**1. Introducción**

El proyecto está basado en la idea de “media center” creada por Microsoft. Bajo este punto de vista, se ha diseñado y construido un proyecto capaz de hacer cosas realmente complicadas de una forma muy sencilla, permitiendo a los usuarios más inexpertos de la informática poder interactuar con las computadoras. Además, la idea principal del proyecto es que permita a las personas mayores hacer la vida cotidiana más fácil. Entonces, se pensó que además de ver fotos, escuchar música,… a este tipo de personas les gusta leer el periódico o necesitan de una comunicación de fácil utilización. Ambas cosas tienen su cabida en este proyecto, gracias a los RSS Feeds y a la API desarrollada por el equipo de “Skype developers”, a los que hay que agradecer por su increíble trabajo. Por último, se ha añadido una sección de juegos, cargados mediante un navegador y hechos en flash, y una sección de créditos, realizados en 3D.

También los pasos para la instalación y ejecución.

**2. Diagrama de Entidad-Relación de la Base de Datos.**

La base de datos de este proyecto es muy limitada, ya que no se desea su uso para ser lo más portable posible. Aún asi, se ha creado una base de datos llamada “Teléfonos” que permite guardar contactos para llamar a través de la sección “Teléfono” del programa. En vez de realizar un diagrama, se explicará lo que contiene la base de datos. Sólo se dispone de una tabla, llamada “Teléfonos”, al igual que la base de datos. Dicha tabla contiene:

* Nombre: para guardar el nombre de la persona en cuestión. Tipo nvarchar(50).
* Apellidos: para guardar los apellidos de la persona en cuestión. Tipo nvarchar(150).
* DNI: para guardar el DNI de la persona en cuestión. Tipo numeric(8,0).
* Handle: este campo indica el “nickname” con el que la persona está registrada en Skype. Tipo nvarchar(50).
* Teléfono: contiene el teléfono completo de la persona en cuestión (incluyendo el código del país, por ejemplo, España es +34). Tipo nvarchar(20).

Como se explicará más adelante, el proyecto no necesita de más tablas para un correcto funcionamiento. En versiones posteriores, está propuesto trasladar esta base de datos a una SQLite, para su portabilidad, o incluso a XML.

**3. Diagrama de Clases**

**TUPUTAMADRE**

**4. Diseño de clases**

Las clases programadas con simplemente un resumen de qué

hace esa clase y para qué está hecha, cómo funciona a grandes

rasgos, y una lista detallada de todos sus atributos y métodos

(puede ser la misma cabecera de cada uno de ellos) comentando

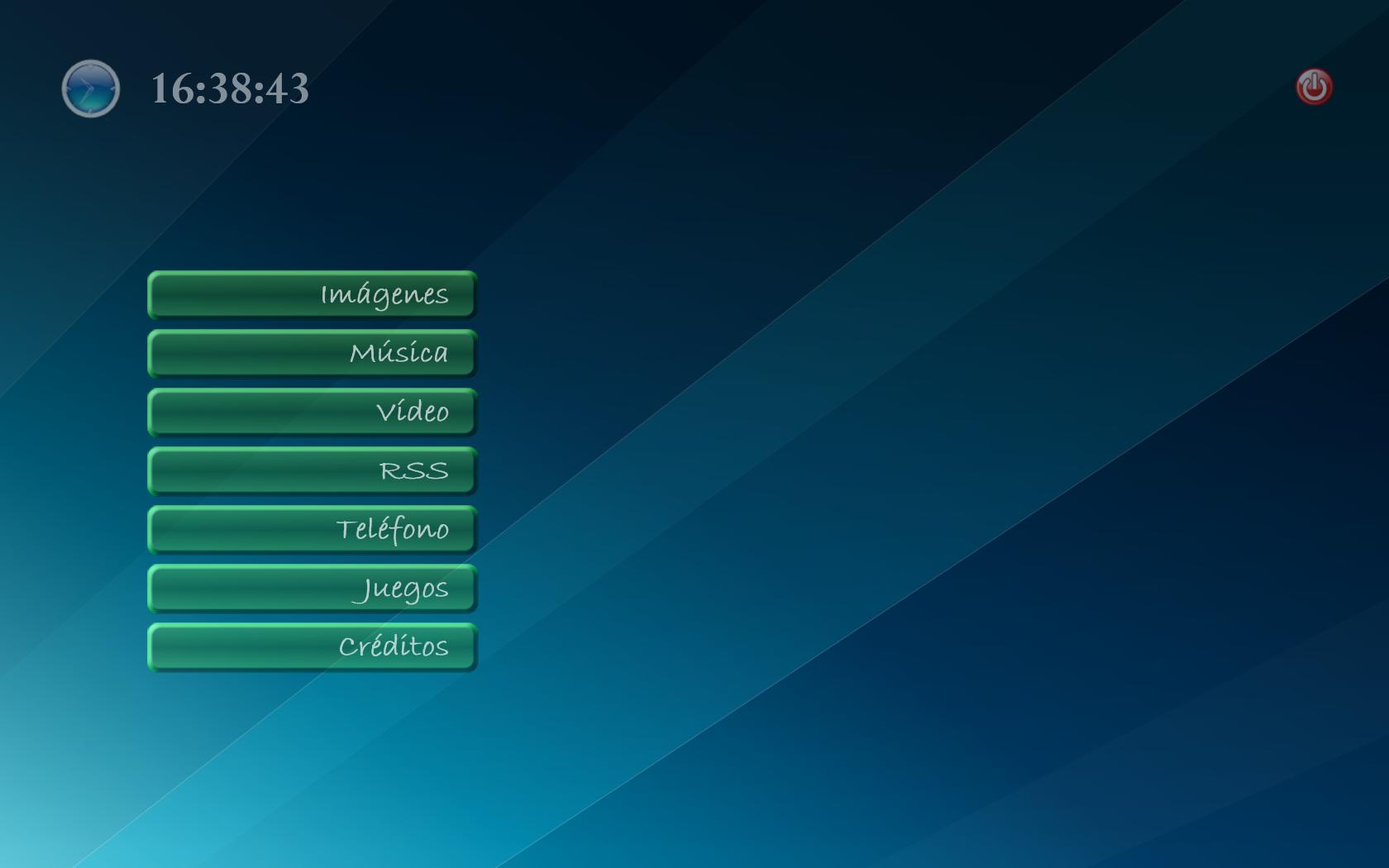
cada atributo y método qué hace (si merece la pena). Toda esta

