Documentación automática

doxygen

¿Porqué documentar?

- Seguirás usando tu código meses después de escribirlo
- Querrás que la gente entienda tu código
- Aumenta la calidad del código
- El código se escribe una vez pero se lee muchas veces

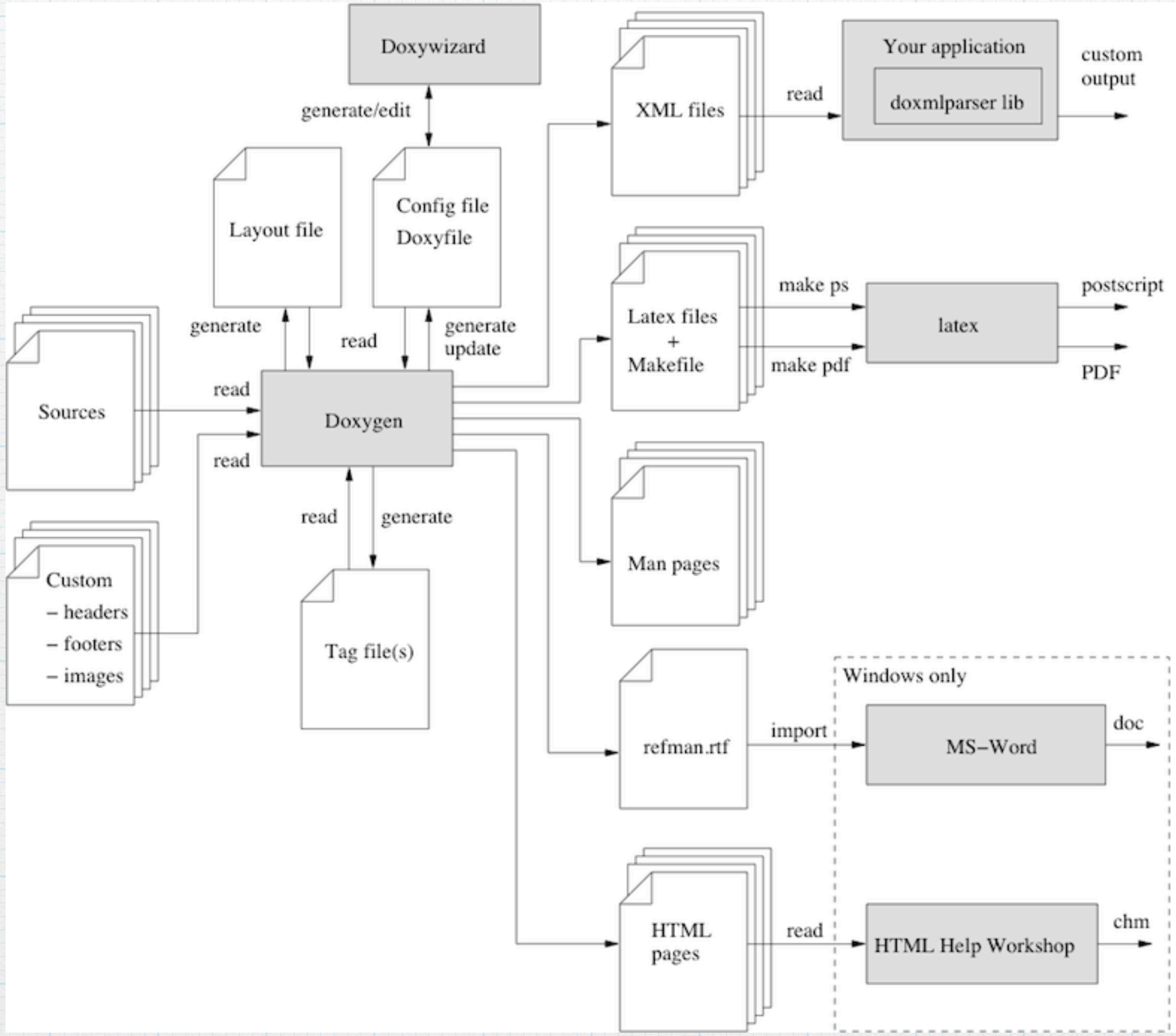
Tipos de documentación

- Captura de requisitos
- Comentarios
- README.md (Markdown)
- End-User documentation
- API ReST

