

Capítulo 1

Doxygen: Haciendo Buena Documentación

1.1. Introducción.

1.1.1. Bloque de documentación.

Tipos de bloques:

```
/ **  
/ *!  
//!  
///
```

1.1.2. Compatibilidad con JavaDoc y QT.

1.1.3. Comandos especiales.

Todo lo que empieza con \ o @.

- param
- return
- brief

1.1.4. Descripción Breve y Detallada.

Para la descripción breve hay que poner un @brief a menos que se use JAVADOC_AUTOBRIEF.

1.1.5. Documentación en línea.

Documentacion detallada:

```
/ **<  
/ *!<
```

Documentacion breve:

```
///  
///  

```

Tambien puede ser usada como detallada si tiene 2 renglones.

1.1.6. Documentación fuera de línea.

Se puede documentar fuera de línea estos elementos:

```
@struct to document a C-struct.  
@union to document a union.  
@enum to document an enumeration type.  
@fn to document a function.  
@var to document a variable or typedef or enum value.  
@def to document a #define.  
@file to document a file.  
@namespace to document a namespace.  
@package to document a Java package.  
@interface to document an IDL interface.
```

1.1.7. Creación y opciones del archivo de configuración.

```
doxygen -g <config-file>
```

Mostrar doxywizard.

1.1.8. Formatos de salida.

- HTML
- LaTeX -> PDF
- RTF
- Man
- PerlMod
- CHM (o algo asi)

1.1.9. Lenguajes soportados.

Muchos, el español está :)

1.2. Mejorando la calidad de las descripciones.

1.2.1. Links y referencias cruzadas automáticos.

- Detecta URLs y e-mails.

- Detecta clases, funciones, variables y archivos. Formas de especificar scope:

```
funcion(args)
funcion()
::funcion
scope::funcion(args)
scope::funcion()
scope::funcion
```

Donde `scope` pueden ser namespaces y clases anidadas. Se puede usar `#` en vez de `::` para compatibilidad con JavaDoc.

Nota:

Mencionar el tema del scope (hay que poner `::` simbolo para referirse a un simbolo global).

Para funciones no sobrecargadas puede omitirse los argumentos.

Expande typedefs.

1.2.2. Referencias cruzadas manuales.

```
\\ref nombre ["texto"]
```

Si no se especifica el "texto", se pone el titulo de la seccion. Particularmente util para usar con paginas y secciones propias y para darle nombre a los links a clases y funciones.

1.2.3. Hipervínculos (sólo salida HTML).

Aprovechar para nombrar otros comandos HTML.

1.2.4. Incluir código de ejemplo.

Nombrar `@code` y `@example`.

1.2.5. Listas (con bullets o numeradas).

Nombrar:

- Listas comunes con bullets con `'-'`.
- Listas comunes numeradas con `'-#'`. Pueden ser anidadas indentando, para terminar. Para terminar una lista dejar una linea vacia o que tenga solo un punto:

```
- 1
- 1.1
- 1.2
.
Seguimos escribiendo en el item '1'.
- 2.
  Mas texto en el 2
.
Termina la lista y sigue el texto en el mismo párrafo.
```

Resultado:

- 1
 - 1.1
 - 1.2

Seguimos escribiendo en el item '1'.

- 2. Mas texto en el 2

Termina la lista y sigue el texto en el mismo párrafo. Nombrar listas en formato HTML.

1.2.6. Tablas.

Nombrar tablas en HTML.

1.2.7. Fórmulas matemáticas.

Hay formulas en embebidas en el texto:

```
La distancia entre \@f$ (x_1, y_1) \@f$ y
\@f$ (x_2, y_2) \@f$ es
\@f$ \sqrt{ (x_2 - x_1)^2 + (y_2 - y_1)^2 } \@f$
```

Resultado: La distancia entre (x_1, y_1) y (x_2, y_2) es $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ Y formulas en su propio renglón:

```
\@f[ \sqrt{ (x_2 - x_1)^2 + (y_2 - y_1)^2 } \@f]
```

Resultado:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Nota:

Ver como engancharlo con lo que explique Diego.

1.2.8. Agregar imágenes.

```
\@image format file ["caption"] [sizeindication=size]
```

Ejemplo:

```
\@image html diagrama.jpg
\@image latex diagrama.eps "Diagrama de lala" width=10cm
```

Nota:

Las imagenes deben estar en el IMAGE_PATH.

1.3. Características avanzadas.

1.3.1. Páginas relacionadas.

Nombrar bug, todo, example, deprecated.

1.3.2. Páginas propias.

Nombrar `page`, `section`, `subsection`, etc.

1.3.3. Contenido selectivo.

Nombrar `htmlonly`, `latexonly`, `if`, `internal`, etc.

1.3.4. Pulir los resultados.

Mostrar un `refman.tex` y retocarlo. Mostrar PlaQui.

1.4. Preguntas.

1.5. FIN