

// Sistema de gestión de entrada a museos

José Salinas 61886

Sofia Hanna Feilbogen 61889

Sofia Gonzalez Del Solar 62292

Juan Leal de Ibarra 59005

Materia: “Estructura de Datos y Programación”

Año: 2021, 2do cuatrimestre

Introducción;

- * Programa para Gestión de Museos
- * Trabajo de investigación sobre carga de datos a programa
- * Dos tipos de usuarios: Administrador y Cliente
- * Uso de múltiples Estructuras de datos, tipos de instrucciones y Clases
- * Clases: Cliente, Listacliente, Listaticket, Main, Museos, Nodocliente, Nodoticket, Read, Ticket

Indice

Step 01 Carga de datos

Step 02 Recorrido del programa viendo la consigna

Step 03 Detalles a destacar (Decisiones de estructuras)

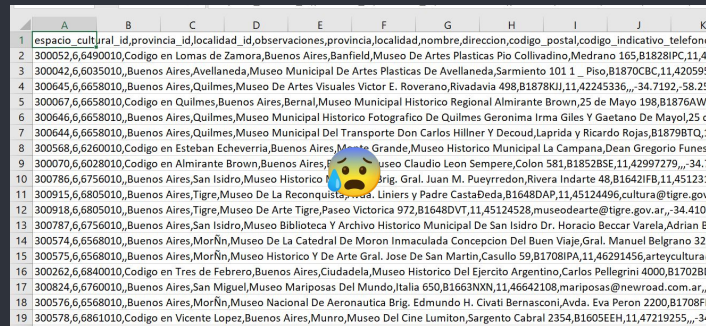
Step 04 Experiencia en el proceso

01 {

[Carga de datos]

< Una de las tareas más desafiantes con las que se encontró el equipo: ¿Cómo llevar información de un CSV, a nuestro programa Java? >

}



	A	B	C	D	E	F	G	H	I	J	K
1	espacio_cultural_id	provincia_id	localidad_id	observaciones	provincia	localidad	nombre	direccion	codigo_postal	codigo_indicativo	telefono
2	300052,6,6490010	Codigo en Lomas de Zamora	Buenos Aires	Banfield	Museo De Artes Plasticas Pio Collivadino	Medrano 165	B1828PC	11,4			
3	300042,6,6035010	Buenos Aires	Avellaneda	Museo Municipal De Artes Plasticas De Avellaneda	Sarmiento 101	1_Piso	B1870CBC	11,4	20595		
4	300645,6,6658010	Buenos Aires	Quilmes	Museo De Artes Visuales Victor E. Roverano	Rivadavia 498	B1878KJ	11,4	2245336	34,7	192	58,25
5	300067,6,6658010	Codigo en Quilmes	Buenos Aires	Bernal	Museo Municipal Historico Regional Almirante Brown	25 de Mayo 198	B1876AW				
6	300646,6,6658010	Buenos Aires	Quilmes	Museo Municipal Historico Fotografico De Quilmes	Geronima Irma Giles Y Gaetano De Mayo	25 de Mayo	B1876AW				
7	300644,6,6658010	Buenos Aires	Quilmes	Museo Municipal Del Transporte Don Carlos Hillner Y Decoud	Laprida y Ricardo Rojas	B1879BTQ	11,4				
8	300568,6,6260010	Codigo en Esteban Echeverria	Buenos Aires	Monte Grande	Museo Historico Municipal La Campana	Dean Gregorio Funes					
9	300070,6,6028010	Codigo en Almirante Brown	Buenos Aires	Monte Grande	Museo Claudio Leon Sempere	Colon 581	B1852BS	11,4	299779	34,7	
10	300786,6,6756010	Buenos Aires	San Isidro	Museo Historico Y Brig. Gral. Juan M. Pueyrredon	Rivera Indarte 48	B1642IFB	11,4	51231			
11	300915,6,6805010	Buenos Aires	Tigre	Museo De La Reconquista	Liniers y Padre Castañeda	B1648DA	11,4	5124496	cultura@tigre.gov		
12	300918,6,6805010	Buenos Aires	Tigre	Museo De Arte Tigre Paseo Victoria 972	B1648DVT	11,4	5124528	museodearte@tigre.gov.ar	34,4	10	
13	300787,6,6756010	Buenos Aires	San Isidro	Museo Biblioteca Y Archivo Historico Municipal De San Isidro	Dr. Horacio Beccar Varela	Adrian B					
14	300574,6,6568010	Buenos Aires	Morón	Museo De La Catedral De Moron Inmaculada Concepcion Del Buen Viaje	Gral. Manuel Belgrano 32						
15	300574,6,6568010	Buenos Aires	Morón	Museo Historico Y De Arte Gral. Jose De San Martin	Casullo 59	B1708IPA	11,4	6291456	arteycultura@moron.gov.ar		
16	300262,6,6840010	Codigo en Tres de Febrero	Buenos Aires	Ciudadela	Museo Historico Del Ejercito Argentino	Carlos Pellegrini 4000	B1702BC				
17	300824,6,6760010	Buenos Aires	San Miguel	Museo Mariposas Del Mundo	Italia 650	B1663NXN	11,4	46642108	mariposas@newroad.com.ar		
18	300576,6,6568010	Buenos Aires	Morón	Museo Nacional De Aeronautica Brig. Edmundo H. Cívati Bernasconi	Avda. Eva Peron 2200	B1708FF					
19	300578,6,6861010	Codigo en Vicente Lopez	Buenos Aires	Munro	Museo Del Cine Lumiton	Sargento Cabral 2354	B1605EEH	11,4	7219255	34,7	

