

DOCUMENTACIÓN AUTOMÁTICA CON DOXYGEN

Noelia Sales Montes

`noelia.sales@uca.es`

Diseño de Videojuegos
Universidad de Cádiz

REPOSITORIO

`http://github.com/nessa/taller-doxxygen`

ÍNDICE

- ➊ INTRODUCCIÓN
- ➋ FUNCIONAMIENTO
- ➌ DOCUMENTACIÓN
- ➍ CONCLUSIÓN

ÍNDICE

- 1 INTRODUCCIÓN
- 2 FUNCIONAMIENTO
- 3 DOCUMENTACIÓN
- 4 CONCLUSIÓN

DOXYGEN

Doxygen es una herramienta de generación automática de documentación.

Permite:

- Generar documentación on-line y manuales de referencia off-line.
- Extraer la estructura de ficheros de código **no documentados**.
- Generar documentación de ficheros **expresamente documentados** al “estilo Doxygen”.

DOXYGEN

Doxygen es una herramienta de generación automática de documentación.

Permite:

- 1 Generar documentación on-line y manuales de referencia off-line.
- 2 Extraer la estructura de ficheros de código **no documentados**.
- 3 Generar documentación de ficheros **expresamente documentados** al “estilo Doxygen”.

DOXYGEN

Doxygen es una herramienta de generación automática de documentación.

Permite:

- ➊ Generar documentación on-line y manuales de referencia off-line.
- ➋ Extraer la estructura de ficheros de código **no documentados**.
- ➌ Generar documentación de ficheros **expresamente documentados** al “estilo Doxygen”.

DOXYGEN

Doxygen es una herramienta de generación automática de documentación.

Permite:

- ➊ Generar documentación on-line y manuales de referencia off-line.
- ➋ Extraer la estructura de ficheros de código **no documentados**.
- ➌ Generar documentación de ficheros **expresamente documentados** al “estilo Doxygen”.

¿QUIÉN USA DOXYGEN?

- Asterisk
- Gaim
- GNU (Standard C++ Library)
- GraphViz
- OGRE
- Pingus
- ScummVM
- Synergy
- ...¹

¹<http://www.stack.nl/~dimitri/doxygen/projects.html>

ÍNDICE

- 1 INTRODUCCIÓN
- 2 FUNCIONAMIENTO**
- 3 DOCUMENTACIÓN
- 4 CONCLUSIÓN

INSTALACIÓN

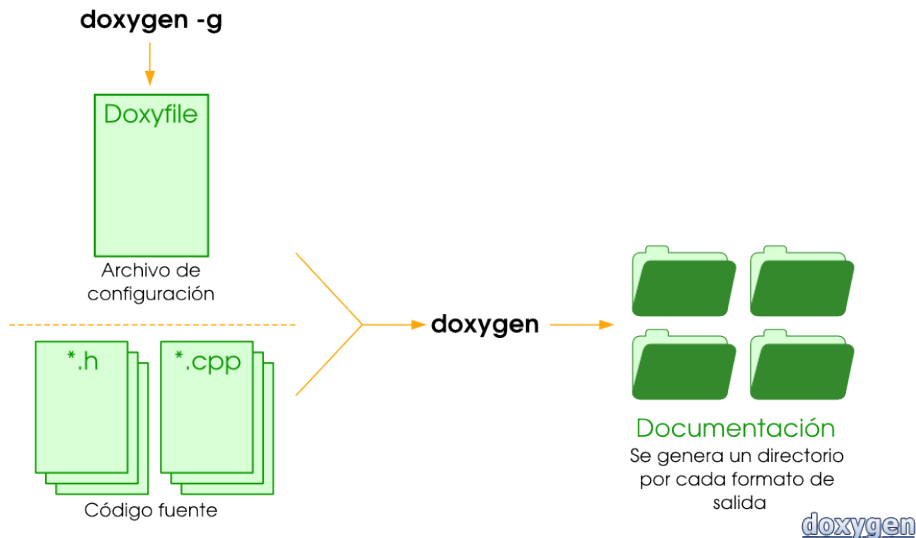
GNU/Linux² - Distribuciones tipo Debian:

```
$ aptitude install doxygen texlive graphviz  
$ aptitude install doxygen-gui doxymacs
```

- doxygen → Paquete completo
- texlive → \LaTeX
- graphviz → Generación de grafos
- doxygen-gui → Interfaz gráfica (opcional)
- doxymacs → Modo para Emacs

²Está disponible para la mayoría de distribuciones de GNU/Linux y para otros sistemas operativos.

