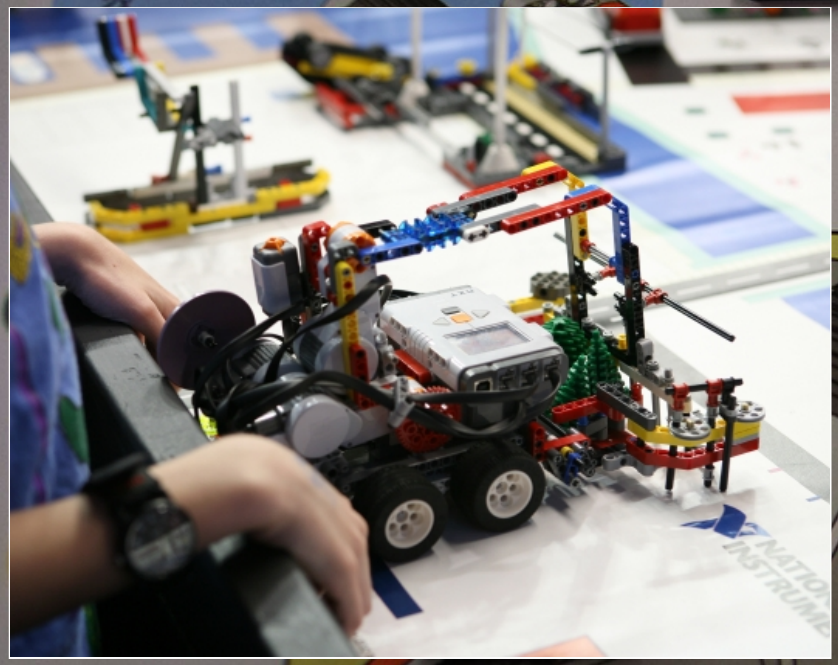


Seminario sobre Introducción a la Robótica.

Dr. José Maria Cañas Plaza
Juan Antonio Breña Moral

Universidad Rey Juan Carlos

ROBOTICA

Campus de Fuenlabrada
ETS. de Ingeniería de Telecomunicación

LEJOS
Java for LEGO Mindstorms

Índice

1. Introducción
2. Objetivos del seminario
3. Proyecto leJOS
4. Robocup Junior
5. Comandos en leJOS
6. Tu primera clase
7. Trabajando con Motores
8. Trabajando con Sensores
9. Proyectos con leJOS:
    - Robot evita obstáculos
    - Robot sigue líneas
10. Anexos
    - Construcción del robot Tribot

#1 Introducción

Lego Mindstorms NXT, es una plataforma educativa para el desarrollo y construcción de robots que interaccionen con el mundo real.

<u>En el ámbito de Robótica educativa, Lego Mindstorms es considerado uno de los estándares. Se emplea en los niveles educativos de Secundaria y Universidad a nivel mundial.</u>

NXT Brick

#1 Introducción

Lego Mindstorms NXT se puede emplear dentro del sistema educativo en múltiples niveles para iniciar al alumno en el mundo de la robótica a través de una plataforma robusta y amigable.

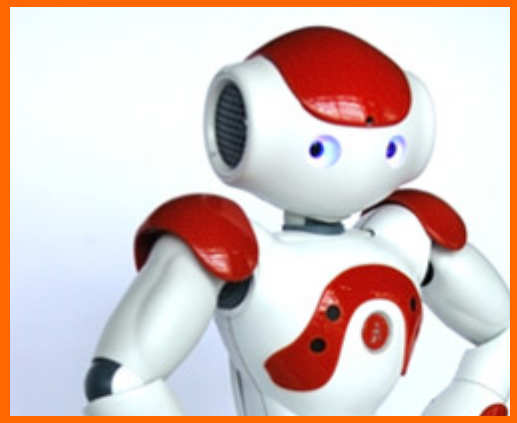
- Universidad (Ingenierías)
- Educación Secundaria

NXT Brick

#1 Introducción

Fase 1

Aprender
Facilidad



Fase 2

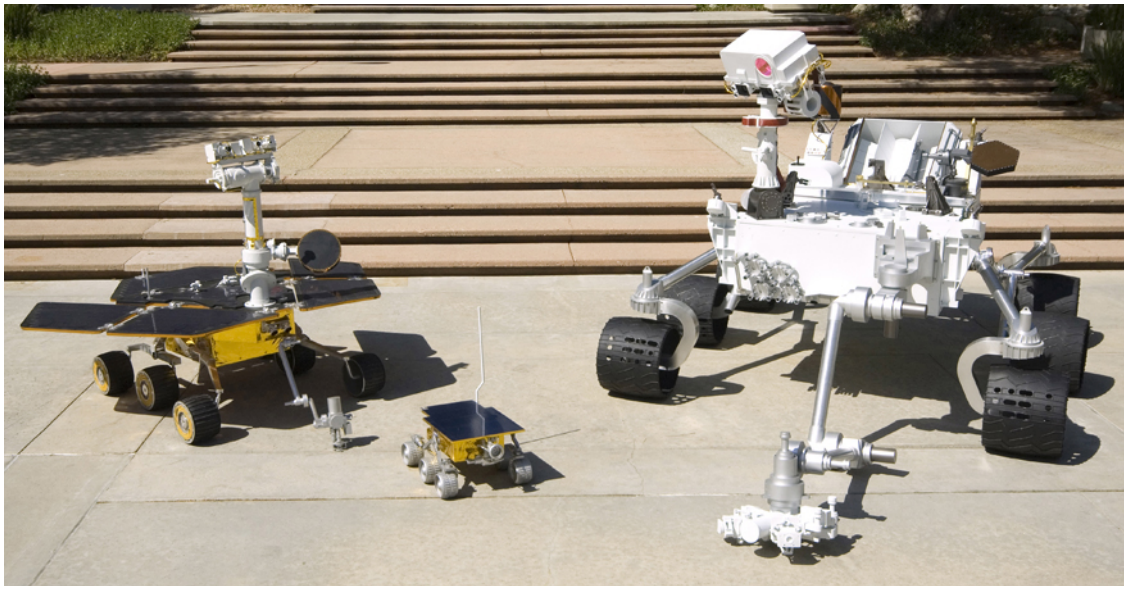
Linux RT
Visión Artificial

Fase 3

Aplicaciones
Productividad

#1 Introducción

Para que un alumno en un futuro pueda participar en proyectos punteros a nivel de robótica industrial, militar / aeroespacial, es necesario formarse en el sistema educativo español en los niveles iniciales.



NXT Brick

#2 Objetivos del seminario

Los objetivos del seminario son los siguientes:

1. Aprender a usar la plataforma educativa Lego Mindstorms NXT
2. Conocer el proyecto LeJOS
3. Aprender la metodología de desarrollo Java
4. Desarrollar robots sencillos con LeJOS

NXT Brick

#3. El proyecto leJOS

El proyecto leJOS tiene como proposito el desarrollo de una máquina virtual Java para los productos Lego Mindstorms NXT.

El proyecto leJOS originalmente se basa en un fork del proyecto TinyVM. LeJOS incluye una VM para Java bytecodes y permite cargar y correr software basado en la tecnología Java.

NXT Brick

LEJOS
Java for LEGO Mindstorms

#3. El proyecto leJOS

El proyecto leJOS se caracteriza por:

- **Usar un lenguaje orientado a objetos. (Java)**
- **Uso de Threads**
- **Capacidad de uso de Arrays simples y multidimensionales.**
- **Recursion**
- **Excepciones**
- **Uso de tipos de datos Java como: float, double, long y String**
- **Librerias matemáticas**
- **Una buena documentación del API de leJOS.**

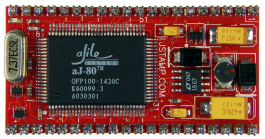
NXT Brick

LEJOS
Java for LEGO Mindstorms

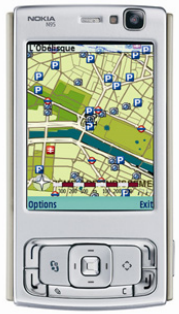
#3. El proyecto leJOS

Compativa de JVM actuales:

Pervasive Java



Embebbed Java



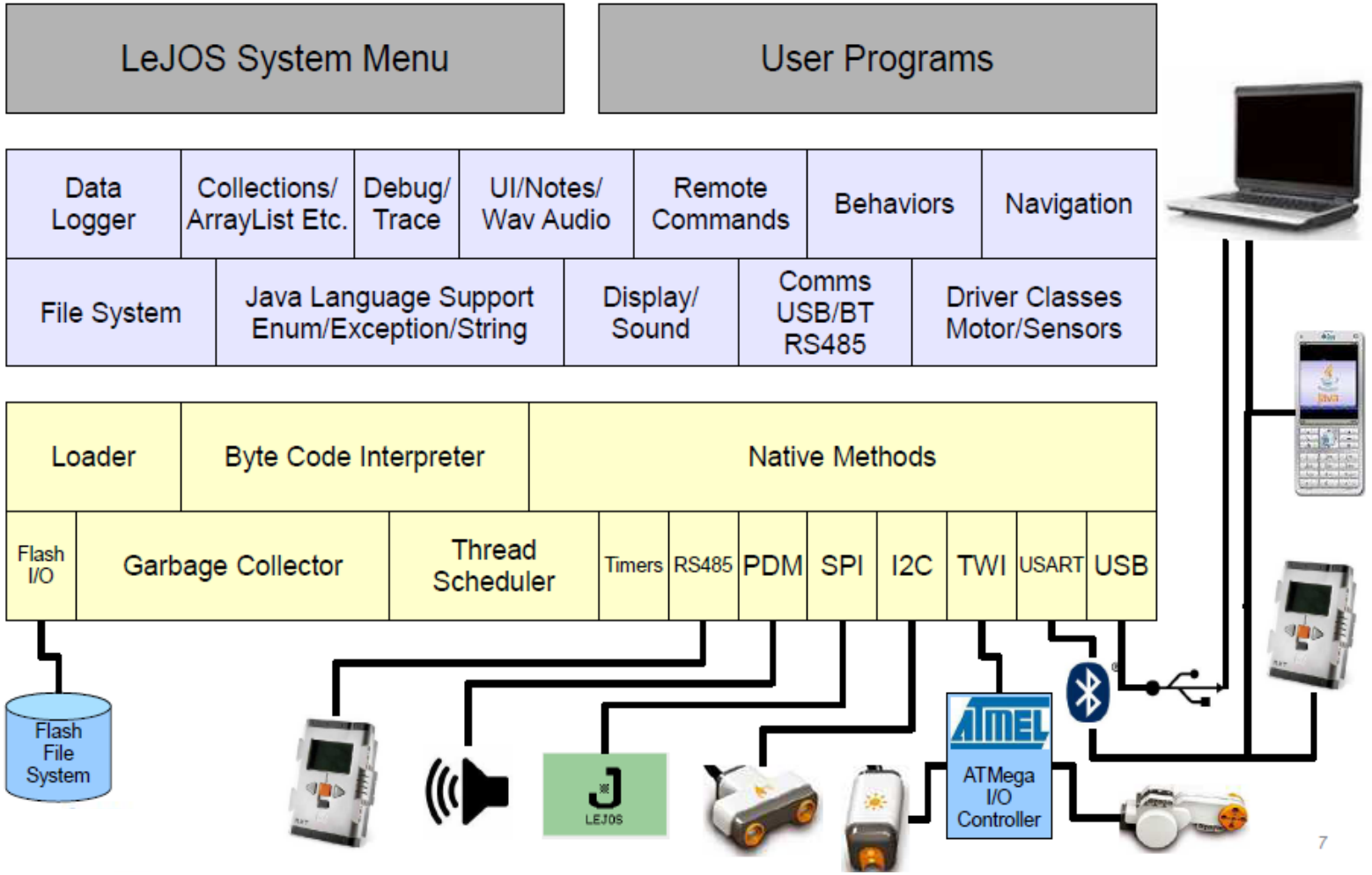
Mobile Java

Desktop Java

Server Java

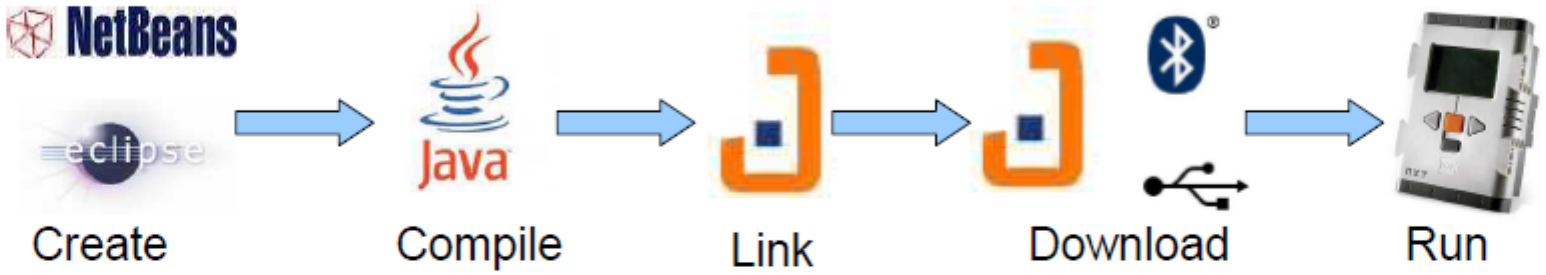
LEJOS
Java for LEGO Mindstorms

#3. El proyecto leJOS



#3. El proyecto leJOS

La metodología de desarrollo de cualquier proyecto leJOS es la siguiente:



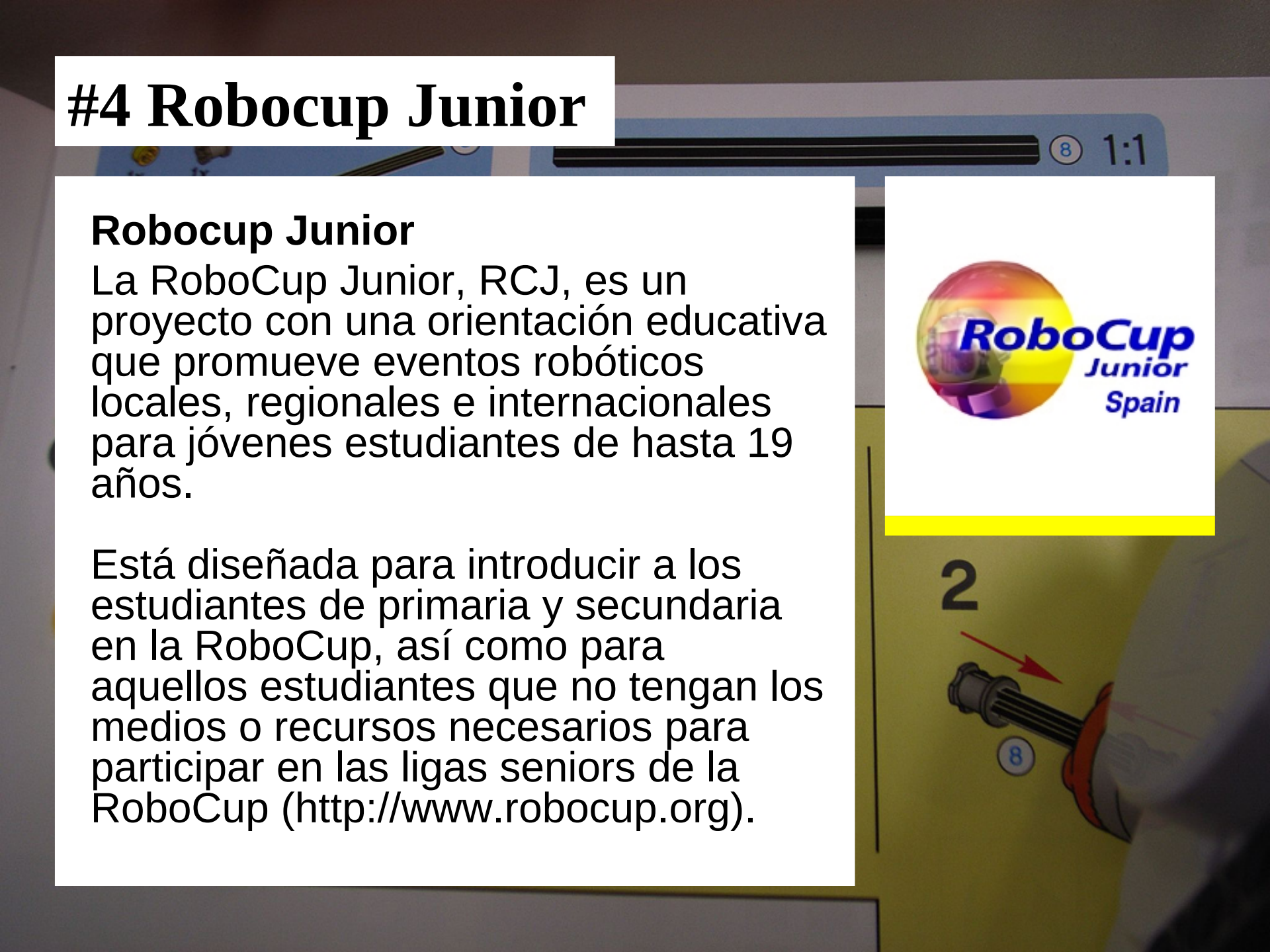
1. Crear y configurar un proyecto Java leJOS en tu IDE favorito.
2. Desarrollar el proyecto en 1 o mas paquetes
3. Compilar y descargar el proyecto en tu NXT
4. Probar proyecto en distintas situaciones

#4 Robocup Junior

Robocup Junior

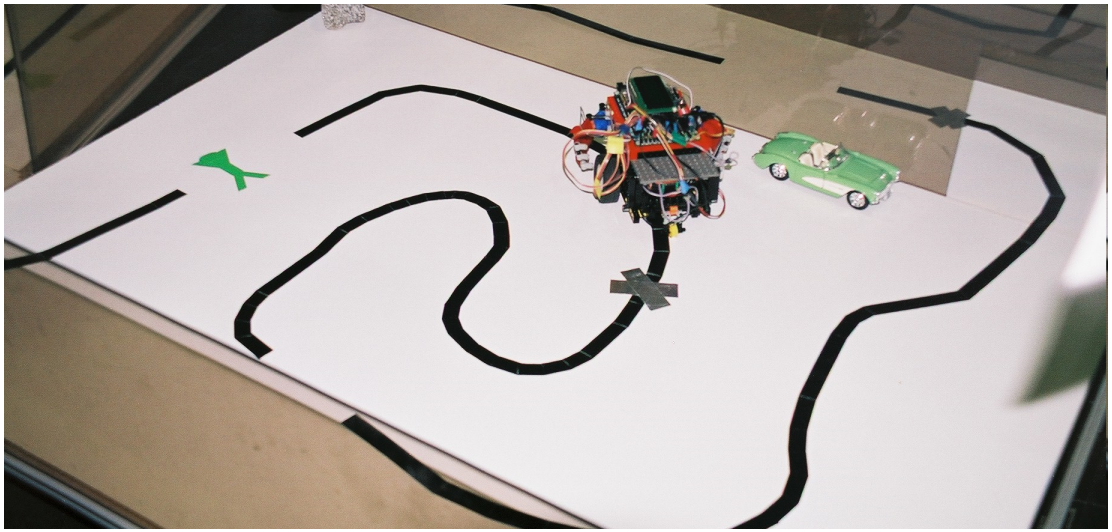
La RoboCup Junior, RCJ, es un proyecto con una orientación educativa que promueve eventos robóticos locales, regionales e internacionales para jóvenes estudiantes de hasta 19 años.

Está diseñada para introducir a los estudiantes de primaria y secundaria en la RoboCup, así como para aquellos estudiantes que no tengan los medios o recursos necesarios para participar en las ligas seniors de la RoboCup (http://www.robocup.org).



#4 Robocup Junior

En la prueba de Robocup Junior Rescue, los robots deben identificar víctimas en un escenario de catástrofe recreado. La complejidad varía entre el seguimiento de líneas hasta el sorteo de obstáculos en superficies con desniveles.



RoboCup Junior Spain

+

#5 Comandos leJOS

En leJOS, los comandos que son necesario aprender son los siguientes:

1. **nxjflashg**: Permite actualizar el firmware del NXT Brick.
2. **nxjc**: Permite compilar tu proyecto leJOS
3. **nxj**: Permite descargar tu proyecto compilado en tu NXT brick.

NXT Brick

#6 Tu primera clase

Una clase esta organizada en la siguientes partes:

- Importación de paquetes
  La clase usa otras clases
- Definición de variables
  La clase usa una serie de magnitudes
- Definición de métodos
  La clase realiza una serie de acciones
- Definición de método principal
  La clase ejecuta una cierta lógica.

NXT Brick

#6 Tu primera clase

Una clase esta organizada en la siguientes partes:

```
import lejos.nxt.*;
```

```
public class HelloWorld {


      public static void main
(String[] args) {


      }
}
```

NXT Brick

#6 Tu primera clase

Una clase esta organizada en la siguientes partes:

```
import lejos.nxt.*;

public class HelloWorld {
        private static String message =
"Hello World";

        public static void main
(String[] args) {

System.out.println(message);
                Button.waitForPress();
        }
}
```

NXT Brick

#6 Tu primera clase

Practica el concepto en tu equipo.

Abre el ejemplo HelloWorld.java y compilalo a traves del comando nxjc.

```
nxjc HelloWorld.java
```

Una vez que tengas compilado el proyecto, es necesario ejecutarlo en tu brick, para eso emplea el comando nxj

```
nxj –r HelloWorld
```

NXT Brick

#6 Tu primera clase

Variantes de la clase HelloWorld.

```
import lejos.nxt.*;

public class HelloWorld2 {

    public static void main (String[] args)
    {
        System.out.println("Hello World");
        Button.waitForPress();
    }

}
```

NXT Brick

#6 Tu primera clase

Variantes de la clase HelloWorld.

```
import lejos.nxt.*;

public class HelloWorld3 {

public static void main (String[] args) {
        while(!Button.ESCAPE.isPressed()){
        LCD.drawString("Hello World", 0, 0);
        }
}
}
```

NXT Brick

#6 Tu primera clase

Variantes de la clase HelloWorld.

```
import lejos.nxt.*;

public class HelloWorld4 {

    public static void main (String[] args)
    {
        LCD.drawString("Hello World", 0, 0);
        try {Thread.sleep(5000);} catch
    (Exception e) {}
    }
}
```

NXT Brick

#6 Tu primera clase

Variantes de la clase HelloWorld.

```
import lejos.nxt.*;

public class HelloWorld5 {

    public static void main (String[] args)
    {
        showMessage();
    }

    private static void showMessage(){
        LCD.drawString("Hello World", 0, 0);
        try {Thread.sleep(5000);} catch
    (Exception e) {}
    }
}
```

NXT Brick

#6 Tu primera clase

Practica el concepto en tu equipo.

Abre las diferentes variantes y prueba haber como reacciona tu robot.

```
nxjc HelloWorld2.java
nxjc HelloWorld3.java
nxjc HelloWorld4.java
nxjc HelloWorld5.java

nxj –r HelloWord2
nxj –r HelloWord3
nxj –r HelloWord4
nxj –r HelloWord5
```

¿Descubriste algo?

NXT Brick

#6 Tu primera clase

Conceptos aprendidos.

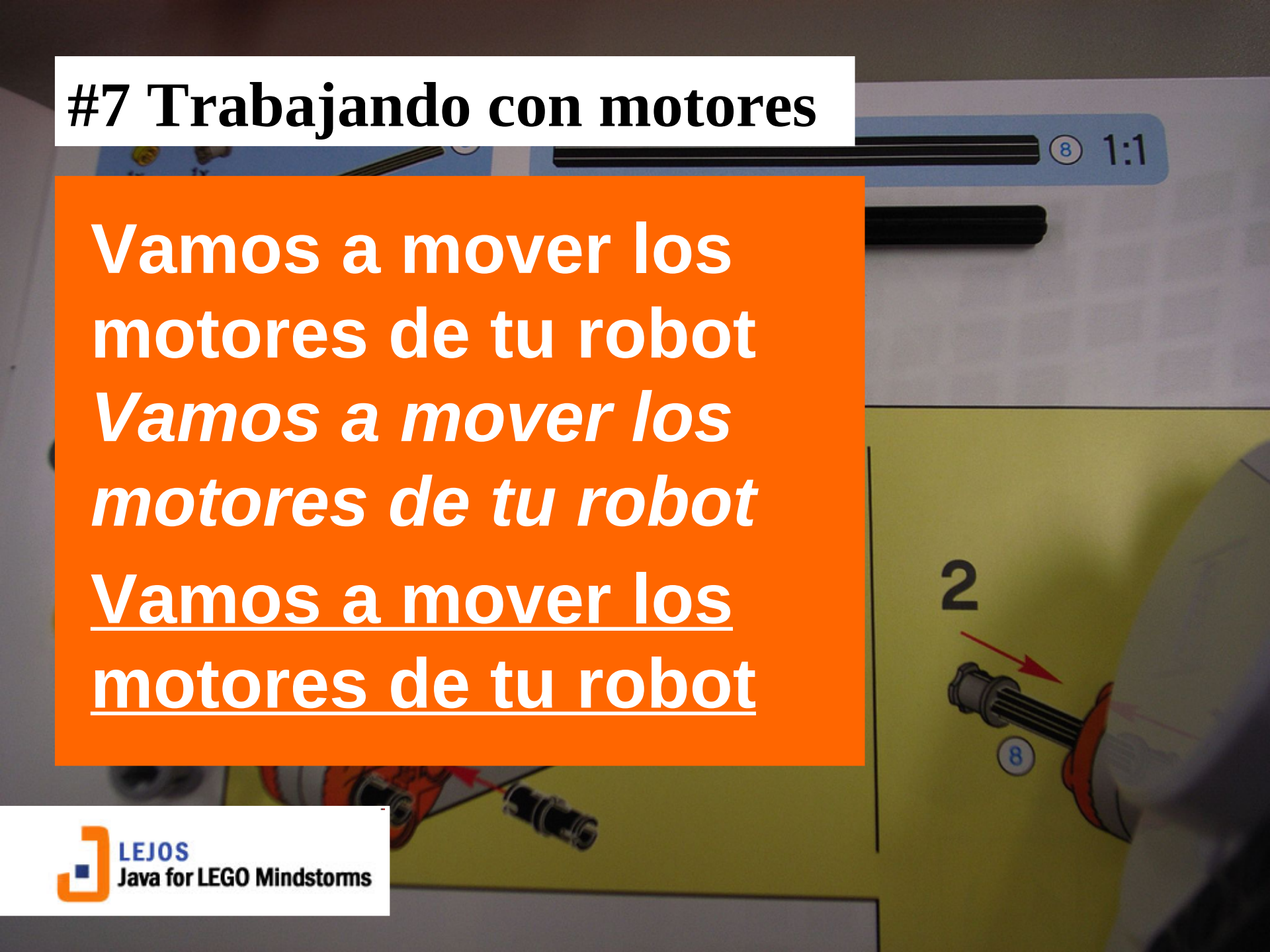
En estos primeros pasos, debereis haber aprendido:

1. Compilar un programa en leJOS
2. Ejecutar un programa leJOS
3. Conocer los elementos basicos de una clase en Java
4. Mostrar por pantalla un mensaje
5. Concepto de bucle de control

NXT Brick

#7 Trabajando con motores

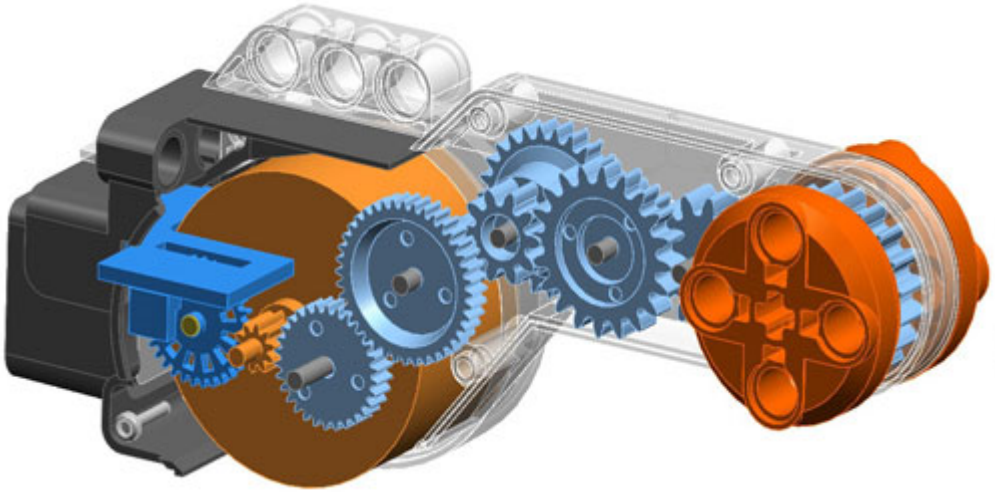
Vamos a mover los motores de tu robot
Vamos a mover los motores de tu robot
Vamos a mover los motores de tu robot



LEJOS
Java for LEGO Mindstorms

#7 Trabajando con motores

Tu NXT brick dispone de 3 servomotores por defecto.



NXT Brick

Para trabajar con ellos simplemente tienes que referirte a ellos de esta forma:

```
Motor motorLeft = new Motor(MotorPort.A)
```

#7 Trabajando con motores

Abramos el ejemplo Motor1.java para entender como se trabaja con motores:

```
import lejos.nxt.*;

public class Motor1 {
    public static void main(String[] args){
        Motor motorLeft = new Motor(MotorPort.A);
        Motor motorRight = new Motor(MotorPort.B);

        motorLeft.setSpeed(900);
        motorRight.setSpeed(900);

        motorLeft.rotate(360);
        try {Thread.sleep(500);} catch (Exception e) {}
        motorRight.rotate(360);
    }
}
```

#7 Trabajando con motores

```
import lejos.nxt.*;

public class Motor2 {
    private static Motor motorLeft;
    private static Motor motorRight;

    public static void main(String[] args){
        motorLeft = new Motor(MotorPort.A);
        motorRight = new Motor(MotorPort.B);
        motorLeft.setSpeed(900);
        motorRight.setSpeed(900);

        turnLeft90();
        try {Thread.sleep(500);} catch (Exception e) {}
    }

    private static void turnLeft90(){
        motorLeft.rotate(360);
    }
}
```

#7 Trabajando con motores

Practica el concepto en tu equipo.

Abre las diferentes variantes y prueba haber como reacciona tu robot.

```
nxjc Motor2.java
nxjc Motor3.java
nxjc Motor4.java
nxjc Motor5.java

nxj –r Motor2
nxj –r Motor2
nxj –r Motor2
nxj –r Motor2
```

¿Descubriste algo?

NXT Brick

#7 Trabajando con motores

Conceptos aprendidos.

En estos primeros pasos, debereis haber aprendido:

1. Usar motores en tus robots
2. Usar un sencillo navegador

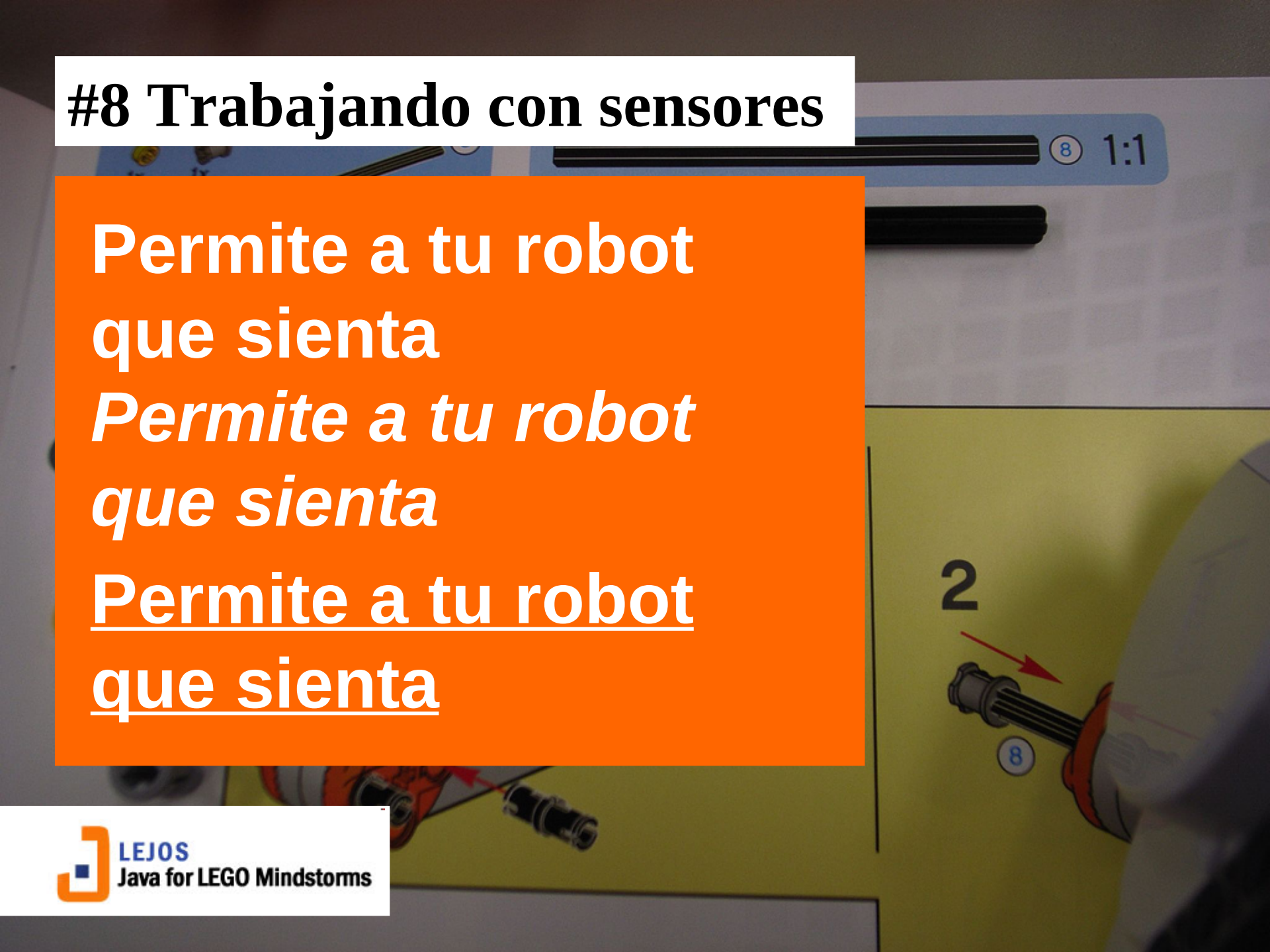
NXT Brick

#8 Trabajando con sensores

Permite a tu robot que sienta

Permite a tu robot que sienta

Permite a tu robot que sienta



LEJOS
Java for LEGO Mindstorms

#8 Trabajando con sensores

Tu robot NXT, puede manejar múltiples sensores. En este seminario aprenderemos a usar 3 sensores principalmente:

NXT Brick

#8 Trabajando con sensores

```
import lejos.nxt.*;

public class USSensor1 {
    private static UltrasonicSensor US;

    public static void main(String[] args){
        US = new UltrasonicSensor(SensorPort.S4);
        int distance = 0;
        Sound.setVolume(Sound.VOL_MAX);

        while(!Button.ESCAPE.isPressed()){
            distance = US.getDistance();

            Sound.playTone(distance+400, 100);
            LCD.drawString("   ", 0, 0);
            LCD.drawString("" + distance, 0, 0);
        }
    }
}
```

#8 Trabajando con sensores

```
import lejos.nxt.*;

public class TouchSensor1 {
    private static TouchSensor TS;

    public static void main(String[] args){
        TS = new TouchSensor(SensorPort.S3);
        Sound.setVolume(Sound.VOL_MAX);

        while(!Button.ESCAPE.isPressed()){
            if(TS.isPressed()){
                Sound.beep();
            }
        }
    }
}
```

#8 Trabajando con sensores

```
import lejos.nxt.*;import lejos.navigation.*;

public class TouchSensor2 {
    private static TouchSensor TS;
    private static SimpleNavigator SN;

    public static void main(String[] args){
        TS = new TouchSensor(SensorPort.S3);
        SN = new SimpleNavigator(5.5f,13.5f,Motor.A, Motor.B);
        SN.setSpeed(900);

        while(!TS.isPressed()){}
        Sound.beepSequenceUp();
        try {Thread.sleep(1000);} catch (Exception e) {}

        SN.forward();
        try {Thread.sleep(1000);} catch (Exception e) {}
    }
}
```

#8 Trabajando con sensores

```
import lejos.nxt.*;
public class LightSensor1 {
    private static LightSensor LS;
    private static final int minBlack = 27;
    private static final int maxBlack = 33;

    public static void main(String[] args){
        int colorDetected = 0;
        LS = new LightSensor(SensorPort.S1);
        LS.setFloodlight(true);

        while(!Button.ESCAPE.isPressed()){
            colorDetected = LS.readValue();
            LCD.drawString("   ", 0, 0);
            LCD.drawString("" + colorDetected, 0, 0);
            if( (minBlack <= colorDetected) &&
                (colorDetected <= maxBlack)){
                Sound.beep();
            }
        }}}
```

#8 Trabajando con sensores

Conceptos aprendidos.

En estos primeros pasos, debereis haber aprendido:

1. Usar sensores en tus proyectos
2. Hacer que tus robots tomen decisiones en funcion de los datos recogidos de los sensores.

NXT Brick

#9 Proyectos con leJOS

Robots autónomos en el aula.
Robots autónomos en el aula
Robots autónomos en el aula

LEJOS
Java for LEGO Mindstorms

#9 Proyectos con leJOS

Ejemplos previos:

Antes de desarrollar un robot desde 0, veremos algunos ejemplos previos

NXT Brick

#9 Proyectos con leJOS

```
import lejos.nxt.*;
import lejos.navigation.*;

public class Robot1 {
    private static UltrasonicSensor US;
    private static SimpleNavigator SN;

    static void main(String[] args) {
        US = new UltrasonicSensor(SensorPort.S4);
        SN = new SimpleNavigator(5.5f,13.5f,Motor.A, Motor.B);
        SN.setSpeed(900);

        while(!Button.ESCAPE.isPressed()){
            SN.forward();
            if(US.getDistance() <= 40){
                SN.rotateLeft();
            }
        }
    }
}
```

#9 Proyectos con leJOS

Robot evita obstáculos.

En este primer reto, desarrollaremos un robot el cual ha sido diseñado con la siguiente arquitectura:

1. Actuadores
    1. Motor A
    2. Motor B
2. Sensores
    1. Sensor de tacto
    2. Sensor de ultrasonidos

NXT Brick

#9 Proyectos con leJOS

Robot evita obstáculos.

