

# Guía de básica para el desarrollo de robots con Java LeJOS

Programa de enriquecimiento educativo 2009-2010. Área Científico-Tecnológica

Juan Antonio Breña Moral

Versión: 20100115

## Índice de contenido

Concepto de clase.....	1
Usando una clase en otra clase.....	2
La clase robot de ejemplo.....	2
Comandos NXJ.....	3
Fragmentos de código interesantes.....	4
Bucle de control .....	4
Sensor de Ultrasonidos.....	5
Sensor de Luz.....	5

## Concepto de clase

El concepto de una clase es la siguiente:

```
//Area de librerias
import lejos.nxt.*;

//Definicion de clase
public class HelloWorld{

    //Area donde se definen los componentes internos de una clase
    public static String message = "Hello World";

    //Programa principal
    public static void main (String[] args) {
        showMessage();
        Button.waitForPress();
    }

    //Lo que puede la clase
    private static void showMessage(){
        LCD.drawString(message, 0, 0);
        try {Thread.sleep(5000);} catch (Exception e) {}
    }
}
```

Viendo este ejemplo, deberias haber aprendido:

1. Una clase se compone de su definicion interna de sus componentes y de la definicion de sus acciones. En el ejemplo HelloWorld existe una variable que define el mensaje y un metodo que realiza una accion.

## ***Usando una clase en otra clase***

```
//Area de librerias
import lejos.nxt.*;

//Definicion de clase
public class GestorLCD{

    //Constructor
    public GestorLCD(){

    }

    //Area donde se definen los componentes internos de una clase
    private String message = "Hello World";

    //Lo que puede la clase
    public void showMessage(){
        LCD.drawString(message, 0, 0);
        try {Thread.sleep(5000);} catch (Exception e) {}
    }
}
```

Definamos una clase que usa la clase anterior:

```
//Definicion de clase
public class PruebaGestor{

    //La clase usa un objeto de tipo GestorLCD
    private static GestorLCD glcd;

    //Programa principal
    public static void main (String[] args) {
        glcd = new GestorLCD();
        glcd.showMessage();
    }
}
```

## ***La clase robot de ejemplo***

```

import lejos.nxt.*;

public class Robot1 {

    private Motor motorLeft;
    private Motor motorRight;

    public Robot1(){
        motorLeft = Motor.A;
        motorRight = Motor.B;
    }

    public void avanzar(int miliseconds){
        motorLeft.forward();
        motorRight.forward();
        try {Thread.sleep(miliseconds);} catch (Exception e) {}
    }

    public void parar(){
        motorLeft.stop();
        motorLeft.stop();
    }

    public void esperar(int miliseconds){

        try {Thread.sleep(miliseconds);} catch (Exception e) {}

    }
}

```

Usando la clase desde otro programa:

```

//Definicion de clase
public class PruebaRobot{

    //La clase usa un objeto de tipo GestorLCD
    private static Robot1 robot;

    //Programa principal
    public static void main (String[] args) {
        robot = new Robot1();
        robot.avanzar(1000);
    }

}

```

## Comandos NXJ

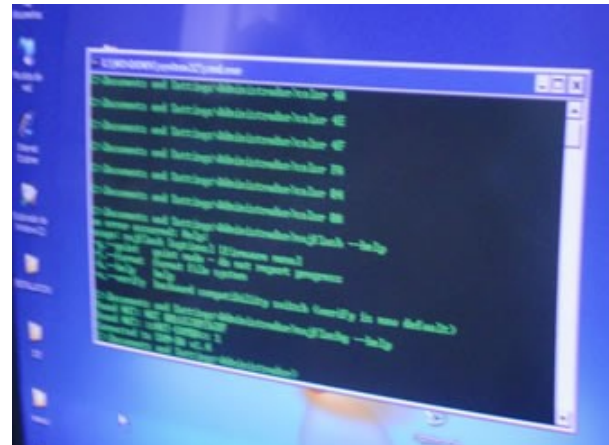
La metodología de desarrollo de los robots a través de Java LeJOS, siempre es de la misma manera:

1. Desarrollo
2. Compilación
3. Linkaje y distribución



Para conseguir ejecutar cada uno de los puntos anteriormente citados, es necesario abrir una ventana de comandos en tu sistema operativo favorito y ejecutar los siguientes comandos:

1. **nxjflashg**: comando que permite sustituir el sistema operativo del brick a través de una herramienta gráfica
2. **nxjc**: comando que permite compilar un programa. Ej: nxjc MiProgramaParaElRobot.java
3. **nxj**: comando que realiza el linkaje y distribución en tu robot de tu programa compilado. Ej: nxj MiProgramaParaElRobot (En este comando no es necesario poner la extensión .java)



## Fragmentos de código interesantes

### ***Bucle de control***

```
public static void main(String[] args){  
    while(!Button.ESCAPE.isPressed()){  
        //Poner código de tu programa  
    }  
}
```

## ***Sensor de Ultrasonidos***

```
private UltrasonicSensor US;  
US = new UltrasonicSensor(SensorPort.S4);  
US.getDistance();
```

## ***Sensor de Luz***

```
LightSensor light = new LightSensor(SensorPort.S1);  
light.setFloodlight(true);
```