

Relación de Ejercicios U2

A1_Numeros

```
public class A1_Numeros {  
  
    public static void main(String[] args) {  
        A1_Numeros1 nNumeros1 = new A1_Numeros1();  
        A1_Numeros2 nNumeros2 = new A1_Numeros2();  
        nNumeros1.start();  
        nNumeros2.start();  
    }  
  
}
```

```
4 public class A1_Numeros1 extends Thread(  
5  
6     public static final String ANSI_YELLOW = "\033[33m";  
7  
8     @Override  
9     public void run() {  
10         int suma = 0;  
11  
12         for(int i = 100; i < 1000; i++){  
13             if(i%5 == 0){  
14                 suma += i;  
15             }  
16         }  
17         System.out.printf(format: "%sLa suma de los numeros divisibles entre 100 y 1000 es: %d\n", args: ANSI_YELLOW, args: suma);  
18  
19     }  
20  
21 }  
22  
23
```

```
3  
4 public class A1_Numeros2 extends Thread(  
5  
6     public static final String ANSI_GREEN = "\u001B[32m";  
7  
8     @Override  
9     public void run() {  
10         for(byte i = 1; i <= 10; i++){  
11             System.out.printf(format: "%sMultiplos de 10: %d\n", args: ANSI_GREEN, (10 * i));  
12         }  
13     }  
14  
15 }  
16
```

```
run Single:
Multiplos de 10: 10
Multiplos de 10: 20
Multiplos de 10: 30
Multiplos de 10: 40
Multiplos de 10: 50
Multiplos de 10: 60
Multiplos de 10: 70
Multiplos de 10: 80
Multiplos de 10: 90
Multiplos de 10: 100
La suma de los numeros divisibles entre 100 y 1000 es: 98550
BUILD SUCCESSFUL (total time: 0 seconds)
```

A2_Caracteres

```
1
2 package psp.u2;
3
4 public class A2_Caracteres {
5
6     public static void main(String[] args) {
7         A2_Caracteres1 a1 = new A2_Caracteres1();
8         a1.start();
9
10        A2_Caracteres2 a2 = new A2_Caracteres2();
11        a2.start();
12    }
13
14 }
15
```

```
public class A2_Caracteres1 extends Thread {
    public static final String ANSI_TELLOW = "\033[33m";
    Random r = new Random();

    @Override
    public void run() {
        for(byte i = 0; i < 5; i++){
            System.out.printf(format: "%sCaracter (%c)\n", ANSI_TELLOW, (char) r.nextInt(origin: 97, bound: 123));
        }
    }
}
```

```

public class A2_Caracteres2 extends Thread {

    public static final String ANSI_GREEN = "\u001B[32m";

    @Override
    public void run() {
        Scanner sc = new Scanner(System.in);

        System.out.println("Escribeme una cadena de texto");
        String cadena = sc.nextLine();

        for (int i = 0; i < cadena.length(); i++) {
            if (cadena.charAt(index: i) == 'a' || cadena.charAt(index: i) == 'e' || cadena.charAt(index: i) == 'i' || cadena.charAt(index: i) == 'o' || cadena.charAt(index: i) == 'u') {
                System.out.print(ANSI_GREEN + cadena.charAt(index: i));
            }
        }
        System.out.println();
    }
}

```

```

run-single:
Character (c)
Character (e)
Character (i)
Character (c)
Character (k)
Escribeme una cadena de texto
actividad realizada
aiaaeiaaa
BUILD SUCCESSFUL (total time: 6 seconds)

```

A3_Inversa

```

public class A3_Inversa {

    public static void main(String[] args) {

        A3_Inversa1 a1 = new A3_Inversa1();
        Thread hilo = new Thread(target: a1);
        hilo.start();

        A3_Inversa2 a2 = new A3_Inversa2();
        Thread hilo2 = new Thread(target: a2);
        hilo2.start();

    }

}

```

```

public class A3_Inversal implements Runnable{

    public static final String ANSI_GREEN = "\u001B[32m";

    @Override
    public void run() {
        String cadena = "hilo";
        String ret = "";

        for(int i = cadena.length()-1; i >= 0; i--){
            ret += cadena.charAt(index: i);
            System.out.println(ANSI_GREEN + ret);
        }

    }

}

