



Programming
language

Apreniendo a Documentar Código - Doxygen -

R1042

Curso: R1042

Docente: Ing. Nahuel Gonzalez

Ayudantes:

Pose, Fernando

Demski, Andrés

Spataro, Hector

Año 2015
V.1.0





Estructura de la Presentación:

- ☐ Buenas prácticas de programación.
- ☐ Introducción.
- ☐ Instalación.
- ☐ Configuración.
- ☐ Preparación del código a documentar.
- ☐ Generando documentación.
- ☐ Ejemplo.
- ☐ Referencias.





Buenas prácticas de programación (1)

1. El código debe ser escrito de manera **legible**:
 - ✓ Identar el código;
 - ✓ 35 líneas por función máximo.
 - ✓ Espaciar el código;
2. Llevar un código autodocumentado y **documentado**:
 - ✓ Usar nombres para las variables y funciones convenientes;
 - ✓ Realizar comentarios en el código (no en exceso);
 - ✓ Rotular los programas – Usar doxygen





Programming
language

Buenas prácticas de programación (2)

Documentar código ayuda a:

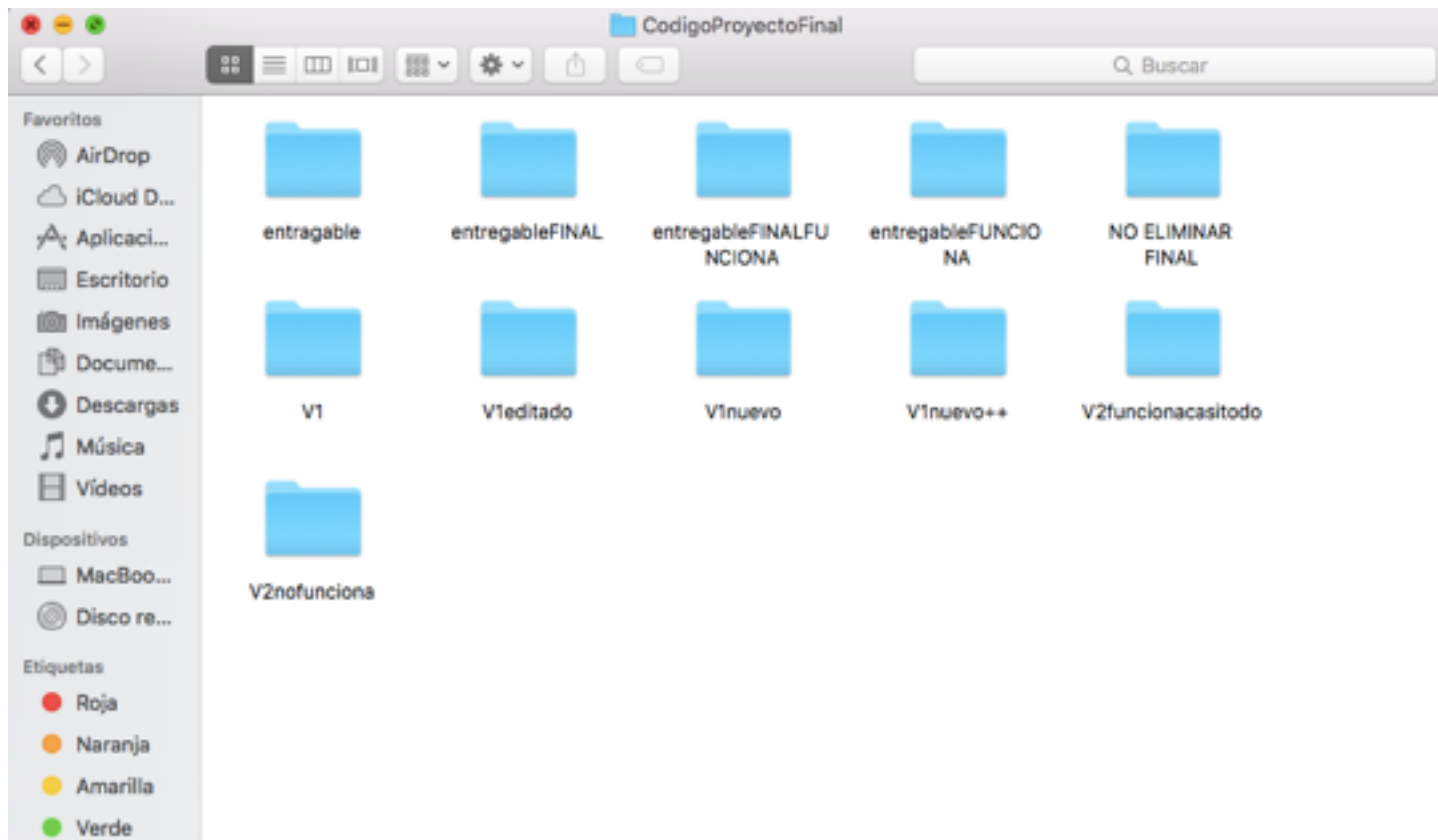
1. Reutilizar código;
2. Depurar código mas fácil;
3. Mantener los proyectos de forma organizada;
4. **Poder trabajar de forma pacífica con compañeros de trabajo.**





Programming
language

Buenas prácticas de programación (2)





Programming
language

Introducción (1)

Doxigen **NO** es

Una estructura de control en C

Una función de C

Algo que nos sirva para programar
(De forma directa)





Programming
language

Introducción (2)

Entonces..
¿Qué es Doxygen?

