

# LENGUAJES Y AUTÓMATAS II

## ME XENIA PADILLA MADRID



### ACTIVIDAD DE LABORATORIO

06

### NOMBRE DE LA ACTIVIDAD

Detector inteligente de palabras

### OBJETIVO DE APRENDIZAJE

Aplicar una gramática libre de contexto y a través del uso de la semántica, realizar búsquedas inteligentes en un archivo de texto.

Elaborar un analizador de palabras reservadas para un lenguaje de frutas; el sistema funcionará de la siguiente manera:

Se le proporcionará una lectura como entrada al sistema, el cual deberá identificar las frutas de temporada que hay en todo el año, evitando reconocer aquellas frutas que terminen con la letra «e», p.e. —aguacate—

### INSTRUCCIONES

El sistema deberá ser inteligente y ser capaz de reconocer las frutas en singular y plural (fresa, fresas, Fresa, Fresas, FRESAS, FRESA), en este caso solo deberá identificar que se trata de una sola fruta, «fresa»

Cuando se identifique(n) la(s) fruta(s), deberá mencionar los meses que más abunda, p.e.

**fresa**; abunda los meses de: enero, febrero, marzo, abril, mayo, noviembre, diciembre

### NOMBRE ESTUDIANTE

Itsai Zempoaltecatl Mejia

### GRUPO

6SA

### FECHA

2/marzo/2022

# LENGUAJES Y AUTÓMATAS II

## ME XENIA PADILLA MADRID



- 1) Se creo una gramática que es capaz de detectar los meses, frutas en sus variantes de escritura (mayus/minus).

**frutas** contiene todas las opciones de frutas, **meses** es la combinación de meses y frutas, ya que en la entrada del texto contiene un mes y las frutas de esa temporada. el lexema de grupo permite varias entradas de este tipo.

- 2) Se crean los .java para usar los métodos del visitor.

- 3) Se creo una clase "MyVisitor.java" en la cual usaremos los métodos heredados que visita a cada mes y en un HashMap (memory) guardamos el valor del mes (ctx.ENERO) como key y de frutas (ctx.frutas) como value. Es necesario recorrer ctx.frutas() para obtener cada una de las frutas que vienen en el texto. Esto se replicara en cada visit de cada mes para obtener todos los meses y todas las frutas de cada uno de ellos.

1)

```
grammar Alimentos;  
  
grupo : (meses)+;  
meses :  
    ENERO frutas+ #enero  
    |  
    FEBRERO frutas+ #febrero  
    |  
    MARZO frutas+ #marzo
```

```
FRESA : ('fresa' | 'fresas' | 'Fresa' | 'Fresas' | 'FRESA' | 'FRESAS');  
GUAYABA : ('guayaba' | 'guayabas' | 'Guayaba' | 'Guayabas' | 'GUAYABA' | 'GUAYABAS');  
JAMAICA : ('jamaica' | 'jamaicas' | 'Jamaica' | 'Jamaicas' | 'JAMAICA' | 'JAMAICAS');
```

```
ENERO : 'Enero';  
FEBRERO : 'Febrero';  
MARZO : 'Marzo';
```

```
frutas :  
    FRESA #fresa  
    |  
    GUAYABA #guayaba  
    |  
    JAMAICA #jamaica  
    |  
    LIMA #lima
```

2)

```
Alimentos.interp  
Alimentos.tokens  
AlimentosBaseVisitor  
AlimentosLexer  
AlimentosLexer.interp  
AlimentosLexer.token  
AlimentosParser  
AlimentosVisitor
```

3)

```
public static HashMap<String, List<String>> memory = new HashMap<>();  
  
@Override public String visitEnero(AlimentosParser.EneroContext ctx) {  
    String mes = ctx.ENERO().getText();  
    List<String> frutas = new ArrayList<>();  
    for (int i=0; i<ctx.frutas().size(); i++){  
        frutas.add(ctx.frutas(i).getText());  
    }  
  
    memory.put(mes, frutas);  
    return null;  
}
```

# LENGUAJES Y AUTÓMATAS II

## ME XENIA PADILLA MADRID



4) Ya que el resultado deseado es saber en que meses se encuentra cada una de las frutas, lo que se hizo fue extraer todas las frutas existentes y agregarlas en un nuevo HashMap donde la key es el nombre de la fruta y el value es una lista de los meses en los cuales esta contenida esa fruta. El primer foreach recorre el hasmap memory para extraer las frutas y agregarles su meses, el segundo recorre el nuevo hashmap creado en el anterior y a su vez muestra el resultado deseado.

