

1.2 ASPECTOS TÉCNICOS



El componente base del proyecto es la cámara científica modelo MicroPublisher 3.3 RTV de la marca QImaging, expuesta en la Figura 2. Si empezamos sabiendo que esta es la cámara de la que disponen los investigadores del Oceanográfico, todas las demás decisiones, cuestionables por supuesto, parten de este punto concreto.



Figura 2.- Cámara científica MicroPublisher 3.3 RTV de QImaging

Como se adelanta en la introducción, el lenguaje de programación elegido debido a que las librerías de la cámara (QCam Version: 2.0.10, Document Version: 1.3) están definidas así, es C++, y aunque se podrían haber elegido otros lenguajes como por ejemplo Java o C#, se hubieran necesitado "parches" adicionales para la traducción, por eso un primer paso es programar en C/C++. El sistema operativo sobre el que se construye la aplicación es Windows XP debido a que es el que tienen instalado los ordenadores del Oceanográfico, aunque también se podría haber hecho sobre Windows 2000, Mac OS X (10.4.7 o superior) o en Linux (versión de kernel 2.6 o superior).

Otra decisión importante al comenzar el proyecto, es saber en qué entorno de desarrollo se va a programar (Integrated Development Environment, IDE). En principio se usó Microsoft Visual Studio 2010 Express (versión de prueba), pero debido a razones que se explicarán a fondo más adelante, se cambió a Microsoft Visual Studio 2008 Professional (con licencia académica), aunque en pocas palabras fue por poder disponer de unas opciones sin las cuales se imposibilitaba el avance. También se podría haber elegido otro entorno como por ejemplo Eclipse, que es de código abierto, añadiéndole los plugins (programas adicionales) necesarios para trabajar en C++.

Una vez escogido el lenguaje C++ en el entorno Visual Studio, se puede optar a diferentes niveles de programación, y en este caso podemos diferenciar entre tres, el nivel más bajo sería programar en C++ nativo, sin ninguna clase de librerías preprogramadas, de manera que habría que crear todo desde cero, incluso el interfaz gráfico (Graphical User Interface, GUI); un nivel medio sería uno que ya posee clases predeterminadas que facilitan el desarrollo del software (Microsoft Foundation Classes, MFC), con facilidades para conseguir buenas interfaces para el usuario (Application Programming Interface, API); y por último un nivel alto que ya posee herramientas tanto para la programación (Common Language Runtime, CLR), como para versiones extendidas (Common Language Infrastructure, CLI). En nuestro proyecto se ha decidido programar en el más alto nivel, ya que así hay la posibilidad de usar los Formularios de Windows.

Los Formularios de Windows fueron una de las primeras ventajas que supuso un cambio en la balanza a la hora de decidir el nivel de programación con el que se desarrollaría todo el proyecto, en favor del nivel más alto. Esto se debe a que se pueden crear fácilmente aplicaciones con interfaces de usuario de estilo Windows (con ventanas, botones, etc.), en las que gran parte del código se genera automáticamente y sólo hay que configurar algunas propiedades y cuidar las acciones en forma de eventos entre unos componentes y otros del formulario.

Una decisión más que fue preciso tomar debido a que el procesamiento y posterior tratamiento de imágenes era complicado usando sólo las herramientas disponibles de los formularios, fue la búsqueda de librerías específicas para trabajar con las fotografías capturadas por la cámara científica. La elección tomada fue apoyarse en las librerías CImg⁶ de código abierto, e incluso también en las librerías Magick++⁷ (cuando sea necesario trabajar con imágenes de formato .jpeg o .tiff), que cumplen las funciones que quedaban por cubrir.

Por tanto, podemos resumir que el software del proyecto toma como base los Formularios de Windows para crear una aplicación con un buen interfaz, práctico para el usuario, y además hace de puente entre las librerías QCam API de la cámara que captura las imágenes, y la librería CImg que se encarga del posterior tratamiento de las imágenes capturadas, como representa la Figura 3.



Figura 3.- Esquema software del proyecto

Y así es como se ha desarrollado paso a paso el proyecto, haciendo referencia a las bifurcaciones que es necesario tomar cada vez que se escoge un camino a seguir, a favor siempre de encontrar la mejor solución posible y disponible a un determinado problema. A partir de ahora se profundizará en lo que hay entre decisión y decisión, propiamente dicho en la labor de programación de la aplicación.

⁶Librerías CImg
⁷Librerías Magick++