```

```

public class A3_Inversa2 implements Runnable {

    public static final String ANSI_YELLOW = "\033[33m";

    @Override
    public void run() {
        int suma = 0;
        for(int i = 500; i <= 1000; i++){
            if(i%2 != 0){
                suma += i;
            }
        }
        System.out.printf(Format: "%sLa suma de todos los numeros impares es: %d\n", args: ANSI_YELLOW, args: suma);
    }

}

```

La suma de todos los numeros impares es: 187500

o

ol

oli

olih

BUILD SUCCESSFUL (total time: 0 seconds)

A4_Divisores

```
3
4 public class A4_Divisores {
5
6     public static void main(String[] args) {
7
8         A4_Divisores1 a1 = new A4_Divisores1();
9         a1.start();
10
11         A4_Divisores2 a2 = new A4_Divisores2();
12         a2.start();
13
14     }
15
16 }
17
```

```
public class A4_Divisores1 extends Thread{
    public static final String ANSI_YELLOW = "\033[33m";

    @Override
    public void run() {
        for(int i = 1; i <= 200; i++){
            if(i % 4 == 0){
                System.out.printf(format: "%sEl numero %d es divisible entre 4\n", args: ANSI_YELLOW, args: i);
            }
        }
    }
}
```

```
public class A4_Divisores2 extends Thread{
    public static final String ANSI_GREEN = "\u001B[32m";

    @Override
    public void run() {
        for(int i = 1; i <= 200; i++){
            if(i % 4 == 0){
                System.out.printf(format: "%sEl numero %d es divisible entre 4\n", args: ANSI_GREEN, args: i);
            }
        }
    }
}
```

```
compile-single:
run-single:
El numero 4 es divisible entre 4
El numero 8 es divisible entre 4
El numero 12 es divisible entre 4
El numero 16 es divisible entre 4
El numero 20 es divisible entre 4
El numero 24 es divisible entre 4
El numero 28 es divisible entre 4
El numero 32 es divisible entre 4
El numero 36 es divisible entre 4
El numero 40 es divisible entre 4
El numero 44 es divisible entre 4
El numero 48 es divisible entre 4
El numero 52 es divisible entre 4
El numero 56 es divisible entre 4
El numero 60 es divisible entre 4
El numero 64 es divisible entre 4
El numero 68 es divisible entre 4
El numero 72 es divisible entre 4
El numero 76 es divisible entre 4
El numero 80 es divisible entre 4
El numero 84 es divisible entre 4
El numero 88 es divisible entre 4
El numero 92 es divisible entre 4
El numero 96 es divisible entre 4
El numero 100 es divisible entre 4
El numero 104 es divisible entre 4
El numero 108 es divisible entre 4
El numero 112 es divisible entre 4
El numero 116 es divisible entre 4
El numero 120 es divisible entre 4
El numero 124 es divisible entre 4
El numero 128 es divisible entre 4
El numero 132 es divisible entre 4
El numero 136 es divisible entre 4
El numero 140 es divisible entre 4
El numero 4 es divisible entre 4
El numero 8 es divisible entre 4
El numero 12 es divisible entre 4
El numero 16 es divisible entre 4
El numero 20 es divisible entre 4
El numero 24 es divisible entre 4
El numero 28 es divisible entre 4
El numero 32 es divisible entre 4
El numero 36 es divisible entre 4
El numero 40 es divisible entre 4
```

A5_Argumentos

```
public class A5_Argumentos {  
  
    public static void main(String[] args) {  
  
        for (int i = 0; i < args.length; i++) {  
            Thread hilo = new Thread(new A5_Argumentosl(args[i]));  
            hilo.start();  
        }  
    }  
}
```

```
public class A5_Argumentosl implements Runnable(  
  
    private String palabra;  
  
    public A5_Argumentosl(String palabra) {  
        this.palabra = palabra;  
    }  
  
    @Override  
    public void run() {  
        for (int i = 0; i < palabra.length(); i++) {  
            if (palabra.charAt(index: i) == 'a' || palabra.charAt(index: i) == 'e' || palabra.charAt(index: i) == 'i' || palabra.charAt(index: i) == 'o' || palabra.charAt(index: i) == 'u') {  
                System.out.printf(format: "%s", args: "\u001B[32m", args: palabra.charAt(index: i));  
            }  
        }  
    }  
}
```

