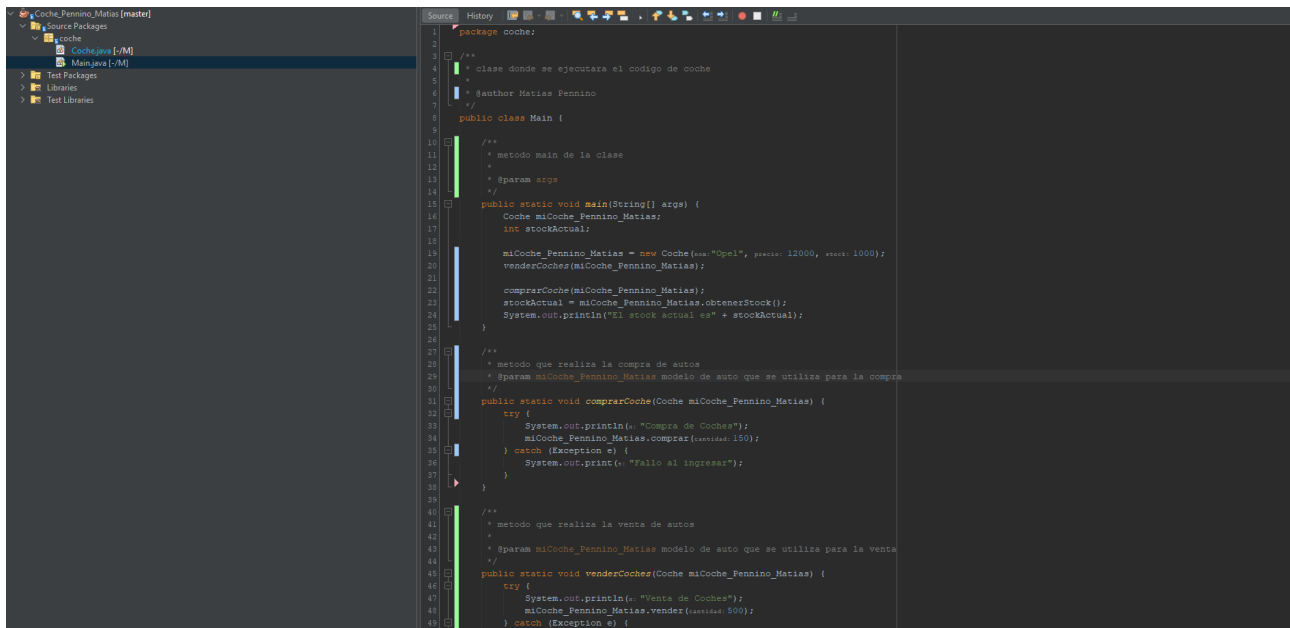


Actividad 4.4.1 Documentación javadoc proyecto Coche

Alumno: Matias Pennino



```
1 package coche;
2
3 /**
4  * clase donde se ejecutara el codigo de coche
5  *
6  * @author Matias Pennino
7  */
8
9 public class Main {
10
11     /**
12      * metodo main de la clase
13      *
14      * @param args
15      */
16     public static void main(String[] args) {
17         Coche miCoche_Pennino_Matias;
18         int stockActual;
19
20         miCoche_Pennino_Matias = new Coche("Opel", precio: 12000, stock: 1000);
21         venderCoche(miCoche_Pennino_Matias);
22
23         comprarCoche(miCoche_Pennino_Matias);
24         stockActual = miCoche_Pennino_Matias.obtenerStock();
25         System.out.println("El stock actual es" + stockActual);
26     }
27
28     /**
29      * metodo que realiza la compra de autos
30      *
31      * @param miCoche_Pennino_Matias modelo de auto que se utiliza para la compra
32      */
33     public static void comprarCoche(Coche miCoche_Pennino_Matias) {
34         try {
35             System.out.println("Compra de Coches");
36             miCoche_Pennino_Matias.comprar(cantidad: 150);
37         } catch (Exception e) {
38             System.out.print(e + "Fallo al ingresar");
39         }
40     }
41
42     /**
43      * metodo que realiza la venta de autos
44      *
45      * @param miCoche_Pennino_Matias modelo de auto que se utiliza para la venta
46      */
47     public static void venderCoche(Coche miCoche_Pennino_Matias) {
48         try {
49             System.out.println("Venta de Coches");
50             miCoche_Pennino_Matias.vender(cantidad: 500);
51         } catch (Exception e) {
52             System.out.print(e + "Fallo al ingresar");
53         }
54     }
55 }
```

