

Entornos de programación integrados

Temporización

2 horas (2 Presenciales + 0 No presenciales)

Objetivos

- Manejar un entorno integrado de desarrollo de programas en lenguaje C y usarlo para desarrollar programas.
- Ejecutar los programas desarrollados bien desde el sistema operativo o bien desde el soporte de ejecución del entorno integrado.
- Conocer las principales prestaciones del editor integrado.
- Identificar algunos elementos sintácticos del lenguaje de programación C y corregir errores en la fase de compilación.
- Presentar adecuadamente los resultados de salida de un programa mediante el formateo de salida de datos numéricos y textuales.

Competencias a desarrollar

- ☒ RD1: Poseer y comprender conocimientos
- ☒ RD2: Aplicación de conocimientos
- ☒ UAL1: Conocimientos básicos de la profesión
- ☒ UAL3: Capacidad para resolver problemas
- ☒ UAL6: Trabajo en equipo
- ☒ FB3: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en la ingeniería.

Tareas a realizar

- Formar el equipo de trabajo
- Inicializar los entornos de trabajo - Cuenta de github
- Revisar y probar los ejemplos incluidos
- Realizar los ejercicios 1 al 6 de esta ficha y dejar los resultados en el repositorio creado.

Formación del equipo de trabajo - Aprendizaje Cooperativo

Nótese que esta competencia es transversal a la competencia específica de la asignatura (programación), por lo que un indicador importante a tener en cuenta será el nivel competencial alcanzado por cada miembro del equipo en esta última, el cual es evaluado de forma individual mediante una prueba escrita, utilizándose las calificaciones de dichas pruebas como un indicador cuantitativo de los objetivos de la competencia de trabajo en equipo de acuerdo con la siguiente fórmula: $\text{Mínimo}(\text{nota1}, \text{nota2}, \text{nota3}, \text{nota4})/5$

¿Qué es el aprendizaje cooperativo?

Es una forma de trabajo en grupo basado en la construcción colectiva del conocimiento y el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del grupo es

responsable tanto de su aprendizaje como del de los restantes miembros del grupo.

Los 5 ingredientes del aprendizaje cooperativo son:

Interdependencia positiva. Las tareas están estructuradas de manera que los estudiantes se necesitan entre sí para completarlas

“O nos salvamos todos o nos hundimos juntos”

Responsabilidad individual. Cada estudiante debe ser responsable de su trabajo y del resto de sus compañeros

¡No vale descansar en el trabajo de otros!

Interacción cara a cara. Para que los miembros promuevan las explicaciones y discusiones

“Cuando enseñas aprendes dos veces”

Habilidades interpersonales y de grupo. Conocer y confiar en el otro, comunicarse acertadamente, resolver conflictos constructivamente

“Pagaré más por la capacidad de tratar con la gente que por cualquier otra habilidad que exista”

Reflexión del grupo. ¿Qué fue útil? ¿Qué se puede mejorar?>

Había una vez...

Un equipo de alumnos tenía cuatro miembros. Se llamaban **Todo el Mundo**, **Alguien**, **Cualquiera** y **Nadie**. Había un trabajo importante por hacer. **Todo el Mundo** estaba seguro de que **Alguien** lo haría. **Cualquiera** podría haberlo hecho, pero al final **Nadie** lo hizo. **Alguien** se molestó ante este hecho, pues era un deber de **Todo el Mundo**. **Todo el Mundo** pensó que **Cualquiera** podía hacerlo, pero **Nadie** comprendió que al final **Todo el Mundo** no lo haría. En conclusión, **Todo el Mundo** culpó a **Alguien** cuando **Nadie** hizo lo que **Cualquiera** podía haber hecho.

