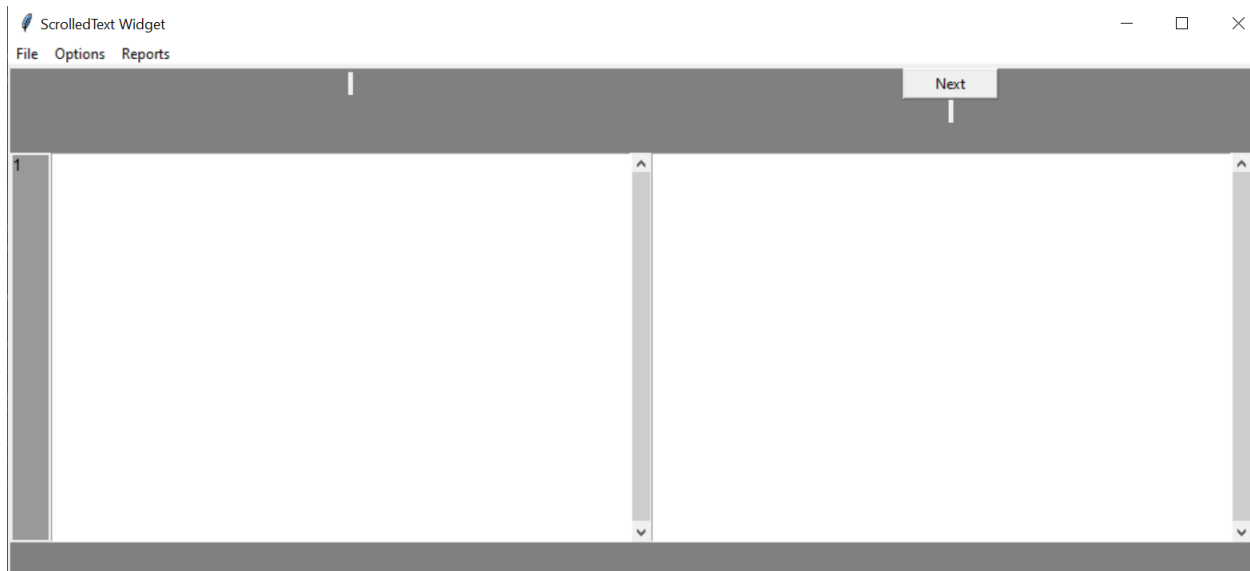


# Manual de Usuario

La aplicacion de escritorio JPR es un interprete que ejecuta instrucciones de alto nivel el cual cuenta con analisis lexico, sintactico y semantico de un compilador. Al ejecutar la aplicacion se mostrara una Ventana como aparece a continuacion la cual cuenta con algunas opciones, un area de texto para el editor y otro para la consola, un boton para el debugger, un label para mostrar la fila y columna en la que se encuentre el cursor en el editor y otro label para mostrar la fila en la que se encuentre ejecutando el editor.



Como se observa en la parte superior izquierda de la ventana cuenta con 3 opciones las cuales son File, Options y Reports. La primera opción(File) contiene las siguientes opciones.



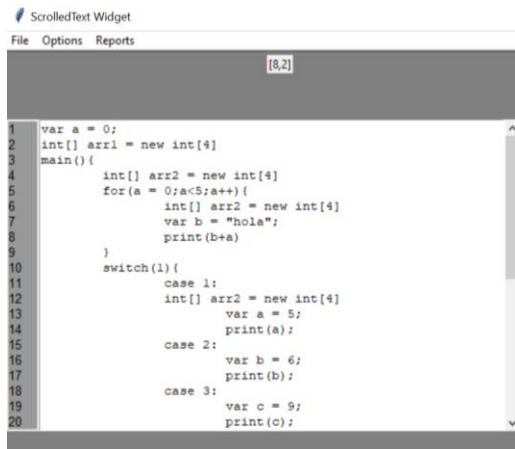
La opción de Open permitirá abrir cualquier archivo JPR , si por algún motivo o error se abre un archivo que no es JPR se mostrara un mensaje de error. En la opción de New permite crear un nuevo archivo JPR eliminando del editor el texto si es que existe alguno. Para la opción de Save permite guardar cambios que se hayan realizado en algún archivo recientemente abierto o guardar algún nuevo archivo, el cual pedirá el nombre y la ubicación donde se desea almacenar. Para el Save As guardara el archivo solicitando un nombre y la ubicación donde se almacenara.