documentación se debe generar en xml a través del compilador

csc.

**5. Diseño de interfaz**

Nada más arrancar la aplicación, nos aparece la pantalla principal del programa. En esta pantalla (figura 1) podremos seleccionar a qué parte de la aplicación nos queremos dirigir. Los botones son rectángulos redondeados y con un estilo para cuando el ratón pase por encima. Además de iluminarse mediante el estilo, a la derecha aparece una imagen que simboliza lo que realiza dicho botón.



**Figura 1.** Pantalla inicial del sistema.

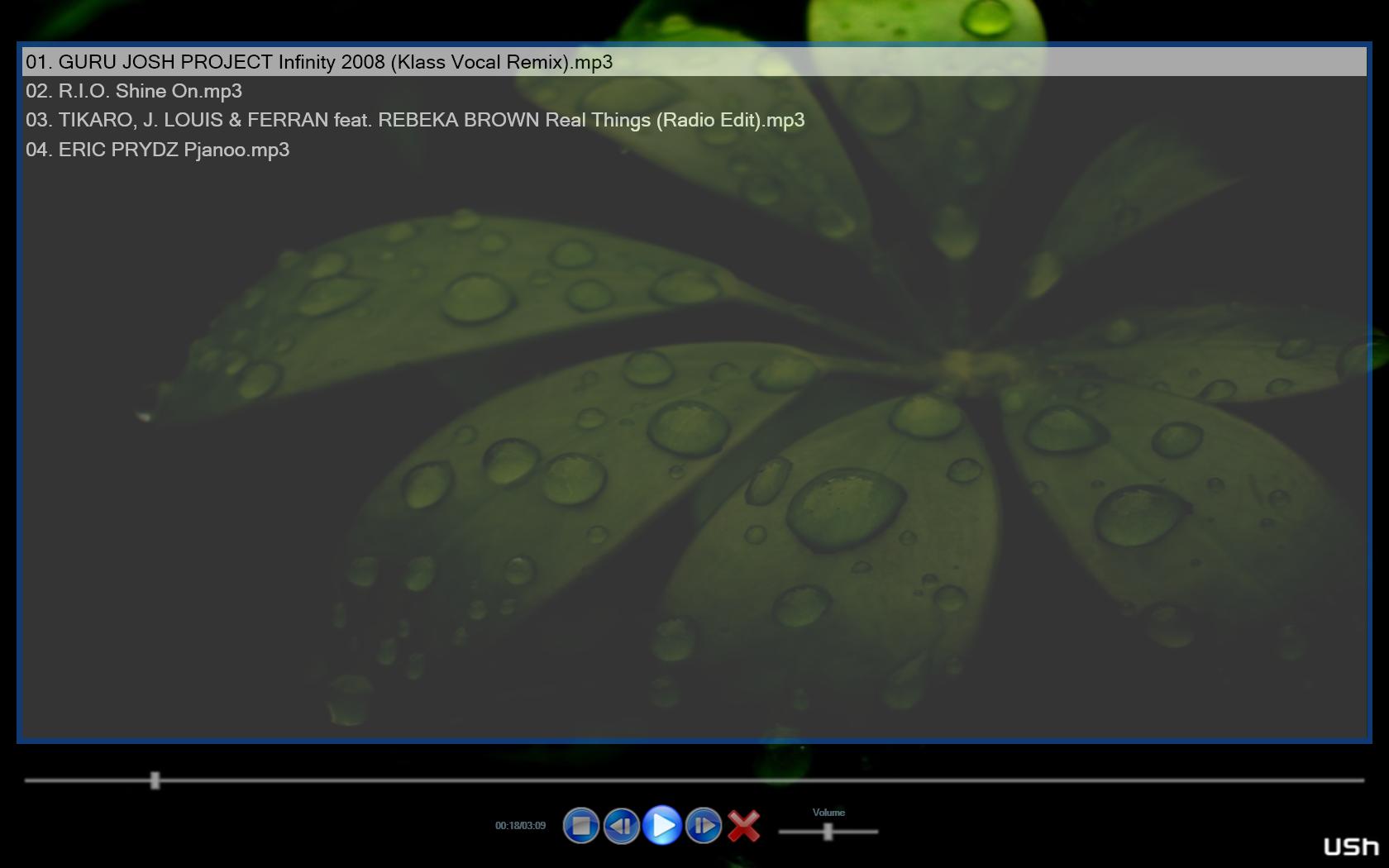
Arriba a la izquierda se puede ver un reloj, que recoge la hora del sistema, y que se ilumina si se pasa por encima el ratón; arriba a la derecha está el botón de salida del sistema. Como se ve, el sistema posee siete opciones. Cabe destacar que al pasar de una opción a otra, las ventanas se fusionan entre ellas para hacer un efecto espectacular.

