

Entornos de Desarrollo

UT4 - Práctica de Optimización y Generación de Javadoc

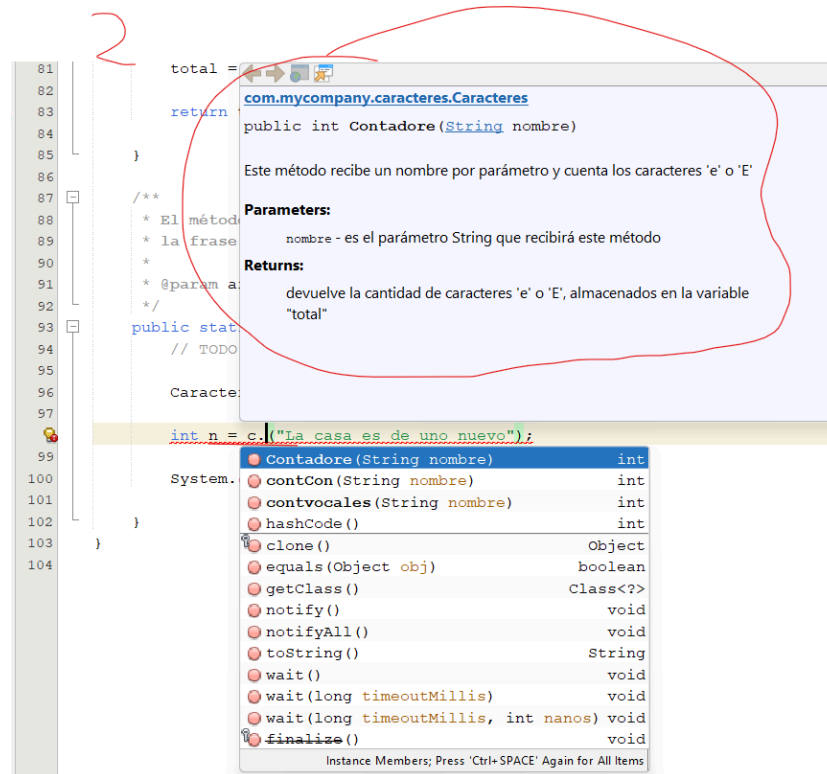
En esta actividad aprenderemos a generar la documentación Javadoc en nuestros proyectos de Java para proporcionar descripciones a los métodos, tanto en NetBeans como en Eclipse.