Mision del robot:
El robot navegara por un espacio no conocido. En caso de descubrir obstaculos frontales con el sensor de ultrasonidos, retrocedera y realizara un giro.

NXT Brick

#9 Proyectos con leJOS

Robot evita obstáculos.

Mision del robot:
El robot navegara por un espacio no conocido. En caso de descubrir obstaculos frontales con el sensor de ultrasonidos, retrocedera y realizara un giro.

NXT Brick

#9 Proyectos con leJOS

Robot evita obstáculos.

Seudocodigo:

```
Mientras(apagarRobot){
        avanzar;
        si(detectarObstaculo){
                hacerMovimientoEvasion
        }
}
```

NXT Brick

#9 Proyectos con leJOS

```
import ?

public class ChocaYGira{
        private static ?

public static void main (String[] args){
        ¿

    while(!Button.ESCAPE.isPressed()){

        if ((distance <= 40) || (TS.isPressed())){
            ?
        }
    }
}
```

#9 Proyectos con leJOS

Robot evita obstáculos.

Fragmentos de codigo:

```
private static int getRandomAngle(){
    double rnd = 0.0d;
    rnd = java.lang.Math.random();
    return (int)(rnd*(315-45))+45;
}


if (angulo>180){
    angulo = 360 - angulo;
    pilot.rotate(angulo-(2*angulo));
}else{
    pilot.rotate(angulo);
}
```

NXT Brick

#9 Proyectos con leJOS

Robot sigue lineas.

En este segundo reto, el robot sera capaz de seguir un conjunto de lineas, para ello, el robot estara compuesto de:

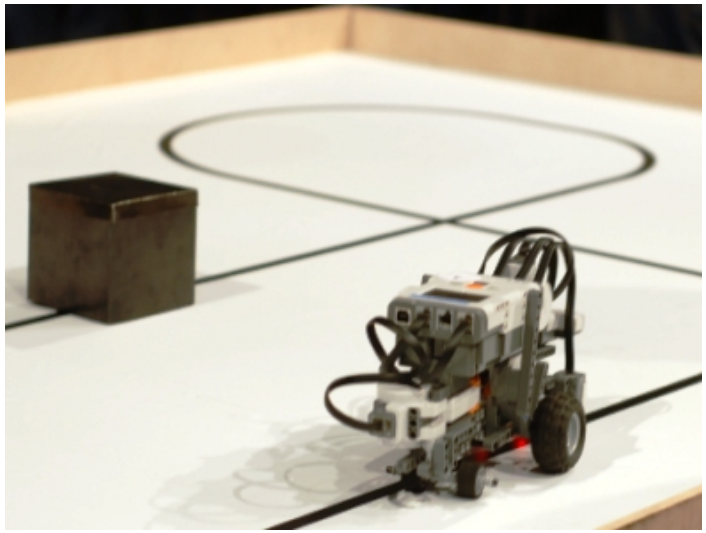
1. Actuadores
    1. Motor A
    2. Motor B
2. Sensores
    1. Sensor luz

NXT Brick

#9 Proyectos con leJOS

Robot sigue lineas.

Mision del robot:
El robot tiene como mision recorrer un circuito. La caracteristica del circuito es que se compone de una franga negra sobre un fondo blanco.



NXT Brick

#9 Proyectos con leJOS

```
public class LineFollower {

    public static void main (String[] aArg){
        String left = "Left ";
        String right= "Right";

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

        final int blackWhiteThreshold = 45;
        final int forward = 1;
        final int stop = 3;
        final int power = 80;

        LCD.drawString("Light %: ", 0, 0);

        // Show light percent until LEFT is pressed
        while (! Button.LEFT.isPressed()){
            LCD.drawInt(light.readValue(), 3, 9, 0);
            LCD.refresh();
        }
```

#9 Proyectos con leJOS

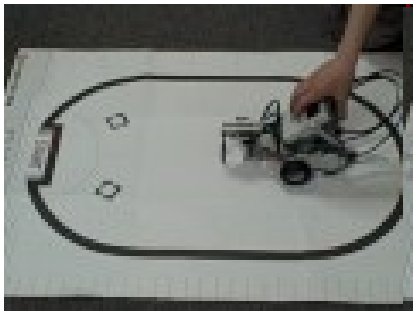
```
        // Follow line until ESCAPE is pressed
        while (! Button.ESCAPE.isPressed()){

            if (light.readValue() > blackWhiteThreshold){
                // On white, turn right
                LCD.drawString(right, 0, 1);
                MotorPort.A.controlMotor(0,stop);
                MotorPort.B.controlMotor(power, forward);
            }else{
                // On black, turn left
                LCD.drawString(left, 0, 1);
                MotorPort.A.controlMotor(power, forward);
                MotorPort.B.controlMotor(0,stop);
            }
                LCD.drawInt(light.readValue(), 3, 9, 0);
                LCD.refresh();
                try {Thread.sleep(100);} catch (Exception e) {}
            }
        }
}
```

#9 Proyectos con leJOS

Robot sigue lineas.

<u>Conclusiones de este enfoque:</u>
Este enfoque esta bien cuando el circuito es conocido y tus curvas siempre van en un mismo sentido, en este caso a la derecha.



NXT Brick

#9 Proyectos con leJOS

Robot sigue líneas.

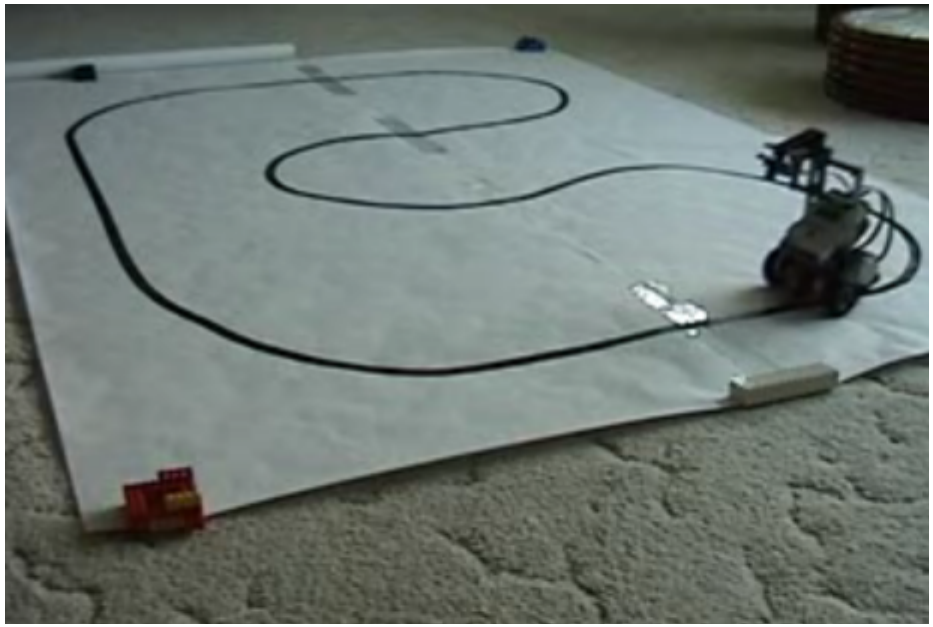
Pero en la vida real, los circuitos son aleatorios y con curvas de diferente naturaleza.

NXT Brick

#9 Proyectos con leJOS

Robot sigue líneas.

Los enfoques actuales, permiten usar cámaras para seguir líneas.



http://www.youtube.com/watch?v=o6U7Xtvkj_A

NXT Brick

#9 Proyectos con leJOS

Conceptos aprendidos.

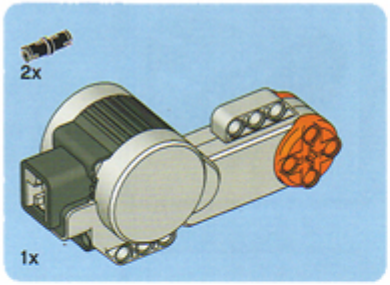
En estos primeros pasos, deberéis haber aprendido:

1. A desarrollar sencillos robots autónomos.
2. Cuanto mas procesamiento, mas fácil es ser preciso en la toma de decisiones
3. La visión artificial es un campo de enorme crecimiento. El ojo humano es el principal sensor de los humanos.

