

Introducción a la herramienta doxygen

Doxygen es una herramienta que permite generar documentación con formato a partir de los comentarios incluidos en el propio código fuente. La herramienta soporta diferentes lenguajes de programación como Fortran, C, C++, Java, VHDL, PHP, C# o Python, entre los más populares.

La inclusión de comentarios que pueden ser procesados por *doxygen* se realiza utilizando una sintaxis especial.

A continuación se muestra un ejemplo de un comentario en una cabecera de un fichero .h

```
/**
 * @brief Utilidades de manejo de caracteres
 *
 * Este modulo contiene los prototipos de las funciones de manejo de
 * caracteres.
 * @file utilcad.h
 * @author Nombre Apellido
 * @version 1.0
 * @date 25-01-2010
 */
```

Los comentarios que son interpretados por *doxygen* tienen que comenzar con `/**` y terminar con `*/`. Las etiquetas comienzan con `@`, y permiten definir campos dentro de los comentarios que serán tratados de manera especial por la herramienta.

Las funciones siguen una sintaxis similar:

```
/**
 * @brief compara dos cadenas de caracteres
 *
 * strcmp() compara lexicograficamente dos cadenas de
 * caracteres s1 y s2.
 * @param s1 cadena de caracteres correctamente formada.
 * @param s2 cadena de caracteres correctamente formada.
 * @return entero mayor, igual, o menor que 0 dependiendo de si
 * s1 es mayor, igual o menor que s2
 */
int strcmp(const char *s1, const char *s2);
```

La etiqueta `@brief` se emplea al comienzo de un módulo y permite definir una frase resumen del mismo. La etiqueta `@param` permite incluir comentarios para los parámetros de la función, mientras que la etiqueta `@return` permite describir los valores de retorno.

Constantes, variables globales, enumerados y estructuras se comentan de manera ligeramente distinta:

```
const double NA = 6.02214179E23;          /*!< Numero de Avogrado */

/**
 * @brief Lista de posibles errores
 */
enum Errores {
    OK,          /*!< No hay error */
```

```

        ERROR_MEM,    /*!< Error en la reserva de memoria */
        ERROR_FICH    /*!< Error al abrir el fichero */
};

/**
 * @brief Punto en 2D
 *
 * Esta estructura define un punto en dos dimensiones.
 */
typedef struct
{
    float x;          /*!< Coordenada x */
    float y;          /*!< Coordenada y */
} Punto;

```

Nótese que los campos individuales se comentan con `/*!< ... */`. Hay que tener en cuenta que *doxygen* dispone de varios estilos de sintaxis habiéndose elegido la que se muestra en los ejemplos por su similitud con la documentación *javadoc* de Java.

Ejecución

Antes de ejecutar *doxygen* se recomienda crear el fichero de configuración *Doxyfile*:

```
> doxygen -g
```

Para ejecutar *doxygen*:

```
> doxygen Doxyfile
```

Uso de doxygen

La herramienta *doxygen* incluye un conjunto bastante amplio de etiquetas¹. Ahora bien, como se ha podido comprobar, la inclusión de estas etiquetas reduce la legibilidad del código fuente, por lo que se recomienda su uso moderado. Las etiquetas incluidas en este documento están dentro de las más habituales².

Las partes del código fuente que tiene sentido comentar empleando la sintaxis de *doxygen* siempre van a ser las partes públicas, es decir, aquellas que se corresponda con ficheros `.h`. Queda desaconsejado su uso en el resto del código. En especial, dentro del código de las funciones.

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

² En algunos proyectos `@date` y `@version` se omiten ya que existen herramientas de control de versiones que generan automáticamente esta información.