



# Estructuras de Datos y Algoritmos (EDA)

## Tarea 3: Calculadora Interactiva

Prof: José M. Saavedra Rondo

Ayudante: Lukas Pavez

Octubre 2023

### 1. Objetivo

Comprender y aplicar eficientemente estructuras dinámicas en la resolución de problemas.

### 2. Descripción

En esta tarea deberás implementar una calculadora interactiva que permita:

- Asignar un valor a una variable (ej.  $x = 5$ ).
- Calcular una expresión que puede incluir cinco posibles operaciones:
  - Suma (+) : ejemplo  $5 + 3$
  - Resta (-): ejemplo  $5 - 3$
  - Multiplicación (\*): ejemplo  $5 * 3$
  - División (/): ejemplo  $5 / 3$
  - Potenciación (^) : ejemplo  $5 ^ 3$
- Además, se pueden usar paréntesis como signo de agrupación.
- El resultado de la última operación deberá quedar guardada en una variables global llamada *ans*, esta variable puede ser reusada para otros cálculos.
- Deberás construir un árbol con la última expresión evaluada

### 3. Ejemplo de ejecución

```
$ ./calculadora
"""""""" CALCULADORA INTERACTIVA """"""""
$ x = 6
$ x + 1
$ x + x * 2 + ans
$ ans = 25
```

```

$ tree
$ --- +
----- +
----- x
----- *
----- x
----- 2
----- ans
$ FIN

```

## 4. Solución

Para resolver una expresión, deberás convertir la expresión de notación infija a postfija. Para esto puedes seguir el algoritmo descrito en <https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression>. Luego resuelve la notación postfija, tal como en el control 2.

Por otro lado, las variables tienen que ser guardadas en memoria, puedes usar un diccionario como `std::map`.

**Puedes suponer que la expresión se escribe separando operadores y operandos por espacio en blanco.**

## 5. Restricciones

1. Pueden trabajar en grupos de 2 estudiantes.
2. Todos los programas deben ser propios, permitiendo solamente utilizar el código disponible en el repositorio del curso [https://github.com/jmsaavedrar/eda\\_cpp](https://github.com/jmsaavedrar/eda_cpp). **Así, todo el código debe ser realizado sobre el código del repositorio del ramo.**
3. El hallazgo de plagio será penalizado con nota 1.0, para todos los grupos involucrados.
4. Todas las implementaciones deben ser realizadas en C++.
5. **La entrega del informe es obligatorio.** Un trabajo sin informe no será calificado, asignando la nota mínima igual a 1.0.

## 6. Entrega

La entrega se debe realizar por canvas hasta el domingo 27 de octubre, 2023, 23:50 hrs. La entrega debe incluir:

1. Código fuente (en C++), junto a un README con los pasos de compilación.
2. Informe