

An interpreter for a robot language

Contenido

- **Objetivo**
- **El Lenguaje**
- **Descripción de los módulos**
- **Descripción de las fases**
- **Las Librerías**
- **Ejemplos**

Objetivo

Desarrollar un lenguaje que:

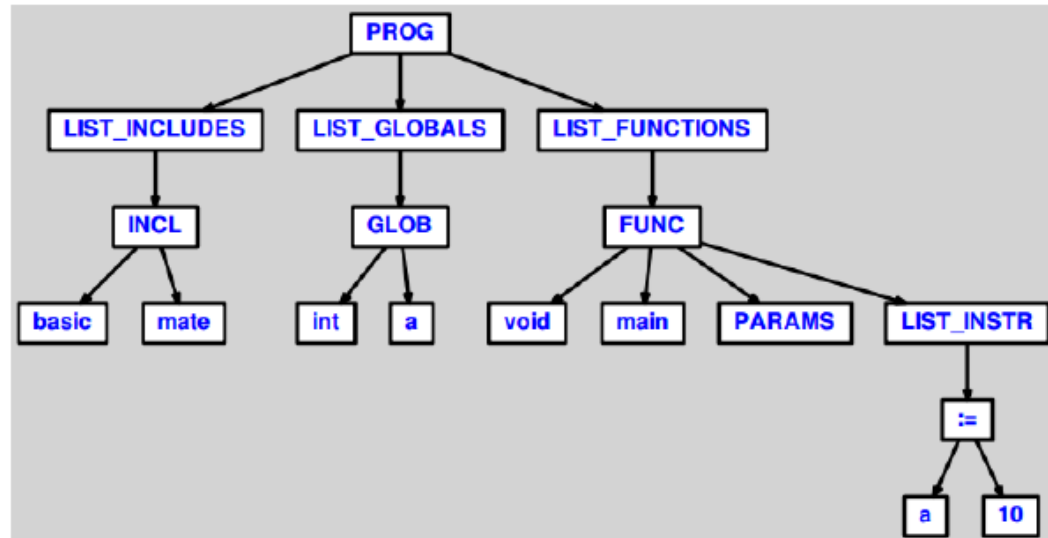
- Sea sencillo y a su vez potente
- Interactúe con el robot *LEGO Mindstorms nxt 2.0*
- Permita usar acciones predefinidas



El Lenguaje

Estructura básica de un programa:

- Includes *
- Globales *
- Funciones +



El Lenguaje

Funcionalidades

- Todas las que posee un lenguaje imperativo
- Interacción con el mundo real:
 - lectura de sensores
 - movimiento de motores

El Lenguaje

Interacción con el mundo real:

- Sensores:

- Sensor de Color

`c = COLOR(1)`

`c.getColor()`

- Sensor de Tacto

`t = TOUCH(4)`

`t.getTouch()`

- Sensor de Ultrasonido

`us = ULTRA(3)`

`us.getUltrasonic()`

El Lenguaje

Interacción con el mundo real:

- Motores:

- Creadora

- `m1 = MOTOR(1)`

`m2 = MOTOR(3)`

- Setter

- `m1.setSpeed(num)`

`m2.stop()`

- `m2.avanzar([num,bool])`

`m1.retroceder([num,bool])`

- Getter

- `m2.getSpeed()`

El Lenguaje

Otras funciones:

`sleep([num])`

Includes y Objetos:

- Includes:

`include paquete/clase`

- Objetos:

`o = OBJECT(clase)`

`o.funcionClase([params])`

Descripción de los módulos

Módulos:

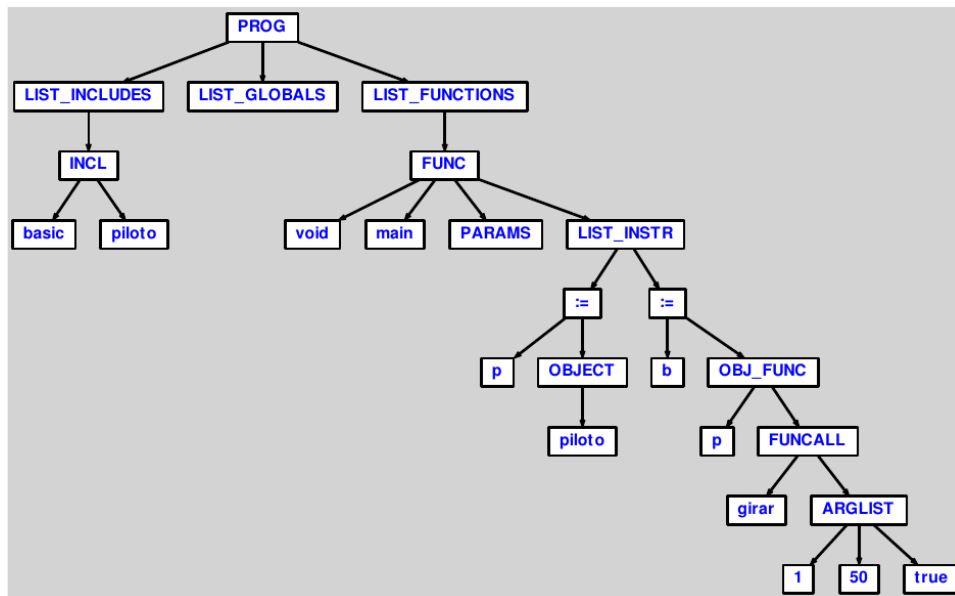
- Analizador
- Traductor
- Software/Hardware Externo

Descripción de los módulos

Analizador:

```
include basic/piloto
```

```
void main()  
  p = OBJECT(piloto);  
  b = p.girar(1,50,true);  
endfunc
```



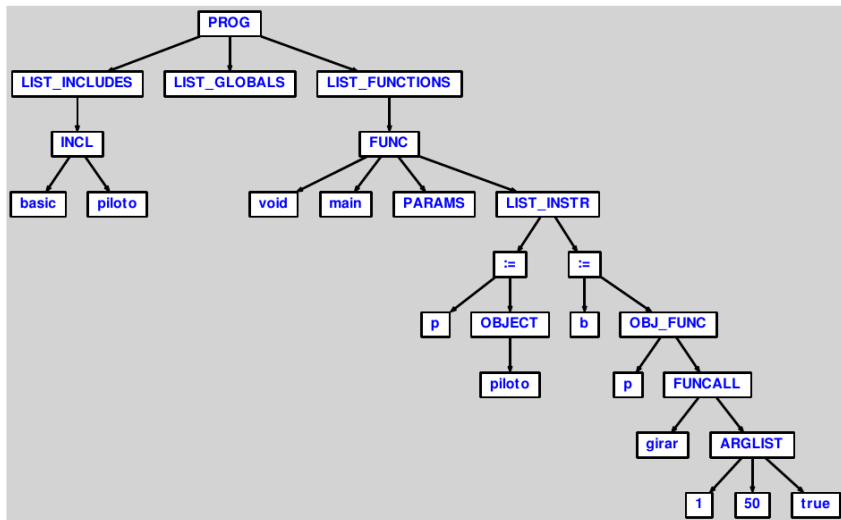
Descripción de los módulos

Traductor:

- Compilación de includes
- Recopilación información:
 - funciones de includes
 - variables globales
- Type Check → Traducción de instrucción

Descripción de los módulos

Traductor:



Traducción a leJOS

```
import basic.piloto;
import lejos.nxt.*;
import lejos.util.Delay;

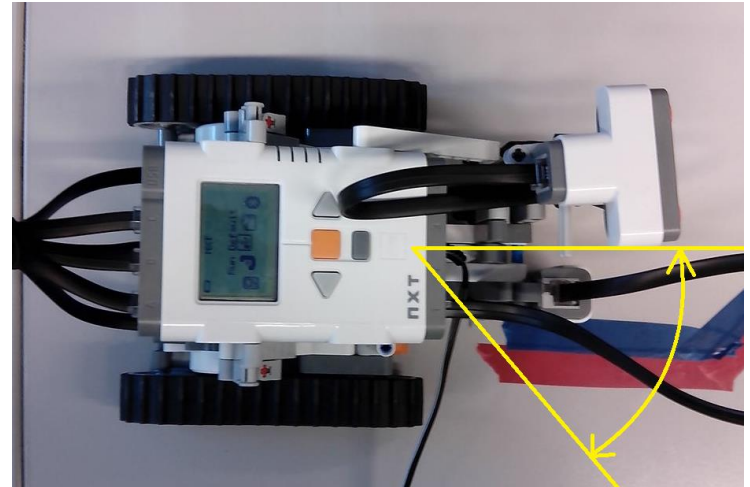
public class basico {
    public static void main(String args[]) {
        piloto p = new piloto();
        boolean b = p.girar(1, 50, true);
    }
}
```

Descripción de los módulos

Software/Hardware Externo:

LEGO Mindstorms nxt 2.0

```
import basic.piloto;  
import lejos.nxt.*;  
import lejos.util.Delay;  
  
public class basico {  
    public static void main(String args[]) {  
        piloto p = new piloto();  
        boolean b = p.girar(1, 50, true);  
    }  
}
```



Descripción de las fases

Fases:

- Familiarización con Robots y recopilación info
- Desarrollo de Gramática de nuestro lenguaje
- Type check y Traducción a lejos
- Añadir potencia: Librerías y Objetos
- Testing, testing, testing

Las Librerías

Sonar:

- Usa el sensor ultrasónico
- Acciones:
 - Busca objetos cercanos
 - Mantiene distancia con obstáculos
 - Elige el giro con menos obstáculos

Las Librerías

Sonar:

```
void laberintoRapido(int lim)
    i = 0;
    while i < lim do
        write "Caminando hasta tope";
        despejado = checkUltra(10);
        while(despejado) do
            mover(50);
            despejado = checkUltra(10);
        endwhile;
        write "Tope encontrado";
        g = grados();
        write "Girando";
        write g;
        M1.avanzar(6 * g, true);
        M2.retroceder(6 * g, false);
        i = i + 1;
    endwhile;
endfunc
```

```
void mover(int g)
    M1.avanzar(g,true);
    M2.avanzar(g,true);
endfunc

bool checkUltra(int tope)
    d = U.getUltrasonic();
    return d > tope;
endfunc
```


Las Librerías

Sonar:

```
int grados()
  girarUltra(- 75);
  grad = 0;
  dist = 0;
  i = -75;
  cerca = true;
  while i <= 75 do
    d = U.getUltrasonic();
    if (d > dist) then
      dist = d;
      grad = i;
    endif;
    if (d > 20) then
      cerca = false;
    endif;
    girarUltra(10);
    i = i + 10;
  endwhile;
```

```
  girarUltra(- 85);
  if (cerca) then grad = 90; endif;
  return grad;
endfunc
```

```
void girarUltra(int g)
  M.avanzar(g,false);
  sleep(100);
endfunc
```

Las Librerías

Piloto:

- Objeto para el manejo fácil de un cierto robot
- Acciones:
 - definir características de nuestro robot
 - mover el robot de manera intuitiva
 - funcionalidad seguir líneas 1 y 2 colores

Las Librerías

Piloto:

```
bool girar(int dire,int grados, bool check)
    if dire>=0 then
        izq.avanzar(relacion*2*grados,true);
        while (izq.isMoving() and ( (check and checkSensors()) or not check)) do
            sleep(100);
        endwhile;
    else
        der.avanzar(relacion*2*grados,true);
        while (der.isMoving() and ( (check and checkSensors()) or not check)) do
            sleep(100);
        endwhile;
    endif;
    checkeo = (check and checkSensors()) or not check;
    if not checkeo then
        stopMov();
    endif;
    return checkeo;
endfunc
```

Las Librerías

Piloto:

```
bool girarInSitu(int dire,int grados, bool check)
    if dire<0 then
        grados = -1*grados;
    endif;|
    izq.avanzar(relacion*grados,true);
    der.retroceder(relacion*grados,true);
    return checkMotorSensor(check);
endfunc
```

```
bool mover(int dire,int grados,bool check)
    if dire<0 then
        grados = -1*grados;
    endif;
    izq.avanzar(grados,true);
    der.avanzar(grados,true);
    return checkMotorSensor(check);
endfunc
```

```
void moverNoLimit(int dire)
    if dire<0 then
        izq.retroceder();
        der.retroceder();
    else
        izq.avanzar();
        der.avanzar();
    endif;
endfunc
```

```
void stopMov()
    izq.parar();
    der.parar();
endfunc
```

Las Librerías

Mate:

- Contiene funciones matemáticas básicas
 - Exponenciación
 - Absoluto

Ejemplos

Ejemplo 1: Resolución de un laberinto

1. Avanza hasta que encuentra un obstáculo
2. Busca una dirección sin obstáculos
3. Repite 1 y 2 unas X veces

Ejemplos

Ejemplo 2: Sigue una línea bicolor

1. Lee color izq y der de la línea a seguir
2. Avanza y realiza lectura de color
3. Si la lectura es el color izq o der de la línea → 2
4. Decide si salida por la izq o der
5. Corrige hacia izq o der según decisión punto 4
6. Si consigue volver a la línea → 2
7. Termina ejecucion

Ejemplos

Ejemplo 3: Seguir un objeto

1. Busca el objeto más cercano
2. Gira y avanza en esa dirección
3. Repite 1 y 2 unas X veces

FIN