

# Lenguajes para programar robots

(P2: T7)

Yeray Méndez Romero  
[yeray.mendez@udc.es](mailto:yeray.mendez@udc.es)

Daniel Rivera López  
[d.rivera1@udc.es](mailto:d.rivera1@udc.es)

# ÍNDICE

- Introducción.
- Programación de robots:
  - Nivel gestual.
  - Nivel textual.
- Relación de los lenguajes de programación de robots y los habituales.
- Lenguajes en empresas.
- Lenguajes en educación.
- Conclusión
- Bibliografía.

# INTRODUCCIÓN



- Robot:
  - Viene del vocablo checo “robota”, que significa “servidumbre”, “trabajo forzado” o “esclavitud”, especialmente los llamados “trabajadores alquilados” que vivieron en el Imperio Austrohúngaro hasta 1848.
  - Sistema electromecánico conducido por un cierto programa que le permite realizar una determinada tarea de forma autónoma.

# PROGRAMACIÓN DE ROBOTS

- Existen dos niveles de programación:
  - Gestual



FIG.17

- Textual

```
1  ' Globales -----
2  Var Variable0:Booleano
3  Var Variable1:Cadena
4  ' Fin Globales -----
5  Proc Procedimiento ' <- Procedimiento sin retorno.
6    Var Variable2:Entero ' Locales
7    Var Variable3:Real
8
9    Si Variable0 = Falso Entonces ' Condición "If"
10     Contar Variable2 = 0 a 9 ' Bucle "For"
11     Variable1 = Variable1 + "1"
12     Seguir ' "End For"
13   FinSi ' "End If"
14
15   Variable3 = 5.13
16 FinProc
```

# PROGRAMACIÓN DE ROBOTS

## NIVEL GESTUAL

- Gestual:
  - El robot interviene en el proceso como un digitalizador de posiciones.
  - No es necesaria la experiencia del usuario, pues se realiza con métodos sencillos:
    - Moviendo manualmente el robot a la posición deseada.
    - Sistema maestro-esclavo.
    - Mando.

# PROGRAMACIÓN DE ROBOTS

## NIVEL TEXTUAL

- Textual:
  - El movimiento del robot y su relación con el entorno es definida mediante un programa. Es necesaria experiencia en programación.
  - Existen tres tipos:
    - Nivel robot
    - Nivel objeto
    - Nivel tarea

# PROGRAMACIÓN DE ROBOTS

## NIVEL TEXTUAL

- Nivel robot:
  - Orientados a definir la secuencia de operaciones para que el robot realice la tarea.
  - Ejemplos:
    - ANORAD
    - EMILY

# PROGRAMACIÓN DE ROBOTS

## NIVEL TEXTUAL

- Nivel objeto:
  - Las instrucciones se dan en función de los objetos a manejar.
  - Ejemplos:
    - RAPT y AUTOPASS utilizados en el ensamblaje de piezas.

PLACE C<sub>1</sub>

SUCH THAT C<sub>1</sub> BOT CONTACTS C<sub>2</sub>TOP

AND B<sub>1</sub> A<sub>1</sub> IS ALIGNED WITH C<sub>2</sub>A<sub>1</sub>

AND B<sub>1</sub> A<sub>2</sub> IS ALIGNED WITH C<sub>2</sub>A<sub>2</sub>



# PROGRAMACIÓN DE ROBOTS

## NIVEL TEXTUAL

- Nivel tarea:
  - El programador define las acciones que debe ejecutar el robot.
  - El sistema decide qué movimientos y comprobaciones sensoriales debe realizar y en que orden. Estas decisiones se toman en función de :
    - Los objetivos propuestos.
    - El estado del mundo del robot

# PROGRAMACIÓN DE ROBOTS

## NIVEL TEXTUAL

- Nivel tarea:
  - Ejemplos:
    - STRIPS: lenguaje formal utilizado por el planificador con el mismo nombre por el primer robot que podía razonar sobre sus acciones, Shakey.

```
Initial state: At(A), Level(low), BoxAt(C), BananasAt(B)
Goal state:   Have(bananas)
```

