

A decorative graphic on the right side of the page. It features three sets of concentric circles in shades of blue. The top set is the largest, the middle set is smaller, and the bottom set is the largest again. Thin blue lines intersect the circles, creating a geometric pattern.

# Configurando SFML y Visual Studio

Modelos Y Algoritmos para videojuegos I

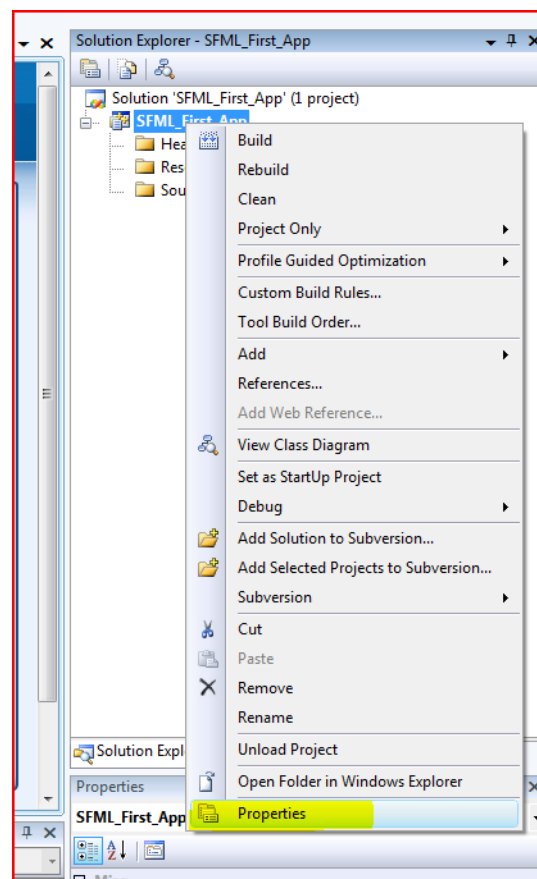
El propósito de este tutorial es explicar brevemente los pasos para configurar Visual Studio para usar SFML.

**Sebastián Rojas Fredini & Emmanuel Rojas Fredini**

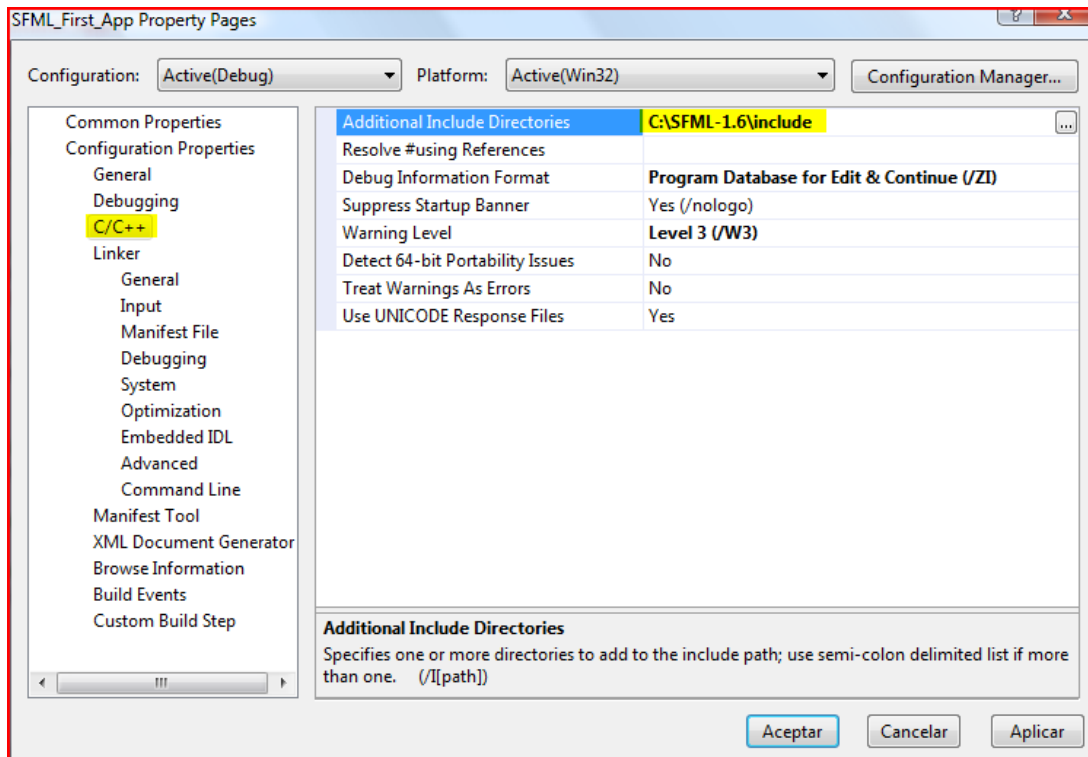
Para trabajar con SFML en Visual Studio, debemos configurar al menos 3 elementos: Directorio de los include, directorio de las librerías y las librerías. Vamos por parte a explicar que es cada uno y cómo se configura. Para ello vamos a crear un proyecto de consola nuevo, como ya se explicó en un tutorial anteriormente.

