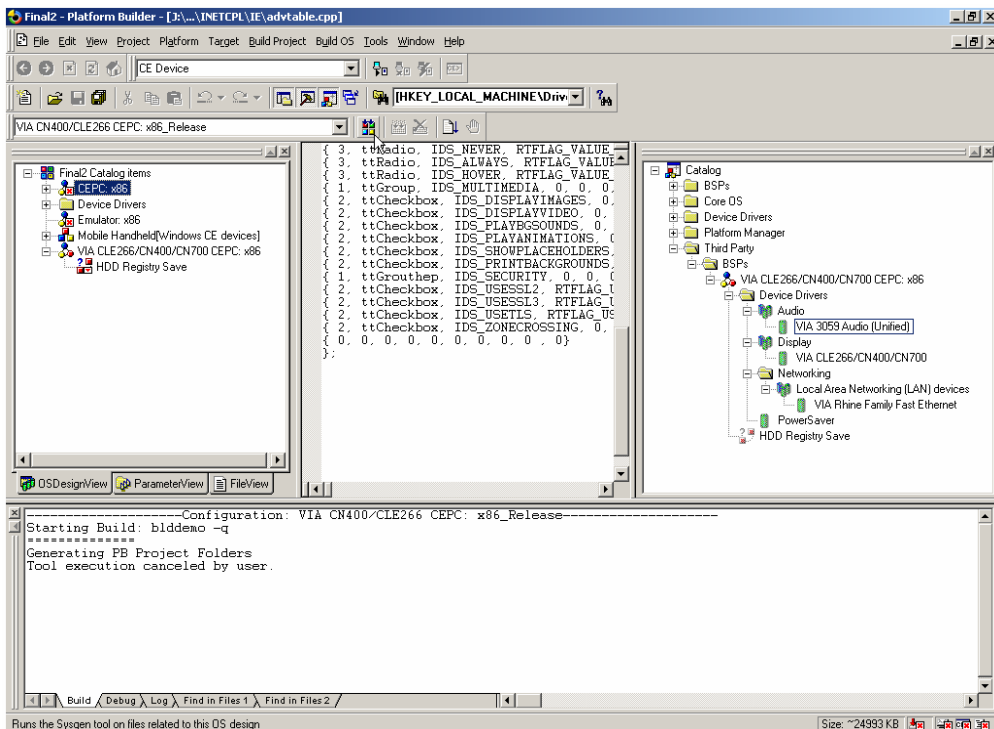
Como ya carece de interés probarlo sobre el emulador porque ya lo hemos experimentado procederemos a compilar y generar el sistema operativo utilizando el BSP de VIA CN400/CLE266 CEPC :x86 Release. El sistema generado con este BSP será el que finalmente correrá sobre el dispositivo.

Para esto seleccionamos el BSP correspondiente en el textbox desplegable que hay arriba a la derecha como podemos observar en la siguiente figura.



**Fig. 76.8** Captura de Platform Builder. Seleccionamos BSP de VIA

Ahora hacemos clic sobre el botón de “Sysgen” para comenzar la compilación y generación del sistema.



**Fig. 77.8** Captura de Platform Builder. Generar sistema operativo

Podemos ver en el estado final de Platform Builder que nos informa de los errores y warnings en la generación del sistema. Esta vez no ha habido ningún error ni aviso pero BLDEMO detectó alguno que después seguramente se verá reflejado en la generación de la imagen \*.bin.

