Guido Van Rossum - Socializar la programación

Developers and Motivations - MSWL Universidad Rey Juan Carlos I

Ricardo García Fernández

3 de julio de 2013

Índice

1. Guido van Rossum and Python in a nutshell

Guido van Rossum es el creador de Python, uno de los lenguajes de programación más utilizados alrededor del mundo[1]. Es un lenguaje de programación bajo una Licencia Libre *Python Software Foundation License -PSFL-*[2]. Está respaldado por una comunidad grande que evoluciona según las necesidades de los usuarios a través de la organización *Python Software Foundation* de la que Guido es su actual presidente, el cual actúa como *Benevolent Dictator for Life -BDFL-*[3] y apoyando la comunidad a través de las propuestas y recomendaciones definidas a través de los *Python Enhancement Proposal - PEPs*[4].

El nombre del lenguaje Python proviene del nombre del grupo de cómicos ingleses Monty Python[5] a diferencia de lo que la gente pueda pensar, no tiene nada que ver con serpientes.

2. Biografía

Guido Van Rossum nació el 31 de Enero de 1956 en los Países Bajos. Estudió la carrera de matemáticas y ciencias de la computación en la Universidad de Ámsterdam[6] *-Curiosamente es la misma Universidad en la que Linus Torvalds desarrolló el primer esbozo del conocido sistema operativo-*. No tuvo acceso a un ordenador hasta que no entró en la Universidad a los 18 años, para el era algo alucinante pero que tenía que compartir con los demás alumnos.

Empezó a trabajar en la Universidad con las tarjetas perforadas, sus primeros pasos en la comunicación con una máquina. Más adelante se rodeó de un grupo de programadores que se reunía en el sótano de la facultad en donde se encontraban las lectoras de tarjetas, las impresoras y ellos. Hablando, discutiendo, al fin y al cabo aprendiendo unos de otros alrededor de un tema que les ilusionaba, la programación.

Guido creó su primer "Hello World!.ª través del lenguaje Algol-60[7] algo tosco para el pero que le ayudó a entrar en el mundo de la programación. El siguiente lenguaje de programación en conocer fue Pascal a través de la recomendación de un profesor, fue directo a conseguir nuevos conocimientos del único libro que había en la Universidad[8] y se quedó con lo más significativo.

Tiene una habilidad para con los lenguajes y por supuesto interés en aprenderlos.

[1] http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
[2] http://docs.python.org/2/license.html
[3] BDFL no se debe confundir con el término usado por Eric Raymon en su ensayo sobre los Líderes de los proyecto de Software Libre el cual los reconoce como dictadores benevolentes en 1997. Este sobrenombre se le acuñó durante un workshop en 1995 en el resumen del mismo:http://www.artima.com/weblogs/viewpost.jsp?thread=235725
[4] http://www.python.org/dev/peps/
[5] http://www.imdb.com/title/tt0063929/
[6] http://www.uva.nl/en
[7] http://www.masswerk.at/algol60/report.htm
[8] http://www.anvari.org/fortune/Miscellaneous_Collections/228172_silver-book-n-jensen-and-wirths-infamous-pascal-user-manual-and-repo.html

Podríamos definir a Guido como un políglota ya que poco a poco se sumaban más lenguajes de programación en su haber; *Fortran, Lisp, Basic, Cobol.* Los lenguajes de programación le atraían más que la programación en si, la comunicación y la forma de ejercerla con la máquina.

El conocimiento de una gran variedad de lenguajes de programación existentes en los 80 le llevó a trabajar para el data center de la Universidad manteniendo el sistema operativo. Aquí tuvo el primer contacto con un sistema Unix donde aprendió C y Shell Scripting. Una de las cosas a destacar de esta posición es que le permitió el acceso directo a una computadora. Programaba porque le gustaba, por placer no por necesidad de crear nada en cuestión, es decir no tenía un *leitmotive*[9].

Hasta que la unión entre sus conocimientos matemáticos junto al interés por los lenguajes de programación, propiciados en el lugar donde se encontraba estudiando y trabajando, tuvo como resultado su inclusión en el proyecto *ABC group*[10] en 1982. ABC es un lenguaje interactivo de programación. Empezó su trabajo en el *Çentre for Mathematics and Computer Science - CWI"* de la Universidad de Amsterdad en donde se puede clasificar como uno de los puntos de inflexión en la historia de Guido.

Su experiencia obtenida en el desarrollo de ABC durante 4 años le sirvió para embarcarse en el siguiente proyecto Amoeba[11], un sistema operativo distribuido. Un proyecto de colaboración entre el CWI y la Universidad Vrije de Amsterdam (Universidad Libre)[12].

Aquí es donde Guido encuentra su motivación, según sus propias palabras:

> Python is a direct product of my experience at CWI. As I explain later, ABC gave me the key inspiration for Python, Amoeba the immediate motivation, and the multimedia group fostered its Michael McLaygrowth.
>
> My original motivation for creating Python was the perceived need for a higher level language in the Amoeba project. I realized that the development of system administration utilities in C was taking too long

Desestimó la portabilidad de cualquier otro lenguaje, incluso de Perl 3 debido a que está fuertemente ligado a Unix y no le gustaba su sintaxis, dato importante que se verá reflejado en Python, la importancia del a sintaxis que está fuertemente influenciado por los lenguajes Algol 60, Pascal, Algol 68. Comienza el desarrollo de Python a finales de 1989. El desarrollo de Python se basa en ABC corrigiendo los errores que contenía bajo su perspectiva. Se publicó internamente la primera versión funcional en el primer tercio del año 1990 en la cual recuerda con cariño sus primeras funcionalidades de las cuales aún perduran como *pgen.*

[9] http://es.wikipedia.org/wiki/Leitmotiv
[10] http://homepages.cwi.nl/ steven/abc/
[11] http://www.cs.vu.nl/pub/amoeba/
[12] http://www.vu.nl/en/index.asp

3. Evolución del proyecto

Se introdujo el uso del sistema de control de versiones *CVS* que fue creado por uno de sus compañeros del *proyecto ABC*, Dick Grune[13] y Guido creó un FAQ que distribuía a través de distintas listas de correo.

En 1993 se creó la propia lista de correo que todavía sigue activa y bajo un software hecho en Python. La eclosión de la comunicación de los usuarios mediante la lista de correo centralizada significó la expansión en si misma de Python. La lista de correo fue el siguiente paso de la evolución a partir de un simple mensaje en el que el asunto es impactante y encierra mucho más significado:

> If Guido was hit by a bus?

Este mensaje proviene de *Michael McLay*[14] y traslada esa cuestión a Guido, *¿ que pasaría con Python si atropella a Guido un autobús ?*. La evolución del sistema estaba centralizada a través de Guido por lo que si el desaparecía, el proyecto caería en una recesión no deseada. El cuerpo del mensaje también explica un parte importante en la evolución de Python, la estandarización del lenguaje, en otras palabras la profesionalización. Debido que si se ha de extender el uso de Python ha de cumplir una serie de estándares para dotarlo de una fiabilidad para que las empresas, organizaciones, etc, lo adopten como su tecnología transmitiéndoles seguridad, robustez y madurez.

Guido fue invitado a trabajar en el *US National Institute for Standards and Technology NIST* a través de M.McLay para seguir evolucionando Python y promoverlo mediante workshops dentro de los EEUU a través del NIST desde 1995 a 2000.

El primer workshop se llevó a cabo en Noviembre de 1994 junto *Ken Manheimer*[15], un desarrollador del NIST, a la que asistieron 20 personas, de las cuales la mitad sigue participando activamente en la comunidad Python e incluso unos pocos se han convertido en líderes FLOSS de proyecto propios(Jim Fulton de Zope y Barry Warsaw de GNU mailman). A través del soporte y la promoción desarrollada en el NIST Guido acabó publicitando workshops delante de más de 400 asistentes lo que hacía que creciera mediante la difusión el conocimiento de Python.

3.1. Python Software Foundation

Python Software Foundation[16] es la fundación de Python que fue constituida en 2001[17]. Se anunció durante el noveno año de la conferencia anual de Python a través de Guido siguiendo los pasos del modelo creado por la Apache Software Foundation. Las funciones destacadas están relacionadas con la educación, el mantenimiento de la web y el fomento del uso de Python a través de sus grupos. Y como más significativo, Guido le

[13] http://www.dickgrune.com/
[14] http://www.python.org/search/hypermail/python-1994q2/1040.html
[15] http://myriadicity.net/
[16] http://www.python.org/psf/
[17] http://www.python.org/psf/press-release/pr200103/

otorga la responsabilidad, es decir los derechos de propiedad intelectual haciendo así el traspaso de poder y liberando Python totalmente a la comunidad.

Y no, no fue Guido el primer presidente, en este caso fue *Dick Hardt* por parte de la una de las primeras empresas en esponsorizar la PSF, ActiveState[18].

4. Contribuciones

Como creador de Python Guido van Rossum ha hecho muchas contribuciones al proyecto mediante código y otras un poco menos tangibles a simple vista que son las contribuciones a la comunidad dotándola de vida y solidez mediante herramientas y acciones. Una comunidad no se autogestiona si no la enseñas a autogestionarse mediante herramientas que lo faciliten.

Para hacer un análisis a groso modo del estado de la comunidad y de las contribuciones de Guido a la misma vamos a utilizar el repositorio de código fuente[19], el bugtracker[20], los archivos de las listas de correo[21] y las PEP[22] como herramientas que han dotado de vida a la comunidad que ha ido alimentando Guido con ellas.

4.1. SCM repository

Mediante el uso de la herramienta online Ohloh podemos ver fácilmente un histórico del repositorio distribuido Mercurial de Python *http://www.ohloh.net/p/python*. A primera vista muestra nos un muestra un resumen del repositorio Python en el que destacamos: un total de *80,648 commits* hecho por *186 contribuidores* y representan *866,300 líneas de código*.

Podemos observar desde los inicios únicamente existía un contribuidor al proyecto, Guido. A partir de Julio 1992 se aprecian dos nuevos contribuyentes más que van alternando su trabajo él durante un año llegando a estabilizar esta cifra durante 6 meses, desde Septiembre de 1993 a Febrero de 1994. En junio de 1994 se produce el famoso mensaje de Michael McLay y esto parece que repercute en la vida del proyecto haciendo crecer el número de desarrolladores base a dos a principios de 1995.

Cuando se muda a EEUU para trabajar durante 5 años en el NIST evolucionando Python con un nuevo grupo de desarrolladores lo contribuyentes básicos crecen linealmente, desde el mínimo de 2 hasta un máximo de 8 al empezar el año 2000 estabilizando un mínimo de usuarios mayor que 10 en ese mismo año llegando a existir 21 diferentes contribuyentes. El trabajo hecho a través del NIST, desarrollo, workshopsks y la gestión de la comunidad mediante listas de correo apoyados por la distribución del código fuente a través de CVS consolidaron la autosuficiencia de la comunidad.

que el proyecto a partir de la creación de la PSF duplicó el número de contribuyentes al mismo de

[18] http://www.activestate.com/
[19] http://hg.python.org/cpython
[20] http://bugs.python.org/
[21] http://mail.python.org/pipermail/python-list/
[22] http://www.python.org/dev/peps/

Existe un perfil de Guido asociado al proyecto Python[23]. Del que podemos destacar que el último año únicamente ha hecho dos aportaciones al núcleo pero sigue siendo el usuario que más commits ha aportado durante toda la historia *10931*.

4.2. Mail list

La lista de correo fue el primer y principal canal de comunicación con los usuarios de Python. Fue lanzada en 1993 y posibilitó la evolución de la comunidad Python a la socialización hacia más niveles. Existe un archivo histórico de la lista[24] pero únicamente a partir de 1999. Desde partir de 2009 se aprecia un descenso de cantidad correo pasando de 2MB a ¡1MB y estabilizando el tráfico a ¡1MB en Noviembre de 2012.

4.3. Workshops

Los workshop como hemos presentado antes crean un efecto positivo dentro de la comunidad de una forma más activa ya que posibilitan la interacción con las personas en el mismo instante creando vínculos más grandes dentro de la misma. Estas reuniones afianzan más las relaciones entre los desarrolladores haciendo más cercano el trato entre ellos, desde Guido hasta el último desarrollador interesado en Python, todos están en el mismo plano compartiendo sus inquietudes de una manera directa. Esta implicación que humaniza el mundo de la tecnología humaniza a Guido frente a la comunidad sin ninguna barrera delante. Entendiendo por barrera una pantalla de ordenador.

4.4. Bugtracker

El bugtracker es una herramienta centralizada que unifica las peticiones conocidas como 'issues'. Un 'issue' puede ser cualquier mejora, error o propuesta encontrados en Python dotándoles de una gran trazabilidad. Se implantó en 2007[25] facilitando la interacción entre desarrolladores, técnicos, usuarios, es decir toda la comunidad. Esta herramienta unifica su potencial junto a la lista de correo enlazados de manera intrínseca exponiendo comentarios y alertando del avance en el estado de las peticiones.

4.5. PEPs

Otro paso significativo fue la creación de los PEPs[26] en Julio del año 2000. Los PEPs son *Python Enhancement Proposals*, las propuestas de mejora de Python. Se dividen varios grupos y están todos reflejados:

- *Meta-PEPs*: las normas para crear un PEP.
- *Other Informational PEPs*: Normas de codificación, HOWTOs y estándares.

[23] http://www.ohloh.net/p/python/contributors/113816964319
[24] http://mail.python.org/pipermail/python-list/
[25] el dato que he obtenido es el del primer bug registrado el 27 de Agosto de 2007 First issue: http://bugs.python.org/issue1034
[26] http://www.python.org/dev/peps/

- *Accepted PEPs (accepted; may not be implemented yet)*: En espera de ser implementados.
- *Open PEPs (under consideration)*: En espera de evaluarse.
- *Finished PEPs (done, implemented in code repository)*: Finalizados e implementados.
- *Historical Meta-PEPs and Informational PEPs*: PEP históricos e informativos.
- *Deferred PEPs*: Diferidos.
- *Abandoned, Withdrawn, and Rejected PEPs*: Abandonados, no tenidos en consideración.

Sirven para definir en que se convierte Python, una hoja de ruta trazada por la comunidad en la que Guido sigue otorgando poder, participación y facilidad de interacción a la comunidad de Python. Están en constante evolución ya que a partir de estos PEPs se define la hoja de ruta del lenguaje, el esqueleto de Python.

5. ¿ Continúa siendo un líder ?

Guido trabaja actualmente (2012) en Google dedicando el 50 % de su tiempo al desarrollo de Python desde su posición de *BDFL*. Es el actual presidente de la *PSF* desde 2002 y participa activamente en workshops. En los distintos perfiles online que posee (ver sección 8), siempre queda bien a la vista el título de 'Python's BDFL 1989 Present'.

Como presidente de la PSF su trabajo se ve reflejado más a través del rol de *BDFL*, mediante charlas, ponencias, talleres que en el desarrollo del propio núcleo de Python como se aprecia en el análisis del repositorio en donde únicamente aparecen 2 commits en el último año a su nombre[27] aunque eso sí, el primer commit es de 1990 y aporta un total de 10931 commits al proyecto siendo el que mayor número ha aportado[28].

Es una persona orgullosa de su trabajo, de que signifique algo y de que no sólo se quede ahí. Guido desde sus inicios con Python repite una y otra vez que la programación debería estar en el día a día de cada uno. Que Python sea conocido y utilizado no sólo por los desarrolladores de software si no que también sea un lenguaje utilizado por la gente no técnica. Guido mantiene la idea en su mente que todo el mundo debería de aprender a programar durante sus estudios de una manera relativamente fácil. Es un medio para socializar la programación y así demostrar lo que aporta siendo una herramienta útil.

> Computer Programming for Everybody, in which he further defined his goals for Python:
>
> - an easy and intuitive language just as powerful as major competitors
> - open source, so anyone can contribute to its development

[27] http://www.ohloh.net/p/python/contributors/113816964319
[28] http://www.ohloh.net/p/python/contributors?query=&sort=commits

- code that is as understandable as plain English
- suitability for everyday tasks, allowing for short development times

6. Recognized rol

Guido consolidó el proyecto gradualmente junto a la comunidad. La primera distribución del proyecto se licenció mediante una Licencia de Software Libre (MIT) y la base de usuarios como hemos visto eran sus propios compañeros de despacho en Ámsterdam. Los usuarios se fueron expandiendo más allá de su despacho al empezar a utilizar la distribución publicada, la comunicación funcionaba ahora a través de la lista de correo.

Guido es un Líder FLOSS del que no he encontrado ninguna polémica. A simple vista parece que le gusta su trabajo y rodearse de gente motivada que cree en lo que hace. Se puede destacar como una cualidad a raíz de haber buscado información que se deja aconsejar, sabe escuchar y más importante sabe valorar los consejos, no todos tienen porque ser buenos dentro de su ámbito. Por otra parte comenzando con el nombre de Python en referencia a los Monty Python, demuestra un buen sentido del humor y que él incentiva orgulloso.

Enlazando este dato con lo estricto que es Python con respecto a la identación podemos ver la parte más profesional de Guido, sabiendo mantener el equilibrio entre el cachondeo y la profesionalización aportando la seriedad y solidez al Python.

Su éxito en el trabajo con la creación y mantenimiento de la comunidad que fue evolucionando, por lo que se ve a mejor, tiene sus bases en el anterior párrafo, *saber escuchar y rodearse de determinada gente*. Las listas de correo como inicio y nutrirlas de información, el paso del código al VCS (Version Control System), la licencia de tipo BSD que acompañaba a Python, los PEPs, la donación de Python a la comunidad a través de la PSF, el bugtracker, los workshops, la cantidad de información producida por el mismo alrededor de Python y la cercanía hacia la comunicación hacen de Guido un Líder FLOSS admirado y respetado porque es de forma recíproca.

Con respecto a las licencias de Software Libre y sus licenciadores, Python ha tenido 4; CWI, CNRI, BeOpen.com y PSF. Esto marca su evolución a través del tiempo ligada a la vida de Guido desde la universidad de Ámsterdam, su mudanza a EEUU para trabajar en el CNRI, la conversión del núcleo de desarrolladores en BeOpen que más tarde pasaría o se transfirió a PSF y por lo tanto pasó a ser la comunidad el propietario de Python.

Al pasar a manos de la PSF solucionó la compatibilidad existente de la licencia de Python con la GPL ya que guido no quería licenciar Python con GPL debido a que organizaciones de software propietario también utilizaban Python y ésto desharía cualquier tipo de colaboración perdiendo una parte de los usuarios. Entre los años 2000 y 2001 entró en juego Richard Stallman[29] presidente de la Free Software Foundation (FSF) abogando para que Python se licenciase como GPL en su totalidad. Mientras que Guido no quería cerrar puertas de alguna manera a todos los posibles socios o colaboradores existentes y por haber dentro del universo Python, como muestra esta entrevista[30]. Por

[29] http://stallman.org/

[30] http://www.linuxtoday.com/infrastructure/2000090701121OSCYSW

lo que en ese par de años se produjeron muchos cambios alrededor de Python y las licencias con las que se publicaba, hasta que se encontró la solución que promovía Guido, la posibilidad del doble licenciamiento creando así una licencia PSF compatible con la GPL de la FSF. Abogó de nuevo por escuchar a todos los actores existentes teniendo en cuenta su experiencia y su opinión para que confluyesen los intereses de todos, un gran mediador reconocido.

En última instancia y como idea general me gustaría hacer hincapié desde una perspectiva más personal remarcaría la implicación de Guido desde un principio de cómo comprender el lenguaje de comunicación con las máquinas que empezó cuando estuvo trabajando con tarjetas perforadas hasta la última versión de Python. Donde al especializarse en lenguajes de programación se aprecia que se convierte casi más en un *lingüista* en su intento de conseguir que la comunicación entre humano-máquina fluya en sintonía. Dando importancia a la idea de querer acercar generando interés a cualquier persona del mundo los lenguajes de programación para que se conviertan en una herramienta importante dentro de nuestras vidas, volviendo al título:

Socializar la programación.

7. Curiosidades

Unas pequeñas curiosidades que he encontrado que rodean a la historia de Python y por supuesto a Guido van Rossum.

La explicación de la sintaxis de Python, su rigidez a la par que su legibilidad en donde la resume en la siguiente frase:

> Python's indentation was invented by the wife of Robert Dewar

Que la explica en alusión a la mujer de Robert Dewar uno de los creadores de B (conocido como ABC del que Python hereda esta robustez) cuando se dispone a explicar a su mujer que estaban implementando mediante la explicación de un *for*. Ella pregunta; ¿ lo que quieres decir que haga el for tiene que hacerlo en las líneas siguientes o sólo la primera ?, se dieron cuenta el error de lectura que acompañaban los *dos puntos* al final de línea creando una confusión lectora. De aquí vino la focalización en la identación en B y varios estudios posteriores referentes al diseño de la sintaxis de un lenguaje de programación.

Se definió una filosofía del desarrollador en Python la cual se conoce como *The Zen of Python* que fue implementada por Tim Peters[31] un desarrollador muy importante dentro del núcleo de Python y publicada como un PEP0020[32]. En ella se describen las buenas prácticas del desarrollo mediante Python en un lenguaje coloquial se podría decir, una especie mandamientos transcritos a una narrativa cercana.

[31] http://www.ohloh.net/p/python/contributors/113816811789
[32] http://www.python.org/dev/peps/pep-0020/

8. Información de contacto

Página web personal: http://www.python.org/ guido/, en la que destaca un apartado llamado *How to Reach Me* donde nos emplaza a enviarle un mail para cualquier pregunta a menos que se refiera al uso de Python, advirtiendo que no lo leerá.

Google Plus: https://plus.google.com/115212051037621986145/posts en donde se describe a si mismo como 'Python's BDFL'.

LinkedIn: http://www.linkedin.com/pub/guido-van-rossum/0/756/4a0 perfil más profesional de Guido donde redacta los proyectos en los que ha trabajado y vuelve a remarcar orgulloso su título de 'Python's BDFL'.

Referencias

[1] Personal home page,
Guido van Rossum,
http://www.python.org/ guido/

[2] The History of Python,
Guido van Rossum & Greg Stein,
http://python-history.blogspot.com

[3] Meet the Mighty Python,
http://pytalent.zandstrasystems.com/meet_py1.html