

USAC/CUNOC

Ingeniería en Ciencias y Sistemas

Laboratorio de Lenguajes Formales y de Programación

Ronald Danilo Chávez Calderón

200130586

# Manual Técnico

## CLASE IDE

### Atributos:

private tokensPink1: string[]

Tokens color rosado con 1 caracter de tamaño

private tokensBlue1: string[]

Tokens color azul con 1 caracter de tamaño

private tokensBlue2: string[]

Tokens color azul con 2 caracteres de tamaño

private tokensGreen2: string[]

Tokens color verde con 2 caracteres de tamaño

private tokensGreen4: string[]

Tokens color verde con 4 caracteres de tamaño

private tokensGreen5: string[]

Tokens color verde con 5 caracteres de tamaño

private tokensGreen6: string[]

Tokens color verde con 6 caracteres de tamaño

private tokensPurple6: string[]

Tokens color púrpura con 6 caracteres de tamaño

private tokensGray6: string[]

Tokens color gris con 6 caracteres de tamaño

private tokensGreen7: string[]

Tokens color verde con 7 caracteres de tamaño

private tokensCyan7: string[]

Tokens color cyan con 7 caracteres de tamaño

private tokensGreen8: string[]

Tokens color verde con 8 caracteres de tamaño

private tokensOrange8: string[]

Tokens color naranja con 8 caracteres de tamaño

private tokensBrown8: string[]

Tokens color cafe con 8 caracteres de tamaño

private tokensGreen10: string[]

Tokens color verde con 10 caracteres de tamaño

private tokensGreen12: string[]

Tokens color verde con 12 caracteres de tamaño

private tokensGreen14: string[]

Tokens color verde con 14 caracteres de tamaño

private tokensGreen15: string[]

Tokens color verde con 15 caracteres de tamaño

Métodos:

public IDE()

Método para crear el objeto IDE.

public colorText(): int

Método para dar color al texto del RichTextBox.

public compareToDefinedTokens(): string

Método para comparar un string de entrada con los tokens definidos previamente, devuelve el token que encuentre.

public compile(): string

Método para compilar el texto completo y devolver una lista de tokens hallados.

public processText(): void

Método para procesar cada línea de texto donde nos hallemos y colorearla adecuadamente.

public openFileGT(): void

Método para abrir los archivos de código fuente.

public openFileLogGTE(): void

Método para abrir los archivos de log en el IDE.

public openFileProjectGTP(): void

Método para abrir los archivos de proyecto.

public saveFileGT(): void

Método para guardar los archivos de código fuente

public saveFileLogGTE(): void

Método para guardar los archivos de log en el IDE.

public saveFileProjectGTP(): void

Método para guardar los archivos de proyecto.

public isString(): bool

Método para averiguar si el texto ingresado es un string.

public isCharacter(): bool

Método para averiguar si el texto ingresado es un booleano.

public isDecimal(): bool

Método para averiguar si el texto ingresado es un decimal.

public isInteger(): bool

Método para averiguar si el texto ingresado es un entero.

## CLASE AUTOMATON

Atributos:

private states: string[]

Estados definidos del autómata.

private alphabet: string[]

Alfabeto de letras que se ingresan al autómata

private transitionFunction: string[,]

Función de transición que rige los cambios de estados del autómata.

private initialState: string

Estado inicial del autómata.

private actualState: string

Estado actual del autómata.

private actualLetter: string

Letra del alfabeto actual que se ingresa al autómata.

Métodos:

public Automaton()

Método para crear el objeto Automaton.

public nextState(): string

Método para pasar al siguiente estado, se ingresa un estado y se devuelve el siguiente.

public AFD(): string

Método para devolver el siguiente estado de un AFD.