

# Manual de Usuario

## Árboles AVL en

## Compiladores

Hecho en C#

[Link de GitHub](#)

## Grupo 22

---

### ¿Qué hace nuestro código?

*Nuestro código es una muestra muy básica de como los compiladores implementan los árboles AVL para facilitar el almacenamiento de variables y sus datos.*

#### Árboles AVL

*Un árbol AVL (Adelson-Velsky y Landis) es una estructura de datos de árbol binario balanceado en la que la diferencia de alturas entre sus subárboles izquierdo y derecho (conocida como el factor de equilibrio) se mantiene en un rango específico, generalmente -1, 0 o 1, lo que garantiza un tiempo de búsqueda y inserción eficiente,  $O(\log n)$ , en operaciones de árbol.*


#### Impresiones

*Nuestro código imprime las variables detectadas en un programa main de cualquier proyecto de C#. Las muestra indentadas tal como las almacena el árbol AVL*

# Requisitos

## *.NET Core*

*Debido a que la mayoría del programa fue hecho en Linux, el IDE utilizado fue JetBrains Rider, donde el .NET que se instaló fue el 4.8.*

 *No fue probado para otras versiones posteriores del .NET*

## *IDE*

*Para el desarrollo del programa y para las pruebas del código se utilizaron los siguientes IDEs*

- 1. JetBrains Rider 2023.2.2*
- 2. Visual Studio 2019*

## *Computadora*

*Seguir las recomendaciones de los requisitos mínimos para ejecutar los IDEs*

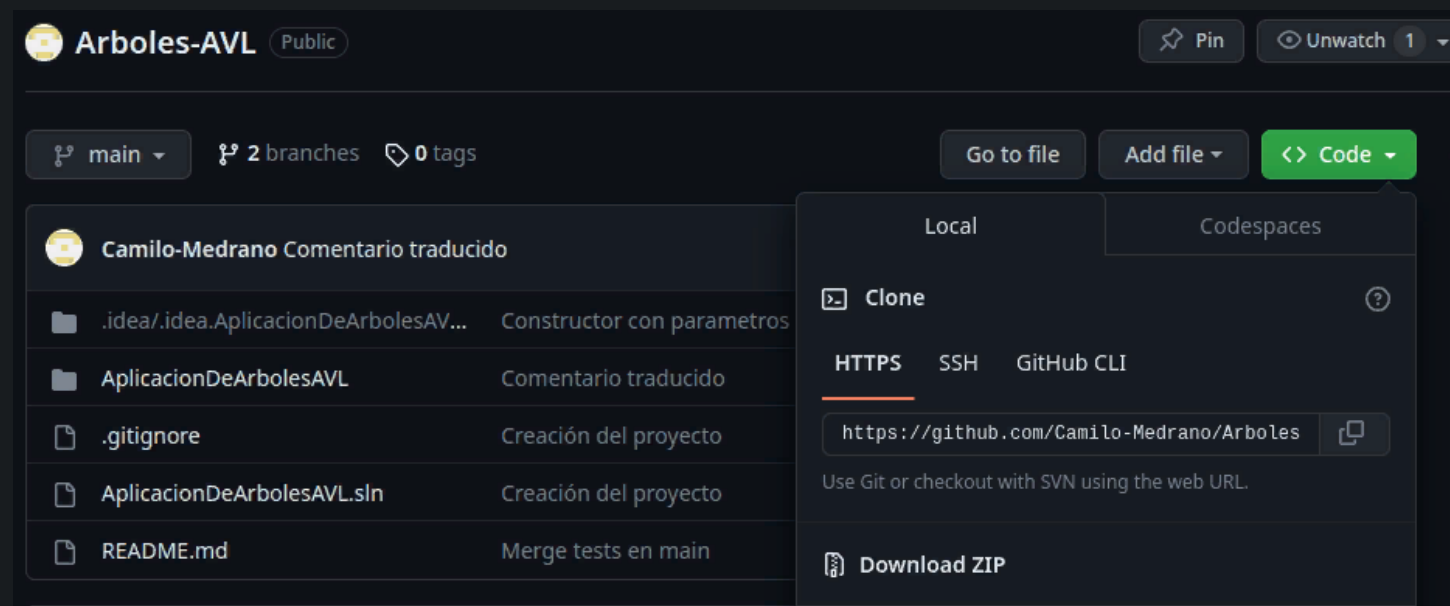
# Instalación

## *Descarga de GitHub*

*Navegar en la terminal hasta la carpeta deseada y ejecutar:*

```
git clone https://github.com/Camilo-Medrano/Arboles-AVL
```

