

# Tutorial sobre los comandos más empleados con el API de LeJOS para el desarrollo de robots autónomos

Juan Antonio Breña Moral

## índice

Introducción.....	3
Los comandos del proyecto LeJOS.....	3
Comandos de desarrollo.....	4
Nxjc.....	4
Nxj.....	4
Nxjflashg.....	6
Utilidades.....	7
Nxjcontrol.....	7

## Índice de las ilustraciones

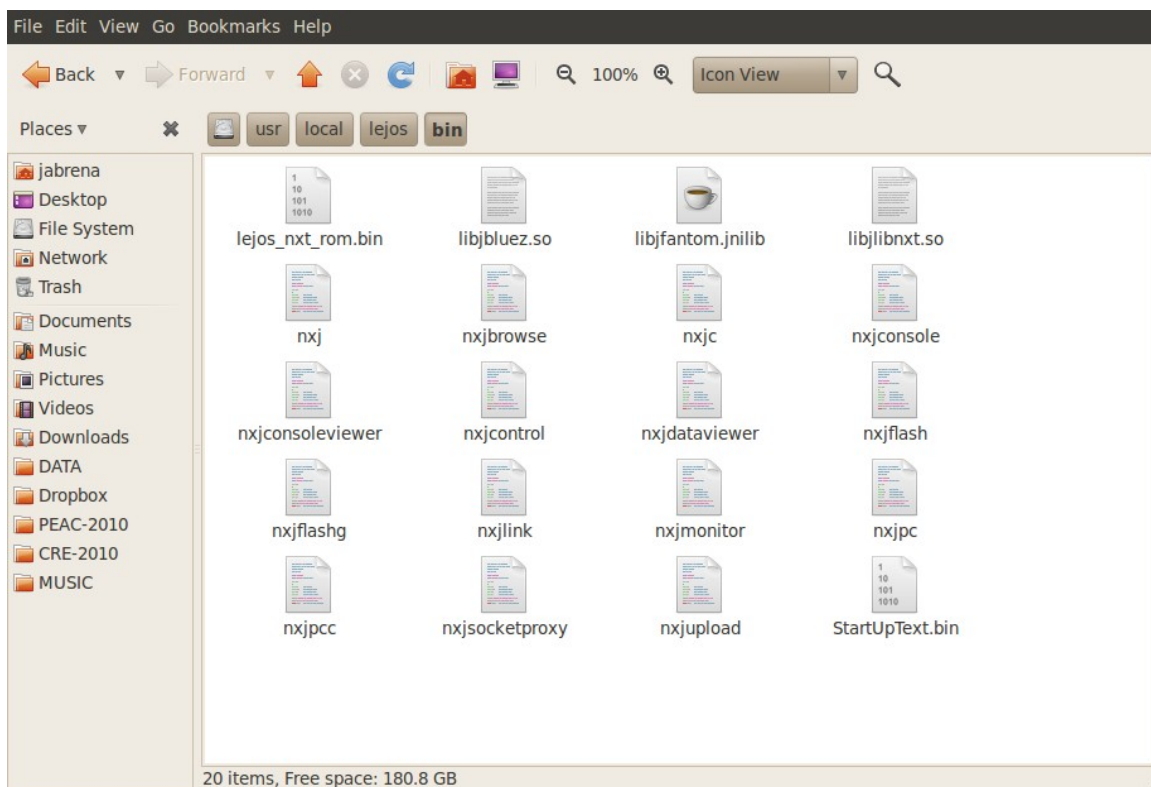
Illustration 1: Comandos de LeJOS.....	3
Illustration 2: Carpeta bin generada por Eclipse.....	4
Illustration 3: Ejecución del comando nxj.....	5
Illustration 4: Ejecución del comando nxj con opcion verbose.....	6
Illustration 5: Herramienta para actualizar firmware.....	7
Illustration 6: Ventana inicial de conexion.....	8
Illustration 7: Listado de ficheros instalados en el brick.....	8
Illustration 8: Valores de los sensores y actuadores.....	9
Illustration 9: Pantalla para la gestión de actuadores.....	9
Illustration 10: Utilidades diversas.....	10

