

## Guía de Prácticas

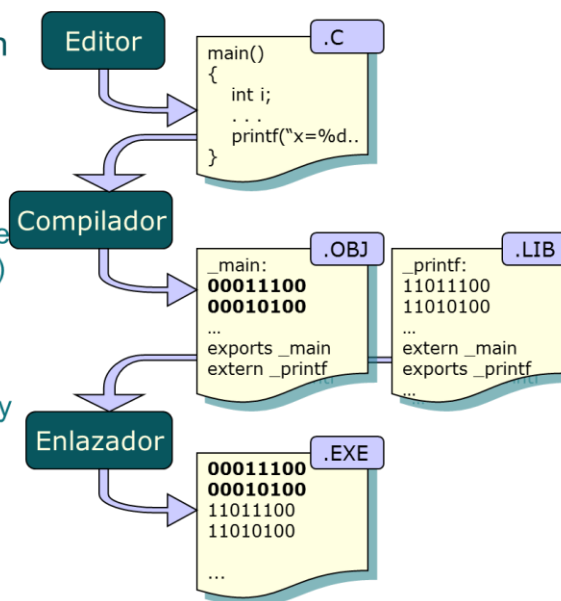
ASIGNATURA: Informática Industrial y Comunicaciones  
CENTRO: Escuela Politécnica de Ingeniería de Gijón  
ESTUDIOS: Grado en Ingeniería Electrónica y Automática  
CURSO: 3º CUATRIMESTRE: 1  
CARÁCTER: Obligatoria CRÉDITOS ECTS: 6  
PROFESORADO: Ignacio Alvarez, José Mª Enguita, Borja Millán

PRACTICA 01: Introducción a la programación en C en entorno Qt Creator

1. Pasos en la generación de un programa:

□ Pasos en la obtención de código máquina:

- Escribir código fuente (programa editor)
- Compilar código fuente (programa compilador)
- Enlazar código objeto y librerías (programa enlazador)



2. Los datos de partida y resultados intermedios del proceso de compilación son archivos en disco, y los programas editor, compilador, etc. leen de archivo de entrada, hacen su trabajo, y escriben archivo de salida. Un integrado de desarrollo (IDE) es un programa que integra a todos los necesarios. Concepto de proyecto (Project).
3. Proponer programa a realizar: se han medido las distancias recorridas por un vehículo en tiempos determinados, obteniéndose la siguiente tabla de valores.

$\Delta t$ (min)	2	3	2	1
$\Delta d$ (Km)	2.75	4	3.25	1.5

Se desea calcular la distancia total, el tiempo total y la velocidad media del vehículo (Km/h) durante su trayecto.

4. Realizar pasos para crear un programa en Qt Creator, modo consola, siguiendo los pasos de <http://isa.uniovi.es/~ialvarez/Curso/descargas/InstalarQt-2022.pdf?version=0001> (aptdo 2).

```

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

int main()
{
    int incT0,incT1,incT2,incT3;
    float incD0,incD1,incD2,incD3;
    int tTotal;
    float dTotal,velMedia;

    incT0=2;
    incT1=3;
    incT2=2;
    incT3=1;

    incD0=2.75;
    incD1=4;
    incD2=3.25;
    incD3=1.5;

    tTotal=(incT0+incT1+incT2+incT3); // Tiempo total = suma de tiempos
    dTotal=(incD0+incD1+incD2+incD3); // Distancia total = suma de distancias
    velMedia=dTotal/tTotal; // velocidad = distancia entre tiempo

    // Escribir resultados
    printf("Tiempo total = %d \n",tTotal);
    printf("Distancia total = %f \n",dTotal);
    printf("Velocidad media = %f \n",velMedia);

    return 0;
}

```

5. Recordar los elementos del programa (diapositiva 28 del documento [http://isa.uniovi.es/~ialvarez/Curso/descargas/lenguaje\\_c.pdf](http://isa.uniovi.es/~ialvarez/Curso/descargas/lenguaje_c.pdf) ). Introducir concepto de printf (diapo 26 del mismo documento).
6. Compilar y enlazar.
7. Mostrar el directorio de la solución y el proyecto, presentando los diversos archivos que aparecen: .pro, .c, .o, .exe.
8. El “.exe” obtenido es directamente un programa ejecutable, como cualquier otro, se podría ejecutar haciendo doble-click sobre él (pero hay que tener el PATH del sistema actualizado).
9. Ejecutar desde el entorno con la opción Run (previamente, activar Projects → Run → Run In Terminal. El resultado debe ser:

```

Tiempo total = 8
Distancia total = 11.500000
Velocidad media = 1.437500
Press <RETURN> to close this window..

```

10. Mejorar la salida del programa: los valores de las magnitudes (tiempo, distancia, velocidad) ¡¡¡¡ deben figurar con unidades !!!! Comproba que las unidades de la velocidad media no son las "habituales". Corregir el programa para que así sea. El resultado debe ser:

```

Tiempo total = 8 min
Distancia total = 11.500000 Km
Velocidad media = 86.250000 Km/h
Press <RETURN> to close this window..

```

11. Generar errores de compilación, recompilar, y mostrar cómo buscarlos y corregirlos. Insistir en que no se genera ejecutable si hay errores:
  - a. Modificar `incd0=2.5;` (D minúscula en lugar de mayúscula)
  - b. Borrar `;` después del primer `printf()` (el error aparece en la línea siguiente).
12. Generar warning de compilación. Aunque el ejecutable se crea, puede contener errores. Por tanto, hay que evitar todos los warning.
  - a. Quitar línea `incD0=2.5;` con lo que `incD0` no tiene valor inicial y aparece un warning.
13. Generar errores de enlazado, recompilar, y mostrar cómo buscarlos y corregirlos. No se genera ejecutable si hay errores.
  - a. Cambiar `printf()` por `Printf()` (mayúscual). Aparecerá un warning también.
14. Generar errores de ejecución. Disponer de un ejecutable no significa que el programa haga lo que se pide.
  - a. Eliminar uno de los sumandos el cálculo de la velocidad media.
  - b. Intercambiar `%f` por `%d` al escribir `velMedia`.
15. Mejorar el código inicializando variables en la declaración.
16. Mejorar el código utilizando `#define` y nombres más adecuados:

```
#define MIN_POR_HORA 60
...
int tTotal_min;
float dTotal_Km,velMedia_Km_h;
velMedia_Km_h=dTotal_Km/tTotal_min*MIN_POR_HORA;
```

17. Cambiar los valores iniciales de `incT...` e `incD...` por peticiones por teclado.

```
printf("Introduzca incremento de tiempo: ");
scanf("%d",&incT0);
```

Ojo al `&` necesario en el `scanf()` (no en el `printf`). El por qué ya se explicará más adelante.

Ahora el mismo programa sirve para valores distintos de las variables de partida, antes había que recompilar (generar otro programa) para cada uno.

### **AMPLIACIONES PROPUESTAS:**

- I) Calcular la media de las velocidades de cada tramo, y comprobar que el resultado es distinto que la velocidad media.
- II) Solicitar un valor adicional por teclado (desviación máxima respecto a la media de velocidades). Calcular las velocidades de cada tramo, y escribir un aviso para aquellas velocidades de tramo que se diferencian de la media más que esa desviación. Requiere uso de `if` (se verá en la siguiente clase de teoría).
- III) Lo mismo que el anterior, pero para aquellas velocidades de tramo que se separan más de  $\pm\sigma$  de la media de velocidades.

Desviación típica 
$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Se requiere el uso de `sqrt()` para el cálculo de raíz cuadrada, para lo cual se precisa:

```
#include <math.h>
```