

Proyecto: Etapa 5

MiniJava 2018: Compiladores e Intérpretes

Germán Alejandro Gómez

16/11/2018

INDICE

Instrucciones para ejecutar.....	3
Casos de prueba.....	4
Diagramas de clases.....	5
Consideraciones de diseño	8
Logros	10

Instrucciones para ejecutar.

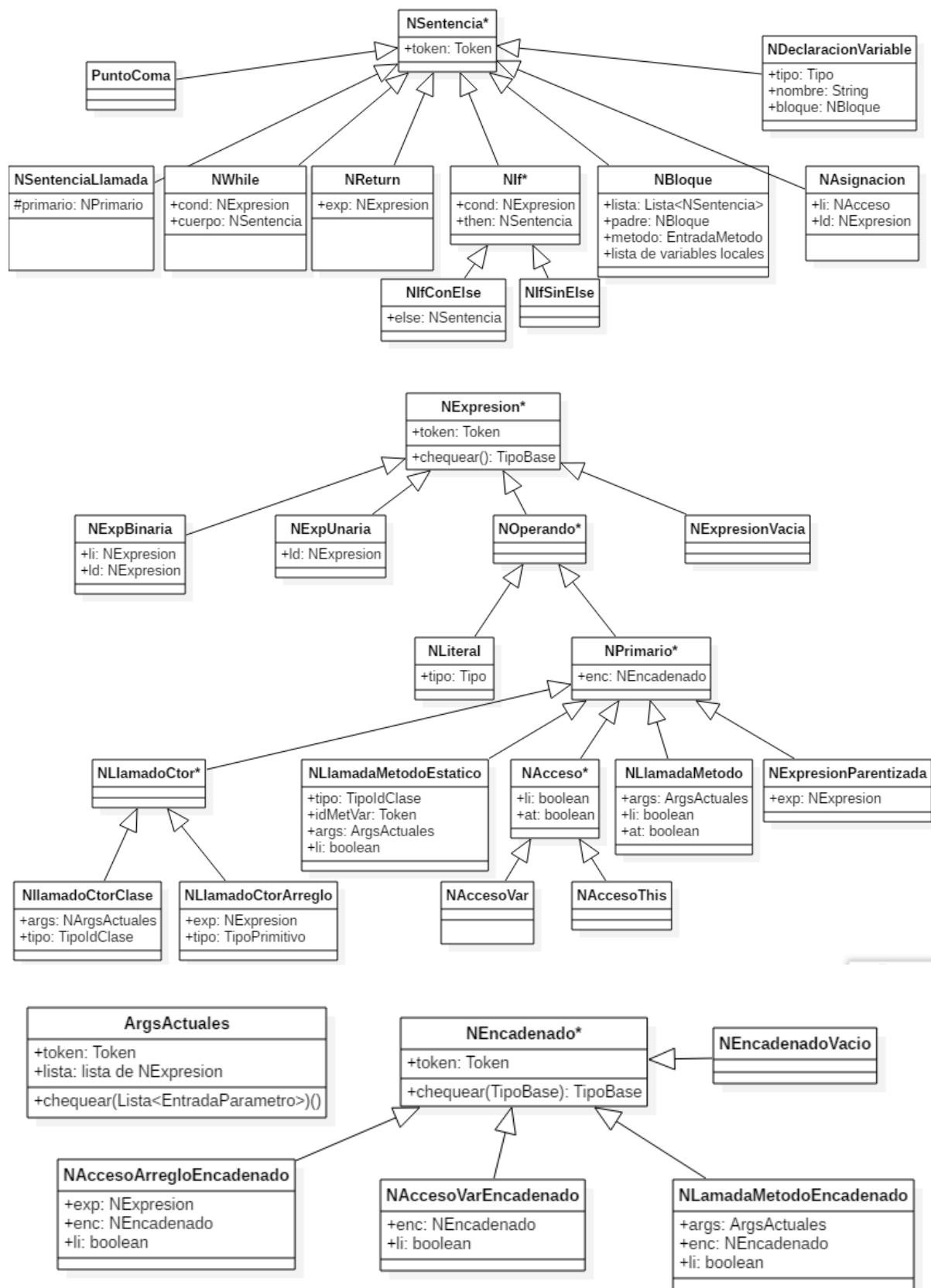
Para poder utilizar el programa hace falta seguir las siguientes instrucciones:

1. Abrir una consola en Windows, por ejemplo cmd.
2. Usando cmd, ir al directorio donde se encuentra Principal.java.
3. Utilizar el comando 'javac Principal.java' para compilar el código.
4. Para ejecutarlo hay que ingresar la siguiente sentencia: 'java Principal <IN_FILE> <OUT_FILE>'. IN_FILE es el archivo a compilar. OUT_FILE es el archivo generado con código intermedio. Cualquier error léxico, sintáctico y semántico se mostrará por consola.

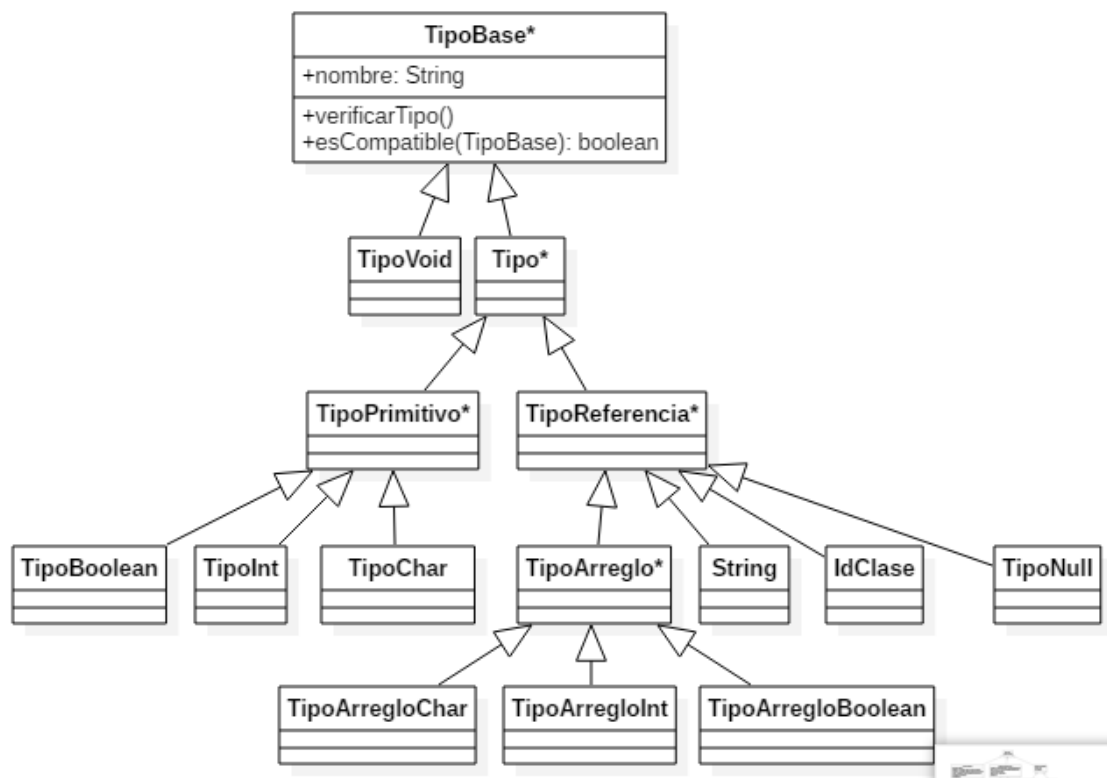
Casos de prueba

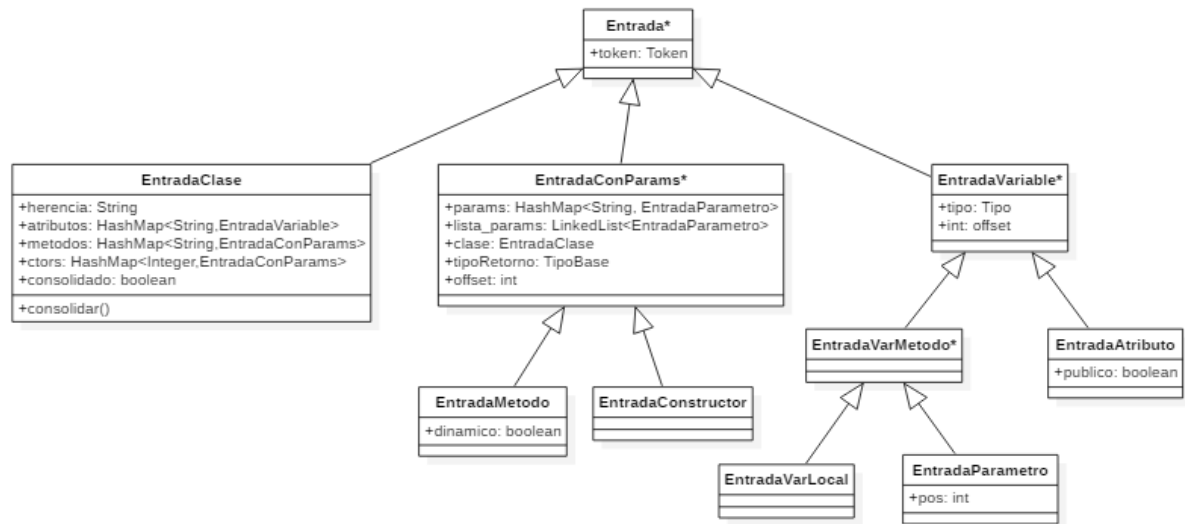
Se adjunta una carpeta con el nombre test. Cada archivo corresponde a un caso de prueba, y el resultado esperado, se especifica en forma de comentario dentro de cada uno.

Diagramas de clases



TablaDeSimbolos
+clases: HashMap +claseActual: EntradaClase +metodoActual: EntradaConParams +metodoMain: EntradaMetodo +errores: boolean
+addClase(EntradaClase) +addAtributos(EntradaVariable[]) +addConstructor(EntradaConParams) +addMetodo(EntradaMetodo) +hayStaticMain(): boolean +chequear() +consolidar()





Consideraciones de diseño

- Se corrigieron los errores especificados de la etapa 4 que correspondía al logro recuperación de errores de declaración.
- El compilador hace una pasada para chequear el código a través de los AST creados en el analizador sintáctico. Luego de esto se procede a realizar una nueva pasada donde se genera el código maquina.
- Calculo de offset:
 - o Métodos: al final de consolidar los métodos de la clase, se recorren todos los métodos dinámicos y se le asigna un offset, dependiendo de su contexto, ya sea que es heredado, redefinido o nuevo. El primer método tiene offset 1, el segundo 2, y el ultimo método tiene offset n.
