Uso del depurador y algunos atajos de Eclipse

Bienvenido, tengo aquí una pequeña guía para facilitarte el trabajo si algún día en tu tiempo de ocio deseas programar con el lenguaje Java (uno de los lenguajes de programación más utilizados en la actualidad). Si amigo mío, Java está en todas partes. Muchas personas empiezan utilizando NetBeans, no obstante recomiendo encarecidamente que programéis con Eclipse para iniciarse en Java con muy buen pie.

¿Qué es Eclipse?

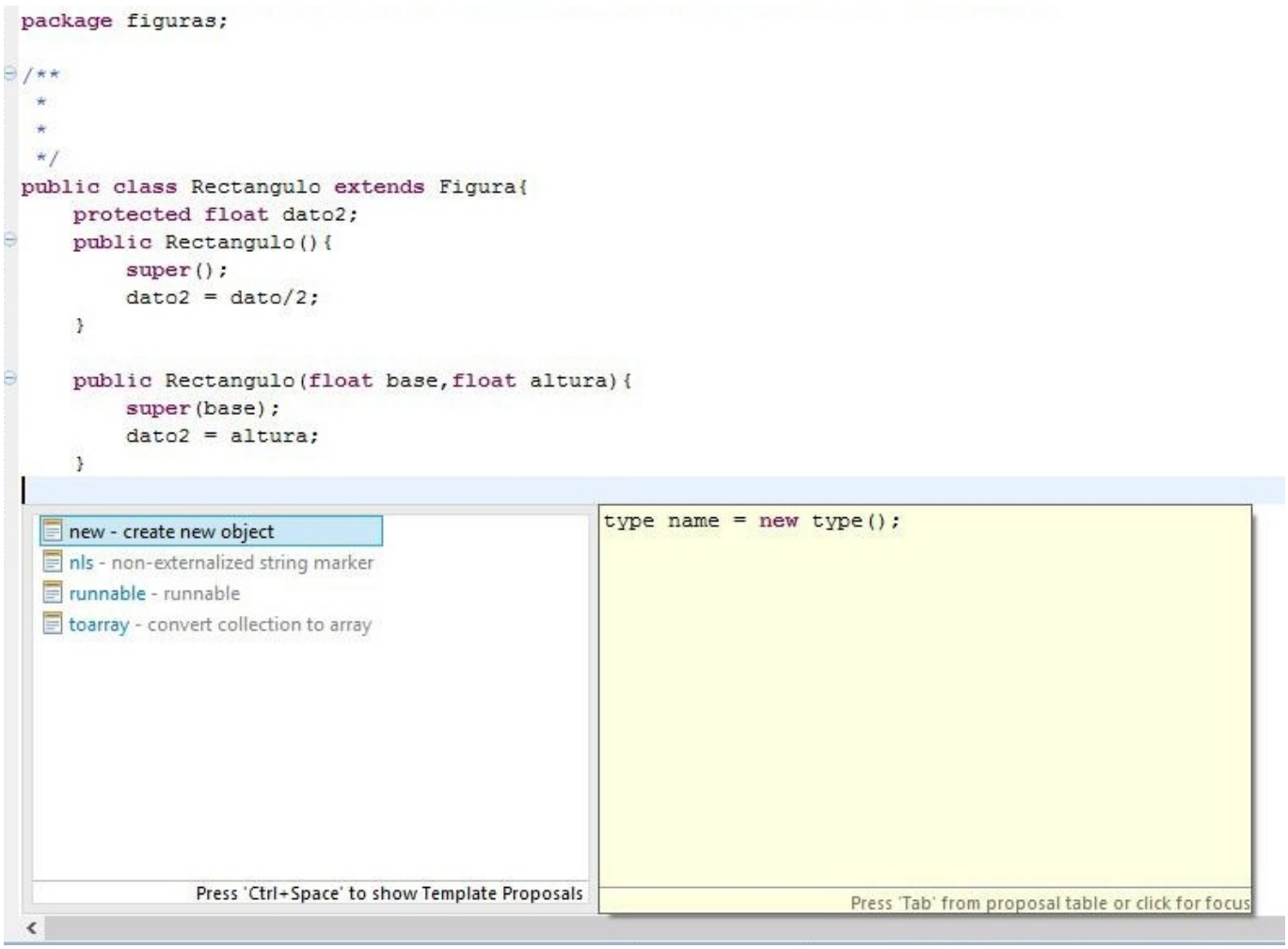
Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por si misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java (JDT)

