



1-6-2021

Manual de usuario

Proyecto JPR

Eduardo Ixen
INGENIERIA USAC

Contenido

| | |
|---|----------|
| Requisitos para ejecutar el programa | 2 |
| Instalando librerías | 2 |
| Ejecutando el programa | 2 |
| Partes del editor | 3 |
| Menú archivo | 3 |
| • Crear archivos | 3 |
| • Abrir archivos: | 3 |
| • Guardar: | 3 |
| • Guardar Como | 3 |
| Menú Reportes | 3 |
| • Reporte de errores: | 3 |
| • Generar Árbol AST: | 3 |
| • Tabla de símbolos: | 3 |
| Ejecutando código | 3 |
| Cargando un archivo: | 3 |
| Escribiendo código: | 6 |
| Ver reportes | 6 |
| Ver reporte de errores | 6 |
| Generar Árbol AST | 7 |
| Ver tabla de símbolos | 8 |

Requisitos para ejecutar el programa

- Tener instalado Python 3.9
- Instalar la librería numpy de Python
- Instalar la librería tabulate de Python

Instalando librerías

➤ Numpy

Para instalar la librería numpy únicamente debemos abrir la consola de nuestra computadora y escribir el siguiente comando.

```
Pip install numpy
```

➤ Tabulate

Para instalar la librería tabulate se escribe el siguiente comando en símbolo del sistema.

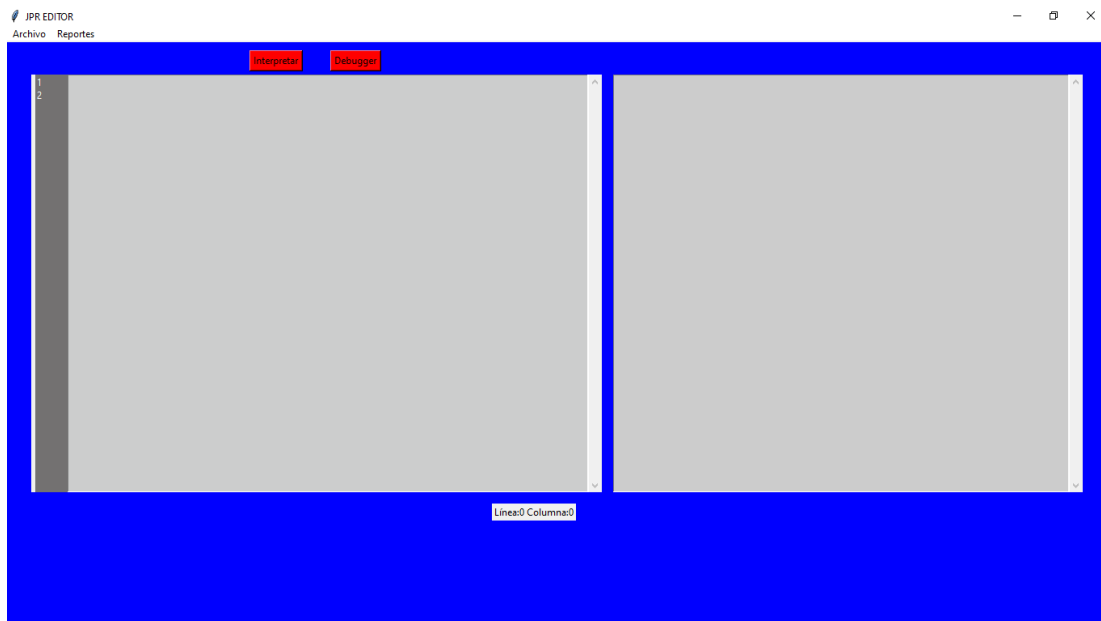
```
Pip install tabulate
```

Ejecutando el programa

Para ejecutar el programa luego de haber descargado el repositorio, desde la consola navegamos a la carpeta principal del proyecto y ejecutamos el siguiente comando.

Py main.py

Obteniendo como resultado lo siguiente:



Como podemos observar contamos con un editor de texto donde se podrá escribir código jpr y una consola donde se mostraran las salidas obtenidas del código ingresado.

Partes del editor

Menú archivo

- **Crear archivos:** El editor crear archivos en blanco y limpia el editor si contenía algo previamente.
- **Abrir archivos:** El editor abre archivos con extensión .jpr.
- **Guardar:** El editor guardar el estado del archivo en el que se estará trabajando si ya se ha guardado previamente, de lo contrario se guardara el archivo especificando el nombre y la ruta deseada.
- **Guardar Como:** El editor guardar el contenido del editor en el que se estará trabajando con un nuevo nombre.

