



UNIVERSIDAD AUTÓNOMA DE ZACATECAS

Practica 9: Pilas

Estudiante:

José Francisco Hurtado Muro

Profesor:

Dr. Aldonso Becerra Sánchez

April 7, 2025

Tabla de Contenidos

1	Introducción	2
2	Actividad 2:	3
2.0.1	Actividad 1: Programa de Encriptación	3
2.0.2	Actividad 2: Programa de Desencriptación	5
3	Actividad 3:	6
4	Código Agregado - UML	8
5	Pre-evaluación del Alumno	9
6	Conclusión	9
7	Referencias:	9

1 Introducción

"Las pilas son estructuras de datos fáciles de entender, sin embargo, sus aplicaciones muchas veces no son tan sencillas de visualizar en la práctica. Las pilas facilitan muchos procesos, un caso común son el manejo de notaciones infija, prefija y postfija."

2 Actividad 2:

Una manera sencilla de encriptar mensajes es colocar paréntesis de manera arbitraria, y todo lo que está dentro de un paréntesis ponerlo al revés (invertir el contenido). Por ejemplo, “Curso de Informática” se puede encriptar como “Cur(os) de I(rofn)mática”.

2.0.1 Actividad 1: Programa de Encriptación

Escriba un programa que dado un mensaje en texto plano (ej. “Curso de Informática”), encripte el mensaje pidiendo:

- Número de agrupaciones (pares de paréntesis) a usar
- Para cada agrupación:
 - Posición inicial en la cadena (ej. 4 y 11)
 - Tamaño del contenido a encerrar (ej. 2 para “os”, 4 para “rofn”)

Restricciones:

- Las posiciones no deben empalmarse
- Se debe validar que las posiciones y tamaños sean válidos con respecto a la longitud de la cadena
- Si no se cumplen las reglas, el programa no debe encriptar

Ejemplo:

- Entrada: “Curso de Informática”
- 3 agrupaciones
- Posiciones: 4 y 11
- Tamaños: 2, 4, 3
- Salida: “Cur(os) de I(rofn)má(cit)a”

