

Como instalar Dev-C++ y SDL en Windows

Créditos

- » **Autor:** Hugo Ruscitti
- » **Fecha:** Sábado 9 de Julio del 2005

Sobre Dev-C++

Dev-C++ (o devcpp) es un entorno de desarrollo integrado (IDE) para los lenguajes de programación C y C++.

Incluye el compilador Mingw, una adaptación del compilador GCC (GNU Compiler Collection) para Windows, y un conjunto de herramientas que facilitan el desarrollo en sistemas Windows.



Obtener e instalar Dev-cpp:

Primero debemos obtener el compilador Mingw y el entorno de desarrollo Dev-C++.

En la página de descargas del proyecto dev-cpp:

- » <http://www.bloodshed.net/dev/devcpp.html>

podremos obtener una versión del programa que incluye el compilador Mingw con GCC y GDB (depurador). Al momento de escribir esta guía la última versión del programa es la 4.9.9.2 (beta 5) y ocupa unos 9 MB. Puede obtener esa versión directamente desde:

- » http://prdownloads.sourceforge.net/dev-cpp/devcpp-4.9.9.2_setup.exe

Una vez descargado, instale el programa en "c:\dev-cpp", al menos si busca seguir esta guía paso a paso.

El primer programa con Dev-c++:

Cuando inicie el programa por primera vez verá un diálogo de configuración rápida donde podrá configurar el idioma y la apariencia del programa. Luego puede cambiar esas opciones desde:

```
Herramientas -> Opciones del entorno -> Interfaz  
(Tools -> Environment Options -> Interface)
```

Para crear un nuevo programa debemos generar un proyecto desde:

```
Archivo -> Nuevo Proyecto  
(File -> New Project)
```

y seleccionar el tipo de programa como "Console Application".

Crear un nuevo proyecto nos resultará de utilidad a la hora de modificar, distribuir y compilar el programa.

Cuando seleccione "Aceptar", tendrá un código de programa inicial, cambie ese código por:

```
#include <stdlib.h>  
#include <stdio.h>  
  
int main (void)  
{  
    printf("hola mundo\n");  
  
    system("pause");  
    return 0;  
}
```

y pulse la tecla F9 para generar y ejecutar el programa.

Todas las operaciones relacionadas con la compilación del proyecto se encuentran en el menú Ejecutar (Execute).

Recomendaciones adicionales:

Es recomendable, antes de continuar, definir algunos parámetros de configuración del programa:

En "Herramientas -> Opciones del Editor" ("Tools -> Editors Options") seleccione "Permitir el uso del caracter Tabulador" ("Use Tab Character") y quite la marca de "Tabuladores Inteligentes" ("Smat Tabs"). De esta manera el ancho de las tabulaciones del código fuente se pueden alterar con la opción Tab Size (desde las opciones de ese editor) a gusto de cada programador.

Añadiendo las bibliotecas SDL:

Existen bibliotecas complementarias a SDL, como por ejemplo SDL_mixer, SDL_net y SDL_gui. Cada una de estas bibliotecas añade capacidades adicionales que resultan de mucha utilidad.

En esta guía instalaremos solo 3 bibliotecas: SDL (base), SDL_image y SDL_mixer. Puede obtener mas información acerca de todas las bibliotecas complementarias en www.libsdl.org; el procedimiento de instalación es similar para todas ellas.

Dado que buscamos desarrollar y ejecutar programas desarrollados con SDL, nuestro sistema requerirá de 2 versiones de cada biblioteca:

- » La versión de desarrollo, que incluye todo lo necesario para "generar" programas (archivos de cabecera...).
- » La versión de ejecución (runtime) que será necesaria para ejecutar los programas creados con SDL (.DLLs en Windows).

Las bibliotecas de ejecución (runtime) deben estar presentes en cada equipo que quiera ejecutar nuestros programas, habitualmente se incluyen en el mismo paquete del programa.

Descargar la biblioteca principal de SDL:

SDL (base)

- » Pagina principal: <http://www.libsdl.org/download-1.2.php>
- » Ejecución: <http://www.libsdl.org/release/SDL-1.2.8-win32.zip>
- » Desarrollo: <http://www.libsdl.org/release/SDL-devel-1.2.8-mingw32.tar.gz>

Instalar SDL (base)

Debemos descomprimir SDL-1.2.8-win32.zip y SDL-devel-1.2.8-mingw32.tar.gz en diferentes carpetas, por ejemplo SDL_run y SDL_dev respectivamente.