Carga de datos { Objetivo:

Clases importadas

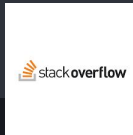
```
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.LinkedList;  
import java.util.Queue;
```

Lograr que cada FILA
del CSV pueda ser un
OBJETO (clase Museos)
dentro del vector
MUSEOS dentro de
nuestro programa JAVA

Otras instrucciones

While
Try - catch

Research



01

{

En el Main...

```
Museos[] vectormuseos=new Museos[1046];
vectormuseos=Read.vector();
```

}

```
public class Read {
    public static void main(String[] args) {
        vector();
    }

    public static Museos[] vector() {
        String file = "C:\\Users\\Familia feliz\\eclipse-workspace\\final\\src\\Museos.csv"; //
        BufferedReader reader = null;
        String line;
        Museos museos[] = new Museos[1046];
        Queue<Cliente> colaclientes = new LinkedList<>();

        try {
            reader = new BufferedReader(new FileReader(file));
            int cont = 0;

            while ((line = reader.readLine()) != null) {
                String[] columna = line.split(",");

                if (cont > 0 && cont < 1047) {

                    Museos museo1 = new Museos(colaclientes,columna[0], columna[1], columna[2],

                    museos[cont - 1] = museo1;

                }
                cont++;
            }
        }
    }
}
```

02 {

[Recorrido del programa]

< Clases implementadas: Main, Cliente, Ticket,
Listacliente, Listaticket, Nodocliente,
Nodoticket, Museos, Read >

