

MANUAL TÉCNICO – IDE LFYDP

Descripción: El proyecto “IDE – LFYDP” es un IDE que se encuentra en fase de desarrollo, sin embargo, el IDE actualmente es capaz de crear, abrir y editar archivos y proyectos, también es capaz de reconocer expresiones determinadas y colorearlas.

INTERFAZ GRÁFICA (Clases):

AperturaArchivo: Clase encargada de mostrar un Windows Form, el cual recibirá una ruta de archivo, y creará un proyecto, en base a su ubicación (la cual se puede ingresar de manera manual, a través de un TextBox o se puede seleccionar a través de un FolderBrowserDialog que se mostrará presionando un botón).

CreacionDeProyecto: Clase encargada de mostrar un Windows Form, el cual recibirá el nombre del proyecto, la ubicación que tendrá el proyecto (la cual se puede ingresar de manera manual, a través de un TextBox o se puede seleccionar a través de un FolderBrowserDialog que se mostrará presionando un botón), y también le permite al usuario decidir si junto a su proyecto creará un archivo inicial (.gt) a través de un CheckBox.

NuevoArchivo: Clase encargada de mostrar un Windows Form, el cual creará un archivo (.gt) dentro de un proyecto ya abierto. El usuario únicamente debe ingresar el nombre del archivo nuevo en un TextBox.

IDE: Clase encargada de mostrar un Windows Form, el cuál muestra el IDE, compuesto por:

- **Área de Texto:** RichTextBox en el que el usuario escribirá.
- **Barra de Herramientas:** Menú desplegable, que brinda opciones al usuario.
- **Barra de Acceso Rápido:** Panel que contiene un grupo de objetos tipo Button, los cuales servirán para la facilidad del usuario, también contiene un Label, el cual indica la posición que el cursor tiene dentro del área de texto.
- **Área de Archivos de Código Fuente:** ListBox que muestra el nombre de todos los archivos que el proyecto contiene, si el usuario presiona el nombre de algún archivo, el programa debe mostrar en el área de texto el código del archivo seleccionado.
- **Área Log:** TextBox multilínea, el cuál se encuentra vacío al inicio de la ejecución del programa, este TextBox debe ser llenado cuando el usuario decida compilar su código.

PantallaInicial: Clase encargada de mostrar un Windows Form, el cual le permitirá al usuario decidir si crear un proyecto, abrir un proyecto, crear un archivo o abrir un archivo; esto a través de botones.

ARCHIVOS (Clases):

Automata: Clase encargada de manejar todo lo relacionado con la validación o descarte de tokens recibidos. El autómata ha sido programado por medio de métodos, por lo cual, cada método representa cada estado del autómata y las transiciones se dan en las condiciones que cada método posee. Sus métodos a describir:

- **public bool Comprobar(String cadenaIngresada):** Método encargado de comprobar si el resultado final de las transiciones entre estados es verdadero. Cadena ingresada se refiere al token recibido.
- **private void Inicializacion():** Inicializa las palabras reservadas que el lenguaje tiene.
- **Public String Color:** Getter que devuelve el color del cuál se deberá pintar el token ingresado.

AnalizadorDeToken: Clase encargada de manejar el token, previo a mandarlo al autómata. Sus métodos a describir:

- **public AnalizadorDeToken(RichTextBox txtArea):** Constructor de la clase. El RichTextBox crea una instancia en la clase del área de texto del IDE.
- **Private void Inicializacion():** Inicializa aquellos tokens que son el final de una cadena de dos caracteres.
- **Public void AnalizarToken():** Este método analiza los diferentes tipos de tokens que el usuario puede ingresar, desde tokens de un carácter, hasta tokens representados por una cadena de caracteres.
- **Private void ActualizarDatos():** Actualiza el número de fila y columna, en el que se encuentra el cursor que está dentro del área de texto.
- **Private void Pintar(int strt, int length):** Pinta la cadena de caracteres según lo establecido por el autómata. Strt marca el inicio de la cadena y length marca la longitud de la cadena.

AnalizadorLog: Clase encargada de Analizar los errores que se mostrarán en el área log. Sus métodos a describir:

- **Public void Analizar(RichTextBox txtArea, TextBox txtLog, int index):** Analiza los errores que se encuentran en el área de texto y los escribe en el área log. txtArea crea en la clase una instancia del área de texto del IDE, txtLog crea en la clase una instancia del área Log del IDE, index es la ubicación en la que se encuentra el cursor dentro del área de texto.