

**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**FACULTAD DE INGENIERIA EN ELECTICIDAD Y COMPUTACION**

**PROYECTO ANAGRAMA**

**GRUPO**

**ROMERO RODRIGUEZ LUIS FERNANDO**

**ZURITA GUERRERO ANGELO SAUL**

**MATERIA:**

**ORGANIZACIÓN DE COMPUTADORES**

**TUTOR:**

**MSIG. VELOZ DE LA TORRE FREDY RONALD**

# **Marco Teórico**

Según el equipo editorial de Etecé (2016), un anagrama implica la generación de dos o más palabras diferentes por el solo hecho de cambiar de posición sus letras. Las nuevas palabras o frases que se forman pueden no tener ninguna relación lógica o gramatical con las originales.

Existen dos tipos de anagramas:

* **Anagramas de palabras**. Son los anagramas más comunes y se forman cuando se utilizan todas las letras de una palabra pero modificando el orden. Por ejemplo: *delira – lidera.*
* **Anagramas de frases**. Son un poco más difíciles de encontrar. En estos anagramas se parte de una o más palabras para luego extraer más términos (o una frase) alterando el orden de las letras. Por ejemplo:*anagrama – amar gana.*

# **Planteamiento del Problema**

El proyecto requiere la elaboración de un programa en Assembly que permita al usuario el ingreso de dos cadenas de caracteres, independientes entre sí, para luego determinar si las palabras o frases ingresadas son anagramas.

# **Requerimientos Funcionales**

* + 1. El usuario podrá ingresar dos cadenas de caracteres de hasta 40 caracteres (palabra o frase), y la utilización de la tecla ENTER indicará la finalización de cada cadena. NO CASE SENSITIVE. b)
    2. Posteriormente, el programa analizará el contenido de las dos cadenas y mostrará un mensaje indicando si son anagramas o no. c)
    3. Después, el usuario tendrá la opción de realizar más ingresos, hasta que decida lo contrario.

# **Funciones del programa**

|  |  |
| --- | --- |
| **Etiquetas - Funciones** | |
| main proc | * Es donde empieza todo. |
| start | * Es el punto donde el programa puede volver a empezar. * Se usa cuando el usuario quiere probar nuevas palabras. |
| continue\_prompt | * Pregunta al usuario si quiere continuar (S/N) * Y vuelve al start. |
| not\_anagram\_output | * Muestra el mensaje de que las palabras NO son anagramas. * Se usa cuando ya se sabe que las palabras no pueden ser anagramas. |
| read\_input proc | * Lee lo que escribe el usuario. * Se asegura de no leer más de 40 letras |
| normalize\_string proc | * Limpia el texto que escribió el usuario * Quita espacios * Convierte todo a MAYÚSCULAS |
| sort\_string proc | * Ordena las letras alfabéticamente. |
| compare\_strings proc | * Compara las dos palabras ya ordenadas * Decide si son anagramas o no |
| clear\_buffers proc | * Limpia todos los espacios de memoria * Como "borrar el pizarrón" para empezar de nuevo |

Tabla1. Tabla de funciones y etiquetas del programa ANAGRAMA

# **Demo – Programa**

|  |  |
| --- | --- |
| **Capturas de Pantalla** | **Descripción** |
|  | Compilación e inicio del programa. |
|  | Prueba con las mismas palabras para verificar que el programa devuelva correctamente que son anagramas. |
|  | Prueba con las mismas frases para verificar que el programa devuelva correctamente que son anagramas. |
|  | Prueba con palabras diferentes de distinto tamaño, para verificar que el programa devuelva que NO son anagramas. |
|  | Prueba con palabras no anagramas de igual tamaño para verificar que el programa devuelva que NO son anagramas. |
|  | Prueba con palabras alfanuméricas. |
|  | Prueba de los inputs para continuar (S o s) y terminar (N o n). |
|  | Prueba con palabra y frase, que si son anagramas para verificar que el programa devuelva que son anagramas. |
|  | Prueba de nuestro programa hasta que yo decida (en el sistema se representa con input N o n). |

# **Código en Assembly**

**Repositorio GitHub :**

<https://github.com/aszurita/Anagrama.git>

# **Referencias y Bibliografía**

*Etecé, equipo editorial. (2016, julio 22). 50 ejemplos de anagramas. Ejemplos.co.* [*https://www.ejemplos.co/50-ejemplos-de-anagramas/#ixzz8tCsIN521*](https://www.ejemplos.co/50-ejemplos-de-anagramas/#ixzz8tCsIN521)