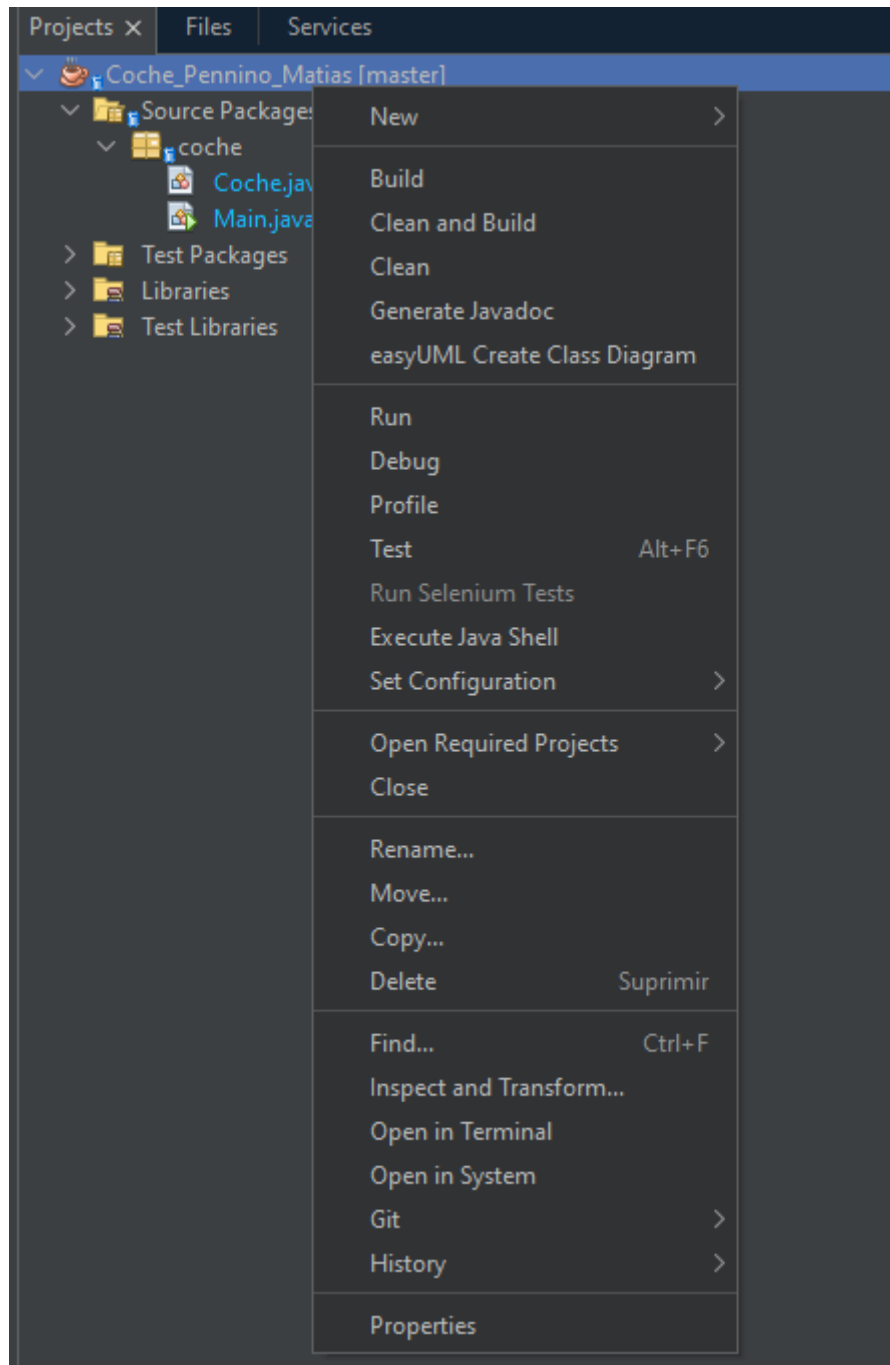
Documentación de la clase main

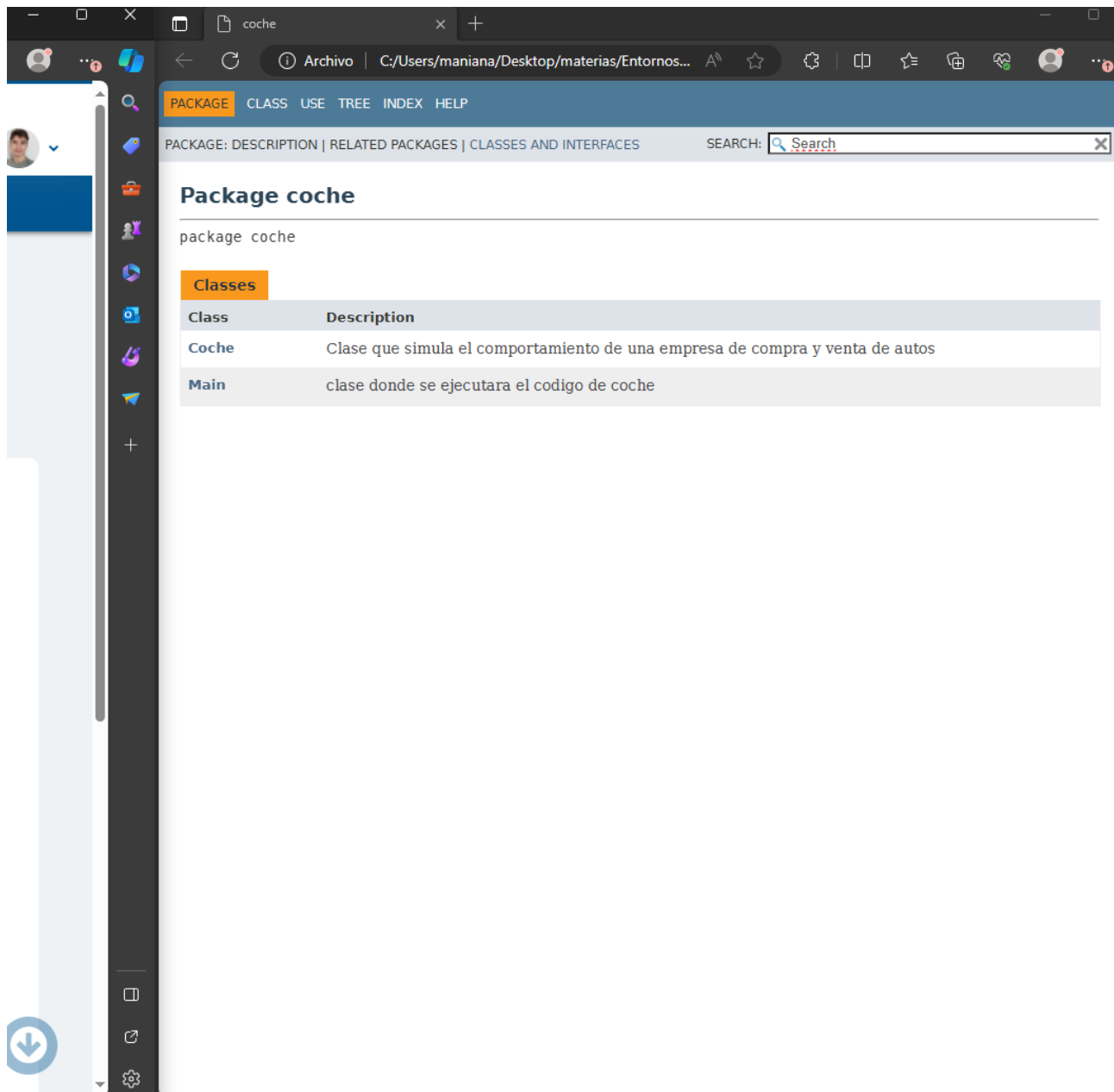
```
1 package coche;
2
3
4 /**
5  * Clase que simula el comportamiento de una empresa de compra y venta de autos
6  * @author Matias Perrino
7  */
8 public class Coche {
9
10     private String nombre;
11     private double precio;
12     private double precioIVA;
13     private int stock;
14
15     /**
16      * Constructor sin parametros
17      */
18     public Coche ()
19     {
20     }
21
22     /**
23      * constructor con tres parametros
24      * @param nom nombre del auto
25      * @param precio precio del auto
26      * @param stock cantidad de autos
27      */
28     public Coche (String nom, double precio, int stock)
29     {
30         this.nombre=nom;
31         this.precio=precio;
32         this.stock=stock;
33     }
34
35     /**
36      * metodo que asigna un nombre a un auto
37      * @param nom nombre a asignar
38      */
39     public void asignarNombre(String nom)
40     {
41         nombre=nom;
42     }
43
44     /**
45      * metodo que devuelve el nombre del auto
46      * @return nombre del auto
47      */
48     public String obtenerNombre()
49     {
```

```
46     * @return nombre del auto
47     */
48     public String obtenerNombre()
49     {
50         return nombre;
51     }
52
53     /**
54      * metodo que devuelve la cantidad de autos que hay en stock de ese modelo