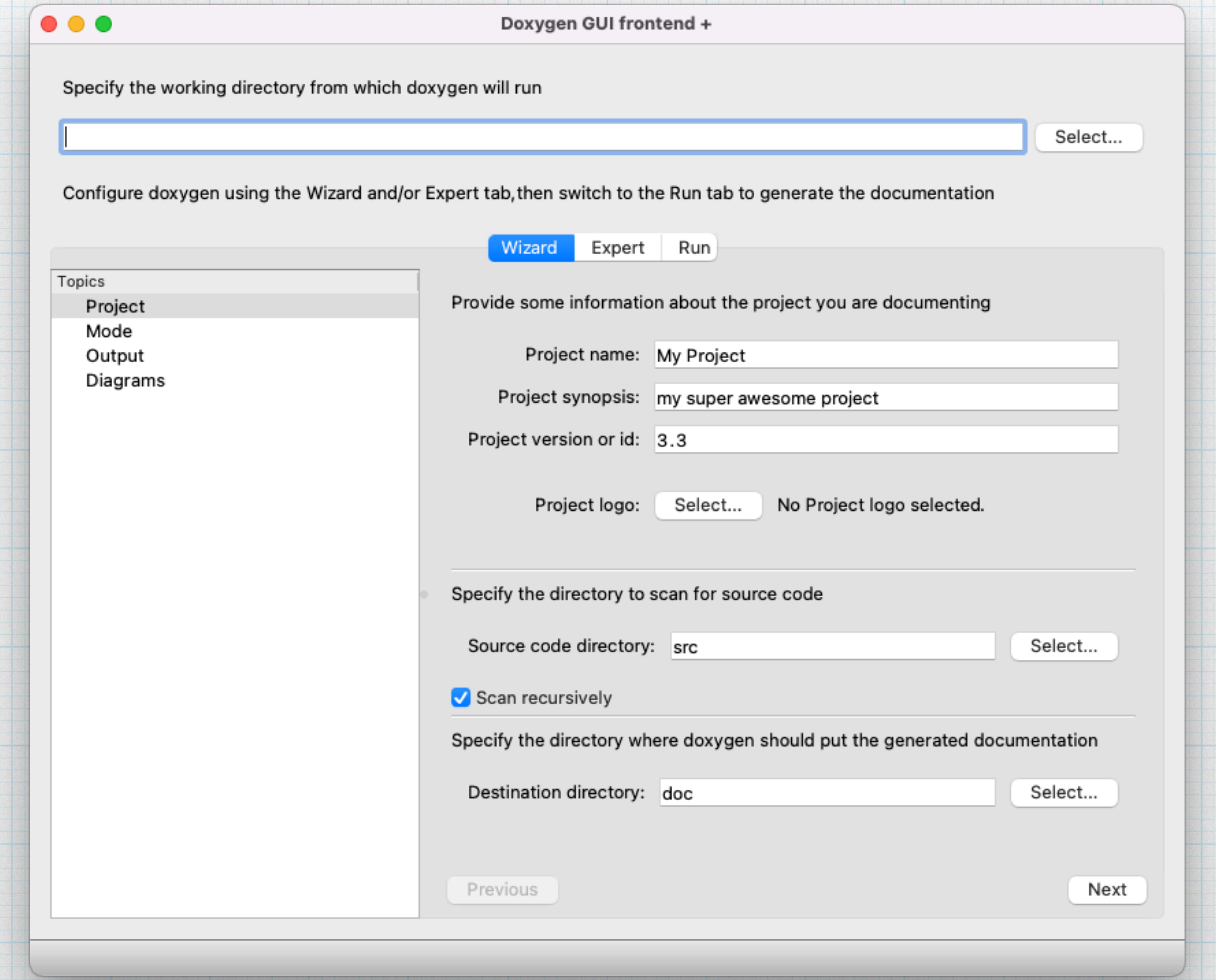
¿Qué incluir en un README?

- Descripción del problema que el proyecto resuelve
- Pequeños ejemplos de código
- Enlaces al gestor de issues o al repositorio de código
- Preguntas Frecuentes (FAQ)
- Información sobre cómo contribuir al proyecto
- Instrucciones de instalación
- La licencia del proyecto
- …

Doxygen


Doxywizard
generate/edit
Layout file
Config file
Doxyfile
generate
read
generate
update
Sources
read
Doxygen
read
Custom
– headers
– footers
– images
read
generate
Tag file(s)
XML files
read
Your application
doxmlparser lib
custom
output
Latex files
+
Makefile
make ps
make pdf
latex
postscript
PDF
Man pages
Windows only
refman.rtf
import
MS–Word
doc
HTML
pages
read
HTML Help Workshop
chm

Doxygen



Doxygen GUI frontend +

Specify the working directory from which doxygen will run

Configure doxygen using the Wizard and/or Expert tab,then switch to the Run tab to generate the documentation

Provide some information about the project you are documenting

Project name: My Project

Project synopsis: my super awesome project

Project version or id: 3.3

Project logo: No Project logo selected.