- La documentación, los artículos y las descargas de Eclipse se encuentran disponibles en Eclipse.org.

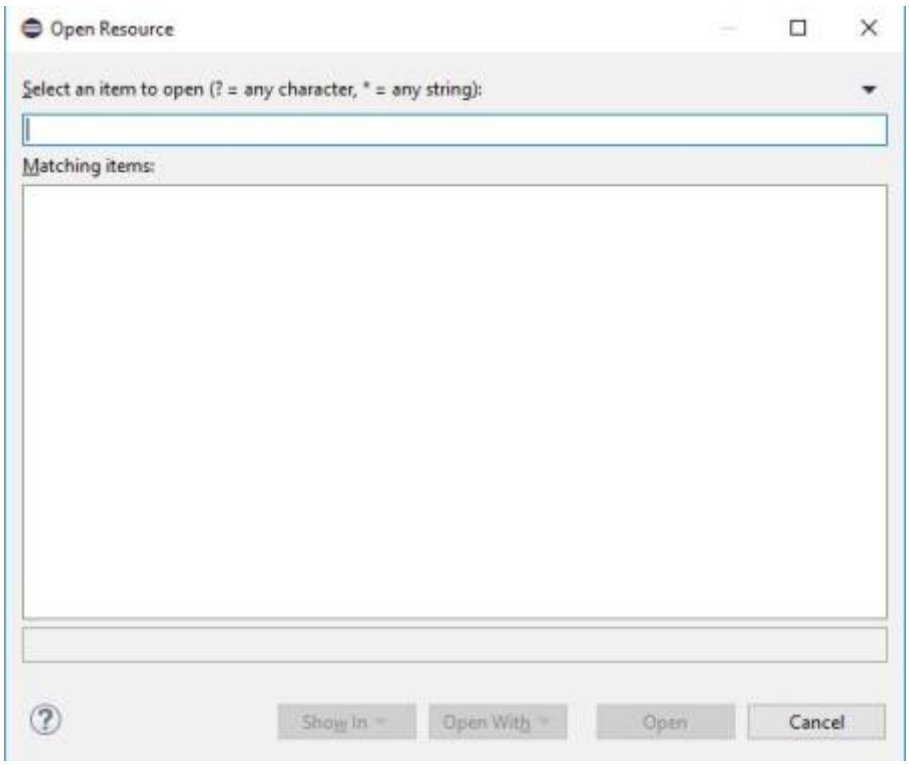
Manual de uso del depurador de Eclipse

A continuación os enseñaré un manual de uso del depurador así como de los atajos de la plataforma, tomad nota y con un poco de práctica os sabréis estos atajos de memoria:

<color #1C0068>**CTRL + BARRA_ESPACIADORA: Autocompletado de metodos/variables/etc