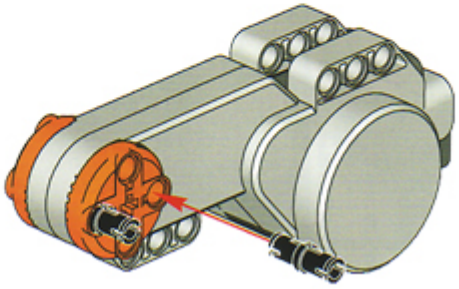
NXT Brick

#11 Anexos

Construcción del robot Tribot



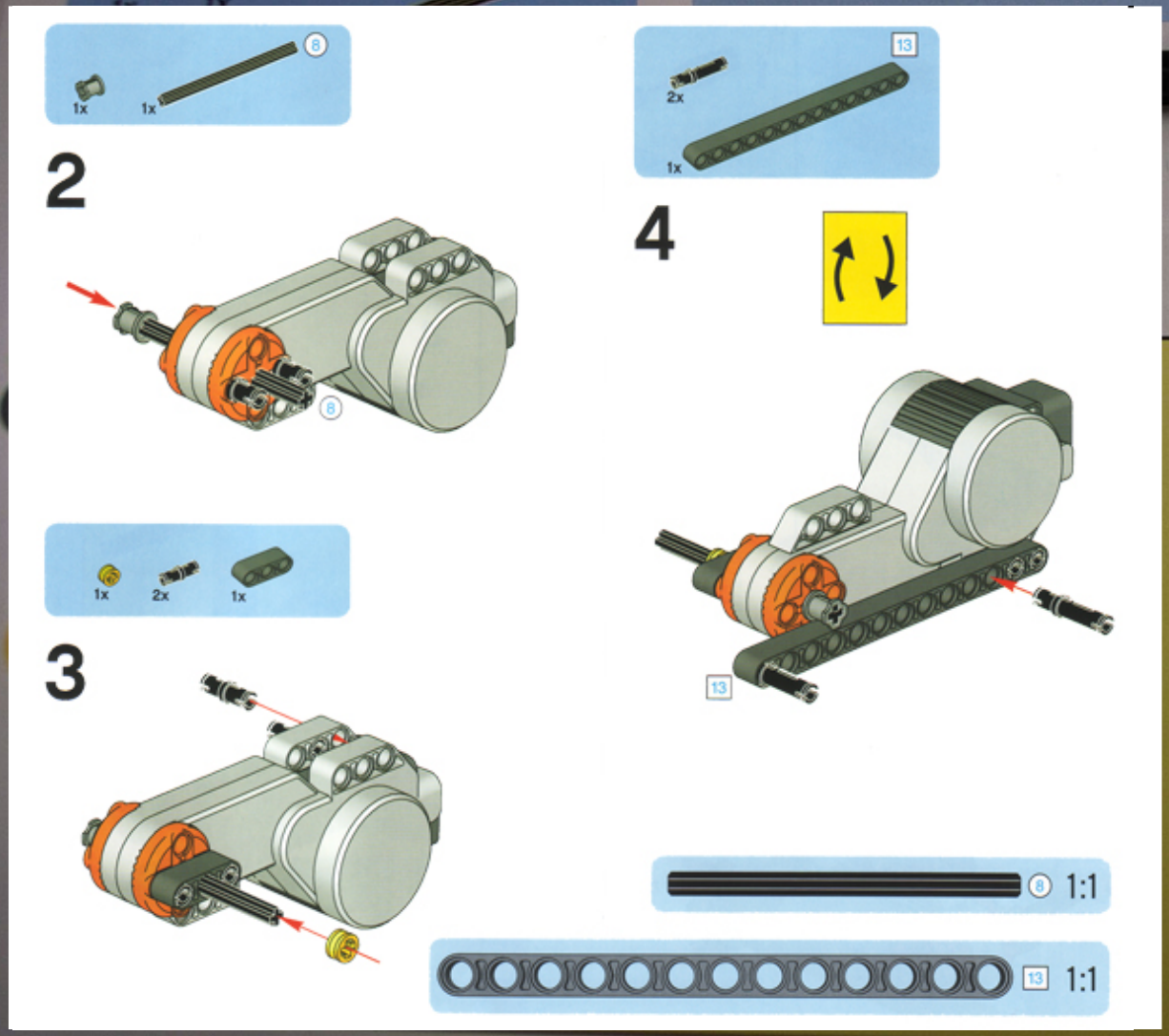
1

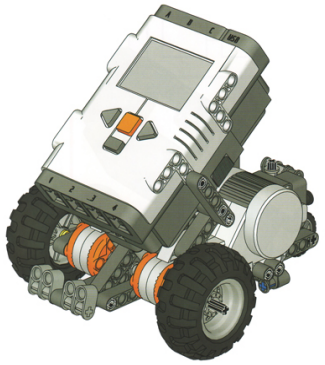


Robot Tribot

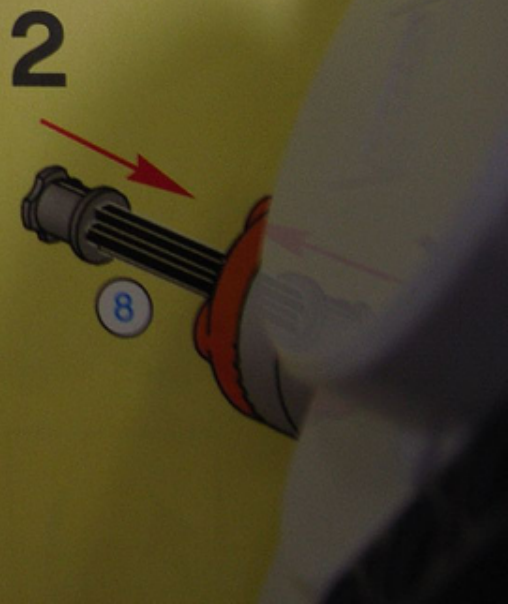
2

#11 Anexos

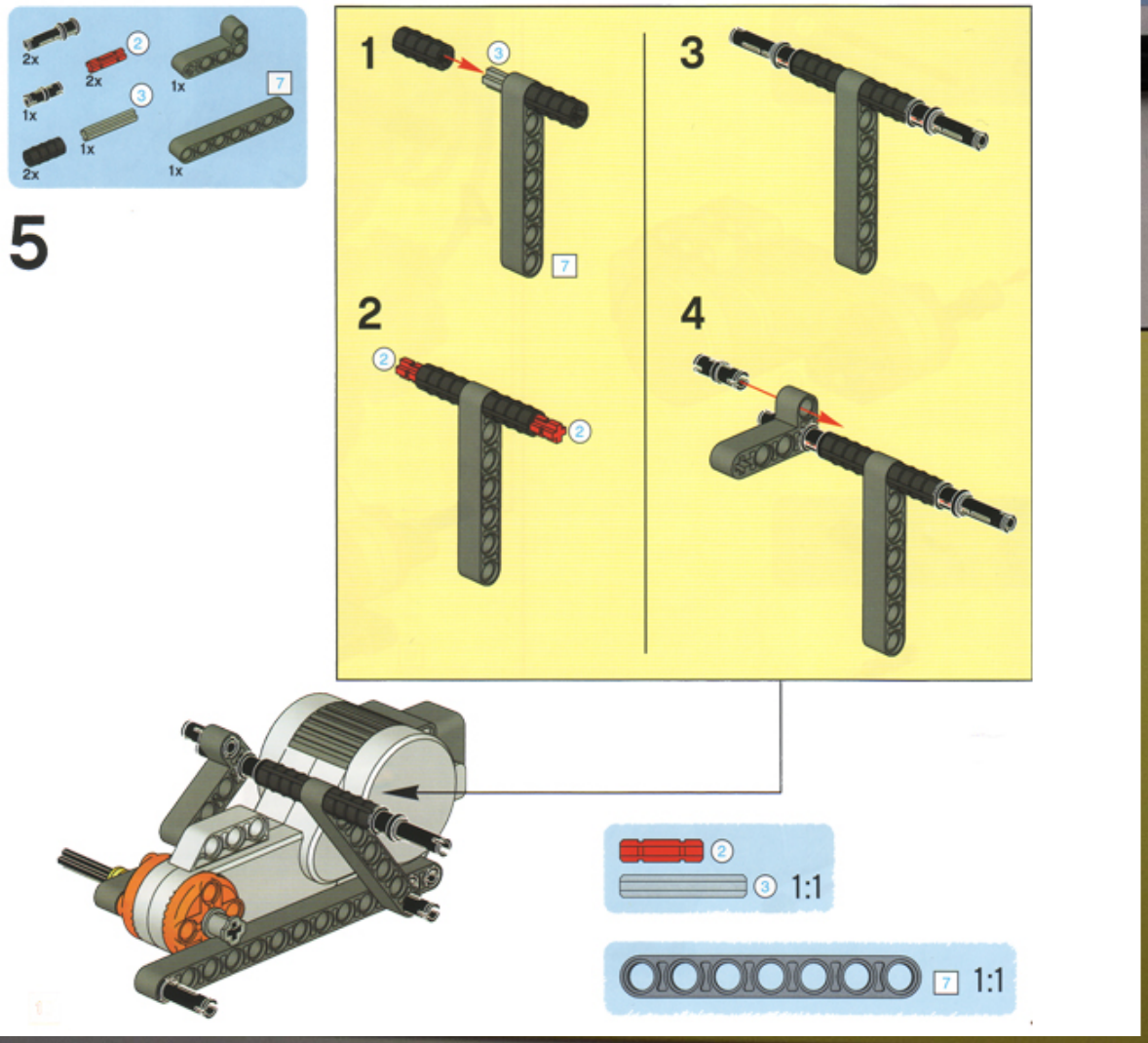


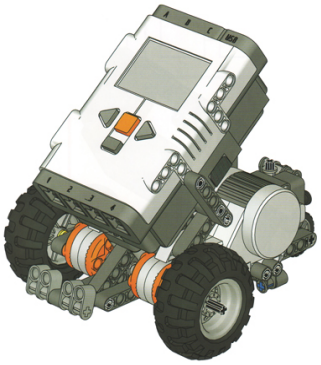


Robot Tribot

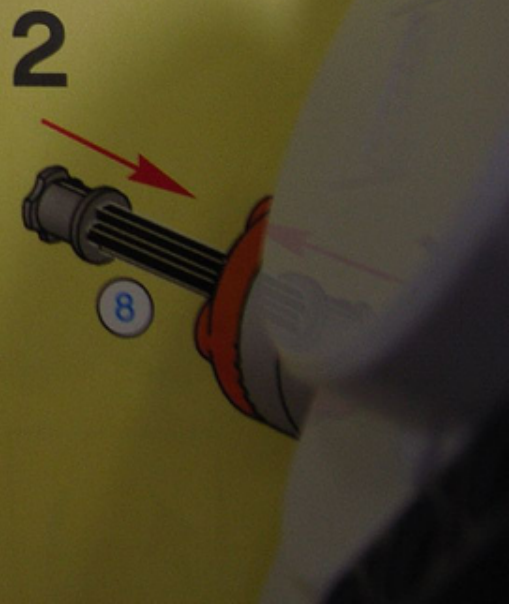


#11 Anexos

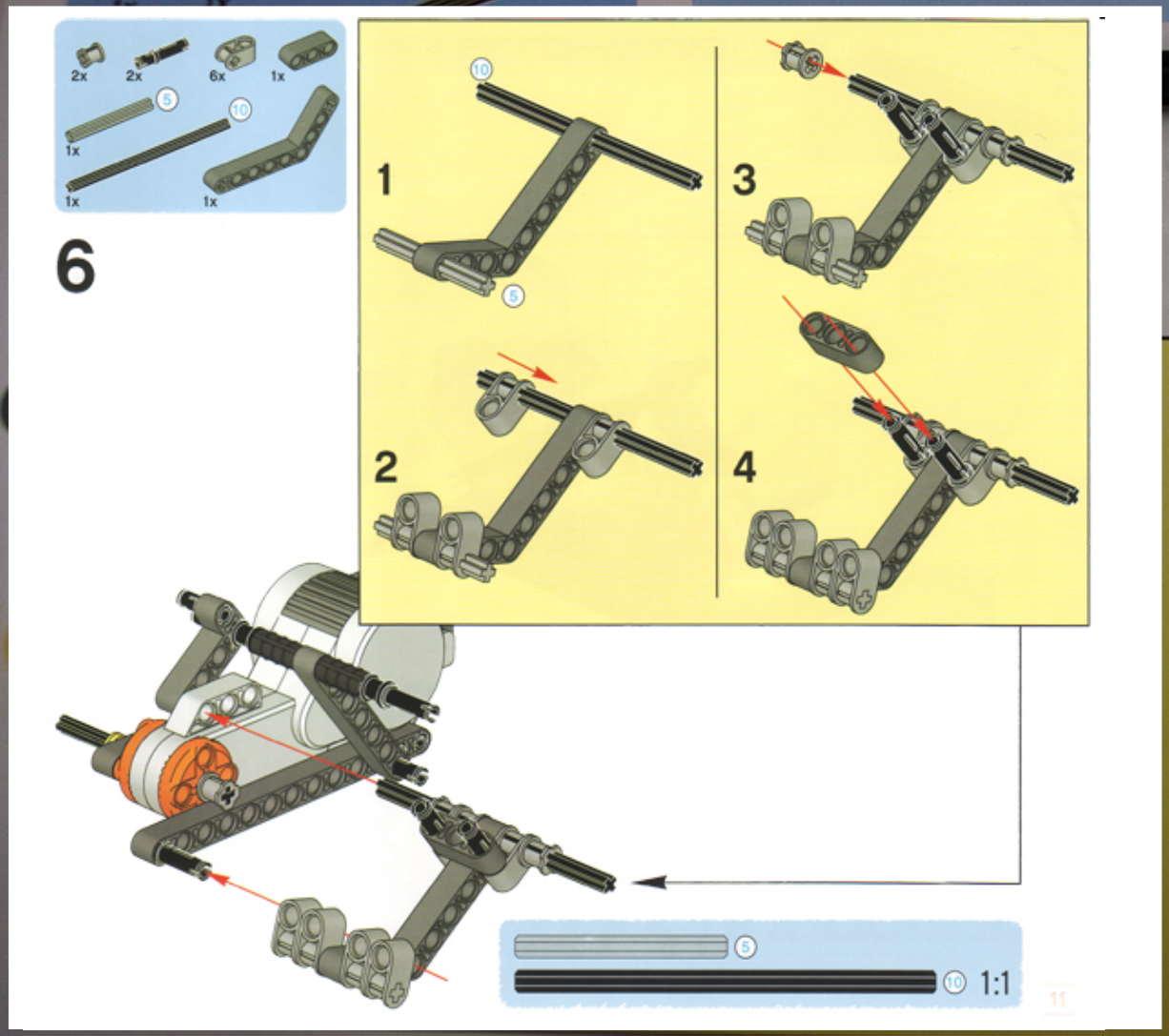


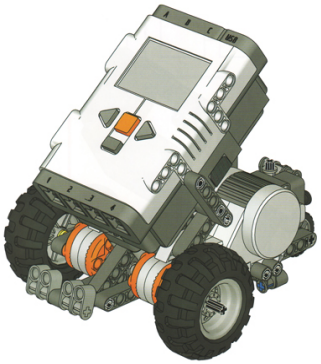


Robot Tribot

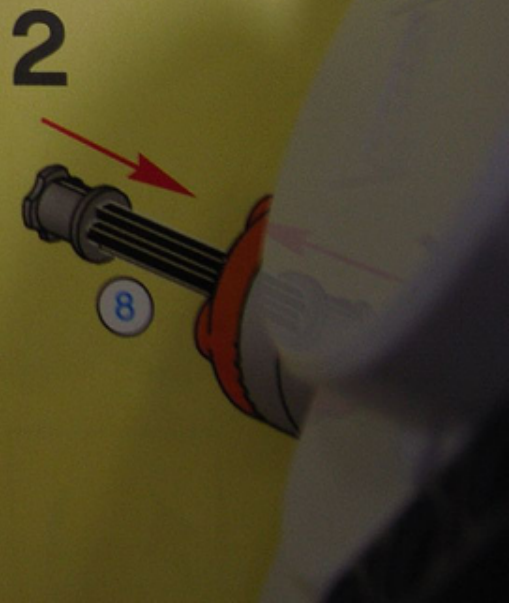


#11 Anexos

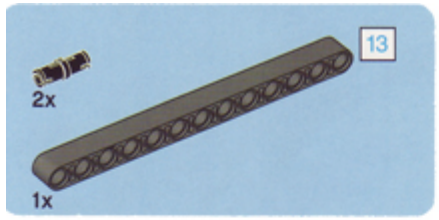




