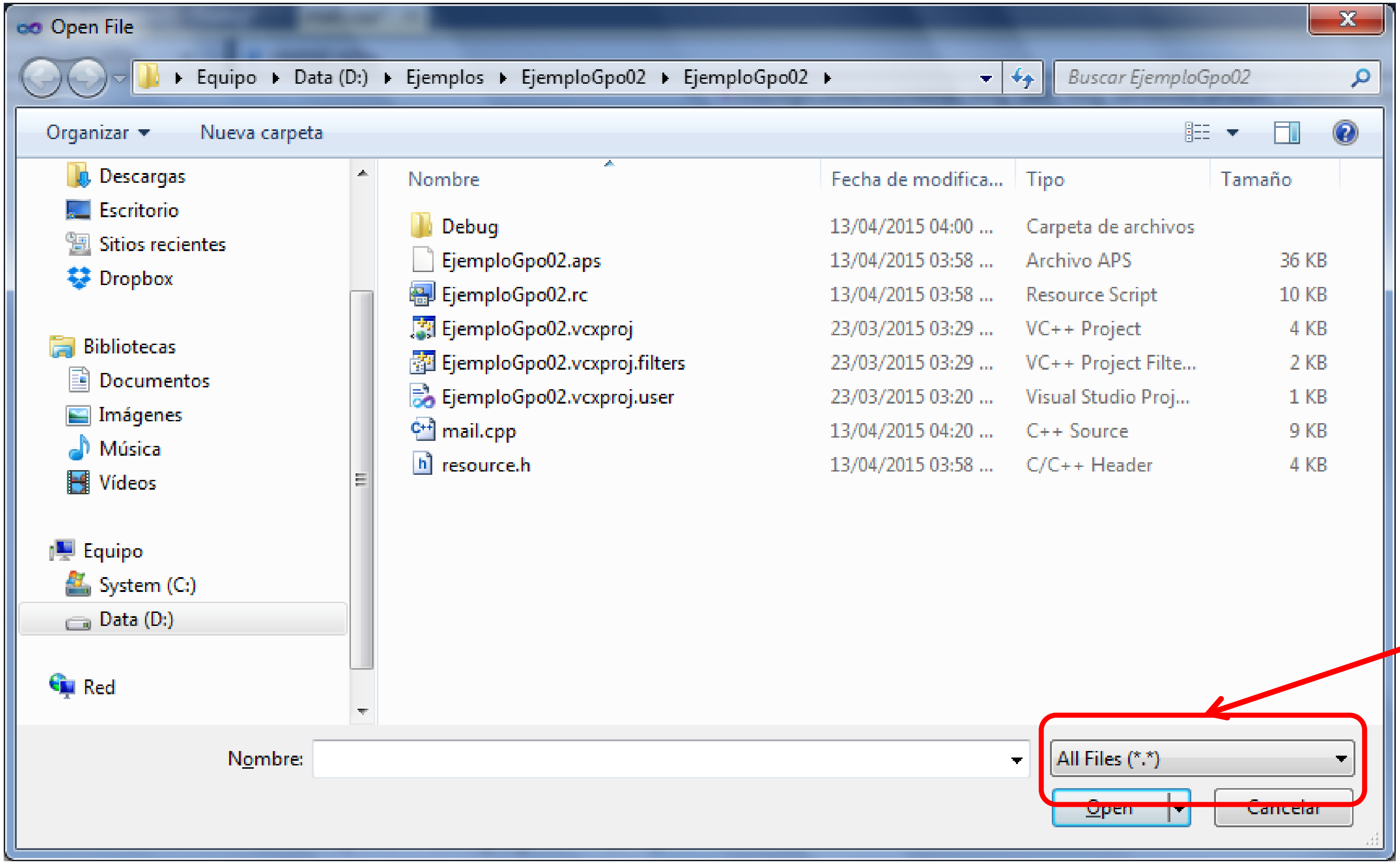
Win32 API (Application Programming Interface).

