

# Manual de instalación de la librería *Allegro 5* para *Microsoft Visual Studio 2015*

Joaquin Olazabal, Dominic Schialer  
Programación I  
UPC

26 de septiembre de 2016

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Instalación y configuración</b>	<b>2</b>
2.1. Instalando Allegro . . . . .	2
2.2. Configurando Allegro . . . . .	3
<b>3. Ejemplos</b>	<b>3</b>
3.1. Ejemplo 1: ¡Bienvenido a Allegro! . . . . .	3
3.2. Ejemplo 2: Visualización de una imagen . . . . .	4

## 1. Introducción

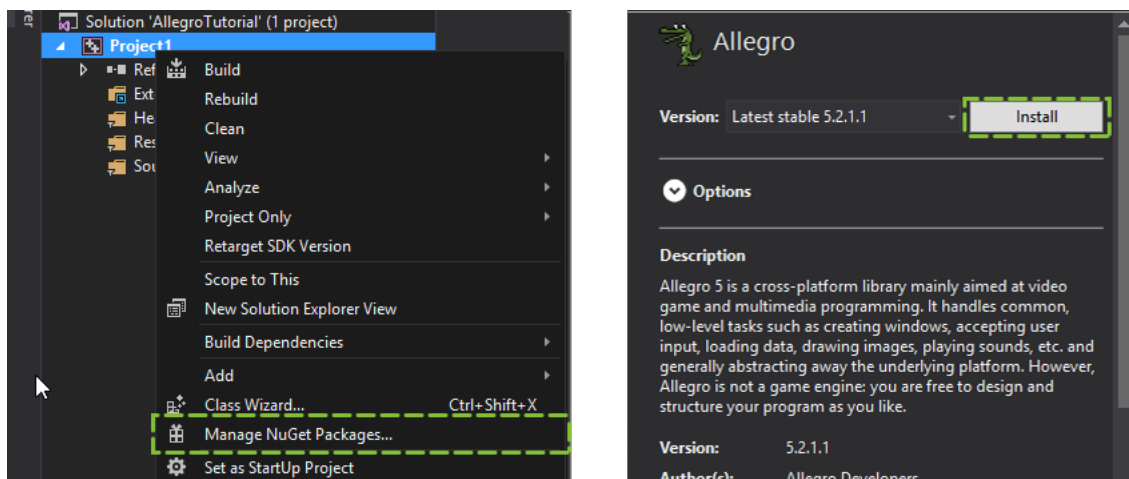
Allegro es una librería multiplataforma dirigida al desarrollo de videojuegos y programas multimedia. Maneja operaciones de bajo nivel como crear ventanas, leer el input del usuario, cargar datos, dibujar imágenes, reproducir sonidos, etc; generalmente abstrayendo la plataforma subyacente. Sin embargo, Allegro *no* es un motor de videojuego (game engine): eres libre de diseñar y estructurar tu programa a tu gusto.

## 2. Instalación y configuración

En Visual Studio seleccione *File>New>Project*, nombre y cree un nuevo CLR Empty Project.

### 2.1. Instalando Allegro

Haga clic derecho en la solución y seleccione *Manage NuGet Packages*(Fig.1). A continuación, seleccione el tab *Browse* e ingrese *Allegro* en el buscador. El paquete principal de Allegro 5 se verá al inicio de la lista. Finalmente presione el botón *Install* (Fig.2). Una vez terminada la instalación deberá ver el mensaje de finalizado en la consola del output (Fig.3).



Figs.1 y 2

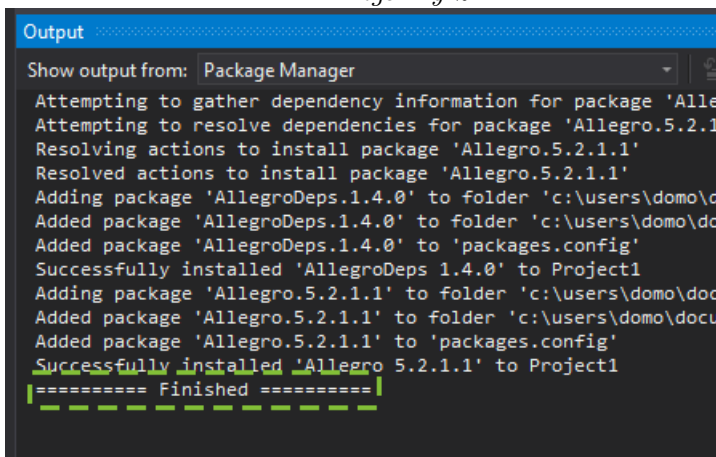
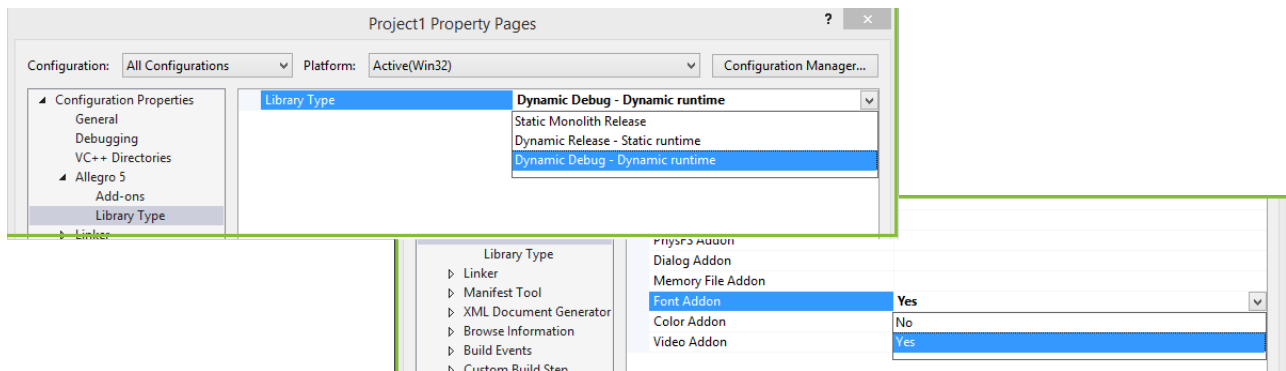


