



## Guía de Prácticas

ASIGNATURA:	Control de Procesos en Tiempo Real		
CENTRO:	Escuela Politécnica de Ingeniería de Gijón		
ESTUDIOS:	Ingeniero Industrial (especialidad Electrónica y Automática)		
CURSO:	5º	CUATRIMESTRE:	1
CARÁCTER:	Optativa	CRÉDITOS ECTS:	7.5
PROFESORADO:	Ignacio Alvarez García, José M <sup>a</sup> Enguita González		

**PRACTICA 01:** Introducción a la programación en C en entorno Visual Studio 2008.

1. Pasos en la generación de un programa ([transparencias funcionamiento computador](#)): edición, compilación, enlazado, carga, ejecución (depuración).
2. Descripción de Entorno Integrado de Desarrollo (IDE): programa que integra a todos los necesarios. Conceptos de solución (Workspace) y proyecto (Project).
3. Realizar los pasos para crear un programa en Visual Studio, modo consola, siguiendo puntos 1 y 2 del documento disponible en la página de enlaces de la asignatura [http://isa.uniovi.es/~ialvarez/Curso/descargas/Crear\\_Proyecto\\_Visual\\_Studio.pdf](http://isa.uniovi.es/~ialvarez/Curso/descargas/Crear_Proyecto_Visual_Studio.pdf). Se utilizará como ejemplo el programa siguiente, cuyo objetivo es generar una alarma si un valor entero supera el doble de otro :

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int x,y;

    x=7;
    y=3;
    if (x>=y*2)
        printf("!!! ALARMA !!! X es superior a Y*2\n");
    else
        printf("Todo va bien\n");
}
```

4. Compilar y enlazar.
5. Comprobar el directorio de la solución y el proyecto, en que aparecen los diversos archivos que forman la solución: .sln, .vcproj, .c, .obj, .exe.
6. El “.exe” obtenido es directamente un programa ejecutable, como cualquier otro, haciendo doble-click sobre él.
7. Ejecutar desde el entorno con la opción Run. Tanto en el caso anterior como en este, el programa se ejecuta y la ventana se cierra inmediatamente, por lo que no se ve el resultado. Solucionar añadiendo al final de main() las líneas:

```
printf("Pulse INTRO para terminar");
fflush(stdin);
getchar();
```

Volver a probar.

8. Generar errores de compilación, recompilar, y determinar cómo buscarlos y corregirlos. No se genera ejecutable si hay errores de compilación:
  - a. Usar `X=7;` (mayúscula en lugar de minúscula)
  - b. Borrar `;` después del primer `printf()` (el error aparece en la línea siguiente).
9. Generar warning de compilación. Aunque el ejecutable se crea, son avisos que pueden esconder errores. Por tanto, hay que evitar todos los warning.
  - a. Quitar línea `x=7;` con lo que x no tiene valor inicial.
10. Generar errores de enlazado, recompilar, y mostrar cómo buscarlos y corregirlos. No se genera ejecutable si hay errores de enlazado.
  - a. Cambiar `printf()` por `Printf()`.
11. Generar errores de ejecución. Disponer de un ejecutable no significa que el programa haga lo que se pide.
  - a. Intercambiar `>=` por `<=` en la comparación.
12. Ejecutar paso a paso para buscar el error, viendo el valor de las variables y corregirlo.
13. Cambiar el valor inicial de x e y por la petición por teclado.

```
printf("Introduzca valor de x: ");
scanf("%d",&x);
```

Atención al `&` del `scanf()`. El porqué ya se explicará más adelante.

Al añadir `scanf()` aparecerá un warning (función insegura) que se quita poniendo al principio:

```
#pragma warning(disable : 4996)
```

14. Modificar el programa para que, en lugar de que la salida sea:

```
!!! ALARMA !!! X es superior a Y*2
```

escriba:

```
!!! ALARMA !!! 7 es superior a 3*2
```

15. Modificar el programa para que sea  $|x|$  quien se compare con  $y*2$ .

Cálculo de  $|x|$ :  $x$  (si  $x \geq 0$ ) ó  $-x$  (si  $x < 0$ ).

Necesidad de una variable auxiliar para el cálculo de  $|x|$ . Necesidad de una estructura de control (`if`) para el cálculo de  $|x|$ .

16. Ejercicios (voluntarios):

- a) Pedir 4 valores, calcular su media, y comprobar si la media está entre 2 límites. Introducir la necesidad de `float` (la media de 4 enteros puede ser un real).