

Índice

▼ Índice

- Ejercicio 1. Número par o impar
- Ejercicio 2. Conversión de temperatura
- Ejercicio 3. Calculadora de salario
- Ejercicio 4. Promedio de calificaciones
- Ejercicio 5. Calculadora básica
- Ejercicio 6. Comparación de números
- Ejercicio 7. Adivina el número
- Ejercicio 8. Verificación de contraseña
- Ejercicio 9. Coste de vacaciones
- Ejercicio 10. Semáforo textual
- Ejercicio 11. Clasificador de edad
- Ejercicio 12. Calculo de notas
- Ejercicio 13. Clasificador de triángulos

Ejercicios Unidad 5

[Descargar estos ejercicios](#)



Antes de empezar

Para realizar estos ejercicios, deberás descargar los recursos del enlace [proyecto_condicionales](#). Cada ejercicio se codificará en el cuerpo del método que corresponda con el nombre del ejercicio. Es decir, sustituirás las líneas `//TODO:` por el código de solución del ejercicio. Para probar el funcionamiento correcto de los ejercicios, deberás pasar los Test adjuntos a este proyecto (como se explica en la unidad 4).

Ejercicio 1. Número par o impar

Escribe un programa que solicite al usuario un número entero y determine si es par o impar.

```
Ejercicio 1: Número par o impar
Introduce un número entero: 6
El número 6 es par.
```

Requisitos:

- Usa el condicional `if - else` para decidir la operación.
- Usa el operador módulo `%` para verificar si el número es divisible entre 2.
- Muestra un mensaje indicando si el número es par o impar.

Ejercicio 2. Conversión de temperatura

Escribe un programa que convierta una temperatura de grados Celsius a Fahrenheit y viceversa. El usuario debe ingresar la temperatura y el tipo de conversión que desea realizar.

```
Ejercicio 2: Conversión de temperatura
Introduce la temperatura: 35
Convertir a (F)ahrenheit o (C)elsius? F
35°C son 95,00°F
```

Fórmulas:

- De Celsius a Fahrenheit: $F = C * 9/5 + 32$
- De Fahrenheit a Celsius: $C = (F - 32) * 5/9$

Requisitos:

- Usa `char.Parse()` para convertir la cadena de entrada del tipo de conversión a `char`.
- Usa el condicional `if - else` para decidir la operación.
- Usa operadores aritméticos para realizar las conversiones.
- Muestra el resultado con dos decimales.

Ejercicio 3. Calculadora de salario

Crea un programa que calcule el salario semanal de un empleado. Solicita al usuario las horas trabajadas y el pago por hora. Si el empleado trabaja **más de 40 horas**, las horas extra se pagan al 50% adicional.

```
Ejercicio 3: Calculadora de salario
Introduce las horas trabajadas: 45
Introduce el pago por hora: 11
El salario semanal es: 522,50 ?
```

Requisitos:

- Usa el condicional `if - else` para decidir la operación.
- Usa operadores aritméticos para calcular el salario base y las horas extra.
- Muestra el salario total.

Ejercicio 4. Promedio de calificaciones

Escribe un programa que solicite al usuario tres calificaciones, calcule el promedio y determine si el estudiante aprobó o suspendió. Considera que la calificación mínima aprobatoria es de 5.

```
Ejercicio 4: Promedio de calificaciones
Introduce la primera calificación: 4
Introduce la segunda calificación: 7
Introduce la tercera calificación: 6
El promedio es 5,67. Resultado: Aprobado
```

Requisitos:

- Usa operadores aritméticos para calcular el promedio.
- Usa el condicional de expresión o operador ternario `?:` para determinar si aprobó o suspendió.

Ejercicio 5. Calculadora básica

Crea un programa que solicite al usuario dos números y un operador (+, -, *, /). Según el operador ingresado, realiza la operación correspondiente y muestra el resultado.

```
Ejercicio 5: Calculadora básica
Ingresa el primer número: 6
Ingresa el segundo número: 3
Ingresa un operador (+, -, *, /): *
Resultado: 18
```

Requisitos:

- Usa `char.Parse()` para convertir la cadena de entrada del operador a tipo `char`.
- Utiliza una estructura `switch` para manejar los operadores con la expresión de tipo `char`.
- Maneja la división entre cero con un mensaje de error.

Ejercicio 6. Comparación de números

Crea un programa que solicite al usuario dos números y determine cuál es mayor, menor o si son iguales.

```
Ejercicio 6: Comparación de números
Ingrese el primer número: 4
Ingrese el segundo número: 5
4 es menor que 5.
```

Requisitos:

- Usa operadores relacionales (`>`, `<`, `==`) para realizar las comparaciones.
- Usa el condicional `if - else` para decidir la operación.

