



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FES ARAGÓN
INGENIERÍA EN COMPUTACIÓN

ESTRUCTURAS DE DATOS
CON JAVA

Programas

Autor: J. Eduardo Sánchez Posadas
e-Mail: jeduardo.schz@gmail.com

5 de julio de 2018

ID	Tema	Actividad	Descripción
1 _i	Estructuras básicas	Reseña artículo	Realizar una reseña de una cuartilla sobre el artículo
2	Estructuras básicas	Estándar IEEE754	Realizar una clase que convierta números al estándar IEEE754
3	Estructuras lineales	Implementación de ListaSimple	Clase donde se implementan todos los métodos de una lista Ligada simple
4	Estructuras lineales	Pruebas de ListaSimple	Tres pruebas de la lista simple: Integers, String, clase Automóvil
5	Estructuras lineales	Paréntesis	Verificar que los paréntesis en una expresión estén balanceados
6	Estructuras lineales	Signos de agrupamiento	Verificar que los paréntesis en una expresión estén balanceados y colocados de manera correcta
7!	Estructuras lineales	Verificar operaciones	Verificar que una operación esté bien escrita
8	Estructuras lineales	Conversión infijo-posfijo	Convertir una expresión infija a una prefija, y viceversa
9	Estructuras lineales	Torres de Hanoi	Simular con pilas el juego de las torres de Hanoi
10	Estructuras lineales	Supermercado	Simular con una cola la fila de un supermercado
11 _i	Estructuras no lineales	Preguntas	Responder a las preguntas
12	Estructuras no lineales	Grafo Metro	Implementar en una matriz de un grafo no dirigido de la red del metro e imprimirlo en un archivo de texto
13	Estructuras no lineales	Arboles de expresion	Dar los arboles de expresión para las operaciones dadas

1. Apéndice A. Diagrama de clases para la actividad 10

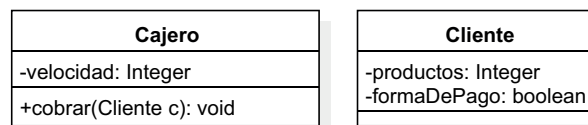


Figura 1: Diagrama de clases

```
1 public class Cajero {
2
3     private Integer velocidad;
4
5     public Cajero() {
6         //Velocidad por cada 10 productos
7         this.velocidad = new Random().nextInt(10) + 1;
8     }
9
10    public Integer getVelocidad() {
11        return velocidad;
12    }
13
14    public void setVelocidad(Integer velocidad) {
15        this.velocidad = velocidad;
16    }
17
18    public void cobrar(Cliente c) {
19        int tiempoPago = c.getFormaDePago().equals(true) ? 30 : 10;
20        long tiempo = ((c.getProductos() / 10) * velocidad) + tiempoPago;
21    }
22 }
23
```

Figura 2: Código clase Cajero

```
1 public class Cliente {
2
3     private Integer productos;
4     private Boolean formaDePago; //True=efectivo->30, False=tarjeta->10
5
6     public Integer getProductos() {
7         return productos;
8     }
9
10    public void setProductos(Integer productos) {
11        this.productos = productos;
12    }
13
14    public Boolean getFormaDePago() {
15        return formaDePago;
16    }
17
18    public void setFormaDePago(Boolean formaDePago) {
19        this.formaDePago = formaDePago;
20    }
21
22
23    public Cliente() {
24        this.productos = new Random().nextInt(30) + 1;
25        this.formaDePago = new Random().nextBoolean();
26    }
27
28 }
```

Figura 3: codigo clase **Cliente**

2. Apéndice B. Preguntas de la actividad 11

1. ¿Cómo saber si el grafo tiene bucles o ciclos?
2. ¿Cómo saber el número de vértices?
3. ¿Cómo saber si hay algún vértice sin adyacentes?
4. ¿Cómo calcular el número de adyacentes de un vértice?
5. ¿Cómo sería la matriz de adyacencia de un grafo no dirigido?
6. ¿Cómo representar un grafo con aristas ponderadas usando una matriz de adyacencias?
7. Dibujar el grafo de la siguientes matrices de adyacencia e indicar qué tipo de grafo representan:

•	0	1	2	3	4	5
0	false	true	false	false	false	false
1	false	true	false	true	true	false
2	false	false	false	false	true	false
3	true	true	false	false	false	false
4	false	false	false	true	false	false
5	false	false	true	false	true	false

•	A	B	C	D	E
A	0	1	0	0	1
B	1	1	1	0	0
C	0	1	0	1	1
D	0	0	1	0	0
E	1	0	1	0	1

•	1	2	3	4	5
1	-	40	25	-	56
2	-	-	-	95	-
3	10	-	-	-	-
4	-	-	14	-	-
5	-	30	36	75	-

3. Apéndice C. Mapa para la actividad 12

SISTEMA DE TRANSPORTE COLECTIVO

Red del Metro



4. Apéndice D. Expresiones para la actividad 13

1. $a + b + c + d$
2. $a * (b - c)$
3. $a + a * (b - c) + (b - c) * d$
4. $i = i + 10$
5. $((x + y) - ((x + y) * (x - y))) + ((x + y) * (x - y))$
6. $a + b + (a + b)$
7. $a + b + a + b$
8. $a + a + ((a + a + a + (a + a + a + a)))$