Menú Reportes

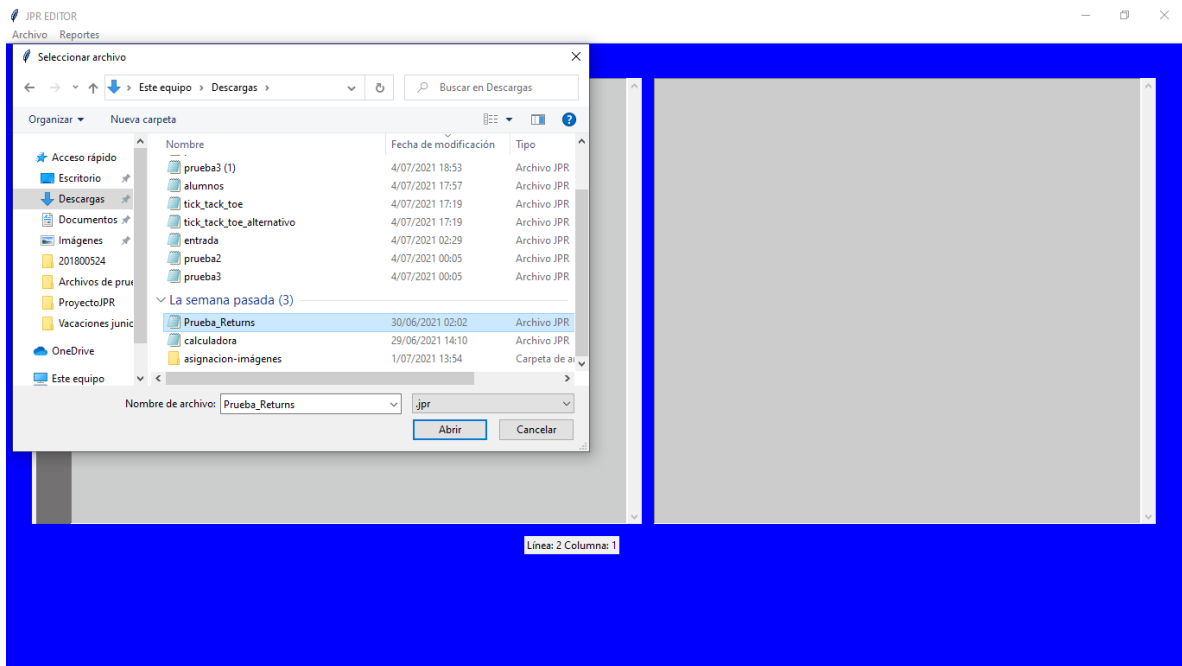
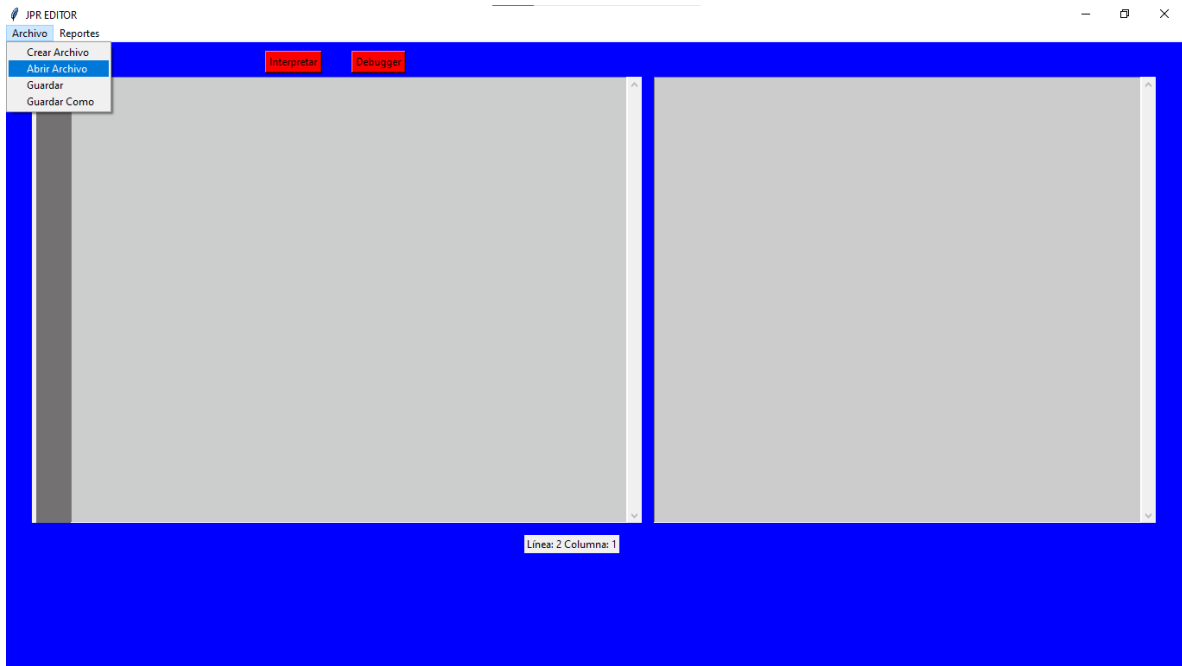
- **Reporte de errores:** El editor almacena todos los errores léxicos, sintácticos y semánticos que se producen al ejecutar el código ingresado y los agrega en una tabla de html para una mejor visualización.
- **Generar Árbol AST:** Se crea el árbol de análisis sintáctico del código ingresado en el editor, en un documento pdf.
- **Tabla de símbolos:** Se crear la tabla de símbolos del código ingresado en el editor, mostrándose en una tabla de html para una mejor visualización.