}

```
1 public class Listacliente {
2     private Nodocliente inicio;
3     private int longitud;
```

```
4 public class Nodocliente {
5     Nodocliente siguiente;
6     Cliente cliente;
```

```
7 public class Listaticket {
8     private Nodoticket inicio;
9     private int longitud;
```

```
10 public class Nodoticket {
11     Nodoticket siguiente;
12     Ticket ticket;
```

```
public class Ticket {
    private Cliente cliente;
    private Museos museo;
    private int numticket;
```

```
public class Cliente {
    private int dni;
    private String nombre;
    private String apellido;
```

```
public class Museos {
    private String espacio_cultural_id;
    private String provincia_id;
    private String localidad_id;
    private String observaciones;
    private String provincia;
    private String localidad;
    private String nombre;
    private String mail;
    private String web;
    private String jurisdiccion;
    private String anio_de_creacion;
    private String descripcion_de_patrimonio;
    private String anio_de_inaguracion;
    public Queue<Cliente>colaclientes=new LinkedList<>();
```



```
1  < 1er switch > {
2
3  |
4      < El usuario puede optar por ser Administrador,
5      Cliente o Salir del programa >
6  }
7
8  < Switches internos > {
9
10 |
11     < Tanto el Administrador como el Cliente tienen
12     un conjunto de funcionalidades distintas que
13     pueden optar por utilizar >
14 }
```

Método más utilizado:

Nombre: validoytraigomuseo

Funcionamiento:

1. Pide la provincia y localidad del museo que se requiere y valida que existan ambas
2. Se muestran todas las opciones de museos en esa provincia y localidad, cada uno con un número de identificación
3. Se pide que se ingrese el número de identificación del museo que se desea
4. Se le pide al usuario que confirme su elección
5. El método devuelve el museo seleccionado

Funcionalidades administrador {

1. Atender clientes
2. Consultar tickets e ingresar clientes a un museo
3. Salir del menú administrador

}

1. Atender clientes

- 1- Se utiliza el método `validoytraigomuseo`
- 2- Se atienden a todos los clientes de ese museo, sacándolos de la cola y brindándoles un ticket
- 3- Se agrega el ticket creado a la lista universal de tickets

```
public static void atendemosclientes(Museos museo, Listaticket ticketslista) {  
    if (museo.colaclientes.isEmpty()) {  
        System.out.println("No hay nadie en esta cola para atender. ");  
    } else {  
        while(!museo.colaclientes.isEmpty()) {  
            System.out.println("Atendemos el siguiente cliente " + museo.colaclientes.peek());  
            Ticket ticket = new Ticket(museo.colaclientes.poll(), museo);  
            System.out.println("Se le ha asignado el ticket: "+ticket);  
            Nodoticket nodo = new Nodoticket(ticket);  
            ticketslista.agrego(nodo);  
        }  
    }  
}
```

2. Consultar tickets e ingresar clientes a un museo

Ingresamos los datos del cliente que se quiere ingresar y en caso de que exista le preguntamos al administrador en qué museo quiere ingresarlo.

3 posibilidades:

- a) Cliente no existente
- b) Cliente existente sin ticket
- c) Cliente existente con ticket

Funcionalidades cliente {

Ingresar tus datos al sistema

1. Sacar entradas a un museo
2. Consultar mis tickets
3. Consultar posición en alguna cola de un museo
4. Cancelar adquisición de tickets (salir de la cola)
5. Salir del menú cliente

}

1 Ingresar tus datos al sistema

- 2
- 3 - Ingresamos el nombre, apellido y DNI del cliente.
- 4
- 5 - Si la lista se encuentra vacía lo agregamos al sistema.
- 6
- 7 - Verificamos que no se encuentre en la **lista general de clientes**
- 8 recorriendo la listaclientes comparando con el cliente en cuestión
- 9 [buscandocliente].
- 10
- 11 - Si ya se encuentra, no lo agregamos, sino creamos un nodo y lo
- 12 agregamos a la lista clientes.
- 13
- 14

```
1      },
2      public boolean buscandocliente(int dni,String
3      nombre,String apellido){
4          boolean encontrado=false;
5          Nodocliente aux=this.inicio;
6          while (aux!=null){
7              if(dni==aux.getCliente().getDni() &&
8              nombre.equals(aux.getCliente().getNombre()) &&
9              apellido.equals(aux.getCliente().getApellido())){
10                  encontrado=true;
11                  break;
12              }
13              else{
14                  aux=aux.siguiente;
15              }
16          }
17          return encontrado;
18      }
19  }
```