 - o Atributos de instancia: al final de consolidar los métodos de la clase, se recorren todos los atributos y se le asigna un offset, dependiendo de su contexto, ya sea que es heredado o nuevo. El primer atributo tiene offset 0, el segundo 1, y el ultimo atributo tiene offset n-1.
 - o Parámetros en métodos/constructores: al final de consolidar los métodos de la clase, se recorren todos los parámetros y se le asigna un offset. El offset del parámetro se calcula como: $cons + cantidad\ de\ parámetros\ del\ método - posición\ de\ parámetro$, donde cons es 3 si el método es dinámico y 2 si es estático.
 - o Variables locales: en la generación código se le asigna un offset. La primer variable local tiene offset 0, la segundo -1, y a si sucesivamente.
- Las etiquetas de los métodos se generan a través de método `getEtiqueta()` que se encuentra en `EntradaConParams`, el cual se compone de “ `mt_<nombre_metodo> $ <Cantidad_parametros> _ <Clase_del_metodo>` ”

- Las etiquetas de los constructores se generan a través de método `getEtiqueta()` que se encuentra en `EntradaConstructor`, el cual se compone de “ `ctor_ <Clase_del_ctor> $ <Cantidad_parametros>` ”
- En el ejemplo `(System.println(new int[1][0]));` el analizar semántico arroja error. Esta decisión fue consultada vía mail con Sebastián Gottifredi.

Logros

Se intentan cumplir los logros de:

- Inicializaciones Inline controladas (etapa 4)
- Inicializaciones Inline generadas (etapa 5)
- Generación Sobrecargada (etapa 5)