



```
package dam1delacruzgonzalez_ruth_ej030306fibonacci02;
import java.util.Scanner;
public class DAM1DeLaCruzGomez_Ruth_Ej030306Fibonacci02 {

    public static void main(String[] args) {

        int num1 = 0;
        int num2 = 1;
        int resultado;
        String num1text;
        String num2text;

        Scanner entrada = new Scanner(System.in);
        System.out.println(x: "Escriba un numero minimo: ");
        num1text = entrada.nextLine();
        num1 = Integer.valueOf(s: num1text);

        System.out.println(x: "Escriba un numero maximo: ");
        num2text = entrada.nextLine();
        num2 = Integer.valueOf(s: num2text);

        while (num2 <= 1000) {
            num1 = (num1 + num2);
            resultado = num1;
            System.out.println(x: resultado);
            System.out.print(s: "\n");

            num2 = (num1 + num2);
            resultado = num2;
            System.out.print(i: resultado);
            System.out.println(x: "\n");
        }
    }
}
```

### BLOQUE 1 : Nombres

Tiene variables con nombres significativos como `int : numero1` , que seria una variable entera con un valor 0 . A la hora de leer el código tiene nombres fácil de pronunciar como en este caso seria `String` que sabemos que es variable referenciada , y usa nombres que puedan ser buscados ya que tiene constantes `while ( numero2 es menor o igual que 1000)`. Las variables son nombres significativos como podemos observar en la variable `int = 0`

### BLOQUE 2 : Funciones

tiene funciones pequeñas ya que ocupan como mucho 4 líneas donde te describe toda la función que va a realizar ese programa , cada función realiza única cosa y ademas tiene asociada su única variable como por ejemplo `num1 = Integer.valueOf(num1text)`“el numero 1 convierte la cadena del numero1texto en un numero entero” En este código no se utiliza los `switch` ya que son complicados de usar ya que normalmente no realizan una única cosa si no que realizan mas y ya no estaría cumpliendo una de las reglas del Clean Code. En este código no existe ninguna repetición de código ya que solo realiza una función que en este caso es un `while`.

### BLOQUE 3 : Comentarios

En este código se añaden comentarios para que a la hora de leer el código sea mucho más fácil de entenderlo ya que te explican que va a realizar el código pero no te explican como lo hace.

Normalmente se utilizan códigos auto explicativos ya que es mucho más sencillo que el código explique por sí solo lo que hace que se utilicen comentarios que algunas veces mienten

```
DeLaCruzGomez_Ruth_Ej030306Fibonacci02.java x
History
public class DAM1DeLaCruzGomez_Ruth_Ej030306Fibonacci02 {

    public static void main(String[] args) {

        int num1 = 0;
        int num2 = 1;
        int resultado;
        String num1text;
        String num2text;

        Scanner entrada = new Scanner(source: System.in);
        System.out.println(x: "Escriba un numero minimo: "); // pide por pantalla ingresar un numero minimo
        num1text = entrada.nextLine();
        num1 = Integer.valueOf(s: num1text);

        System.out.println(x: "Escriba un numero maximo: "); // pide por pantalla ingresar un numero maximo
        num2text = entrada.nextLine();
        num2 = Integer.valueOf(s: num2text);

        while (num2 <= 1000) { // condicional donde ingresa un numero menor o igual que 1000
            num1 = (num1 + num2);
            resultado = num1;
            System.out.println(x: resultado);
            System.out.print(s: "\n");

            num2 = (num1 + num2); // la suma de ambos numeros es el resultado de el numero 2
            resultado = num2;
            System.out.print(i: resultado);
            System.out.println(x: "\n");
        }
    }
}
```