Specify the directory to scan for source code

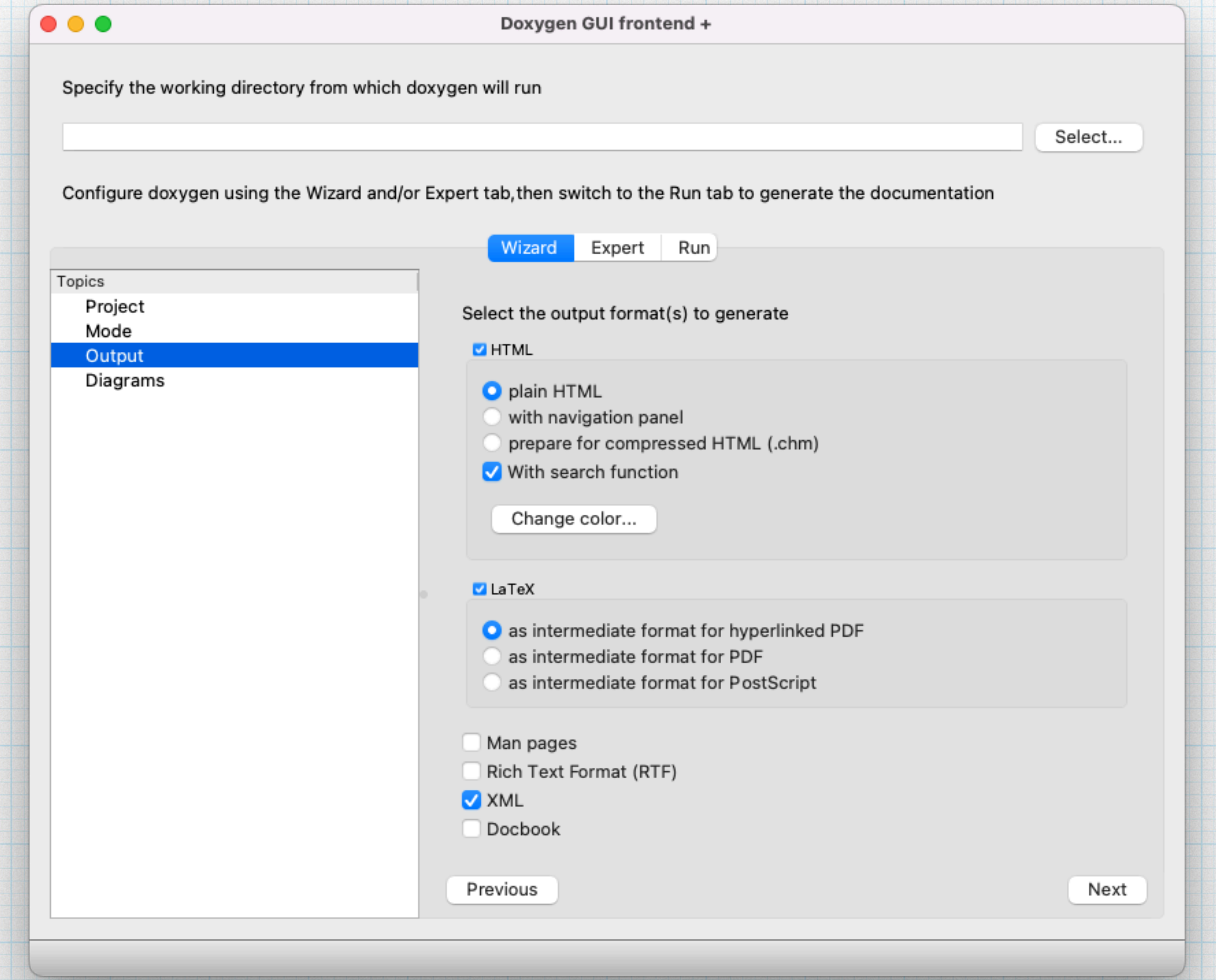
Source code directory: src

☑ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: doc

Doxygen



Specify the working directory from which doxygen will run

Configure doxygen using the Wizard and/or Expert tab,then switch to the Run tab to generate the documentation

Select the output format(s) to generate

☑ HTML

● plain HTML
○ with navigation panel
○ prepare for compressed HTML (.chm)
☑ With search function

☑ LaTeX

● as intermediate format for hyperlinked PDF
○ as intermediate format for PDF
○ as intermediate format for PostScript

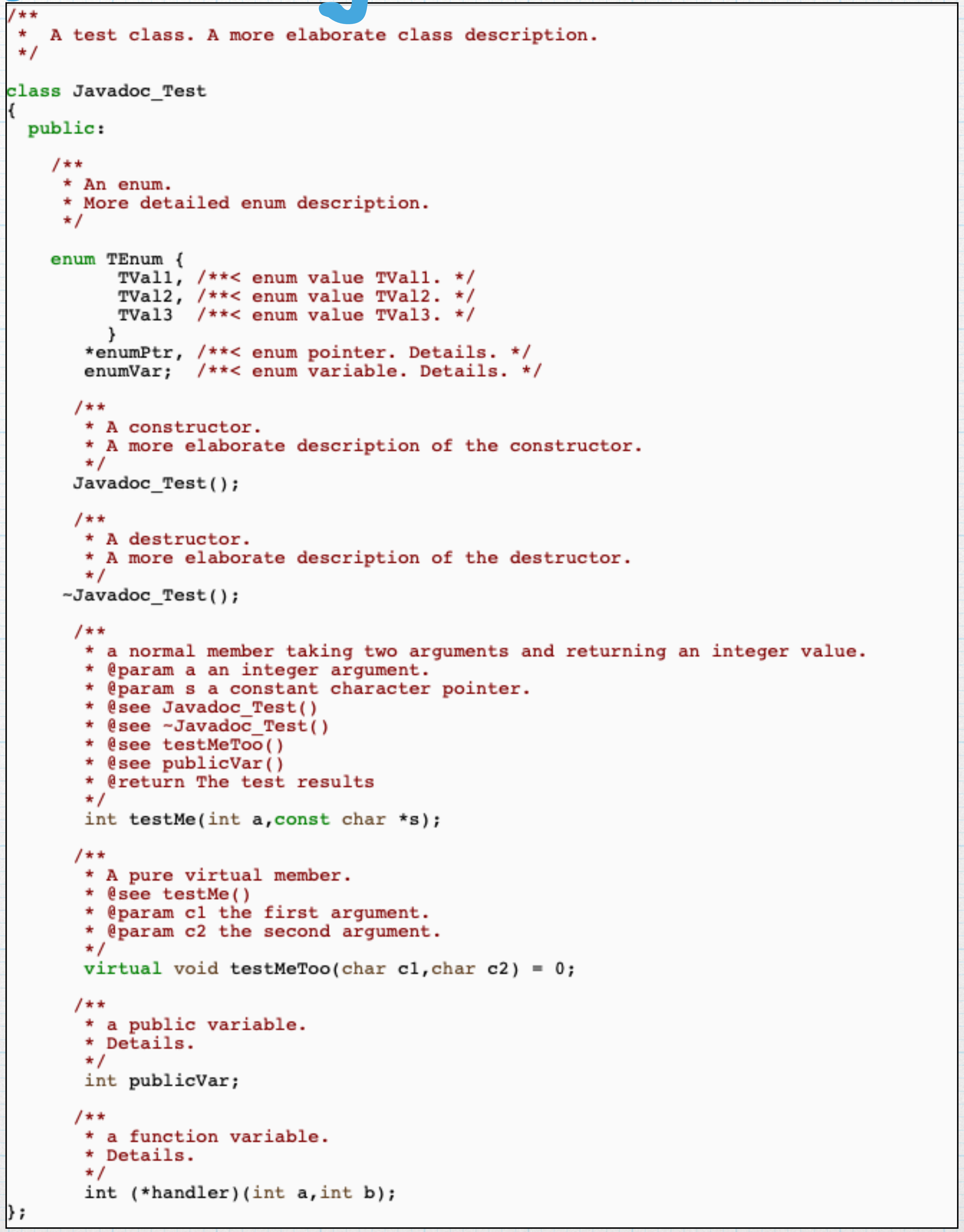
□ Man pages
□ Rich Text Format (RTF)
☑ XML
□ Docbook