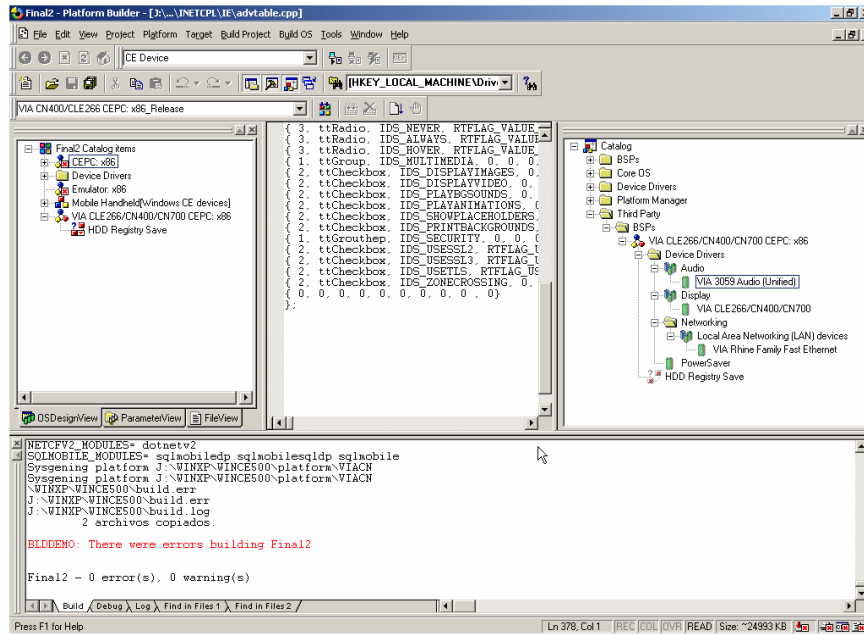


Fig. 78.8 Captura de Platform Builder. Estado después de generar

Hacemos clic en “Build OS” -> “Copy files to Release Directory” para copiar todos los archivos del sistema operativo generado anteriormente al directorio de la versión release. Después haremos clic en “Make Run-time Image”.

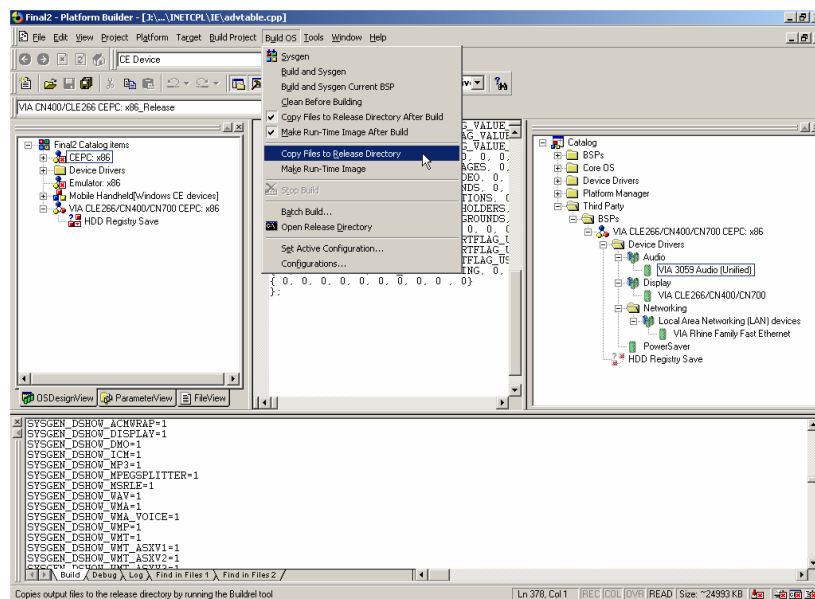
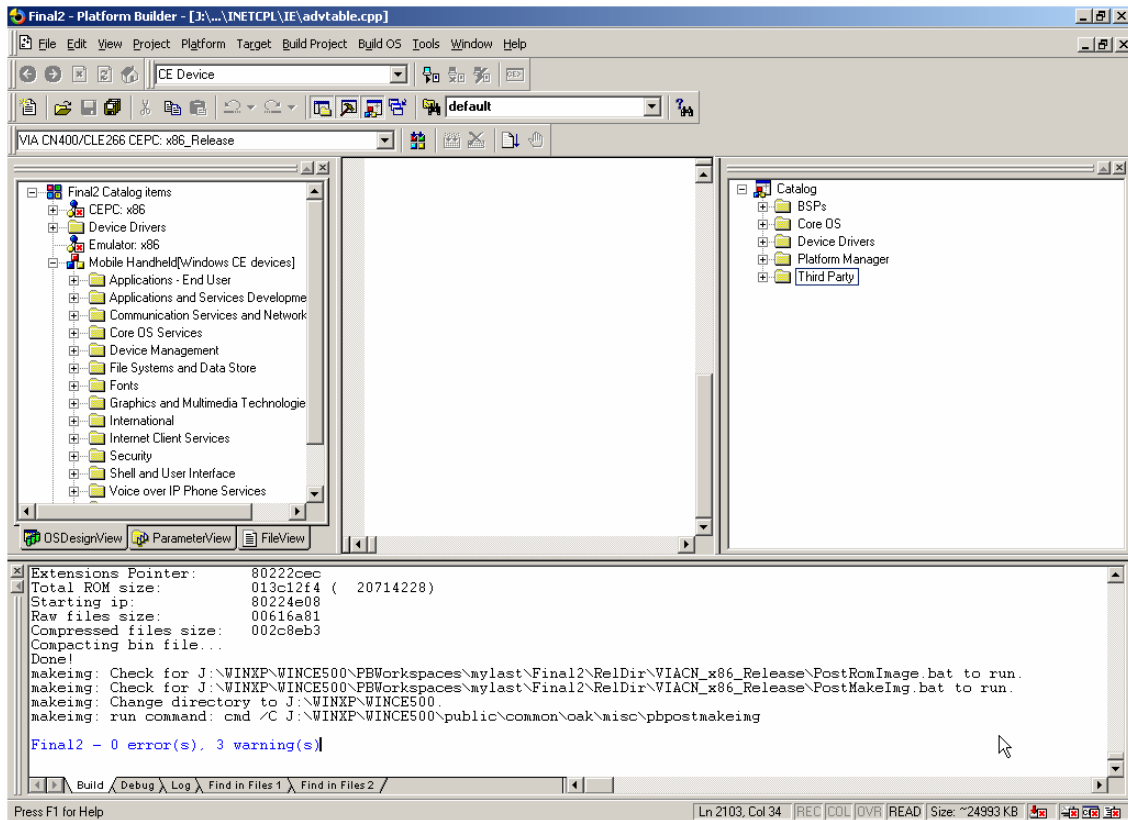


