



**Tecnológico Nacional de México**  
**Instituto Tecnológico de Tepic**  
**Ingeniería en Sistemas Computacionales**



**Proyecto**  
**Compilador – FarmBot**

**Lenguajes y Automatas II**  
**Tema 2. Generación de código intermedio**  
**Docente: Sheyla Janett Tapia García**  
**Grupo: 5A**

**Datos de los alumnos:**

- Fonseca Lares Uriel Alejandro.
- Guizar Escobar Andrea Vianney.
- Luna García Angel Yaniv.
- Moreno Duran Erik.

**Tepic, Nayarit, a lunes 07 de abril de 2025**

# FarmBot

## Descripción del lenguaje de programación.

El lenguaje de programación desarrollado en el proyecto "Los Escombros" es un lenguaje diseñado para la automatización agrícola mediante el uso de sensores y actuadores. Permite el monitoreo de variables ambientales y la toma de decisiones automatizadas para optimizar el uso de recursos en cultivos.

Este lenguaje incluye características específicas para la programación de robots agrícolas, tales como comandos para movimiento, monitoreo de sensores y control de plagas.

### Características principales:

- Monitoreo de variables ambientales como temperatura, humedad, pH y peso de los cultivos.
- Automatización de procesos agrícolas como riego, aplicación de insecticidas y control de plagas.
- Integración con sensores y robots agrícolas para optimizar el uso de recursos.
- Definición de comandos específicos para la manipulación de cultivos.

### Tokens del lenguaje.

Los tokens en este lenguaje incluyen:

Operadores
<ul style="list-style-type: none"><li>→ SUMA (+)</li><li>→ RESTA (-)</li><li>→ DIVISION (/)</li><li>→ MULTIPLICACION (*)</li><li>→ ASIGNACION (=)</li><li>→ IGUAL (==)</li><li>→ DIFERENTE (!=)</li><li>→ MAYORQUE (&gt;)</li><li>→ MENORQUE (&lt;)</li><li>→ MENORIGUAL (&lt;=)</li><li>→ MAYORIGUAL (&gt;=)</li></ul>

### Palabras reservadas

- BEGIN, END
- TRUE, FALSE
- IF, ELSE, WHILE, FOR
- RETURN, FUN, FROM
- IMPORT
- Tipos de datos: INT, BOOL, STG, REAL

### Tokens específicos del proyecto

- **Movimientos:** MOVERADELANTE, MOVERATRAS, GIRARIZQUIERDA, GIRADERECHA, DETENER, VELOCIDAD
- **Sensores y datos obtenidos:** SENSORTEMPERATURA, SENSORHUMEDAD, SENSORPH, SENSORPESO, SENSORCONTADOR
- **Monitoreo de cultivos:** MEDIRALTURA, MEDIRCE, REVISARCOLOR, REVISARTAMANO
- **Control y cuidado:** CONTROLPLAGAS, INSECTICIDA, FUNGICIDA, ASPERSION, APLICAR, REGISTRAR

### Delimitadores y símbolos de puntuación

- PUNTO (.)
- COMA (,)
- PUNTOCOMA (;)
- COMILLASIMPLE ('')
- COMILLADOBLE ("")
- PARENTESIS\_A, PARENTESIS\_B ((, ))
- LLAVE\_A, LLAVE\_C ({, })
- CORCHETE\_A, CORCHETE\_B ([, ])

### Gramáticas.

La gramática del lenguaje de "Los Escombros" sigue un enfoque basado en reglas formales que definen cómo deben estructurarse las sentencias y expresiones dentro del código, algunas de las gramáticas en las que se basa el lenguaje de programación son las siguientes.

## 1. Estructura General del Programa

Un programa en este lenguaje tiene una estructura delimitada por las palabras reservadas BEGIN y END. Dentro de este bloque se define la lógica de ejecución.

### Regla:

```
programa ::= "BEGIN" bloque_codigo "END"  
          | plantas "BEGIN" bloque_codigo "END"
```

- Un programa puede comenzar directamente con BEGIN seguido de un bloque\_codigo y finalizar con END.

