

Sphinx como herramienta para documentar proyectos técnicos

Javier García Tobar

Sphinx as a tool for documenting technical projects

RESUMEN

La documentación es una tarea importante de un proyecto técnico que puede consumir bastante tiempo. Por esta razón, algunas veces se pasa por alto. Un sistema de documentación de éxito beneficiará a las organizaciones e individuos involucrados directamente. El presente artículo se centrará en resumir las principales características de Sphinx, una herramienta de código abierto creada para generar documentación de Python. La información se formateará adecuadamente en archivos estándar (HTML, PDF y ePub). Basado en esos resultados, este trabajo recomienda Sphinx para escribir eficientemente la documentación técnica de un proyecto. Esta herramienta cumple con los posibles requisitos que necesitará la documentación de un proyecto.

Recibido: 4 de octubre de 2016
Aceptado: 20 de octubre de 2016

ABSTRACT

Documentation is an important task of a technical project and can be quite time consuming. For this reason, it is sometimes overlooked. A successful documentation system will benefit the organizations and individuals directly involved. The present paper will focus on summarizing the main features of Sphinx, an open-source tool created to generate the Python documentation. Information will be nicely formatted to standard files (HTML, PDF and ePub). Based on those results, this study recommends the use of Sphinx to write efficiently the technical documentation of a project. This tool satisfies the possible requirements that project documentation will need.

Received: October 4, 2016
Accepted: October 20, 2016

Palabras clave

Documentación, proyecto, lenguaje de marcas, código abierto.

Keywords

Documentation, project, markup language, open-source.

```

({- extends "layout.html" })

({- block title })
  {{ document.title }}
({- endblock })

({ block css })
  {{ super() }}
  {{ document.css|safe }}
  <link rel="stylesheet" href="/static/websupport-custom.css" type="text/css">
({ endblock })

({- block script })
  {{ super() }}
  {{ document.script|safe }}
({- endblock })

({- block relbar })
  {{ document.relbar|safe }}
({- endblock })

({- block body })
  {{ document.body|safe }}
({- endblock })

({- block sidebar })
  {{ document.sidebar|safe }}
({- endblock })

```



SPHINX

Introducción

Hoy en día, cualquier tipo de proyecto técnico produce una gran cantidad de documentos y hasta que se generan otro tipo de entregables (patentes, programas informáticos, planos, etc.), se convierten en los únicos elementos generados. Esta información se consulta tanto a nivel directivo como entre los propios miembros del equipo del proyecto. Para resaltar la importancia que tienen los documentos dentro de una organización se plantean tres situaciones posibles (Rakos et al., 2005):

- En el caso de que un miembro del equipo abandone el proyecto en curso, este deberá ser reemplazado por otro miembro que deberá utilizar la documentación cedida por su predecesor como legado. Si no existe la documentación o esta es deficiente, la nueva incorporación va a necesitar de un tiempo de aprendizaje mayor para desempeñar la tarea encomendada.

- Si en el transcurso de un proyecto, cambia el equipo de una fase a otra. La documentación es la forma preferente de intercambio de información entre ambos equipos con independencia de que se acuerden otras formas complementarias de comunicación.

- En ocasiones, los proyectos se demoran en el tiempo o se suspenden por diversas causas, como puede ser la falta de financiación. Para poder reanudar un proyecto se requiere analizar la documentación generada hasta la fecha y comprobar el estado actual del proyecto. Si no hay documentación o esta es escasa, entonces podrían repetirse tareas ya realizadas con la consecuente pérdida de tiempo asociada.

Este artículo tiene su origen en una práctica habitual que se lleva a cabo en proyectos de desarrollo de *software* que consiste en la documentación del código fuente de un programa. A esta actividad se considera una buena práctica en programación (McConnell, 1993) y no siempre ha sido del agrado de los programadores, ya sea por falta de tiempo o por no disponer de una herramienta sencilla para hacerlo. En este campo es recomendable la documentación porque sirve al propio programador para recordar lo que escribió hace tiempo, ayuda a otros programadores a que modifiquen su código y favorece la puesta en común de información dentro de una organización. La documentación del *software* debería centrarse en transmitir información útil y con significado más que en in-

formación precisa y exacta (Forward y Lethbridge, 2002).