Fig. 79.8 Captura de Platform Builder. Copiar ficheros al directorio release

Por último, en el estado final del Platform Builder podemos observar, como siempre, los avisos y errores producto de la generación de la imagen \*.bin generada. Como podemos ver se han producido 3 warnings o avisos.



**Fig. 80.8** Captura de Platform Builder. Estado final de Platform Builder

A continuación podemos ver los avisos que se han producido. Hay dos que como he comentado anteriormente se deben a una versión obsoleta de los archivos \*.rel de algunos módulos actualizados. Mientras que el otro aviso se debe a un simple duplicado de un fichero en el directorio raíz.

Ninguno de estos avisos es peligroso para la estabilidad del sistema porque no implica que no se hayan incluido correctamente sino que simplemente están obsoletos según su versión actual. El otro aviso adquiere menos mención si cabe, porque es un duplicado y Platform Builder utilizará la versión compatible del sistema más reciente en haberse copiado en el directorio release.

```

Warning: Found duplicate entry 'irsir.dll' NK J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\kern.exe to J:\WINXP\WINCE500\PBWorkspaces\m
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\tfat.dll to J:\WINXP\WINCE500\PBWorkspaces\m
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\com_card.dll to J:\WINXP\WINCE500\PBWorkspac
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\systemcpl.cpl to J:\WINXP\WINCE500\PBWorkspac
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\pcc_pcm.dll to J:\WINXP\WINCE500\PBWorkspac
Copying J:\WINXP\WINCE500\PBWorkspaces\mylast\Final2\RelDir\VIACN_x86_Release\Rbd8042US.dll to J:\WINXP\WINCE500\PBWorkspac
Pass 1...
Rel files are required to load the code section of modules into slot 1 and therefore freeing up more virtual memory space
Warning: No .rel file found for module sqlcecompact30.dll, using old fixup style.
Warning: No .rel file found for module dbnetlib.dll, using old fixup style.