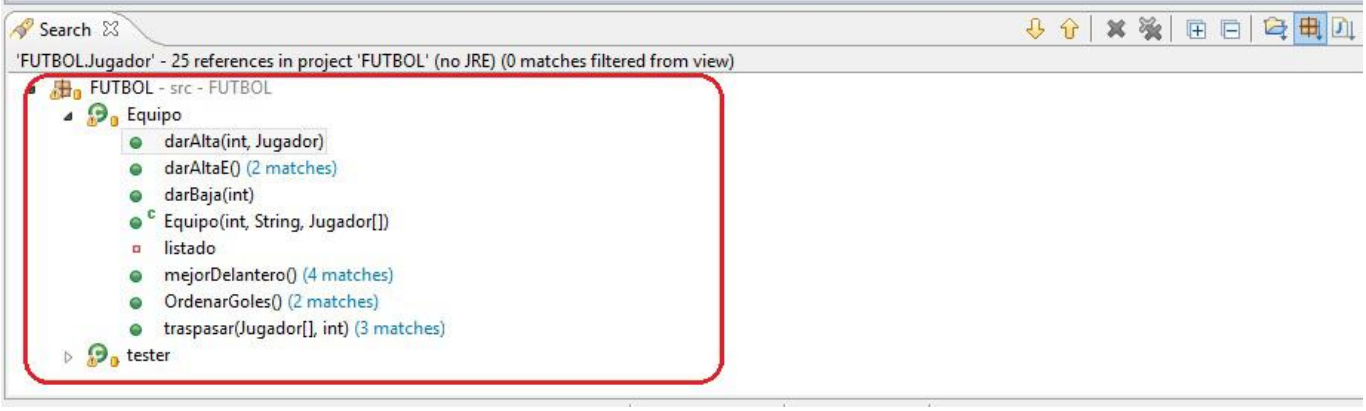


CTRL + SHIFT + R: Búsqueda de un recurso (código fuente, jar, xml, properties...) en los proyectos abiertos (sin tener que ir al explorador de paquete a buscarlo)



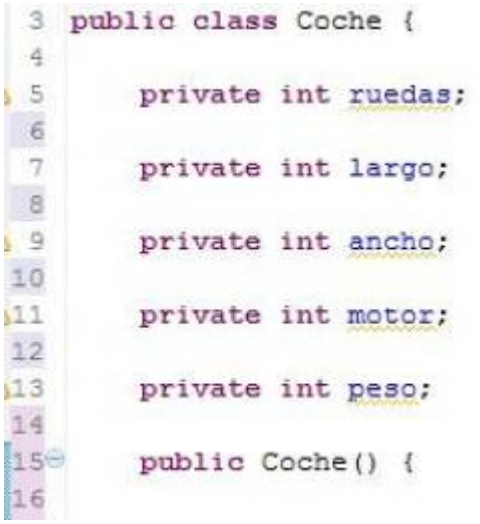
CTRL + SHIFT + G: Buscar referencias al método/clase en el Workspace. El único caso en que yo prefiero usar el botón derecho → References → Project, porque si hay muchos proyectos abiertos tarda más en hacer la búsqueda

```
1  package FUTBOL;
2
3
4  public class Jugador {
5      protected int idJugador;
6      protected String nombre;
7      protected int idEquipo;
8      protected int canPartidos;
9      protected int goles;
10
11     public Jugador(int idJugador, String nombre, int idEquipo, int canPartidos, int goles) {
12         this.idJugador = idJugador;
13         this.nombre = nombre;
14         this.idEquipo = idEquipo;
15         this.canPartidos = canPartidos;
16         this.goles = goles;
17     }
18 public Jugador(){
```

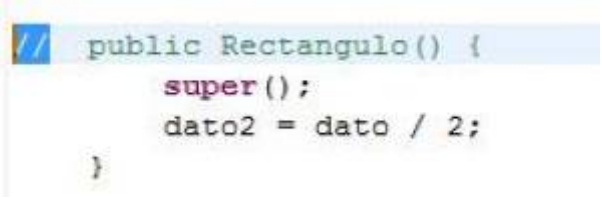


Ctrl+Shift+F: formatear código (tabulaciones, saltos de línea,...)

