**2.2 EL PROGRAMA** 

Empezamos a partir de ahora a estudiar a fondo el código de la aplicación. Y se hace de manera exhaustiva debido a que este es un proyecto abierto y habrá más estudiantes que continúen desarrollándolo, y en pos de facilitar la comprensión del trabajo ya realizado para poder seguir desde el punto en que se dejó anteriormente, se muestra todo el código, convenientemente comentado y con cualquier explicación necesaria para aclarar el funcionamiento del programa.

Al empezar a escribir código, lo primero que hay que hacer es ir a la Página de Inicio de Visual Studio y crear un proyecto correspondiente a la plantilla "Aplicación de Consola Win32", ya que si lo hiciéramos directamente con una "Aplicación de Windows Forms", tarde o temprano se descubre que no se puede por ejemplo usar la librería CImg para tratamiento de imágenes (ver errores de compilación LNK2028 y LNK2019), y por eso nos apoyamos en la Consola de Win32.

Al crear un proyecto de consola, la función "main" pertenece a este primer archivo, por lo que para trabajar con Formularios de Windows, será desde OTOLIVE.cpp (como se ha llamado al proyecto) desde donde se haga una llamada para ejecutar el Formulario Principal, el que se encargará de llamar a los formularios secundarios de Información General, de Calibración del Microscopio o de conteo de anillos en los Transectos.

El formulario principal denominado Form\_Ppal.h, posee menús desplegables y botones para esa labor de conexión con todas las funcionalidades de la aplicación. Al mostrarse la ventana de diseño de un formulario comprobaremos la sencillez de añadir elementos para nuestra interfaz. Desde la ventana del cuadro de herramientas sólo hay que arrastrar por ejemplo un botón para colocarlo en la posición que deseemos del formulario. Lo que es más complicado es dar las órdenes que se ejecutarán cuando se pulse ese botón. Visual Studio dispone de gran cantidad de tipos de eventos distintos que puedan desencadenar unas funciones u otras. Veamos toda esta casuística a lo largo de los archivos de la aplicación OTOLIVE.

Pero antes, hay que configurar algunas propiedades en nuestro proyecto:

- Ya lo hemos dicho, pero por poner todo en orden desde el principio, se crea un proyecto con la plantilla de Consola de Win32.
- En las propiedades de nuestro proyecto recién creado, vamos a Propiedades de Configuración, de ahí a General, y en la opción de Compatibilidad con Common Language Runtime, cambiamos a que sea Compatible con Common Language Runtime (/clr). Esto es lo que permite hacer llamadas a otras funciones aparte del Formulario Principal.
- En las mismas propiedades del proyecto, pero ahora en Propiedades Comunes, elegimos Agregar una nueva referencia..., y ahí buscamos el componente Microsoft.Office.Interop.Excel con la versión 11.0.0.0 y lo agregamos, para poder acceder a las bases de datos, cuyo formato serán documentos de Excel.
- De nuevo en las Propiedades de Configuración, vamos ahora al Vinculador, y de ahí a Entrada. De sus opciones, en Dependencias Adicionales se añade el macro

"\$(InputDir)QCamDriver.lib". Haciendo esto junto con incluir el archivo QCamDriver.lib en la carpeta del proyecto (../OTOLIVE/OTOLIVE/), podremos usar las librerías de la cámara científica. El archivo QCamDriver.lib se encuentra en la carpeta ../QImaging/SDK/libs/ de nuestro disco duro, que se crea al instalar el archivo QCamSDKInstaller.exe.

 Una vez que hemos realizado los pasos anteriores, si se desea usar la librería de la cámara, es necesario incluir en cada archivo que la vaya a necesitar, la cabecera #include "QCamAPI.h". Lo mismo ocurre si necesitamos la librería CImg de procesamiento de imágenes, para lo que añadimos la cabecera #include "CImg.h" y también la línea using namespace cimg\_library;. A esto va igualmente unida la acción de añadir respectivamente los archivos CImg.h y QCamAPI.h a la carpeta del proyecto, donde también pusimos QCamDriver.lib, que es además donde se irán guardando los archivos que se vayan creando para el proyecto.

Con estos pasos ya estamos listos para empezar a programar.

## 38 2.2 EL PROGRAMA