Actions:

```
// move from X to Y
_Move(X, Y)_
Preconditions: At(X), Level(low)
Postconditions: not At(X), At(Y)

// climb up on the box
_ClimbUp(Location)_
Preconditions: At(Location), BoxAt(Location), Level(low)
Postconditions: Level(high), not Level(low)

// climb down from the box
_ClimbDown(Location)_
Preconditions: At(Location), BoxAt(Location), Level(high)
Postconditions: Level(low), not Level(high)

// move monkey and box from X to Y
_MoveBox(X, Y)_
Preconditions: At(X), BoxAt(X), Level(low)
Postconditions: BoxAt(Y), not BoxAt(X), At(Y), not At(X)

// take the bananas
_TakeBananas(Location)_
Preconditions: At(Location), BananasAt(Location), Level(high)
Postconditions: Have(bananas)
```

# RELACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN DE ROBOTS Y LOS HABITUALES

- Podemos agrupar los lenguajes de programación de robots en dos tipos fundamentales:
  - Extensiones de los lenguajes habituales
  - Lenguajes diseñados específicamente para programar robots.

# RELACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN DE ROBOTS Y LOS HABITUALES

- Extensiones de los lenguajes habituales:
  - Lenguajes como C , Java, Python, Matlab, se les añade las funcionalidades requeridas para programar robots a través de librerías importadas.
  - Las diferencias con los lenguajes habituales solo estaría en las librerías importadas.

# RELACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN DE ROBOTS Y LOS HABITUALES

- Lenguajes diseñados específicamente para programar robots:
  - Normalmente diseñados por una empresa fabricante de robots para emplearlo en sus propias unidades.
  - Las principales diferencias con los lenguajes habituales son:
    - Posesión de funciones de movimiento,.
    - Control de sensores.
    - Un menor número de tipos de datos y estructuras de control.
    - Menor complejidad.

# LENGUAJES EN EMPRESAS

- Ámbito empresarial: Muchos lenguajes.
- Robotics Business Review(RBR) es la empresa que elabora la lista anual RBR50 en la que se encuentran las empresas más innovadoras, influyentes y exitosas en el ámbito de la robótica. En la de este año encontramos a :
  - ABB con el lenguaje RAPID.
  - Fanuc con el lenguaje KAREL.
  - Adept con V+.

# LENGUAJES EN EMPRESAS

- RAPID:
  - Lenguaje de programación textual de alto nivel.
  - Utilizado actualmente en la controladora IRC5 de ABB.
  - Estructura:
    - Una aplicación RAPID consta de:
      - Programa que está compuesto por:
        - » Rutina principal(main).
        - » Datos del programa
        - » Subrutinas.
      - Módulos del sistema.

# LENGUAJES EN EMPRESAS

- RAPID:

```
%%%  
VERSION:1  
LANGUAGE:ENGLISH  
%%%  
MODULE EJEMPLO  
  CONST robtarget A:=[[0,0,0],[0,0,0,0],[0,-1,0,0], [9E+09,...]];  
  
  CONST tooldata pinza:= [TRUE, [[0,0,0],[1,0,0,0]],  
                           [0,[0,0,0],[1,0,0,0],0,0,0]];  
  
  PROC cerrar_pinza()  
    Set spinza;  
  ENDPROC  
  
  PROC coger_pieza()  
    MoveJ B1,v100,z5,pinza;  
    MoveL B,v80,fine,pinza;  
    cerrar_pinza;  
  ENDPROC
```

```
  PROC main()  
    CONST dionum listo:=1;  
    abrir_pinza;  
    WHILE TRUE DO  
      MoveJ A,v100,fine,pinza;  
      WaitDI econtrol,listo;  
      coger_pieza;  
      MoveL B1,v80,z5,pinza;  
      MoveJ D,v100,z100,pinza;  
      MoveJ C1,v100,z5,pinza;  
      MoveL C,v80,fine,pinza;  
      abrir_pinza;  
      MoveL C1,v80,z5,pinza;  
    ENDWHILE  
  ENDPROC  
ENDMODULE
```



# LENGUAJES EN EMPRESAS

- KAREL:
  - Lenguaje que combina características de alto y bajo nivel.
  - Similar a Pascal:
    - Variable fuertemente tipadas.
    - Constantes.
    - Tipos personalizados.
    - Procedimientos.
    - Funciones de movimiento y movimiento concurrente.
  - Lenguaje compilado: Fichero.kl se traduce a un fichero.pc para ser cargado y ejecutado en el controlador.

