

# O Uso de Robótica como Ferramenta de Apoio ao Ensino de Programação em Cursos de Engenharia e Computação

Paula Ceccon Ribeiro

# Uma Breve Introdução ao Java

# Estrutura Geral de Um Programa C

```
/*comentário*/
```

```
<declaração de variáveis>
```

```
<declaração das funções>
```

```
int main(void){
```

```
    <comandos>
```

```
return 0;
```

```
}
```

# Estrutura Geral de um Programa em Java

```
public class <nomedaclasse> {  
  
    /*comentário*/  
  
    <declaração de variáveis>  
  
    <declaração dos métodos>  
  
    public static void main (String[] args) {  
  
        <comandos>  
  
    }  
}
```

# Declaração de Variáveis

## C

1. float x, y;
2. -----;
3. char c;
4. int l, j;
5. int A[5];
6. float B[] = {3, 6, 9, 2};
7. #define pi;
8. char s[] = "Hello World";

## Java

1. float x, y;
2. boolean b;
3. char c;
4. int i, j;
5. int A[] = new int[5];
6. float B[] = {3, 6, 9, 2};
7. static final float pi;
8. String s = "Hello World!";

# Atribuições

C

```
i = i + 1;
```

```
i++;
```

Java

```
i = i + 1;
```

```
i++;
```

# Expressões e Operadores

## C

1. ++ -- !
2. \* / %
3. + -
4. <= >= < >
5. = !=
6. &&
7. ||
8. = (atribuição)

## Java

1. ++ -- !
2. \* / %
3. + -
4. <= >= < >
5. == !=
6. &&
7. ||
8. =

# Condicionais

## C

```
if ( <condition> ) {  
    <statements>  
}  
else {  
    <statements>  
}
```

## Java

```
if ( <condition> ) {  
    <statements>  
}  
else {  
    <statements>  
}
```



# Loops

## C

```
for (i=1; i<=n; i++) {  
    <statements>  
}
```

```
while ( <expression> ) {  
    <statements>  
}
```

## Java

```
for (i=1; i<=n; i++) {  
    <statements>  
}
```

```
while ( <expression> ) {  
    <statements>  
}
```

# Switch e Break

## C

```
switch ( <expression> ) {  
    case <value1>: <statement1> ;  
        break;  
    case <value2>: <statement2> ;  
        break;  
    ...  
}
```

## Java

```
switch ( <expression> ) {  
    case <value1>: <statement1> ;  
        break;  
    case <value2>: <statement2> ;  
        break;  
    ...  
}
```

# Programa Mínimo

## C

```
int main(void){  
  
    printf("Oi \n");  
  
    return 0;  
}
```

## JAVA

```
public class oi{  
  
    public static void main (String[] args){  
        System.out.println("Oi");  
    }  
  
}
```

# Programa Básico

C

```
#define c 2

void soma (int p){
    int v;
    v = 2;
    v = v + c + p;
    printf(“%d \n”, v);
}

int main(void){
    soma(5);
    return 0;
}
```

JAVA

```
public class soma{

    final int c = 2;

    public static void soma(int p){
        int v ;
        v = 2;
        v = v + c + p;
        System.out.println(v);
    }

    public static void main (String[] args){
        soma(5);
    }
}
```

# Lego Mindstorms NXT 2.0

# Visão Geral

- É uma linha do brinquedo LEGO, lançada comercialmente em 2006, voltada para a educação tecnológica.
- É equipado com sensores de luz, toque e som, permitindo a criação, programação e montagem de robôs com noções de distância, capazes de reagir a movimentos, ruídos e cores, e de executar movimentos com razoável grau de precisão.



# Visão Geral

- Possui 4 portas para a saída dos sensores (S1, S2, S3, S4) e três portas para a entrada dos motores (A, B, C).
- Os empregos didáticos abrangem as áreas de automação, controle, robótica, física, matemática, programação e projetos.