## Introducción

El presente documento, tiene como objeto enseñar los principales comandos empleados en todo proyecto usando el API de programación del proyecto Open Source LeJOS, Java for Lego Mindstorms

## Los comandos del proyecto LeJOS

La ruta por defecto de la instalación de LeJOS es la siguiente: /usr/local/lejos/ sobre dicha ruta, si te situas en la carpeta bin/ encontraras los siguientes comandos:



*Illustration 1: Comandos de LeJOS*

Normalmente los comandos mas empleados serán los siguientes:

1. Comandos de desarrollo
  1. nxjc
  2. nxj
  3. nxjflash / nxjflashg

2. Utilidades

1. nxjcontrol

## Comandos de desarrollo

### Nxjc

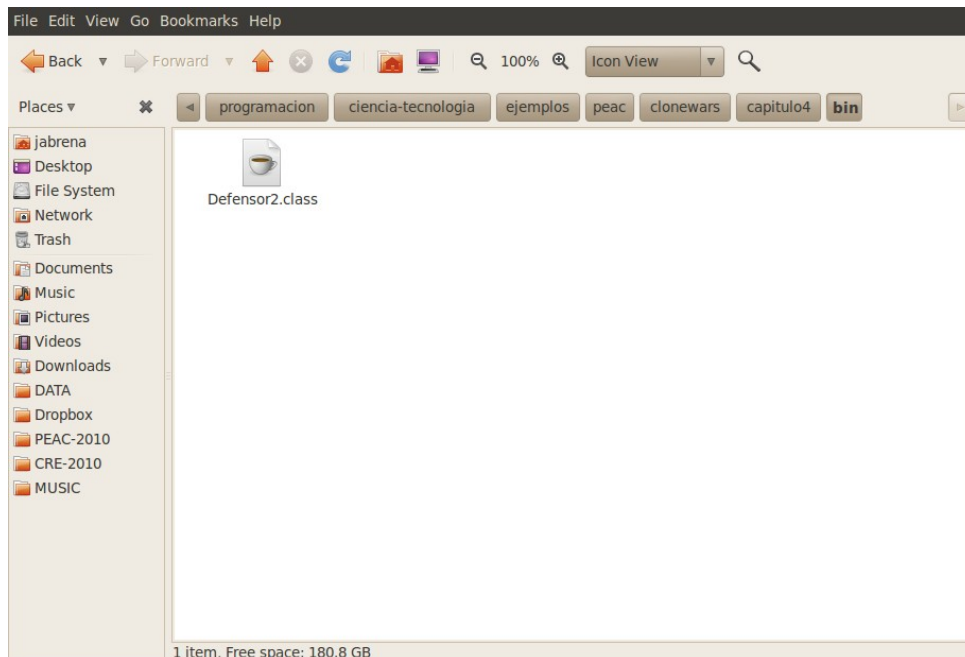
El comando nxjc es un comando que permite compilar un programa en Java que emplea las librerías de LeJOS.

#### Nota:

Si empleas el entorno de desarrollo Eclipse no es necesario emplearlo.

