

Tema 4. Optimización y documentación

Hecho por Juan Alfonso López, Michelle Díaz y Pablo Murillo



Tabla de contenidos

Refactorización.....	3
Código sin refactorizar.....	3
Código refactorizado.....	4
Git y GitHub.....	5
Inicialización del repositorio.....	5
Preparación para la subida.....	5
Añadir repositorio local a remoto.....	5
Configurando subida del proyecto.....	5
Subida del proyecto.....	6
Javadoc.....	7
Comentarios del código.....	7
Método “numeroMaximo”.....	7
Método “main”.....	7
Métodos get y set “valor_maximo”.....	7
Visualización del Javadoc.....	7

Refactorización

Código sin refactorizar

```
package entornos;
public class Main {
    public static boolean maximo(int a, int b){
        boolean valor_maximo = false;
        int num1 = a;
        int num2 = b;
        if (num1>num2) {
            valor_maximo = true;
        } else if (num2>num1) {
            valor_maximo = false;
        }
        return valor_maximo;
    }
    public static void main(String[] args) {
        maximo(10, 20);
    }
}
```

Código refactorizado

```
public class Main {
    private static boolean valor_maximo;

    public static boolean numeroMaximo(int b, int a){
        setValor_maximo(false);
        if (a>b) {
            setValor_maximo(true);
        } else if (b>a) {
            setValor_maximo(false);
        }
        return isValor_maximo();
    }

    public static void main(String[] args) {
        numeroMaximo(20, 10);
    }

    private static boolean isValor_maximo() {
        return valor_maximo;
    }

    private static void setValor_maximo(boolean valor_maximo) {
        Main.valor_maximo = valor_maximo;
    }
}
```

- Hemos cambiado el nombre al método.
- Hemos convertido la variable “valor_maximo” en un campo.
- Hemos cambiado la signatura del método y hemos intercambiado los valores de entrada.
- Hemos incorporado la variable “num1”.
- Hemos incorporado la variable “num2”.
- Encapsulamos campo “valor_maximo”.

Git y GitHub

Para crear el repositorio vamos a usar los comandos de consola.

Inicialización del repositorio

Con el comando “git init” iniciamos el repositorio en el directorio que nosotros queramos.

Para ello creará un archivo “.git”.

Preparación para la subida

Para subir el proyecto a GitHub usaremos el comando “git add <nombre-del-archivo>” y si hacemos uso del comando “git status” podremos ver como el directorio con el proyecto se va a subir.

```
pablo@pablo-ubuntu:~/Escritorio/Proyecto-Tema4$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
nuevos archivos: entornos/src/entornos/Main.java
```

Con el comando “git commit -m “*mensaje*”” para añadir un mensaje a la hora de la subida y dejar los directorios/ficheros listos para la subida.

Añadir repositorio local a remoto

Para poder subir los ficheros a GitHub deberemos usar el comando “git remote add origin <link-del-repositorio>” y luego ya podríamos subir el proyecto.

Configurando subida del proyecto

Con el comando “git push” subiremos el proyecto, pero al ser un repositorio local convertido en remoto tendremos que usar el comando “git push --set-upstream origin master” donde nos pide usuario y contraseña, una vez asignadas se juntarán las dos ramas y ya se podrá subir todo.

```
pablo@pablo-ubuntu:~/Escritorio/tst$ git push --set-upstream origin master
Username for 'https://github.com': PabloMurilloR
Password for 'https://PabloMurilloR@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 219 bytes | 219.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/PabloMurilloR/test_entornos.git
 * [new branch]      master -> master
Rama 'master' configurada para hacer seguimiento a la rama remota 'master' de 'origin'.
```

Subida del proyecto

Una vez hecho todo lo anterior ya tendremos subido todo el repositorio a GitHub, en nuestro caso todo el Javadoc, el código y la documentación. Podremos comprobarlo a través del enlace “[este enlace](#)”.

Javadoc

Comentarios del código.

Método “numeroMaximo”

```
/** -----
 * @author Pablo, Juan Alfonso, Michelle
 * @since 15-04-2022
 * @return Valor de la funcion "isValor_maximo"
 * @param b es el segundo número
 * @param a es el primer número */
public static boolean numeroMaximo(int b, int a){
    setValor_maximo(false);
    if (a>b) {
        setValor_maximo(true);
    } else if (b>a) {
        setValor_maximo(false);
    }
    return isValor_maximo();
}
```

Método “main”

```
/** -----
 * @author Pablo, Juan Alfonso, Michelle
 * @since 13-04-2022 */
public static void main(String[] args) {
    numeroMaximo(20, 10);
}
```

Métodos get y set “valor_maximo”

```
/** -----
 * @author Pablo, Juan Alfonso, Michelle
 * @since 17-04-2022
 * @return Retorna el atributo "valor_maximo" */
private static boolean isValor_maximo() {
    return valor_maximo;
}

/** -----
 * @author Pablo, Juan Alfonso, Michelle
 * @since 17-04-2022
 * @param valor_maximo es el booleano para saber si es el número máximo o no */
private static void setValor_maximo(boolean valor_maximo) {
    Main.valor_maximo = valor_maximo;
}
```

Visualización del Javadoc

Para visualizar el Javadoc accedemos desde GitHub.