```

**Fig. 81.8** Captura de Platform Builder. Avisos producidos

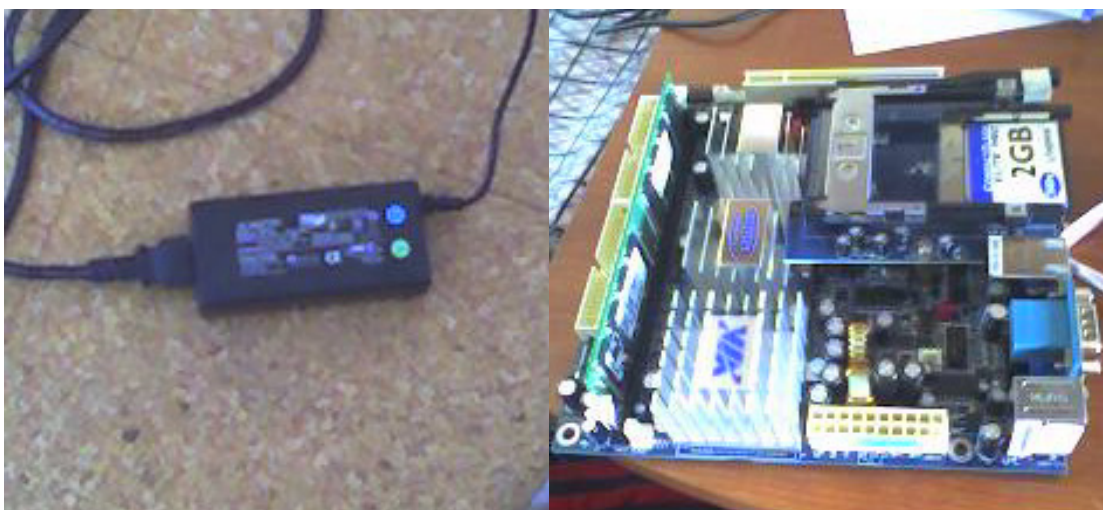
## 8.2. Montaje del dispositivo

En las siguientes imágenes podemos ver el montaje de la placa. El montaje es igual que un ordenador personal común, la única variación es el tamaño limitado, que tenemos menos tomas de corriente para dispositivos IDE y que no tenemos salida posterior para añadir una tarjeta extra.



**Fig. 82.8** Chasis y frontal

A continuación se muestra el adaptador de corriente alterna que transforma la corriente alterna de 220 V a corriente continua de 12 V apta para nuestro dispositivo. Además podemos ver la imagen de la placa sin montar en el chasis pero con la RAM y una tarjeta Compact Flash de 2 GBytes.