Para documentar el código fuente de un programa se utilizan herramientas informáticas. En este artículo se presenta la herramienta Sphinx¹ desde un punto de vista práctico y se propone su uso como sistema de documentación en proyectos técnicos, para lo cual se describen sus características principales y se identifican las necesidades que puedan surgir en este proceso de documentación.

Sphinx

Descripción

Sphinx es un *software* generador de documentación para convertir ficheros en lenguaje marcado *reStructuredText* (en adelante, ReST)² en páginas web HTML (incluyendo los archivos de ayuda de Windows) y en otros formatos como PDF o ePub. Por tanto, el usuario escribe las páginas en ReST, cuya sintaxis es sencilla pero al igual que ocurre en Python³ es sensible a la indentación. El logotipo de la herramienta se inspira en el símbolo egipcio del ojo de Horus, *Udyat*, en coherencia con el nombre del programa (figura 1).

George Brandl creó Sphinx en el año 2008 para documentar el lengua-



Figura 1. Logotipo de la herramienta Sphinx.

je de programación Python (Brandl, 2016). Como la herramienta tiene licencia BSD, se ha utilizado no solo para documentar código Python en proyectos de relevancia como *Scipy*⁴ (librería científica) y *Matplotlib*⁵ (librería para representar datos), sino también para documentar otros lenguajes e incluso con fines tan dispares como la publicación *online* de libros o, con las modificaciones oportunas, para escribir un blog personal. Para esta última aplicación, lo más conveniente es utilizar *Pelican*⁶, que es un generador de sitios web estáticos en Python con funciones propias de blogs.

La herramienta Sphinx se actualiza con periodicidad, y su versión estable más reciente es la 1.4.8 (1 de octubre de 2016). Hay otras herramientas similares disponibles en el mercado para documentar código como, por ejemplo, *Natural Docs*⁷ y *Doxygen*⁸. En la actualidad, la documentación técnica en línea se puede realizar mediante los lenguajes de marcas ligero más importantes que son ReST o, por el contrario, usando *Markdown*. Hay cierta controversia sobre qué lenguaje es mejor, pero en cualquier caso ambas sintaxis son similares y se usan ampliamente.

Instalación

Antes de proceder a la instalación de Sphinx, es necesario tener instalado en nuestra máquina el lenguaje de programación Python en cualquiera de sus dos ramas de desarrollo (2.x o 3.x). La instalación de Sphinx es similar tanto en el entorno Windows como en Linux, porque esta herramienta podrá instalarse de forma convencional mediante el comando pip de Python o a partir de la distribución Anaconda⁹.

La herramienta se ejecuta en línea de comandos mediante la sentencia sphinx-quickstart. Para crear un proyecto nuevo de documentación, el usuario administrador debe responder un cuestionario que aparece por pantalla y en el que indica el título del proyecto, su directorio raíz, qué módulos o extensiones específicas se



Figura 2. Página de documentación del proyecto OpenCV.

utilizarán, etc. Esta información se guarda dentro de un archivo de configuración denominado config.py que puede modificarse con posterioridad. Como resultado de la ejecución del comando anterior, se crean las carpetas Source y Build. El código fuente en sentido estricto se incluye dentro de la carpeta Source, que contiene el citado archivo de configuración y un documento maestro en formato ReST denominado index.rst. El documento maestro es la página de bienvenida o portada y muestra, entre otros elementos, el índice de contenidos que enlaza la página principal con otras páginas ReST. En el transcurso del proceso de documentación, el administrador o los usuarios colaboradores añadirán páginas ReST dentro de la carpeta Source. Por otra parte, en la carpeta Build se guardan las páginas en formato HTML, que se han generado a partir de los archivos ReST, y todos aquellos archivos que requieren para su correcta visualización como son: imágenes y documentos, hojas de estilo en cascada (CSS) y archivos javascript (JS).