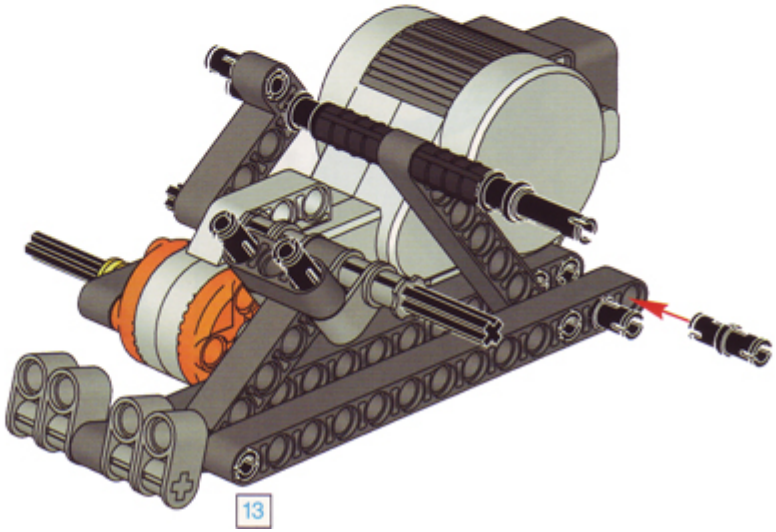
Robot Tribot

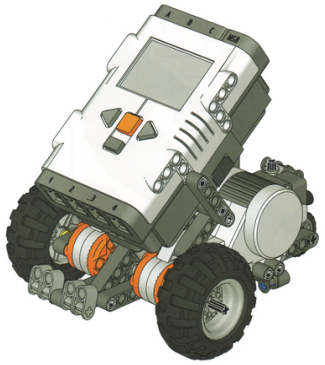


#11 Anexos



7

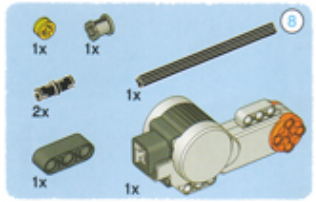


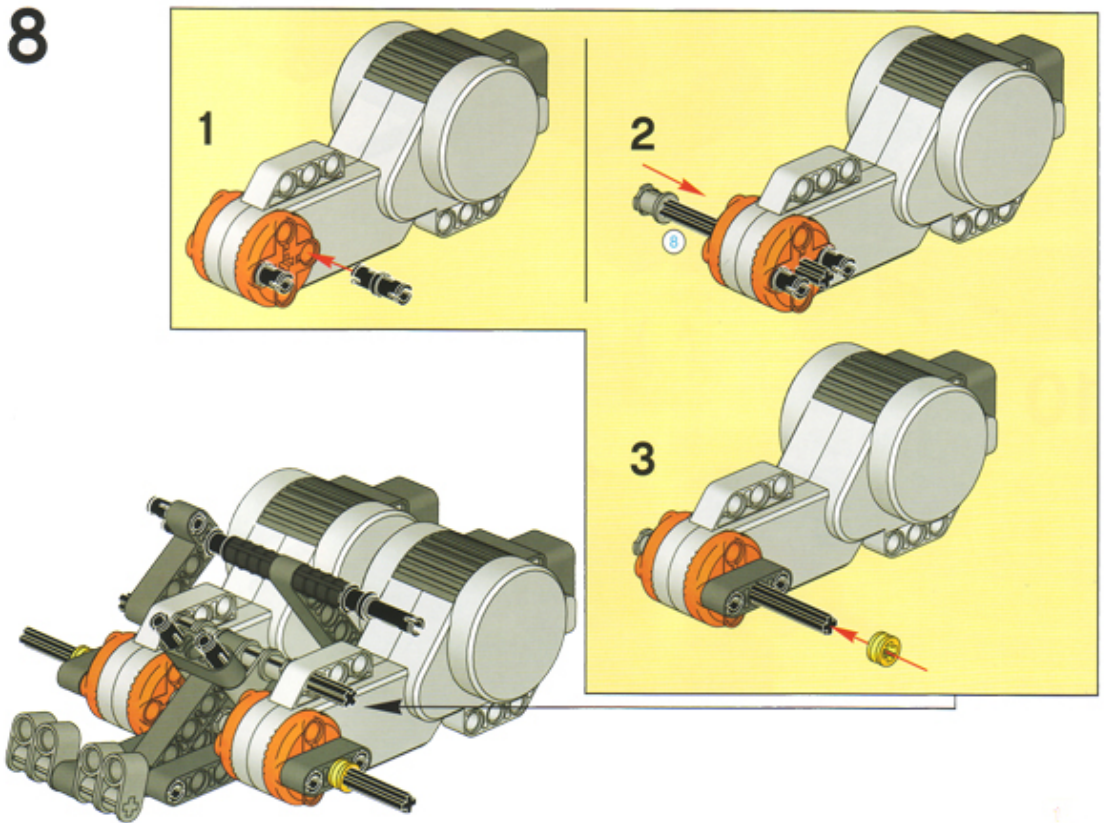


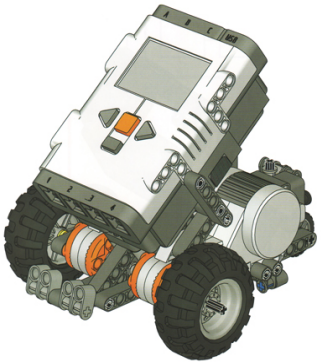
Robot Tribot

2

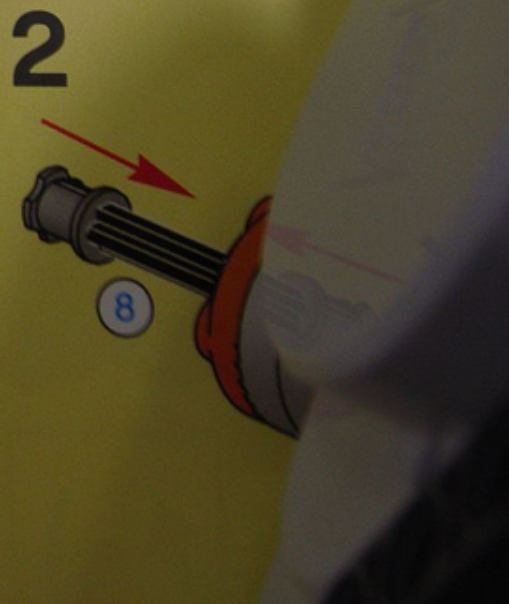
#11 Anexos



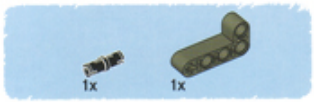




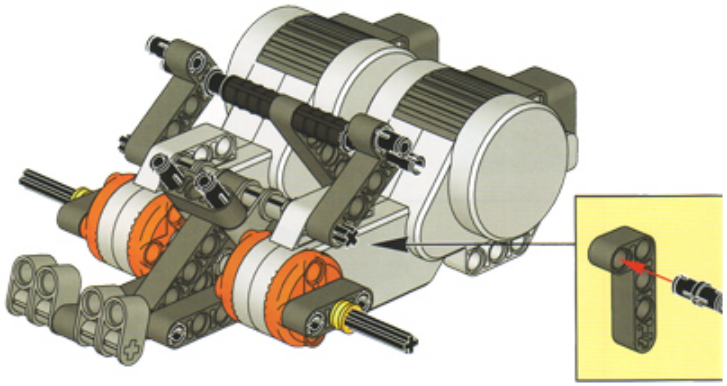
Robot Tribot

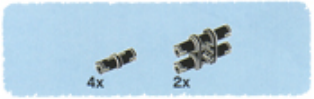


#11 Anexos

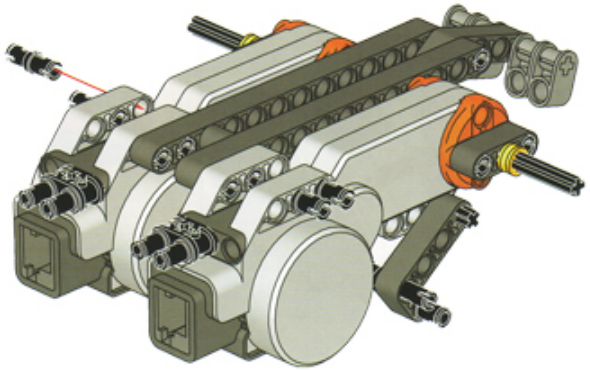


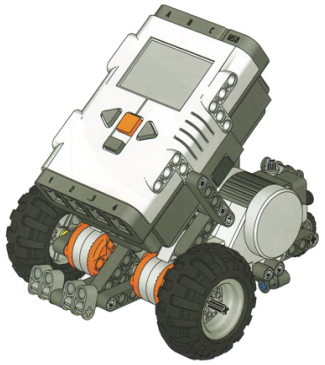
9





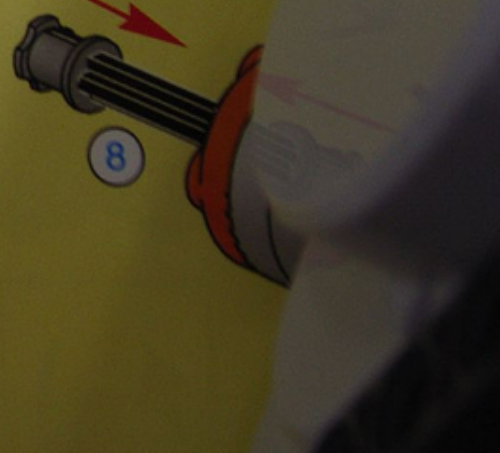