5) Por ultimo, en nuestro main usamos las sentencias necesarias para leer el archivo y analizarlo. Despues mandamos a llamar el método "ShowList()" y con esto finalizamos el proyecto.

```
4) public void ShowList(){
    HashMap<String,List<String>> frutaMes = new HashMap<>();

    for (String key: memory.keySet()) {
        List<String> value = memory.get(key);
        for (int f=0; f<value.size(); f++){
            String currentFruta = value.get(f).toLowerCase(Locale.ROOT);
            if(!frutaMes.containsKey(currentFruta)){
                frutaMes.putIfAbsent(currentFruta, new ArrayList<String>());
                frutaMes.get(currentFruta).add(key);
            }else{
                frutaMes.get(currentFruta).add(key);
            }
        }
    }

    for (String key: frutaMes.keySet()) {
        System.out.print("[*] "+key+" abunda en los meses: ");
        List<String> meses = frutaMes.get(key);
        for(int m=0; m<meses.size(); m++){
            System.out.print(meses.get(m)+" ");
        }
        System.out.println();
    }
}
```

```
5) public static void main(String[] args) throws Exception{
    CharStream input = CharStreams.fromFileName("Food.txt");
    AlimentosLexer lexico = new AlimentosLexer(input);
    CommonTokenStream tokens = new CommonTokenStream(lexico);
    AlimentosParser sintactico = new AlimentosParser(tokens);
    ParseTree arbol = sintactico.grupo();
    MyVisitor visitas = new MyVisitor();
    visitas.visit(arbol);
    visitas.ShowList();
}
```

# LENGUAJES Y AUTÓMATAS II

## ME XENIA PADILLA MADRID

### 6) Resultados.

6)

```
[*] melón abunda en los meses: junio, enero, abril, mayo, julio, septiembre,  
[*] caña abunda en los meses: diciembre,  
[*] papaya abunda en los meses: diciembre, diciembre, noviembre, octubre, enero, abril, marzo, febrero, mayo, julio, agosto, septiembre,  
[*] tuna abunda en los meses: julio, agosto, septiembre,  
[*] limón abunda en los meses: junio, diciembre, diciembre, noviembre, octubre, enero, abril, marzo, febrero, mayo, julio, agosto, septiembre,  
[*] cereza abunda en los meses: octubre,  
[*] plátano abunda en los meses: junio, diciembre, diciembre, noviembre, octubre, enero, abril, marzo, mayo, julio, agosto, septiembre,  
[*] pera abunda en los meses: junio, diciembre, diciembre, noviembre, octubre, mayo, julio, agosto, septiembre,  
[*] mandarina abunda en los meses: diciembre, diciembre, noviembre, octubre, enero, marzo, febrero,  
[*] jamaica abunda en los meses: enero,  
[*] chabacano abunda en los meses: junio, mayo,  
[*] tamarindo abunda en los meses: enero, abril, marzo, febrero,  
[*] higo abunda en los meses: junio, julio, agosto,  
[*] tejocote abunda en los meses: diciembre,  
[*] toronja abunda en los meses: junio, diciembre, noviembre, octubre, enero, abril, marzo, febrero, mayo, julio, agosto, septiembre,  
[*] chichazapote abunda en los meses: diciembre,  
[*] sandía abunda en los meses: junio, abril, marzo, febrero, mayo, julio, septiembre,  
[*] maney abunda en los meses: junio, abril, marzo, febrero, mayo,  
[*] uva abunda en los meses: junio, octubre,  
[*] guayaba abunda en los meses: diciembre, noviembre, octubre, enero, agosto, septiembre,  
[*] ciruela abunda en los meses: junio, mayo, julio, agosto,  
[*] lima abunda en los meses: diciembre, noviembre, enero,  
[*] naranja abunda en los meses: diciembre, diciembre, noviembre, octubre, enero, abril, marzo, mayo, septiembre,  
[*] uvas abunda en los meses: noviembre, julio, agosto,  
[*] chirimoya abunda en los meses: diciembre,  
[*] mango abunda en los meses: junio, diciembre, abril, marzo, febrero, mayo, julio, agosto,  
[*] durazno abunda en los meses: junio, octubre, julio, agosto, septiembre,  
[*] granada abunda en los meses: diciembre, noviembre, octubre, enero, febrero, julio, agosto, septiembre,  
[*] fresa abunda en los meses: diciembre, noviembre, enero, abril, marzo, febrero, mayo,  
[*] membrillo abunda en los meses: octubre, julio, agosto, septiembre,  
[*] capulín abunda en los meses: agosto,  
[*] piña abunda en los meses: junio, diciembre, diciembre, enero, abril, febrero, mayo, julio, agosto,
```

**CONCLUSIÓN:** En la clase BaseVisitor trabajamos con cada Visit de los tokens, la información que almacenan vienen en el Context de cada token cuando el Lexema coincide.

Para este ejemplo se uso un HashMap que va almacenando los valores de cada uno de los contextos y con este mismo fue con el que se trabajo al final.

Fue necesario “invertir” el acomodo de los valores, ya que se empezó como key los meses y value las frutas, pero para obtener el resultado como se solicita las frutas se convirtieron en key y los meses en value.