

Documentación de código con Doxygen

Doxygen es una herramienta de documentación de código que se utiliza para generar documentación técnica a partir del código fuente de un programa. Se utiliza principalmente en proyectos de software para crear documentación legible y fácilmente navegable que describe el propósito, la funcionalidad, la estructura y el diseño del código.

Palomeque Nestor Levi

```
#include <stdio.h>
```

```
int main(){  
    int a = 10;  
    int b;  
    b ~= a;  
    return 0;  
}
```

Bloque de comentario

Hay varias formas de marcar un bloque de comentarios como descripción detallada:

```
/**  
 * ... texto ...  
 */
```

```
/*!  
 * ... texto ...  
 */
```

```
/*!  
 ... texto ...  
 */
```

```
///  
/// ... texto ...  
///
```

```
////////////////////////////////////  
/// ... texto ...  
////////////////////////////////////
```

```
/******  
 * ... texto  
*****/
```

Comentando archivos

La práctica común es realizar los comentarios en el archivo de encabezado (.h) en lugar del archivo de implementación (.c) en Doxygen, ya que el archivo de encabezado contiene las definiciones de las estructuras, funciones y variables que son visibles para otros archivos que incluyen el encabezado.

Ahora bien, para comentar un archivo cabecera llamado “*punto.h*”, arriba del mismo utilizaremos:

```
/**  
 * @file punto.h  
 * @brief Definición de la estructura Punto y sus funciones asociadas.  
 * @author Nestor Palomeque  
 * @version 1.0  
 * @date 2024-04-10  
 */
```

//Aquí sigue tú código

Comentando funciones

En una función es interesante dar una descripción de su cometido, indicar qué parámetros de entrada necesita y qué devuelve como resultado. Para ellos se pueden emplear las siguientes etiquetas:

- **@brief**: Proporciona una descripción corta y concisa del cometido de la función o del elemento documentado.
- **@param**: Utilizado para describir el cometido de un parámetro de función.
- **@returns**: Especifica qué valor devuelve una función.
- **@verbatim** y **@endverbatim**: Encierra un bloque de texto literal, como ejemplos de uso de una función o cualquier otro texto que no necesita ser formateado.

```
/**
 * @brief Calcula el promedio de una lista de números.
 * @param numeros Un array de números enteros.
 * @param longitud La longitud del array de números.
 * @returns El promedio de los números en el array.
 */
float calcularPromedio(int numeros[], int longitud) {
    float suma = 0.0;
    for (int i = 0; i < longitud; ++i) {
        suma += numeros[i];
    }
    return suma / longitud;
}
```

◆ calcularPromedio()

```
float calcularPromedio ( int numeros[],
                        int longitud )
```

Calcula el promedio de una lista de números.

Parámetros

numeros Un array de números enteros.

longitud La longitud del array de números.

Devuelve

El promedio de los números en el array.

Definición en la línea 7 del archivo `main.c`.

Comentando clases

A continuación veremos como comentar una clase y los atributos (o variables de la clase). Para comentar el constructor, destructor y las funciones miembros se emplea el mismo método comentado anteriormente para las funciones.

```
/**
 * @class Punto
 * @brief Clase que representa un punto en un plano cartesiano.
 */
class Punto {
public:
    /**
     * @brief Constructor de la clase Punto.
     * @param x Coordenada x del punto.
     * @param y Coordenada y del punto.
     */
    Punto(float x, float y);

    //.....Continúa con tu código.....

private:
    float x; /**< Coordenada x del punto. */
    float y; /**< Coordenada y del punto. */
};
```

```
/// @class Punto
/// @brief Clase que representa un punto en un plano cartesiano.

class Punto {
Public:

    /// @brief Constructor de la clase Punto.
    /// @param x Coordenada x del punto.
    /// @param y Coordenada y del punto.
    Punto(float x, float y);

    //.....Continúa con tu código.....

private:
    float x; /**< Coordenada x del punto.
    float y; /**< Coordenada y del punto.
};
```

Instalación

Para poder utilizar esta herramienta primero debemos realizar la instalación de los siguientes paquetes:

- Instala Doxygen:
sudo apt update
sudo apt install doxygen doxygen-gui
- Instala LaTeX:
sudo apt install doxygen-latex
- Graphviz (opcional):
sudo apt install graphviz

