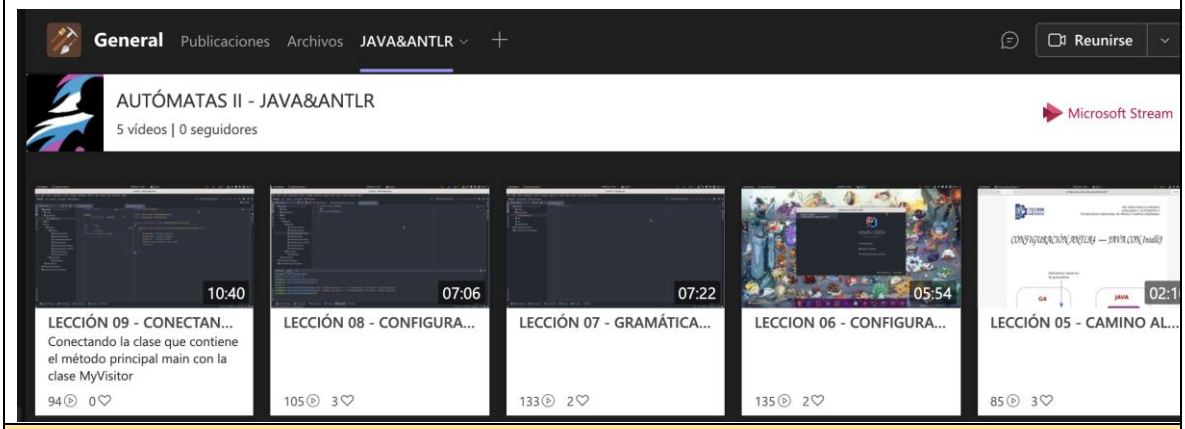
ACTIVIDAD DE LABORATORIO	**05**
NOMBRE DE LA ACTIVIDAD	Conexión ANTLR4 - JAVA
OBJETIVO DE APRENDIZAJE	Implementar la conexión Java con ANTLR4
INSTRUCCIONES	Sigue los videos para poder lograr la configuración pertinente
NOMBRE ESTUDIANTE	Itsai Zempoaltecatl Mejia
GRUPO	6SA
FECHA	23 febrero 2022

Saludar.g4

Sigue la serie de videos incluidos en la pestaña llamada -Configuración JAVA-ANTLR que encontrarás en TEAMS.

- Elaborar la práctica que aparece en la LECCIÓN 09
- Elaborar una guía general para conectar la gramática con java
- Poner una conclusión

HAPPY CODING!

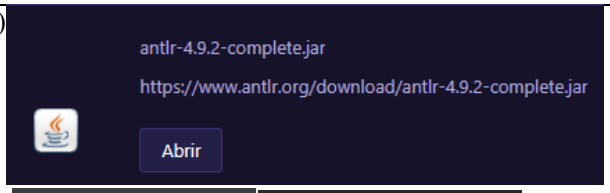


PRODUCTO A ENTREGAR

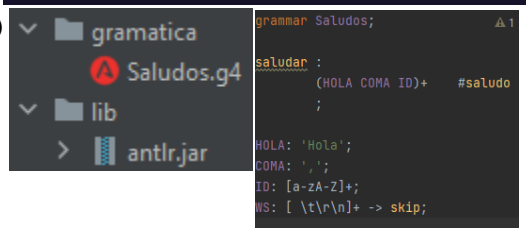
Nota: Previamente tenemos que instalar Intelij y crear un proyecto java.

1) Descargar el .jar de antlr
2) Creamos 2 carpetas “lib” y “gramática”, en lib agregamos nuestro archivo antlr.jar y en gramática creamos nuestro archivo .g4 para la gramática y llenamos nuestro archivo .g4 con nuestra gramática

1)

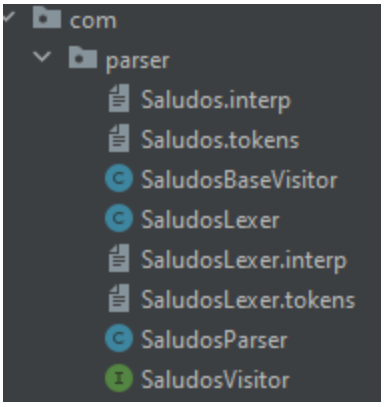


2)