Ejecutando código

Es posible ejecutar código de dos formas:

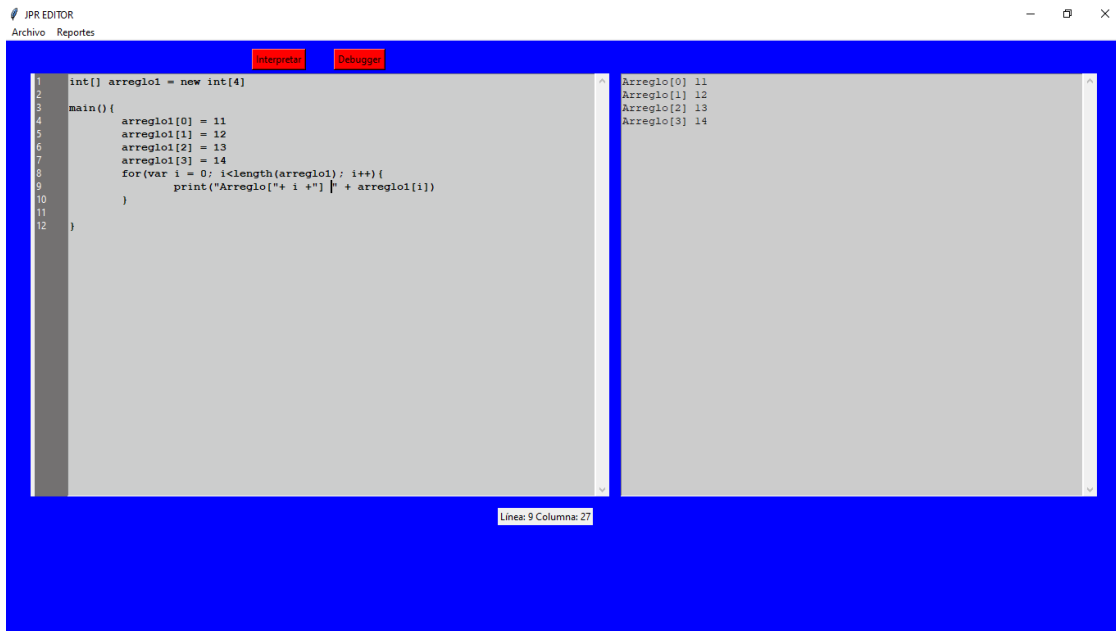
Cargando un archivo:

Para cargar un archivo jpr nos dirigiremos a la barra de menú en la opción “Archivo” y seleccionaremos la opción “Abrir archivo”, luego navegaremos hasta la ruta de nuestro archivo y seleccionaremos.



Escribiendo código:

También es posible escribir código para ejecutarlo, únicamente nos ubicamos en el editor de la parte izquierda y escribimos para luego ejecutar.



The screenshot shows the JPR Editor interface. On the left, the code editor contains a Java program that initializes an array and prints its elements. On the right, the output window displays the result of the program execution.

```
1 int[] arreglo1 = new int[4]
2
3
4 main() {
5     arreglo1[0] = 11
6     arreglo1[1] = 12
7     arreglo1[2] = 13
8     arreglo1[3] = 14
9     for(var i = 0; i<length(arreglo1); i++){
10        print("Arreglo["+ i +"] |" + arreglo1[i])
11    }
12 }
```

```
Arreglo[0] 11
Arreglo[1] 12
Arreglo[2] 13
Arreglo[3] 14
```

Linea: 9 Columna: 27

Ver reportes

Ver reporte de errores

Si al ejecutar el código ingresado, se encuentra con algún error léxico, sintáctico o semántico; se mostrará en la consola y se almacenara para crear un archivo de html para visualizar todos los errores como una tabla.