**Fig. 83.8** Adaptador y placa





**Fig. 84.8** Vistas de la placa montada en el chasis



**Fig. 85.8** Panel trasero



Fig. 86.8 Vista del dispositivo final

### 8.3. Errores de la aplicación sobre WCE

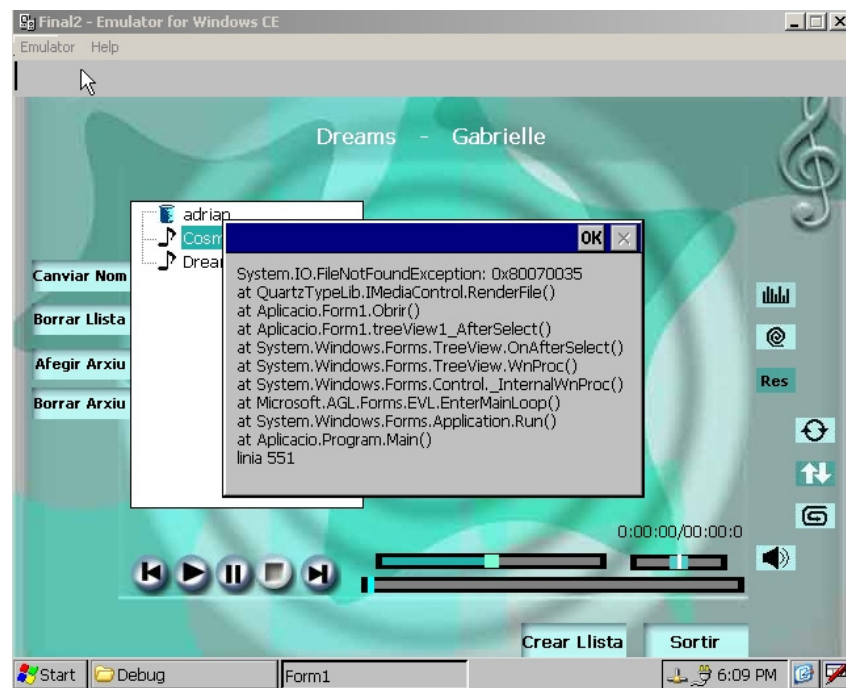


Fig. 87.8 Error de la aplicación

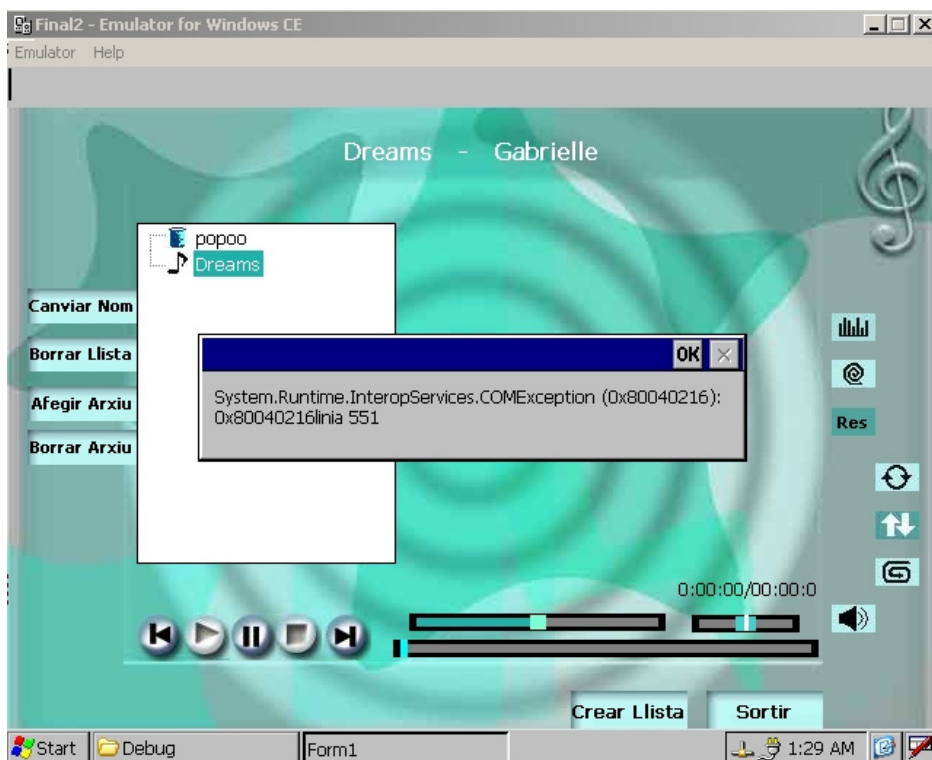


Fig. 88.8 Error de la aplicación

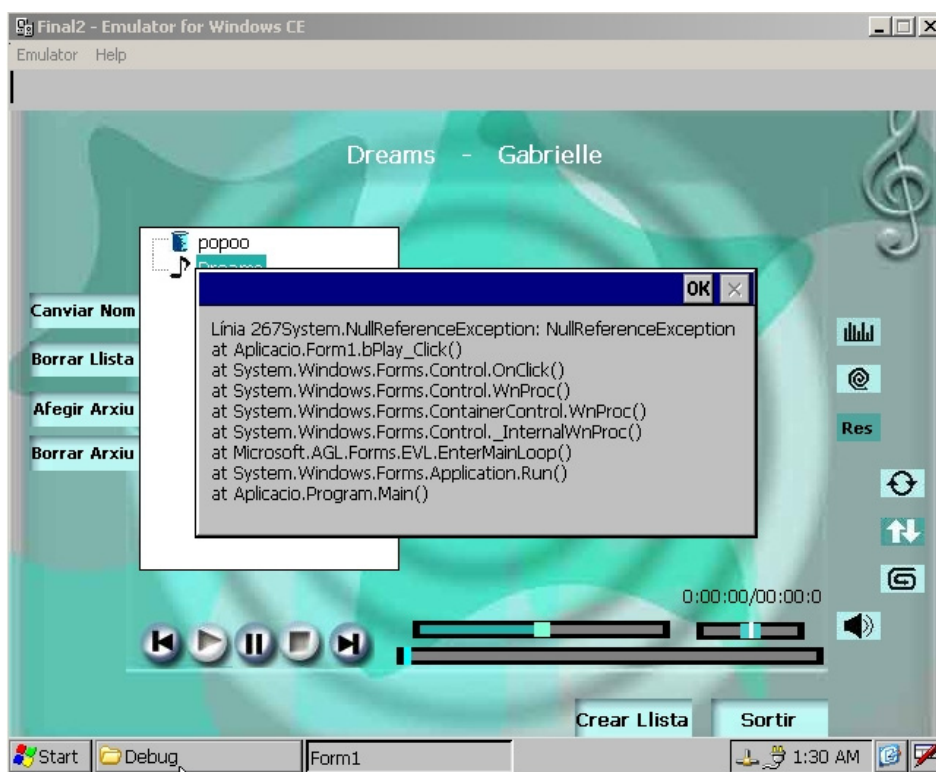


Fig. 89.8 Error de la aplicación



## CAPÍTULO 9. BIBLIOGRAFÍA

### 9.1. Bibliografía y fuentes de información

#### 9.1.1. Bibliografía

- [1] Murray, John. Inside Microsoft Windows CE. Redmon: Microsoft Press, 1998.
- [2] Stallings, William. Operating systems. 4th ed. Upper Saddle River, NJ: Prentice-Hall International, 2001.
- [3] A. Goggin, Terence. Windows CE Developer's Handbook. SYBEX, 1999.
- [4] Boling, Douglas. Programming Microsoft® Windows® CE .NET. 3rd ed. Microsoft, 2003.
- [5] Wilson, James Y. Building Powerful Platforms with Windows CE. Addison-Wesley, 2001

#### 9.1.2. Otras fuentes de información

- **TFC/PFC y trabajos de años anteriores:**