Parte 1: NetBeans

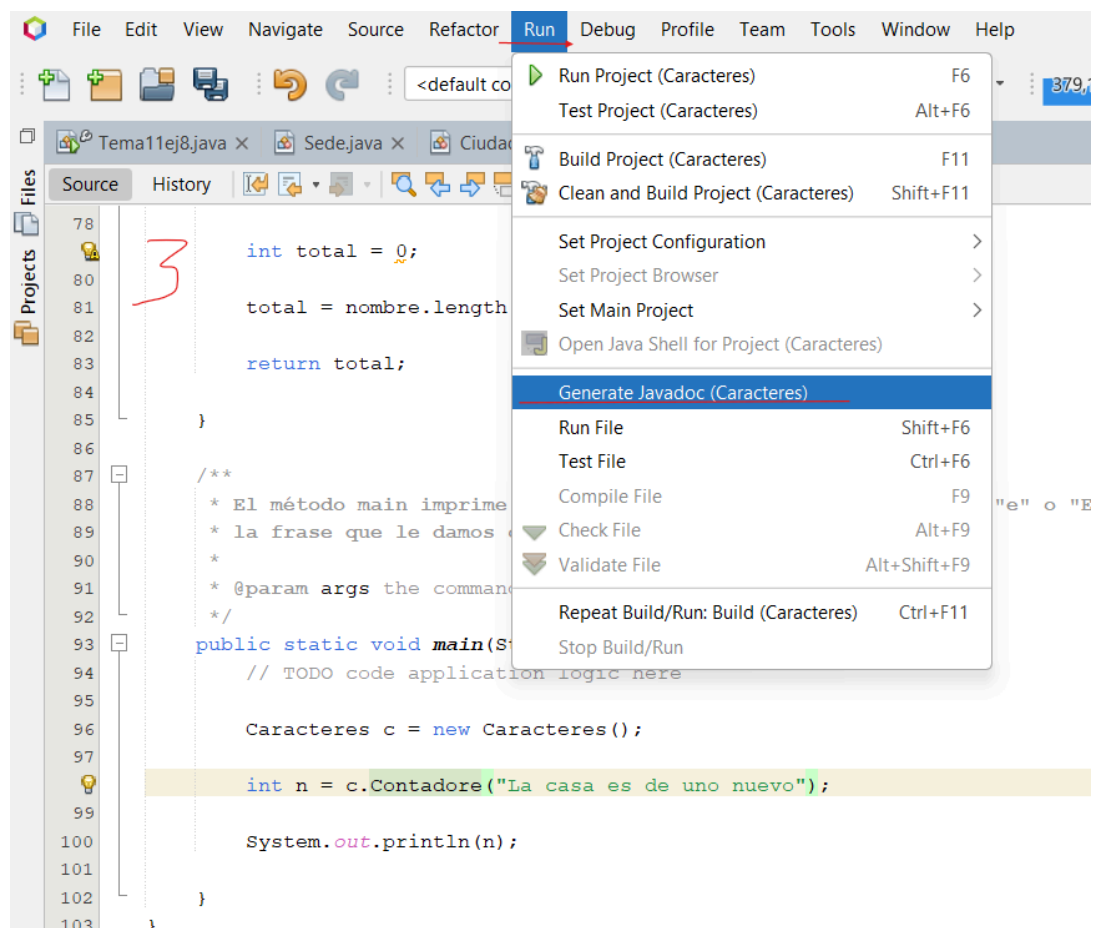
1) Lo primero que debemos hacer es escribir la documentación encima de cada método. Usamos `/**` para comenzar el modo comentario y `*/` para cerrarlo. De forma automática se nos generará el `@` en los parámetros y en el return para que escribamos una descripción.

```
public class Caracteres {  
  
    /**  
     * Este método recibe un nombre por parámetro y cuenta los caracteres 'e' o  
     * 'E'  
     *  
     * @param nombre es el parámetro String que recibirá este método  
     * @return devuelve la cantidad de caracteres 'e' o 'E', almacenados en la  
     * variable "total"  
     */  
    public int Contadore(String nombre) {  
        int total = 0;  
  
        int i = 0;  
  
        while (i < nombre.length()) {  
            if ((nombre.charAt(i) == 'e') || (nombre.charAt(i) == 'E')) {  
  
                total++;  
  
            }  
  
            i++;  
        }  
    }  
}
```

2) Si llamamos al método "Contadore" veremos que la documentación que hemos escrito aparece.