¿CÓMO FUNCIONA DOXYGEN?



¿CÓMO FUNCIONA DOXYGEN?

- 1 Documentar el código.
- 2 Generar el fichero de configuración:

```
doxygen -g
```

- 3 Editar el fichero de configuración.
- 4 Generar la configuración:

```
doxygen [doxyfile]
```

Si tienes un *doxyfile* generado con una versión anterior de Doxygen, puedes actualizarlo:

```
doxygen -u [doxyfile]
```

DOXYFILE³

Conjunto de:

- ETIQUETA = VALOR
- # Comentarios

Ejemplo:

```
DOXYFILE_ENCODING      = UTF-8
```

```
# The PROJECT_NAME tag is a single word (or a sequence of  
# words surrounded by quotes) that should identify the project.
```

```
PROJECT_NAME           =
```

```
# The PROJECT_NUMBER tag can be used to enter a project or revision  
# number. This could be handy for archiving the generated  
# documentation or if some version control system is used.
```

```
PROJECT_NUMBER         =
```

³<http://www.stack.nl/~dimitri/doxygen/config.html>

DOXYFILE

- `PROJECT_NAME = Proyecto`
- `INPUT = <fichero_o_directorio_a_leer>`
- `FILE_PATTERNS = *.h, *.cpp`
- `OUTPUT_LANGUAGE = English`
- `HAVE_DOT = YES`
 - Usar GraphViz
- `EXTRACT_ALL = YES`
 - Doxygen asumirá que todas las entidades están documentadas, aunque no sea así.

LENGUAJES DE ENTRADA

- Lenguajes similares a C:
 - C
 - C++
 - C#
 - Objective-C
 - PHP
 - Java
- Python
- Fortran
- VHDL
- ...

FORMATOS DE SALIDA

DIRECTAMENTE SOPORTADOS

HTML GENERATE_HTML = YES

L^AT_EX GENERATE_LATEX = YES

UNIX MAN GENERATE_MAN = YES

XML GENERATE_XML = YES

INDIRECTAMENTE SOPORTADOS

- PDF
- ENABLE_PDFLATEX = YES
 - PDF_HYPERLINKS = YES

EJEMPLO 1

❶ Descargar *octave* y crear el *doxyfile*:

```
mkdir ejemplo1 && cd ejemplo1
wget ftp://ftp.octave.org/gnu/octave/octave-3.6.1.tar.bz2
tar xjf octave-3.6.1
doxygen -g
```

❷ Editar el *doxyfile*:

```
INPUT = octave-3.6.1/liboctave
FILE_PATTERNS = *.h
GENERATE_HTML = YES
GENERATE_LATEX = NO
EXTRACT_ALL = YES
```

❸ Ejecutar *doxygen*

❹ Ver *html/index.html*

OBSERVACIONES DEL EJEMPLO 1

- Se generan enlaces de forma automática:
 - Listado de ficheros
 - Listados de clases (alfabético/jeraquizado) y de miembros
 - En cada clase:
 - Enlace al fichero fuente
 - Enlace a la documentación de cada miembro
- Se generan automáticamente diagramas de herencia
 - `CLASS_DIAGRAMS = YES`
- Se pueden generar gráficos avanzados (GraphViz + `HAVE_DOT = YES`):
 - `GRAPHICAL_HIERARCHY` (por defecto)
 - `rCOLLABORATION_GRAPH` (por defecto)
 - `CALL_GRAP`, `CALLER_GRAPH`

ÍNDICE

- 1 INTRODUCCIÓN
- 2 FUNCIONAMIENTO
- 3 DOCUMENTACIÓN**
- 4 CONCLUSIÓN

BLOQUES DE DOCUMENTACIÓN

```
/**
 * Comentario al estilo JavaDoc
 */

///
/// Comentarios C++ (son tres barras, no dos)
///

/*!
 * Comentario al estilo Qt
 */

/*****
 * Comentario muy visible
 *****/
```

DOCUMENTACIÓN DE UN ELEMENTO⁴

❶ DESCRIPCIÓN CORTA

- Explicación breve en una línea.
- Se puede indicar explícitamente con el parámetro `@brief`.

