

DISEÑO DEL SISTEMA DE CONTROL BASADO EN SOFTWARE LIBRE PARA UN BRAZO ROBÓTICO DE 6 GRADOS DE LIBERTAD CON FUNCIONALIDAD DE MECANIZADO Y PALETIZADO

EDISON VELASCO (*Autor*)

Universidad de las Fuerzas Armadas ESPE-L
Departamento de Eléctrica y Electrónica
Latacunga-Ecuador
edipatvel@hotmail.com

JAVIER MAMARANDI (*Autor*)

Universidad de las Fuerzas Armadas ESPE-L
Departamento de Eléctrica y Electrónica
Latacunga-Ecuador
xavieralexander@gmail.com

Abstract—El presente artículo se centra en el diseño y desarrollo de un sistema de control para un brazo robótico diseñado en la Universidad de las Fuerzas Armadas ESPE extensión Latacunga que utiliza servomotores Dynamixel; el software empleado para dicho sistema es Python que por sus ventajas y características de ser un lenguaje de programación libre brinda al proyecto de fiabilidad y facilidad de comunicación del brazo con un computador. Esto permitirá resolver el problema financiero que presenta varios sistemas robóticos ya que los software que vienen con ellos son propietarios representan grandes inversiones.

PALABRAS CLAVE — BRAZO ROBÓTICO; DYNAMIXEL; PYTHON; SERVOMOTOR; SOFTWARE LIBRE; RS-485; MECANIZADO; PALETIZADO.

I. INTRODUCCIÓN

En el campo de la robótica se puede notar grandes avances a lo largo del tiempo, vemos como robots desempeñan labores que para el humano se han vuelto muy difíciles de realizar y necesitan de estos para ingresar a ambientes peligrosos o inaccesibles, o en el campo médico la necesidad de prótesis robóticas para pacientes discapacitados; este tipo de desarrollo e investigación requiere de grandes esfuerzos e inversiones financieras, es por esa razón que los software se han vuelto propietarios y difíciles de adquirir ya que cuestan un gran valor económico, debido a esto se ha optado en investigar proyectos basados en software libre lo cual reduce costos y da la facilidad de que cualquier persona pueda acceder a estos programas.

La necesidad de obtener más ventajas y viabilidad en el uso de sistemas robóticos empuja a encontrar maneras de manipular estos sistemas a través del uso

de software libre, ya que con el avance de la investigación en lenguajes de programación hacen posibles el desarrollo de nuevas herramientas, extensiones, librerías, etc. que facilitan un lazo de comunicación entre un robot y una PC por medio de un algoritmo codificado en software libre.

Por ejemplo el software libre Python es un lenguaje de programación útil y fácil de codificar, además de que tiene extensiones que son de gran ayuda para las aplicaciones en las que se quiera desarrollar un algoritmo. En este caso el establecimiento de la comunicación con una red de servomotores Dynamixel y el desarrollo de una interfaz gráfica que permita la visualización del proceso.

II. BRAZOS ROBÓTICOS

Los robots industriales son dispositivos mecánicos de funciones múltiples programables diseñados para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos variables programados para realizar varias tareas. Un sistema robótico industrial no sólo incluye los robots industriales, sino también a todos los dispositivos y/o sensores necesarios para que el robot pueda realizar sus tareas, así como la secuenciación o monitorear las interfaces de comunicación. Los robots son generalmente utilizados para realizar tareas inseguras, peligrosas e incluso repetitivas para el operador. Ellos tienen muchas funciones diferentes, tales como el manejo de materiales, montaje, soldadura, carga y descarga de una máquina o herramienta y de funciones como: pintura, pulverización, etc. La mayoría de los robots están configurados para una operación mediante la técnica de enseñanza y repetición. [4]

Un brazo robótico es un tipo de brazo mecánico, normalmente programable, con funciones parecidas a las de un brazo humano; este puede ser la suma total del mecanismo o puede ser parte de un robot más complejo. Las partes de estos manipuladores o brazos son interconectadas a través de articulaciones que permiten, tanto un movimiento rotacional (tales como los de un robot articulado), como un movimiento traslacional o desplazamiento lineal.