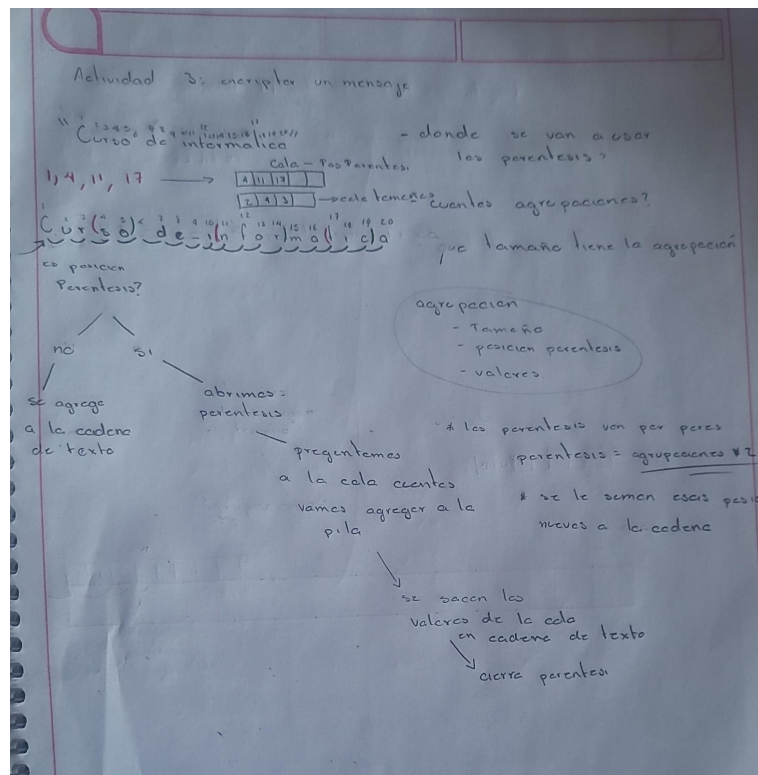


Figure 1: Análisis: Encriptación

Explicación

Para este caso lo que se hizo fue recuperar letra por letra de una cadena dada y es guardada en una cadena nueva, pero si la posición de la cadena concuerda con un posición de un paréntesis dada por una cola, se agrega un paréntesis de apertura a la cadena y las letras siguientes se agregan a una pila, para esto a una cola donde se guardaron los valores de los tamaños de las agrupaciones dadas se le pregunta cuantos caracteres almacenara, por lo cual agrega esa cantidad de caracteres y una vez ingresada la cantidad de caracteres, se regresan a la cadena resultado siguiendo las normas de una pila, una vez de terminar de regresar los valores se cierra el paréntesis y se procesan los siguientes caracteres de la misma forma como podemos ver en la figura 1 y figura 2.

```

Ingrese el mensaje que desea encriptar:
curso de informatica
¿Cuántas agrupaciones tendrá el mensaje?
3
Ingrese la posición del paréntesis de apertura para la agrupación 1:
4
Ingrese la posición del paréntesis de apertura para la agrupación 2:
11
Ingrese la posición del paréntesis de apertura para la agrupación 3:
17
Ingrese el tamaño de la agrupación 1:
2
Ingrese el tamaño de la agrupación 2:
4
Ingrese el tamaño de la agrupación 3:
3
Mensaje encriptado: cur(os) de i(rofn)ma(cit)a
Seleccione una opción:
1. Encriptar un mensaje
2. Desencriptar un mensaje
3. Salir
  
```

Figure 2: Funcionamiento: Programa de Encriptación

2.0.2 Actividad 2: Programa de Desencriptación

Escriba un programa que permita desencriptar mensajes siguiendo las mismas reglas del punto 1, identificando los segmentos entre paréntesis y revirtiendo su contenido.

Ejemplo:

- Entrada: "Cur(os) de I(rofn)má(cit)a"
- Salida: "Curso de Informática"

