

IC5701 - Compiladores e intérpretes

Gramática de TerraCode

Estudiantes:

Jozafath Pérez Fernández - 2023107460

Alana Calvo Bolaños - 2022040915

Mary Paz Álvarez Navarrete - 2023138604

José Gabriel Jiménez Chacón - 2021128841

Docente:

SANABRIA RODRIGUEZ AURELIO

Año:

IS 2025

Tabla de contenidos

Introducción.....	3
Estructura de Repetición.....	3
Ejemplo:.....	3
Estructura de Bifurcación.....	3
Ejemplo:.....	3
Variables y Tipos de Datos.....	4
Ejemplo:.....	4
Soporte de Funciones.....	4
Ejemplo:.....	4
Soporte de Comentarios.....	5
Ejemplo:.....	5
Soporte de Textos, números enteros y flotantes.....	5
Ejemplo:.....	5
Expresiones Matemáticas.....	5
Ejemplo:.....	6
Expresiones condicionales(para la estructura de bifurcación).....	8
Ejemplo:.....	8
Funciones de Ambiente Estándar.....	8
Ejemplo:.....	8
Estructura de Datos Personalizada.....	9
Ejemplo:.....	9
Gramáticas libres de contexto.....	10

Introducción

TerraCode es un lenguaje de programación inspirado en la naturaleza y la jardinería. Sus palabras reservadas están diseñadas para facilitar la comprensión a través de términos ecológicos y botánicos.

Estructura de Repetición

Fotosíntesis: Equivalente al bucle “**while**”. Se repite mientras la condición sea verdadera.

Ejemplo:

```
Fotosintesis (humedad > 30) {  
    Podar(#Regar las plantas#)  
}
```

Estructura de Bifurcación

Exterior: Equivalente a “**if**”.

Interior: Equivalente a “**else**”.

Ejemplo:

```
Exterior (luz > 50) {  
    Podar(#Demasiada luz, poner sombra#)  
} Exterior (luz < 10) {  
    Podar(#Muy poca luz, encender lampara#)  
} Interior {  
    Podar(#Luz adecuada#)  
}
```

Variables y Tipos de Datos

Maceta: Para la asignación de una **variable local**.

Semilla: Para la asignación de una **variable global**.

Asignación a variables: =

Ejemplo:

```
Semilla temperatura  
  
Sembrar funcion() {  
    Maceta humedad  
    humedad = 70  
    temperatura = 30  
} Florecer
```

Soporte de Funciones

Sembrar: Para la asignación de **iniciar** una función.

Florecer: Para la asignación de **finalizar** una función.

Ejemplo:

```
Sembrar funcion() {  
} Florecer
```

Soporte de Comentarios

... : Para la asignación de **ignorar línea**.

: Para la asignación de **hacer prints**(serían nuestras comillas).

Ejemplo:

```
*Esto es un comentario*  
Podar(#Esto es un texto#)
```

Soporte de Textos, números enteros y flotantes

Texto: Para la asignación de [A-Z]?[a-z]?[0-9]?[ÁÉÍÓÚáéíóúÑñÜü]*

Enteros: Para la asignación de (-+)?[0-9]+

Flotantes: Para la asignación de [-+]?(0-9)?\.[0-9]+

Ejemplo:

```
*Ejemplo de un texto 12*
```

```
Podar(#1234 camión#)
```

```
Podar(#urgüento#)
```

```
Maceta entero1
```

```
Maceta entero2
```

```
entero1 = 25478
```

```
entero2 = -3
```

```
Maceta flotante1
```

```
Maceta flotante2
```

```
flotante1 = 23.89
```

```
flotante2 = - 1.36
```

Expresiones Matemáticas

Operaciones Básicas:

Se decidió usar los operadores ya conocidos en las operaciones matemáticas y de comparación.