A6_Sumatorio

```
2  
3 public class A6_Sumatorio {  
4  
5     public static void main(String[] args) {  
6  
7         Thread hilo = new Thread(new A6_Sumatorisl());  
8         hilo.start();  
9     }  
10 }  
11  
12
```

```

3
4 public class A6_Sumatoriol implements Runnable{
5
6     @Override
7     public void run() {
8         int suma = 0;
9         for(int i = 100; i <= 1000; i++){
10             if(i % 2 == 0){
11                 suma += i;
12             }
13         }
14         System.out.printf(format: "%sLa suma es %d\n", args: "\u001B[32m", args: suma);
15     }
16
17 }
18

```

```

Compiling 1 source file to C:\Users\Sergio\Desktop\Estudios\P
compile-single:
run-single:
La suma es 248050
BUILD SUCCESSFUL (total time: 0 seconds)

```

A7_ParImpar

```

public class A7_ParImpar {

    public static void main(String[] args) {

        A7_ParImpar1 hilo = new A7_ParImpar1();
        Thread hilo2 = new Thread(new A7_ParImpar2());

        hilo.start();
        hilo2.start();

    }

}

```



```

public class A7_ParImpar1 extends Thread {

    @Override
    public void run() {
        for (int i = 1; i <= 10; i += 2) {
            System.out.printf(format: "%sImpar: %d\n", args: "\033[33m", args: i);
        }
    }
}

```

```

public class A7_ParImpar2 implements Runnable{

    @Override
    public void run() {
        for (int i = 0; i <= 10; i += 2) {
            System.out.printf(format: "%sPar: %d\n", args: "\u001B[32m", args: i);
        }
    }
}

```

```

Par: 0
Par: 2
Par: 4
Par: 6
Par: 8
Par: 10
Impar: 1
Impar: 3
Impar: 5
Impar: 7
Impar: 9
BUILD SUCCESSFUL (total time: 0 seconds)

```

A8_Series

```
public class A8_Series {  
  
    public static void main(String[] args) {  
        A8_Series1 hilo1 = new A8_Series1();  
        A8_Series1 hilo2 = new A8_Series1();  
        A8_Series1 hilo3 = new A8_Series1();  
  
        hilo1.start();  
        hilo2.start();  
        hilo3.start();  
    }  
}
```

```
public class A8_Series1 extends Thread{  
  
    @Override  
    public void run() {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print(s: "Introduce un número entero para la serie: ");  
        int n = scanner.nextInt();  
  
        int suma = 0;  
        for (int i = 1; i <= n; i++) {  
            suma += 2 * i;  
        }  
  
        System.out.printf(format: "%sLa suma de la serie para n=%d es: %d\n", args: "\u001B[32m", args: n, args: suma);  
    }  
}
```

```
Introduce un número entero para la serie: Introduce un número entero para la serie: Introduce un número entero para la serie: 5  
2  
1  
La suma de la serie para n=1 es: 2  
La suma de la serie para n=5 es: 30  
La suma de la serie para n=2 es: 6  
BUILD SUCCESSFUL (total time: 7 seconds)
```

A9_Factorial

```
public class A9_Factorial {  
    public static void main(String[] args) {  
        if (args.length == 0) {  
            System.out.println("Debes proporcionar al menos un número como argumento.");  
            return;  
        }  
  
        for (String arg : args) {  
            try {  
                int number = Integer.parseInt(arg);  
                Thread thread = new Thread(new A9_Factorial1(numero: number));  
                thread.start();  
            } catch (NumberFormatException e) {  
                System.out.println("Error: " + arg + " no es un número válido.");  
            }  
        }  
    }  
}
```

```
2  
3 public class A9_Factorial1 implements Runnable {  
4  
5     private int numero;  
6  
7     public A9_Factorial1(int numero) {  
8         this.numero = numero;  
9     }  
10  
11     @Override  
12     public void run() {  
13         long factorial = calculateFactorial(numero);  
14         System.out.println(numero + "! = " + factorial);  
15     }  
16  
17     private long calculateFactorial(int n) {  
18         if (n < 0) {  
19             return -1;  
20         } else if (n == 0 || n == 1) {  
21             return 1;  
22         } else {  
23             long result = 1;  
24             for (int i = 2; i <= n; i++) {  
25                 result *= i;  
26             }  
27             return result;  
28         }  
29     }  
30  
31 }  
32
```