Main.java

```
public boolean esVacia() {  
    if(this.inicio==null) {  
        return true;  
    }  
    else  
        return false;  
}
```

```
public void agrego(Nodocliente nodo) {  
    if(this.esVacia()) {  
        setInicio(nodo);  
        longitud++;  
    }  
    else{  
        if(this.buscar(nodo.getCliente())==false) {  
            nodo.setSiguiete(this.inicio);  
            this.inicio=nodo;  
            longitud++;  
        }  
    }  
}
```

Museos.java

```
public boolean buscar(Cliente cliente) {  
    Nodocliente cliente2=this.inicio;  
    boolean verdad=false;  
    while(cliente2!=null) {  
        if(cliente2.equals(cliente)) {  
            verdad=true;  
        }  
        cliente2=cliente2.getSiguiete();  
    }  
    return verdad;  
}
```



1. Sacar entradas a un museo

2

3 - Ingresamos museo a ingresar a la cola para comprar una entrada.

4

5 - Validamos que exista y sea el deseado, confirmamos.

6

7 - Si el cliente no se encuentra en la cola (`contains`) lo agregamos a
la respectiva cola (`museo.colaclientes.add()`) (Se encuentran colas
8 por museo) y visualizamos la cola recorriendola

9

10 - Si el cliente ya se encuentra la comentamos que ya se encuentra en
la cola del museo ingresado

11

12

13

14

```
1  
2  
3  
4  
5     public void visualizamoscola() {  
6         System.out.println("Los clientes en la cola hasta  
7         ahora son: ");  
8         for (Cliente i : this.colaclientes) {  
9             System.out.println(i);  
10        }  
11  
12  
13  
14
```



2. Consultar mis tickets

Recorremos la `listageneraldetickets`, si es que el cliente se encuentra en la lista imprimimos su ticket (`getTicket()`), si es que no tiene le comentamos que no tiene tickets el cliente.

```
public void muestroticketscliente(Cliente cliente) {  
    Nodoticket aux = this.inicio;  
    boolean bande=false;  
    while (aux != null) {  
        if (cliente.equals(aux.getTicket().getCliente()))  
        {  
            System.out.println(aux.getTicket());  
            bande=true;  
            aux=aux.getSiguiente();  
        } else {  
            aux = aux.siguiente;  
        }  
    }  
    if (!bande){  
        System.out.println("Usted no tiene tickets. ");  
    }  
}
```

3. Consultar posición en alguna cola de un museo

- Preguntamos de qué museo quiere saber la posición en la que se encuentra en la cola.
- Validamos el museo
- Si no contiene el cliente en la cola de clientes imprimimos que no se encuentra el cliente en esta cola
- Si es que si se encuentra busco la posicion y visualizo la cola

```
public static void buscoposicionescolas (Museos
museos, Cliente cliente) {

    if (museos.colaclientes.isEmpty()) {
        System.out.println("No se encuentra usted en esta
cola del museo: "+ museos);
    } else {
        int count=1;
        for (Cliente i : museos.colaclientes) {
            if(i.equals(cliente)) {
                System.out.println("Usted se encuentra en
la posicion: "+count);}
                count++;
            }
        }
    }
}
```

11

12

13

14

```
public void visualizamoscola() {
    System.out.println("Los clientes en la cola hasta
ahora son: ");
    for (Cliente i : this.colaclientes) {
        System.out.println(i);
    }
}
```



1 4. Cancelar adquisición de tickets (salir de la cola)

2

3

- El cliente selecciona de que museo desea irse de la cola

4

5

- Verificamos el museo

6

7

- Si el cliente se encuentra en la cola de este museo (`contains`)
actualizamos la cola y luego la visualizamos

8

9

- Si no se encuentra imprimimos que no puede salir de la misma ya
que no se encuentra en esta