### **Encabezados:**

En primer lugar tenemos que indicarle a Visual Studio dónde buscar los .h (encabezados con las definiciones de la librería) de SFML. Para ello hacemos click derecho sobre el proyecto en el explorador de soluciones y elegimos propiedades.



Esto nos va a desplegar la ventana de propiedades del proyecto. Para configurar los directorios de búsqueda de .h debemos ir al ítem que dice C++. En donde dice Additional Include Directories ( o su equivalente en español) debemos poner la ubicación de los archivos .h de SFML, si SFML fue descomprimida en C:\ como se explicó en el apunte, entonces la dirección se vería de la siguiente manera:

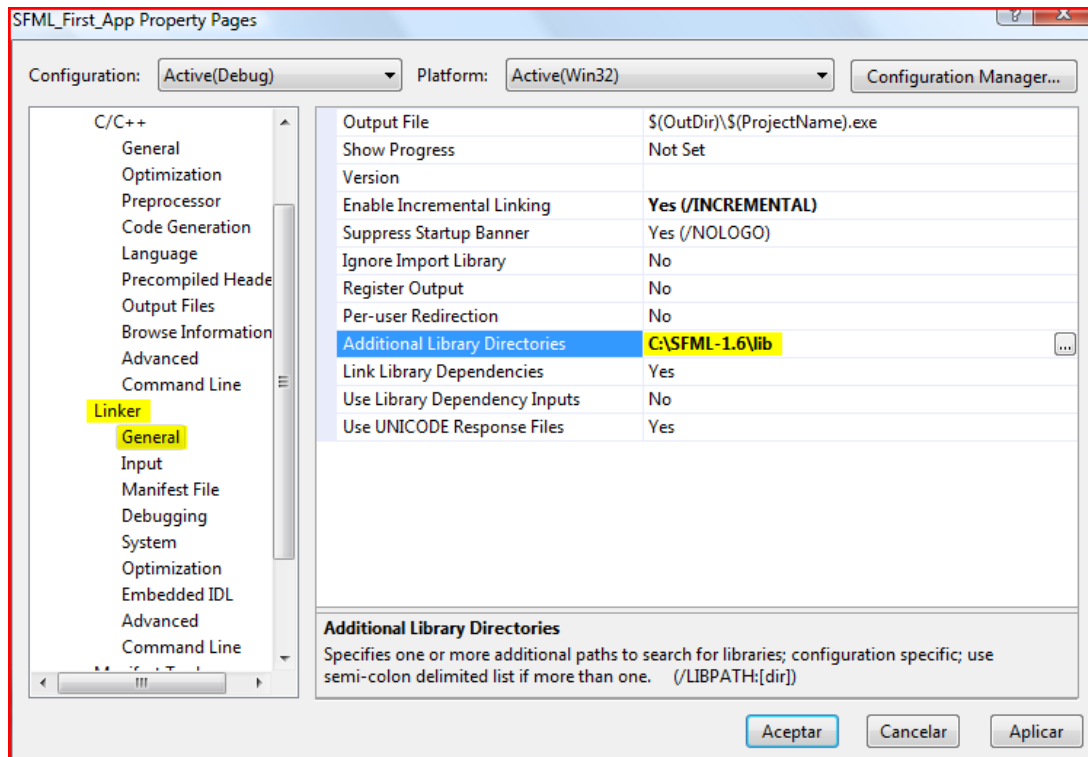


Esto va a permitir que SFML encuentre las definiciones de la librería.

### Librerías:

Ahora debemos indicarle dónde se encuentran los archivos .lib que contienen la implementación de la misma.

Para ello vamos a Linker->General y en additional Library Directories (o su equivalente en español) le damos la dirección dónde se encuentran los archivos .lib de SFML. (la carpeta lib). Si la instalamos en C:\ entonces el path quedaría:



De esta forma Visual C++ ya sabe dónde buscar los archivos .h y los .lib que contienen la implementación de las definiciones de los .h. (Los .lib son generados con los .cpp).