El archivo contenido en SDL-1.2.8-win32.zip, SDL.dll, se debe copiar a la carpeta System dentro de la carpeta principal de Windows, generalmente la ruta completa es "c:\windows\system".

El archivo SDL-devel-1.2.8-mingw32 incluye varios archivos y carpetas:

- » grabe el contenido de la carpeta "lib" (4 archivos) en "c:\dev-cpp\lib"
- » entre en la carpeta include y copie la carpeta SDL completa en "c:\dev-cpp\include", de forma que los archivos .h queden dentro de "c:\dev-cpp\include\SDL"

Primer programa con SDL en Dev-cpp

Solo hace falta crear un proyecto y configurar la compilación para que funcione con la biblioteca que instalamos:

Iniciamos Dev-C++, y generamos un nuevo proyecto de tipo "Console Application".

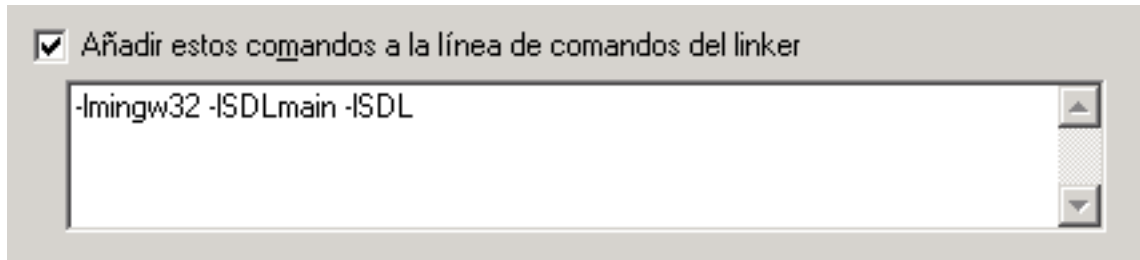
Tendremos que definir los parámetros de compilación para vincular las bibliotecas al programa.

Abrimos el menú "Herramientas -> Opciones del Compilador" ("Tools" -> "Compiler Options") y seleccionamos la opción:

```
"Añadir estos comandos a la línea de comandos del Linker"  
(Add these commands to the linker...)
```

Luego se debe completar el campo siguiente con:

```
-lmingw32 -lSDLmain -lSDL
```



Código de prueba

Para verificar que la instalación funciona correctamente, reemplace el contenido del archivo principal del proyecto por otro ejemplo sencillo de SDL:

```
#include <stdio.h>
#include <SDL/SDL.h>

void imprimir_rectangulo (SDL_Surface *screen, int x, int y);

int main(int argc, char *argv[])
{
    SDL_Surface *screen;
    SDL_Event event;

    if (SDL_Init(SDL_INIT_VIDEO) == -1)
    {
        printf("Error: %s\n", SDL_GetError());
        return 1;
    }

    screen = SDL_SetVideoMode(320, 240, 16, SDL_SWSURFACE);

    if (screen == NULL)
    {
        printf("Error: %s\n", SDL_GetError());
        return 1;
    }

    SDL_WM_SetCaption("Ejemplo básico", NULL);

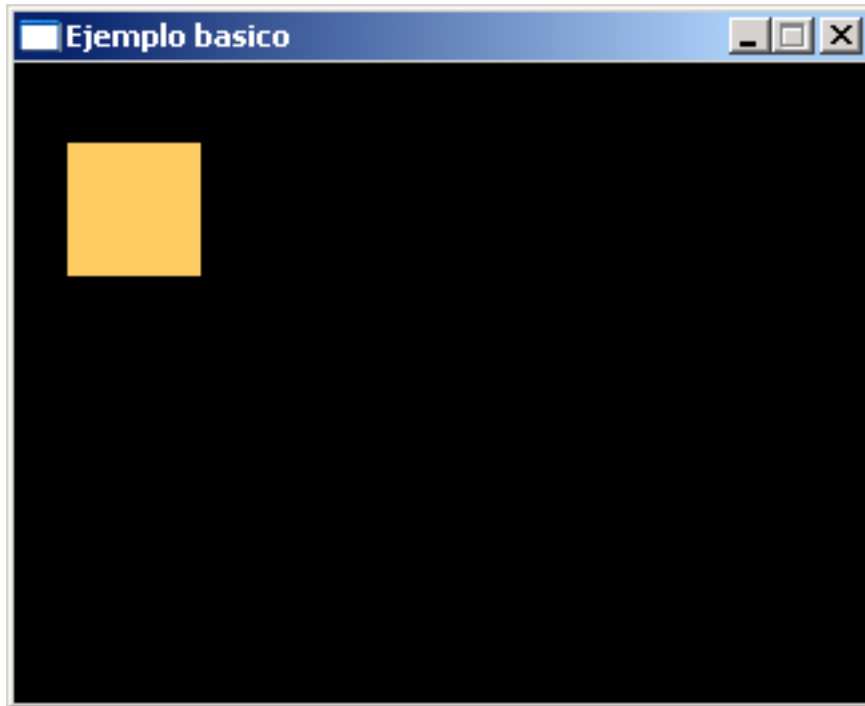
    imprimir_rectangulo(screen, 20, 30);

    while (SDL_WaitEvent(&event))
    {
        if (event.type == SDL_QUIT)
            break;
    }

    SDL_Quit();
    return 0;
}
```

```
void imprimir_rectangulo (SDL_Surface *screen, int x, int y)
{
    SDL_Rect rect = {x, y, 50, 50};
    Uint32 a = SDL_MapRGB(screen->format, 255, 200, 100);

    SDL_FillRect(screen, &rect, a);
    SDL_UpdateRect(screen, rect.x, rect.y, rect.w, rect.h);
}
```



