





-  Laboratorio 2: Sistema de Tipos con ANTLR
 -  Descripción General
 -  Instrucciones de Configuración
 -  Entregables

Laboratorio 2: Sistema de Tipos con ANTLR

Descripción General

En este laboratorio trabajarás con **ANTLR**, un generador de analizadores sintácticos. Hemos proporcionado un **Dockerfile** para ayudarte a configurar el entorno rápidamente. Utilizaremos Python para hacer pruebas, ya que es más sencillo que Java para pruebas pequeñas.

Experimentarás con un sistema de tipos básico, extenderás una gramática y completarás el sistema de tipos. Con ello, aprenderás sobre la marcha lo básico al utilizar sistemas de tipos en el análisis semántico.

- **Modalidad: Individual**

Instrucciones de Configuración

1. **Construir y Ejecutar el Contenedor Docker** Desde el directorio raíz de este laboratorio, ejecuta el siguiente comando para construir la imagen y lanzar un contenedor interactivo:

```
docker build --rm . -t lab2-image && docker run --rm -ti -v  
"$(pwd)/program":/program lab2-image
```

2. Entender el Entorno

- El directorio **program** se monta dentro del contenedor.
- Este contiene la **gramática de ANTLR**, un archivo **Driver.py** (punto de entrada principal) y un archivo **program_test.txt** (entrada de prueba).



- En este caso usamos un Visitor para visitar los nodos del árbol y aplicar análisis semántico.
- También se un Listener para este efecto.

3. Generar Archivos de Lexer y Parser: Dentro del contenedor, compila la gramática ANTLR a Python con:

```
antlr -Dlanguage=Python3 -visitor SimpleLang.g4          *** Esto es
para utilizar un Visitor ***
antlr -Dlanguage=Python3 -listener SimpleLang.g4        *** Y esto es
para utilizar un Listener ***
```

4. Ejecutar el Analizador Usa el driver para analizar el archivo de prueba:

```
python3 Driver.py program_test_pass.txt
python3 DriverListener.py program_test_pass.txt
```

-  Si el archivo es sintácticamente correcto y, además, no hay problemas de tipo, **se mostrará que la validación de tipos fue exitosa.**
-  Si existen errores sintácticos, o errores de tipo, ANTLR los mostrará en la consola.



Entregables

- **Deben utilizar ambos Visitor y Listener para realizar las actividades de este lab.**
- Analice la ejecución con los archivos provistos, comente acerca de porqué el archivo "pass" si "pasa" y por qué el archivo "no pass" pues, "no pasa" lol.
- Extienda la gramática de ANTLR para incluir otras dos operaciones, las que sean de su agrado.
- Ahora extienda más el sistema de tipos para validar al menos otros 3 conflictos de tipos.
- **Video de YouTube no listado** (pero público) con los resultados de ejecutar los puntos anteriores y sus comentarios.
- Repo de Github con todo su código.