10

11

12

13

14

```

    public static void actualizocola(Museos museo, Cliente
cliente) {
        Queue<Cliente> ayuda= new LinkedList<>();
        while(!museo.colaclientes.isEmpty()){
            if(!
museo.colaclientes.peek().equals(cliente)){

                ayuda.add(museo.colaclientes.poll());

            }
            else{
                System.out.println("Se esta removiendo a:
"+cliente);
                museo.colaclientes.remove(cliente);
            }
        }
        for(Cliente i:ayuda){
            museo.colaclientes.add(i);
        }
        System.out.println("Ahora la cola de museo
actualizada es ");
        System.out.println(museo.colaclientes);
    }

```

13

14

```

    public void visualizamoscola() {
        System.out.println("Los clientes en la cola hasta
ahora son: ");
        for (Cliente i : this.colaclientes) {
            System.out.println(i);
        }
    }

```



03 {

[Detalles a destacar]

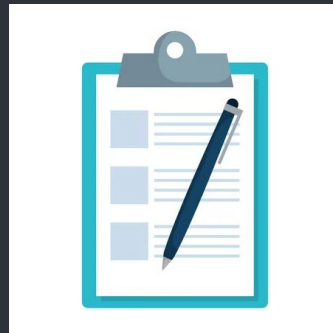
<Lista universal de tickets,
Cola dentro de museos, Lista
universal de clientes >

}

¿El por que de nuestras listas universales? {

Los clientes ingresarán al sistema con nombre, apellido y dni. Si el cliente ya ingresó previamente, se debió haber guardado todos los datos pertinentes a este y continuar a partir de ellos. Por ejemplo, si este ya había sacado tickets, esos tickets deben estar asociados a ese cliente cuando vuelva a ingresar.

}



```
1
2
3  Cliente cliente =new Cliente(dni, nombre, apellido);
4  if(!listaclientes.buscandocliente(dni,nombre,apellido)){
5      System.out.println("El cliente no esta ingresado en la
6  }
7
8  else{
9      Museos museochequeo= validoytraigomuseo(vectormuseos);
10     if(!listatickets.chequeoticket(cliente,museochequeo)){
11         System.out.println("El cliente "+cliente.getNombre(
12     }
13
14
```

Colas como atributos y vector de museos

Los clientes deberán poder buscar museos, refinando su búsqueda por provincia y localidad. Luego de buscarlos, pueden seleccionar un museo específico para sacar una entrada. Cuando elijan sacar una entrada estos serán admitidos en una cola específica de ese museo.



```
1
2 public class Museos {
3     private String espacio_cultural_id;
4     private String provincia_id;
5     private String localidad_id;
6     private String observaciones;
7     private String provincia;
8     private String localidad;
9     private String nombre;
10    private String mail;
11    private String web;
12    private String juridiccion;
13    private String anio_de_creacion;
14    private String descripcion_de_patrimonio;
15    private String anio_de_inaguracion;
16    public Queue<Cliente>colaclientes=new LinkedList<>();
```

1 ¿Cómo nos ayudó la cola dentro de museos?
2
3
4

case 1.

```
Museos museo = validoytraigomuseo(vectormuseos);  
if(!Museos.buscoclienteencola(museo,cliente)){  
    museo.colaclientes.add(cliente);  
    museo.visualizamoscola();  
}  
else {  
    System.out.println("Usted ya se encuentra en la cola de este museo: "+museo);  
}  
break;
```

```
System.out.println("Seleccione el museo del cual quiere salir de la cola: ");

Museos museo3 = validoytraigomuseo(vectormuseos);
if(museo3.getColaclientes().contains(cliente)){
    Museos.actualizocola(museo3,cliente);
    museo3.visualizamoscola();
}
else {
    System.out.println("Usted no puede salir de la cola ya que no se encuentra en la misma.");

}

break;
```

1
2 04 {
3
4
5
6
7
8
9
10
11
12 }
13
14

[Experiencia en el proceso]

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Muchas
gracias