El programa imprime un rectángulo y espera hasta que el usuario cierra la ventana.

Instalar las bibliotecas Auxiliares

Inicialmente, el soporte de formatos de SDL está limitado a los archivos BMP (para las imágenes) y WAV (como formato de audio).

Si queremos ampliar el soporte de formatos para imágenes a PNG, PCX, JPEG etc. podremos utilizar una biblioteca adicional como SDL_Image, y si buscamos ampliar el soporte a otros formatos de audio como MP3, OGG, S3M, Mod etc. podremos utilizar SDL_mixer.

Obtener e Instalar SDL_image y SDL_mixer:

El procedimiento de instalación es similar al de SDL (base), primero obtenemos las versiones de desarrollo y ejecución de cada biblioteca:

SDL_image:

- » Pagina principal: http://www.libsdl.org/projects/SDL_image/
- » Ejecución: http://www.libsdl.org/projects/SDL_image/release/SDL_image-1.2.4-win32.zip
- » Desarrollo: http://www.libsdl.org/projects/SDL_image/release/SDL_image-devel-1.2.4-VC6.zip

SDL mixer:

- » Pagina principal: http://www.libsdl.org/projects/SDL_mixer/
- » Ejecución: http://www.libsdl.org/projects/SDL_mixer/release/SDL_mixer-1.2.6-win32.zip
- » Desarrollo: http://www.libsdl.org/projects/SDL_mixer/release/SDL_mixer-devel-1.2.6-VC6.zip

Nota: Si bien algunas versiones de desarrollo están disponibles en Windows solo para VC, en lugar de Mingw, funcionarán correctamente en este último.

Descomprimos los archivos .zip con las versiones runtime (SDL_mixer-1.2.6-win32.zip y SDL_image-1.2.4-win32.zip) y copiamos todos los archivos .dll contenidos en ambos a la carpeta System dentro de Windows, generalmente "c:\windows\system".

También debemos descomprimir e instalar los archivos .zip de las versiones de desarrollo:

Del contenido de SDL_image-devel-1.2.4-VC6.zip tendremos que copiar el archivo "include/SDL_image.h" a "c:\dev-cpp\include\SDL\" (es importante copiar el archivo dentro de la carpeta SDL). Y el archivo "sdl_image.lib" de la carpeta "lib" se debe copiar a "c:\dev-cpp\lib"

Realizamos lo mismo para SDL_mixer-devel-1.2.6-VC6.zip: copiamos el archivo "SDL_mixer.h" de la carpeta "include" a "c:\dev-cpp\include\SDL\". Y el archivo "SDL_mixer.lib" de la carpeta "lib" a "c:\dev-cpp\lib"

Segundo programa con SDL, SDL_image y SDL_mixer:

Creamos un nuevo proyecto, como en el ejemplo 1, pero ahora añadimos vínculos a las nuevas bibliotecas: seleccionamos "Herramientas -> Opciones de Compilador" ("Tools -> Compiler Options") y agregamos las opciones dentro del campo indicado luego de "Añadir estos comandos a la línea de comandos del linker" ("Add these commands to the linker..."):

```
-lmingw32 -lSDLmain -lSDL -lSDL_image -lSDL_mixer
```

como indica la imagen:



nota: El siguiente ejemplo requiere de 2 archivos: "c:\ima.png" y "c:\cancion.xml". Busque 2 archivos como esos

(el segundo no es necesario para que funcione el programa), copie el siguiente código y genere el programa con F9.

```
#include <stdio.h>
#include <stdlib.h>;
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_mixer.h>

void limpiar (SDL_Surface *screen);
void imprimir(SDL_Surface *screen, SDL_Surface *ima, int x, int y);
void mover_objeto(int *x, int *y, int ancho, int alto);

int main(int argc, char *argv[])
{
    SDL_Surface *screen;
    SDL_Surface *ima;
    Mix_Music *cancion;
    SDL_Event event;
    int salir = 0;
    int x = 100;
    int y = 100;

    if (SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO) == -1)
    {
        printf("Error: %s\n", SDL_GetError());
        return 1;
    }

    screen = SDL_SetVideoMode(320, 240, 16, SDL_HWSURFACE);

    if (screen == NULL)
    {
        printf("Error: %s\n", SDL_GetError());
        return 1;
    }

    if (Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, 2, 1024) == -1)
    {
        printf("Error en Mix_OpenAudio: %s\n", Mix_GetError());
        return 1;
    }

    SDL_WM_SetCaption("Ejemplo 2", NULL);

    ima = IMG_Load("c:/ima.png");

    if (ima == NULL)
```

```
{
    printf("Error en IMG_Load= %s\n", SDL_GetError());
    return 1;
}

cancion = Mix_LoadMUS("c:/cancion.xml");

if (!cancion)
    printf("Error en Mix_LoadMUS: %s\n", Mix_GetError());

if (cancion)
{
    if (Mix_PlayMusic(cancion, -1) == -1)
        printf("Error en Mix_PlayMusic: %s\n", Mix_GetError());
}

while (! salir)
{
    SDL_PollEvent(&event);

    if (event.type == SDL_QUIT)
        salir = 1;

    mover_objeto(&x, &y, ima->w, ima->h);

    limpiar(screen);
    imprimir(screen, ima, x, y);
    SDL_Flip(screen);
    SDL_Delay(10);
}

if (cancion)
{
    Mix_HaltMusic();
    Mix_FreeMusic(cancion);
}

Mix_CloseAudio();
SDL_Quit();
return 0;
}

void limpiar (SDL_Surface *screen)
{
    Uint32 a = SDL_MapRGB(screen->format, 200, 200, 200);
    SDL_FillRect(screen, NULL, a);
}
```

```
void imprimir(SDL_Surface *screen, SDL_Surface *ima, int x, int y)
{
    SDL_Rect rect = {x, y, 0, 0};

    SDL_BlitSurface(ima, NULL, screen, &rect);
}

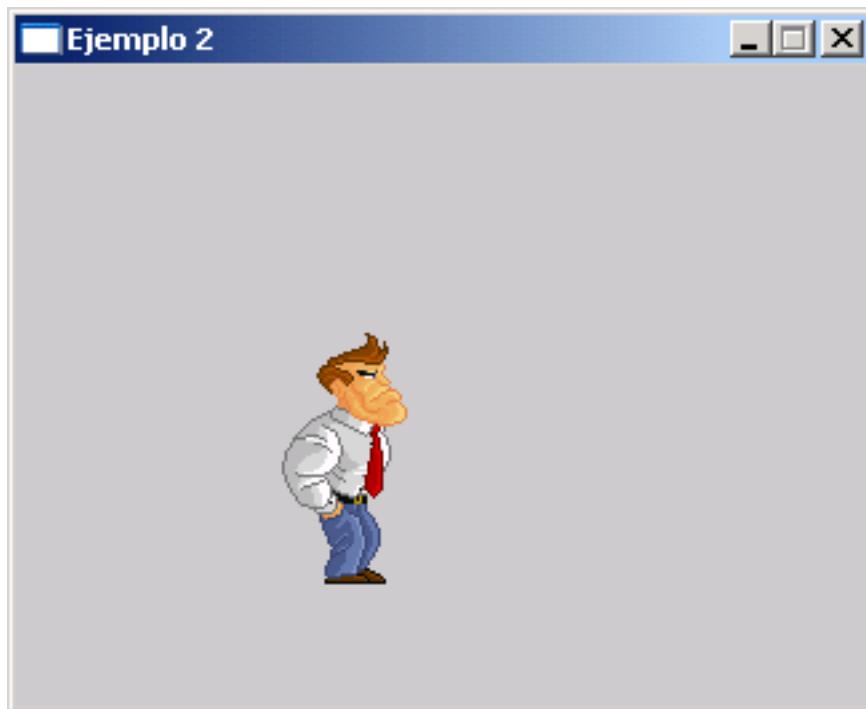
void mover_objeto(int *x, int *y, int ancho, int alto)
{
    Uint8 *key = SDL_GetKeyState(NULL);

    if (key[SDLK_LEFT] && *x > 0)
        (*x) -= 5;

    if (key[SDLK_RIGHT] && (*x + ancho) < 320)
        (*x) += 5;

    if (key[SDLK_UP] && *y > 0)
        (*y) -= 5;

    if (key[SDLK_DOWN] && (*y + alto) < 240)
        (*y) += 5;
}
```