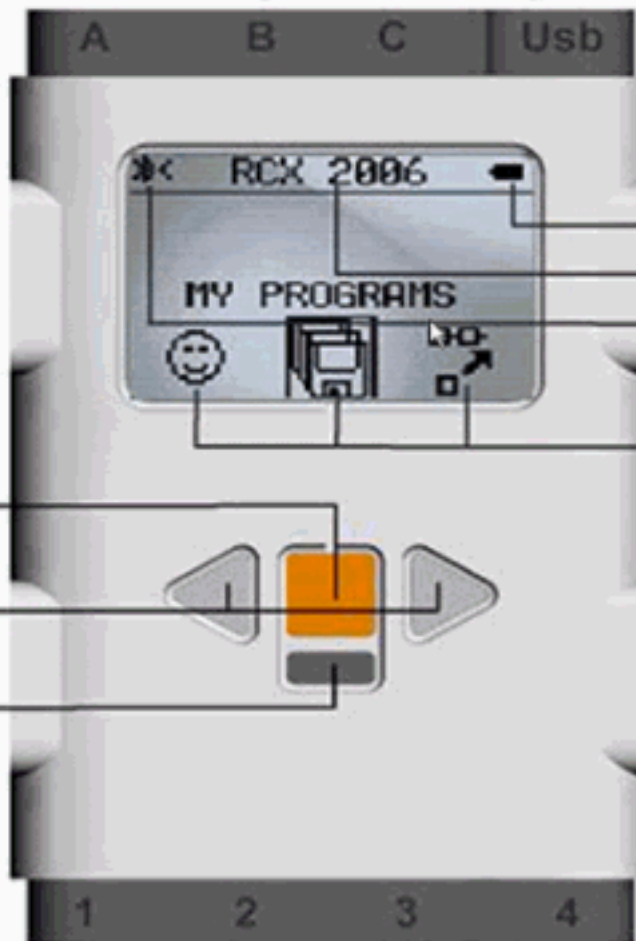


### Output ports

Here you attach the motors

### Usb port

You use this port to download and upload data.



Battery level

Name of the RCX

BT level

Icons

Enter/ON

Select

Back

### Input ports

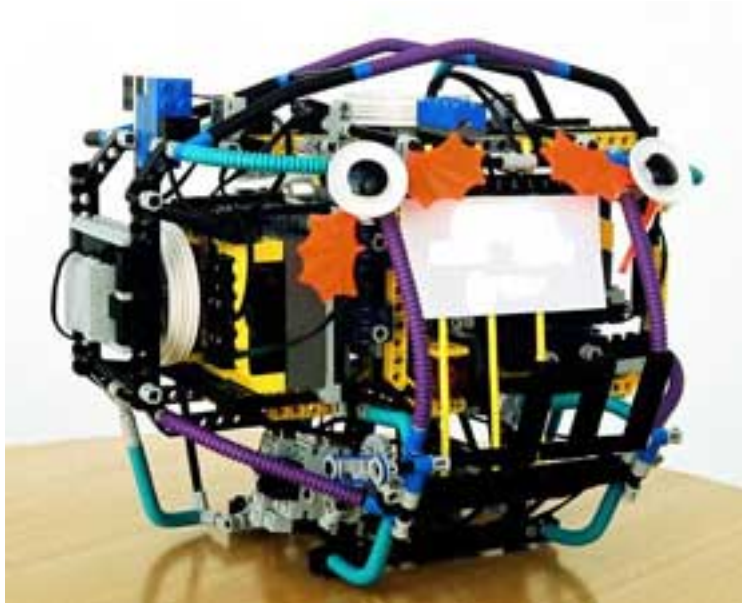
Here you attach the sensors





# Visão Geral

- Lejos é um firmware de substituição para o Lego Mindstorms.
- É muitas vezes utilizado para o ensino de Java à estudantes do primeiro ano do curso de Ciência da Computação.
- O robô Jitter, programado em leJOS, voou ao redor da Estação Espacial Internacional em 2001.



# Programa Mínimo

## C

```
int main(void){  
  
    printf("Oi \n");  
    return 0;  
}
```

## JAVA

```
public class oi{  
    public static void main (String[] args){  
        System.out.println("Oi");  
    }  
}
```

## leJOS

```
public class oi {  
    public static void main(String[] args) {  
        LCD.drawString("Oi", 3, 3);  
    }  
}
```