G. Gibbs, “Aprendiendo en Equipos”

GRUPO DE PRACTICAS: XXXXX

Alumno Apellidos y nombre DNI

Alumno	Apellidos y nombre	DNI
1		
2		
3		
4		

Edición, compilación y ejecución de un programa

Abrir el archivo de nombre SALUDO.c

Completarlo para que su contenido sea:

```
#include <stdio.h>

/* Programa inicial */
int main(){
    printf(Hola.\n");
    printf("Este es mi primer programa\n");
    printf("escrito en C de la carrera.\n");
    printf("Adios.\n");
    printf("\n\n\nPulse una tecla para finalizar");
    getchar();
    return 0;
}
```

Compilar el programa

Ejercicio 1. Modificación del archivo de Saludo

Cargar en el editor el programa SALUDO.C creado en el ejemplo de sesión de trabajo. Modificar el mensaje de salida, de forma que ahora escriba:

```
Aqui estoy de nuevo.
Este es mi segundo programa de la asignatura.
Me he limitado a cambiar el texto de una orden de escritura.
```

- Grabar dicha modificación en un archivo con el nombre ejercicio1.c,
- Compilar dicho archivo creando un archivo ejecutable en disco.
- Descargar el archivo ejercicio1.exe si trabaja con el IDE online.
- Ejecutar desde el sistema operativo dicho programa.

Ejercicio 2. Saludo Formal

Copie lo siguiente en el editor del IDE y guárdelo en el disco de prácticas con el nombre ejercicio2.c; tras comprobar que es ejecutable, realice los pasos que se indican al final del ejercicio

```
/* Ejercicio 2 del Trabajo cooperativo 1 */
#include <stdio.h>
int main(){
    char nombre[30];
    printf("Por favor, introduzca su nombre: ");
    scanf(" %[^\\n]s", nombre);
    printf("\\n\\nSaludos D. %s", nombre);
    printf("\\nBienvenido al fantastico mundo de la programacion");
    printf("\\n\\nPulse una tecla para finalizar");
    getchar();
    return 0;
}
```

- Seleccione como bloque las líneas 5 y 6 del programa.
- Copie el bloque inmediatamente debajo.
- Cambie el identificador nombre, por apellidos en la copia.
- Seleccione como bloque la 4 línea del programa.
- Cópielo inmediatamente debajo (queda como línea 5).
- Cambie el identificador nombre, por apellidos en la nueva línea.
- Elimine la línea 10 utilizando una única orden.
- Inserte una nueva línea en la posición que tenía la que acaba de borrar:

```
printf("\\n\\nSaludos D. %s %s", nombre, apellidos);
```

Compruebe que la nueva versión del programa funciona correctamente y grábela.

Ejercicio 3. Calculadora básica

Entrar en el editor del IDE y copiar las siguientes líneas:

```
/* Ejercicio 3 */

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

int main(){
```

```

int a, b;
char operacion, c;

do{
    printf("Vamos a realizar una operacion aritmetica simple:\n\n");
    printf("Introduzca el primer valor con que vamos a operar: ");
    scanf(" %d",&a);
    printf("Introduzca el segundo valor con que vamos a operar: ")
    scanf(" %d",&b);
    printf("\n");
    printf("Introduzca + para sumar, - para restar o * para multiplicar: ");
    scanf(" %c",&operacion);
    switch(operacion){
        case '+': printf("El resultado es: %d\n", a+b);
                  break;
        case '-': printf("El resultado es: %d\n", a-b);
                  break;
        case '*': printf("El resultado es: %d\n", a*b);
                  break;
        default:  printf("Operacion incorrecta\n");
                  break;
    }
    printf("\nDesea efectuar una nueva operacion (S/N)? ");
    scanf(" %c",&c);
}while ((c!='N') && (c!='n'));
return 0;
}

```

Compile el programa anterior, que se habrá almacenado previamente como ejercicio3. Aparecerá un error al compilarlo:

En este caso, se trata de que en la línea 18, una antes del punto en que encuentra el error, falta un punto y coma. Corrija el error, colocando el punto y coma al final de la instrucción:

```
printf("Introduzca el segundo valor con que vamos a operar: ");
```

Vuelva a compilarlo y una vez que esté sin errores, ejecútelo. Pruebe con los valores que se le indican a continuación la corrección del programa y anote en la siguiente tabla los valores de prueba utilizados y el resultado obtenido.

Valor 1	Valor 2	Código de operación	Resultado
10	10	+	
10	10	*	
-50	50	*	
-50	0	*	
0	0	-	

Valor 1	Valor 2	Código de operación	Resultado
0	0	*	
10	10	+	
20000	20000	+	
50000	50000	*	

¿Ha observado algún resultado anómalo? ¿Cuál puede ser la causa del mismo?