Diseño

Sphinx tiene una gran variedad de plantillas (en inglés, *templates*) que permiten adaptar los contenidos a nuestras necesidades. Estas plantillas suelen estar diseñadas para que los contenidos se visualicen correctamente en dispositivos móviles. Por otra parte, también se puede crear una plantilla desde cero o, más sencillo, modificar una plantilla existente y cambiar su diseño o funcionalidades según nuestras necesidades. El proyecto OpenCV utiliza para su página de documentación la plantilla

por defecto de Sphinx (figura 2). En la mayoría de plantillas, el contenido se estructura en los bloques: cabecera, bloque lateral, bloque central y pie de página.

Para crear o modificar una plantilla se requieren conocimientos de CSS, HTML, JS y, en menor medida, algunas nociones de Python si se desea implementar alguna funcionalidad más compleja. En la mayoría de los casos, los cambios de diseño se limitarán a modificaciones dentro del archivo CSS, que determina básicamente las dimensiones de la página y de sus bloques de contenido así como otros aspectos gráficos (colores, imágenes y formato del texto). Para facilitar la maquetación de una plantilla puede emplearse los navegadores Mozilla Firefox o Chrome y el complemento CSS Viewer de Nicolas Huon u otro similar.

Contenidos

En este apartado se incluye un ejemplo de código que sigue la sintaxis típica de una página en formato ReST, que es la forma de añadir contenidos mediante Sphinx. En este código se introducen algunos elementos de uso frecuente como son los títulos, subtítulos, párrafos, comentarios, formateadores de texto (cursiva y negrita), tablas y listas.

El resultado de la transformación del anterior código a formato HTML y ePub se muestra en la figura 3a y 3b, respectivamente.

Como se observa, la sintaxis es legible en texto plano y, una vez creado el documento de salida (por ejemplo, HTML), los contenidos se visualizan correctamente según la jerarquía de

```

Esto es un título principal
=====
Aquí hay un párrafo.

.. Esto es un comentario

Título de sección
-----
Otro párrafo, pero dentro de la sección. Esta frase
es cursiva* y esta otra aparece en negrita**. A
continuación, se muestra una tabla de una fila y tres
columnas.

+-----+-----+-----+
| 1       | 2       | 3       |
+-----+-----+-----+

Las listas de elementos pueden mostrarse sin
enumerar.
* Item casa
* Item árbol
O por el contrario, enumerarse:
#. Item 1
#. Item 2

```

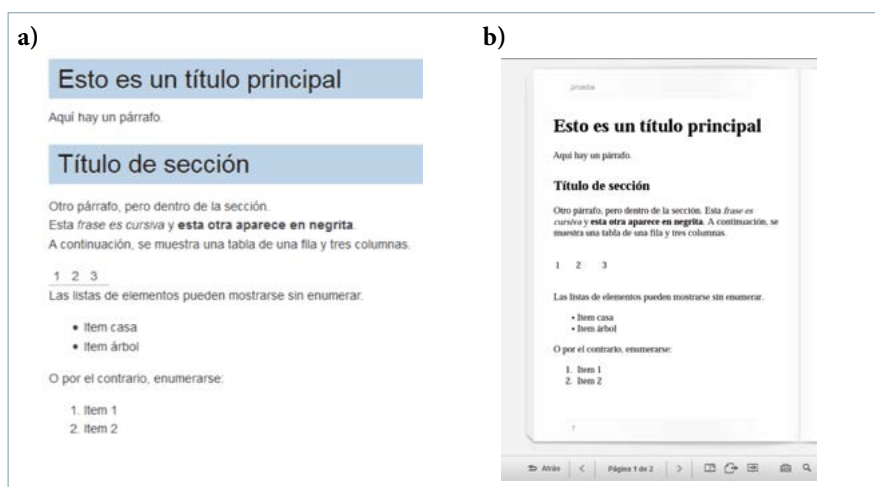


Figura 3. Transformación del código en lenguaje ReST a formato (a) HTML y (b) ePub.

```

html          to make standalone HTML files
dirhtml       to make HTML files named index.html in
              directories
singlehtml    to make a single large HTML file
pickle        to make pickle files
json          to make JSON files
htmlhelp      to make HTML files and a HTML help project
qthelp        to make HTML files and a qthelp project
applehelp     to make an Apple Help Book
devhelp       to make HTML files and a Devhelp project
epub          to make an epub
latex         to make LaTeX files, you can set PAPER=a4 or
              PAPER=letter

[...]
```

títulos, la enumeración de los listados y otros aspectos como el formato del texto. También es posible añadir estructuras de contenidos más complejas como son las tablas.