The screenshot shows the JPR Editor interface with the 'Reportes' (Reports) menu open. The code editor contains a Java program with several errors. The output window displays the error report, which includes a table of errors.

```
1
2
3
4
5 var numeroFactorial = n2;
6 while (numeroFactorial > -1) {
7     mostrarFactorial(numeroFactorial);
8     numeroFactorial--;
9 }
10 print("-----Para Calificar Ciclos-----");
11 SentenciasAnidadas ();
12 print("-----CICLO WHILE Y FOR-----");
13
14
15
16 main() {
17     Principal(7);
18     a var = 12
19 }
20
21 func Principal(int start){
22     print("*****ARCHIVO 2*****");
23     print("VALOR: 15 PTS");
24     FactorialIterativo(start);
25     RecursividadBasica();
26     multiPlicacionPorSumas(7,9);
27     print("*****");
28 }
29
30
31 func mostrarFactorial(int n2){
32     var fact = 1;
```

| Linea | Columna | Descripción |
|-------|---------|--|
| 16,1 | | Lexico - Error lexico.5 |
| 18,7 | | Sintactico - Error Sintactico.var |
| 17,5 | | Semantico - NO SE ENCONTRO LA FUNCION: Principal |
| 15 | | VALOR: 15 PTS |
| 24,24 | | Semantico - Variable start no encontrada |

Linea: 51 Columna: 15

| Reportes de errores | | | | |
|---------------------|------------|--------------------------------------|------|---------|
| No | Tipo | Descripcion | Fila | Columna |
| 1 | Lexico | Error lexico.\$ | 16 | 1 |
| 2 | Sintactico | Error Sintactico.var | 18 | 7 |
| 3 | Semantico | NO SE ENCONTRO LA FUNCION: Principal | 17 | 5 |
| 4 | Semantico | Variable start no encontrada. | 24 | 24 |

Generar Árbol AST

El programa crear una representación gráfica del AST del código ingresado, para visualizarlo nos dirigiremos al menú "Reportes" en la opción "Generar árbol AST" el cual nos mostrará un archivo pdf con el árbol graficado.

The screenshot shows the JPR Editor interface. The 'Reportes' menu is open, showing options: 'Reporte De Errores', 'Generar Árbol AST', and 'Tabla De Símbolos'. The 'Generar Árbol AST' option is selected. The main editor displays C code with line numbers 36 to 67. The code includes a function 'SentenciasAnidadas()' and a 'switch' statement. The output pane on the right shows the generated AST output, which includes a header '*****ARCHIVO 2*****', a value 'VALOR: 15 PTS', and a detailed calculation of factorials for numbers 1 through 7. The output also shows a 'SWITCH CASE' section and a 'WHILE ANIDADO' section. The status bar at the bottom indicates 'Lineas: 48 Columna: 6'.

Reporte

file:///ProyectoJPR/Reporte/ReporteSimbolos.html

Tabla de símbolos

| No | Identificador | Tipo | Tipo2 | Entorno | Valor | Fila | Columna |
|----|--------------------|-------------------|--------|--------------------|----------------------|------|---------|
| 1 | numeroFactorial | Variable | ENTERO | factorialiterativo | 7 | 5 | 5 |
| 2 | factorialiterativo | Funcion | ---- | Global | ---- | 1 | 1 |
| 3 | Main | Funcion Principal | ---- | Global | ---- | 15 | 1 |
| 4 | principal | Funcion | ---- | Global | ---- | 19 | 1 |
| 5 | fact | Variable | ENTERO | mostrarfactorial | 1 | 30 | 5 |
| 6 | cadena1 | Variable | CADENA | mostrarfactorial | El factorial de: 0 = | 31 | 5 |
| 7 | mostrarfactorial | Funcion | ---- | Global | ---- | 29 | 1 |
| 8 | numero1 | Variable | ENTERO | sentenciasanidadas | 0 | 49 | 5 |
| 9 | sentenciasanidadas | Funcion | ---- | Global | ---- | 48 | 1 |
| 10 | i | Variable | ENTERO | figura0 | 0 | 96 | 5 |
| 11 | j | Variable | ENTERO | figura0 -> While | 0 | 98 | 9 |
| 12 | numeroMostrar | Variable | ENTERO | figura0 -> While | 1 | 99 | 9 |
| 13 | unaFila | Variable | CADENA | figura0 -> While | | 100 | 9 |
| 14 | figura0 | Funcion | ---- | Global | ---- | 94 | 1 |
| 15 | cadenaFigura | Variable | CADENA | figura1 | | 115 | 5 |