Tipos de comentarios

- Comentarios de línea
- Comentarios descriptivos de bloque

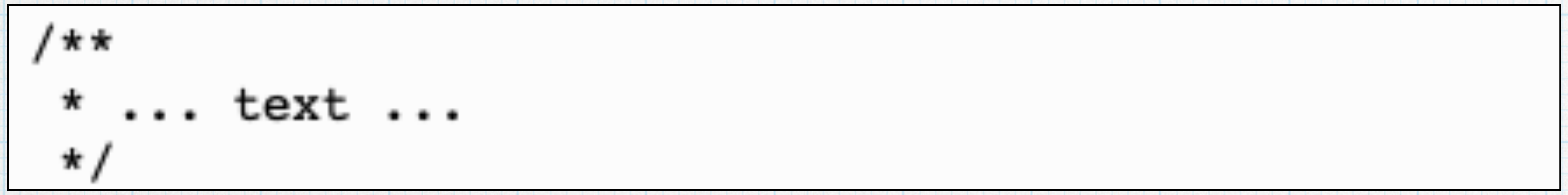
Documentando el código

(C/C++/C#/Objective-C/PHP/Java)



```
/**
 *  A test class. A more elaborate class description.
 */

class Javadoc_Test
{
  public:

    /**
     * An enum.
     * More detailed enum description.
     */

    enum TEnum {
          TVal1, /**< enum value TVal1. */
          TVal2, /**< enum value TVal2. */
          TVal3  /**< enum value TVal3. */
         }
       *enumPtr, /**< enum pointer. Details. */
       enumVar;  /**< enum variable. Details. */

      /**
       * A constructor.
       * A more elaborate description of the constructor.
       */
      Javadoc_Test();

      /**
       * A destructor.
       * A more elaborate description of the destructor.
       */
     ~Javadoc_Test();

      /**
       * a normal member taking two arguments and returning an integer value.
       * @param a an integer argument.
       * @param s a constant character pointer.
       * @see Javadoc_Test()
       * @see ~Javadoc_Test()
       * @see testMeToo()
       * @see publicVar()
       * @return The test results
       */
       int testMe(int a,const char *s);

      /**
       * A pure virtual member.
       * @see testMe()
       * @param c1 the first argument.
       * @param c2 the second argument.
       */
       virtual void testMeToo(char c1,char c2) = 0;

      /**
       * a public variable.
       * Details.
       */
       int publicVar;

      /**
       * a function variable.
       * Details.
       */
       int (*handler)(int a,int b);
};
```

Bloque de código

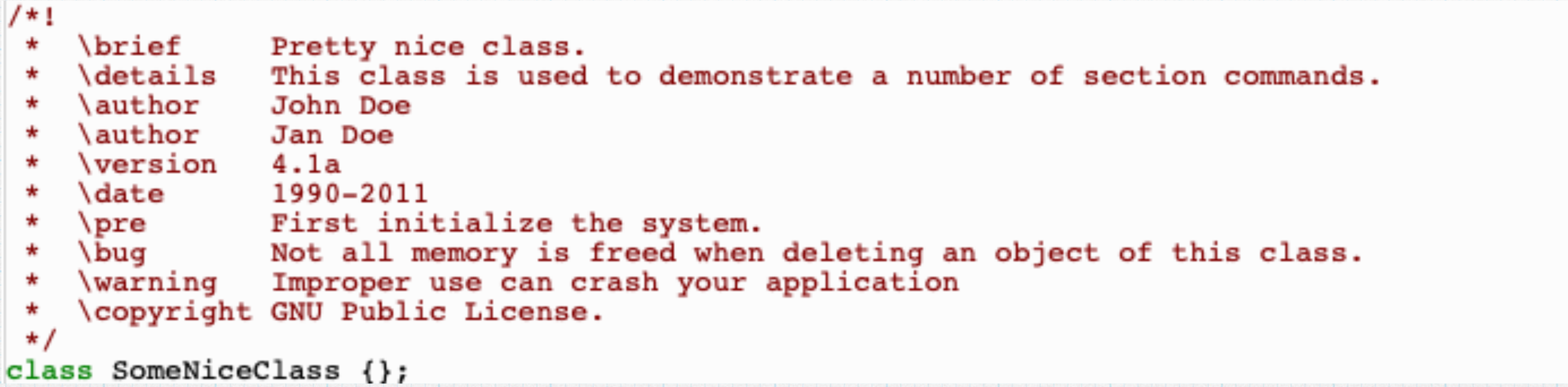


```
/**
 * ... text ...
 */
```

Documentando el código

(C/C++/C#/Objective-C/PHP/Java)

Cabecera de ficheros o clases



```
/*!
 *  \brief     Pretty nice class.
 *  \details   This class is used to demonstrate a number of section commands.
 *  \author    John Doe
 *  \author    Jan Doe
 *  \version   4.1a
 *  \date      1990-2011
 *  \pre       First initialize the system.
 *  \bug       Not all memory is freed when deleting an object of this class.
 *  \warning   Improper use can crash your application
 *  \copyright GNU Public License.
 */
class SomeNiceClass {};
```

