

Refactorizar

Refactorizar en Eclipse

Refactorizar es el término que se usa en Programación para indicar que vamos a hacer un cambio en el código de la aplicación, pero que no afectará a la funcionalidad del mismo, sino sólo a su estructura.


Es usado para limpiar el código, hacerlo más legible, hacer mejoras en la calidad del mismo...

Vamos a ver algunos ejemplos.

Cambio de nombre de variables

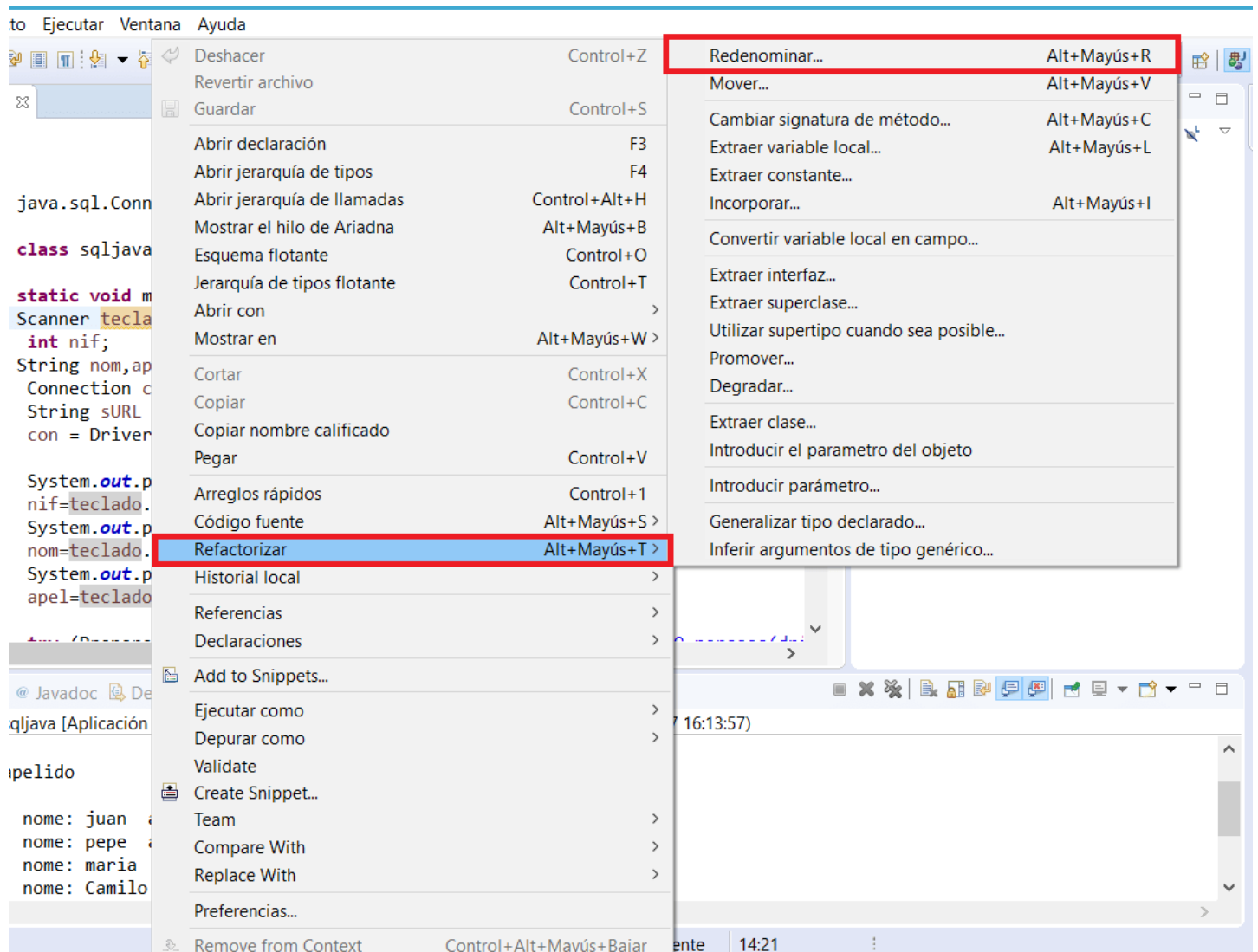
El cambio de nombre de variables y métodos puede ser una labor complicada en programas muy grandes, para ello [Eclipse](#) cuenta con una opción, que con sólo un par de clicks, cambiará de nombre esas variables en todo el proyecto, sin tener que ir buscando línea a línea dónde se usan

```
1 package trabajoeclipse;
2
3 import java.util.Scanner;
4
5 public class PruebaDepurar {
6 //Programa que va sumando cervezas hasta que lleve 5, entonces da un mensaje.
7     public static void main(String[] args) {
8         int n = 0;
9         while (n < 6) {
10             if (n == 5) {
11                 System.out.println("No bebas mas");
12             } else {
13                 System.out.println("llevas " + n);
14             }
15             n +=2;
16         }
17         System.out.println("Estas bebido");
18     }
19 }
```