10



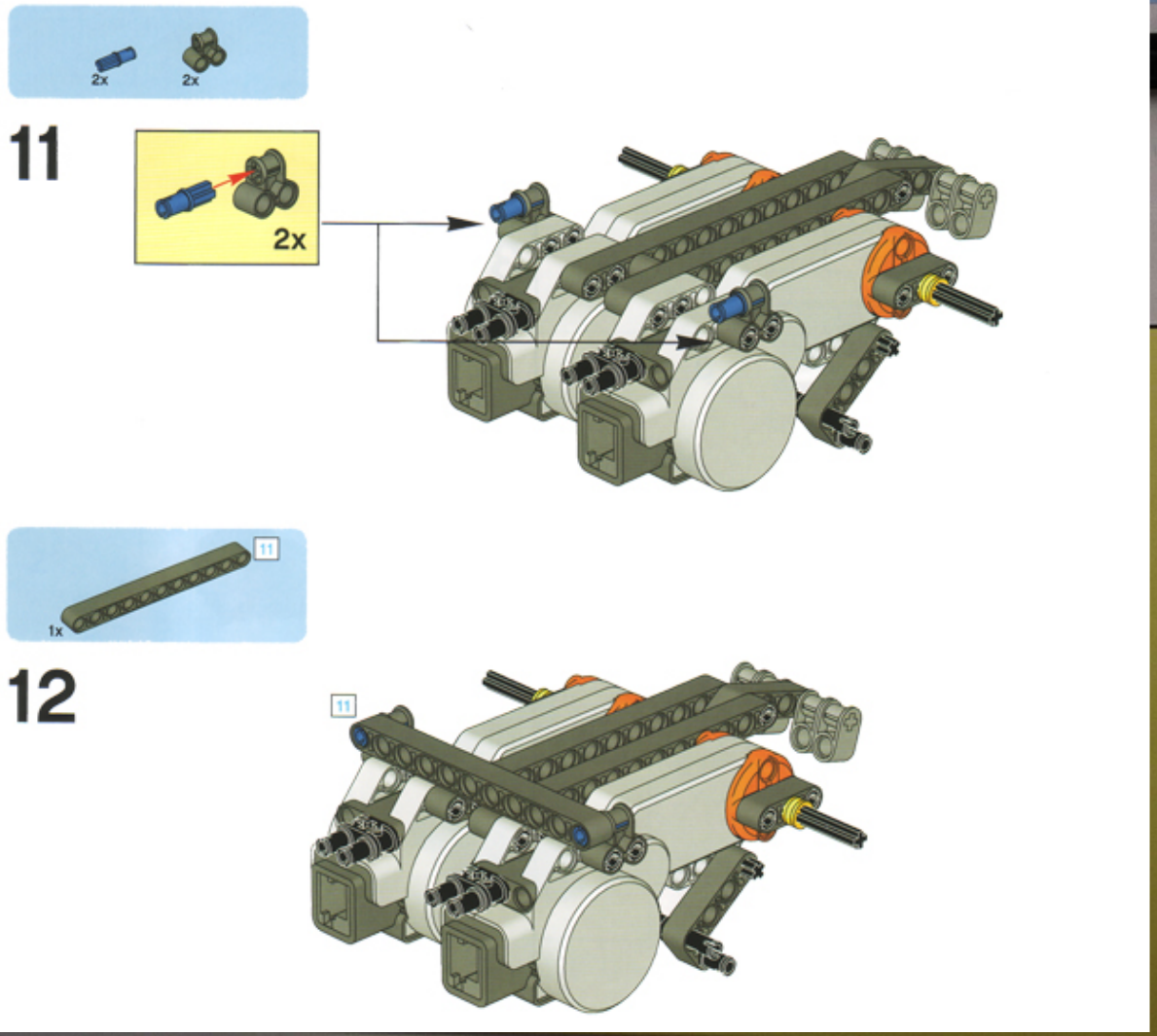


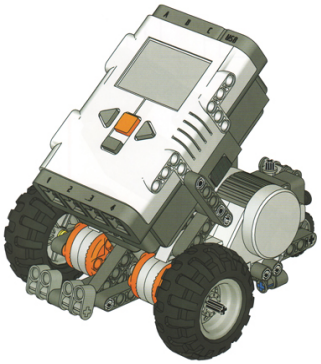
Robot Tribot

2

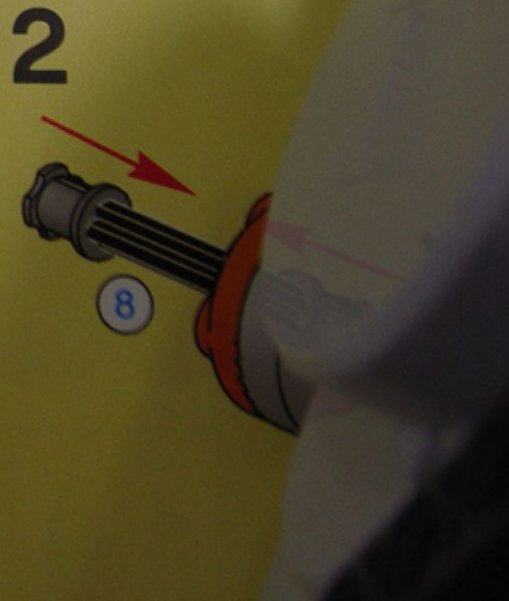


#11 Anexos

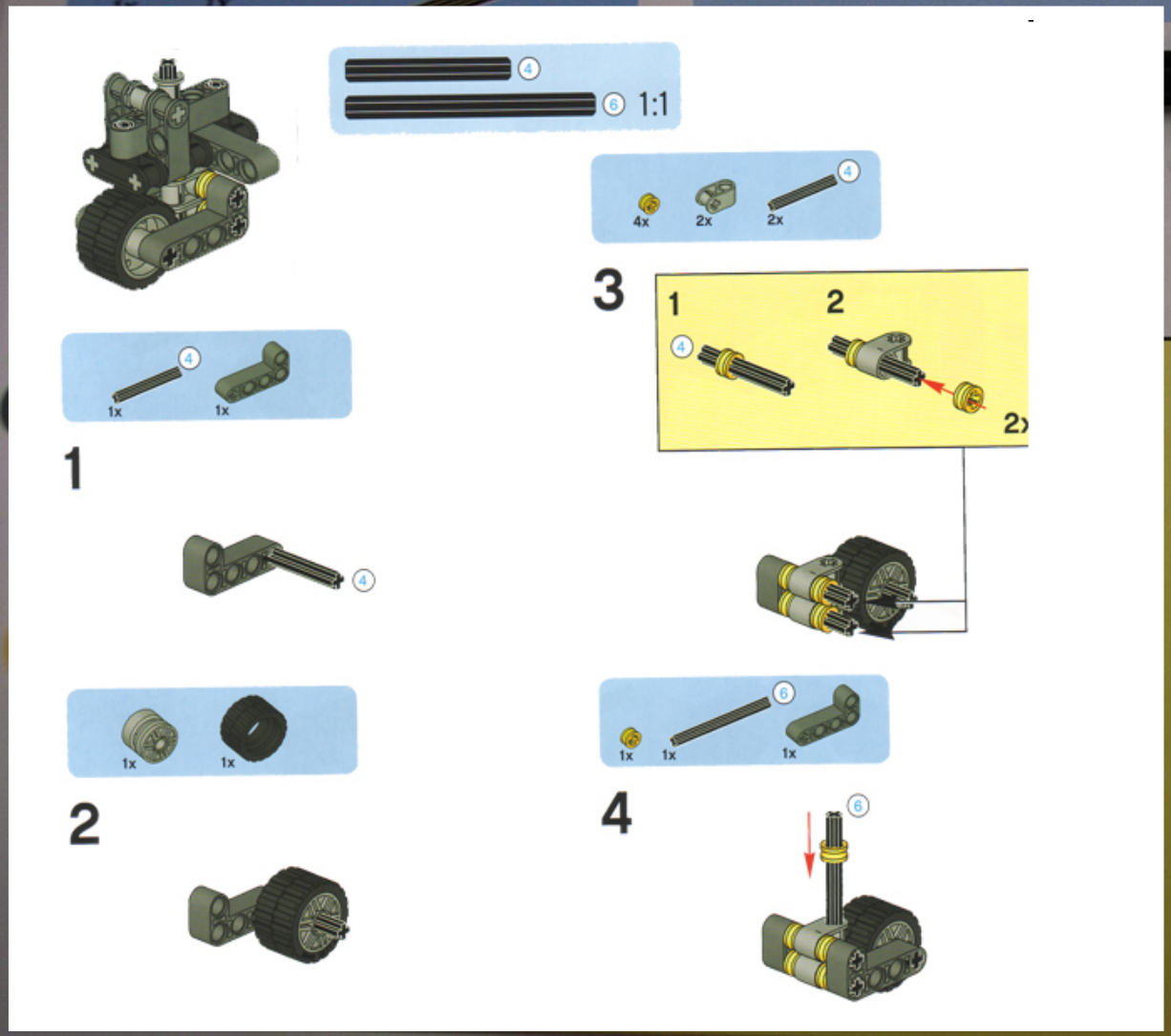


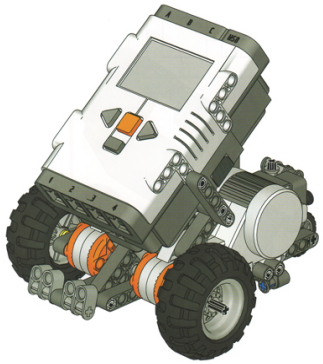


Robot Tribot

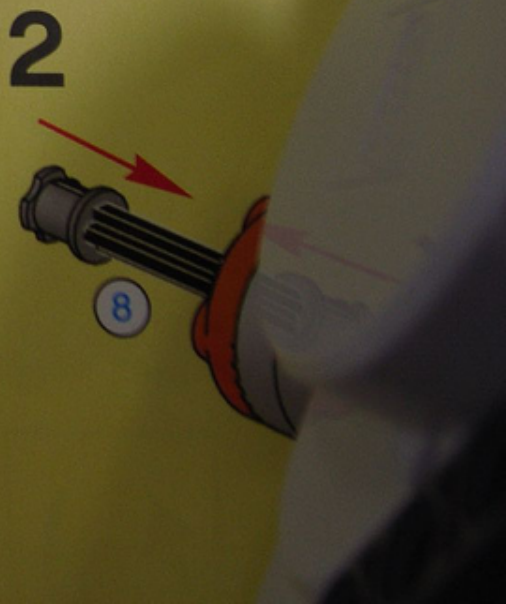


#11 Anexos

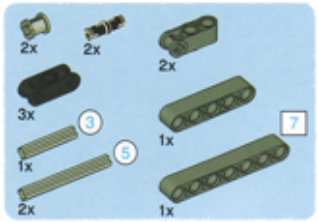




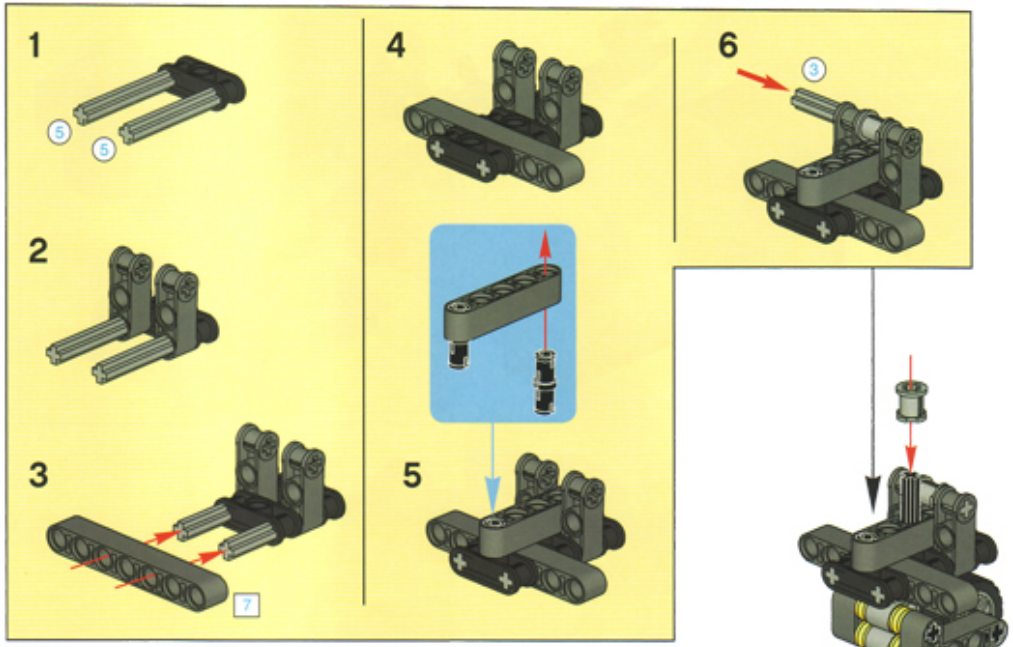
Robot Tribot

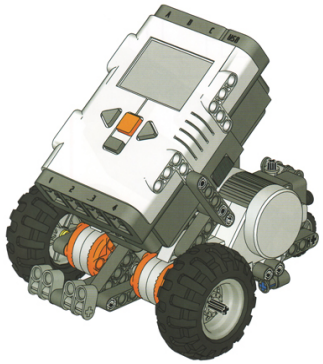


#11 Anexos

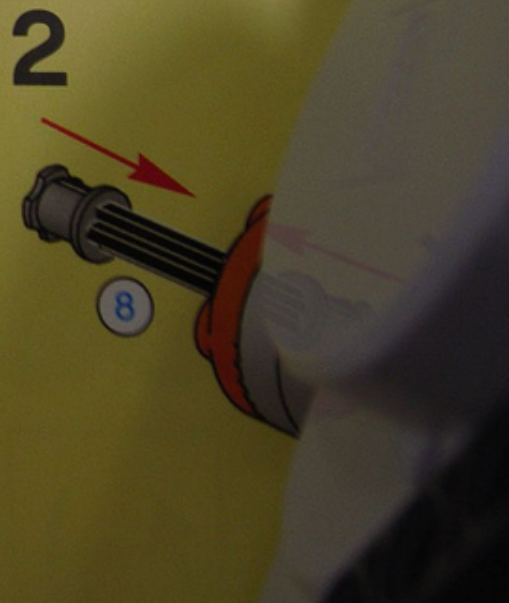


5



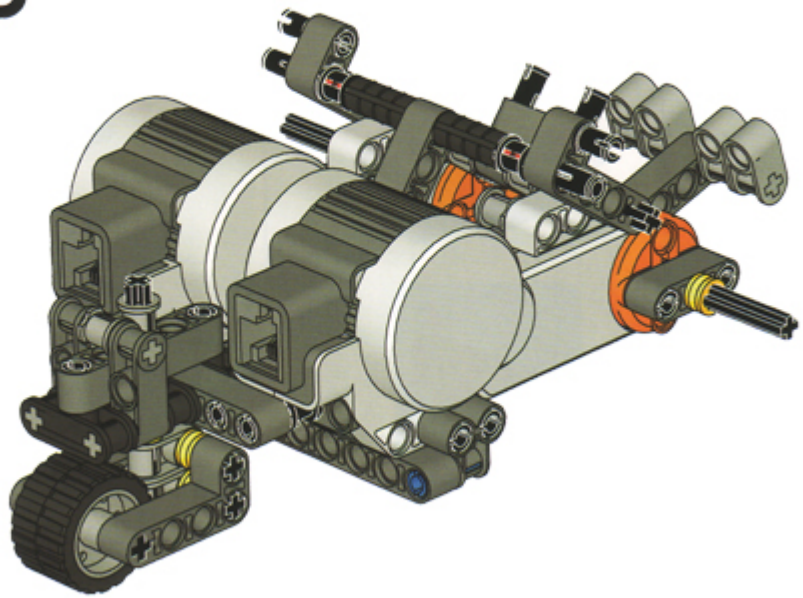


