



Especificación

DISEÑO E IMPLEMENTACIÓN DE UN LENGUAJE

v1.0.0

1. Equipo	1
2. Repositorio	1
3. Dominio	2
4. Construcciones	2
5. Casos de Prueba	3
6. Ejemplos	3

1. Equipo

Nombre	Apellido	Legajo	E-mail
Mateo	Reino	61.307	mreino@itba.edu.ar
Tomás	Scheffer	63.393	tscheffer@itba.edu.ar

2. Repositorio

La solución y su documentación serán versionadas en: [CONCAT](#)

3. Dominio

Desarrollar un lenguaje que permita crear, modificar y operar sobre cadenas de texto. El lenguaje debe poder realizar operaciones sobre cadenas estáticas de texto estáticas. La idea es facilitar la construcción e implementación de aplicaciones que hacen uso extensivo de cadenas de texto evitando el uso de funciones que necesitan ejecutarse en tiempo de ejecución y que ofuscan la interpretación del programa, también así, haciéndolos más seguros.

La correcta implementación de este lenguaje permitirá realizar aplicaciones más eficientes, seguras y fáciles de interpretar, minimizando los riesgos y costos finales.

4. Construcciones

El lenguaje desarrollado debería ofrecer las siguientes construcciones, prestaciones y funcionalidades:

- (I). Se podrá crear un nuevo string como concatenación de dos.
- (II). Se podrá encriptar un string que contenga información valiosa ya sean contraseñas o información personal.
- (III). Se podrá obtener el inverso de un string.
- (IV). Alternancia entre dos strings char a char .
- (V). Conversiones de int a String[] o de String[] a int.
- (VI). Se podrá pasar un string a mayúsculas.
- (VII). Se podrá obtener el número de apariciones de una cadena dentro de otra.
- (VIII). Se proveerán operaciones aritméticas básicas como +, -, * y /.
- (IX). Se proveerá una calculadora .

5. Casos de Prueba

Se proponen los siguientes casos iniciales de prueba de **aceptación**:

- (I). Concatenar dos cadenas.
- (II). Encriptar un string.
- (III). Obtener el reverso de un string.
- (IV). Alternar caracteres entre dos strings.
- (V). Conversión de string a int y viceversa.
- (VI). Operaciones aritméticas básicas.
- (VII). Número de subcadenas en una cadena.
- (VIII). Uso de la calculadora.

Además, los siguientes casos de prueba de **rechazo**:

- (I). Errores de sintaxis.
- (II). Errores de tipo, por ejemplo string en vez de String.
- (III). Operación aritmética entre un string y un int .
- (IV). Aplicar una función de encriptación a un entero.

6. Ejemplos

A continuación se detallan distintos ejemplos de uso de nuestro lenguaje:

```
// Calcular sobre un string una operación de suma:
String miFormula1 = "${5+5}";

// Posible equivalente al anterior
String miFormula2 = CAL("5+5");

// Encriptar un string con AES 128
String crypto = AES("clave", "datos");

// Generar el reverso de un string
String reverso = REV("hola mundo");

// Encriptar el reverso de un string
String cryptoRev = AES("clave", REV("hola mundo"));

// Crear un String aleatorio de caracteres alfanumericos con largo máximo de 10
String random = RND(ALP, MAX(10));

// Crear un String aleatorio solo con caracteres 'a', 'b' y 'c' de largo 5
String random = RND("abc", 5);
```

Más Ejemplos:

```
// Retorna un string en mayúsculas
String upper = TUP("string");

// Cantidad de apariciones del string 2 en el string1
int appearance = APC(string1, string2);

// Operacion aritmetica 5/2. Puede recibir + , * , - , / , %
Double answer = NUM(CAL("5/2"));
```