

1. Que es la documentación de código en Java

Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, punto por punto, de forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué.

2. Como documentar una clase

```
/**
 * Clase Empleado
 *
 * Contiene informacion de cada empleado
 *
 * @author Fernando
 * @version 1.0
 */
public class Empleado {

    //Atributos

    /**
     * Nombre del empleado
     */
    private String nombre;
    /**
     * Apellido del empleado
     */
    private String apellido;
    /**
     * Edad del empleado
     */
    private int edad;
    /**
     * Salario del empleado
     */
    private double salario;
```

Como documentar un constructor

```
3. //Constructores
4. /**
5.  * Constructor por defecto
6.  */
7. public Empleado(){
8.     this("", "", 0, 0);
9. }
```

```

10.
11.  /**
12.   * Constructor con 4 parametros
13.   * @param nombre nombre del empleado
14.   * @param apellido nombre del empleado
15.   * @param edad edad del empleado
16.   * @param salario salario del empleado
17.   */
18.  public Empleado(String nombre, String apellido, int edad, double salario){
19.      this.nombre=nombre;
20.      this.apellido=apellido;
21.      this.edad=edad;
22.      this.salario=salario;
23.  }
24. }

```

4. Como documentar un método

//Metodos publicos

```

/**
 * Suma un plus al salario del empleado si el empleado tiene mas de 40 años
 * @param sueldoPlus
 * @return <ul>
 *     <li>true: se suma el plus al sueldo</li>
 *     <li>false: no se suma el plus al sueldo</li>
 * </ul>
 */
public boolean plus (double sueldoPlus){
    boolean aumento=false;
    if (edad>40 && compruebaNombre()){
        salario+=sueldoPlus;
        aumento=true;
    }
    return aumento;
}

```

//Metodos privados

```

/**
 * Comprueba que el nombre no este vacio
 * @return <ul>
 *     <li>true: el nombre es una cadena vacia</li>
 *     <li>false: el nombre no es una cadena vacia</li>
 * </ul>
 */
private boolean compruebaNombre(){
    if(nombre.equals("")){

```

```

return false;
    }
    return true;
}

```

5. Que son los comandos @param y @return

Las etiquetas @author y @version se usan para documentar clases e interfaces. Por tanto no son válidas en cabecera de constructores ni métodos. La etiqueta @param se usa para documentar constructores y métodos. La etiqueta @return se usa solo en métodos de tipo función.

6. Realizar un ejemplo en Netbeans de la documentación de una clase, constructor y método

```

package miniCalculator;
/**
 *Calcula el valorAbsoluto y raiz cuadrada de un valor.
 * @version 1.00.0
 * @author Daniels
 */
public class miniCalculadora {
    /**
     * Constructor de la clase miniCalculadora
     */
    public miniCalculadora(){}
    /**
     * Calcula el valor absoluto de un numero.
     * @param numeroAbs: Recibe numero para sacar valor absoluto.
     * @return: Devuelve valor absoluto.
     */
    public double valorAbsoluto(double numeroAbs){
        return Math.abs(numeroAbs);
    }
    /**
     * Calcula la raiz cuadrada de un numero
     * @param num: Recibe numero para sacar raiz cuadrada.
     * @return: Devuelve la raiz cuadrada del numero recibido.
     */
    public double raizCuadrada(double num){
        return Math.sqrt(num);
    }
}

```