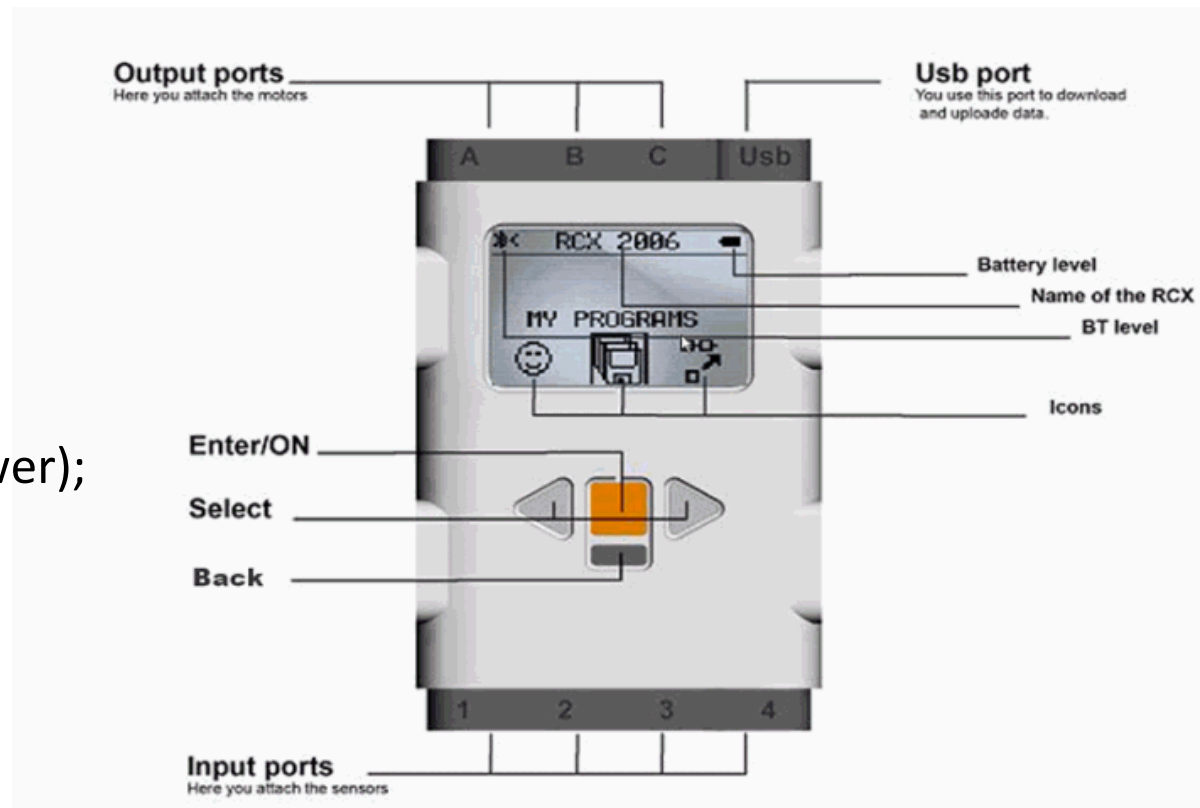
# Comandos Básicos

## Motores:

```
Motor.A.forward();
```

```
Motor.A.backward();
```

```
Motor.A.setPower(int power);
```



# Comandos Básicos

## LCD:

```
LCD.drawChar(char c, int x, int y);
```

```
LCD.drawInt(int i, int x, int y);
```

```
LCD.drawString(String s, int x, int y);
```

```
LCD.clear();
```