# LENGUAJES EN EMPRESAS

- KAREL:

```
PROGRAM hello
-- this is a comment

-- translator directive
%COMMENT = 'Hello'

-- const declarations
CONST
    LANG_EN = 1
    LANG_ES = 2
    LANG_FR = 3

-- custom type
TYPE
    person_t = STRUCTURE
        name      : STRING[16]
        lang_id   : INTEGER
    ENDSTRUCTURE

-- program vars
VAR
    people : ARRAY[3] OF person_t

-- custom routines
-- Returns the proper greeting for a given language
ROUTINE greeting(lang_id : INTEGER) : STRING
BEGIN
    SELECT lang_id OF
        CASE (LANG_EN) :
            RETURN ('Hello')
        CASE (LANG_ES) :
            RETURN ('Hola')
        CASE (LANG_FR) :
```

# LENGUAJES EN EMPRESAS

- V+:
  - Lenguaje de programación interpretado y estructurado de alto nivel que proporciona :
    - Inteligibilidad.
    - Fiabilidad.
    - Adaptabilidad.
    - Transportabilidad.
    - Estructura:
      - Cada línea se interpreta como una instrucción del programa y poseen prioridades asignables.

# LENGUAJES EN EMPRESAS

- V+:

Algoritmo:

```
PROGRAM cogergarra(garra, salir)
LEFTY ; comportamiento de brazo izquierdo
APPRO garra, 100 ; mueve la base a 100mm encima de la garra
DELAY(0.5) ; retardo de 0.5 segundos
OPENI ; abre los enganches de la base
DELAY(0.5)
MOVES garra ; mueve la base hasta entrar en la garra
DELAY(0.5)
CLOSEI ; cierra los enganches de la garra
DELAY(0.5)
MOVES salir ; saca la garra de su plataforma
DELAY(0.5)
DEPARTS 100 ; sube la garra 100mm
```

# LENGUAJES EN EMPRESAS

- Ámbito empresarial: Muchos lenguajes.
- Problema:
  - Cada empresa solo tiene en cuenta sus necesidades en cuanto a software.
  - El código no es portable debido a :
    - Lenguajes propietarios .
    - Fabricantes solo soportan ciertos sensores y controladores.
    - Interfaces no estandarizadas.

# LENGUAJES EN EMPRESAS

- RESULTADO :
  - Usuario final se queda bloqueado en un determinado fabricante.
  - Pérdida de tiempo y económica.



# LENGUAJES EN EMPRESAS

- En respuesta a esta situación han surgido proyectos tratando de proporcionar un lenguaje estándar.
- ROS-Industrial, que es de código abierto y extiende las capacidades de ROS, Robot Operating System:
  - Conjunto de herramientas y librerías que simplifican la tarea de crear un comportamiento complejo.

# LENGUAJES EN EMPRESAS

- Miembros actuales de ROS-Industrial





# LENGUAJES EN EDUCACIÓN

- En la actualidad los robots:
  - Forma educativa de interesar a la juventud en el mundo de la programación, la IA y la robótica.
- Compañías dedicadas a la educación:
  - No suelen tener un lenguaje propietario.
  - Permiten programar con lenguajes habituales.
- Ejemplos:
  - Scratch.
  - Arduino.

# LENGUAJES EN EDUCACIÓN

- Scratch:
  - Lenguaje de programación visual y educativo.
  - Desarrollado por Lifelong Kindergarten Group en el MIT .
  - Paradigma imperativo.
  - Dirigido por eventos.
  - Primera versión en 2003.