La siguiente opción de la barra de menú es el de Options, el cual cuenta con 3 opciones la primera es la de Interpret la cual usara el interprete para leer código JPR que haya en el editor y los resultados los mostrar en la consola

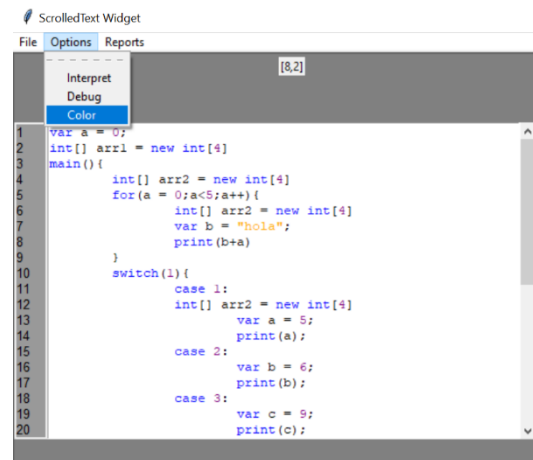


La otra opción es la de Debug la cual ejecutara paso a paso las instrucciones del editor, para ejecutarlos de primero se deberá de activar el debug el cual esta en la barra de menú en Options > Debug al hacer esto es debug esta activo entonces se procede a presionar el botón de Next que esta en la ventana y se iran ejecutando instrucción por instrucción cada vez que se presione, al terminar de ejecutar todas las instrucciones el debugger se desactivara y para volver a usarlo hay que activarlo nuevamente.

Al presionar la opción de Color pintara las palabras reservadas, num y comentarios. También los pinta automáticamente al abrir un archivo o al ejecutar el código.

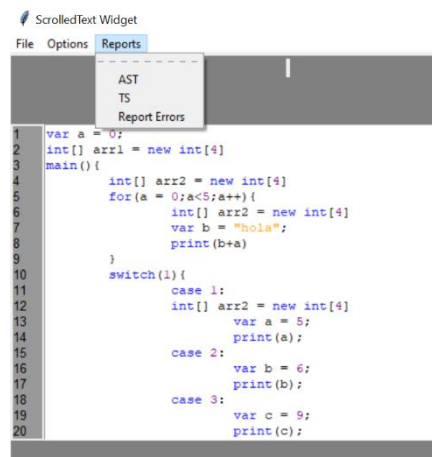


```
1 var a = 0;
2 int[] arr1 = new int[4]
3 main() {
4     int[] arr2 = new int[4]
5     for(a = 0; a < 5; a++) {
6         int[] arr2 = new int[4]
7         var b = "hola";
8         print(b+a)
9     }
10    switch(1) {
11        case 1:
12            int[] arr2 = new int[4]
13            var a = 5;
14            print(a);
15        case 2:
16            var b = 6;
17            print(b);
18        case 3:
19            var c = 9;
20            print(c);
```



```
1 var a = 0;
2 int[] arr1 = new int[4]
3 main() {
4     int[] arr2 = new int[4]
5     for(a = 0; a < 5; a++) {
6         int[] arr2 = new int[4]
7         var b = "hola";
8         print(b+a)
9     }
10    switch(1) {
11        case 1:
12            int[] arr2 = new int[4]
13            var a = 5;
14            print(a);
15        case 2:
16            var b = 6;
17            print(b);
18        case 3:
19            var c = 9;
20            print(c);
```

La ultima opción que es la de Reports en la cual se podrán crear 3 tipos de reportes. El primero es el árbol AST el cual creara un archivo pdf en donde se podrá visualizar el árbol de lo que se encuentre en el editor. Para ver el reporte de la tabla de símbolos esta la opción de TS, al presionarlo se abrirá automáticamente un archivo html en el navegador mostrando la tabla de símbolos con todos sus datos. Y por ultimo esta el reporte de errores que al igual que la TS abrirá automáticamente un html mostrando los error léxicos, sintácticos y semánticos.



```
1 var a = 0;
2 int[] arr1 = new int[4]
3 main() {
4     int[] arr2 = new int[4]
5     for(a = 0; a < 5; a++) {
6         int[] arr2 = new int[4]
7         var b = "hola";
8         print(b+a)
9     }
10    switch(1) {
11        case 1:
12            int[] arr2 = new int[4]
13            var a = 5;
14            print(a);
15        case 2:
16            var b = 6;
17            print(b);
18        case 3:
19            var c = 9;
20            print(c);
```