	Título: Test C# v4 + WebAPI + UI	
	Versión: 4.0	Fecha modificación: 01/12/2023

Goal of the test

Evaluate the coding style, the quality of the code, the approach for the solution and the speed of coding.

What we want to evaluate and find in this code test is the following (all considered as *must have*):

- Apply SOLID principles application
- Apply one or more design patterns
- Use of dependency injection
- Write a Readable code
- Write testable or tested code
- Apply MVVM, you can choose any framework or library.
- Forbidden Business logic on UI Codebehind
- Front-End part its mandatory, so, you can start before part two mocking part one.

Description

Generate and deliver us one or more Visual Studio Solutions to ***Text Process***.

The final deliverable should be an UI that user can input order option and text to order plus a button to process the ordering and can also input text do analyze and a button to perform the analysis. Each button should show the result of text process.

The input text will be any text (phrase, a book paragraph, ...) with words separated by a space.

Must be delivered

1. WebAPI Rest

Exposing a Controller with 3 methods:


- Get ORDER OPTIONS
 - o With NO input parameters
 - o Returning 3 options: AlphabeticAsc, AlphabeticDesc, LenghtAsc)
- Get ORDERED TEXT
 - o With 2 input parameters (TextToOrder and OrderOption)
 - o Returning a list of words ordered
- Get STATISTICS
 - o With 1 input parameter (TextToAnalyze)
 - o Returning a json with text statistics defined bellow (TextStatistics)

Method 1

- Deliver the list of Order Options

Method 2

- Receives the text + one sort option
- Should recover all words of the string and sort them based on the received sort option, there are 3 types of sorting (AlphabeticAsc, AlphabeticDesc, LenghtAsc)

	Título: Test C# v4 + WebAPI + UI	
	Versión: 4.0	Fecha modificación: 01/12/2023

Tecnologías de la Información

- Should deliver a list of ordered words as a result

Method 3

- Receives text only
- Should calculate 3 statistics of the text: hyphens quantity, word quantity, spaces quantity
- Should deliver a complex POCO object called TextStatistics, with the calculated data

2. WPF UI

A **WPF MVVM** Window with the fields/components necessary to fulfill the indicated requirements.

This Window should call the 3 WebAPI methods to perform the operations and show the results

Remember, the test MUST have

- readable and maintainable code
- design patterns
- SOLID principles
- use a framework for Dependency Injection
- testable and tested code
- Apply MVVM
- Forbidden Business logic on UI Codebehind