55      * @return cantidad de ese modelo
56      */
57     public int obtenerStock ()
58     {
59         return stock;
60     }
61
62     /**
63      * metodo que modifica la cantidad de autos, simulando la compra de estos
64      * @param cantidad autos a comprar
65      * @throws Exception si la cantidad es negativa
66      */
67     public void comprar(int cantidad) throws Exception
68     {
69         if (cantidad<0)
70             throw new Exception(message:"No se puede comprar un n° negativo de coches");
71         stock = stock + cantidad;
72     }
73
74     /**
75      * metodo que modifica la cantidad de autos, simulando la venta de estos
76      * @param cantidad autos a vender
77      * @throws Exception si la cantidad es igual o menor a 0, o si no hay la
78      * cantidad suficiente de autos
79      */
80     public void vender (int cantidad) throws Exception
81     {
82         if (cantidad <= 0)
83             throw new Exception (message:"No se puede vender una cantidad negativa de coches");
84         if (obtenerStock())< cantidad)
85             throw new Exception (message:"No se hay suficientes coches para vender");
86         stock = stock - cantidad;
87     }
88
89 }
90
91
92
93
```

Documentación de la clase coche

Al hacer click derecho sobre el proyecto podemos generar la documentacion javadoc





Se puede ver que la documentacion fue generada con exito