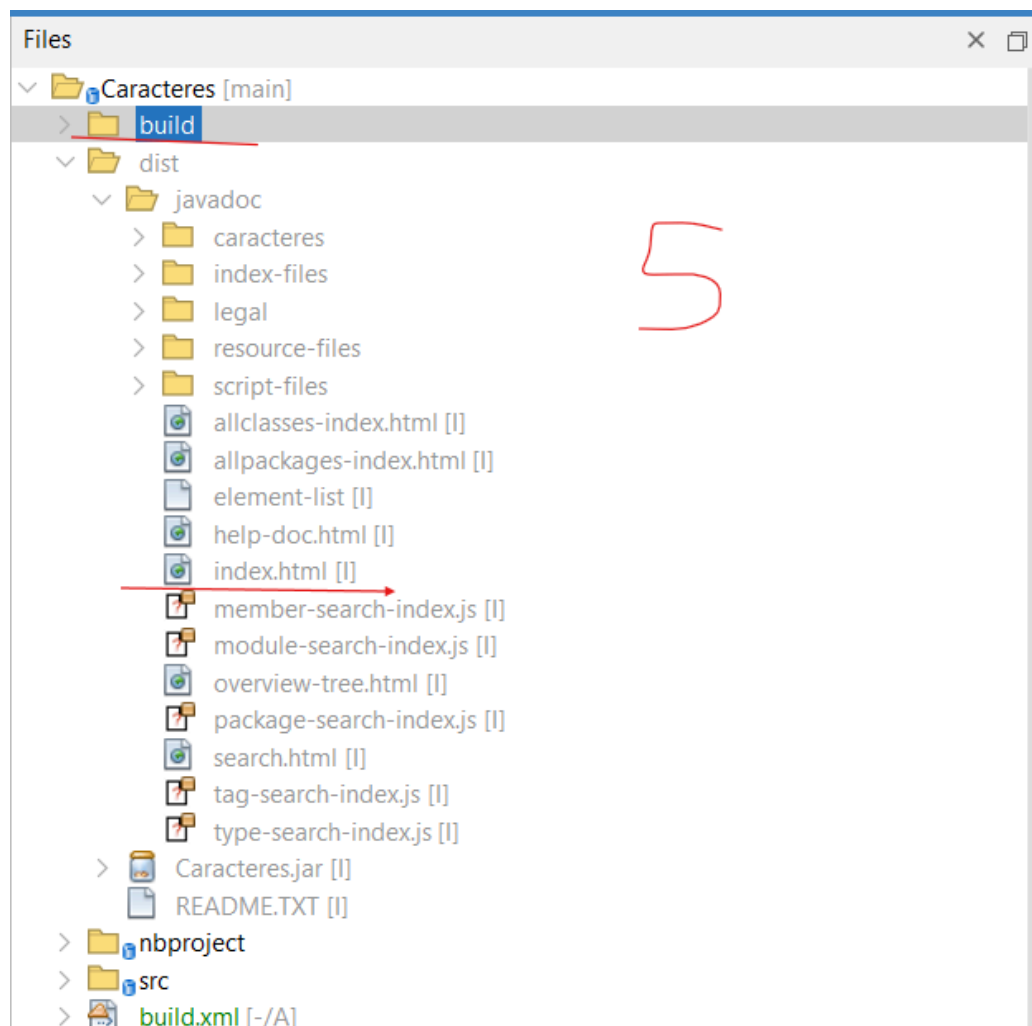
3) Hacemos click en Run, y después en Generate Javadoc



- 4) El navegador se abre, mostrándonos un menú con nuestro javadoc. Si usamos Maven, es posible que haya problemas que solucionar con el archivo `index.html` y `pom.xml`. En Ant no debería darnos problemas:

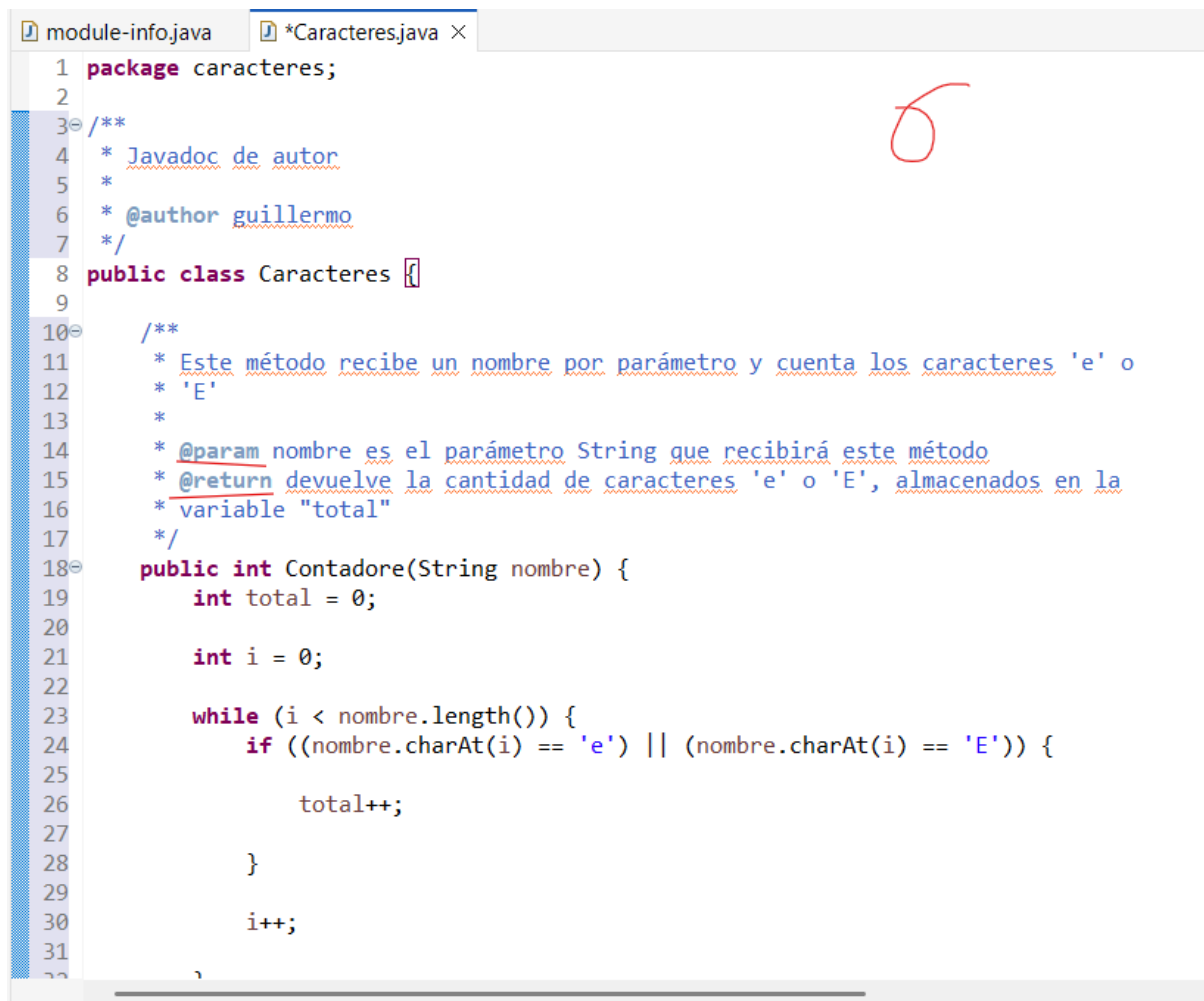


- 5) En cualquier caso, deberíamos encontrar el archivo ***index.html*** en la carpeta ***build/dist/javadoc***:



Parte 2: Eclipse

6) Igual que con NetBeans, escribimos los comentarios usando `/**` y `*/` junto con los `@` para describir lo que estimemos pertinente (en este caso los parámetros y el return, aunque al escribir la `@` veremos más posibilidades)

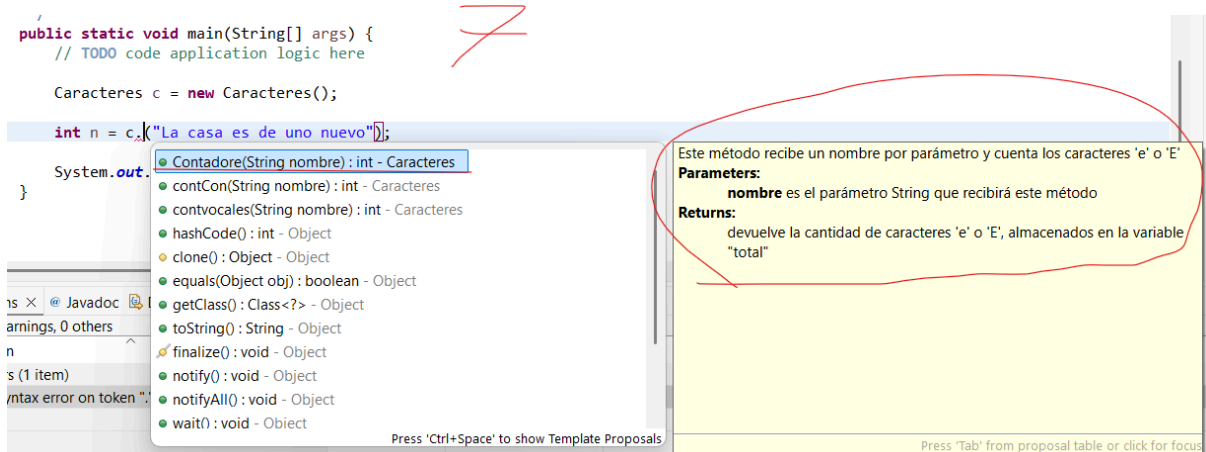


```

1 package caracteres;
2
3 /**
4  * Javadoc de autor
5  *
6  * @author guillermo
7  */
8 public class Caracateres {
9
10    /**
11     * Este método recibe un nombre por parámetro y cuenta los caracteres 'e' o
12     * 'E'
13     *
14     * @param nombre es el parámetro String que recibirá este método
15     * @return devuelve la cantidad de caracteres 'e' o 'E', almacenados en la
16     * variable "total"
17     */
18    public int Contadore(String nombre) {
19        int total = 0;
20
21        int i = 0;
22
23        while (i < nombre.length()) {
24            if ((nombre.charAt(i) == 'e') || (nombre.charAt(i) == 'E')) {
25
26                total++;
27
28            }
29
30            i++;
31
32        }
33    }
34 }