A. Servomotores Dynamixel

Los actuadores Dynamixel han sido utilizados por cada universidad importante, laboratorios de investigación civil y militar. Cada actuador inteligente tiene un microprocesador incorporado para facilitar la comunicación de bus, retroalimentación de la posición, la temperatura y la supervisión de la carga. La carcasa de cada servo está construida específicamente para la robótica, proporcionando una facilidad de usar rieles de montaje y un amplio sistema de soporte disponible para la construcción de las extremidades robóticas. Las comunicaciones serial TTL y RS-485 permiten conexiones bus en cadena con transferencias de 1-3 Mbps. Además, los MCU (Microcontroller Unit) a bordo de los servos Dynamixel tienen un conjunto de características personalizables por el usuario, permitiendo a los usuarios ajustar los servos en concreto para su funcionalidad en las aplicaciones requeridas. [5]

B. Lenguaje de programación Python

Python es un lenguaje de programación multiparadigma, esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.

La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.

La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac y otros. Además, Python es gratuito, incluso para propósitos empresariales. [6]

III. DESCRIPCIÓN DEL PROYECTO

En el campo de la robótica se puede notar grandes avances a lo largo del tiempo, vemos como robots desempeñan labores que para el humano se han

vuelto muy difíciles de realizar y necesitan de estos para ingresar a ambientes peligrosos o inaccesibles o en el campo médico la necesidad de prótesis robóticas para pacientes discapacitados; este tipo de desarrollo e investigación requiere de grandes esfuerzos e inversiones financieras, es por esa razón que los software se han vuelto propietarios y difíciles de adquirir ya que cuestan una gran cantidad de dinero, debido a esto se ha optado en investigar proyectos basados en software libre lo cual reduce costos y da la facilidad de que cualquier persona pueda acceder a estos programas.

A. Resumen del proyecto

El presente proyecto está afín a esta ideología por lo que se requiere desarrollar las librerías necesarias utilizando el lenguaje de programación en la plataforma Python para que una red de servomotores de un brazo robótico realice las funciones de mecanizado y paletizado siendo el objetivo primordial del proyecto reducir costos de adquisición en software.

B. Ensamblaje del brazo robótico

Las características que poseen los servomotores Dynamixel los hace requeridos para el desarrollo de distintas aplicaciones robóticas, su control mediante una aplicación en software libre permitirá desarrollar proyectos donde el conocimiento de su control sea más fácilmente entendible y no requiera de compra de licencias para poder avanzar con una investigación futura.

Al tener los servomotores se necesita únicamente de la estructura que conforme el brazo robótico. Para lo cual es prescindible que las piezas que sirvan de unión y soporte para todo el sistema robótico deben ser lo suficientemente resistentes para soportar el peso y fuerza de los servomotores, pero a la vez deben ser ligeras para que el peso total del brazo robótico sea óptimo y no desperdicie potencia tratando de mover su propio peso.

Las piezas y uniones fueron diseñadas en la Universidad de las Fuerzas Armadas ESPE extensión Latacunga y fabricadas en la ciudad de Quito-Ecuador. La

Figura 1 muestra el diseño final del brazo robótico con las piezas manufacturadas por la Universidad.

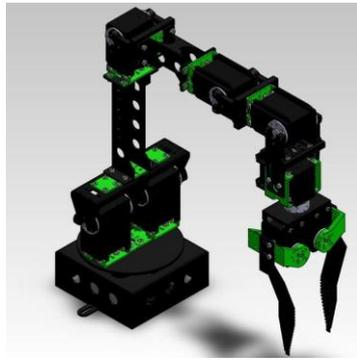


Figura 1.- Diseño final de la estructura del brazo robótico

C. Comunicación USB2Dynamixel

La interfaz USB2Dynamixel es un dispositivo fabricado por la misma empresa de donde se adquirió el kit del brazo robótico utilizado para operar con dispositivos Dynamixel directamente desde una computadora personal.

USB2Dynamixel se conecta al puerto USB del PC, y tiene dos conectores 3P y 4P que están instalados de manera que varios dispositivos Dynamixel pueden ser conectados a estos puertos.



