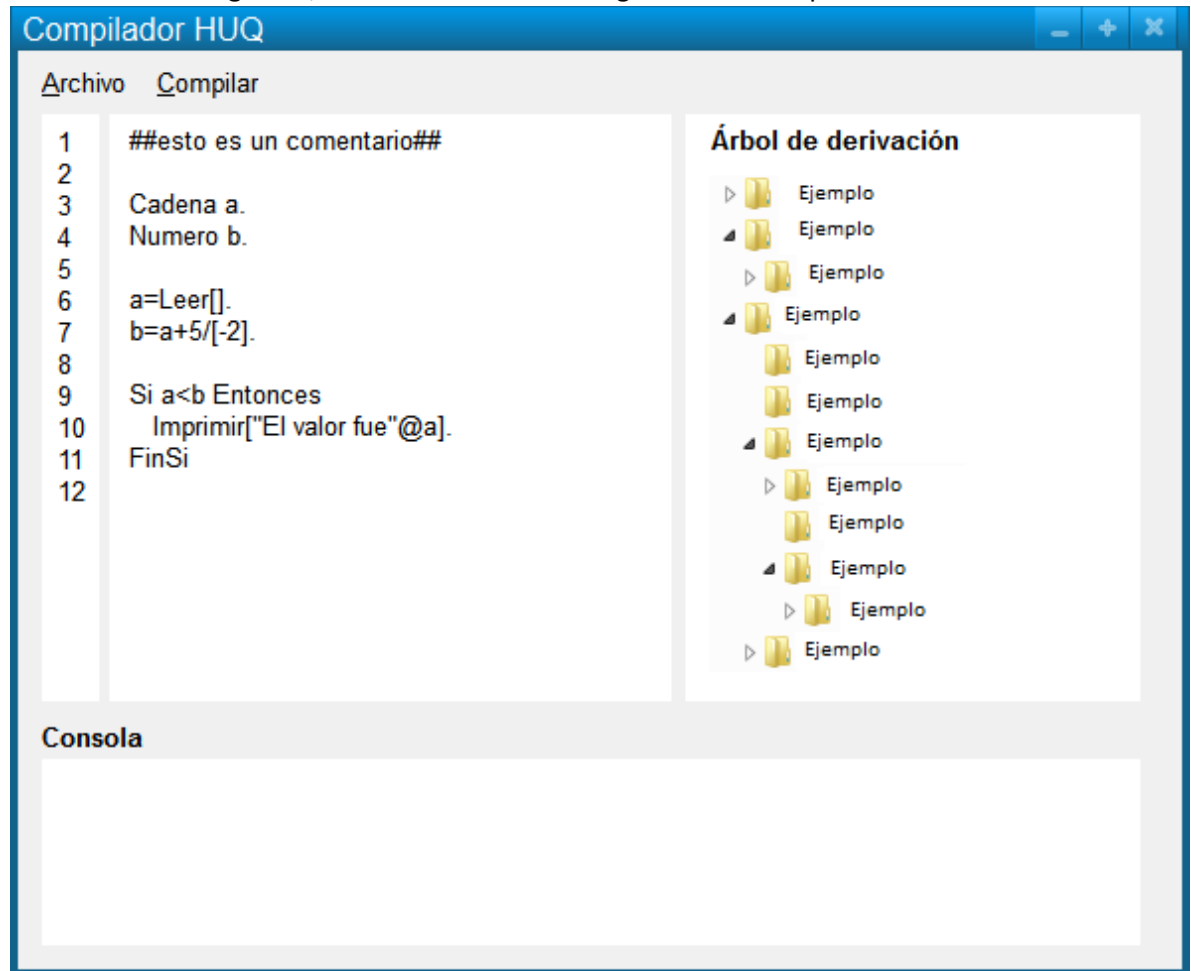


Proyecto Final Compiladores

FASE 2 Analizador Sintáctico

Para la fase 2 del proyecto final de compiladores, se deberá realizar una aplicación usando JAVACC, esta aplicación deberá realizar las funcionalidades descritas a continuación.

1. Se deberá implementar la interfaz gráfica
2. Al abrir la interfaz gráfica, Se mostrará una interfaz gráfica como la que se ve a continuación:



- Tiene un menú, con 2 opciones Archivo y Compilar
 - Tiene una zona de edición que es donde se escribe el código que vamos a traducir
 - Tiene una zona de árbol de derivación, que es donde se muestra en forma explorador de carpetas el árbol de derivación.
 - Tiene una consola, donde se mostrarán los errores que se detecten o si todo esta bien, el mensaje de éxito
3. Menú Archivo: Contiene un submenú con las siguientes opciones
 - a. Abrir: Permite abrir archivos con extensión .huq, el cual contiene el código a analizar. Al escoger el archivo este se cargará en la zona de Edición
 - b. Guardar Como: Si no hemos abierto un archivo y por el contrario iniciamos de cero un código fuente en la zona de edición, al dar clic en guardar como, se guardará el archivo, se pide el nombre y debe quedar con extensión .huq

- c. Nuevo: Borra lo que este en la zona de edición y empieza un nuevo archivo, por ahora sin nombre, hasta que le den guardar como.
 - d. Guardar: Guarda en el archivo seleccionado los cambios generados en el editor.
- 4. Menú Compilar: Al dar clic en este ítem de menú, se realizará el proceso de Análisis gramatical. Si hay cambios no guardados deberá indicarlo.
- 5. Proceso de Análisis gramatical: Este proceso incluye
 - a. Verificar que el código no tenga errores sintácticos, en caso de tenerlos deberá reportarlos en la zona de errores. Debe recuperarse de los errores
 - b. Debe crear el árbol de derivación.

Entregables (Se debe subir todo a classroom, en caso de pesar mucho, favor subirlo a un drive o dropbox y poner en el comentario la url.)

- 1. Código fuente de la aplicación
- 2. Ejecutable de la aplicación (.jar)
- 3. Un informe, donde se muestre los tokens y expresiones regulares, la gramática en BNF, como implementaron la recuperación de errores, los problemas presentados durante el desarrollo.
- 4. 2 ejemplos de código fuente sin errores
- 5. 2 ejemplos de código fuente con errores
- 6. Un video demostrando el funcionamiento del programa.