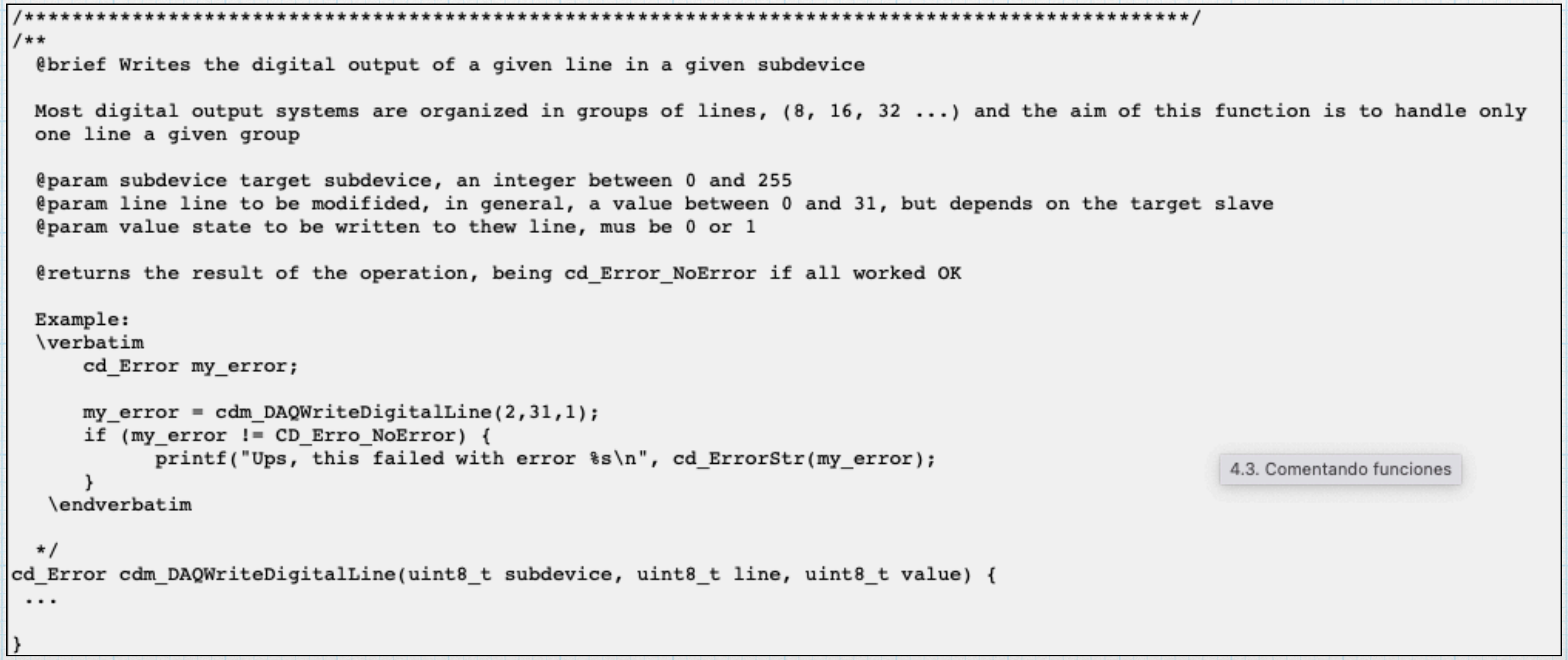
Documentando el código

(C/C++/C#/Objective-C/PHP/Java)

Comentando funciones

En una función es interesante dar una descripción de su cometido y, por supuesto, indicar qué parámetros de entrada necesita y qué devuelve como resultado. Para esta tarea se puede usar:

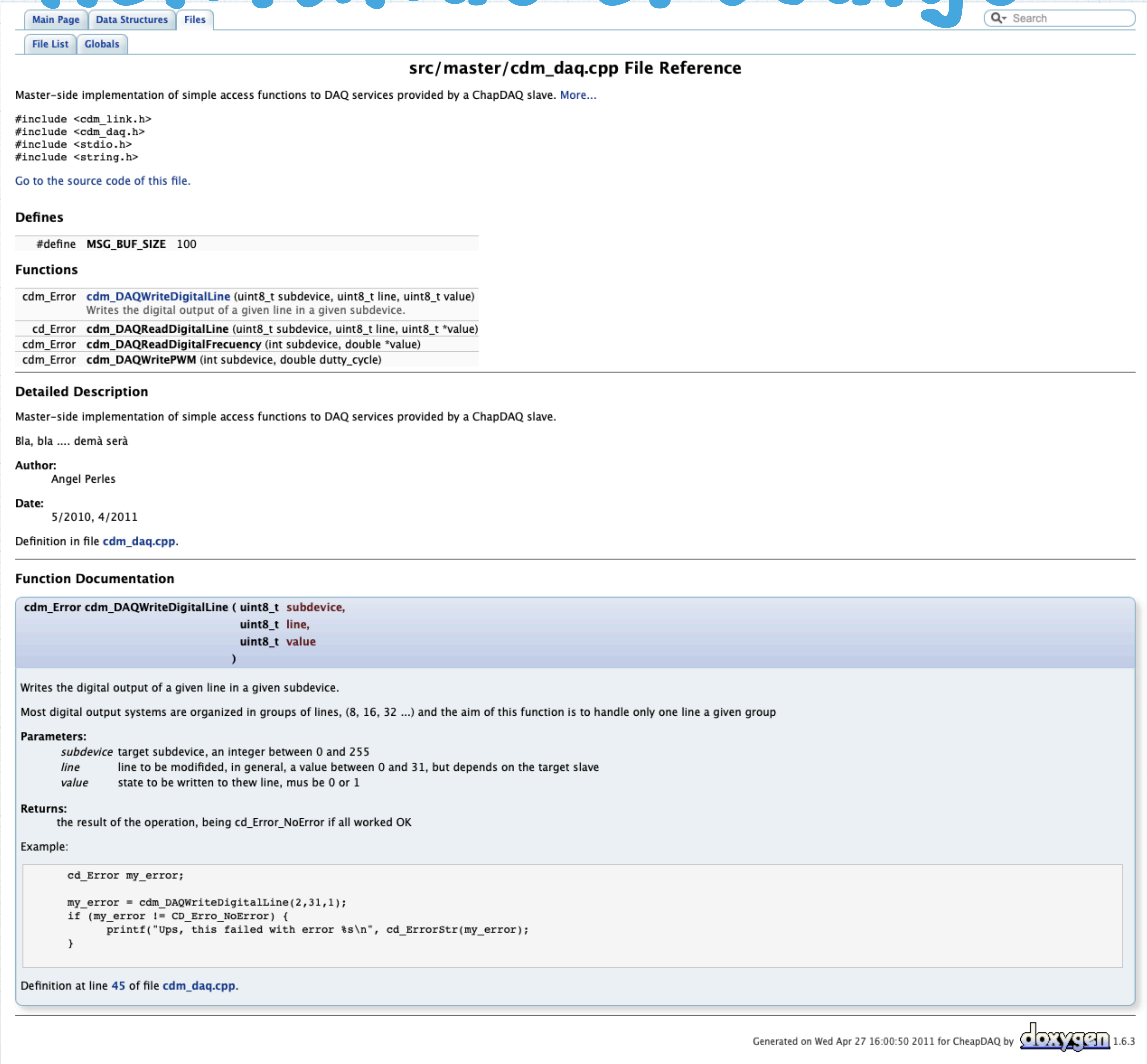
@brief descripción corta del cometido de la función
@param para indicar el cometido de un parámetro
@returns para indicar qué devuelve una función
@verbatim y @endverbatim para insertar un trozo de texto "literal", quizá un ejemplo de uso



```
/***********************************************************************************************/
/**
  @brief Writes the digital output of a given line in a given subdevice

  Most digital output systems are organized in groups of lines, (8, 16, 32 ...) and the aim of this function is to handle only
  one line a given group

  @param subdevice target subdevice, an integer between 0 and 255
  @param line line to be modified, in general, a value between 0 and 31, but depends on the target slave
  @param value state to be written to thew line, mus be 0 or 1

  @returns the result of the operation, being cd_Error_NoError if all worked OK

  Example:
  \verbatim
      cd_Error my_error;

      my_error = cdm_DAQWriteDigitalLine(2,31,1);
      if (my_error != CD_Erro_NoError) {
             printf("Ups, this failed with error %s\n", cd_ErrorStr(my_error);
      }
   \endverbatim

  */
cd_Error cdm_DAQWriteDigitalLine(uint8_t subdevice, uint8_t line, uint8_t value) {
 ...

}
```

Documentando el código

Resultado



src/master/cdm_daq.cpp File Reference

Master-side implementation of simple access functions to DAQ services provided by a ChapDAQ slave. More...

```
#include <cdm_link.h>
#include <cdm_daq.h>
#include <stdio.h>
#include <string.h>
```

Go to the source code of this file.

Defines

#define	**MSG_BUF_SIZE** 100

Functions

cdm_Error	cdm_DAQWriteDigitalLine (uint8_t subdevice, uint8_t line, uint8_t value) Writes the digital output of a given line in a given subdevice.
cd_Error	**cdm_DAQReadDigitalLine** (uint8_t subdevice, uint8_t line, uint8_t *value)
cdm_Error	**cdm_DAQReadDigitalFrecuency** (int subdevice, double *value)
cdm_Error	**cdm_DAQWritePWM** (int subdevice, double dutty_cycle)

Detailed Description

Master-side implementation of simple access functions to DAQ services provided by a ChapDAQ slave.

Bla, bla .... demà serà

Author:
Angel Perles

Date:
5/2010, 4/2011

Definition in file cdm_daq.cpp.

Function Documentation

```
cdm_Error cdm_DAQWriteDigitalLine ( uint8_t  subdevice,
                                    uint8_t  line,
                                    uint8_t  value
                                  )
```

Writes the digital output of a given line in a given subdevice.

Most digital output systems are organized in groups of lines, (8, 16, 32 ...) and the aim of this function is to handle only one line a given group

Parameters:

subdevice	target subdevice, an integer between 0 and 255
line	line to be modified, in general, a value between 0 and 31, but depends on the target slave
value	state to be written to thew line, mus be 0 or 1

Returns:
the result of the operation, being cd_Error_NoError if all worked OK

Example:

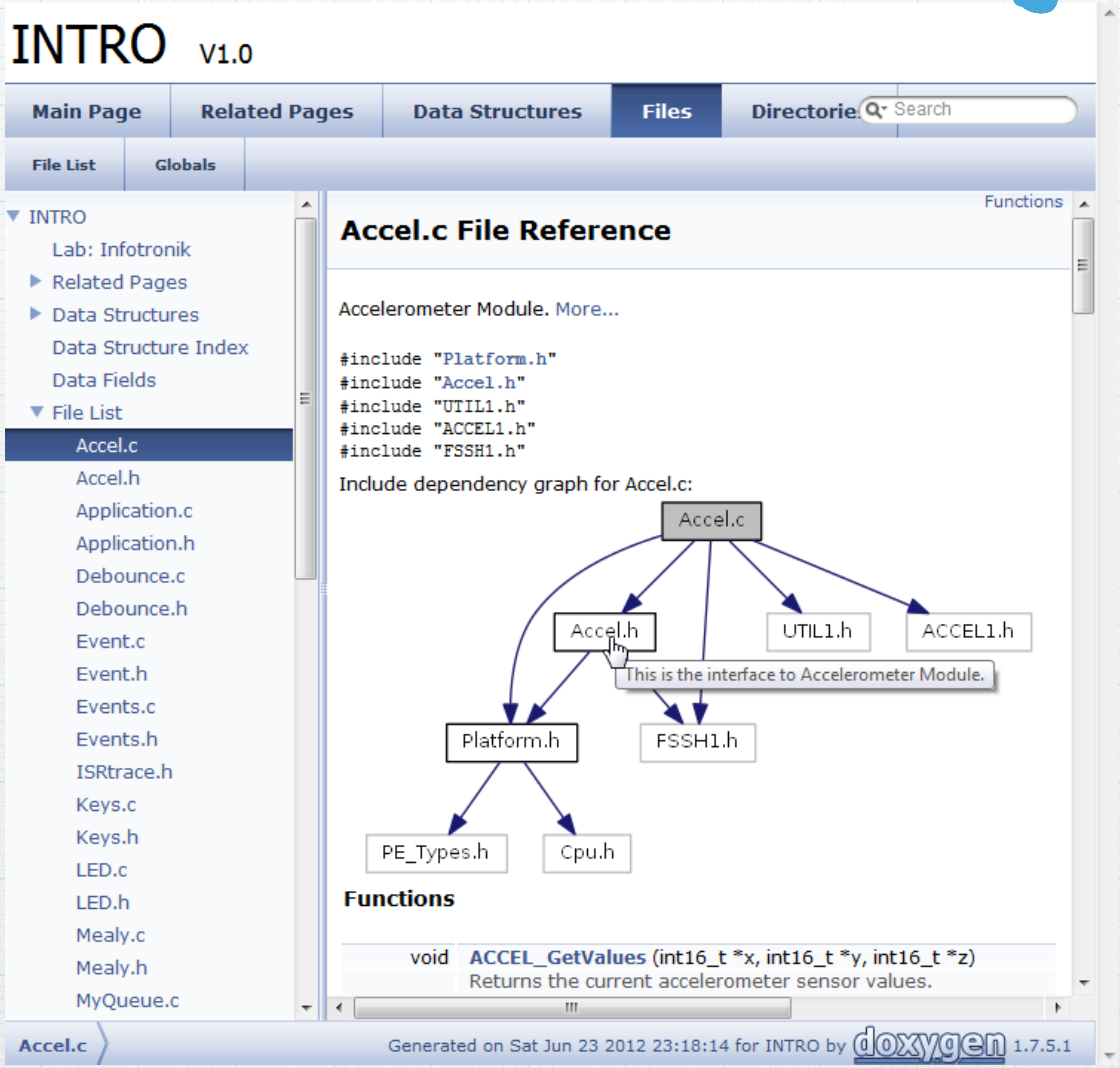
```
      cd_Error my_error;

      my_error = cdm_DAQWriteDigitalLine(2,31,1);
      if (my_error != CD_Erro_NoError) {
            printf("Ups, this failed with error %s\n", cd_ErrorStr(my_error);
      }
```

Definition at line 45 of file cdm_daq.cpp.

Generated on Wed Apr 27 16:00:50 2011 for CheapDAQ by doxygen 1.6.3

Documentando el código

Diagramas



Accel.c File Reference

Accelerometer Module. More...

```
#include "Platform.h"
#include "Accel.h"
#include "UTIL1.h"
#include "ACCEL1.h"
#include "FSSH1.h"
```

Include dependency graph for Accel.c:

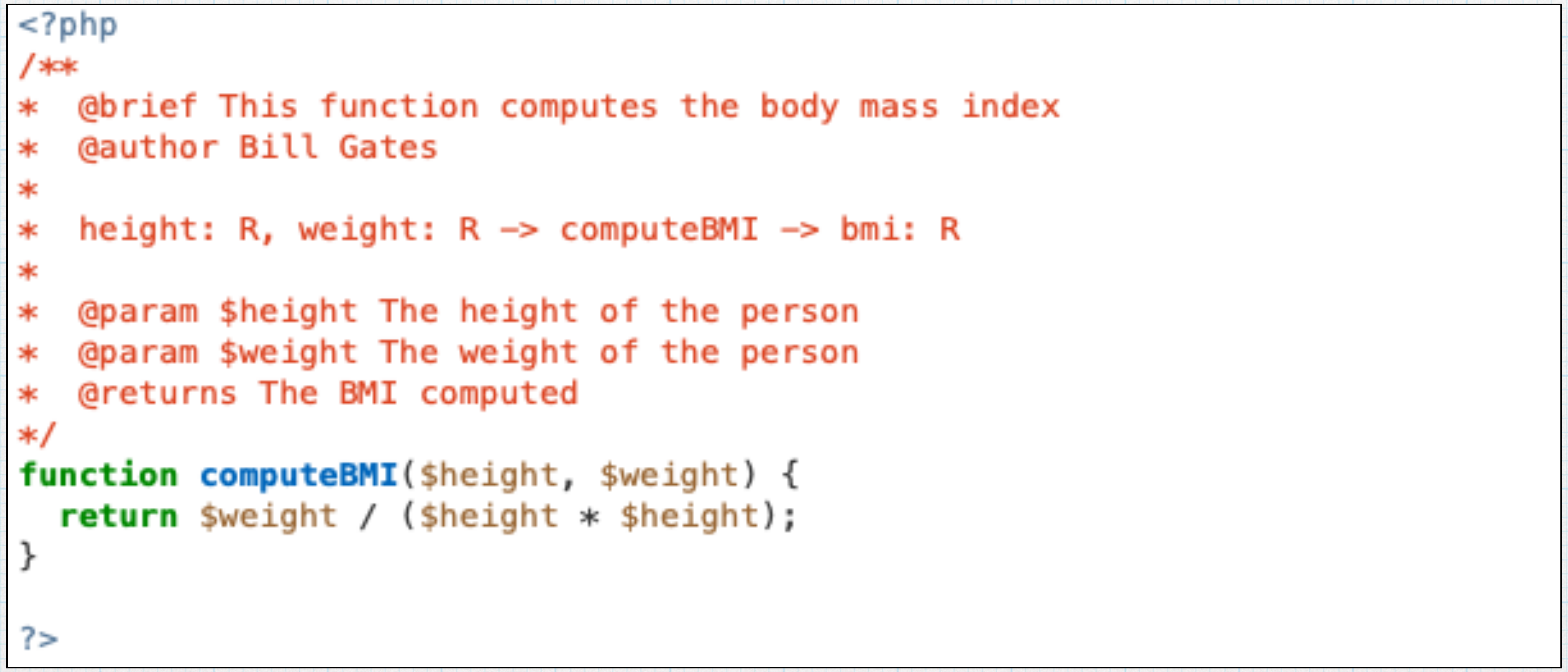
Functions

void **ACCEL_GetValues** (int16_t *x, int16_t *y, int16_t *z)
Returns the current accelerometer sensor values.

Accel.c — Generated on Sat Jun 23 2012 23:18:14 for INTRO by doxygen 1.7.5.1

Documentando el código

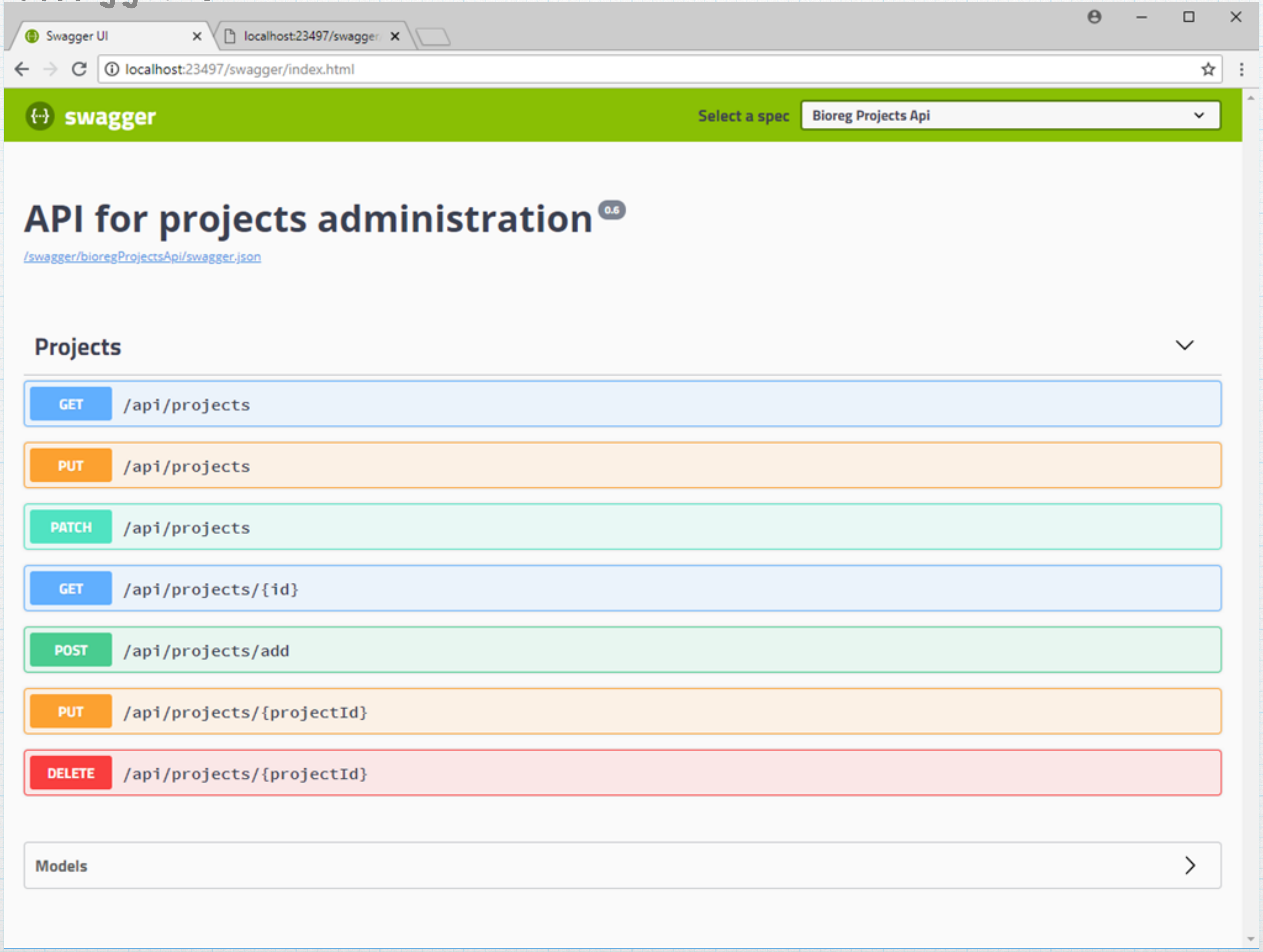
Se debe incluir el diseño!



```php
<?php
/**
*  @brief This function computes the body mass index
*  @author Bill Gates
*
*  height: R, weight: R -> computeBMI -> bmi: R
*
*  @param $height The height of the person
*  @param $weight The weight of the person
*  @returns The BMI computed
*/
function computeBMI($height, $weight) {
  return $weight / ($height * $height);
}

?>
```

Documentación API ReST

Swagger UI



API for projects administration 0.6

/swagger/bioregProjectsApi/swagger.json

Projects

GET /api/projects

PUT /api/projects

PATCH /api/projects

GET /api/projects/{id}

POST /api/projects/add

PUT /api/projects/{projectId}

DELETE /api/projects/{projectId}

Models

Recuerda!

Si tu código necesita un comentario para entenderse...
¡ cambia tu código !