Nota: Los mensajes que se imprimen con printf se almacenan en el archivo stdout.txt dentro de la carpeta del programa (ver más adelante).

Notas:**Compatibilidad:**

Una de las características mas interesantes de SDL es su compatibilidad con varios Sistemas Operativos, no lo desaproveche:

Dev-cpp genera automáticamente un archivo llamado Makefile.win cuando realiza la compilación, este archivo se almacenará en misma carpeta del proyecto.

Los archivos Makefile incluyen reglas que especifican cómo debe generarse el programa. Es recomendable incluir este archivo junto con el código fuente de su programa, eso facilita la tarea de adaptarlo a otros sistemas como GNU/Linux.

Dentro del código en C o C++ no es aconsejable indicar las rutas utilizando barras invertidas "\". Reemplace una ruta como "c:\imagen.jpg" por "c:/imagen.jpg" (como en Unix). Algunos programadores acostumbran a utilizar dos barras, como en "c:\\image.png"; lo cual funcionará sólo en Windows, no adopte esa costumbre, estará desperdiciando las SDL.

Versiones Binarias (.exe):

Tenga en cuenta que la versión compilada de su programa necesita de los archivos .dll de las bibliotecas. Si distribuye su programa puede incluir los archivos .dll en la carpeta principal del proyecto, o bien, pedir a los usuarios que obtengan las bibliotecas SDL por su cuenta y las instalen en "c:\windows\system".

Mensajes de consola:

Si realiza impresiones en la consola, con printf (por ejemplo), los parámetros que añadimos al compilar ("-lmingw32 -lSDLmain ...") redireccionarán la impresión a los archivos stdout.txt y stderr.txt. Estos archivos se generarán en la misma carpeta del proyecto cuando lo ejecute.

Es muy importante respetar el formato de la función principal del programa:

```
int main (int argc, char * argv [])
{
    [...]
}
```

en lugar de utilizar "WinMain" o "main" con otros parámetros como "main (void)" o "void main (char argc, char **argv)".

Dado que los archivos stdout.txt y stderr.txt reemplazan la consola de mensajes, podemos ocultar esa consola añadiendo el parámetro '-mwindows' al vincular el programa. Finalmente nos quedará:

```
-lmingw32 -lSDLmain -lSDL -lSDL_image -lSDL_mixer -mwindows
```

Tambien puede evitar que los mensajes de consola se oculten, para ello debe añadir las siguientes lineas:

```
freopen ("CON", "w", stdout );  
freopen ("CON", "w", stderr );
```

Licencia

Se permite la copia, modificación y distribución de este artículo sólo bajo los términos de la Licencia Creative Commons.

Este documento ha sido generado automáticamente a partir del archivo 'instalar_devcpp.xml' el Wed Jan 28 10:32:20 2009

La versión mas reciente de este documento se almacena en www.losersjuegos.com.ar. Visitenos para obtener mas recursos y actualizaciones.