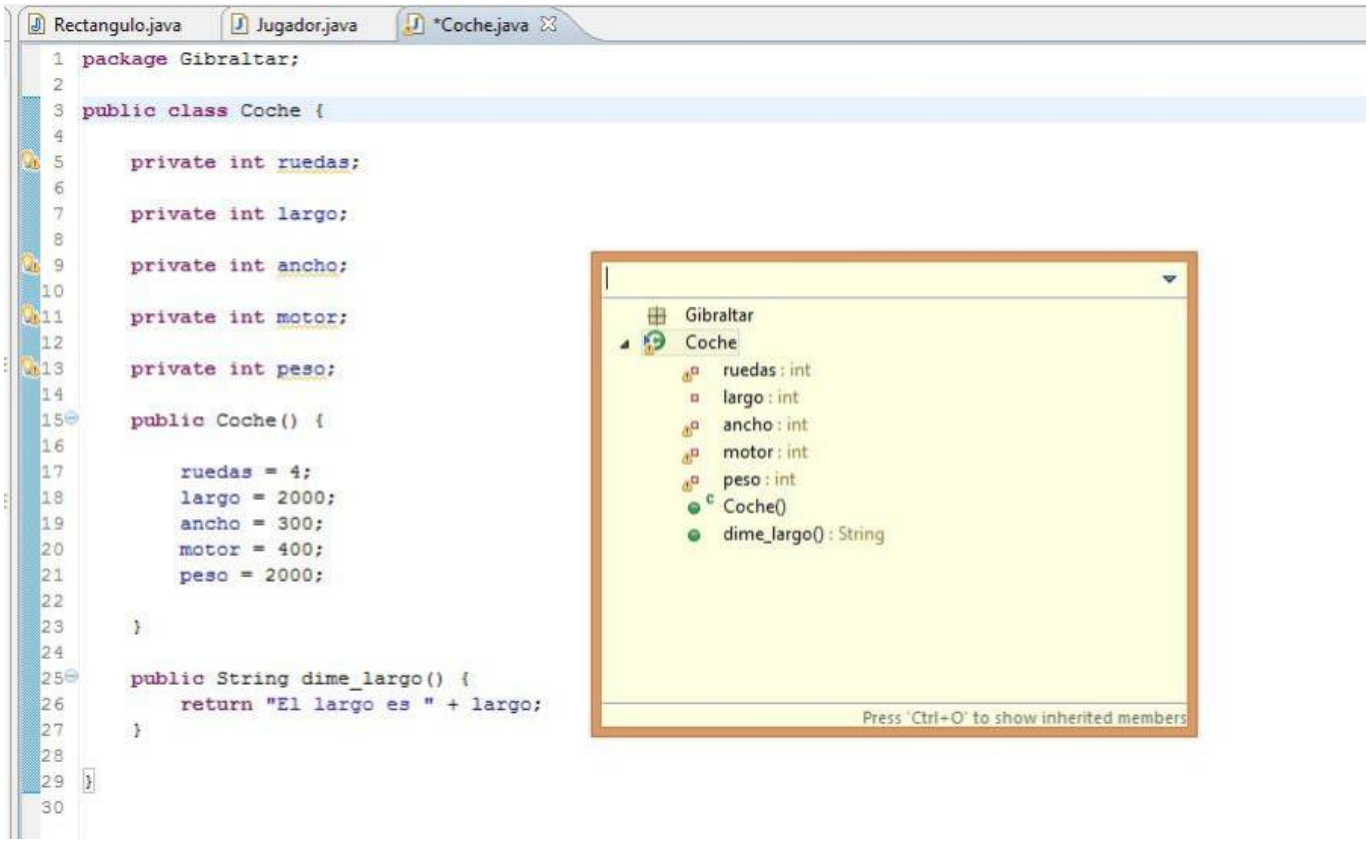
```
3  public class Coche {
4
5                      private int ruedas;
6
7                                  private int largo;
8
9
10
11             private int ancho;
12
13
14                     private int motor;
15
16                             private int peso;
17
18                             public Coche() {
19
```



Ctrl+Shift+C: comentar/descomentar con // las líneas seleccionadas



CTRL + O : Ver métodos/atributos de la Clase.