# Comandos Básicos

## Button:

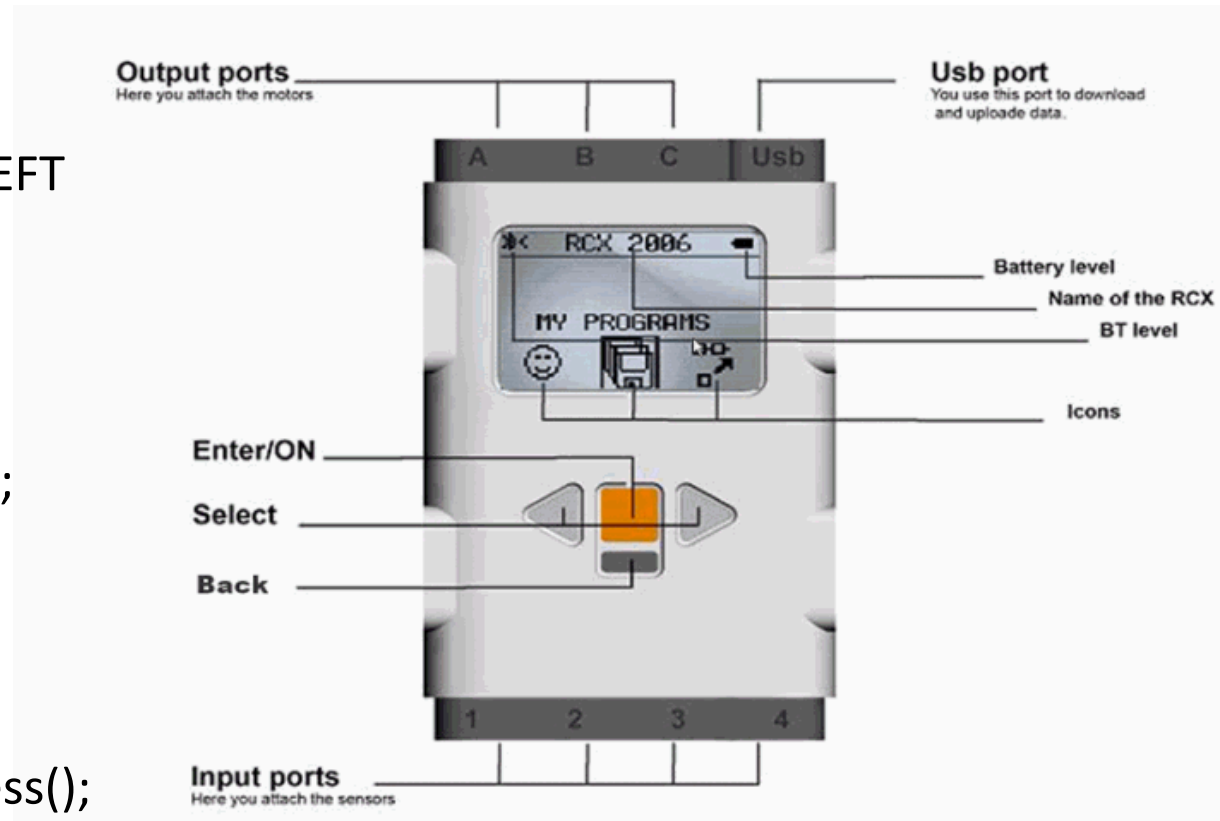
ENTER, ESCAPE, RIGHT, LEFT

```
Button.isPressed();
```

```
Button.ENTER.isPressed();
```

```
Button.waitForPress();
```

```
Button.ENTER.waitForPress();
```