Renombrar variables

Hacemos click sobre la variable «n» con el botón derecho, se abre el menú y hacemos click en «Refactor», «Rename», escribimos el nuevo nombre y se aplican los cambios en todo el proyecto:



Renombrar

El resultado será:

```

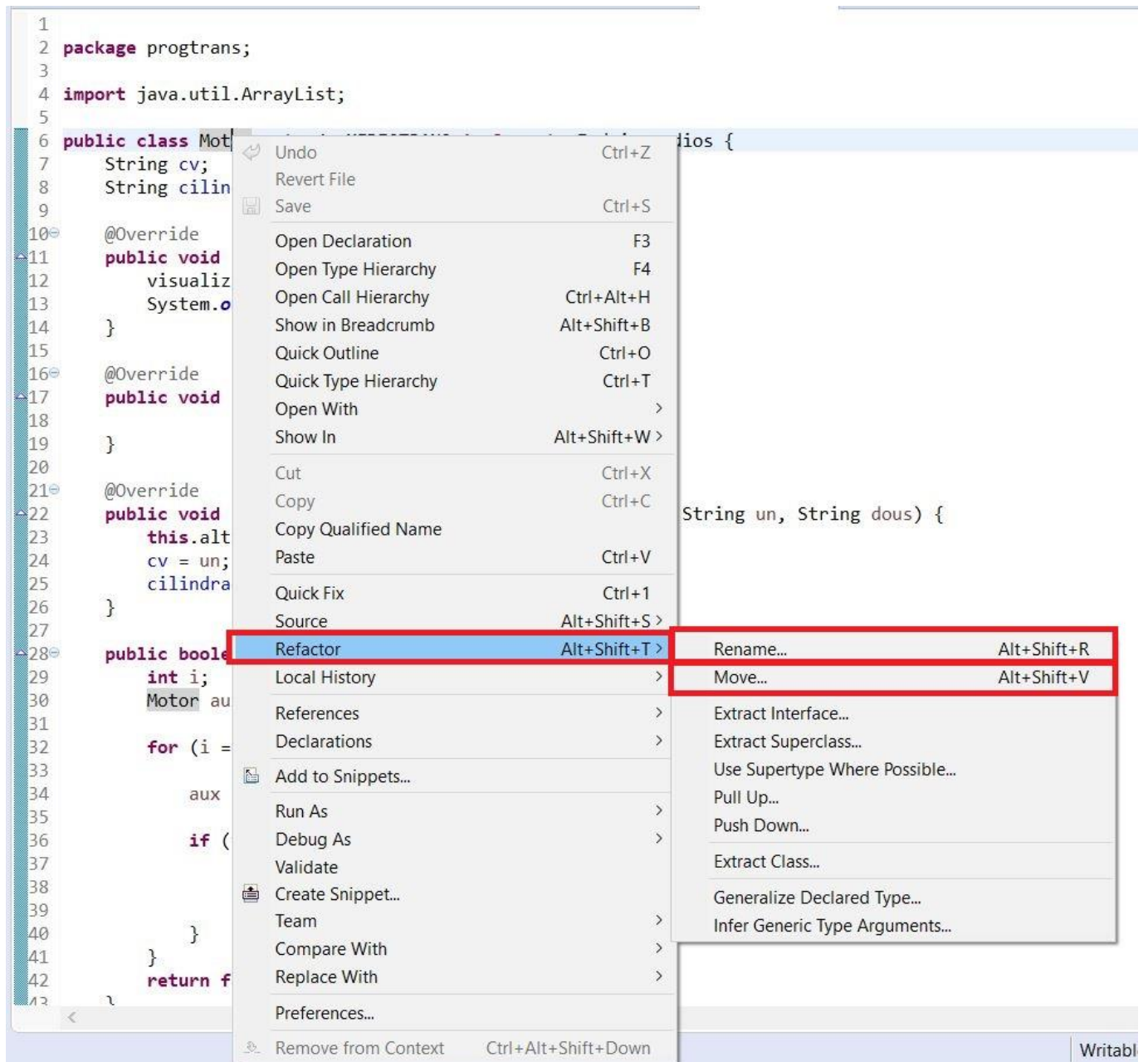
1 package trabajoeclipse;
2
3 import java.util.Scanner;
4
5 public class PruebaDepurar {
6 //Programa que va sumando cervezas hasta que lleve 5, entonces da un mensaje.
7     public static void main(String[] args) {
8         int nCervezas = 0;
9         while (nCervezas < 6) {
10             if (nCervezas == 5) {
11                 System.out.println("No bebas mas");
12             } else {
13                 System.out.println("llevas " + nCervezas);
14             }
15             nCervezas +=2;
16         }
17         System.out.println("Estas bebido");
18     }
19 }

