

Tarea Final de la Unidad PROG05

Realizar un ejercicio en JAVA sobre tipos avanzados de datos.

Entrega: Deberás crear una carpeta comprimida en formato ‘zip’ con el nombre “Apellido_Nombre_Tarea_Final_PROG05”. Dentro de esta carpeta, incluye las clases del ejercicio. Sube el archivo comprimido en la tarea correspondiente del aula virtual.

SISTEMA DE GESTIÓN DE DISPOSITIVOS ELECTRÓNICOS

Descripción Desarrollar una aplicación en Java que permita gestionar dispositivos electrónicos y registrar información de uso, aplicando estructuras de datos estáticas y dinámicas.

Con esta tarea se trabajan los siguientes contenidos:

- Arrays (estructura de tamaño fijo).
- ArrayList (estructura dinámica).
- HashMap (estructura clave/valor para búsquedas rápidas).
- Recorridos de los tipos de datos anteriores.
- Encapsulación, métodos, validación de entradas.

El sistema gestionará un conjunto de dispositivos (portátil, tablet, smartphone, etc.).

Cada dispositivo tendrá:

- Un código identificador único.
- Un nombre descriptivo.
- Un tipo (por ejemplo: “portátil”, “tablet”...).
- Un historial de sesiones de uso, donde cada sesión registra:
 - horas de uso
 - consumo en porcentaje
 - temperatura máxima alcanzada

La aplicación permitirá:

- Dar de alta dispositivos.
- Registrar sesiones de uso.
- Consultar estadísticas por dispositivo.
- Consultar estadísticas globales.

CLASE SesionUso

Atributos:

- private double horas
- private double consumo
- private double temperaturaMax

Constructor:

- SesionUso(double horas, double consumo, double temperaturaMax)

Métodos:

- getters necesarios
- toString() con un formato claro

CLASE Dispositivo

Atributos:

- private String codigo
- private String nombre
- private String tipo

Estructuras de datos obligatorias dentro de Dispositivo:

- private ArrayList<SesionUso> sesiones
(almacena dinámicamente todas las sesiones del dispositivo)

Constructor:

- `Dispositivo(String codigo, String nombre, String tipo)`
Inicializa atributos y crea el `ArrayList` vacío.

Métodos mínimos:

- `public void registrarSesion(SesionUso sesion)`
Añade la sesión al listado.
- `public double mediaConsumo()`
Media del consumo de todas las sesiones (si no hay sesiones, devuelve 0).
- `public double mediaTemperatura()`
Media de temperatura máxima de todas las sesiones (si no hay sesiones, devuelve 0).
- `public double totalHorasUso()`
Suma total de horas.
- `public SesionUso sesionMayorConsumo()`
Devuelve la sesión con mayor consumo (si no hay, devuelve null).
- `public String toString()`
Debe mostrar: código, nombre, tipo, total horas, media consumo y media temperatura.
- `public void mostrarSesiones()`
Muestra todas las sesiones registradas (una por línea).

CLASE AlmacenDispositivos

Esta clase será el “gestor” principal y debe obligatoriamente combinar:

- Un array para almacenamiento fijo.
- Un `HashMap` para búsquedas rápidas.

Atributos obligatorios:

- `private Dispositivo[] inventario`
Tamaño fijo: 30 dispositivos máximo.
- `private int contador`
Número de dispositivos almacenados en el array.
- `private HashMap<String, Dispositivo> indicePorCodigo`
Permite buscar un dispositivo por código.

Constructor:

- Inicializa el array (tamaño 30), contador a 0 e inicializa el HashMap vacío.

Métodos mínimos:

- `public boolean altaDispositivo(Dispositivo d)`

Reglas:

- No se permiten códigos duplicados (consultar HashMap).
- Si el array está lleno, devolver false.
- Si se inserta correctamente:
 - guardar en el array en la siguiente posición libre
 - incrementar contador
 - añadir al HashMap (clave = código)
 - devolver true
- `public Dispositivo buscarPorCodigo(String codigo)`
Devuelve el dispositivo o null si no existe.
- `public void listarDispositivos()`
Recorre el array y muestra los dispositivos existentes con `toString()`.
- `public double mediaGlobalConsumo()`
Calcula la media de consumo de todos los dispositivos (`mediaConsumo()`).

- `public Dispositivo dispositivoMasUsado()`
Devuelve el dispositivo con mayor `totalHorasUso()` (si no hay dispositivos, `null`).

CLASE Principal

La clase Principal debe contener el método `main` y mostrar un menú repetitivo hasta salir.

Menú obligatorio:

1. Dar de alta un dispositivo
 - Solicitar: código, nombre, tipo.
 - Crear objeto `Dispositivo`.
 - Llamar a `altaDispositivo(...)`
 - Informar si se ha creado correctamente o si el código existe / inventario lleno.
2. Registrar sesión de uso en un dispositivo
 - Pedir código del dispositivo.
 - Si no existe, mostrar mensaje y volver al menú.
 - Pedir:
 - horas (`double > 0`)
 - consumo (`double >= 0`)
 - temperatura máxima (`double >= 0`)
 - Crear `SesionUso` y registrarla en el dispositivo.
3. Mostrar información de un dispositivo
 - Pedir código.
 - Mostrar el `toString()` del dispositivo.

- Mostrar también el listado de sesiones (mostrarSesiones()).
- 4. Listar todos los dispositivos
 - Mostrar todos los dispositivos registrados en el inventario (array).
- 5. Mostrar estadísticas globales
 - Mostrar:
 - mediaGlobalConsumo()
 - dispositivoMasUsado() (si existe)
- 6. Salir

VALIDACIONES OBLIGATORIAS

- El código no puede ser vacío.
- Horas debe ser > 0 .
- Consumo y temperatura deben ser ≥ 0 .

No se debe permitir que el programa se caiga por entradas no numéricas: se debe controlar con try/catch.

Ejemplos de ejecución:

```
=== SISTEMA DE GESTIÓN DE DISPOSITIVOS ===
1. Dar de alta un dispositivo
2. Registrar sesión de uso en un dispositivo
3. Mostrar información de un dispositivo
4. Listar todos los dispositivos
5. Mostrar estadísticas globales
6. Salir
Elige una opción (1-6): 1

--- Alta de dispositivo ---
Código (no vacío): SM01
Nombre: S24
Tipo: Móvil
Dispositivo dado de alta correctamente.

=== SISTEMA DE GESTIÓN DE DISPOSITIVOS ===
1. Dar de alta un dispositivo
2. Registrar sesión de uso en un dispositivo
3. Mostrar información de un dispositivo
4. Listar todos los dispositivos
5. Mostrar estadísticas globales
6. Salir
Elige una opción (1-6): 2

--- Registrar sesión ---

Código del dispositivo: SM01
Horas de uso (> 0): 45
Consumo (>= 0): 23
Temperatura máxima (>= 0): 45
Sesión registrada correctamente.

=== SISTEMA DE GESTIÓN DE DISPOSITIVOS ===
1. Dar de alta un dispositivo
2. Registrar sesión de uso en un dispositivo
3. Mostrar información de un dispositivo
4. Listar todos los dispositivos
5. Mostrar estadísticas globales
6. Salir
Elige una opción (1-6): 3

--- Información de dispositivo ---
Código del dispositivo: SM01
Dispositivo [codigo=SM01, nombre=S24, tipo=Móvil, total hora de uso=45.0, media consumo= 23.0, media temperatura= 45.0]
Sesiones:
- Sesión [horas=45,00, consumo=23,00%, tempMax=45,00°C]
Sesión con mayor consumo: Sesión [horas=45,00, consumo=23,00%, tempMax=45,00°C]

=== SISTEMA DE GESTION DE DISPOSITIVOS ===
1. Dar de alta un dispositivo
2. Registrar sesión de uso en un dispositivo
3. Mostrar información de un dispositivo
4. Listar todos los dispositivos
5. Mostrar estadísticas globales
6. Salir
Elige una opción (1-6): 4

--- Listado de dispositivos ---
Dispositivo [codigo=SM01, nombre=S24, tipo=Móvil, total hora de uso=45.0, media consumo= 23.0, media temperatura= 45.0]

=== SISTEMA DE GESTIÓN DE DISPOSITIVOS ===
1. Dar de alta un dispositivo
2. Registrar sesión de uso en un dispositivo
3. Mostrar información de un dispositivo
4. Listar todos los dispositivos
5. Mostrar estadísticas globales
6. Salir
Elige una opción (1-6): 5

--- Estadísticas globales ---
Media global de consumo: 23,00%
Dispositivo más usado: Dispositivo [codigo=SM01, nombre=S24, tipo=Móvil, total hora de uso=45.0, media consumo= 23.0, med
```

Rúbrica de Evaluación

Criterios		Puntos
Clase SesionUso	Atributos correctos	0,5
	Constructor con parámetros, getters y toString().	0,5
Clase Dispositivo	Atributos, ArrayList<SesionUso>	0,5
	registrarSesion() y mostrarSesiones().	1,0
	mediaConsumo(), mediaTemperatura(), totalHorasUso()	1,0
	sesionMayorConsumo()	0,5
	toString()	0,5
Clase AlmacenDispositivos	Estructuras obligatorias correctas: Dispositivo[] inventario (tamaño 30), contador, HashMap<String,Dispositivo> indicePorCodigo.	1,0
	altaDispositivo() cumple reglas: no duplicados, controla inventario lleno, inserta en array + hashmap e incrementa contador.	1,0
	buscarPorCodigo() y listarDispositivos()	1,0
	Estadísticas globales: mediaGlobalConsumo() y dispositivoMasUsado()	1,0
Clase Principal	Menú completo y llamadas correctas a los métodos del sistema.	1,0
	Validaciones y control de errores.	0,5
TOTAL		10,0