## 2. Bloques de Código

El bloque de código es el núcleo del programa y contiene declaraciones, estructuras de control y comandos.

### Regla:

```
bloque_codigo ::= "{" lista_declaraciones "}"
```

- Todo el código está encerrado entre llaves { ... }.
- Dentro del bloque, se pueden escribir múltiples declaraciones.

## 3. Declaraciones de Variables

El lenguaje permite definir variables asignándoles valores específicos.

### Regla:

```
declaracion ::= tipo ID "=" expresion ";"
```

- Una declaración comienza con un tipo de dato (INT, BOOL, REAL, STG, COLOR).
- Luego se escribe un identificador (ID) que representa el nombre de la variable.
- Se usa el operador de asignación (=) para darle un valor.
- La declaración finaliza con un punto y coma (;), que es obligatorio.

## 4. Expresiones

Las expresiones pueden ser valores simples o combinaciones de operaciones aritméticas y lógicas.

### Regla:

```
expresion ::= expresion operador expresion
           | "(" expresion ")"
           | NUMERO
           | ID
```

- Las expresiones pueden incluir operaciones matemáticas y lógicas.
- Se permite el uso de paréntesis para establecer prioridad.
- Se pueden utilizar números (NUMERO) o variables (ID).

## 5. Operadores

El lenguaje tiene operadores aritméticos, lógicos y de comparación.

### Regla:

```
operador ::= "+" | "-" | "*" | "/" | "==" | "!=" | ">" | "<" |
           ">=" | "<="
```

- Operadores matemáticos: **+**, **-**, **\***, **/**.
- Operadores de comparación: **==**, **!=**, **>**, **<**, **>=**, **<=**.

## 6. Condicionales (IF-ELSE)

Permiten tomar decisiones basadas en una condición.

### Regla:

```
si ::= "IF" "(" expresion ")" bloque_codigo
     | "IF" "(" expresion ")" bloque_codigo "ELSE" si
```

- Se usa IF seguido de una condición entre paréntesis.
- Si la condición es verdadera, se ejecuta el bloque\_codigo.
- Opcionalmente, se puede agregar ELSE para ejecutar otro bloque si la condición es falsa.

## 7. Bucles (WHILE, FOR)

Sirven para ejecutar código repetitivamente.

### 7.1 Bucle WHILE

Repite el bloque de código mientras se cumpla una condición.

**Regla:**

```
mientras ::= "WHILE" "(" expresion ")" bloque_codigo
```

- WHILE ejecuta el bloque\_codigo mientras la expresion sea verdadera.

### 7.2 Bucle FOR

Se usa para repetir un bloque con una estructura definida.

**Regla:**

```
for_loop ::= "FOR" "(" for_init ";" for_condicion ";"  
for_actualizacion ")" bloque_codigo
```

- Se define un inicio (for\_init), una condición de parada (for\_condicion) y una actualización (for\_actualizacion).

## 8. Comandos para Control.

Este lenguaje permite controlar sensores y actuadores.

**Regla:**

```
comando ::= "MOVERADELANTE" "(" expresion ")" ";"  
          | "GIRARIZQUIERDA" "(" expresion ")" ";"  
          | "ASPERSION" ";"
```

```
| "REGISTRAR" "(" expresion ")" ";"
```

- Se pueden dar comandos para mover un robot (MOVERADELANTE, GIRARIZQUIERDA).
- Se pueden aplicar tratamientos (ASPERSION).
- Se pueden registrar datos (REGISTRAR).

## 9. Declaración de Plantas y Sensores

El lenguaje permite definir características de cultivos.

**Regla:**

```
planta ::= "PLANTA" ID bloque_parametros
```

- Se declara una planta con un ID y un bloque\_parametros con información sobre ella.

**Link a carpeta con vídeo y proyecto.**

<https://drive.google.com/drive/folders/1vBN93PuSwj4s8j781kajy9fH8U9bHkEx>