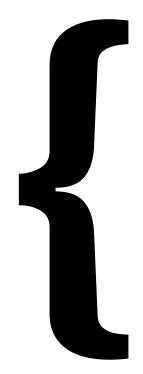


# Programación 2

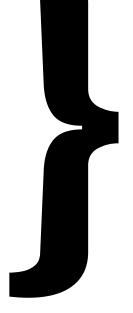
#### UNRN

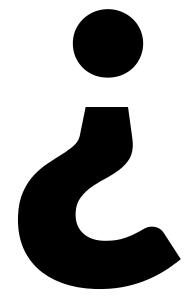
Universidad Nacional de **Río Negro** 

Martín René Vilugrón mrvilugron@unrn.edu.ar



# Recuerden, la entrega es el canal # # entrega





# Como les fue con el TP





### Constantes

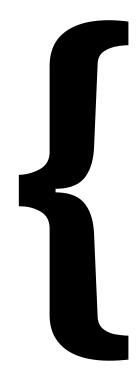


### final tipo NOMBRE = valor;



#### final int MAXIMO = 10000;





### Por suerte no son tan necesarios como en C





# Tipos no tan primitivos



# Ya tuvieron un contacto con String

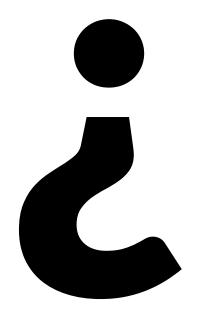
#### Números

Byte byte short Short int □ Integer long Long Float float double Double



# Revisen en la documentación que tienen disponible





## pero... ¿por qué?





### SunSPOT



2007

https://en.wikipedia.org/wiki/Sun\_SPOT



### Tambien tenemos



# Otros números BigInteger BigDecimal



pero...

# No funcionan los operadores





### arrays



# Una secuencia mutable y de elementos de un tipo con un largo fijo



## tipo[] nombre;



$$int[] arr = {1, 2, 3};$$



int[] arr = new int[10];



### int[][] matrix;



$$int[][]$$
 matrix = {{1,2},{3,4}};



int[][] matrix = new int[10][10];

# En el TP2 vamos a darles uso



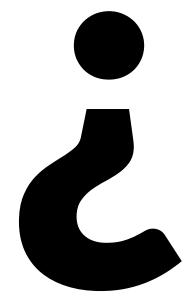


# A probar nuestro código (el primer enfoque)



#### La estructura de una función de pruebas

```
public static void test_suma_positivos(){
    int argumento1 = 10;
    int argumento2 = 20;
    int esperado = 30; // argumento1 + argumento2 esta bien
    int resultado = suma_lenta(argumento1, argumento2);
```



## ¿Cuantas funciones hacen falta?



# Suficientes para ejercitar todas las decisiones (if y lazos)





# Midiendo el tiempo de ejecución



#### A medir tiempo de ejecución

```
long arranque = System.nanoTime();
metodoAProbar();
long parada = System.nanoTime();
long tiempo = (parada - arranque);
//dividir por 1000000 para obtener milisegundos.
```



#### unrn.edu.ar







