Logotipo, nombre de la empresa

Descripción generada automáticamente

Manual del programador para el sistema Compilador con .Net

UNIREMINGTON

**Edwin Rivera Ballesteros**

Corporación Universitaria Remington

Programa de Ingeniería de Sistemas

Manizales, Colombia

2024

**Manual del programador para el sistema Compilador con .Net**

**Versión 1.0**

**Edwin Rivera Ballesteros**

**Asesor:**

Francisco Cardona

Corporación Universitaria Remington

Programa de Ingeniería de Sistemas

Manizales, Colombia

2024

INDICE

[INTRODUCCION 3](#_Toc167795980)

[ARQUITECTURA DEL SISTEMA 4](#_Toc167795981)

[Diagrama de arquitectura 4](#_Toc167795982)

[Tecnologías utilizadas 5](#_Toc167795983)

[REQUISITOS DEL SISTEMA 6](#_Toc167795984)

[INSTALACION 6](#_Toc167795985)

[ESTRUCTURA DEL CODIGO 6](#_Toc167795986)

[Organización del proyecto 6](#_Toc167795987)

[Nombramiento de variables 7](#_Toc167795988)

[Indentación y espacios 7](#_Toc167795989)

# INTRODUCCION

**Este manual está diseñado para proporcionar a los administradores del sistema una comprensión completa y detallada sobre la instalación, configuración y mantenimiento de un compilador o analizador léxico y semántico desarrollado con .net framework, utilizando la librería Irony. Y tiene como analizar y compilar código fuente escrito en un lenguaje de programación específico, proporcionando información detallada sobre la estructura léxica y semántica de este. A través de este proceso, se garantiza que el código cumple con las reglas sintácticas y semánticas definidas para el lenguaje. Pero mas importante que eso es entender el proceso desde que se inserta una cadena de texto hasta la creación de un programa objeto.**

# **ARQUITECTURA DEL SISTEMA**

## Diagrama de arquitectura

El siguiente diagrama ha sido creado para respaldar la arquitectura de componentes de la solución, diseñada para facilitar el uso del producto por parte de los usuarios.

El Compilador tiene la capacidad de permitirle al usuario la carga de archivos de texto y/o escribir texto en un campo el cual será tomado y analizados léxica y semánticamente y le retornaran al usuario el resultado del proceso de análisis.



## Tecnologías utilizadas

Para el proyecto utilizaremos:

* .net framework 4.8 o superior.
* Irony 1.5.1.

Para más información de las librerías y dependencias usadas en los proyectos se puede consultar el archivo **Readmi.MD** del proyecto**.**

# REQUISITOS DEL SISTEMA

* .net framework 4.8 o superior.
* Windows 10 o superior.

# INSTALACION

Para instalar el compilador primero validar que se cumple con el apartado anterior de requisitos del sistema.

# ESTRUCTURA DEL CODIGO

## Organización del proyecto

Texto

Descripción generada automáticamente

* **/Properties:** Contiene todas las propiedades y configuraciones del proyecto .net framework.
* **/Referencias:** En este apartado se pueden apreciar todas las referencias a las librerías que está usando el proyecto.
* **/Capas/Grammar.cs:** Este archivo contiene todas las reglas gramáticas, léxicas y sintácticas del compilador.
* **/Helper/Grammar.html:** Archivo html el cual contiene todas las referencias de ayuda al usuario.
* **/Helper/ManualForm.cs:** Formulario que trae el contenido del html de grammar.html y se muestra al usuario final.
* **App.Config:** Este archivo contiene todas las configuraciones del proyecto.
* **Package.config:** Este archivo hace relación a todos los paquetes que se usan en el proyecto.
* **Program.cs:** Es el archivo de inicio de la aplicación donde se inicial la misma.
* **VistaCompilador.cs:** Es el formulario principal del compilador donde el usuario puede realizar toda la interacción con el proyecto.
* **.gitignore:** Directorio en el cual se indica los archivos que se ignorar al subir los cambios al gestor de versiones GitHub.
* **READMI.md :** Archivo donde se almacena información pertinente al proyecto.

### Nombramiento de variables

* Se usa la notación de camelCase para nombrar variables y métodos. Por ejemplo, miVariable o miMetodo().

### Indentación y espacios

* Se aplica una sangría de 2 espacios.
* Se utiliza un espacio después de los dos puntos en las propiedades de los objetos.