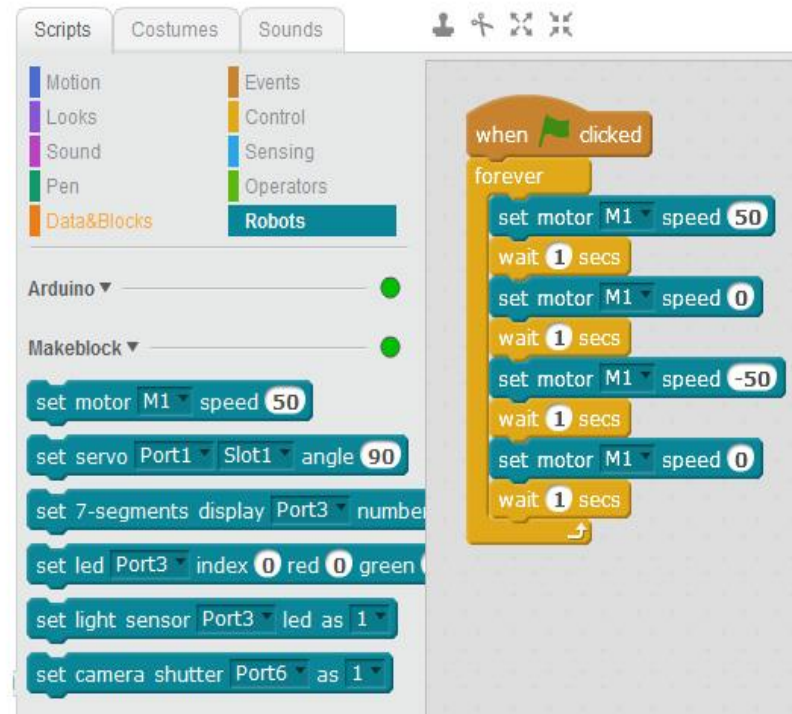
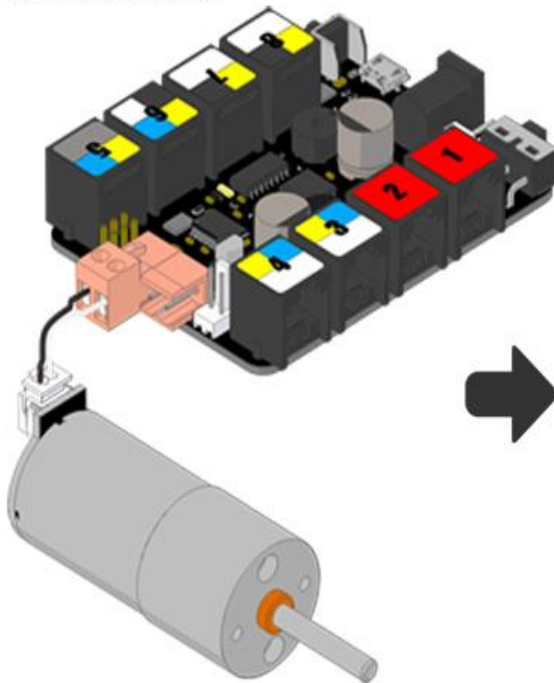
# LENGUAJES EN EDUCACIÓN

- Ejemplos de empresas que usan Scratch:
  - RoboThink.
  - Phiro.
- Ejmplos de empresas que usan Scratch para generar código en otro lenguaje:
  - Scratch4Arduino de ProtoCREA.
  - Mblock utilizado por Makeblock.

# LENGUAJES EN EDUCACIÓN

- Scratch:

Controlar un motor DC

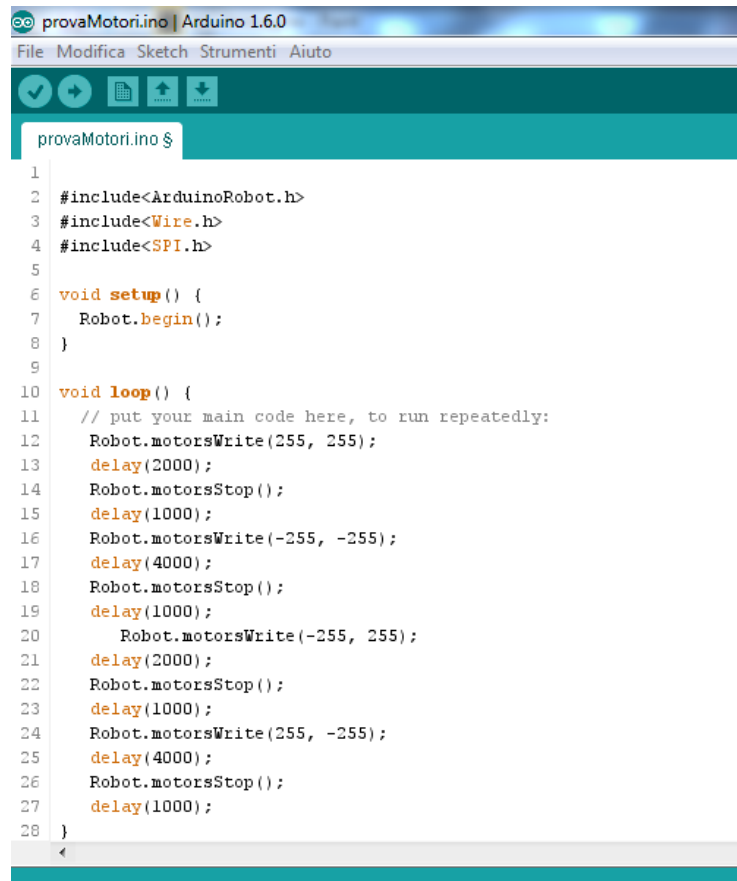


# LENGUAJES EN EDUCACIÓN

- Arduino:
  - Plataforma electrónica open-source basada en hardware y software sencilla de usar.
  - Es simplemente el lenguaje C/C++ con librerías añadidas.
  - Permite programación orientada a robots aunque normalmente se utilizan interfaces visuales, este es el caso de Scratch4Arduino y Mblock.