**Imágenes.** Esta opción permite cargar las imágenes de una carpeta a una lista y mostrar un bonito salvapantallas con transición entre las imágenes. Si se pulsa doble clic sobre la imagen, aparece un menú de control para pasar las fotos hacia adelante, hacia atrás, parar la transición y salir. Se puede ver en la figura 2.



**Figura 2.** Apartado de imágenes con el menú.

**Música.** Al igual que en el apartado anterior, las canciones se cargarán en una colección desde una carpeta. Éstas son mostradas en una lista a través de su título (el nombre del archivo incluido la extensión). Para cargar una canción, puedes seleccionarla y darle al play o simplemente, doble clic. El slider de la parte inferior se irá actualizando según vayan pasando los segundos. Clicar en la barra de desplazamiento supone un movimiento de cinco segundos. El menú nos permite pasar de canción a la siguiente, a la anterior, parar la música, pausarla, reanudarla y subir y bajar el volumen de la misma. La figura 3 muestra el acabado final.



**Figura 3.** Apartado de música.

**Video.** Al igual que en el apartado anterior, los videos se cargarán en una colección desde una carpeta. Éstos son mostrados en una lista lateral a través de su título (el nombre del archivo incluido la extensión). En la primera versión, se utilizó un media element para realizar el sistema de vídeo, pero como consumía muchos recursos, se pasó a una segunda versión, a través de un rectángulo y el videodrawing. En la tercera versión esta previsto hacer un wrapper para incluir mplayer y de esta forma. El resto de la pantalla es exactamente igual al apartado de música.