```

7) De nuevo, se nos muestra la documentación que hemos creado:



```

1 public static void main(String[] args) {
2     // TODO code application logic here
3
4     Caracateres c = new Caracateres();
5
6     int n = c.Contadore("La casa es de uno nuevo");
7
8     System.out.println(n);
9 }

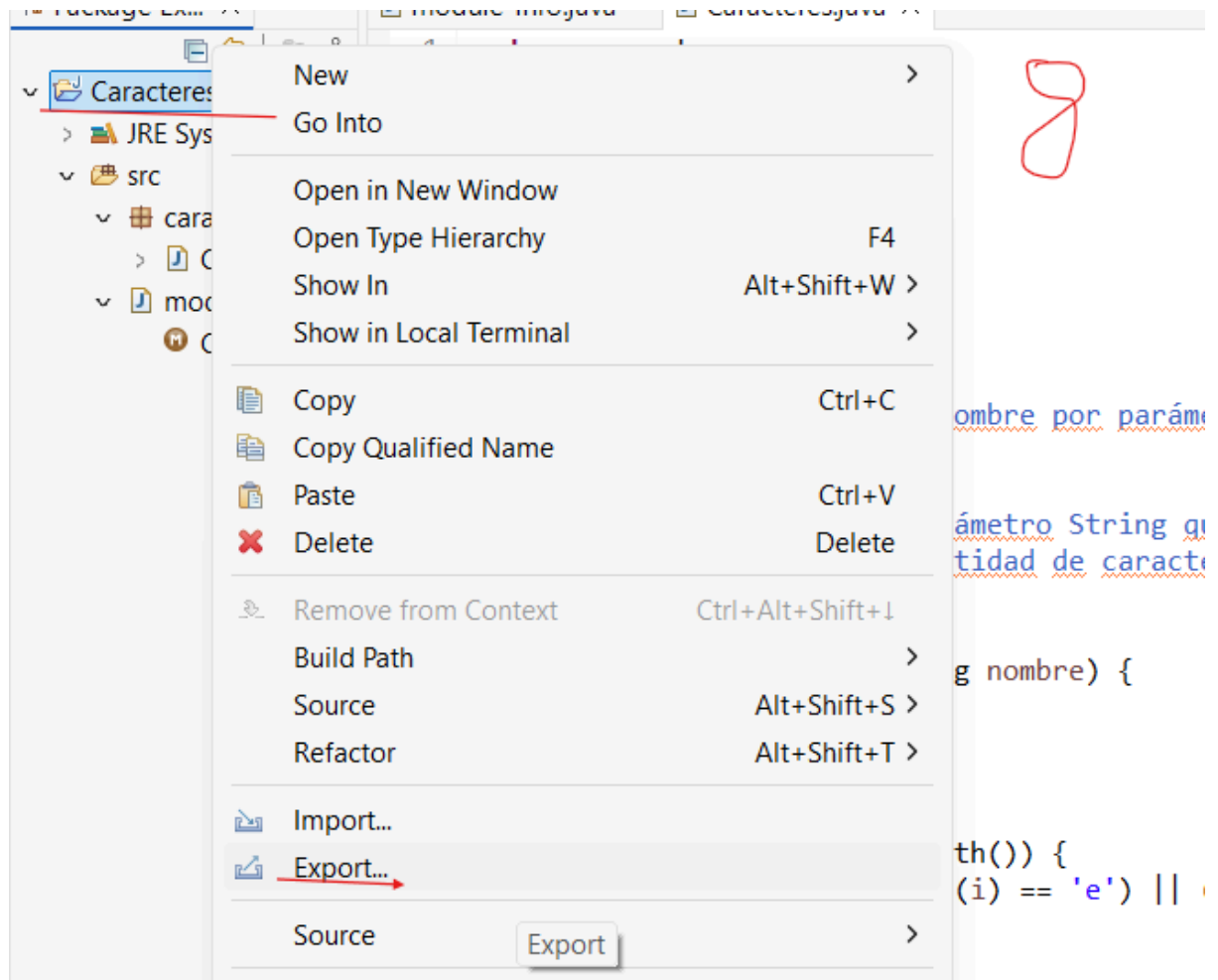
```