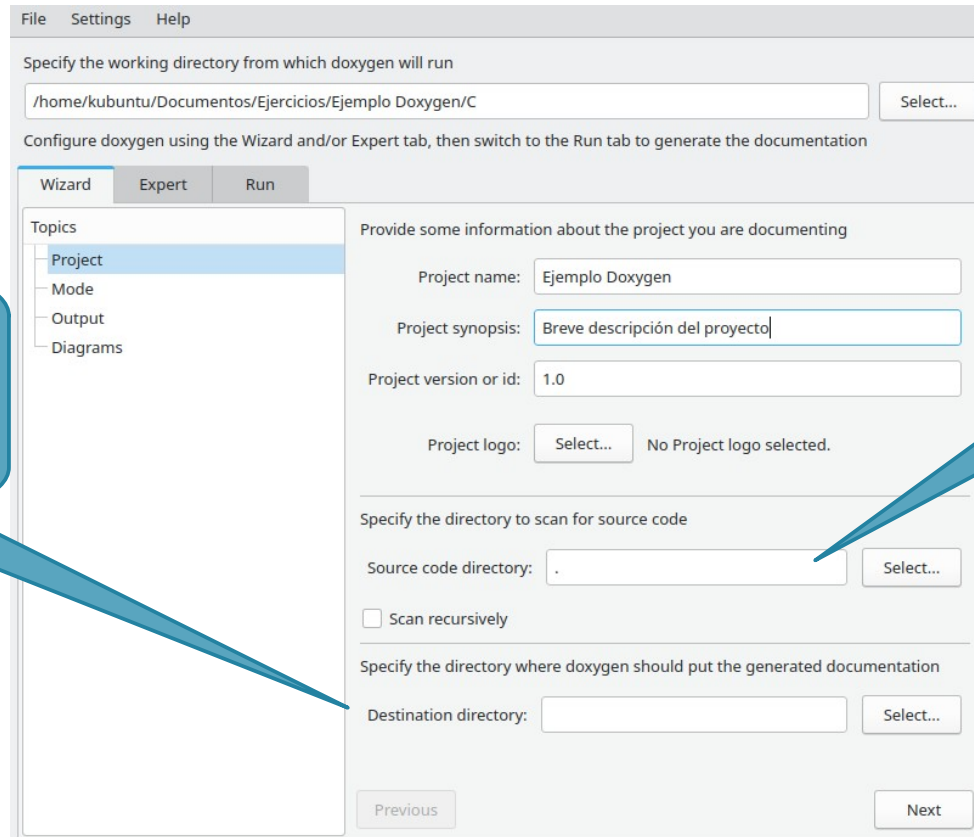
Ejecución desde consola

Para generar la documentación podemos realizarlo desde consola, para ellos debemos crear un archivo Doxifile y configurándolo mediante un editor de texto.

- Generar el archivo Doxyfile:
`doxygen -g`
- Luedo de editarlo haremos:
`doxygen`
- Si quisieramos darle un nombre al archivo Doxyfile:
`doxygen -g NombreDoxyFile`
`doxygen NombreDoxyFile`

Ejecución desde GUI

Si empleamos la interfaz gráfica debemos ejecutar “**doxywizard**” el cuál creará y editará el archivo Doxyfile.



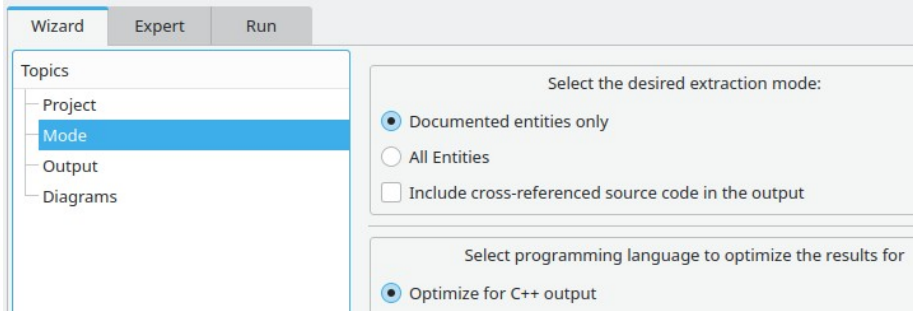
The screenshot shows the Doxywizard application window. At the top, there are menu items: File, Settings, and Help. Below the menu is a text field for the working directory, containing "/home/kubuntu/Documentos/Ejercicios/Ejemplo Doxygen/C", with a "Select..." button to its right. A message below reads: "Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation". The main area has three tabs: Wizard (selected), Expert, and Run. On the left of the Wizard tab is a "Topics" list with items: Project (selected), Mode, Output, and Diagrams. The main content area of the Wizard tab contains several sections: 1. "Provide some information about the project you are documenting" with fields for "Project name:" (Ejemplo Doxygen), "Project synopsis:" (Breve descripción del proyecto), "Project version or id:" (1.0), and "Project logo:" (a "Select..." button and the text "No Project logo selected."). 2. "Specify the directory to scan for source code" with a "Source code directory:" field (containing ".") and a "Select..." button, and a "Scan recursively" checkbox. 3. "Specify the directory where doxygen should put the generated documentation" with a "Destination directory:" field and a "Select..." button. At the bottom are "Previous" and "Next" buttons.

Directorio de salida

Código fuente en el directorio actual

Ejecución desde GUI

En “mode” debemos asegurarnos que esté optimizado para C++.