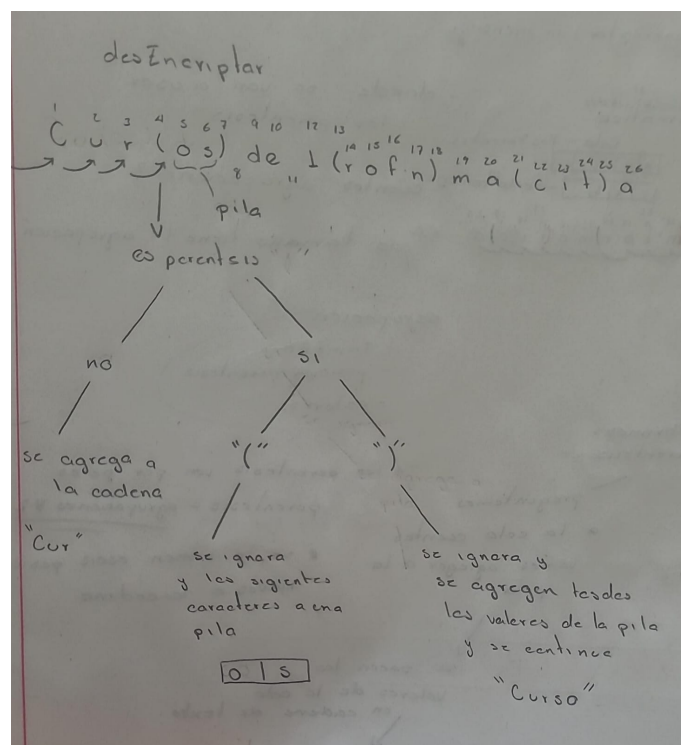


Figure 3: Análisis: desEncriptación

Explicación

En este caso lo que se hizo fue que de un mensaje encriptado, fuéramos extrayendo carácter por carácter, pero si nos encontramos un paréntesis "(" todos los valores siguientes son agregados a una pila hasta que se encuentra un paréntesis ")" y al encontrarlo se vacían todos los valores de la pila en la cadena resultado donde se ignoran los paréntesis, pero si el carácter no formaba parte de estas agrupamientos delimitada por los paréntesis, directamente se agregaban a la cadena resultado, como podemos ver en la figura 4.

```

Seleccione una opción:
1. Encriptar un mensaje
2. Desencriptar un mensaje
3. Salir
1
Ingrese el mensaje que desea encriptar:
curso de informatica
¿Cuántas agrupaciones tendrá el mensaje?
3
Ingrese la posición del paréntesis de apertura para la agrupación 1:
4
Ingrese la posición del paréntesis de apertura para la agrupación 2:
11
Ingrese la posición del paréntesis de apertura para la agrupación 3:
17
Ingrese el tamaño de la agrupación 1:
2
Ingrese el tamaño de la agrupación 2:
3
Ingrese el tamaño de la agrupación 3:
4
Mensaje encriptado: cur(os) de i(ofn)rma(acit)
Seleccione una opción:
1. Encriptar un mensaje
2. Desencriptar un mensaje
3. Salir
2
Ingrese el mensaje encriptado que desea desencriptar:
cur(os) de i(ofn)rma(acit)
Mensaje desencriptado: curso de informatica
Seleccione una opción:
1. Encriptar un mensaje
2. Desencriptar un mensaje
3. Salir

```

Figure 4: Funcionamiento: Programa de Encriptación y desEncriptación

3 Actividad 3:

Hacer un programa que funcione como una calculadora. Este programa permitirá funcionar como una calculadora, debe recibir una expresión aritmética en una caja de texto (o el teclado), por ejemplo de la forma:

$$(a - 3.6) * f / 2^{(c+1)}$$

Asuma que la entrada es una expresión válida, es decir no deberá realizar validaciones de si está bien formada la expresión (eso lo deberá hacer otro módulo en otro programa, que ahorita no se está pidiendo; que pudiera usarse independientemente del funcionamiento del programa que se le pide a usted en esta actividad práctica).

El programa deberá recibir de entrada esta cadena con la expresión aritmética. Posteriormente se deberá obtener el resultado de la operación ejecutada (evaluar la expresión).

Consideraciones de diseño:

- Considere variables de un solo carácter.
- Revise la expresión aritmética para checar si tiene variables (de una sola letra); en caso de que tenga, pida los valores que esas variables debe tener para sustituirlas por su contenido.
- Considere que los números son enteros y decimales, y de cualquier longitud. Ejemplos: 456, 56, 4, 44443, 4.5, 9.99.
- Al final el programa deberá obtener el resultado de la expresión infija ingresada, para ello deberá convertir la cadena en una expresión “postfija” o “prefija” (usando pilas), para posteriormente realizar la evaluación de dicha expresión convertida (usando pilas).

Ejemplo:

Entrada:	$(a - 2.0) * f / 2^{(c-8)}$ donde $a = 1$, $f = 3$, $c = 12.0$
Salida:	-0.1875

Nota:

Recuerde que no puede usar librerías de Java que automaticen ningún proceso de manipulación de grupos de datos, en este caso de cadenas. Por decir, están prohibidos los usos de expresiones regulares.

Implemente el programa según las especificaciones anteriores. Debe incluir:

- Código fuente completo en Java
- Captura de pantalla del programa funcionando
- Explicación del algoritmo utilizado