Este método recibe un nombre por parámetro y cuenta los caracteres 'e' o 'E'

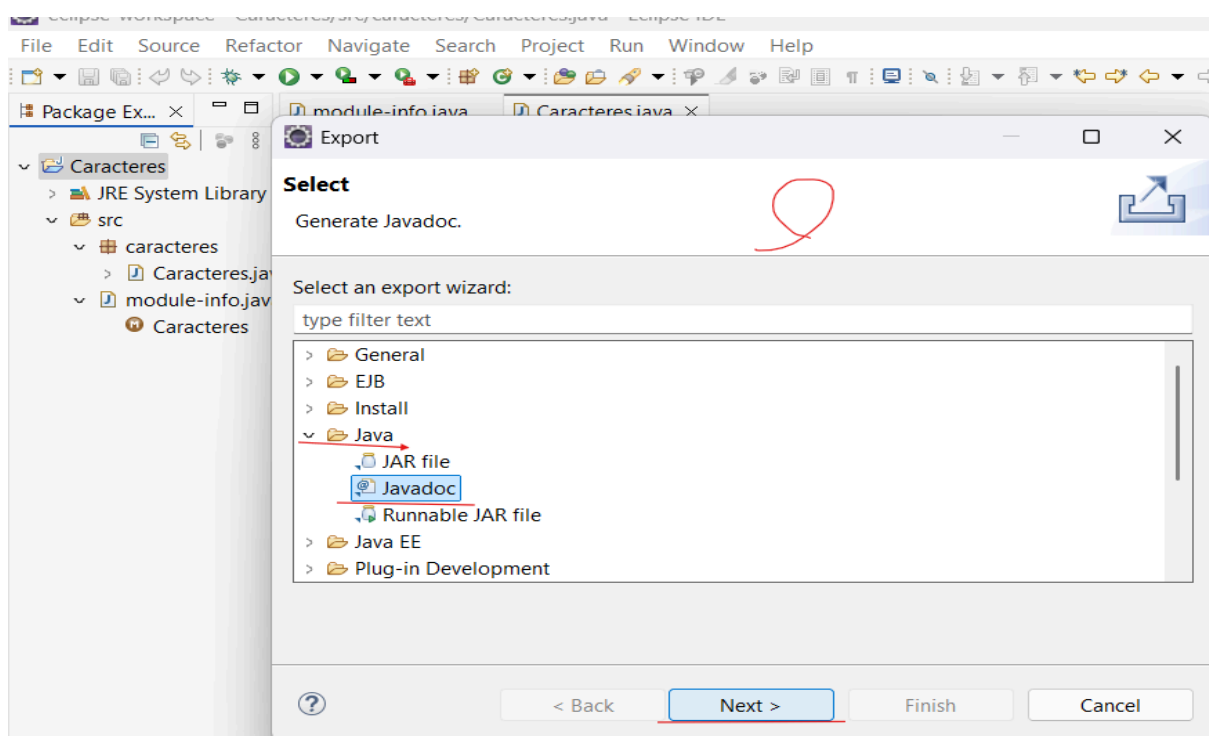
Parameters:
nombre es el parámetro String que recibirá este método

Returns:
devuelve la cantidad de caracteres 'e' o 'E', almacenados en la variable "total"