Figura 2.- Interfaz USB2Dynamixel y sus puertos de conexión

Como este brazo robótico está conformado por dispositivos Dynamixel en este caso servo motores de la serie DX/RX se necesita un protocolo de comunicación ya establecido que es el RS-485 y el puerto de conexión es el conector 4P mostrado en la Figura 2.

La interfaz USB2Dynamixel se debe configurar para este tipo de comunicación, para lo cual lo único que se debe realizar es la selección del modo requerido con el switch seleccionador que se encuentra en el dispositivo.

La comunicación establecida entre el computador y la red de servomotores es RS-485 (Estándar EIA-485) es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar (de 10 a 32) y define las características necesarias para asegurar los valores adecuados de voltaje cuando se tiene la carga máxima. Gracias a esta capacidad, es posible crear redes de dispositivos conectados a un solo puerto RS-485. Esta capacidad, y la gran

inmunidad al ruido, hacen que este tipo de transmisión serial sea la elección de muchas aplicaciones industriales que necesitan dispositivos distribuidos en red conectados a una PC u otro controlador para la colección de datos, HMI, u otras operaciones. RS-485 es un conjunto que cubre RS-422, por lo que todos los dispositivos que se comunican usando RS-422 pueden ser controlados por RS-485. El hardware de RS-485 se puede utilizar en comunicaciones seriales de distancias de hasta 1220 metros de cable. La disposición de pines del conector es el mismo de la comunicación RS-232. [3]

D. Comunicación y configuración de la red de servomotores

Una vez que se tiene el modelo del brazo robótico ensamblado y sujeto a una base lo suficientemente resistente y estable se procede a energizar la red de servomotores, estos funcionan con un voltaje entre 11,1v a 14,8v con un consumo de corriente de 1.7 A, 5.2 A y 6.3 A para los modelos MX-28, MX-64 y MX-106 respectivamente. El modo de conexión de la red de servomotores a una PC utilizando la interfaz USB2Dynamixel se muestra en la Figura 3.

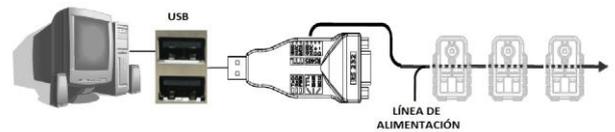


Figura 3. - Conexión de una red de servomotores al PC utilizando la interfaz USB2Dynamixel

IV. PROGRAMACIÓN DEL SISTEMA DE CONTROL E INTERFAZ GRÁFICA

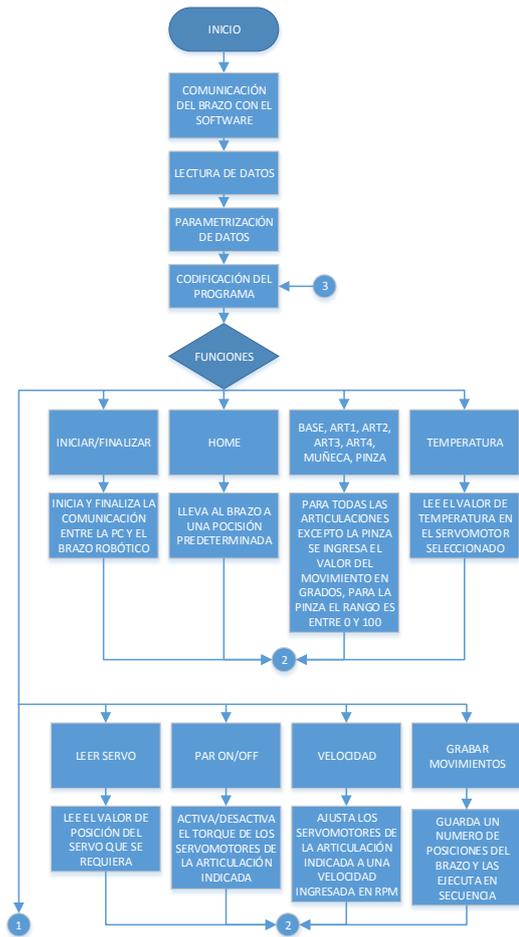


Figura 4.- Diagrama de flujo general de la programación del brazo robótico (parte a)

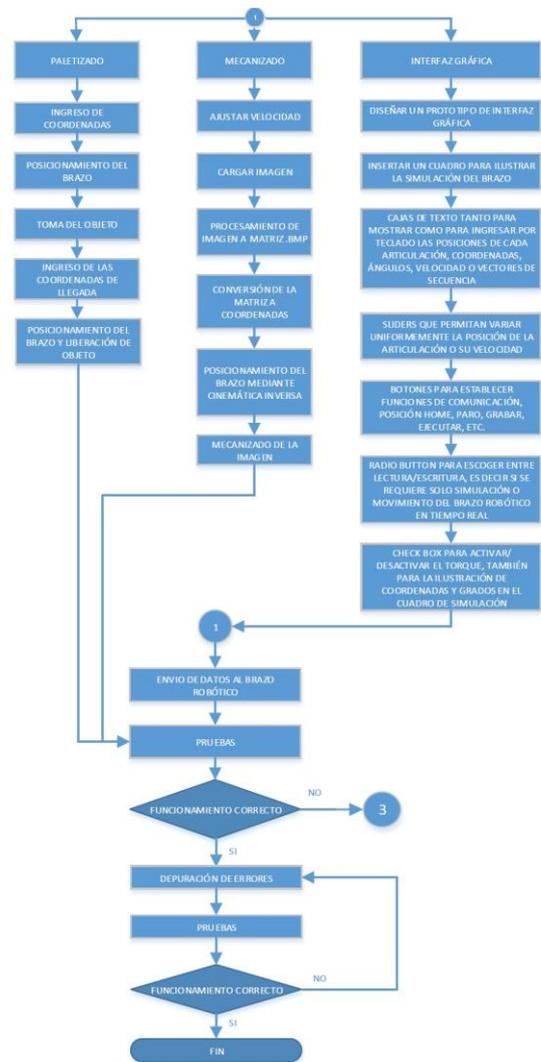


Figura 5.- Diagrama de flujo general de la programación del brazo robótico (parte b)

Al querer desarrollar el sistema de control basado en software libre se empieza por realizar un diagrama de flujo general para poder entablar una comunicación entre el sistema en este caso un brazo robótico y el controlador que es un ordenador de escritorio, todo este proceso se lo puede observar en la Figura 4 y Figura 5.

Como en todo lenguaje de programación se inicia importando librerías si se las necesita, declarando variables, llamando funciones etc.; debido a que este programa no sigue ningún método de control se puede decir que es una programación heurística ya que el brazo robótico sigue las instrucciones del programa línea por línea.

Todo este proceso se lo realizó en dos partes, una de ellas fue el desarrollo del programa encargado de la comunicación con la red de servomotores, de linealizar el número de datos total de una revolución del servomotor a un rango de valores capaz de ser

interpretado por un usuario y que con el mismo pueda realizar varios cálculos y procesos; también en el que se pueda leer los datos entregados por el servomotor como el ID, valor de temperatura, torque, posición, velocidad y entre otros parámetros que dependiendo del modelo de servomotor Dynamixel que se esté utilizando puedan leerse. Por lo tanto en esta primera parte se crearon funciones de lectura y escritura, de donde directamente se puede manipular y configurar los servomotores en forma grupal o individualmente.

En la segunda parte del desarrollo se programó en la extensión de Python llamada Vidle Python, en donde tenemos la opción de crear ventanas, elementos de control, figuras, etc. La necesidad de realizar una interfaz gráfica es sumamente importante ya que se requiere de un sistema amigable y factible de utilizar para usuarios los cuales no tengan conocimientos de programación, entonces en esta interfaz cualquier persona familiarizada con una computadora podrá manipular fácilmente el brazo robótico mediante sliders, botones, cuadros de texto, etc., y observar los cambios realizados tanto en una simulación mostrada en la interfaz gráfica (Figura 6 y Figura 7) como en el sistema robótico físico mismo.

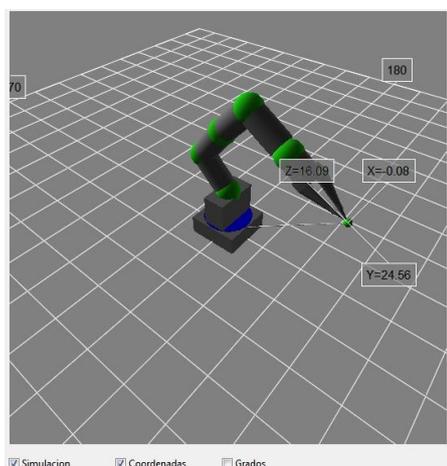


Figura 6. – Cuadro de simulación de la Interfaz Gráfica

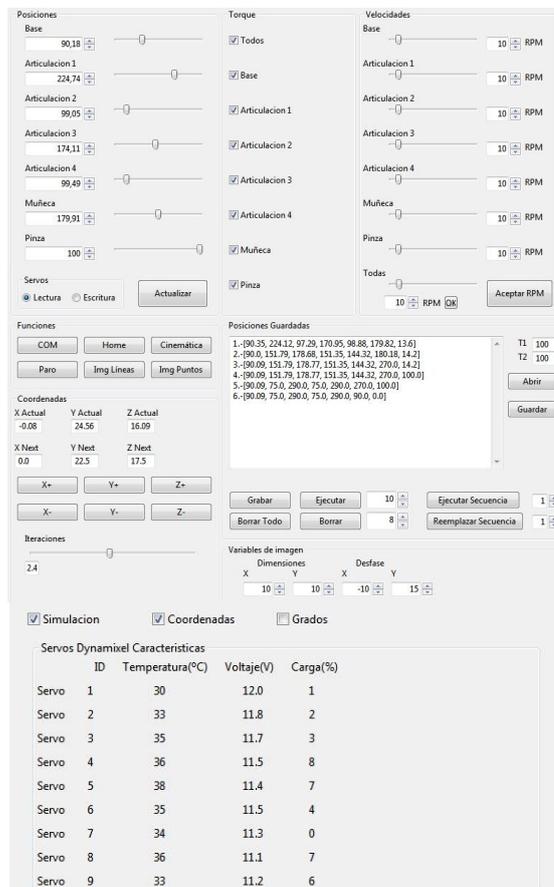


Figura 7. - Controles de la Interfaz Gráfica

Para poder enlazar estas dos partes lo único que se hizo es importar todas las funciones programadas en la primera parte hacia el archivo de programación de la interfaz gráfica como si se tratara de una librería. Los resultados obtenidos se los puede observar en la Figura 8.



Figura 8.- Brazo robótico en comunicación con la interfaz gráfica

V. RESULTADOS

- Los servomotores con comunicación serial presentan grandes ventajas al poseer monitoreo y control interno.
- Con la programación realizada tanto en funciones como en interfaz gráfica se requirió del estudio de otros campos como la cinemática inversa, matemática de vectores, linealización de rangos, procesamiento de imágenes, comprensión de la mecánica en robots y otros temas más para poder relacionar todo como un solo sistema y que trabaje en conjunto.
- Se realizaron funciones de movimiento, ajuste de torque, velocidad, lectura y guardado de los mismos, entre otras; todas estas dependiendo de los parámetros o datos que se podían leer del servomotor.
- Para observar el comportamiento del robot antes de manipularlo se requirió de una función de simulación, gracias a esto se pueden evitar varios accidentes ya que si un usuario ingresa valores de movimiento a una gran velocidad al ejecutar los comandos se corre un riesgo de golpes a las personas cercanas al brazo.
- Se trabajó mucho en lo que es la parametrización de coordenadas, ya que para manipular el brazo robótico al principio solo se ingresaban valores para cada servomotor, manipulando así sus grados de libertad de uno en uno hasta llegar a la posición en el espacio deseada; pero no es lo más óptimo, por ende para movilizar al brazo hasta una posición requerida este tiene que trabajar en conjunto como un sistema en sincronización y utilizando el razonamiento matemático de la suma de vectores se logró que el brazo llegue a una posición únicamente ingresando los valores en los ejes cartesianos x , y y z .
- Para la función del paletizado se trabajó también en coordenadas polares, en donde al brazo robótico le enviamos los datos de las coordenadas en valores de radio, ángulo de giro y altura, se obtuvo los resultados esperados ya que tomaba objetos de dicha posición y los colocaba en otro lugar igualmente designado por el usuario.
- Para la función del mecanizado se tuvo que realizar el estudio e investigación de la cinemática inversa, al comprender este tema la idea fundamental era leer la imagen que se trataba de dibujar y transformarla a una matriz de coordenadas para que el brazo robótico recorra dichos puntos y trace la figura cargada al programa. Para esto lo que

se realizó es procesamiento de imágenes en Python y una relación de escalado de la matriz obtenida de la imagen en unos y ceros a una matriz de coordenadas por donde el brazo robótico recorrería en el espacio.

- Las pruebas realizadas han arrojado resultados favorables y satisfactorios como se esperaban, cada una de las funciones programadas cumplen con sus funciones a cabalidad, una de las funciones principales en la de Programación por recorrido, esta es una técnica en la cual el usuario que hace la programación tiene contacto físico con el brazo robótico y de hecho gana el control y lleva el brazo del robot a través de las posiciones deseadas dentro del área de trabajo, esto es muy usado en la industria para que un robot haga una rutina laboral.

VI. CONCLUSIONES

- De acuerdo a los servomotores utilizados se tiene una configuración diferente de comunicación de la red de servo actuadores con la PC mediante la interfaz USB2Dynamixel.
- El protocolo de comunicación RS-485 nos permite comunicar los nueve dispositivos (servomotores) que conforman el brazo robótico en serie asegurando los valores de voltaje necesarios para que funcione normalmente el sistema robótico cuando estos trabajan al máximo.
- El software libre ofrece posibilidades de adaptar los programas a las necesidades concretas del proyecto.
- El software utilizado Python posee grandes ventajas y múltiples extensiones que permitió el desarrollo de la interfaz gráfica en este proyecto, pero también nos facilita su aplicabilidad en diversos campos de la ingeniería.
- Python permite el manejo de los servomotores Dynamixel, mediante librerías creadas y por medio de una comunicación por el puerto USB y el conversor USB2Dynamixel.
- Los software de diseño y simulación de elementos mecánicos permite usar técnicas de ingeniería inversa.

VII. REFERENCIAS

- [1] R. V. M. Augusta y J. H. F. Valencia, «Construcción de prototipo de brazo robótico controlado por un ordenador,» Escuela Politécnica Nacional, Quito, 2012.
- [2] F. A. C. Herías, «Servomotores,» 20 Septiembre 2007. [En línea]. Available: <http://www.aurova.ua.es:8080/proyectos/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>. [Último acceso: 8 Mayo 2014].
- [3] N. Instruments, «Comunicación Serial: Conceptos Generales,» 6 Junio 2006. [En línea]. Available: <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB68625E0F005888D1>. [Último acceso: 29 Agosto 2014].
- [4] OSHA, «INDUSTRIAL ROBOTS AND ROBOT SYSTEM SAFETY,» 1 Enero 1999. [En línea]. Available: https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html. [Último acceso: 5 Mayo 2014].
- [5] T. Robotics, «Trossen Robotics Dynamixel Guide,» 2006. [En línea]. Available: <http://www.trossenrobotics.com/p/MX-64R-dynamixel-robot-actuator.aspx>. [Último acceso: 20 Junio 2014].
- [6] O. Ocampo, «Programacion en Castellano-Guía de aprendizaje de Python,» 2014. [En línea]. Available: http://programacion.net/articulo/guia_de_aprendizaje_de_python_65. [Último acceso: 20 Agosto 2014].
- [7] P. Org., «The Python Tutorial,» 5 Septiembre 2014. [En línea]. Available: <https://docs.python.org/2/tutorial/index.html#tutorial-index>. [Último acceso: 12 Septiembre 2014].
- [8] C. Tecnologías, «Principales características del lenguaje Python,» 1 Septiembre 2011. [En línea]. Available: http://www.cuatrorios.org/index.php?option=com_content&view=article&id=161:principales-caracteristicas-del-lenguaje-python&catid=39:blogsfeeds. [Último acceso: 20 Agosto 2014].