Es un generador de documentación para cualquier lenguaje de programación





Programming
language

Introducción (3)

Doxygen funciona en múltiples Sistemas Operativos





Introducción (4)

¿por qué usamos Doxygen?

1. Es requerimiento para algunos proyectos.
2. Es una pesadilla mantener los documentos del código cuando están separados.
3. Ayuda a entender código ajeno.
4. Permite generar documentación online en formato HTML
5. Es utilizado en proyectos importantes como los que se exponen a continuación.





Programming
language

Introducción (5)

¿Quién usa doxygen?





Instalación

1. Abrir la consola
2. Instalar los siguientes paquetes:

```
sudo apt-get install doxygen
```

```
sudo apt-get install graphviz
```





Programming
language

Configuración (1)

1. Abrir la consola.
2. Dirigirse hasta la raíz de los fuentes (donde se encuentran los archivos de trabajo)
3. Generamos el archivo de configuración para Doxygen.

```
doxygen -g
```

Esta crea un archivo llamado: Doxyfile cuyas características son:

1. Archivo de configuración de doxygen.
2. Archivo de texto (lo podemos abrir con gedit o kate)





Configuración (2)

Abrir el archivo Doxyfile y configurar los siguientes campos:

Parámetro	Descripción	Valor
PROJECT_NAME	Nombre del proyecto	xxx
PROJECT_NUMBER	Versión del proyecto	xxx
OUTPUT_DIRECTORY	Directorio prin. de archivos	./doxy





Programming
language

Configuración (3)

Parámetro	Descripción	Valor
CREATE_SUBDIRS	Dos niveles de subdirectorio	YES
OUTPUT_LANGUAGE	Lenguaje de documentación	English
TAB_SIZE	Espacios reemplazo de tab	4





Programming
language

Configuración (4)

Parámetro	Descripción	Valor
OPTIMIZE_OUTPUT_FOR_C	Optimización para leng. C	YES
EXTRACT_ALL	Contempla datos no docum.	YES
INPUT	Directorio del proyecto	./





Programming
language

Configuración (5)

Parámetro	Descripción	Valor
INPUT_ENCODING	Sist. de codif. de caracteres	UTF-8
GENERATE_LATEX	Generación de doc. Latex	NO
SOURCE_BROWSER	Código en la documentación	YES

Nota:

Si se trabaja en Windows el sistema de codificación de caracteres con el que se genera la documentación debe ser: ISO-8859





Preparación del código a documentar (1)

Documentando un archivo:

Se agrega en el principio del mismo el siguiente encabezado

```
/**  
    \file      <Nombre del archivo>  
    \brief     <Que contiene el archivo>  
    \author    <Apellido y Nombre (mail)>  
    \date      <Año.Mes.Día>  
    \version   <Versión (ejemplo: 1.0.0)>  
*/
```





Preparación del código a documentar (2)

Documentando una función:

Se agrega sobre el código de la función el siguiente encabezado

```
/**  
    \fn      <Prototipo de la función>  
    \brief   <Función de la función>  
    \author  <Apellido y Nombre (mail)>  
    \date    <Año.Mes.Día>  
    \param   <Parámetro A (si no tiene no se pone)>  
    \param   <Parámetro B (si no tiene no se pone)>  
    \return  <Lo que retorna>  
*/
```





Programming
language

Preparación del código a documentar (3)

Documentando una variable:

Se la puede documentar de forma detallada de la siguiente manera

```
/**  
    \var      <Tipo y nombre de la variable>  
    \brief    <Función de la variable>  
    \details  <Función mas detallada>  
*/
```





Programming
language

Preparación del código a documentar (4)

Documentando una estructura:

Se la puede documentar de forma detallada de la siguiente manera

```
/**  
    \struct  <Nombre de la estructura>  
    \brief   <Función de la estructura>  
    \details <Función mas detallada>  
*/
```

También se pueden documentar: enum, unión, class, typedef, entre otros.





Programming
language

Generación de documentación (1)

Una vez realizada la configuración del Doxyfile se procede a generar la documentación

1. Abrir la consola.
2. Dirigirse hasta la raíz de los fuentes (donde se encuentran los archivos de trabajo)
3. Típear:

```
doxygen Doxyfile
```





Programming
language

Generación de documentación (2)

Una vez generada la documentación abra el archivo:

`./doxy/html/index.html` con su browser



¡Navegue por la documentación!





Programming
language

Ejemplo

TPL1-A 1.0

Primer ejercicio del primer TP1

Página principal Archivada

Lista de archivos Archivos

TPL1-A TPL1-A TPL1-A-Aplicaciones

Referencia del Archivo TPL1-A-main.c

Archivo de entrada de la aplicación. Incluye ...

Incluye "instituto.h"

Dependencia gráfica adjunta para TPL1-A-main.c:

```
graph TD; A["TPL1-A/TPL1-A-Aplicaciones/TPL1-A-main.c"] --> B["Instituto.h"]; B --> C["Oscilador.h"]; B --> D["ReguLPC1000.h"];
```

Ir al código fuente de este archivo

Funciones

int main(void)





Programming
language

Referencias

Diapositivas basadas en:

1. Material de Andres Fontana.
2. Material del Ing. Nahuel Gonzalez.
3. Material del Ing. Jerónimo Atencio.
4. Material del Ing. Alejandro Furfaro.

Ejemplo de referencia

1. Svn -> Curso: R1042
2. Campus virtual -> Curso R1042 -> Material -> Códigos -> Doxygen





Programming
language

