8.B. Documentación del código.

1. Documentación del código.

1.2. Uso de las etiquetas.

¿Cuáles son las etiquetas típicas y su significado? Pasaremos seguidamente a enumerar y describir la función de las etiquetas más habituales a la hora de generar comentarios de documentación.

- `@autor texto con el nombre`: Esta etiqueta sólo se admite en clases e interfaces. El texto después de la etiqueta no necesitará un formato especial. Podremos incluir tantas etiquetas de este tipo como necesitemos.

- `@version texto de la versión`: El texto de la versión no necesitará un formato especial. Es conveniente incluir el número de la versión y la fecha de ésta. Podremos incluir varias etiquetas de este tipo una detrás de otra.

- `@deprecated texto`: Indica que no debería utilizarse, indicando en el texto las causas de ello. Se puede utilizar en todos los apartados de la documentación. Si se ha realizado una sustitución debería indicarse qué utilizar en su lugar. Por ejemplo:

```
@deprecated El método no funciona correctamente. Se recomienda el uso de {@ling metodoCorrecto}
```

```
```

```
```

- `@exception nombre-excepción texto`: Esta etiqueta es equivalente a @throws.

- `@param nombre-atributo texto`: Esta etiqueta es aplicable a parámetros de constructores y métodos. Describe los parámetros del constructor o método. Nombre-atributo es idéntico al nombre del parámetro. Cada etiqueta @param irá seguida del nombre del parámetro y después de una descripción de éste. Por ejemplo:

```
@param fromIndex: El índice del primer elemento que debe ser eliminado.
```

```
```

```
```

- `@return texto`: Esta etiqueta se puede omitir en los métodos que devuelven void. Deberá aparecer en todos los métodos, dejando explícito qué tipo o clase de valor devuelve y sus posibles rangos de valores. Veamos un ejemplo:

```
/**
```

```
* Chequea si un vector no contiene elementos.
```

```
*
```

```
* @return <code>verdadero</code>si solo si este vector no contiene componentes, esto es, su tamaño es cero;
```

```
* <code>falso</code> en cualquier otro caso.
```

```
*/
```

```
public boolean VectorVacio() {
```

```
return elementCount == 0;
```

```
}
```

```

```

```

```

- `@see referencia`: Se aplica a clases, interfaces, constructores, métodos, atributos y paquetes. Añade enlaces de referencia a otras partes de la documentación. Podremos añadir a la etiqueta: cadenas de caracteres, enlaces HTML a páginas y a otras zonas del código. Por ejemplo:

```
* @see "Diseño de patrones: La reusabilidad de los elementos de la programación orientada a objetos"
```

```
* @see <a href="http://www.w3.org/WAI/">Web Accessibility Initiative</a>
```

```
* @see String#equals(Object) equals
```

```

```

```

```

- `@throws nombre-excepción texto`: En nombre-excepción tendremos que indicar el nombre completo de ésta. Podremos añadir una etiqueta por cada excepción que se lance explícitamente con una cláusula `throws`, pero siguiendo el orden alfabético. Esta etiquetas es aplicable a constructores y métodos, describiendo las posibles excepciones del constructor/método.

Autoevaluación

¿Qué etiqueta podría omitirse en un método que devuelve `void`?

- ○ @param.
- ○ @throws.
- ○ @return.