

# Programando con Python y Robots

Fernando López

LINTI  
Facultad de Informática  
Universidad Nacional de la Plata

29 de Agosto de 2013



- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

- Duinobot → Lihuen/GitHub/Pip
  - StandardFirmata → GitHub/Preinstalado
  - Todos con licencias libres.
- 
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
  - Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
    - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
    - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>



- Duinobot → Lihuen/GitHub/Pip
- StandardFirmata → GitHub/Preinstalado
- Todos con licencias libres.
- Duinobot<sup>1</sup> es la API en Python que usamos hasta ahora.
- Duinobot incluye una copia modificada de PyFirmata<sup>2</sup>:
  - PyFirmata implementa el protocolo Firmata<sup>3</sup>.
  - Escribe en el puerto serie que le indiquemos, en nuestro caso /dev/ttyUSB0 (el XBee).

---

<sup>1</sup><https://github.com/Robots-Linti/duinobot>

<sup>2</sup><https://github.com/Robots-Linti/n6firmata>

<sup>3</sup><http://firmata.org/>

# StandardFirmata del lado del robot

Programando  
con Python y  
Robots

Fernando  
López

- StandardFirmata<sup>4</sup> es un sketch de código Arduino que implementa el protocolo Firmata.
- Lee desde una de las interfaces serial del robot, en nuestro caso el XBee está conectado a esa interfaz.
- Usamos una versión modificada<sup>5</sup> que agrega comandos específicos del robot.

---

<sup>4</sup><https://github.com/firmata/arduino>

<sup>5</sup><https://github.com/Robots-Linti/n6firmata>

# StandardFirmata del lado del robot

Programando  
con Python y  
Robots

Fernando  
López

- StandardFirmata<sup>4</sup> es un sketch de código Arduino que implementa el protocolo Firmata.
- Lee desde una de las interfaces serial del robot, en nuestro caso el XBee está conectado a esa interfaz.
- Usamos una versión modificada<sup>5</sup> que agrega comandos específicos del robot.

---

<sup>4</sup><https://github.com/firmata/arduino>

<sup>5</sup><https://github.com/Robots-Linti/n6firmata>

# StandardFirmata del lado del robot

Programando  
con Python y  
Robots

Fernando  
López

- StandardFirmata<sup>4</sup> es un sketch de código Arduino que implementa el protocolo Firmata.
- Lee desde una de las interfaces serial del robot, en nuestro caso el XBee está conectado a esa interfaz.
- Usamos una versión modificada<sup>5</sup> que agrega comandos específicos del robot.

---

<sup>4</sup><https://github.com/firmata/arduino>

<sup>5</sup><https://github.com/Robots-Linti/n6firmata>

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
  - Beep.
  - Movimientos.
  - Movimientos simples.
  - Figuras.
  - Sensores y condicionales.
  - Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.



- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Libro y materiales.
- Intérprete.
- Conexión
- Beep.
- Movimientos.
- Movimientos simples.
- Figuras.
- Sensores y condicionales.
- Control remoto con `raw_input`.

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

# Algoritmos simples

Programando  
con Python y  
Robots

Fernando  
López

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.



- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

- Detenerse en obstáculo.
- Esquivar.
- Escapar.
- Mantenerse en el perímetro.
- Seguir línea.
- Beep proporcional a la distancia.
- Beep inversamente proporcional.

- 4: Tipos y archivos
- Práctica final para los colegios: armar una demostración y exponerla

- **Interpretado**
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.



- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

- Interpretado
- Tipado dinámico
- Orientado a objetos
- Con herencia múltiple (no hay interfaces como en Java)
- Utilizado principalmente:
  - Programas con interfaz gráfica en Linux (GTK/QT/WxWidgets)
  - Páginas Web (Django/Zope/Otros)
  - Cálculo científico numpy, pylab/matplotlib, scipy.

## Direcciones de contacto

`robots@linti.unlp.edu.ar`  
`soportelihuen@linti.unlp.edu.ar`