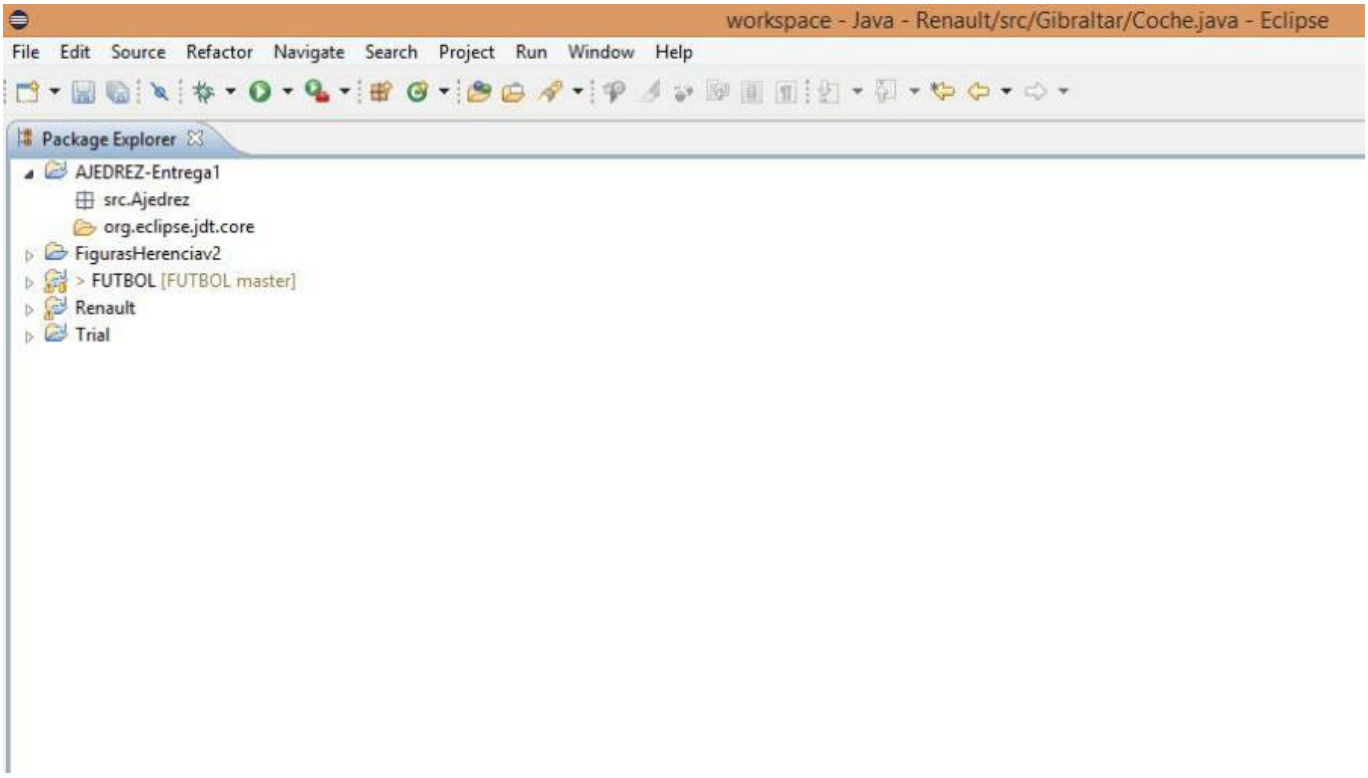


CTRL + D : Eliminar una línea

```
2
3  public class Coche {
4
5      private int ruedas;
6
7      private int largo;
8
9      private int ancho;
10
11     private int motor;
12
13     private int peso;
14
15     private int gasolina;
16
17     public Coche() {
```

```
2
3  public class Coche {
4
5      private int ruedas;
6
7      private int largo;
8
9      private int ancho;
10
11     private int motor;
12
13     private int peso;
14
15
16     public Coche() {
```