Ejercicio 4. Teclee el programa siguiente en el editor:

/* Ejercicio 4 */

```
#include <stdio.h>

int main(){
    int edad;

    printf("Por favor, introduzca su edad: ");
    scanf(" %d", &edad);
    printf("\nPara una edad de %d años\n",edad)
    printf("una lectura adecuada puede ser: ");
    if(edad>60)
        printf("\nLa historia de Roma\n\n");
    else if(edad>40)
        printf("El nombre de la rosa\n");
    else if(edad>18)
        printf("Fundamentos de Programacion\n");
    else if(edad>15)
        printf("Aventuras en los mares del sur\n");
    else if(edad>10)
        printf("Harry Potter\n");
    else printf("Cuentos\n");
    printf("\n\nPulse una tecla para finalizar");
    getch();
    return 0;
}
```

Compile el programa anterior. Si aparecen errores tendrá que corregirlos y volver a compilar el programa. Cuando el programa quede sin errores, ejecútelo desde el entorno integrado. Seguidamente genere un ejecutable y ejecútelo desde el sistema operativo.

Realice un listado de todos los errores sintácticos que haya detectado en los programas fuente de esta práctica y anótelos en la siguiente tabla, indicando cómo los ha corregido:

Error sintáctico Corrección del error

Error sintáctico	Corrección propuesta
------------------	----------------------

Error sintáctico Corrección propuesta

Ejercicio 5. Teclee el siguiente programa en el editor y analice los resultados que aparecen en la pantalla:

```
/* Ejercicio 5 */

#include <stdio.h>

int main(){
    char nombre[30];

    printf("Pruebas de formatos de salida\n");
    printf("=====\n");
    printf("\nDiferentes formatos de salida para el texto:\n\n");
    printf("%s\n", "Esta es una linea de texto de prueba");
    printf("%50s\n", "Esta es una linea de texto de prueba");
    printf("%-50s\n", "Esta es una linea de texto de prueba");
    printf("%-50.10s\n", "Esta es una linea de texto de prueba");
    printf("\nDiferentes formatos de salida para numeros enteros:\n");
    printf("\n1)\n%d\n%d\n%d\n%d\n",8,1234,23,12000);
    printf("\n2)\n%10d\n%10d\n%10d\n%10d\n",8,1234,23,12000);
    printf("\nDiferentes formatos de salida para numeros reales:\n");
    printf("\n1)\n%f\n%f\n%f\n",123.89,-4.0,2345.8999);
    printf("\n2)\n%15f\n%15f\n%15f\n",123.89,-4.0,2345.8999);
    printf("\n3)\n%15.2f\n%15.2f\n%15.2f\n",123.89,-4.0,2345.8999);
    printf("\n\nPulse una tecla para finalizar");
    getchar();
    return 0;
}
```

Ejercicio 6. Intente averiguar el significado del control del formato de salida que se está utilizando en las diferencias instrucciones printf.

```
printf("cadena de control", lista de argumentos);
```

¿Para qué se utiliza el símbolo \n que aparece en bastantes ocasiones en la cadena de control de la instrucción printf?

RESPUESTA:

¿Qué relación encuentra entre las secuencias de salida que aparecen en la cadena de control (secuencias de caracteres que empiezan con el símbolo % y acaban en s, d ó f) y los argumentos de la función printf?

¿Encuentra alguna relación lógica entre dichas letras y los correspondientes argumentos?

RESPUESTA:

Explique el efecto que producen sobre el texto de salida los siguientes formatos:

%s

RESPUESTA:

%50s

RESPUESTA:

%-50s

RESPUESTA:

%-50.10s

RESPUESTA:

Explique el efecto que producen sobre la presentación de los números enteros los siguientes formatos:

%d

RESPUESTA:

%10d

RESPUESTA:

Explique el efecto que producen sobre la presentación de los números reales los siguientes formatos:

%f

RESPUESTA:

%15f

RESPUESTA:

%15.2f

RESPUESTA:

¿Cuál piensa que puede ser el resultado de la ejecución de la siguiente instrucción? Pruébela copiando dicha línea en cualquiera de los programas de esta práctica y compruebe si coincide con lo que pensó.

```
printf("Resultado de %d + %d = %f\n",1,1,1+1);
```

RESPUESTA: