

Prueba Sumativa 1

Programación Orientada a Objetos – 29 de abril 2025

| | | | |
|----------|------------------------|-----|--|
| Nombre | | RUT | |
| Paralelo | () APaolini () CRabi | | |

Antecedentes generales:

| | | | |
|---|---|------------------|--|
| Puntaje total de la prueba/Puntos para nota aprobatoria (4.0) | 1 puntos 0.6 puntos | Puntaje Obtenido | |
| Duración de la prueba | 3 horas | Nota final | |
| Resultados de Aprendizaje a evaluar | 1. Aplicar técnicas de ingeniería de software en la creación de software legible, mantenible y testeable. 2. Aplicar técnicas de programación orientada al objeto en la resolución de problemas. 3. Crear tipos de datos abstractos con bajo acoplamiento entre la implementación y su comportamiento que permitan la resolución problemas. 4. Analizar las relaciones causa efecto de los procesos en estudio. 6. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos. | | |
| Fecha de entrega de resultados | 14 mayo 2025 | | |

Instrucciones:

- Esta evaluación tiene 4 páginas (incluyendo la portada). Compruebe que dispone de todas las páginas.
- Lea la prueba completamente DOS veces antes de hacer cualquier pregunta**
- Durante la prueba no se puede utilizar: teléfonos móviles, apuntes. Está prohibido intentar conectarse a internet de cualquier manera (excepto a Campus Virtual, y solo para subir su solución). Si es sorprendido obtendrá la calificación mínima. Tampoco puede utilizar dispositivos de almacenamiento externos o cualquier otro dispositivo como relojes inteligentes, tables, audífonos, etc.
- Una prueba respondida correctamente en un 60%, de acuerdo con las ponderaciones asignadas, corresponde a una nota 4,0.
- La prueba es individual, cualquier sospecha de copia será calificada con la nota mínima y el caso será remitido al comité de ética.
- En su espacio personal no debe haber nada más que hojas de papel en blanco, lápiz y goma.
- El resto de sus implementos debe guardarlos dentro de su mochila/bolso y ésta debe posicionarse al frente debajo de la pizarra. Si leyó hasta este punto, felicidades, para saber que lo hizo dibuje una palta al final de esta página.
- Los estudiantes quienes se les compruebe falta de honestidad académica o cualquier otro acto contrario a las normas de permanencia universitaria o al espíritu universitario, serán sancionados, según sea la gravedad de la falta, con medidas desde la amonestación verbal hasta la suspensión o pérdida de la condición de estudiante. Los estudiantes expulsados no podrán volver a ingresar a ninguna carrera, programa o curso de la institución. El estudiante que incurriere en falta de honestidad, durante la realización de un proceso evaluativo, será calificado con la nota mínima 1,0.

Acepto las condiciones firmando: _____

Problema 1. Ruteo (0.3 puntos)

Rutee el siguiente código e indique la salida esperada. Dibuje la **situación final** de los objetos involucrados al terminar la ejecución. No es necesario indicar todo el paso a paso.

```
1 public class Main {
2
3     public static void main(String[] args) {
4         Cliente cliente1 = new Cliente(5202, "Mauro Lombardo", "555-1996");
5         Cliente cliente2 = new Cliente(5203, "Luis Torres", "555-5678");
6         Cliente cliente3 = new Cliente(5204, "Carla Pino", "555-9911");
7         Cliente[] clientesDisponibles = { cliente1, cliente2, cliente3 };
8         ArrayList<Pedido> listaPedidos = new ArrayList<>();
9         int numeroInicial = 500;
10        listaPedidos.add(new Pedido(++numeroInicial, "Ositos Haribo", 3590, clientesDisponibles[0]));
11        listaPedidos.add(new Pedido(++numeroInicial, "Rollito de Canela", 3590, clientesDisponibles[1]));
12        listaPedidos.add(new Pedido(++numeroInicial, "Pizza", 6590, clientesDisponibles[1]));
13        listaPedidos.add(
14            new Pedido(++numeroInicial, "Burger Bacon", 3590, new Cliente(5206, "Camilo Villaruel", "555-9969")));
15        listaPedidos.get(2).getCliente().setTelefono("555-2896");
16        listaPedidos.remove(1);
17        listaPedidos.add(new Pedido(++numeroInicial, "Coca-Cola", 1590, clientesDisponibles[0]));
18        listaPedidos.add(new Pedido(++numeroInicial, "Handroll", 3990, null));
19        System.out.println(listaPedidos.get(2).getCliente());
20        listaPedidos.remove(4);
21        System.out.println("\n--- Lista de Pedidos Registrados ---");
22        if (listaPedidos.isEmpty()) {
23            System.out.println("No hay pedidos registrados.");
24        } else {
25            for (Pedido p : listaPedidos) {
26                System.out.println(p);
27            }
28        }
29    }
30 }
31
32
33 public class Cliente {
34     private int rut;
35     private String nombre;
36     private String telefono;
37
38     public Cliente(int rut, String nombre, String telefono) {
39         this.rut = rut;
40         this.nombre = nombre;
41         this.telefono = telefono;
42         System.out.println(Pedido.isBeep());
43     }
44     public String getTelefono() {
45         return telefono;
46     }
47     public void setTelefono(String telefono) {
48         this.telefono = telefono;
49     }
50     public int getRut() {
51         return rut;
52     }
53     public String getNombre() {
54         return nombre;
55     }
56     @Override
57     public String toString() {
58         return "Cliente [RUT=" + rut + ", Nombre=" + nombre + ", Teléfono=" + telefono + "];";
59     }
60 }
```

Ingeniería en Tecnologías de Información
Ingeniería Civil en Computación e Informática
Ingeniería Civil Industrial

```
61
62 public class Pedido {
63     private int numeroPedido;
64     private String descripcion;
65     private double total;
66     private Cliente cliente;
67     private static boolean beep;
68
69     public Pedido(int numeroPedido, String descripcion, double total, Cliente cliente) {
70         this.numeroPedido = numeroPedido;
71         this.descripcion = descripcion;
72         this.total = total;
73         this.cliente = cliente;
74         if (beep) System.out.println("VVS");
75         beep = !beep;
76     }
77     public int getNumeroPedido() {
78         return numeroPedido;
79     }
80     public String getDescripcion() {
81         return descripcion;
82     }
83     public double getTotal() {
84         return total;
85     }
86     public Cliente getCliente() {
87         return cliente;
88     }
89     public static boolean isBeep() {
90         return beep;
91     }
92     @Override
93     public String toString() {
94         String nombreCliente;
95         if (cliente != null) {
96             nombreCliente = cliente.getNombre();
97         }else {
98             nombreCliente = "N/A";
99         }
100
101         return "Pedido [Número= " + numeroPedido +
102             ", Descripción= " + descripcion +
103             ", Total= $" + (int) total +
104             ", Cliente= " + nombreCliente + "];";
105     }
106
107 }
108
```

Ingeniería en Tecnologías de Información
Ingeniería Civil en Computación e Informática
Ingeniería Civil Industrial
Problema 2. Music Analytics (0.7 puntos)

Usted tiene gustos musicales bastante amplios, escucha una gama variada de estilos y bandas musicales. Se dio cuenta que las bandas van cambiando con el tiempo y que sus estilos también, además, que las bandas no se cierran a un estilo específico, sino que prueban varios estilos en un sólo álbum.

Le llama la atención el potencial análisis que se puede lograr respecto a estilos, por ejemplo, la banda The Beatles comenzó con Rock and Roll en sus primeros álbumes, para luego pasar al Folk Rock, continuó con Psicodelia y así en adelante...

Por otra parte, tiene una serie de playlist para diferentes situaciones, lo cual también es interesante, porque hay álbumes que se presentan en circunstancias específicas de su vida. Por ejemplo, en su playlist llamada “estudio”, el 50% de las canciones son del álbum Fome de Los Tres y el otro 50% del álbum Rust in Peace de Megadeth.

Para poner a prueba sus conocimientos de Programación Orientada a Objetos, tomó la decisión de crear un sistema para registrar toda esta información y organizarla, con el objetivo de generar análisis sobre sus selectos y variados gustos.

Ya adelantó un poco de trabajo y creó un archivo llamado musica.txt donde se encuentran los datos de los álbumes y de sus canciones respectivas. Con este formato:

ID Album, Nombre Album, ID Banda, Año lanzamiento, Cantidad de canciones

ID Canción, Nombre Canción, Duración, Estilo, Rating

| musica.txt | Explicación |
|---|--|
| 1,Album1,1,2000,3 1,Cancion1,3:30,Rock,4 2,Cancion2,4:30,Glam,1 3,Cancion3,5:31,Balada,2 2,Album2, 1,2010,2 4,Cancion4,3:30,Reggae,4 5,Cancion5,4:30,Blues,1 ... | La 1a línea es el álbum con estos datos: ID Álbum: 1, Nombre: Álbum1, ID Banda: 1, Lanzamiento: 2000, Cantidad Canciones: 3 Luego vienen 3 líneas con canciones: La 2da línea es la primera canción del Álbum1, tiene estos datos: ID Canción: 1, Nombre: Canción1, Duración 3:30, Estilo: Rock, Rating: 4 El archivo continúa con más álbumes y canciones. |

Tiene un archivo con bandas:

| bandas.txt | Explicación |
|--|---|
| 1,Banda1,1970 2,Banda2,2010 3,Banda3,2001 ... | Cada línea tiene los datos de una banda: ID Banda, Nombre y Año de fundación. La 1ra línea tiene a la banda con ID: 1, Nombre: Banda1, Año fundación: 1970. |

Tiene un archivo con playlists:

| playlists.txt | Explicación |
|---|---|
| Viaje,3 1,2,3 Estudio,2 5,1 ... | La primera línea tiene a la playlist llamada “Viaje” y tiene 3 canciones, en la 2da línea están los ID de las 3 canciones (1,2,3). Luego viene otra playlist llamada “Estudio”, que tiene 2 canciones, sus ID están en la línea siguiente (5,1). |

Ingeniería en Tecnologías de Información
Ingeniería Civil en Computación e Informática
Ingeniería Civil Industrial
Funcionalidades

- Cargar datos de archivos: Al iniciar el programa debe leer los archivos
- Menú con reportes

Reportes

1. Mostrar la composición estilos de todas las bandas en base a sus canciones. Ejemplo:

-Banda1:
Cumbia (70%)
Bachata (10%)
Rock (20%)
-Banda2:
Heavy Metal (20%)
Glam Rock (80%)
...

2. Para cada playlist, mostrar la proporción de álbumes que la componen en base a sus canciones. Ejemplo:

-Viaje
Album1 (40%)
Album3 (20%)
Album8 (40%)
-Estudio
Album2(90%)
Album1(10%)...

3. Mostrar los álbumes de todas las décadas (desde 1950 a 2030). Ejemplo:

1950-1959:
ÁlbumX (AñoLanzamiento) – BandaA
ÁlbumY (AñoLanzamiento) – BandaB
1960-1969:
ÁlbumZ (AñoLanzamiento) – BandaC
...
2020-2030: ...

4. Mostrar el rating de todas las bandas, este se calcula mediante el promedio de todos los ratings de sus canciones.
Ejemplo:

-Banda1: 4.9
-Banda2: 2.4

Debe entregar:

- Diagrama de clases (25%)
- Código Java (75%)

Consideraciones:

- Una canción tiene solo un estilo
- Debe usar orientación a objetos.
- El código fuente debe exportarlo en un solo archivo .zip y subirlo a Campus Virtual.
- Los archivos .txt de ejemplo se encuentran en CampusVirtual.
- Los diagramas de clases debe escribirlos en papel y entregarlos junto a la prueba.
- Hojas sin nombre no se revisarán.
- Debe usar referencias
- Si lee esto quiere decir que leyó las consideraciones, dibuje un tenedor al final de la página.

Ejemplos de ejecución: Revise el archivo ejemplos.txt