Ejercicio 7. Adivina el número

Escribe un programa que genere un número aleatorio entre 1 y 100. El usuario debe adivinar el número, y el programa debe indicar si el número ingresado es mayor, menor o igual al número generado.

```
Ejercicio 7: Adivina el número
Adivina el número entre 1 y 100.
Ingresa tu número: 56
El número es mayor.
```

Requisitos:

- Usa la clase `Random` para generar el número aleatorio.
- Usa operadores relacionales para comparar los números.

Ejercicio 8. Verificación de contraseña

Escribe un programa que solicite al usuario una contraseña y verifique si coincide con una contraseña predefinida en una **constante**.

```
Ejercicio 8: verificación de contraseña
Ingrese la contraseña: pedro
Contraseña incorrecta.
```

Requisitos:

- Usa operadores de comparación (`==`) para verificar la contraseña.
- Crea una constante con el modificador `const` para la contraseña base (revisa como debe ser el identificador de una constante).
- Muestra un mensaje indicando si la contraseña es correcta o incorrecta.

Ejercicio 9. Coste de vacaciones

Una agencia de viajes utiliza para calcular el coste de unas vacaciones **dos** premisas: el tipo de vacaciones y la duración en días.

Las vacaciones **tipo A** cuestan **250€** por día y las vacaciones **tipo B** cuestan **150€** por día. Las vacaciones incluyen el coste por día más un plus único por cargo de servicio de **50€**.

Realiza un programa que solicite el tipo de vacaciones y el número de días y visualice en pantalla el total a pagar.

```
Ejercicio 9: Coste de vacaciones
Introduce el tipo de vacaciones (A o B): A
Introduce el número de días: 5
Total a pagar: 1.300,00 ?
```

Requisitos:

- Se deberá introducir el carácter `'A'` o `'B'` como entrada.
- Estos caracteres pueden ser introducidos en mayúsculas o en minúsculas.
- Se usará el método `Toupper()` de la clase `Char` para pasar la entrada a mayúsculas.

Ejercicio 10. Semáforo textual

Solicita al usuario que escriba un color del semáforo: "rojo", "amarillo" o "verde", y según la entrada, muestra una instrucción:

Info

Rojo → Detente

Amarillo → Precaución

Verde → Avanza

Cualquier otro → Color no válido

Ejercicio 10: Semáforo textual

Escriba un color del semáforo (rojo, amarillo, verde): amarillo
Precaución

Requisitos:

- Transforma la entrada a minúsculas para estandarizarla.
- Usa una `expresión switch` sobre cadenas.
- Muestra la acción correspondiente que devuelve la expresión.

Ejercicio 11. Clasificador de edad

Crea un programa que solicite al usuario su edad y le diga en qué grupo etario se encuentra:

Info

Niñez (0-12)

Adolescencia (13-17)

Adulthood (18-64)

Vejez (65 o más)

```
Ejercicio 11: Clasificador de edad
Ingrese su edad: 45
Se encuentra en el grupo etario: Adultez
```

Requisitos:

- Utiliza switch con expresión y **patrones relacionales**.
- Evalúa mediante rangos.
- Muestra el resultado de la expresión.

Ejercicio 12. Calculo de notas

Crea un programa que pedirá dos notas exactas (sin decimales): la nota del examen y la nota de prácticas. El resultado será una nota numérica que puede tener decimales, además tanto las prácticas como los exámenes **solo se podrán evaluar con tres notas (4, 7, 10)**.

Con todo esto y las siguientes valoraciones, calcula la nota numérica final:

- Si la nota del examen es 4, la nota será la misma que la del examen independientemente de la de las prácticas.
- Si la nota del examen es 7 y la de prácticas es mayor o igual a 7 la nota será la media entre ambas
- Si la nota del examen es 7 y la de prácticas es 4 la nota final será 5
- Si la nota del examen es 10 y la de prácticas menor o igual a 7 la nota final será 9
- Si la nota del examen es 10 y la de prácticas es 10, la nota final será 11

```
Ejercicio 12: calculo de notas
Introduce la nota del examen (4, 7, 10): 7
Introduce la nota de prácticas (4, 7, 10): 4
La nota final es: 5
```

Requisitos:

- Se indicará nota incorrecta en caso de introducir una nota no permitida.
- Se usará el switch de expresión con `Tuplas`.
- Se mostrará la salida del switch de expresión.

Ejercicio 13. Clasificador de triángulos

Solicita al usuario tres valores enteros correspondientes a los lados de un triángulo. Indica qué tipo de triángulo es:

Info

Equilátero: todos los lados iguales

Isósceles: dos lados iguales

Escaleno: todos los lados distintos

```
Ejercicio 13: Clasificación de triángulos
Ingrese la longitud del primer lado: 4
Ingrese la longitud del segundo lado: 7
Ingrese la longitud del tercer lado: 4
El triángulo es isósceles.
```

Requisitos:

- Usa switch con patrones posicionales sobre una tupla de tres elementos.
- Valida que se pueda formar un triángulo con patrones, de forma que para que los lados sean correctos deberán de ser mayores a 0.