- Operaciones matemáticas:

- **Suma:** SUMATERA
- **Resta:** RESEDA
- **Multiplicación:** BAMBU
- **División:** DIOSMA
- **División Entera:** DALIA
- **Módulo:** MAGNOLIA

- Comparación:

- **Mayor que:** SECUOYA

- **Menor que:** BONSÁI
- **Igual que:** CACTUS
- **Mayor o igual:** ROSA
- **Menor o igual:** LIRIO

-Especiales:

- **INJERTO** (unión de dos plantas): Es el equivalente a la expresión lógica de **AND/and**.
- **DIOICA** (planta que tiene flores masculinas o femeninas pero no las dos): Es el equivalente a la expresión lógica de **OR/or**.
- **ÁCARO**: Es equivalente a la expresión lógica de **NOT/not**.

Ejemplo:

- Operaciones matemáticas:

```
Sembrar calculos() {
    *Calcular la cantidad de plantas*
    Maceta plantas
    Maceta rosas
    Maceta orquideas

    rosas = 3
    orquideas = 4
    plantas = rosas SUMATERA orquideas

    *Calcular los sacos disponibles*
    Maceta sacosDisponibles
    Maceta sacosUsados

    sacosDisponibles = 20
    sacosUsados = 6
    sacosDisponibles = sacosDisponibles RESEDA sacosUsados

    *Calcular los litros de agua requerida para las rosas*
    Maceta agua
    agua = rosas BAMBÚ 1.5

    *Calcular los metros de cerca requeridas por sector*
    Maceta cercaDisponible
    Maceta cercaSector

    cercaDisponible = 15}
```

```

cercaSector = cercaDisponible DIOSMA 3

*Cuantas macetas alcanzan x cantidad de semillas*
Maceta macetas
Maceta semillas

semillas = 20
macetas = semillas DALIA 3
semillas = semillas MAGNOLIA 3

} Florecer

```

- Operaciones de comparación:

```

Exterior (semillas SECUOYA sacosDisponible) {
    Podar(#Hay que comprar sacos#)
} Exterior (semillas BONSÁI sacosDisponible) {
    Podar(#Ya casi no hay semillas#)
} Exterior (semillas CACTUS sacosDisponible) {
    Podar(#Apenas alcanzan#)
} Exterior ((semillas ROSA sacosDisponible) DIOICA (semillas LIRIO
sacosDisponibles)) {
    Podar(#Revise los sacos y las semillas#)
} Interior {
    Podar(#Algo pasó con los sacos y las semillas#)
}

```

- Especiales:

```

Exterior (humedad >= 70 INJERTO temperatura >= 30) {
    Podar(#Las plantas se están quemando y muriendo de sed#)
}

Exterior (humedad < 70 DIOICA temperatura < 30) {
    Podar(#Las plantas están bien frescas#)
}

Exterior ACARO humedadTierra {
    Podar(#Sus plantas requieren hidratación YA#)
}

```

Expresiones condicionales (para la estructura de bifurcación)

Viva: Para asignar que una variable es **verdadera**.

Muerta: Para asignar que una variable es **falsa**.

Ejemplo:

```
Exterior (humedadTierra == Viva) {  
    Podar(#La tierra está lo suficientemente húmeda#)  
} Exterior (humedadTierra == Muerta) {  
    Podar(#La tierra está tan seca, que la planta se va a morir#)  
} Interior {  
    Podar(#El sensor se desconectó, no hay lecturas#)  
}
```

Funciones de Ambiente Estándar

Podar: Esta función **imprime** el mensaje que se ponga en ella como **print**.

Medir: Devuelve el **largo** de la variable que se especifique como el **len**.

Recolectar: Permite **recibir texto** introducido en la terminal como el **input**.