Con el comando `make` se puede crear a partir de un único código fuente distintos tipos de archivos. En la sección de ayuda de este comando se proporciona un listado con los tipos de archivos de salida que se pueden generar.

Para publicar los contenidos, es decir; para convertir los páginas ReST a HTML, se debe ejecutar el comando `make html`. En este paso de “compilación”, podrán mostrarse avisos que interrumpen el proceso y que suelen estar provocados por la utilización de una sintaxis no permitida en las páginas ReST. Con el desarrollo de los libros electrónicos y otros dispositivos portátiles de lectura (tabletas y *smartphones*), la tendencia actual es generar archivos PDF y especialmente en formato ePub, que son dos opciones disponibles dentro del programa.

Funcionalidades

Aunque se han adelantado algunas características del programa como son los tipos de archivos de salida (HTML, PDF y ePub), esta herramienta también permite utilizar LaTeX. Los documentos que se generan tienen un formato y estructura idónea para su consulta mediante dispositivos electrónicos porque, entre otras razones, en el contenido pueden añadirse referencias cruzadas (citas, glosarios de términos, etc.), enumeraciones automáticas para los títulos u otros elementos como figuras o tablas, fórmulas matemáticas (mediante la extensión `sphinx.ext.mathjax`) y código fuente de lenguajes de programación. Resulta interesante añadir código HTML directamente dentro de una página en formato ReST, lo cual añade más flexibilidad a la herramienta.

En cuanto a la organización de los contenidos, se encuentra el índice de contenidos, denominado *toctree*, que puede mostrarse comúnmente en la página principal (`index.rst`) y/o en su bloque lateral. Otro elemento fundamental en la navegación dentro de Sphinx es el menú horizontal de navegación que suele aparecer en la cabecera de la página y que se denomina *breadcrumb* (o en castellano, “miga de pan”). Este menú resulta de gran utilidad para que el lector sepa en todo momento dónde

se encuentra. Para recuperar la información, destaca el potente buscador interno de la herramienta y que facilita el acceso a la información de forma rápida y precisa.

Como se ha señalado, Sphinx se creó para documentar código fuente Python, por lo que dispone de la extensión específica `sphinx.ext.autodoc` para generar automáticamente la documentación del código Python a partir de sus comentarios (denominados `doctrings`). La herramienta es compatible con otros lenguajes de programación como C++ si se instala un módulo específico para ello.

Sphinx también permite extensiones y módulos para integrar en las páginas, entre otros servicios web, Google Maps, Google Analytics, vídeos de Youtube y presentaciones tipo *slideshare*.

Alojamiento web

Si consideramos que la información, tanto la interna como la externa, es un elemento clave y estratégico dentro de las organizaciones, la forma por la que se produce la documentación en formato electrónico y su acceso son factores esenciales. La documentación generada con Sphinx puede estar dirigida a un público interno o externo a la organización. En el primer caso, tan solo se debe reservar un directorio específico en la intranet para subir la documentación o simplemente habilitar una lista de correo. En el segundo caso, se necesita tener un alojamiento web o más conocido por *hosting* que permita acceder a la información desde el exterior de la organización.

La herramienta Sphinx se diferencia de otros sistemas de gestión de contenidos (CMS), como *WordPress*¹⁰ o *Drupal*¹¹ en que no requiere de una base de datos MySQL para almacenar la información o de PHP para su funcionamiento. Para que funcione Sphinx se precisa de un *hosting* con pocos requisitos, puesto que al servidor se suben exclusivamente páginas estáticas (HTML), archivos CSS y JS. El resultado es que se tiene una alta disponibilidad de las páginas, sin entrar en otros aspectos, como son la simplificación del mantenimiento y la mejora de la seguridad. Además, el alojamiento es considerablemente más económico e incluso, en su lugar, se puede optar por servicios gratuitos y de uso muy extendido como son las plataformas de desarrollo colaborativo *GitHub*¹² y

*Bitbucket*¹³. Para usar las plataformas de desarrollo como *hosting* se necesita configurar el dominio (en concreto, su DNS) y la cuenta de la plataforma para vincular el dominio a un directorio específico de la plataforma. Cada vez que se realice un *commit*, esto es, la actualización de la documentación mediante la aprobación de unos cambios que se consideran definitivos, las páginas podrán actualizarse de forma inmediata. Efectivamente, este tipo de plataformas y la herramienta Sphinx están diseñados para sacar su máximo provecho en trabajos de tipo colaborativo.