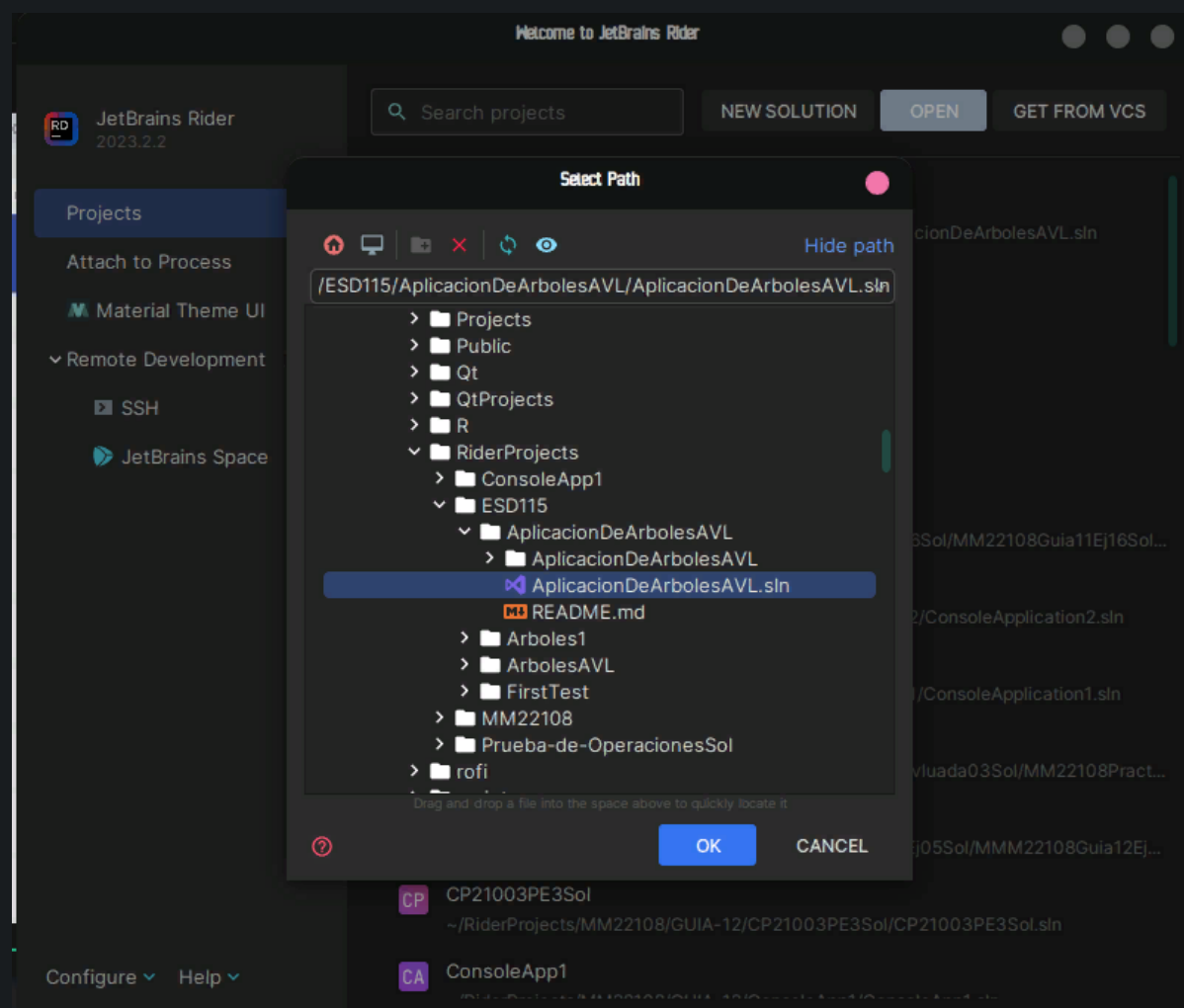
Otra forma es descargando el .zip desde GitHub, para eso abrir el link de GitHub, seleccionar Code y hacer click en Download Zip.



Luego descomprimir el archivo en la ubicación deseada

Abrir solución

Al tener la carpeta descargada, abrir el IDE y darle a la opción de abrir, seleccionar el archivo .sln



# Funciones y Características

## Objetivo

*Simular una parte de un compilador, de como este utiliza árboles AVL para guardar variables y buscarlas.*

## Capacidades

*Capaz de reconocer todo tipo de variables y sus valores dado un programa del main de C#*

## Limitantes


*No validaremos el código proporcionado por el usuario, asumiremos que está hecho sin errores debido a que sale de nuestros objetivos del programa validarlos.*

## Guía de Uso

*El uso de nuestro programa no requiere gran cosa, solamente lo siguiente:*

- 1. Tener abierta la solución del proyecto en el IDE a elección*
- 2. Un código del main de cualquier programa de C#*

*Si ya se tiene eso, se puede continuar con los siguientes pasos*

 Solo se necesita lo que esta adentro del main, no todo el archivo .cs

## Insertar código

*Copiar el código elegido en el paso 2 y pegarlo en la siguiente parte*

```
5  namespace AplicacionDeArbolesAVL
6  {
7      internal class Program
8      {
9          public static void Main(string[] args)
10         {
11             string codigoCSharp = @"
12                 int numero = 3;
13                 string saludo = ""Hola, querido usuario"";
14                 string nombre = ""Camilo Perez"";
15                 string adjetivo = ""honorable"";
16                 string siguiente = ""estimado usuario"";
17                 string descripcion = ""humilde persona"";
18                 string motivo = ""respuesta formal"";
19                 double valor2 = 4.5;
20
21                 // Ejemplo de operación aritmética válida
22                 double resultado = numero + valor2;
23
24                 // Actualización de la variable
25                 valor2 = 3;
26                 Console.WriteLine(valor2);
27
28                 // Concatenación de cadenas de texto
29                 string saludoFinal = saludo + "" Rodrigo. Le damos la bienvenida a nuestro sistema."";
30                 Console.WriteLine(saludoFinal);
31             ";
32         }
```

## Ejecutar el programa

*Presionar el botón de ejecutar en el IDE o F5, se mostrará la consola con la impresión del árbol de diferentes formas.*