Índice

▼ Índice

- Ejercicio 1. Analizador léxico simple
- Ejercicio 1. Longitud, mayúsculas y concatenación
- Ejercicio 2. Reemplazos y recortes en cadena
- Ejercicio 3. Padding y conteo de palabras
- Ejercicio 4. Codificador con `StringBuilder`
- Ejercicio 6. Nivel de seguridad de contraseña
- Ejercicio 7. Comparación y anagramas
- Ejercicio 8. Extraer texto entre delimitadores
- Ejercicio 9. Generador de contraseñas aleatorias

Ejercicios Avanzados Unidad 11

Descargar estos ejercicios

> **Antes de empezar**
>
> Estos ejercicios están pensados para trabajar de forma avanzada el manejo de cadenas, expresiones regulares y objetos mutables como `StringBuilder` en C#. Se recomienda revisar los apuntes de la unidad antes de resolverlos.

Ejercicio 1. Analizador léxico simple

Crea un programa que lea un texto y separe en tokens las palabras, números y signos de puntuación usando expresiones regulares. Muestra cuántos elementos de cada tipo hay.

Ejercicio 1. Longitud, mayúsculas y concatenación

Crea un menú con las siguientes opciones:

1. Pedir una cadena y mostrar su longitud.

2. Pedir una cadena y mostrarla en mayúsculas.
3. Concatenar una frase a otra usando `.Concat`.

Crea métodos separados para cada funcionalidad.

Ejercicio 2. Reemplazos y recortes en cadena

Crea un menú con las siguientes opciones:

1. Pedir una cadena y dos números `P` y `N`. Eliminar `N` caracteres desde la posición `P`.
2. Pedir una cadena y recortar espacios o saltos de línea al final con `.TrimEnd`.
3. Pedir una cadena y sustituir una subcadena por otra usando `.Replace`.

Ejercicio 3. Padding y conteo de palabras

1. Pide una cadena y un número. Añade al final tantos caracteres `!` como indique el número usando `.PadRight`.
2. Pide una cadena y una palabra. Indica cuántas veces aparece la palabra usando `.IndexOf`.

Ejercicio 4. Codificador con `StringBuilder`

Crea un programa que reciba una frase acabada en punto y permita:

1. Invertir cada palabra (estilo espejo).
2. Reemplazar signos de puntuación por caracteres aleatorios (ASCII 224–238).
3. Eliminar todos los espacios.

Cada tipo de codificación debe implementarse en un método que utilice `StringBuilder`.

Ejercicio 6. Nivel de seguridad de contraseña

Escribe una función que devuelva el nivel de una contraseña:

- Muy débil: solo números o < 8 caracteres.

- Débil: solo letras y ≥ 8.
- Fuerte: letras y números y ≥ 8.
- Muy fuerte: letras/números + especiales y ≥ 8.

Ejercicio 7. Comparación y anagramas

Crea un método que determine si dos cadenas son anagramas. Ignora tildes y mayúsculas. Usa métodos como `.ToLower()`, `.Replace()`, y si es posible, `StringBuilder`.

Ejercicio 8. Extraer texto entre delimitadores

Implementa `ExtraerEntre(string texto, string ini, string fin)` que devuelve el contenido entre los delimitadores. Úsalo con cadenas estilo HTML o similares.

Ejercicio 9. Generador de contraseñas aleatorias

Solicita al usuario:

- nº de dígitos, letras y caracteres especiales.
- nº de contraseñas a generar.

Usa `StringBuilder`, `Random` y mezcla aleatoria para construir contraseñas.