Wizard Expert Run

Topics

- Project
- Mode
- Output
- Diagrams

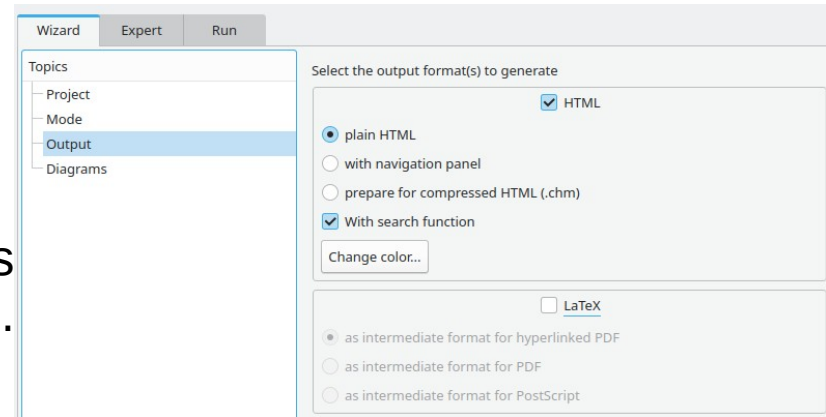
Select the desired extraction mode:

- ☒ Documented entities only
- ☐ All Entities
- ☐ Include cross-referenced source code in the output

Select programming language to optimize the results for

- ☒ Optimize for C++ output

En “output” seleccionaremos solo HTML.



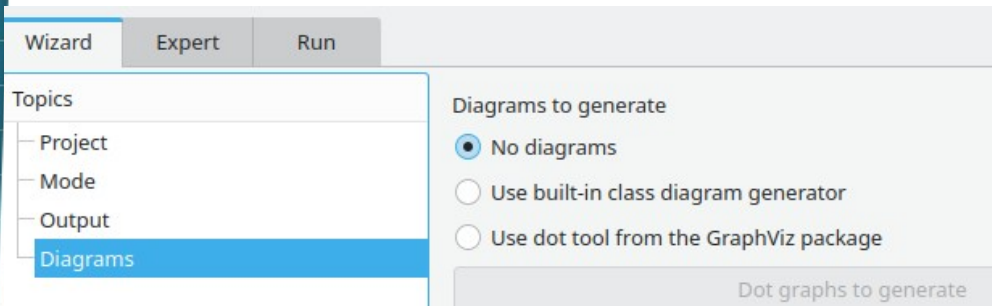
Wizard Expert Run

Topics

- Project
- Mode
- Output
- Diagrams

Select the output format(s) to generate

- ☒ HTML
- ☒ plain HTML
- ☐ with navigation panel
- ☐ prepare for compressed HTML (.chm)
- ☒ With search function
- Change color...
- ☐ LaTeX
- ☒ as intermediate format for hyperlinked PDF
- ☐ as intermediate format for PDF
- ☐ as intermediate format for PostScript



Wizard Expert Run

Topics

- Project
- Mode
- Output
- Diagrams

Diagrams to generate

- ☒ No diagrams
- ☐ Use built-in class diagram generator
- ☐ Use dot tool from the GraphViz package

Dot graphs to generate

En “diagrams” seleccionaremos para que no los cree, por ahora.

Ejecución desde GUI

Wizard Expert Run

Topics

- Project
- Build
- Messages
- Input
- Source Browser
- Index
- HTML
- LaTeX
- RTF

OUTPUT_DIRECTORY

The OUTPUT_DIRECTORY tag is used to specify the (relative or absolute) path into which the generated documentation will be written. If a relative path is entered, it will be relative to the location where doxygen was started. If left blank the current directory will be used.

DOXYFILE_ENCODING UTF-8

PROJECT_NAME Ejemplo Doxygen

PROJECT_NUMBER 1.0

PROJECT_BRIEF Breve descripción del proyecto

PROJECT_LOGO No Project logo selected.

OUTPUT_DIRECTORY

CREATE_SUBDIRS ☐

ALLOW_UNICODE_NAMES ☐

OUTPUT_LANGUAGE Spanish

OUTPUT_TEXT_DIRECTION None

BRIEF_MEMBER_DESC ☒

Previous Next

En "Expert" podemos seleccionar el idioma de la documentación

Por último debemos correr doxygen

Wizard Expert Run

Specify additional command line options for running doxygen

Run doxygen Status: not running

☐ Condensed Save log...

Show HTML output Show configuration

Output produced by doxygen

```
Generating directory documentation...
Generating index page...
Generating page index...
Generating module index...
Generating namespace index...
Generating namespace member index...
Generating annotated compound index...
Generating alphabetical compound index...
Generating hierarchical class index...
Generating member index...
Generating file index...
Generating file member index...
Generating example index...
finalizing index lists...
writing tag file...
Running plantuml with JAVA...
lookup cache used 10/65536 hits=44 misses=11
finished...

*** Doxygen has finished
```

