

# Tema 6 – S2

## Contenidos:

2. La iteración como estrategia de diseño: fases y elementos
  - Estrategias para descubrir la estructura iterativa del problema
  - Mecánica y Semántica del bucle `while`
  - Ejemplos y algunos ejercicios no resueltos en la Sesión 1
  - Resolución de Recurrencias usando un bucle `while`: término, serie y error

# La iteración como estrategia de diseño

## (Examen PoliformaT) Mecánica y semántica del while: ejercicio 2

Sea `n` una variable `int` inicializada a un valor no negativo. Sin usar el método `Math.pow`, completa el siguiente bucle para que muestre por pantalla los `n` primeros **cuadrados perfectos**, de 1 en adelante y en líneas separadas. Por ejemplo, si `n = 4`, se escribirán en líneas separadas 1, 4, 9 y 16

```
int i =  ;  
while (  ) {  
    System.out.println( );  
    i++;  
}
```

actualizar contador

### ¿Por dónde empezar?

Descubre la estrategia subyacente “a partir de ejemplos”



Solución del ejercicio 2 del Examen Poliformat **Actividad Tema 6: Mecánica y Semántica del bucle while** (contiene anexo el slide-show)

# La iteración como estrategia de diseño

## (Examen PoliformaT) Mecánica y semántica del while: ejercicio 3

Sea `n` una variable `int` inicializada a un valor no negativo. Sin usar el método `Math.pow`, completa el siguiente bucle para que muestre por pantalla los primeros cuadrados perfectos **menores o iguales que `n`**, de 1 en adelante y en líneas separadas. Por ejemplo, si `n = 5`, se escribirán en líneas separadas **1** y **4**

```
int i =  ;  
while (  ) {  
    System.out.println( );  
    i++;  
}
```

**actualizar contador**

**¿Por dónde empezar?**

Modifica la estrategia existente para adaptarla al ejercicio

Solución del ejercicio 3 del **Examen Poliformat Actividad Tema 6: Mecánica y Semántica del bucle while** (contiene anexo el slide-show)

# La iteración como estrategia de diseño

## (Examen PoliformaT) Mecánica y semántica del while: ejercicio 4

Sea  $n$  una variable `int` inicializada a un valor no negativo. Completa el bucle del siguiente método para que calcule el factorial de  $n$  tal y como muestra su traza para  $n = 5$

```
// PRECONDICIÓN:  $n \geq 0$ 
public static int factorial(int a) {
    int i = ; int fact = ;
    while (  ) {
        
    }

    return fact;
}
```

i	fact
0	1
1	1
2	2
3	6
4	24
5	120

¿Estrategia?

¿Por dónde empezar?

Ya tienes los ejemplos de los que aprender: la traza que se te da



Solución del ejercicio 4 del Examen Poliformat *Actividad Tema 6: Mecánica y Semántica del bucle while* (contiene anexo el slide-show)

# La iteración como estrategia de diseño - Ejercicio nº 2 Capítulo 8 del libro

Diseña un método que devuelva la suma de los n primeros números naturales

// **PRECONDICIÓN:**  $n \geq 0$

```
public static int sumarHasta(int n) {
```

**Estrategia:**  $i$  es el último nº sumado

```
    int res = ; int i = ;
```

```
    while (  ) {
```

```
          
    }
```

```
    return res;
```

```
}
```



**¿Por dónde empezar?**

Descubre qué significa la estrategia que se te da:  $i$  es el último nº sumado

# La iteración como estrategia de diseño

## Ejercicio propuesto: nº 2 Capítulo 8 del libro con la estrategia indicada

Diseña un método que devuelva la suma de los n primeros números naturales

```
// PRECONDICIÓN:  $n \geq 0$ 
```

```
public static int sumarHasta(int n) {
```

**Estrategia:** i es el siguiente nº a sumar

```
    int res = ; int i = ;
```

```
    while (  ) {
```

```
        
```

```
    }  
    return res;
```

Tienes el código solución en el programa Ejercicio2Capitulo8Libro del package **sesion1** del proyecto BlueJ **ejercicios – Tema 6**

# La iteración como estrategia de diseño

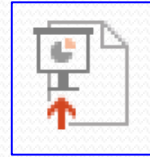
## Ejercicio propuesto: nº 7 del Capítulo 8 del libro

Diseña un método que devuelva  $a^n$ , siendo  $a > 0$  y  $n \geq 0$ , **SIN** usar `Math.pow` a partir de la definición matemática de la función potencia

```
// PRECONDICIÓN:
public static int potenciaSinUsarMath(int a, int n) {
    int res = ; int i = ;
    while (  ) {
        
    }
    return res;
}
```

# Resolución de Recurrencias usando un bucle while

## Término y serie: Ejemplo 3



**Completa** el siguiente método para que devuelva el término  $n$ -ésimo de la sucesión  $a_1 = 7, a_i = a_{i-1} + 2, i \geq 2$  (Recurrencia que expresa el cálculo de un término para todo  $i \geq 1$ )

```
// PRECONDICIÓN:  $n \geq 1$ 
public static int calcularTermino(int n) {
    int i = ; int res = ;
    while (  ) {
        
    }

    return res;
}
```



# Resolución de Recurrencias usando un bucle while

## Término y serie: Ejercicio Nº 2 Transparencias



**Modifica** el bucle del método del Ejemplo 3 para que devuelva el término  $n$ -ésimo de esta otra sucesión:  $a_1 = 7$ ,  $a_i = a_{i-1} + i$ ,  $i \geq 2$

```
// PRECONDICIÓN:  $n \geq 1$ 
public static int calcularTermino(int n) {
    int i = ; int res = ;
    while (  ) {
        
    }

    return res;
}
```

# Resolución de Recurrencias usando un bucle while

## Término y serie: Ejercicio Nº 3 Transparencias



**Completa** los huecos para que el siguiente método devuelva la **suma** de los **n** primeros términos de la sucesión  $a_1 = 7$ ,  $a_i = a_{i-1} + i$ ,  $i \geq 2$

**Estrategia:** Repetir  $n - 1$  veces { termino = termino + i; }

**// PRECONDICIÓN:**  $n \geq 1$

```
public static int calcularSerie(int n) {  
    int i = 1; int termino = 7; // calculado término  $a_1$   
    int res = ;  
    while ( i != n ) { //  $i < n$   
        i++;  
        termino = termino + i; // calculado término  $a_i$   
    }  
    // PARADA :  $i = n$  // calculado término  $a_n$   
    return res;  
}
```

# Resolución de Recurrencias usando un bucle while

## Término y serie - Ejercicio Nº 4 Transparencias



Completa el siguiente método para que devuelva el valor (double) de la siguiente serie para un cierto  $n \geq 1$

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1} + \frac{1}{n}$$

**// PRECONDICIÓN:  $n \geq 1$**

```
public static double calcularSerie(int n) {  
    int i = ; double termino = , res = ;  
    while (  ) {  
          
    }  
    return res;  
}
```

**Estrategia:**

## + Ejercicios propuestos

- **Nº 5 Transparencias:** diseña un programa NPrimerosMúltiplosDe que muestre por pantalla, en líneas separadas, los n primeros múltiplos de x, siendo x y n enteros positivos. Por ejemplo, si  $x = 9$  y  $n = 4$ , se mostraría
  - $9 \times 1 = 9$
  - $9 \times 2 = 18$
  - $9 \times 3 = 27$
  - $9 \times 4 = 36$
- **Nº 6 Transparencias:** escribe un método Java que calcule el valor aproximado de  $\pi$  para un cierto  $n \geq 1$  a partir de la fórmula 
$$\frac{\pi^2}{6} \approx \sum_{i=1}^{\infty} \frac{1}{i^2}$$
- **Nº 7 Transparencias:** escribe un método Java que, dado  $n \geq 0$ , devuelva el n-ésimo término de la sucesión de Fibonacci:  $a_0 = 0$  y  $a_1 = 1$  ;  $a_i = a_{i-1} + a_{i-2}$  si  $i \geq 2$
- **Nº 9 y nº 10 del Capítulo 8 del libro de la asignatura**