CTRL + M : Maximizar/Minimizar la ventana activa



CTRL + I : Corregir indentacion

```
private int motor;

    private int peso;
```

```
private int motor;

private int peso;
```

Ctrl + -: comprimir bloque de llaves

```
public Coche() {

public String dime_largo() {
    return "El largo es " + largo;
}

}
```

Ctrl + +: desplegar bloque de llaves

```
public Coche() {

    ruedas = 4;
    largo = 2000;
    ancho = 300;
    motor = 400;
    peso = 2000;

}
```

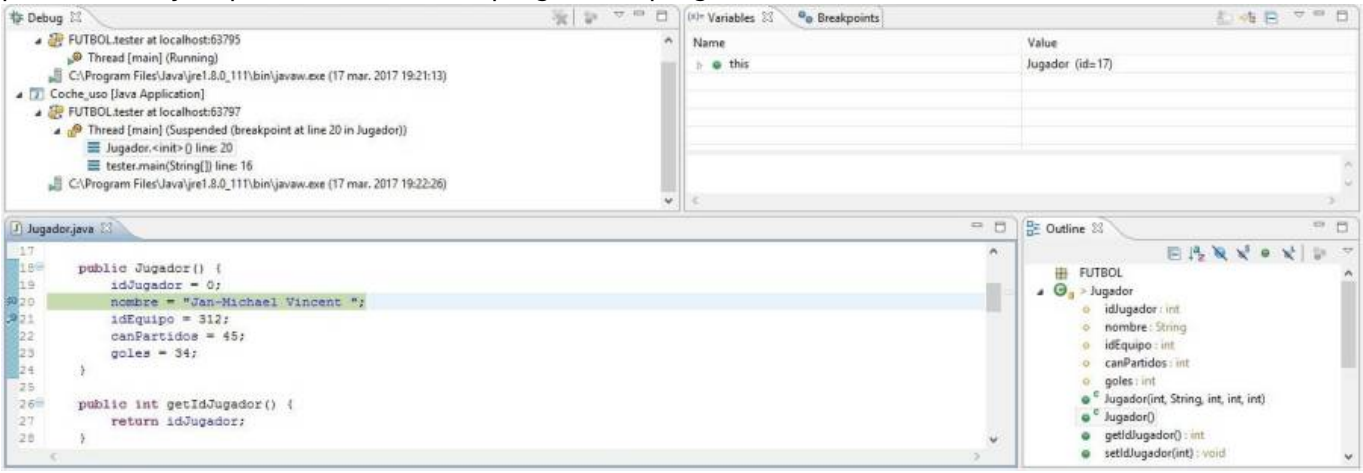
Ctrl+Shift+C: comentar/descomentar con // las líneas seleccionadas''

```java
package Gibraltar;

public class Coche {

//  private int ruedas;
//
//  private int largo;
//
//  private int ancho;
//
//  private int motor;
//
//  private int peso;
//
//
//  public Coche() {
//
//      ruedas = 4;
//      largo = 2000;
//      ancho = 300;
//      motor = 400;
//      peso = 2000;
//
//  }
//
//  public String dime_largo() {
//      return "El largo es " + largo;
//  }
```

DEBUG: Poner un punto inicial de ruptura. Seguidamente se comenzaría con el proceso debug. Si pulsamos f5 y f6 podemos ver como progresa el programa.**</color>



— *Daniel Hoyos 2018/03/10 00:47*

From:
https://server-jk.ddns.net/dokuwiki/ - **IES Palomeras-Vallecas Dep. Electronica**

Permanent link:
https://server-jk.ddns.net/dokuwiki/doku.php?id=aula:programacion:uso_del_depurador_y_algunos_atajos_de_eclipse

Last update: **2025/01/22 01:02**