### Nxj

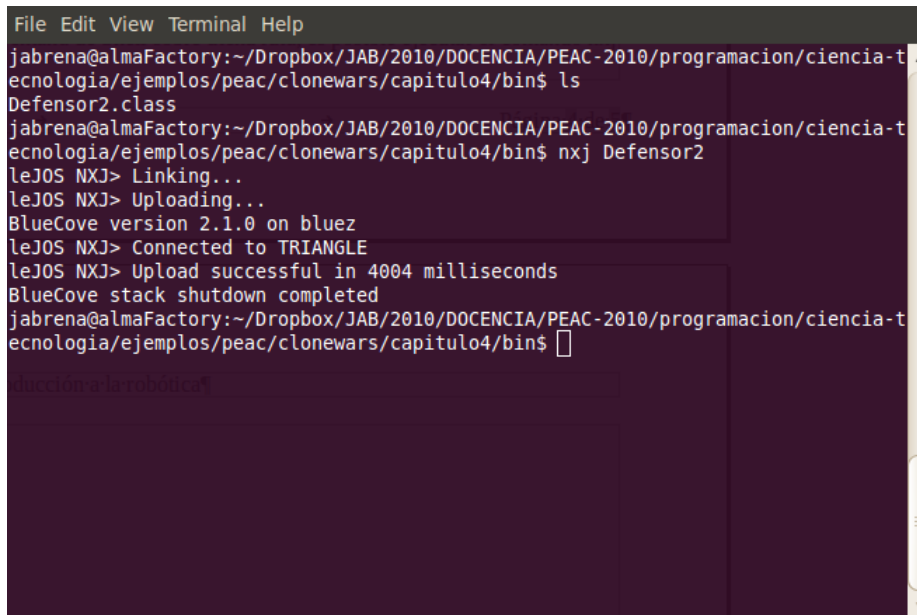
El comando nxj es el responsable de enviar al Brick tu programa. Si empleas Eclipse para el desarrollo de tus programas, los programas compilados estarán en tu carpeta bin:



*Illustration 2: Carpeta bin generada por Eclipse*

Por lo cual, accediendo desde una ventana de Shell sobre dicha ruta, es posible ejecutar el comando nxj y enviar el programa.

*nxj Defensor2*



```
File Edit View Terminal Help
jabrena@almaFactory:~/Dropbox/JAB/2010/DOCENCIA/PEAC-2010/programacion/ciencia-tecnologia/ejemplos/peac/clonewars/capitulo4/bin$ ls
Defensor2.class
jabrena@almaFactory:~/Dropbox/JAB/2010/DOCENCIA/PEAC-2010/programacion/ciencia-tecnologia/ejemplos/peac/clonewars/capitulo4/bin$ nxj Defensor2
leJOS NXJ> Linking...
leJOS NXJ> Uploading...
BlueCove version 2.1.0 on bluez
leJOS NXJ> Connected to TRIANGLE
leJOS NXJ> Upload successful in 4004 milliseconds
BlueCove stack shutdown completed
jabrena@almaFactory:~/Dropbox/JAB/2010/DOCENCIA/PEAC-2010/programacion/ciencia-tecnologia/ejemplos/peac/clonewars/capitulo4/bin$
```

*Illustration 3: Ejecución del comando nxj*

el comando `nxj` es posible hacer que una vez el brick recibe el programa lo ejecute, para ello emplea el parametro `-r`:

*nxj -r Defensor2*

otro parametro muy interesante es `-verbose` que permite darte información sobre el propio programa y si tu programa cuando esta funcionando te da un error, poder buscar en la consola la posible fuente de error.