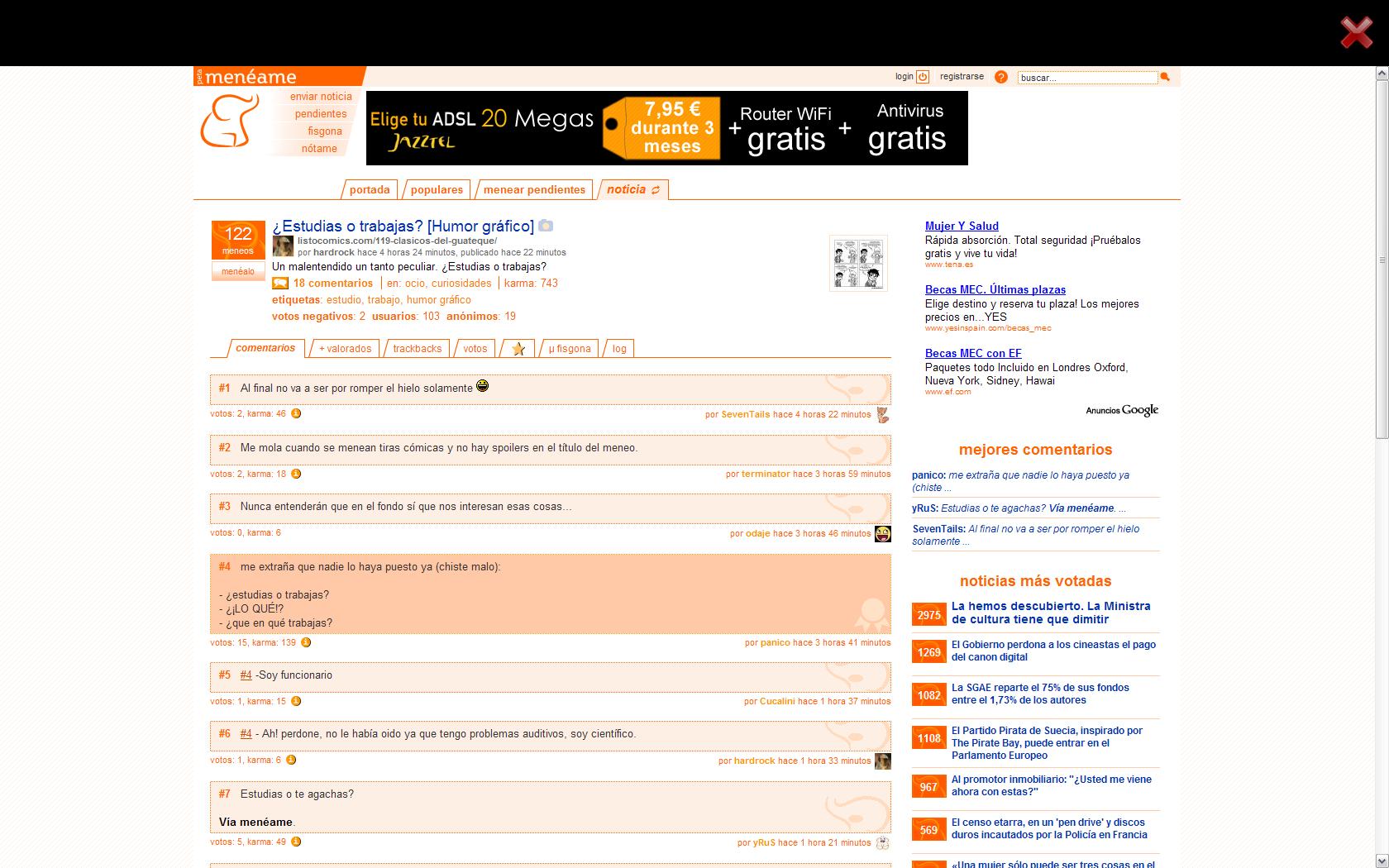
**Figura 4.** Apartado de video.

**RSS.** Mucha gente no conoce lo que es RSS ni la potencia que esto puede tener en un proyecto. RSS permite leer noticias o similar a través de un archivo .rss, que al fin y al cabo, es un XML. Lo que aquí se pretendía era tener una facilidad impresionante y tener en una sola pantalla y con pocos clics todas las fuentes de información conocidas. En vez de poner El Correo, Marca, El Mundo, etc. hemos optado por algo más informático: páginas relacionadas con los juegos (arenazero o mediavida) y páginas de recopilación de noticias (menéame). Como se muestra en la figura 5, se cargan en una lista todas las noticias del rss, y pinchando abajo en el icono correspondiente, se cambiaría la fuente de datos y con ello, el contenido de la lista.



**Figura 5.** Apartado de RSS.

Al seleccionar una noticia, el ítem de la lista (un data template) se seleccionará en un verde transparente. Al pulsar doble clic sobre una noticia, se abrirá un navegador que permitirá la visualización de dicha noticia. El navegador solo tendrá un botón, que es el de salir, tal y como muestra la siguiente figura.



**Figura 6.** Navegador del sistema.

**Teléfono.** Quizás sea la parte más llamativa del sistema desarrollado. Gracias a la API de Skype, se ha podido realizar la comunicación por voz que permite, de una manera facilísima, llamar a través del programa. El funcionamiento se explica en otro apartado, aquí simplemente decir que de la base de datos se carga a una lista, también modificada para que los datos seleccionados se muestren en azul transparente. A la izquierda un teclado numérico para añadir el número de teléfono a llamar. Para llamar a un fijo, hay que disponer de una cuenta SkypeOut, sino, no funcionará. El resultado es el siguiente.



