

## UNIDAD TEMÁTICA 4: Árboles Binarios y recursividad

### PRACTICOS DOMICILIARIOS INDIVIDUALES - FORMULACION DE SEUDOCÓDIGO

#### Ejercicio #1

1. Desarrolla en pseudocódigo, en forma recursiva, un algoritmo para calcular el factorial de un cierto número entero que se pasa como parámetro (descrito en el slide #4 de la ppt de clase sobre recursividad).
  - Identifica claramente el caso base y la sentencia que lo contempla.
  - ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base?
2. Analiza el orden del tiempo de ejecución del algoritmo.
3. Escribe, en lenguaje natural, casos de prueba apropiados para testear este algoritmo
4. Implementa el algoritmo (en JAVA) y pruébalo (mediante test – casos apropiados):
  - ¿Qué sucede si el número es negativo?
  - Verifica que `factorial(4)`, `factorial(5)`, y `factorial(0)` produzcan los resultados esperados

#### Ejercicio #2

1. Desarrolla en pseudocódigo, en forma recursiva, el algoritmo

*Algoritmo SumaLineal(A, n),*

que se describe en el slide #11 de la ppt de clase sobre recursividad.

- Identifica claramente el caso base y la sentencia que lo contempla.
  - ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base?
2. Analiza el orden del tiempo de ejecución del algoritmo
  3. Escribe, en lenguaje natural, casos de prueba apropiados para testear este algoritmo
  4. Implementa en JAVA los test-cases correspondientes al punto 3.
    - ¿Qué sucede si el parámetro `n` es negativo?
    - ¿Qué sucede si el vector `A` está vacío?
  5. Implementa el algoritmo (en JAVA) y pruébalo con los test-cases. Luego repara eventuales errores.

### Ejercicio #3

1. Desarrolla en pseudocódigo, en forma recursiva, un algoritmo para calcular la potencia de un número. El mismo ha de recibir como parámetros el número y el exponente (ver el slide #13 de la ppt de clase sobre recursividad).
  - Identifica claramente el caso base y la sentencia que lo contempla.
  - ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base
2. Analiza el orden del tiempo de ejecución del algoritmo.
3. Implementa el algoritmo (en JAVA) y pruébalo:
  - ¿Tu algoritmo soporta números reales o sólo enteros – para ambos parámetros?
  - ¿qué sucede si uno, otro o ambos parámetros son negativos?

### Ejercicio #4

1. Desarrolla en pseudocódigo, en forma recursiva, un algoritmo para invertir los componentes de un vector pasado por parámetro, entre dos índices indicados también pasados como parámetros. (ver el slide #12 de la ppt de clase sobre recursividad).
  - Identifica claramente el / los caso(s) base y la(s) sentencia(s) que lo contempla(n).
  - ¿Puedes verificar que siempre el algoritmo progresará hacia el caso base?
2. Analiza el orden del tiempo de ejecución del algoritmo.
3. Implementa el algoritmo (en JAVA) y pruébalo:
  - Crea un pequeño vector y prueba el algoritmo. Prueba situaciones de borde (extremos), parámetros fuera de rango, vector vacío, vector con sólo un elemento, etc.