

 71901020		Fundamentos de Programación			901	
		MULTIPLES GRADOS				71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo A Mixto	Nacional 1º Cuatrimestre	
Material: Ninguno					Hoja 1 de 3	

FUNDAMENTOS DE PROGRAMACION	MODELO 1	CONV. ORD. 25/26
GRADO DE INFORMÁTICA y GRADO DE TECNOLOGÍAS DE LA INFORMACIÓN-ETSII-UNED		
MATERIAL AUTORIZADO: NINGUNO	DURACIÓN: 2 HORAS	
INSTRUCCIONES:	1) El test debe ser contestado en la hoja de marcas. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta. 2) El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 preguntas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio. 3) Cada respuesta correcta 1 pto. Respuesta incorrecta, doble o en blanco: 0 pto. 4) Puede quedarse, si lo desea, con esta hoja de examen.	

CUESTIONES DE TEST

1. En C±, el identificador diasemana y DIASEMANA:

- A. son el mismo identificador válido
- B. son válidos y distintos
- C. el identificador con todo en mayúsculas no es válido
- D. no pueden declararse en el mismo ámbito

2. En la declaración:

Tipo1 Tipo2 = {1, 2, 3};

- A. Tipo1 es una variable y Tipo2 es un tipo
- B. Ambos son tipos
- C. Ambos son variables
- D. Tipo1 es un tipo de dato y Tipo2 es una variable

3. En C±, la expresión lógica:

!a || b && c < d

Es equivalente a:

- A. (!a) || (b && (c<d))
- B. (!a || b) && (c<d)
- C. !(a || b) && (c<d)
- D. (!a) || ((b && c) <d))

4. Una función es pura si:

- A. produce efectos laterales y todos sus argumentos son por valor
- B. no produce efectos laterales y tiene algún argumento por referencia
- C. no produce efectos laterales y todos sus argumentos son por valor
- D. produce efectos laterales y tiene algún argumento por referencia

5. En C±, los elementos de un vector:

- A. pueden ser de distinto tipo y el tipo puede ser cualquiera
- B. son todos del mismo tipo y el tipo ha de ser escalar (int, char y float)
- C. son todos del mismo tipo y el tipo puede ser cualquiera
- D. son todos del mismo tipo y el tipo ha de ser escalar (int, char y float) o struct

 71901020		Fundamentos de Programación			901	
		MULTIPLES GRADOS				71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo A Mixto	Nacional 1° Cuatrimestre	
Material: Ninguno					Hoja 2 de 3	

6. Si tenemos la definición de los tipos enumerados:

```
typedef enum TipoTres { Tercero, Segundo, Primero };
typedef enum TipoLetras { u,o,i,e,a };
```

Entonces la asignación:

```
valor = (int(e)-int(Tercero))/(int(u)+1);
```

A la variable valor de tipo entero es:

- A. 2
- B. 4
- C. 1
- D. 3

7. Si tenemos el procedimiento:

```
void miProc(int & miArg){
    miArg = miArg * 2;
}
```

Y hacemos la llamada:

```
A = 4;
miProc(A);
vfinal = A * 2;
```

¿Cuál sería el valor de la variable "vfinal"?

- A. 8
- B. 16
- C. 4
- D. 32

8. ¿Qué imprime el siguiente código?

```
int i = 7;
do {
    try {
        if (i%2==0) {
            throw 1;
        } else {
            throw 0;
        }
    } catch (int e) {
        printf("%d ", e*2);
    }
    i=i-1;
} while (i > 0);
```

- A. 0202020
- B. 2020202
- C. 02020
- D. 20202

 71901020		Fundamentos de Programación			901
		MULTIPLES GRADOS			71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo A Mixto	Nacional 1° Cuatrimestre
Material: Ninguno				Hoja 3 de 3	

9. La notación Backus-Naur Form (BNF) especifica:

- A. el orden de complejidad de un algoritmo
- B. la memoria que consume un programa
- C. la semántica de un lenguaje
- D. la sintaxis de un lenguaje

10. En C±, en la lista de ítems que puede incluirse en un tipo registro no se puede añadir:

- A. un campo puntero
- B. la definición de otro tipo
- C. la definición de una cabecera de subprograma
- D. un campo array

EJERCICIO DE PROGRAMACIÓN

Escriba un Tipo Abstracto de Datos (TAD) para gestionar un catálogo de hasta 100 viviendas en una inmobiliaria. Cada vivienda tiene una dirección (una cadena de caracteres), un número de habitaciones, un precio y puede estar disponible o no.

Implemente las siguientes operaciones:

- **AgregarVivienda:** agrega una nueva vivienda al catálogo, siempre que este no esté lleno. Si el catálogo está lleno, devuelve falso; de lo contrario, devuelve verdadero.
- **FiltrarPorNumeroHabitacionesYPrecio:** imprime los datos de las viviendas que tengan desde un número mínimo de habitaciones y hasta un precio máximo.
- **OrdenarPorPrecio:** ordena las viviendas del catálogo por precio.
- **ListarPorDisponibilidad:** imprime los datos de todas las viviendas disponibles. Las viviendas estarán ordenadas por precio y para ello el procedimiento invocará a "OrdenarPorPrecio".