Programación 2

MGTI Alejandro Villarreal Mojica

Common Dialog

- #include <CommDlg.h>



Nombre	Fecha de modifica...	Tipo	Tamaño
Debug	13/04/2015 04:00 ...	Carpeta de archivos	
EjemploGpo02.aps	13/04/2015 03:58 ...	Archivo APS	36 KB
EjemploGpo02.rc	13/04/2015 03:58 ...	Resource Script	10 KB
EjemploGpo02.vcxproj	23/03/2015 03:29 ...	VC++ Project	4 KB
EjemploGpo02.vcxproj.filters	23/03/2015 03:29 ...	VC++ Project Filte...	2 KB
EjemploGpo02.vcxproj.user	23/03/2015 03:20 ...	Visual Studio Proj...	1 KB
mail.cpp	13/04/2015 04:20 ...	C++ Source	9 KB
resource.h	13/04/2015 03:58 ...	C/C++ Header	4 KB

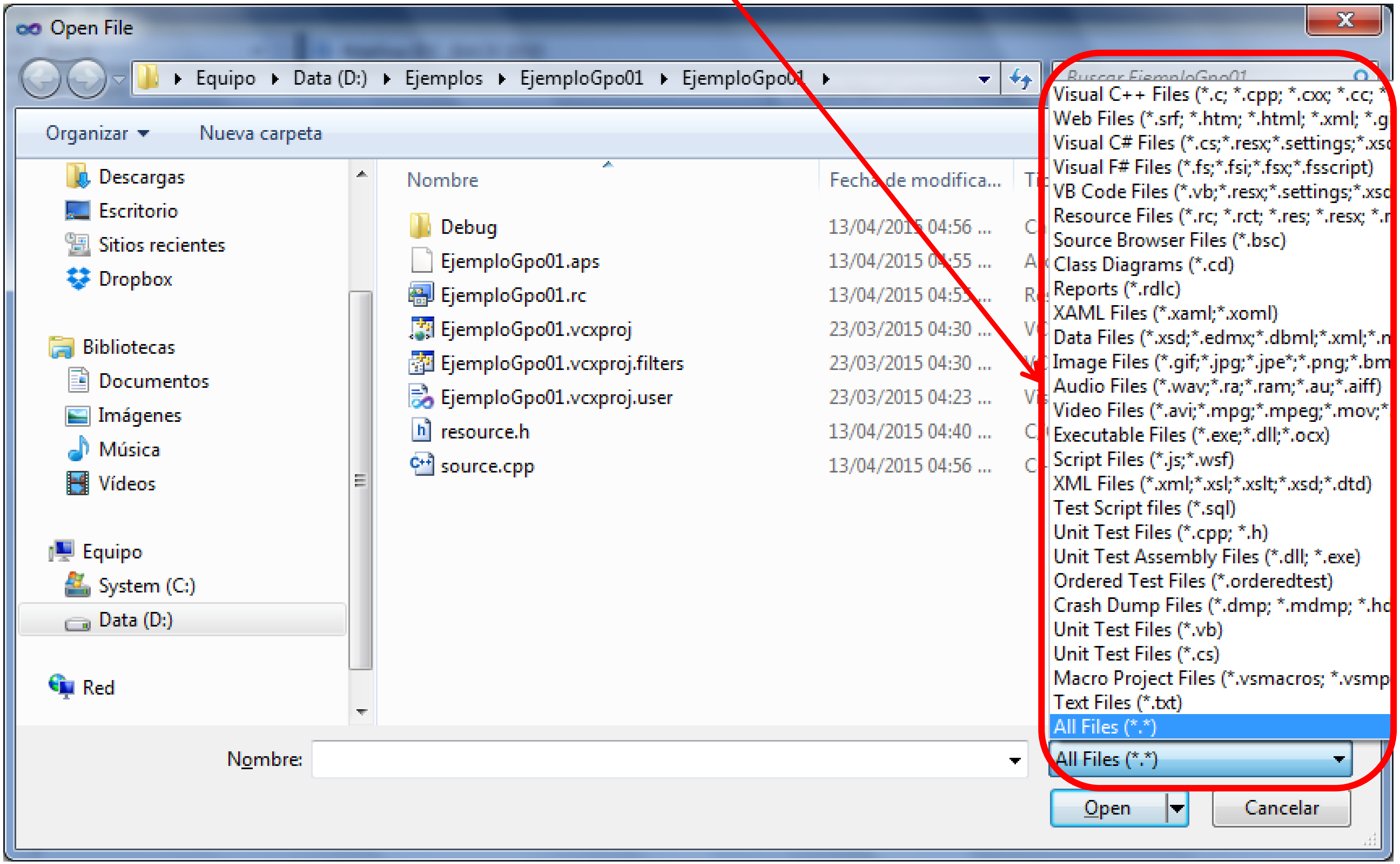
ofn.lpstrFilter

Definición de variable (Objeto)

```
char szFile[MAX_PATH];       // buffer for file name

OPENFILENAME ofn;       // common dialog box structure
HANDLE hf;              // file handle
// Initialize OPENFILENAME
ZeroMemory(&ofn, sizeof(ofn));
ofn.lStructSize = sizeof(ofn);
ofn.hwndOwner = hDlg;
ofn.lpstrFile = szFile;
// Set lpstrFile[0] to '\0' so that GetOpenFileName does not
// use the contents of szFile to initialize itself.
ofn.lpstrFile[0] = '\0';
ofn.nMaxFile = MAX_PATH;
ofn.lpstrFileTitle = NULL;
ofn.nMaxFileTitle = 0;
ofn.lpstrInitialDir = NULL;
ofn.lpstrFilter = "Todos\0*.*\0Archivos Texto\0*.TXT\0Archivos Word
(97-2003)\0*.doc\0Archivos Word\0*.docx\0Imagenes jpg\0*.jpg\0";
ofn.nFilterIndex = 2;
ofn.Flags = OFN_PATHMUSTEXIST | OFN_FILEMUSTEXIST;
```

ofn.lpstrFilter



Open File
Equipo ▸ Data (D:) ▸ Ejemplos ▸ EjemploGpo01 ▸ EjemploGpo01 ▸
Buscar EjemploGpo01
Organizar ▾
Nueva carpeta
Descargas
Escritorio
Sitios recientes
Dropbox
Bibliotecas
Documentos
Imágenes
Música
Vídeos
Equipo
System (C:)
Data (D:)
Red
Visual C++ Files (*.c; *.cpp; *.cxx; *.cc; *
Web Files (*.srf; *.htm; *.html; *.xml; *.g
Visual C# Files (*.cs;*.resx;*.settings;*.xsd
Visual F# Files (*.fs;*.fsi;*.fsx;*.fsscript)
VB Code Files (*.vb;*.resx;*.settings;*.xsd
Resource Files (*.rc; *.rct; *.res; *.resx; *.r
Source Browser Files (*.bsc)
Class Diagrams (*.cd)
Reports (*.rdlc)
XAML Files (*.xaml;*.xoml)
Data Files (*.xsd;*.edmx;*.dbml;*.xml;*.n
Image Files (*.gif;*.jpg;*.jpe*;*.png;*.bm
Audio Files (*.wav;*.ra;*.ram;*.au;*.aiff)
Video Files (*.avi;*.mpg;*.mpeg;*.mov;*
Executable Files (*.exe;*.dll;*.ocx)
Script Files (*.js;*.wsf)
XML Files (*.xml;*.xsl;*.xslt;*.xsd;*.dtd)
Test Script files (*.sql)
Unit Test Files (*.cpp; *.h)
Unit Test Assembly Files (*.dll; *.exe)
Ordered Test Files (*.orderedtest)
Crash Dump Files (*.dmp; *.mdmp; *.hd
Unit Test Files (*.vb)
Unit Test Files (*.cs)
Macro Project Files (*.vsmacros; *.vsmp
Text Files (*.txt)
All Files (*.*)
Open
Cancelar

Nombre	Fecha de modifica...	Ti
Debug	13/04/2015 04:56 ...	C
EjemploGpo01.aps	13/04/2015 04:55 ...	A
EjemploGpo01.rc	13/04/2015 04:55 ...	Re
EjemploGpo01.vcxproj	23/03/2015 04:30 ...	V(
EjemploGpo01.vcxproj.filters	23/03/2015 04:30 ...	V(
EjemploGpo01.vcxproj.user	23/03/2015 04:23 ...	Vi
resource.h	13/04/2015 04:40 ...	C
source.cpp	13/04/2015 04:56 ...	C

Nombre:

All Files (*.*)

Mostrar diálogo - Abrir

```
// Display the Open dialog box.

if (GetOpenFileName(&ofn) == TRUE)
{
SetWindowText(GetDlgItem(hDlg, IDC_EDIT1),
ofn.lpstrFile);
}
else
MessageBox(hDlg, "No eligió archivo", "Aviso",
MB_OK|MB_ICONINFORMATION);
```

```
OPENFILENAME ofn;
char szFileName[MAX_PATH] = "";

ZeroMemory(&ofn, sizeof(ofn));

ofn.lStructSize = sizeof(ofn); // SEE NOTE BELOW
ofn.hwndOwner = hDlg;
ofn.lpstrFilter =
   "Text Files (*.txt)\0*.txt\0All Files (*.*)\0*.*\0";
ofn.lpstrFile = szFileName;
ofn.nMaxFile = MAX_PATH;
ofn.Flags =
   OFN_EXPLORER | OFN_FILEMUSTEXIST | OFN_HIDEREADONLY;
ofn.lpstrDefExt = "txt";

if(GetOpenFileName(&ofn))
{
    // Do something usefull with the filename
   // stored in szFileName
}
```

- The lpstrFile points to the buffer we have allocated to store the name of the file, since filenames can't be larger than MAX_PATH this is the value that I've chosen for the buffer size.
- The flags indicate that the dialog should only allow the user to enter filenames that already exist (since we want to open them, not create them) and to hide the option to open the file in readonly mode, which we aren't going to support.
- Finally we provide a default extention, so if the user types in "foo" and the file is not found, it will try to open "foo.txt" before finally giving up.

- To select a file for saving instead of opening, the code is nearly the same, except for calling GetSaveFileName() we need only change the flags member to options more suitable for saving.

```
ofn.Flags = OFN_EXPLORER | OFN_PATHMUSTEXIST |
OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT;
```

- In this case we no longer want to require the file exist, but we do want the directory to exist since we aren't going to try and create it first. We'll also prompt the user if they select an existing file to make sure they want to overwrite it.

Mostrar diálogo – Guardar como

```
//ofn.Flags = OFN_EXPLORER | OFN_FILEMUSTEXIST |
OFN_HIDEREADONLY;
ofn.Flags = OFN_EXPLORER | OFN_PATHMUSTEXIST |
OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT;
ofn.lpstrDefExt = "dat";

// Display the Save dialog box.

if(GetSaveFileName(&ofn))
{
// Do something usefull with the filename
// stored in szFileName
MessageBox(hDlg, ofn.lpstrFile, ofn.lpstrFile, MB_OK);
}
```

```
OPENFILENAME ofn;
char szFileName[MAX_PATH] = "";

ZeroMemory(&ofn, sizeof(ofn));

ofn.lStructSize = sizeof(ofn); // SEE NOTE BELOW
ofn.hwndOwner = hDlg;
ofn.lpstrFilter = "Text Files (*.txt)\0*.txt\0All Files
(*.*)\0*.*\0";
ofn.lpstrFile = szFileName;
ofn.nMaxFile = MAX_PATH;
ofn.Flags = OFN_EXPLORER | OFN_PATHMUSTEXIST | OFN_HIDEREADONLY |
OFN_OVERWRITEPROMPT;
ofn.lpstrDefExt = "txt";

if(GetSaveFileName(&ofn))
{
// Do something usefull with the filename stored in szFileName
       MessageBox(hDlg, ofn.lpstrFile, ofn.lpstrFile, MB_OK);
}
```

GetCurrentDirectory

```
// Definir variable global para
// guardar la ruta y carpetas donde
// está el programa:
char Folder[MAX_PATH] = "";

// en el WINAPI WinMain:
GetCurrentDirectory(MAX_PATH,
Folder);
```