2. Evaluación de expresión aritmética (1,5 puntos)

Djikstra presentó un algoritmo para la evaluación de una expresión aritmética. Por ejemplo: `(1+((2+3)*(4*5)))`

Se pide implementar el siguiente método:

```
public double evaluar(String exp)
 // Precondición: La expresión aritmética es correcta.
 //      La expresión está TOTALMENTE parentizada, es decir,
 //                por cada pareja de operandos se tienen paréntesis
 //      Todos los operadores son binarios (de la forma "X OPERADOR Y")
 // Postcondición: el resultado es el valor de la expresión
```

Dada la expresión ejemplo anterior, el resultado será 101.

La expresión aritmética parentizada se evaluará de la siguiente manera:

- Se usan 2 pilas, una para operadores y otra para operandos
- Los caracteres de la secuencia de entrada se tomarán de uno en uno, haciendo lo siguiente a cada paso:
  - Si es un paréntesis izquierdo, no se hace nada
  - Meter operandos en la pila de operandos
  - Meter operadores en la pila de operadores
  - Si es un paréntesis derecho, entonces
    - coger 2 operandos de la pila de operandos
    - coger un operador de la pila de operadores
    - calcular el resultado y meter en la pila de operandos

Ejemplo

Dada la siguiente cadena: `(1+((2+3)*(4*5)))`

Pila de operadores	
Pila de operandos	

- Paréntesis izquierdo "(" → no hacer nada

`Cadena: 1+((2+3)*(4*5)))`

- "1" → meter en pila de operandos

Pila de operadores	
Pila de operandos	1

`Cadena: +((2+3)*(4*5)))`

- "+" → meter en pila de operadores

Pila de operadores	+
Pila de operandos	1

Cadena: ((2+3)*(4*5)))

- Paréntesis izquierdo "(" → no hacer nada
- Paréntesis izquierdo "(" → no hacer nada

Cadena: 2+3)*(4*5)))

- "2" → meter en pila de operandos

Pila de operadores	+
Pila de operandos	1 2

Cadena: +3)*(4*5)))

- "+" → meter en pila de operadores

Pila de operadores	+ +
Pila de operandos	1 2

Cadena: 3)*(4*5)))

- "3" → meter en pila de operandos

Pila de operadores	+ +
Pila de operandos	1 2 3

Cadena:)*(4*5)))

- ")" → realizar operación

Pila de operadores	+
Pila de operandos	1 5

Cadena: *(4*5)))

- "*" → meter en pila de operadores

Pila de operadores	+ *
Pila de operandos	1 5

Cadena: (4*5)))

- Paréntesis izquierdo "(" → no hacer nada

Cadena: 4*5)))

- "4" → meter en pila de operandos

Pila de operadores	+ *
Pila de operandos	1 5 4

Cadena: *5)))

- "*" → meter en pila de operadores

Pila de operadores	+ * *
Pila de operandos	1 5 4

Cadena: 5)))

- "5" → meter en pila de operandos

Pila de operadores	+ * *
Pila de operandos	1 5 4 5

Cadena:)))

- ")" → realizar operación

Pila de operadores	+ *
Pila de operandos	1 5 20

Cadena:))

- ")" → realizar operación

Pila de operadores	+
Pila de operandos	1 100

Cadena:)

- ")" → realizar operación

Pila de operadores	
Pila de operandos	101

¡El resultado está en la pila de operandos!

Implementar el algoritmo y calcular su costo de manera razonada.

Nota: para calcular el valor de un número, dado, se puede usar el siguiente método (no hay que implementarlo):

```
public Double parse(String s)
// Post: dado un string que contiene un número entero, devuelve ese entero.
```