```

Resultado refactorización

Renombrar o mover una clase

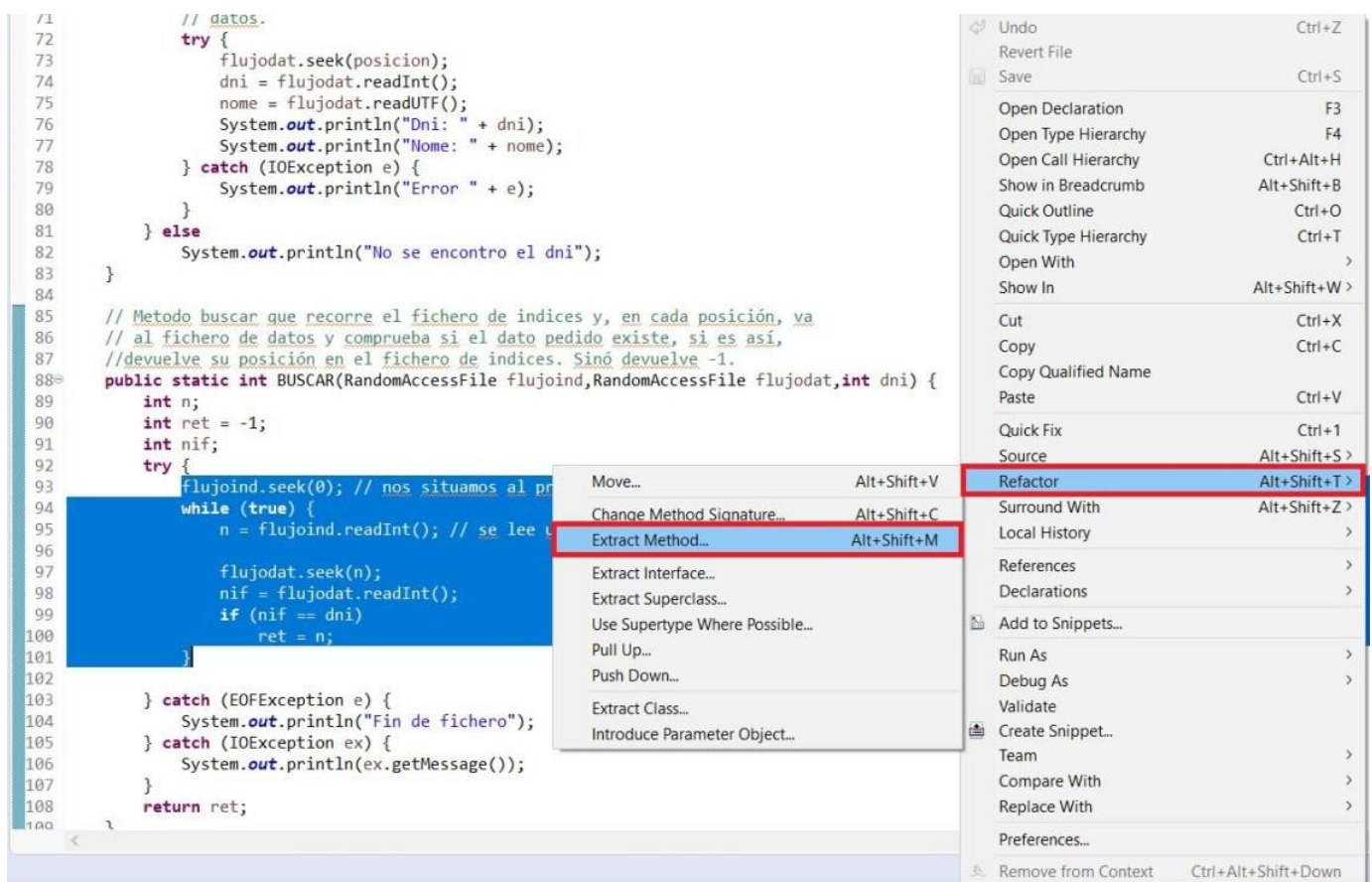
Para renombrar o mover una clase hacemos click con el botón derecho sobre la clase, luego «**Refactor**» y «**Rename**» o «**Move**». Eclipse revisa todas las clases del proyecto y cambia automáticamente los import y el nombre de la clase donde aparezca.



Renombrar clase

Extraer un método

Para extraer un método tenemos que seleccionar varias líneas de código, hacer click con el botón derecho del ratón, «Refactor», «Extract Method».



Extraer método

Escribimos un nombre para el método. En este caso «**buscar_indice**», y hacemos click en «**OK**».

Extract Method

Method name:

Access modifier: ☐ public ☐ protected ☐ package ☒ private

Parameters:

| Type | Name |
|------------------|----------|
| RandomAccessFile | flujoind |
| RandomAccessFile | flujodat |
| int | dni |
| int | ret |
| | |
| | |
| | |

Edit...

Up

Down

☐ Declare thrown runtime exceptions
☐ Generate method comment
☐ Replace additional occurrences of statements with method

Method signature preview:
private static int buscar_indice(RandomAccessFile flujoind, RandomAccessFile flujodat, **int** dni, **int** ret) **throws** IOException

Preview >

OK

Cancel

Refactorizar método

Se crea sólo el método al final del código y la llamada donde estaba el código que extrajimos.


```

85 // metodo buscar que recorre el fichero de indices y, en cada posicion, va
86 // al fichero de datos y comprueba si el dato pedido existe, si es así,
87 // devuelve su posición en el fichero de indices. Sinó devuelve -1.
88 public static int BUSCAR(RandomAccessFile flujoind, RandomAccessFile flujodat, int dni) {
89     int n;
90     int ret = -1;
91     int nif;
92     try {
93         ret = buscar_indice(flujoind, flujodat, dni, ret);
94     } catch (EOFException e) {
95         System.out.println("Fin de fichero");
96     } catch (IOException ex) {
97         System.out.println(ex.getMessage());
98     }
99     return ret;
100 }

```

Llamada al método

Método

```

103 private static int buscar_indice(RandomAccessFile flujoind, RandomAccessFile flujodat, int dni, int ret)
104     throws IOException {
105     int n;
106     int nif;
107     flujoind.seek(0); // nos situamos al principio
108     while (true) {
109         n = flujoind.readInt(); // se lee un entero del fichero
110
111         flujodat.seek(n);
112         nif = flujodat.readInt();
113         if (nif == dni)
114             ret = n;
115     }
116     return ret;
117 }
118 }

```

Crear método