  - Cobos Pozo, Javier. Cliente FTP para Windows CE. 2002
  - Idoia Longan, Laura Val, Pablo González. Windows CE 2006
- **Documentos encontrados en la red:**
  - Boot\_Windows\_CE\_from\_DOM.pdf
  - vtf05\_via\_ems.pdf
  - APEMB0105\_WinCE\_with\_USB\_Support.pdf
  - eBoxII\_WinCE\_50\_JumpStart.pdf
  - WinCEnet.ppt
  - Sem\_emb\_11\_03.ppt

- **Sitios Web:**

- <http://blogs.msdn.com/mikehall/> - Mikehall's Embedded WEblog
- <http://msdn2.microsoft.com/en-us/default.aspx> - MSDN Home Page
- <http://www.via.com.tw/en/index.jsp> - VIA Home Page
- [http://en.wikipedia.org/wiki/Windows\\_CE](http://en.wikipedia.org/wiki/Windows_CE) - Wikipedia Windows CE
- <http://www.awprofessional.com/articles/article.asp?p=167782&rl=1> – Platform Builder construction
- <http://www.lawebdelprogramador.com/temas/mostrar.php?id=104&texto=Windows+CE> – Web del programador
- <http://www.todopocketpc.com> – Todo PocketPC
- <http://www.fortunecity.com/skyscraper/fatbit/607/wince/wince.html> - Arquitectura WCE
- <http://msdn.microsoft.com/embedded/> - MSDN Embedded
- <http://www.fmcpherson.com/knowce/> - Saber sobre WCE