# LENGUAJES EN EDUCACIÓN

- Arduino:

A screenshot of the Arduino IDE interface. The title bar shows 'provaMotori.ino | Arduino 1.6.0'. The menu bar includes 'File', 'Modifica', 'Sketch', 'Strumenti', and 'Aiuto'. Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area shows the code for 'provaMotori.ino' with line numbers from 1 to 28. The code includes headers for 'ArduinoRobot.h', 'Wire.h', and 'SPI.h'. It defines a 'setup' function that calls 'Robot.begin()' and a 'loop' function that controls a robot's motors with various delays and motor writes.

```
1
2 #include<ArduinoRobot.h>
3 #include<Wire.h>
4 #include<SPI.h>
5
6 void setup() {
7   Robot.begin();
8 }
9
10 void loop() {
11   // put your main code here, to run repeatedly:
12   Robot.motorsWrite(255, 255);
13   delay(2000);
14   Robot.motorsStop();
15   delay(1000);
16   Robot.motorsWrite(-255, -255);
17   delay(4000);
18   Robot.motorsStop();
19   delay(1000);
20   Robot.motorsWrite(-255, 255);
21   delay(2000);
22   Robot.motorsStop();
23   delay(1000);
24   Robot.motorsWrite(255, -255);
25   delay(4000);
26   Robot.motorsStop();
27   delay(1000);
28 }
```

# LENGUAJES EN EDUCACIÓN

- Además de los ya mencionados también se utilizan otros lenguajes habituales como:
  - Matlab.
  - Python.
  - C/C++.
  - JavaScript.

# CONCLUSION

- Similitudes:
  - No existe un cambio en la variedad de tareas a realizar en cada uno de los lenguajes mencionados.
- Diferencias:
  - Eficiencia del código y simplicidad(en los lenguajes de empresa).



# BIBLIOGRAFÍA

- <https://es.wikipedia.org/wiki/Robot>
- <https:// analisisyprogramacionoop.blogspot.com.es/2014/10/programacion-de-robots.html>
- [http://platea.pntic.mec.es/vgonzale/cyr\\_0708/archivos/\\_15/Tema\\_5.6.htm](http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.6.htm)
- <https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics>
- [https://en.wikipedia.org/wiki/Educational\\_robotics](https://en.wikipedia.org/wiki/Educational_robotics)
- <http://petercorke.com/wordpress/toolboxes/robotics-toolbox>
- <https://www.makeblock.es/soporte/mblock/>
- <http://codigo21.educacion.navarra.es/autoaprendizaje/primeros-pasos-con-scratch-y-lego-wedo/>
- <http://www.onerobotics.com/posts/2013/introduction-to-karel-programming/>
- <http://www.monografias.com/trabajos3/progrob/progrob.shtml>
- <http://rosindustrial.org/>
- [https://www.roboticsbusinessreview.com/companies/?companyType=rbr\\_50](https://www.roboticsbusinessreview.com/companies/?companyType=rbr_50)
- <https://en.wikipedia.org/wiki/RAPID>
- <http://hopl.info/showlanguage.prx?exp=7482>
- [https://wikivisually.com/lang-de/wiki/ARLA\\_\(Programmiersprache\)](https://wikivisually.com/lang-de/wiki/ARLA_(Programmiersprache))
- <http://www.robot-forum.com/robotforum/fanuc-robot-forum/karel-load-option-in-r30ib/>
- [https://en.wikipedia.org/wiki/Adept\\_Technology](https://en.wikipedia.org/wiki/Adept_Technology)
- [https://es.wikipedia.org/wiki/Scratch\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Scratch_(lenguaje_de_programaci%C3%B3n))
- [https://wiki.scratch.mit.edu/wiki/Scratch#Notable\\_Information](https://wiki.scratch.mit.edu/wiki/Scratch#Notable_Information)
- [https://en.wikipedia.org/wiki/Educational\\_robotics](https://en.wikipedia.org/wiki/Educational_robotics)
- <https://www.arduino.cc/en/Guide/Introduction>