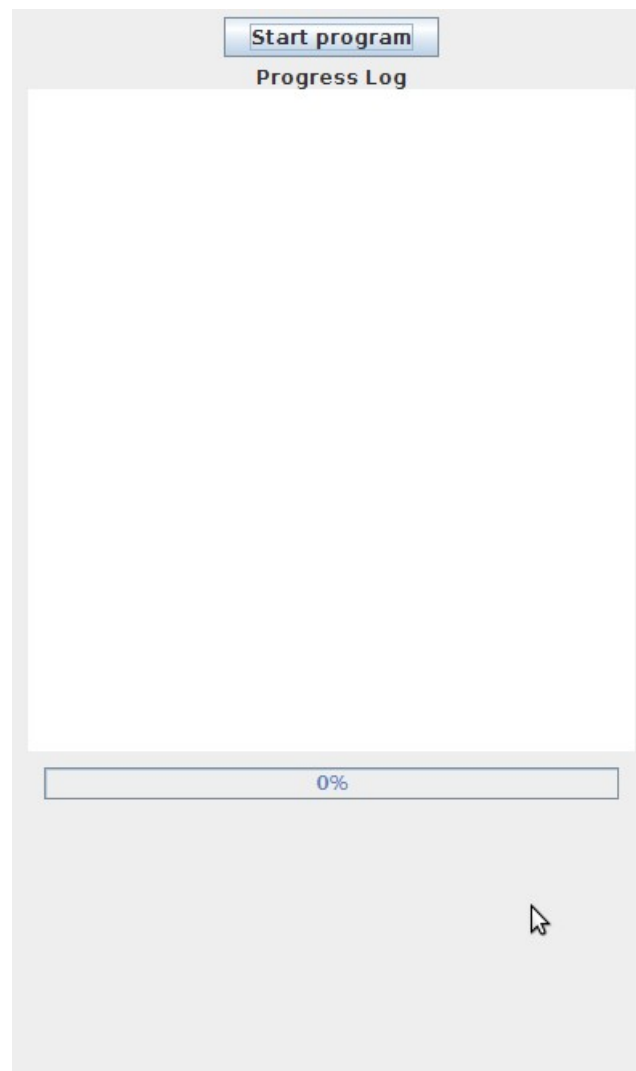
*nxj -r -verbose Defensor2*

```
File Edit View Terminal Help
jabrena@almaFactory:~/Dropbox/JAB/2010/DOCENCIA/PEAC-2010/programacion/ciencia-tecnologia/ejemplos/peac/clonewars/capitulo4/bin$ nxj -r --verbose Defensor2
leJOS NXJ> Linking...
Class 0: java.lang.Object
Class 1: java.lang.Throwable
Class 2: java.lang.Error
Class 3: java.lang.OutOfMemoryError
Class 4: boolean
Class 5: char
Class 6: float
Class 7: double
Class 8: byte
Class 9: short
Class 10: int
Class 11: long
Class 12: void
Class 13: [Ljava/lang/Object;
Class 14: java.lang.NoSuchMethodError
Class 15: java.lang.StackOverflowError
Class 16: java.lang.NullPointerException
Class 17: [Z
Class 18: [C
Class 19: [F
Class 20: [D
```

*Illustration 4: Ejecución del comando nxj con opción verbose*

## Nxjflashg

La herramienta nxjflashg es una herramienta visual que permite actualizar el firmware del brick.



*Illustration 5: Herramienta para actualizar firmware*

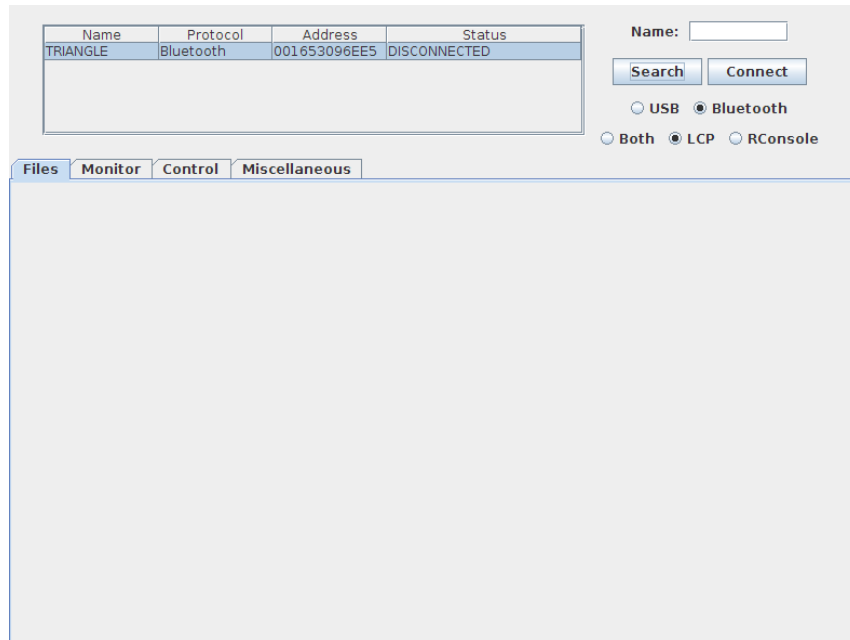
## **Utilidades**

### **Nxjcontrol**

Esta utilidad permite gestionar muchos aspectos del brick desde los ficheros, los datos procedentes de los sensores y actuadores como operaciones diversas.