Fig.3

## 2.2. Configurando Allegro

Para el primer ejemplo, proporcionado por [wiki.allegro.cc](http://wiki.allegro.cc), vamos a hacer uso de el add-on de texto de Allegro llamado *Font Addon*. Además configuraremos Allegro para que use debug y tiempo de ejecución dinámicos. La versión de debug le ayudará a verificar que esté usando Allegro de manera correcta, por eso se le recomienda seleccionarla durante el desarrollo.

Haga clic derecho sobre la solución y seleccione *Properties*. A continuación seleccione la entrada *Allegro 5* y en *Library Type* seleccione la opción *Dynamic Debug - Dynamic Runtime* (Fig.4). En la entrada *Add-ons* seleccione *Yes* en *Font Addon* (Fig.5), para usar ese add-on.



Figs.4 y 5

## 3. Ejemplos

### 3.1. Ejemplo 1: ¡Bienvenido a Allegro!

Para verificar que todo funciona, cree un archivo `source.cpp` y copie el siguiente código.

```
#include <allegro5/allegro.h>
#include <allegro5/allegro_font.h>

int main()
{
    al_init();
    al_init_font_addon();
    ALLEGRO_DISPLAY* display = al_create_display(800, 600);
    ALLEGRO_FONT* font = al_create_builtin_font();
    al_clear_to_color(al_map_rgb(0, 0, 0));
    al_draw_text(font, al_map_rgb(255, 255, 255),
                400, 300, ALLEGRO_ALIGN_CENTER,
                "Welcome to Allegro!");
    al_flip_display();
    al_rest(5.0);
    return 0;
}
```

Al ejecutar, este código creará una ventana e imprimirá la frase *Welcome to Allegro!*, que después de 5 segundos se cerrará.

### 3.2. Ejemplo 2: Visualización de una imagen

En el siguiente ejemplo se mostrará como visualizar una imagen en Allegro. Para que pueda mostrar imágenes, se deberá usar el addon llamado *Image Addon*, además de la configuración de *Dynamic Debug - Dynamic Runtime* (véase sección 2.2). Todas las imágenes que sean utilizadas por el código deben de estar en la misma carpeta que el archivo `source.cpp`. De lo contrario Allegro no encontrará la imagen. El output del programa, según el código escrito, es el siguiente:



La imagen `test.png` con las dimensiones 150 x 150 px que se usará en este ejemplo se puede descargar de <http://i.imgur.com/5uoNsJs.png>.

```
#include <allegro5/allegro.h>
#include <allegro5/allegro_image.h>
```

Las librerías necesarias son `allegro.h`, la librería principal de Allegro, y `allegro_image.h`, la librería del addon de imágenes de Allegro.

```
int main()
{
    al_init();
    al_init_image_addon();
```

Se inicializa Allegro y el addon de imágenes de Allegro.

```
    ALLEGRO_DISPLAY *display = NULL;
    ALLEGRO_BITMAP *fondo = al_load_bitmap("test.png");
    display = al_create_display(750, 600);
```

Se crean dos structs: `ALLEGRO_DISPLAY *display`, la ventana en la que el programa va a mostrar el output, y `ALLEGRO_BITMAP *fondo`, el bitmap que el programa usará.

```
al_draw_scaled_bitmap(fondo, 0, 0, 150, 150, 0, 0, 600, 600, NULL);  
for (int i = 0; i < 4; i++)  
    al_draw_bitmap(fondo, 600, (150 * i), NULL);
```

Se dibujan los bitmaps en el buffer. La función `al_draw_bitmap()` tiene como argumentos el puntero, la posición en la que el bitmap va a ser dibujado en la ventana y flags, que pueden volcar el bitmap a lo largo del eje *y* (`ALLEGRO_FLIP_HORIZONTAL`), el eje *x* (`ALLEGRO_FLIP_VERTICAL`) o cualquier combinación de los dos.

En este caso se le asigna un valor nulo (`NULL`), ya que no se va a utilizar ninguna de las dos opciones.

```
al_flip_display();
```

Se 'vuelca' el buffer a la ventana.

```
al_rest(10.0);
```

Se espera 10.0 segundos antes de ejecutar el resto del código.

```
al_destroy_bitmap(fondo);  
al_destroy_display(display);
```

Finalmente se destruyen los structs, liberando así la memoria.

```
    return 0;  
}
```

## Referencias

- [1] Matthew Leverton, Allegro 5.0 reference manual, <https://www.allegro.cc/manual/5/>, ©1999 - 2016, accedido: 25/09/2016
- [2] Allegro Wiki, <https://wiki.allegro.cc>, accedido: 25/09/2016