### Estático o Dinámico

SFML tiene dos versiones de sus librerías, una estática y la otra dinámica. La estática incluye todo el código de SFML en el ejecutable de la aplicación. La dinámica no copia el código de SFML dentro del ejecutable, sino que utiliza archivos .dll (deja el código SFML dentro de estos archivos y los enlaza dinámicamente cuando se ejecuta la aplicación). Para poder ejecutar la aplicación vamos a necesitar tener los .dll en el mismo lugar que el ejecutable. La ventaja de esto es que un .dll puede ser usado por varios programas. En cambio si linkeamos estáticamente cada aplicación va a tener dentro todo el código de SFML, generando una repetición de código y tamaño de ejecutables considerable.

### Release o Debug

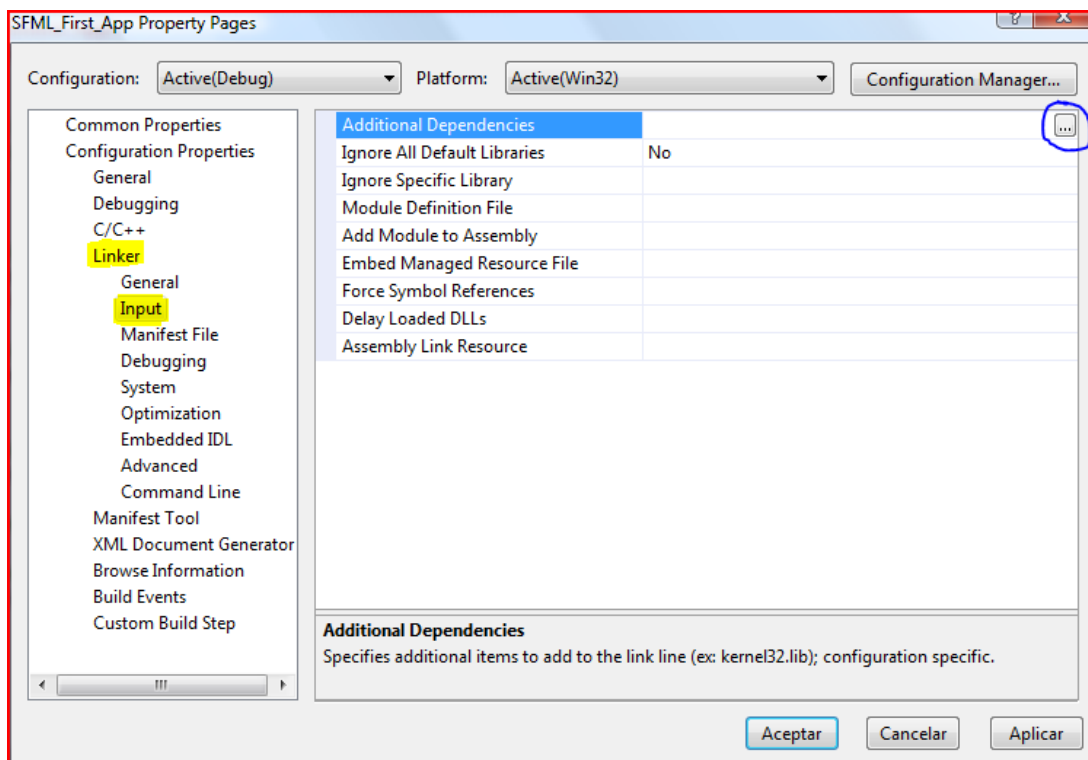
Cuando uno compila un proyecto, puede hacerlo en su versión *debug* (que contiene información de depuración y ninguna optimización) o en su versión *release* (versión sin información de depuración y optimizado). Las propiedades son distintas para cada uno de estos casos, nosotros hasta ahora hemos estado configurando la versión de *debug*. Para cambiar qué versión estamos configurando lo hacemos utilizando la lista desplegable que se encuentra en la esquina superior

