




- Realiza las acciones de gradle run y clean en helloGradle.
 - Sube el proyecto a la tarea.


 Inicializamos el programa

```
yosari@lliurex-alda:~/NetBeansProjects/helloGradle$ gradle init
Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details

> Task :init SKIPPED
The settings file 'settings.gradle' already exists. Skipping build initialization.

BUILD SUCCESSFUL in 6s
```

```
yosari@lliurex-alda:~/NetBeansProjects/helloGradle$ tree
.
├── app
│   ├── build
│   │   ├── classes
│   │   │   ├── java
│   │   │   │   ├── main
│   │   │   │   │   ├── com
│   │   │   │   │   │   ├── iesME
│   │   │   │   │   │   │   ├── edd
│   │   │   │   │   │   │   │   App.class
│   │   │   │   │   └── test
│   │   │   │   │       ├── com
│   │   │   │   │       │   ├── iesME
│   │   │   │   │       │   │   ├── edd
│   │   │   │   │       │   │   AppTest.class
│   │   └── distributions
```

 Miramos donde se encuentra ubicado.

 Ejecutamos el programa para ver los datos de salida.

```
yosari@lliurex-alda:~/NetBeansProjects/helloGradle$ gradle run

> Task :app:run
Hello World!

BUILD SUCCESSFUL in 1s
2 actionable tasks: 2 executed
yosari@lliurex-alda:~/NetBeansProjects/helloGradle$ gradle clean

BUILD SUCCESSFUL in 765ms
1 actionable task: 1 executed
```

- Cread un proyecto con Gradle para vuestro código de la calculadora. A esta calculadora debéis añadirle ahora dos métodos adicionales:
 - `public Boolean isPrime(float op1){...}` esta función nos devolverá un valor lógico, indicando si el número es primo o no (primo=divisible solo por él mismo y por 1).

```

66
67 public boolean isPrime(float op1) {
68     boolean esPrimo = true;
69     float contador = op1 - 1;
70     while (contador > 1) {
71         if (op1 % contador == 0) {
72             esPrimo = false;
73         }
74         contador--;
75     }
76     return (esPrimo == true) ? false : true;
77 }
78

```

- `public Integer nextPrime(float op1){...}` este método nos devolverá el siguiente número primo al que le indiquemos (si indicamos un número primo, nos devolverá el mismo).

```

79 public int nextPrime(float op1) {
80
81     if (op1 < 2) {
82         return 2;
83     }
84
85     int prime = (int) Math.ceil(op1);
86     if (prime % 2 == 0) {
87         prime++;
88     }
89
90     while (isPrime(prime)) {
91         prime += 2;
92     }
93     return prime;
94 }
95

```


```

venancio@venancio:~/NetBeansProjects/calcula$ gradle init

> Task :init SKIPPED
The settings file 'settings.gradle' already exists. Skipping build initialization.

BUILD SUCCESSFUL in 11s

```

 Miramos donde se encuentra ubicado

```


venancio@venancio:~/NetBeansProjects/calcula$ tree
.
├── app
│   ├── build
│   │   ├── classes
│   │   │   └── java
│   │   │       └── main
│   │   │           └── calcula
│   │   │               ├── App.class
│   │   │               └── calculadora.class
│   └── generated

```

 Pasamos los valores por el archivo build.gradle.

```

45     }
46     run{
47         args=['4','3']
48     }
49 
```

 Importamos la libreria Math para llamar en el programa princial.

```

implementation 'com.google.guava:guava:30.1.1-jre'
// https://mvnrepository.com/artifact/org.apache.commons/commons-math3
implementation 'org.apache.commons:commons-math3:3.6.1'
}

```

```

4 package com.iesME.edd.calcula;
5 import org.apache.commons.math3.primes.Primes;
6

```

 Pasamos los valores para poder ejecutar el programa. En este caso, son 4 y 3.

```
venancio@venancio:~/NetBeansProjects/calcula$ gradle run --args " 4 3"
```

```

> Task :app:run
La suma entre 4.0 y 3.0 es :7.0
La resta entre 4.0 y 3.0 es :1.0
La multiplicacion entre 4.0 y 3.0 es :12.0
La division entre 4.0 y 3.0 es :1.3333334
Ultima operacionj realizada : division; ultimo resultado 1.3333334
4.0 es mayor que 3.0 :1
El numero 4.0NO es primo
El siguiente numero primo despues de 4.0 es : 5

```

```

JAVA_HOME="/usr/lib/jvm/jdk-17-oracle-x64"
cd /home/yosari/Escritorio/dam_m/PROGRAMACION/P00BJETOS/calcula/app; ../gradlew --configure-on-demand -x check run
Configuration on demand is an incubating feature.
> Task :app:compileJava UP-TO-DATE
> Task :app:processResources NO-SOURCE
> Task :app:classes UP-TO-DATE

```

```

> Task :app:run
La suma entre 4.0 y 3.0 es :7.0
La resta entre 4.0 y 3.0 es :1.0
La division entre 4.0 y 3.0 es :1.3333334
La multiplicacion entre 4.0 y 3.0 es :12.0
4.0 es mayor que 3.0 :1
Ultima operacionj realizada : MayorQue; ultimo resultado 1.0
El numero 4.0true
El siguiente numero primo despues de 4.0es : 5

```

```

BUILD SUCCESSFUL in 343ms
2 actionable tasks: 1 executed, 1 up-to-date

```

```
venancio@venancio:~/NetBeansProjects/calcula$ gradle clean
```

```

BUILD SUCCESSFUL in 3s
1 actionable task: 1 executed

```