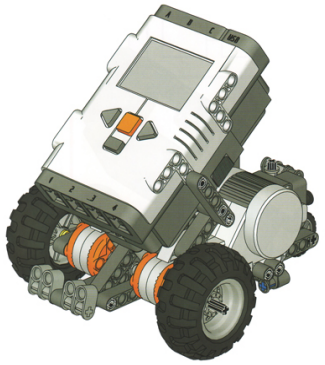
Robot Tribot



#11 Anexos

13

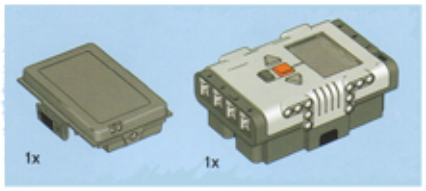




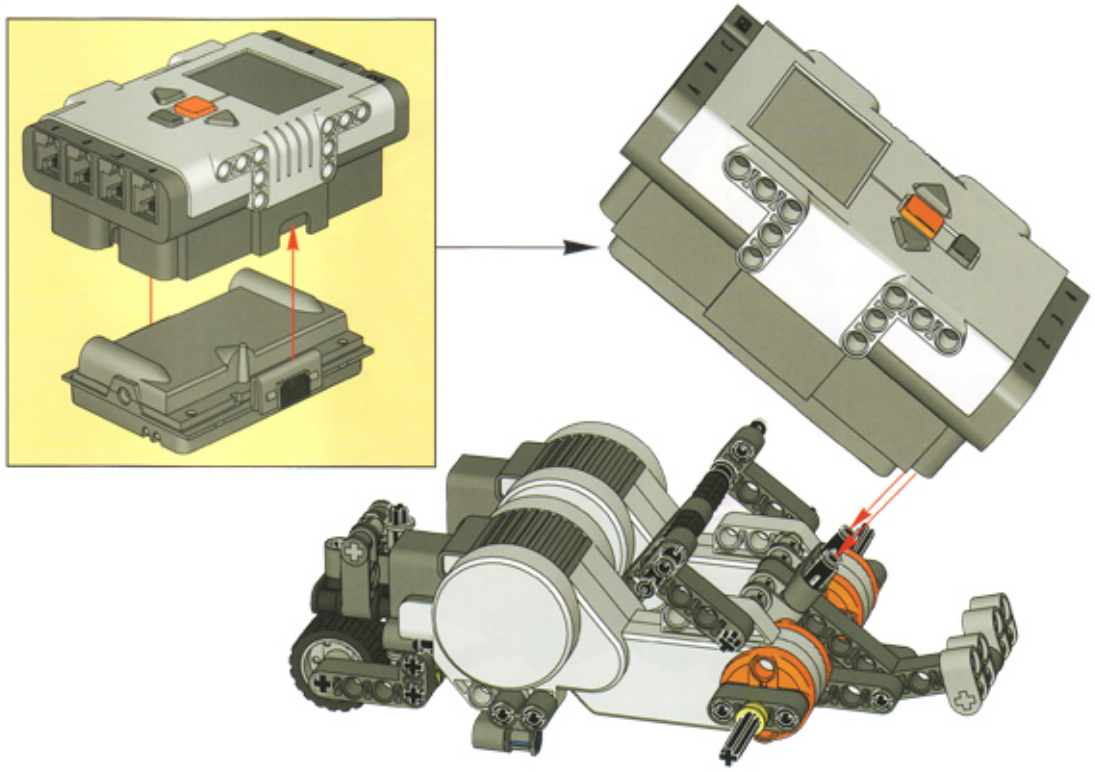
Robot Tribot

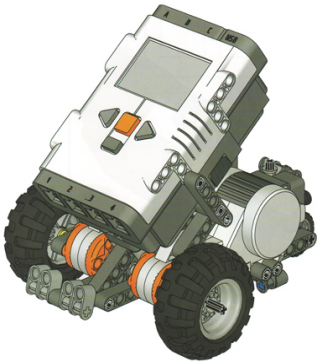
2

#11 Anexos



14

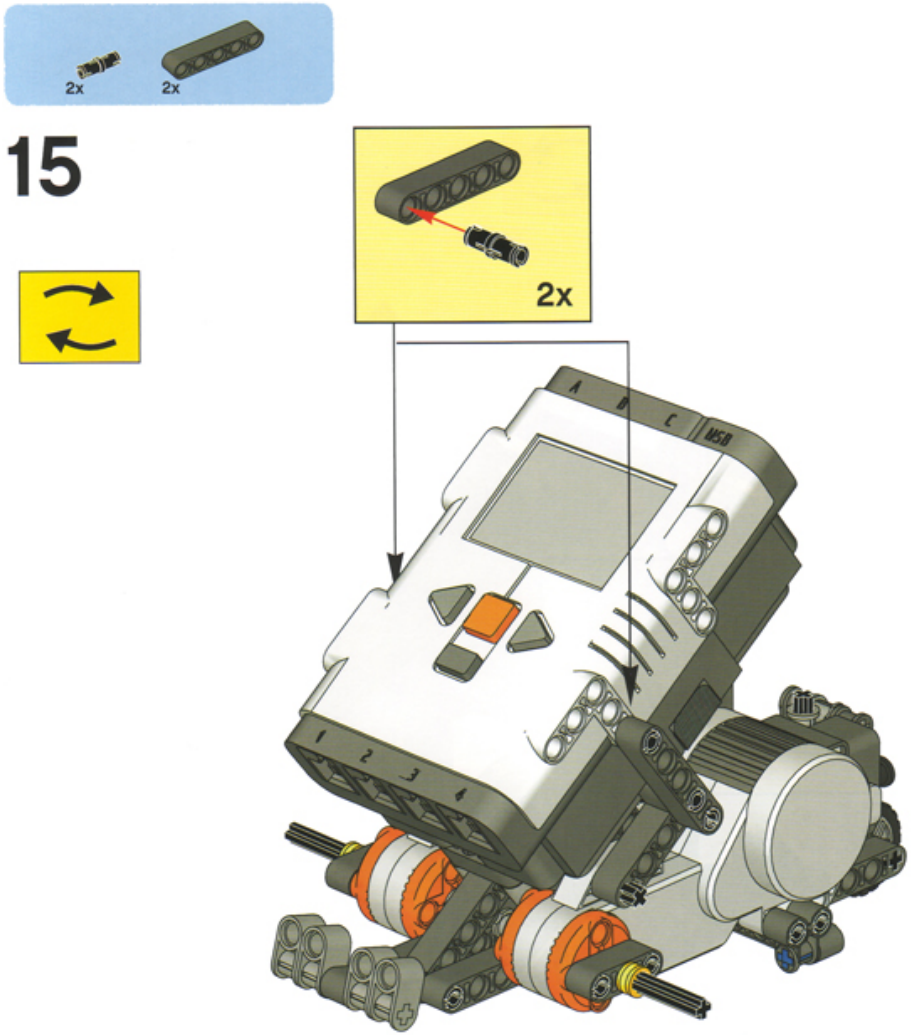




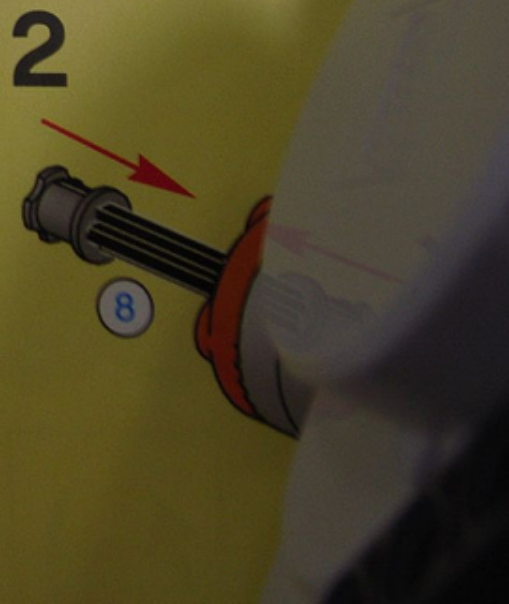
Robot Tribot

2

#11 Anexos

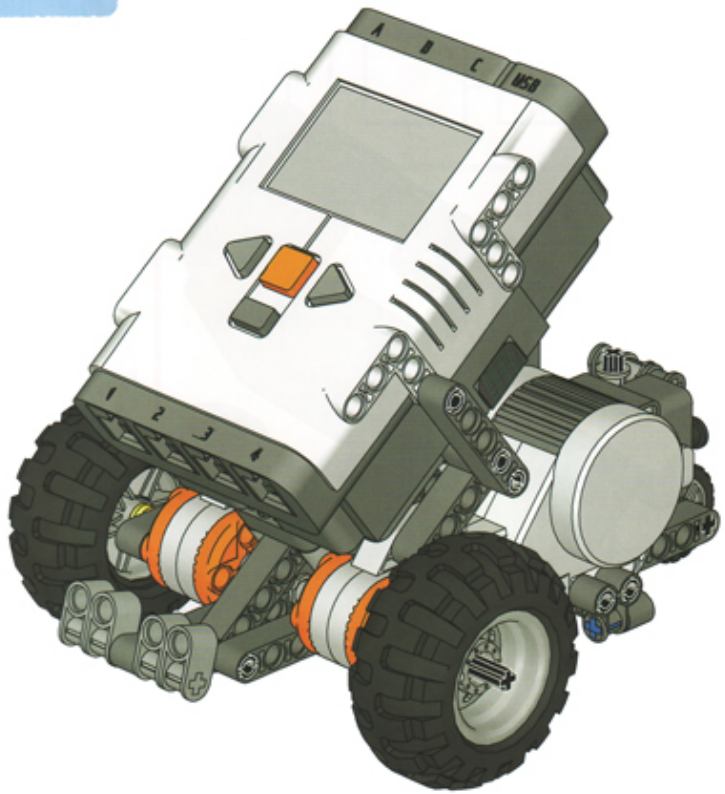


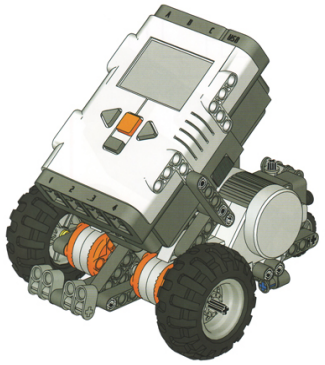
Robot Tribot



#11 Anexos

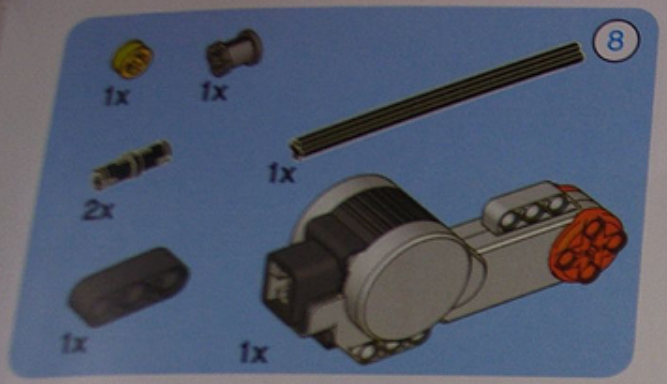
16



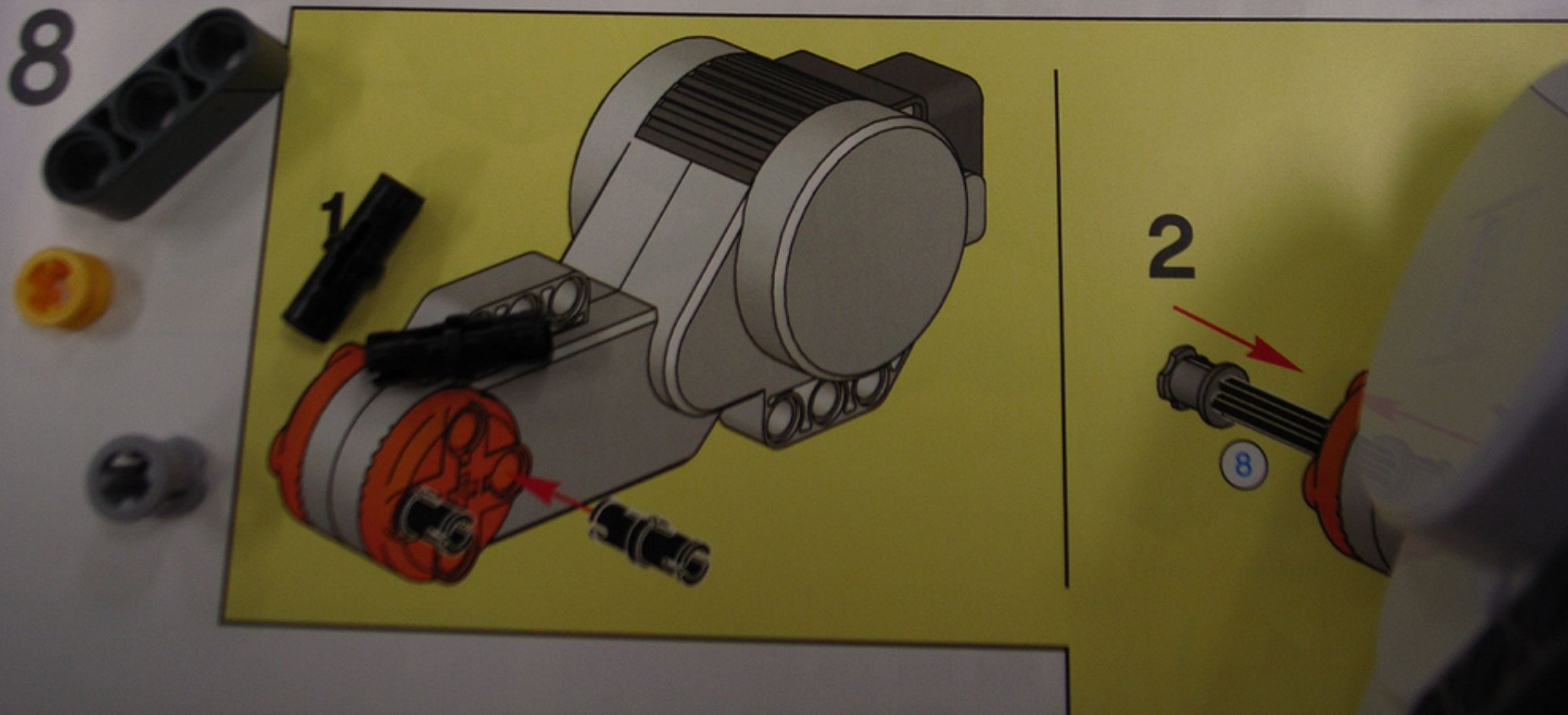


Robot Tribot

2


8
1x
1x
2x
1x
1x
1x


8
1:1


8
1
2
8