Discusión

En un proyecto técnico en cualquiera de sus fases de desarrollo se tienen documentos en distintos formatos (documentos de texto, hojas de cálculo, planos, imágenes, código fuente de programas, etc.). Para integrar esta documentación en un sistema de documentación Sphinx se puede convertir parte del contenido más importante a formato ReST o la opción más sencilla para aquellas documentos de cierta extensión, referenciar los documentos dentro de las páginas. Se debe utilizar la herramienta como un soporte para mostrar información útil y no caer en el error de mostrar una gran cantidad de información, que podría consultarse mejor en su formato convencional (por ejemplo, PDF).

Por otra parte, es posible incluir vídeos y presentaciones, que favorecen que los contenidos sean didácticos y atractivos para el usuario final. Podrán aprovecharse grabaciones o presentaciones realizadas por la organización en reuniones, congresos, etc. Estos formatos de información son idóneos para su visualización en dispositivos móviles como tabletas o *smartphones*, aprovechando que el diseño responde a estos medios.

Se podrán crear varios proyectos de documentación independientes según el tamaño y alcance del proyecto técnico, el equipo de trabajo y la fase del proyecto. Cada proyecto de documentación estará gestionado por un usuario administrador y/o por los usuarios colaboradores. Pese a tratarse de una herramienta de tipo colaborativa, Sphinx no permite gestionar permisos para grupos de usuarios como sí ocurre en las herramientas tipo *wiki* que distinguen entre distintos perfiles de usuario

(administrador, usuario colaborador y lector). Para resolver en parte esta circunstancia, el administrador puede optar por revisar los contenidos en cada puesta en común (o en terminología técnica, *commit*) antes de que los cambios se conviertan en definitivos.

Conclusiones

La herramienta Sphinx es flexible y potente debido al gran abanico de funcionalidades que presenta. Permite organizar la información y acceder a ella de forma eficaz, enfatizando el contenido frente al diseño, y escalable. A esto se suma que es una herramienta gratuita y con una gran comunidad de desarrollo detrás.

El futuro de Sphinx es prometedor porque está cada vez más presente en las organizaciones al satisfacer necesidades como la aplicación que se propone en este artículo. Parte del éxito se debe a que está programado en Python, que es un lenguaje de programación muy demandado en la actualidad, y a que es relativamente sencillo crear y administrar páginas web compatibles con dispositivos móviles y que se adapten al diseño corporativo de nuestra organización.

Bibliografía

- Brandl G (2016). "Sphinx Documentation Release 1.4.9". Disponible en: <https://media.readthedocs.org/pdf/sphinx/stable/sphinx.pdf>
- Forward A y Lethbridge TC (2002). "The relevance of software documentation, tools and technologies: a survey". In *Proceedings of the 2002 ACM symposium on Document engineering*, New York, USA, 26-33.
- McConnell S (1993). "Code Complete". Redmond, WA: Microsoft Press.
- Rakos J, Dhanraj K, Kennedy S, Fleck L y Jackson S (2005). "The Practical Guide to Project Management Documentation". John Wiley & Sons, Hoboken, USA.

Páginas web

(Enlaces consultados y en activo en octubre 2016).

- <http://www.sphinx-doc.org>
- <http://docutils.sourceforge.net/rst.html>
- <https://www.python.org/>
- <https://www.scipy.org/>
- <http://www.matplotlib.org/>
- <http://docs.getpelican.com/>
- <http://www.naturaldocs.org/>
- <http://www.stack.nl/~dimitri/doxygen/>
- <https://www.continuum.io/why-anaconda>
- <https://es.wordpress.org>
- <https://www.drupal.org>
- <https://github.com>
- <https://bitbucket.org>

Javier García Tobar

jgtobar@iies.es
Doctor ingeniero de minas.