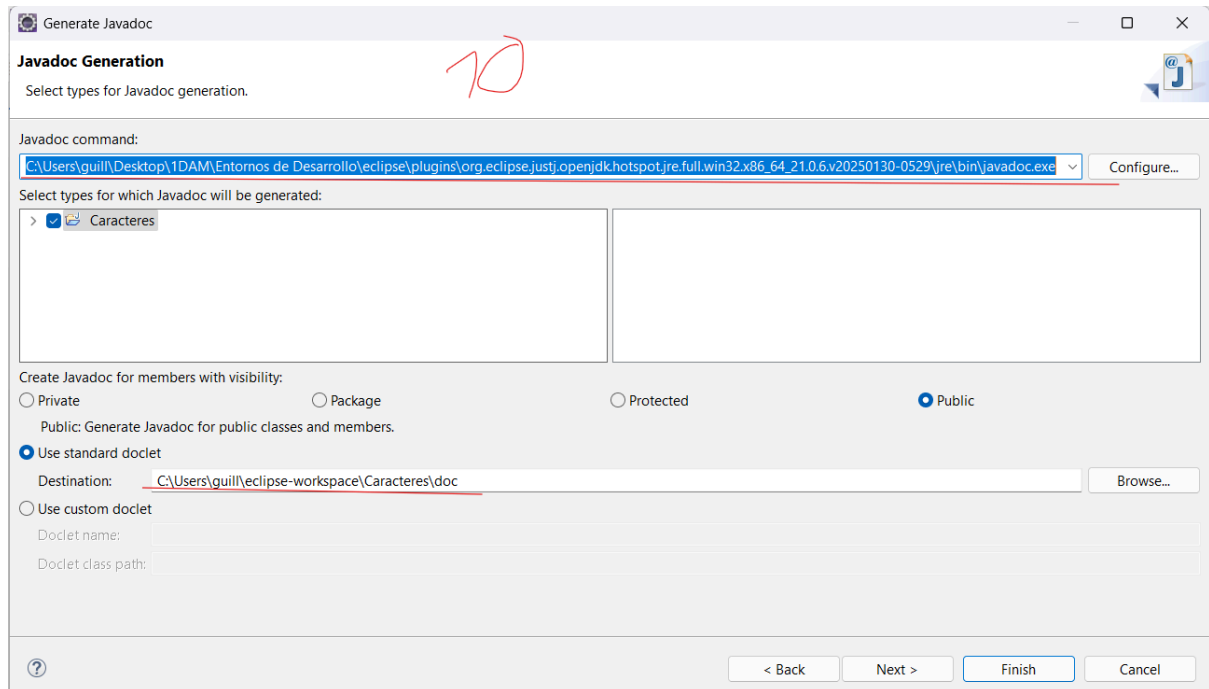
- 8) Para generar el javadoc, primero tenemos que hacer click derecho en el proyecto, y de ahí a **Export**:



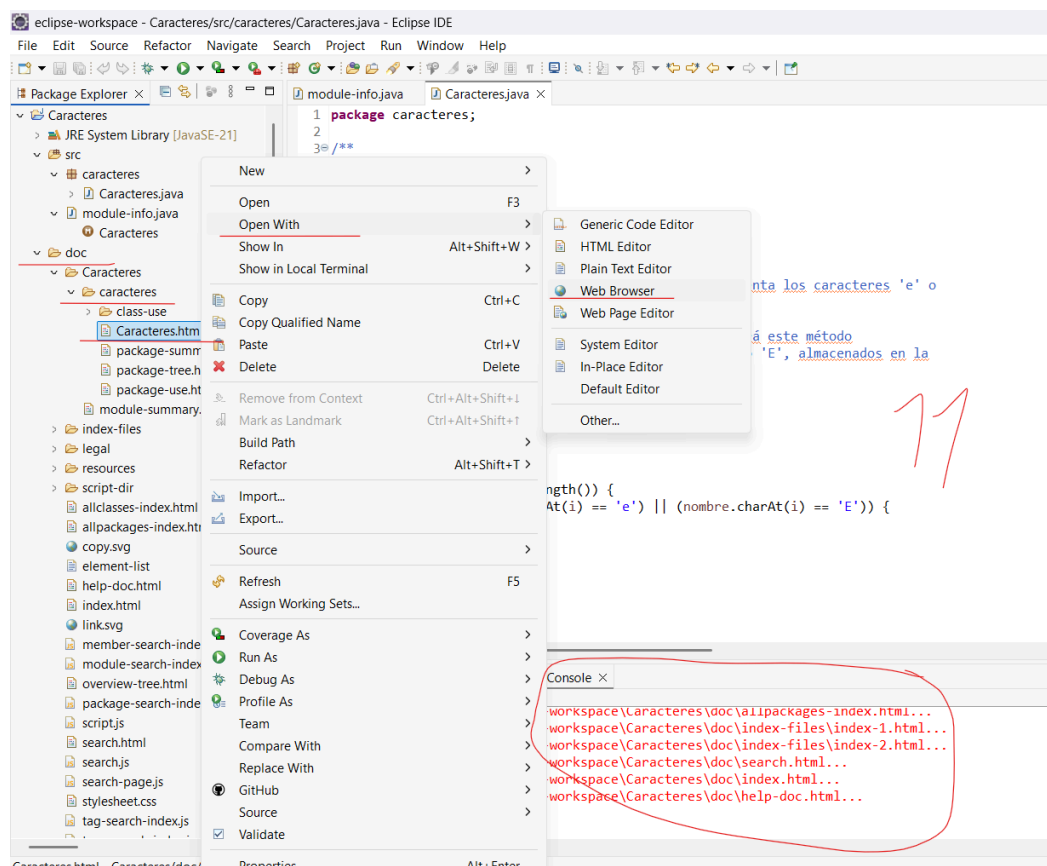
- 9) Después pinchamos en **Java**, **Javadoc**, y **Next**:



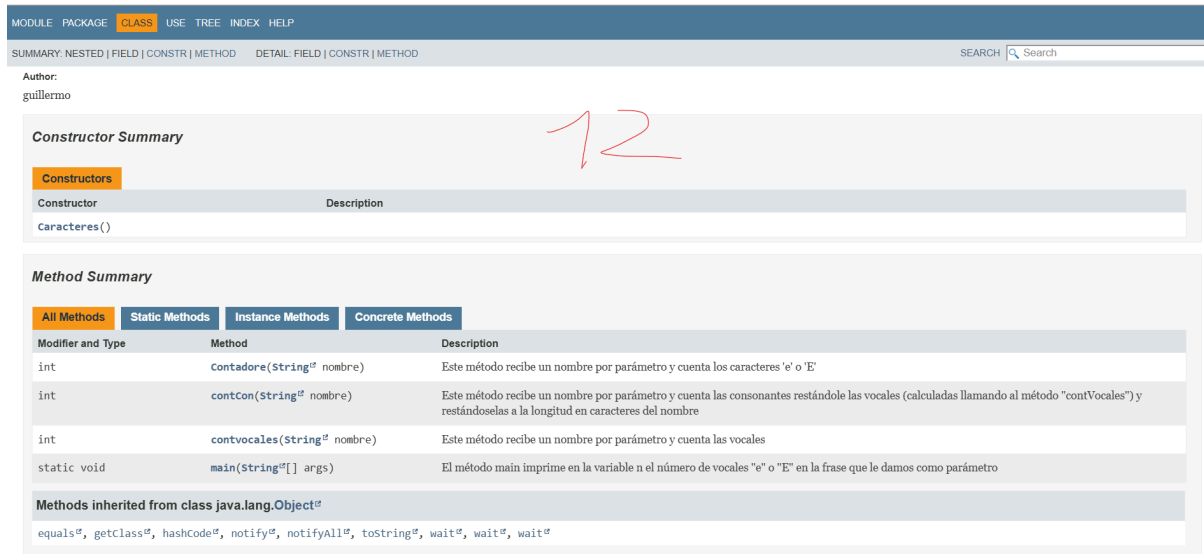
10) Nos aparecerá la opción para elegir dónde queremos guardar la documentación. Lo dejamos tal cual nos aparece para este ejercicio:



11) Abajo a la derecha podemos ver el proceso de creación de la documentación. Buscamos el archivo html del proyecto dentro de la carpeta donde lo hemos guardado, hacemos click derecho, **Open with** y **Web Browser**.



12) Al igual que con NetBeans, el html se abre sin problema:



MODULE PACKAGE **CLASS** USE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH

Author:
guillermo

Constructor Summary

Constructor	Description
Caracteres()	

Method Summary

Modifier and Type	Method	Description
int	contadore(String nombre)	Este método recibe un nombre por parámetro y cuenta los caracteres 'e' o 'E'
int	contCon(String nombre)	Este método recibe un nombre por parámetro y cuenta las consonantes restandole las vocales (calculadas llamando al método "contVocales") y restándoselas a la longitud en caracteres del nombre
int	contvocales(String nombre)	Este método recibe un nombre por parámetro y cuenta las vocales
static void	main(String[] args)	El método main imprime en la variable n el número de vocales "e" o "E" en la frase que le damos como parámetro

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Con esto, hemos generado el Javadoc tanto en NetBeans como en Eclipse, y el ejercicio ha sido finalizado.