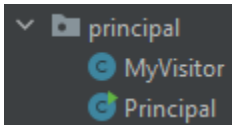


3) Para crear nuestros archivos .java para conectar antlr con java usaremos el siguiente comando donde -package define el paquete donde se generaran los archivos, -o la carpeta donde se guardaran, -no-listener evita que se generen archivos listener y -visitor genera los archivos visitors los cuales son muy importantes

3)
```
antlr -package com.parser -o ..\src\com\parser -no-listener -visitor .\Saludos.g4
```



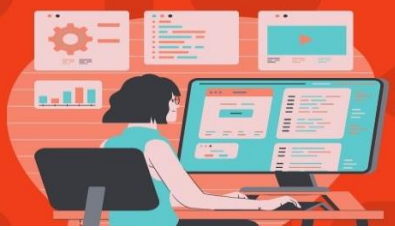
4) Una vez con los archivos .java generados a partir de nuestra gramatica, agregamos una carpeta "principal" donde crearemos los archivos "MyVisitor.java" que será el que nos ayudara con la comunicación entre antlr y java. Por otro lado crearemos un archivo "Principal.java" donde probaremos si hay comunicación.

4)


5) Dentro de "MyVisitor.java" importamos las clases "BaseVisitor" y "Parser" asi como el método heredado "visit" el cual contiene **context (ctx)** que es el encargado de devolverlos el valor de la gramatica.

5)
```
MyVisitor.java
1  package com.principal;
2
3  import com.parser.SaludosBaseVisitor;
4  import com.parser.SaludosParser;
5
6  public class MyVisitor extends SaludosBaseVisitor<String> {
7      @Override public String visitSaludo(SaludosParser.SaludoContext ctx)
```

6) Hasta este punto tenemos comunicación entre antlr y java. Nota: Cada que hagamos una modificación en el archivo .g4 tenemos que repetir el punto 3 (solo ese)	



LENGUAJES Y AUTÓMATAS II

ME XENIA PADILLA MADRID

1) MyVisitor es la clase que editaremos en el método heredado “visit”, dentro de este podremos trabajar con context para analizar los datos que contiene la entrada (archivo de texto).

1)

MyVisitor.java

```
package com.principal;

import com.parser.SaludosBaseVisitor;
import com.parser.SaludosParser;

public class MyVisitor extends SaludosBaseVisitor<String> {
    @Override public String visitSaludo(SaludosParser.SaludoContext ctx) {

//        String hola = visit(ctx.HOLA());
//        String coma = visit(ctx.COMA());
//        String id = visit(ctx.ID());
//        System.out.println(hola + coma + id);
        System.out.println("Hay un saludo ");
        System.out.println(ctx.children);
        return null;
    }
}
```

2) En “Principal.java” probaremos la conexión entre java y antlr que acabamos de crear. Por medio de las siguientes líneas de código (hasta ParseTree) leeremos la entrada y con MyVisitor mandaremos a llamar el método visit que nos devolverá un resultado en base a la entrada.

2)

```
import com.parser.SaludosLexer;
import com.parser.SaludosParser;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.CharStreams;
import org.antlr.v4.runtime.CommonTokenStream;
import org.antlr.v4.runtime.tree.ParseTree;

public class Principal {
    public static void main(String[] args)  throws Exception{
        CharStream input = CharStreams.fromFileName("prueba.txt");
        SaludosLexer lexico = new SaludosLexer(input);
        CommonTokenStream tokens = new CommonTokenStream(lexico);
        SaludosParser sintactico = new SaludosParser(tokens);
        ParseTree arbol = sintactico.saludar();
        MyVisitor visitas = new MyVisitor();
        visitas.visit(arbol);
```

3) Entrada y salida (consola)

3)

prueba.txt

```
Hola, Zempo
Hola, Zempoo
Hola, Zempooo
```

```
Hay un saludo
[Hola, ,, Zempo, Hola, ,, Zempoo, Hola, ,, Zempooo]
```

CONCLUSIÓN: En el video se menciona el resolver porque no muestra 3 mensajes de “Hay un saludo” esto se debe a que context como tal guarda todas las entradas como un solo array, entonces al evaluar la entrada solo devuelve un resultado (el que se ve en la salida) y no 3 como era de esperarse, es por esto que solo vemos un mensaje en lugar de 3