Ejemplo:

```
Sembrar inventario() {  
    Maceta semillas  
    semillas = Recolectar(#Introduzca la cantidad de semillas: #)  
  
    Maceta largo  
    largo = Medir(semillas)  
  
    Exterior(semillas == 0) {  
        Podar(#Por favor ingrese el número de semilla#)  
    }  
} Florecer
```

Estructura de Datos Personalizada

- Switch con ciclo

Planta() : Nombre de la estructura

Flor: Va a ser la forma de decir que es la variable que se va a usar para el switch

Hoja(): Caso

```
Planta(Condicional) {  
    Instrucción para el ciclo  
    Flor valor = Asignación  
  
    Hoja("valor para a comparar") {  
        Instrucción del case  
    }  
    Hoja() {  
        Instrucción del case default  
    }  
}
```

Cosechar = Return

Madurar = int

Germinar = float

Gramáticas libres de contexto

Programa ::= Instrucciones

Instrucciones ::= Instrucion (Instrucciones)*

Instrucion ::= Comentario | Declaracion | Asignacion | Funcion | Repeticion | Condicional | Imprimir | Largo | Entrada | CicloCasos

Comentario ::= "*" Texto "*"

Declaracion ::= TipoVar Identificador

TipoVar ::= "Maceta" | "Semilla"

Asignacion ::= Identificador "=" Expresion

Funcion ::= "Sembrar" Identificador "(" ")" "{" Instrucciones "}" "Florecer"

Repeticion ::= "Fotosintesis" "(" Expresion ")" "{" Instrucciones "}"

Condicional ::= If+ Else?

If ::= "Exterior" "(" Expresion ")" "{" Instrucciones "}"

Else ::= "Interior" "{" Instrucciones "}"

Expresion ::= ExpresionLogica | ExpresionAritmetica | ExpresionComparacion |
ExpresionIgualdad | ExpresionNegacion | "(" Expresion ")" | ValorEstandar

ValorEstandar ::= Identificador | Entero | Flotante | String | ValorBooleano

ExpresionLogica ::= ExpresionLogicaTerm (OpLogico ExpresionLogicaTerm)*

ExpresionLogicaTerm ::= ExpresionComparacion | ExpresionIgualdad | "(" ExpresionLogica ")" | ValorBooleano | ExpresionNegacion

ExpresionComparacion ::= ExpresionAritmetica OpComparacion ExpresionAritmetica

ExpresionIgualdad ::= (ExpresionAritmetica | String | ValorBooleano) "CACTUS"
(ExpresionAritmetica | String | ValorBooleano)

ExpresionAritmetica ::= ExpresionAritmeticaTerm (OpAritmetico ExpresionAritmeticaTerm)*

ExpresionAritmeticaTerm ::= Entero | Flotante | Identificador | "(" ExpresionAritmetica ")"

ExpresionNegacion ::= "ACARO" ExpresionLogica

OpLogico ::= "INJERTO" | "DIOICA"

OpAritmetico ::= "SUAMTERA" | "RESEDA" | "BAMBÚ" | "DIOSMA" | "DALIA" |
"MAGNOLIA"

OpComparacion ::= "SECUOYA" | "BONSÁI" | "ROSA" | "LIRIO"

ValorBooleano ::= "Viva" | "Muerta"

Imprimir ::= "Podar" "(" String ")"

Largo ::= "Medir" "(" Identificador ")"

Entrada ::= "Recolectar" "(" String ")"

CicloCasos ::= "Planta" "(" Expresion ")" "{" Instrucciones Casos CasoDefault "}"

Casos ::= ("Hoja" "(" (String | Entero | Flotante | ValorBooleano)") "{" Instrucciones "}")+

CasoDefault ::= "Hoja" "(" ")" "{" Instrucciones "}"

String ::= "#" Texto "#"

Texto ::= [A-Z]?[a-z]?[0-9]?[ÁÉÍÓÚáéíóúññüü]*

Identificador ::= [a-zA-Z][a-zA-Z0-9]*

Entero ::= ("+"|"-")?[0-9]+

Flotante ::= ("+"|"-")?[0-9]+."[0-9]+