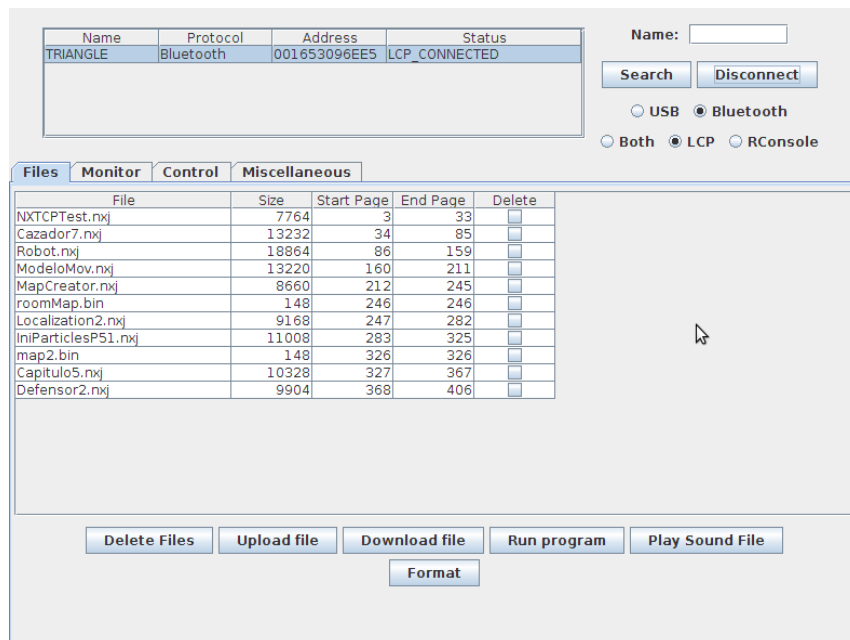
Lo primero que hay que hacer es detectar el brick conectado via usb o bluetooth y conectarse.

## Documentación para cursos de introducción a la robótica



*Illustration 6: Ventana inicial de conexión*

Una vez conectado, es posible ver los ficheros instalados y borrar ficheros antiguos si alguno no te sirve.



*Illustration 7: Listado de ficheros instalados en el brick*

Documentación para cursos de introducción a la robótica

Por otro lado, es posible obtener información de los sensores y actuadores del sistema.