**Figura 7.** Apartado de teléfono.

**Juegos.** Es un apartado muy simple pero a la vez, dota a la aplicación de más y más cosas, lo que hace de ella una aplicación robusta. Al clicar en este apartado, nos sale una imagen de fondo y un icono por cada juego registrado. De momento, solo se ha puesto un juego, pero añadir otro sería tan fácil como crear el botón y con el enlace a la URL del juego. URL porque el juego está en flash colgado en una página externa, y a través del navegador previamente creado para RSS y de la URL del juego, obtenemos un bonito lanzador de juegos SWF. La pantalla de los juegos sería la mostrada en la siguiente figura.



**Figura 8.** Apartado de juegos.

**Créditos.** Si el teléfono era la parte más llamativa hasta el momento, los créditos no se quedan atrás. Unos créditos realizados en 3D que permiten, a través del ratón, girar alrededor de la situación y hacer zoom sobre ella. Algo realmente espectacular que sin la ayuda de 3DTools hubiese sido casi imposible. El resultado son unos créditos originales pero que carecen de sentido debido a su alto consumo de recursos y necesidad de tarjeta gráfica. Aún así, como prueba de utilización, es increíble.



**Figura 8.** Apartado de créditos.

**6. Documentación interna**

Todo lo demás. NO HAY QUE LISTAR EL CODIGO. Aquí se

indicarían las descripciones de los algoritmos principales.

**7. Conclusiones**

Una vez terminado el proyecto, el sabor de boca final es BUENO. Primero he de decir que esto forma parte de mi proyecto fin de carrera, proyecto compuesto por muchas más cosas que complementaran está magnífica herramienta construida, bajo mi punto de vista. ¿Por qué bueno y no muy bueno? Porque WPF es una maravilla en cuanto a sencillez, pero consume demasiados recursos, lo que lo hace inútil para dispositivos de bajo rendimiento, e incluso de medio. Considero el resultado final bastante espectacular, considerando el tiempo dedicado y comparándolo con otras alternativas del mercado.

En este apartado me gustaría comentar también las cosas que se están incluyendo en el proyecto fin de carrera que dotarán a esta aplicación de la mayor espectacularidad posible. En primer lugar, se permitirá a las familias (recordemos, está diseñado para gente de la tercera edad) modificar las diferentes carpetas de las que se leen los archivos. La manera de hacerlo será desde FTP. Se dispondrá de una web, con usuario y contraseña por cada persona que disponga del programa, en donde, a través de un panel de control muy intuitivo y gráfico, se podrán subir fotos, música y videos a un servidor central. La aplicación, al comenzar la parte de fotos, por ejemplo, comprobará los cambios del FTP. Si se dispone de alguna fotografía nueva, descargará las imágenes a su carpeta local, para su posterior visualización. Esto hará posible que toda la familia pueda dejar fotos al *abuelo* y éste poder disfrutar de ellas.

Por otro lado, se está diseñando un sistema de plugins para la extensibilidad de la aplicación. Esto permitirá a cualquier persona que quiera añadir funcionalidad a la aplicación una facilidad increíble para hacerlo. De esta forma, se consigue que no haya que recompilar la aplicación cada vez que se quiera añadir funcionalidad, sino que ésta, al arrancar, comprobará los diferentes plugins que haya para cargarlos.

**8. Apéndices**

Voy a aprovechar este punto para hablar de dos librerías externas que se han utilizado en este proyecto: SKYPE4COMLib y 3DTools.

Skype4Com es una librería desarrollada por el equipo de Skype que permite el uso de su programa de una forma transparente, sin tener que interactuar con él. Eso sí, el programa debe estar instalado y funcionando. Gracias al modo “Silent Mode” existente en la API, la comunicación con el programa sólo abre una ventana de permiso la primera vez que se invoca. Después, nunca más se verá una ventana de Skype, por lo que, al ojo del usuario, las llamadas se realizan completamente desde el programa. El acabado que ha permitido es muy profesional, y estoy muy satisfecho con la documentación que se da de la API, aunque la parte de ejemplos es totalmente nula.

3DTools es una librería de fácil uso que permite, con pocas líneas de código, manejar objetos en 3D a través de cámaras y controles. Añadiendo código en el XAML se ha conseguido poder rotar y hacer zoom sobre el Viewport 3D con el ratón. Hacer esto de forma programática sería más que complicado. En cambio, sabiendo utilizar la librería, con sólo 15 minutos tienes tus cámaras funcionando. En la demo se ve perfectamente el funcionamiento.