❷ DESCRIPCIÓN LARGA

- Párrafo con varias líneas.
- Es **necesario** indicar un salto de línea entre la descripción corta y ésta.
- `JAVADOC_AUTOBRIEF = YES`: la descripción larga comienza tras el primer punto en comentarios tipo JavaDoc.

❸ ELEMENTOS ADICIONALES

⁴Todas son opcionales.

ELEMENTOS ADICIONALES

- `@author`
- `@date`
- `@exception`
- `@todo`
- `@{ - @}`: Agrupación de elementos.
- **Otros:** `@class`, `@union`, `@enum`, `@fn` (función), `@file`, `@typedef`, `@namespace`, `@package`, `@interface`, ...

ELEMENTOS ADICIONALES

```
1  /// @brief Clase de prueba.
2  ///
3  /// Descripción más elaborada de la clase de prueba.
4  /// @author Noelia
5  /// @versión 0.1
6  class Prueba {
7      /// @ {
8      /// @name Coordinadas
9      float x ;
10     float y ;
11     /// @ }
12
13     /// @brief Devuelve la coordenada X
14     /// @return Valor flotante de la coordenada X
15     float getX();
16
17     /// ...
18 }
```


EJEMPLO 2

- ❶ Entra en el subdirectorio `materiales/ejemplo2` del repositorio
- ❷ Genera el doxyfile: `doxygen -g`
 - `HAVE_DOT = NO`
 - `HAVE_DOT = YES`
- ❸ Genera la documentación: `doxygen`
- ❹ Visualiza *`html/index.html`*

También se puede probar con `HAVE_DOT = YES` en el proyecto de Octave, pero... tarda un poco ;)

EJEMPLO 2

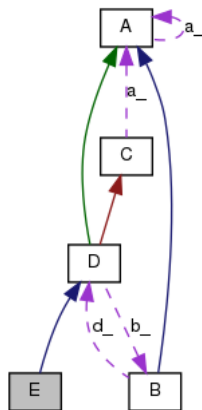


FIGURA: Diagrama de colaboración de la clase E

ÍNDICE

- 1 INTRODUCCIÓN
- 2 FUNCIONAMIENTO
- 3 DOCUMENTACIÓN
- 4 CONCLUSIÓN**

REFERENCIAS Y AGRADECIMIENTOS

Referencias:



<http://www.stack.nl/~dimitri/doxygen/>



<http://www.stack.nl/~dimitri/doxygen/manual.html>



[http://trevinca.ei.uvigo.es/~jgarcia/FP/manuales/
manualDoxygen.pdf](http://trevinca.ei.uvigo.es/~jgarcia/FP/manuales/manualDoxygen.pdf)

Agradecimientos a:

- Rafael Rodríguez Galván y José Tomás Tocino García, por poder inspirarme en sus talleres.
- Dimitri van Heesch, quien desarrolló y liberó Doxygen.

PARA TERMINAR

¿Alguna pregunta?

ESTA PRESENTACIÓN ES LIBRE

Copyright 2012 Noelia Sales Montes

Este trabajo se publica bajo la siguiente licencia:

Creative Commons Attribution-Share Alike License

<http://creativecommons.org/licenses/by-sa/3.0/>

Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:

ATRIBUCIÓN Debes reconocer la autoría de la obra en los términos especificados por el propio autor o licenciante.

RECIPROCIDAD Si alteras, transformas o creas una obra a partir de esta obra, solo podrás distribuir la obra resultante bajo una licencia igual a ésta.