4 Código Agregado - UML

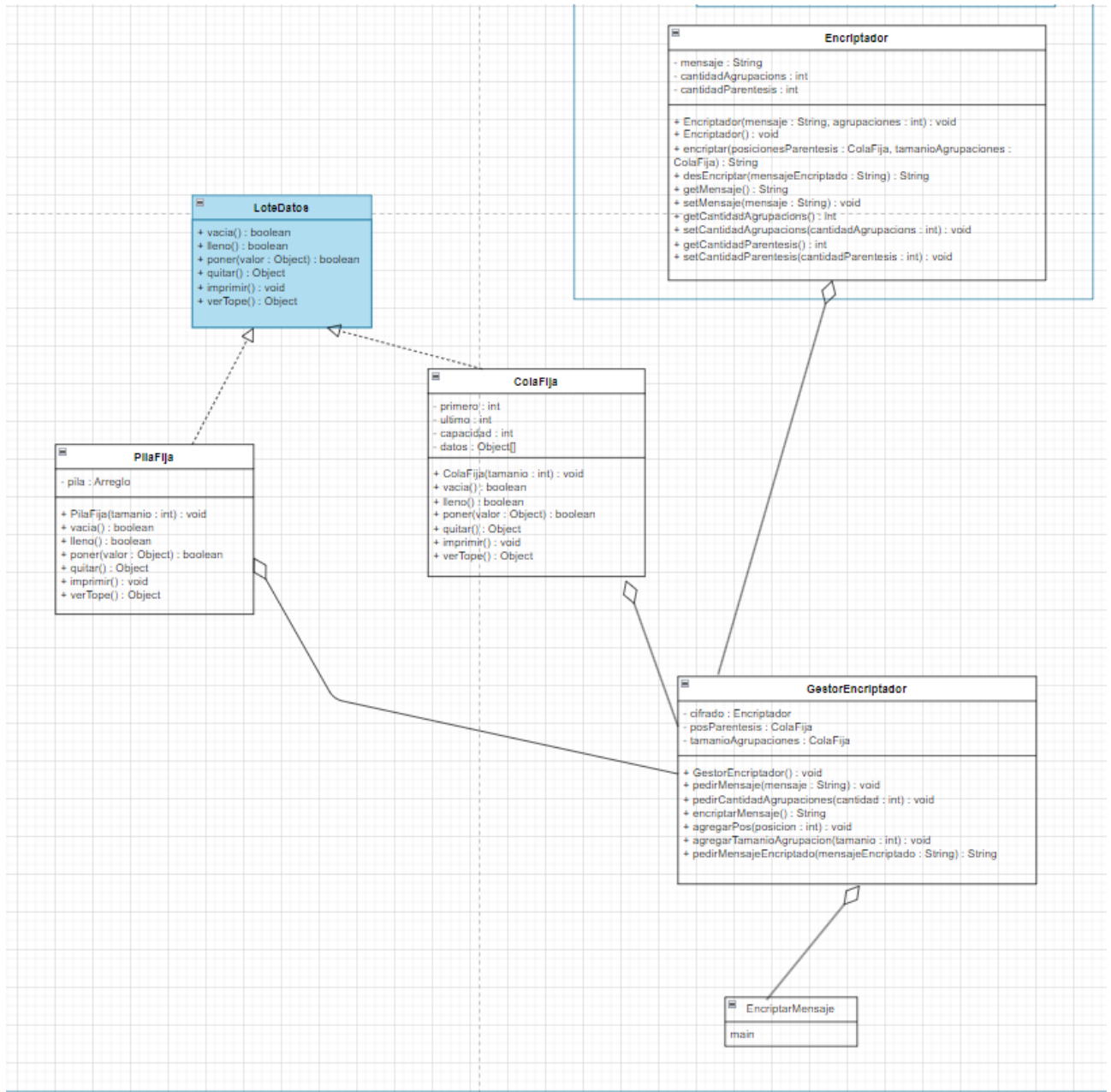


Figure 5: UML

5 Pre-evaluación del Alumno

Criterio	Evaluación
Cumple con la funcionalidad solicitada	Sí
Dispone de código auto-documentado	Sí
Dispone de código documentado a nivel de clase y método	Sí
Dispone de indentación correcta	Sí
Cumple la POO	Sí
Dispone de una forma fácil de utilizar el programa para el usuario	Sí
Dispone de un reporte con formato IDC	Sí
La información del reporte está libre de errores de ortografía	Sí
Se entregó en tiempo y forma la práctica	No
Incluye el código agregado en formato UML	Sí
Incluye las capturas de pantalla del programa funcionando	Sí
La práctica está totalmente realizada (especifique el porcentaje completado)	70%

Table 1: Evaluación de la práctica

6 Conclusión

El uso de pilas y colas para el manejo de datos es bastante importante ya que con esto podemos hacer operaciones respetando valores de precedencia los cuales son importantes para la resolución de problemas como lo son las operaciones aritméticas o algebraicas.

7 Referencias:

- Cairo, Osvaldo; Guardati, Silvia. *Estructura de Datos, Tercera Edición*. McGraw-Hill, México, Tercera Edición, 2006.
- Mark Allen Weiss. *Estructura de datos en Java*. Ed. Addison Wesley.
- Joyanes Aguilar, Luis. *Fundamentos de Programación. Algoritmos y Estructuras de Datos*. Tercera Edición, 2003. McGraw-Hill.