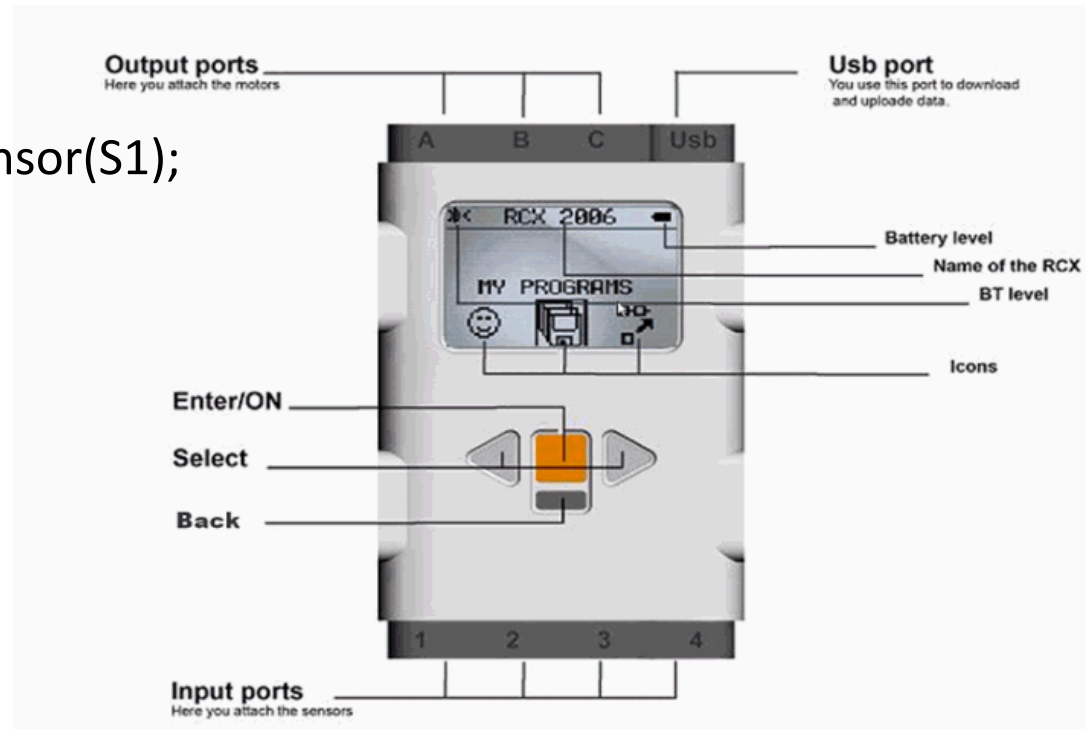


# Comandos Básicos

## Touch Sensor:

```
TouchSensor tc = new TouchSensor(S1);
```

```
tc.isPressed();
```

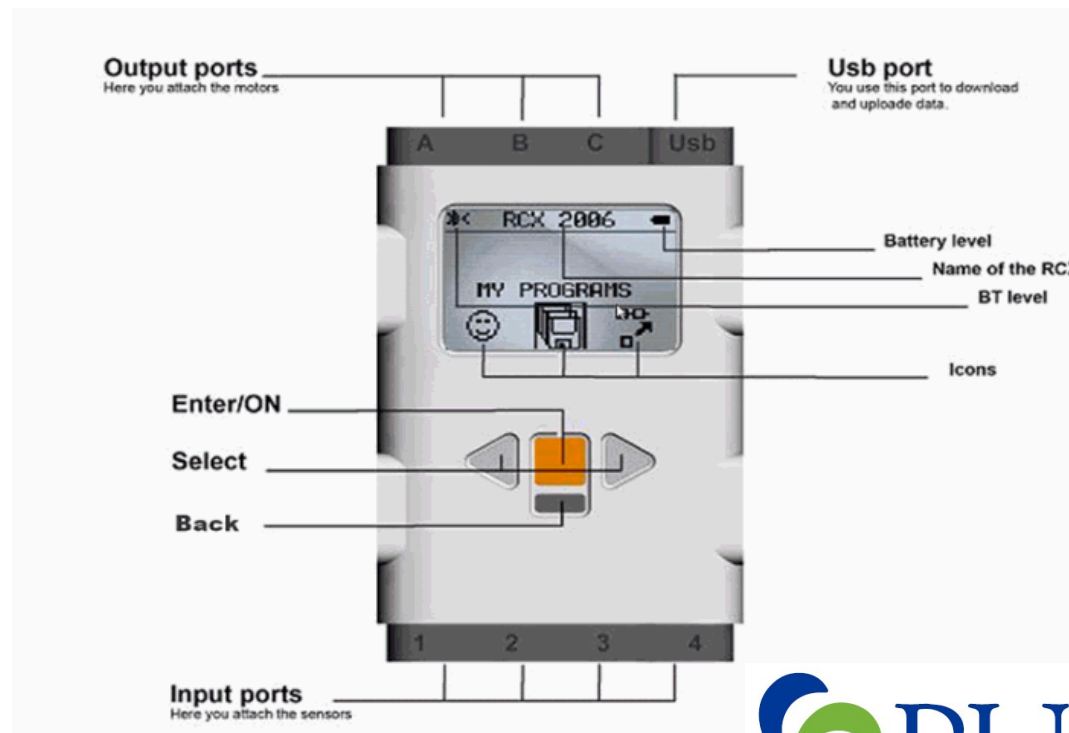


# Comandos Básicos

## Ultrasonic Sensor:

```
UltrasonicSensor us = new UltrasonicSensor(SensorPort.S1);
```

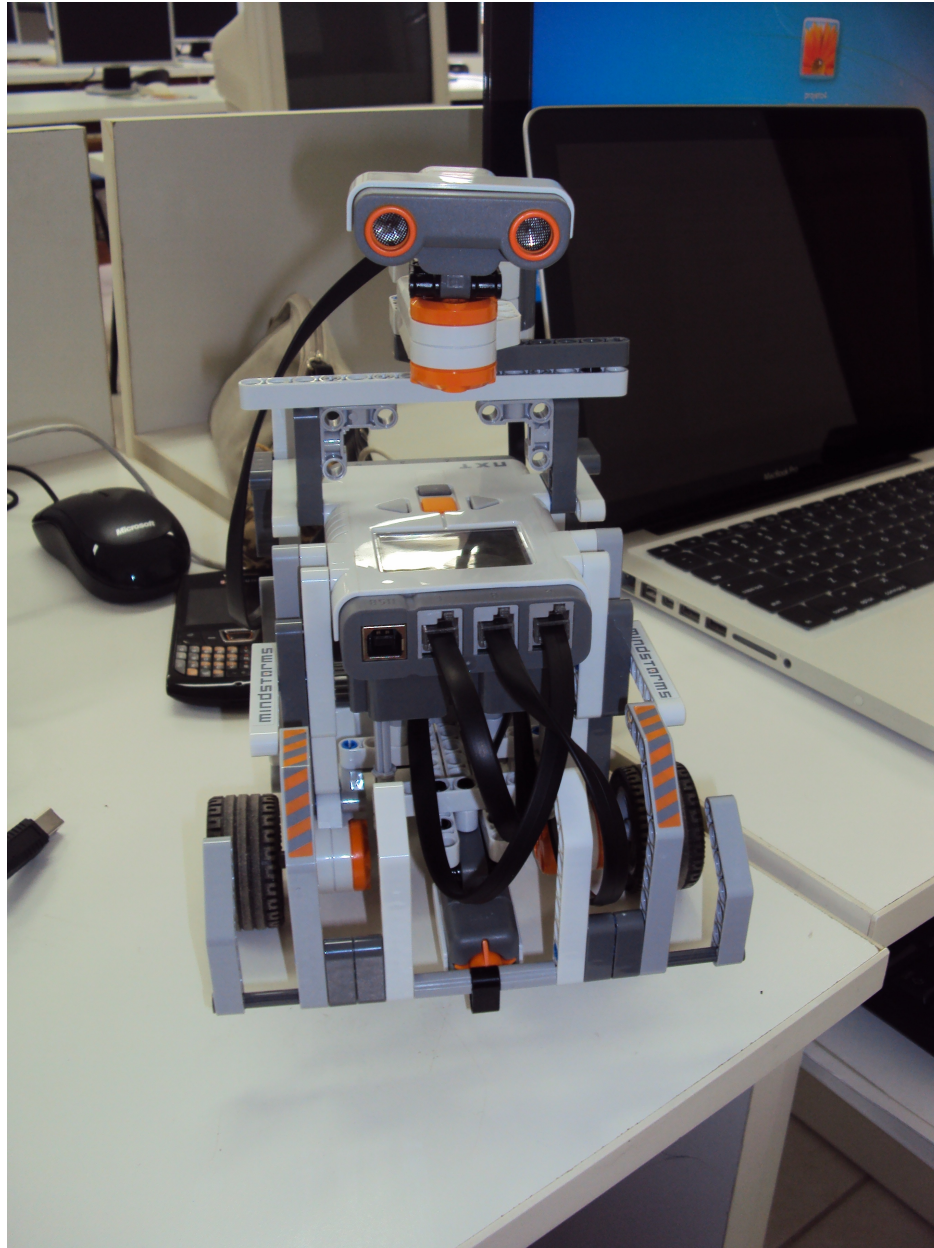
```
us.getDistance();
```





# API

## Experimento : Explorador





# Métodos

setaPotenciaMotores(int  
potencia)

tocouObjeto(TouchSensor  
toque)

inicializaSensorToque()

andaParaFrente()

inicializaSonar()

andaParaTras()

pegaDistância  
(UltrasonicSensor sonar);

# Métodos

viraSonarDireita()

viraSonarEsquerda()

viraSonarPosicaoOriginal()

viraRoboDireita()

viraRoboEsquerda()



# Comandos Importantes

## Color Light Sensor:

```
ColorLightSensor cs = new ColorLightSensor(SensorPort.S1,  
                                             ColorLightSensor.TYPE_COLORFULL);
```

```
Colors.color cor = cs.readColor();
```