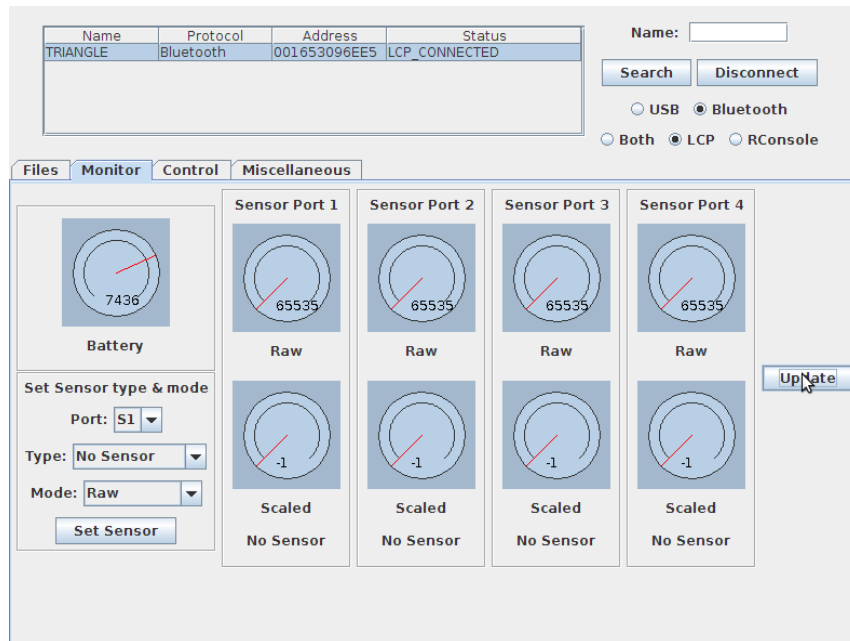


Illustration 8: Valores de los sensores y actuadores

A través de esta ventana, es posible teleoperar el robot de una manera sencilla

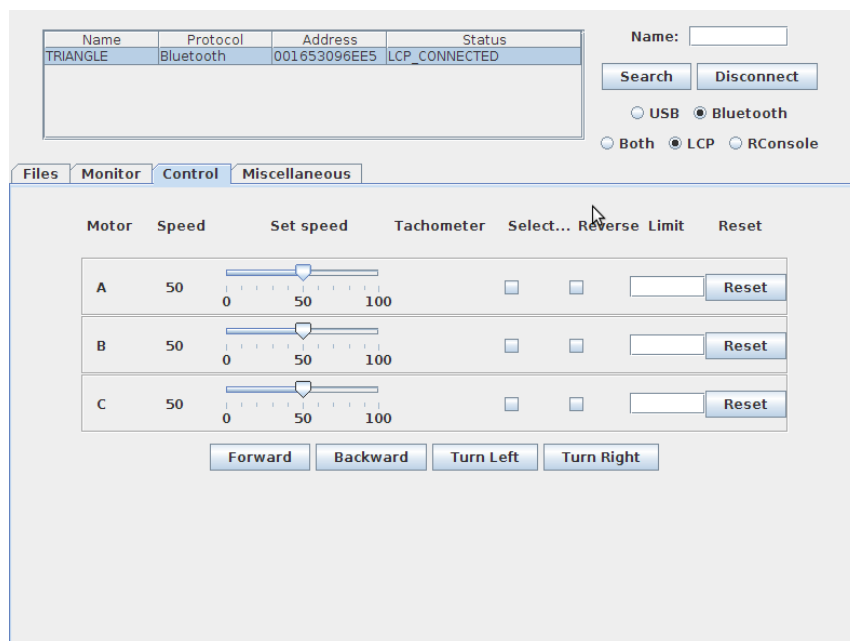
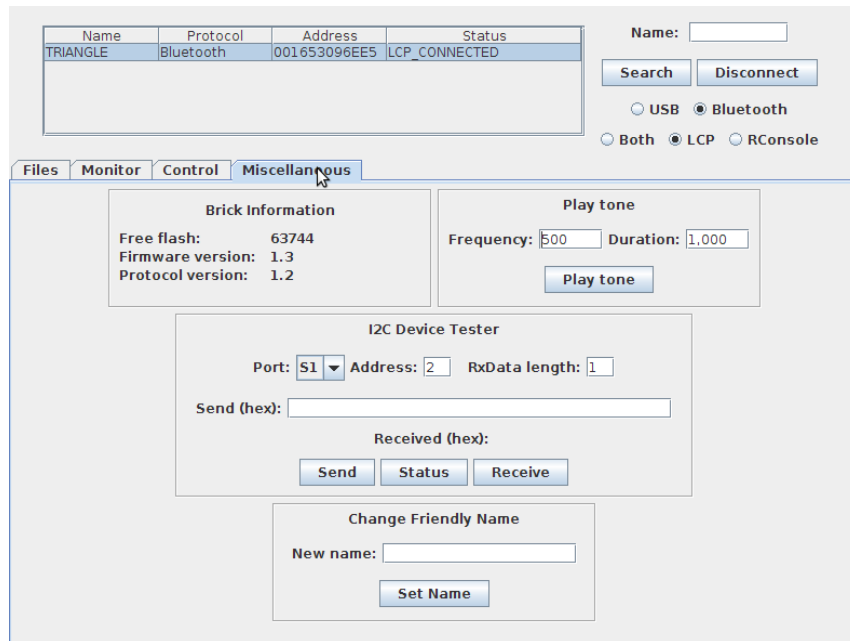


Illustration 9: Pantalla para la gestión de actuadores

Documentación para cursos de introducción a la robótica

Por ultimo, esta venta dispone de ciertas utilidades interesantes.



*Illustration 10: Utilidades diversas*