izquierda de la ventana de propiedades. Si cambian de *debug* a *release* van a ver que todas las configuraciones que habían seteado para *debug*, no están para *release*.

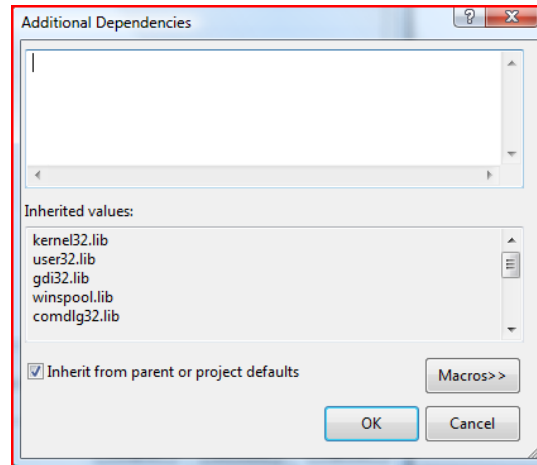
Esto cobra importancia a la hora de decirle qué librería usar (.libs), ya que tiene que haber una correspondencia entre la configuración de nuestra aplicación y los .libs a utilizar. Es decir, si estamos compilando la versión *debug*, entonces vamos a tener que utilizar las librerías de SFML para *debug*. Por otro lado, si estamos compilando para *release*, usaremos las versiones de *release* de SFML.

### Configuración de libs

Para decirle a Visual Studio que libs debe utilizar lo hacemos yendo a Linker->Input como muestra la figura:



Si hacemos click en la zona marcada nos aparecerá una ventana donde podremos ingresar la lista de las librerías a utilizar:



Ahora, veamos cómo SFML estructura los nombres de las librerías. En SFML todas las librerías se llaman sfml-[nombre]-[s]-[d]. El nombre siempre va a estar. La s, puede estar o no e indica si es estático ese .lib. Si no está entonces es dinámico. La d a su vez puede estar o no, e indica si es la versión de debug de la misma o no. Si observan en el directorio lib dentro de SFML van a poder encontrar todas estas versiones que les presentamos. Por ejemplo sfml-graphics-s-d.lib es la librería de gráficos estática en su versión debug, mientras que sfml-graphics-s.lib es la versión estática en su versión release.

No es necesario incluir todos los .lib , sino sólo aquellos que utilizaremos. Para mayor sencillez vamos a utilizar las versiones estáticas de SFML. Por ejemplo lo mínimo que necesitamos para mostrar una ventana son los .lib siguientes:

