



Capturas Practica Volumenes y Superficies

Israel Lozano
Fernández-Pacheco



Enlace al repositorio de GitHub:

<https://github.com/isralfp14/PracticaVolúmenesYSuperficies>

Pasos para implementar archivos en el repositorio:

Crear el repositorio en GitHub:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

isralfp14 / PracticaVolúmenesYSuperficies

✔ PracticaVolúmenesYSuperficies is available.

Great repository names are short and memorable. Need inspiration? How about **super-duper-octo-palm-tree** ?

Description (optional)

Tarea con eclipse de volúmenes y superficies

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

Inicializar el repositorio en la carpeta correspondiente con **git init**

```
C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>git init
Reinitialized existing Git repository in C:/Users/Usuario/Desktop/MEDAC 24-25/Entornos de desarrollo/Ejercicios eclipse/.git/
```



Añadir la carpeta “PracticaVolúmenesYSuperficies” con **git add** y posteriormente hacer un **git commit** con el mensaje “Repositorio practica” por ejemplo.

```
C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>git add PracticaVolúmenesYSuperficies
warning: in the working copy of 'PracticaVolúmenesYSuperficies/.gitignore', LF will be replaced by CRLF the next time Git touches it

C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>git commit -m "repositorio practica"
[main 8818488] repositorio practica
10 files changed, 213 insertions(+)
create mode 100644 PracticaVolúmenesYSuperficies/.classpath
create mode 100644 PracticaVolúmenesYSuperficies/.gitignore
create mode 100644 PracticaVolúmenesYSuperficies/.project
create mode 100644 PracticaVolúmenesYSuperficies/.settings/org.eclipse.core.resources.prefs
create mode 100644 PracticaVolúmenesYSuperficies/.settings/org.eclipse.jdt.core.prefs
create mode 100644 PracticaVolúmenesYSuperficies/src/app/Principal.java
create mode 100644 PracticaVolúmenesYSuperficies/src/clases/Cilindro.java
create mode 100644 PracticaVolúmenesYSuperficies/src/clases/Cubo.java
create mode 100644 PracticaVolúmenesYSuperficies/src/tests/testCilindro.java
create mode 100644 PracticaVolúmenesYSuperficies/src/tests/testCubo.java

C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 2.80 KiB | 191.00 KiB/s, done.
Total 18 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/isralfp14/PracticaVolúmenesYSuperficies.git
   9e204df..8818488  main -> main

C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>
```

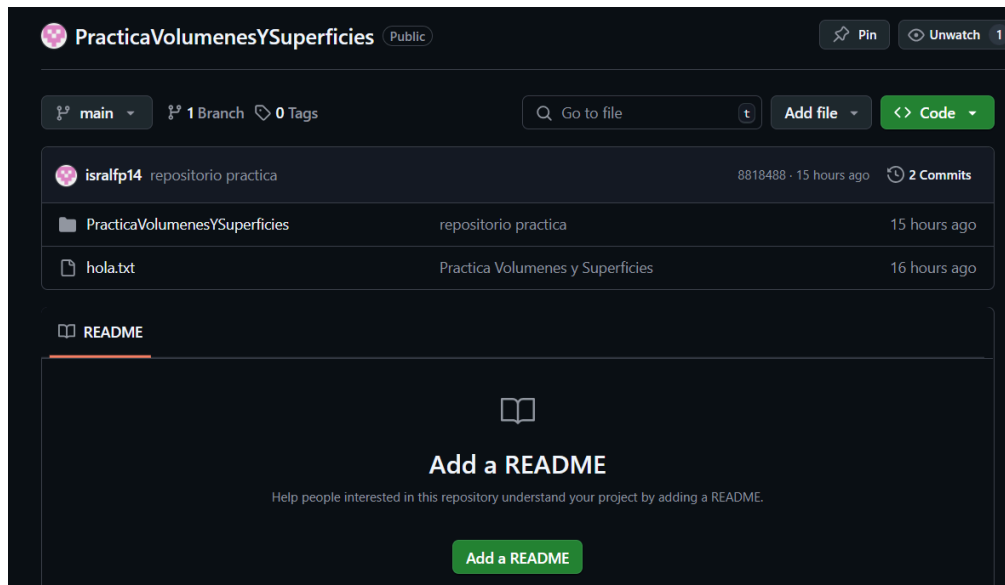
Por último hacer un **git push** para subirlo al repositorio online de GitHub

```
C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 2.80 KiB | 191.00 KiB/s, done.
Total 18 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/isralfp14/PracticaVolúmenesYSuperficies.git
   9e204df..8818488  main -> main

C:\Users\Usuario\Desktop\MEDAC 24-25\Entornos de desarrollo\Ejercicios eclipse>
```



Comprobar que efectivamente se ha subido la carpeta al repositorio:



Código del ejercicio:

Clases:

```
1 package clases;
2
3 public class Cilindro {
4     int radio;
5     int altura;
6     final double pi;
7
8     public Cilindro(int r, int alt) {
9         this.radio = r;
10        this.altura = alt;
11        this.pi = Math.PI;
12    }
13
14    public Cilindro() {
15        this.radio = 1;
16        this.altura = 1;
17        this.pi = Math.PI;
18    }
19
20    /**
21     * Calcula la superficie del cilindro
22     * @return superficie calculada
23     */
24
25    public int superficie() {
26        int sup = 0;
27        sup = (int) (2 * pi * Math.pow(radio, 2)) + (int) (2 * pi * this.radio * this.altura);
28        return sup;
29    }
30
31    /**
32     * Calcula el volumen del cilindro
33     * @return volumen calculado
34     */
35
36    public int volumen() {
37        int area = (int) (pi * Math.pow(radio, 2));
38        return area * this.altura;
39    }
40 }
```



```
Cubo.java x Cilindro.java Principal.java testCilindro.java testCubo.java
1 package clases;
2
3 public class Cubo {
4     int lado;
5
6     public Cubo(int lado) {
7         this.lado = lado;
8         if (this.lado <= 0) {
9             try {
10                 this.lado = lado;
11             }
12             catch (IllegalArgumentException fallo) {
13                 System.out.println("Error. El lado debe ser mayor a 0");
14             }
15         }
16     }
17
18     public Cubo() {
19         this.lado = 1;
20     }
21
22     /**
23      *Calcula la superficie de un cubo
24      * @return superficie calculada
25      */
26     public int superficie() {
27         return (this.lado * this.lado) * 6;
28     }
29
30     /**
31      *Calcula el volumen de un cubo
32      * @return volumen calculado
33      */
34
35     public int volumen() {
36         return (int) Math.pow(this.lado, 3);
37     }
38 }
```

Main:

```
1 package app;
2
3 import clases.Cilindro;
4
5 public class Principal {
6
7     public static void main(String[] args) {
8
9         int random = (int) (Math.random()*(10-1) + 1);
10
11         System.out.print("El numero aleatorio es: " + random + "\n");
12
13         //Cubo
14         Cubo c1 = new Cubo(random);
15         Cubo c2 = new Cubo(random);
16
17         System.out.print("-----CUBO-----" + "\n");
18         System.out.print("Superficie 1: " + c1.superficie() + "\n");
19         System.out.print("Superficie 2: " + c2.superficie() + "\n");
20         System.out.print("Volumen 1: " + c1.volumen() + "\n");
21         System.out.print("Volumen 2: " + c2.volumen() + "\n");
22         System.out.print("=====");
23
24         //Cilindro
25         Cilindro cil1 = new Cilindro(random, random);
26         Cilindro cil2 = new Cilindro(random, random);
27
28         System.out.print("-----CILINDRO-----" + "\n");
29         System.out.print("Superficie 1: " + cil1.superficie() + "\n");
30         System.out.print("Superficie 2: " + cil2.superficie() + "\n");
31         System.out.print("Volumen 1: " + cil1.volumen() + "\n");
32         System.out.print("Volumen 2: " + cil2.volumen() + "\n");
33         System.out.print("=====");
34     }
35 }
```

<terminated> Principal [Java Application] C:\Users\Usuario\Desktop\MEDA
El numero aleatorio es: 4
-----CUBO-----
Superficie 1: 96
Superficie 2: 96
Volumen 1: 64
Volumen 2: 64
=====

-----CILINDRO-----
Superficie 1: 200
Superficie 2: 200
Volumen 1: 200
Volumen 2: 200
=====

Outline x
main(String[]) : void



Tests:

```
1 package tests;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 import clases.Cilindro;
8 import clases.Cubo;
9
10 class testCilindro {
11
12     @Test
13     void testSuperficie() {
14         Cilindro cili1 = new Cilindro(9, 9);
15         int superficieEsperada = 1016;
16         int superficieObtenida = cili1.superficie();
17         assertEquals(superficieEsperada, superficieObtenida);
18     }
19
20     @Test
21     void testVolumen() {
22         Cilindro cili1 = new Cilindro(9, 9);
23         int volumenEsperado = 2286;
24         int volumenObtenido = cili1.volumen();
25         assertEquals(volumenEsperado, volumenObtenido);
26     }
27 }
28
29
```

```
Cubo.java  Cilindro.java  Principal.java  testCilindro.java  testCubo.java X
1 package tests;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 import clases.Cubo;
8
9 class testCubo {
10
11     @Test
12     void testSuperficie() {
13         Cubo c1 = new Cubo(3);
14         int superficieEsperada = 54;
15         int superficieObtenida = c1.superficie();
16         assertEquals(superficieEsperada, superficieObtenida);
17     }
18
19     @Test
20     void testVolumen() {
21         Cubo c1 = new Cubo(3);
22         int volumenEsperado = 27;
23         int volumenObtenido = c1.volumen();
24         assertEquals(volumenEsperado, volumenObtenido);
25     }
26
27 }
28
```