

# Caso de estudio - evaluación formativa 1

## Sistema de Gestión de Biblioteca Digital "GameVerse"

### CONTEXTO DEL PROBLEMA

La plataforma de videojuegos digital "GameVerse" necesita un sistema de consola para gestionar su biblioteca de títulos disponibles. El sistema debe manejar diferentes tipos de videojuegos, calcular métricas de éxito con algoritmos específicos, procesar análisis de manera asíncrona, y generar reportes estadísticos para la toma de decisiones comerciales.

### OBJETIVO PRINCIPAL

Desarrollar una aplicación de consola en Kotlin que permita:

- Gestionar una biblioteca de videojuegos de la plataforma
- Procesar análisis con cálculos de éxito automáticos
- Aplicar filtros y búsquedas según criterios específicos
- Simular el procesamiento asíncrono de datos desde servidores
- Generar reportes de rendimiento utilizando operaciones funcionales

### REQUERIMIENTOS ESPECÍFICOS

#### MODELADO DE CLASES

El sistema debe representar diferentes tipos de videojuegos de la plataforma. Cada juego tiene características básicas como título, precio, año de lanzamiento y calificación de usuarios. Es necesario que algunos juegos puedan tener variaciones en su métrica de éxito según sus características específicas.

El sistema debe manejar al menos tres tipos diferentes de videojuegos:

- a) **JuegoRPG**: Los juegos de rol pueden tener múltiples expansiones y horas de contenido que afectan su valor total de manera exponencial.
- b) **JuegoAccion**: Los juegos de acción pueden tener diferentes presupuestos de desarrollo y ventas que determinan su éxito comercial relativo.
- c) **JuegoIndie**: Los juegos independientes pueden ser innovadores, lo que influye en su valoración y reconocimiento en la plataforma.

#### MANEJO DE ESTADOS ASÍNCRONOS

El sistema debe representar los diferentes estados que puede tener una operación de análisis durante su procesamiento: **Iniciando - Analizando - Completado - Fallido**

Debe simular el procesamiento de análisis de juegos de manera asíncrona, incluyendo un tiempo de espera que represente la carga real de datos desde servidores externos (Se recomienda 2.5 segundos de espera).

## **LÓGICA DE NEGOCIO Y CÁLCULOS**

El sistema debe manejar información sobre diferentes métricas de juegos, valores totales de la biblioteca, y cálculos de éxito específicos.

Es importante seleccionar los tipos de datos más apropiados para cada variable.

El sistema debe realizar cálculos precisos para: • Éxito individual por videojuego • Métricas de éxito por tipo de juego • Valor total de la biblioteca • Estadísticas comparativas entre categorías

### **Reglas de Negocio**

El sistema debe implementar lógica condicional para: • Determinar éxito según el tipo de videojuego • Validar rangos válidos para calificaciones (1.0 - 10.0) • Aplicar filtros especiales cuando corresponda

## **FUNCIONES Y COLECCIONES**

El código debe estar organizado en funciones que encapsulen responsabilidades específicas como la inicialización de la biblioteca, cálculo de éxito, aplicación de filtros y presentación de resultados.

El sistema debe utilizar colecciones apropiadas para almacenar tanto la biblioteca de juegos disponibles como los resultados de análisis actuales.

Debe realizar operaciones avanzadas sobre las colecciones de datos para filtrar juegos por criterios específicos, transformar datos aplicando cálculos, y generar totales automáticamente.

También debe incluir una forma personalizada de mostrar los resultados iterando sobre los datos.

## **MANEJO DE ERRORES**

El sistema debe ser robusto y manejar situaciones de error como datos inválidos (calificaciones fuera de rango), operaciones matemáticas problemáticas, o accesos incorrectos a los datos almacenados. La aplicación debe continuar funcionando correctamente incluso cuando ocurran estos errores.

## **ESTRUCTURA TÉCNICA REQUERIDA**

## Archivos a Crear:

1. **Main.kt** - Punto de entrada del programa
2. **Videojuego.kt** - Clase base y derivadas
3. **EstadoAnalisis.kt** - Sealed class para estados
4. **GestorBiblioteca.kt** - Funciones de negocio

## Flujo del Programa:

1. Mostrar biblioteca de videojuegos
2. Seleccionar opciones de análisis
3. Calcular métricas con algoritmos específicos
4. Procesar análisis de forma asíncrona
5. Mostrar resumen final con estadísticas detalladas

## DATOS DE PRUEBA SUGERIDOS

### Biblioteca Inicial:

- **JuegoRPG**: "The Witcher 3", precio \$39.99, año 2015, expansiones: 2, horas de contenido: 120, calificación 9.3
- **JuegoRPG**: "Elden Ring", precio \$59.99, año 2022, expansiones: 1, horas de contenido: 100, calificación 9.5
- **JuegoAccion**: "Cyberpunk 2077", precio \$49.99, año 2020, presupuesto \$174M, ventas \$780M, calificación 7.8
- **JuegoAccion**: "Call of Duty: MW", precio \$69.99, año 2023, presupuesto \$250M, ventas \$1200M, calificación 8.2
- **JuegoIndie**: "Hades", precio \$24.99, año 2020, tema "Mitología", innovador=true, calificación 9.0
- **JuegoIndie**: "Hollow Knight", precio \$14.99, año 2017, tema "Metroidvania", innovador=true, calificación 8.7

### Fórmulas de Éxito:

- **JuegoRPG**:  $(\text{calificación} \times \text{expansiones} \times \text{horas de contenido}) / 15$
- **JuegoAccion**:  $(\text{calificación} \times \text{ventas}) / \text{presupuesto}$
- **JuegoIndie**:  $\text{calificación} \times (\text{innovador} ? 1.8 : 1.0)$

## Funcionalidades Mínimas Obligatorias:

• Sistema de herencia con clase base y clases derivadas • Polimorfismo mediante sobrescritura de métodos • Manejo de estados para operaciones asíncronas • Implementación de operaciones asíncronas con simulación de tiempo • Uso de colecciones para gestionar datos • Operaciones funcionales avanzadas sobre colecciones • Lógica condicional para reglas de negocio • Manejo robusto de errores y excepciones • Código organizado en funciones modulares • Documentación y comentarios en secciones clave

**Entrega:** Proyecto comprimido subido a AVA antes del término de la sesión. Código fuente completo y funcional.

## EJEMPLO DE EJECUCIÓN ESPERADA

```
=== SISTEMA PIXELHUB ===
Biblioteca disponible:
1. The Witcher 3 (RPG) - Rating: 9.3 - Éxito: 148.8
2. Elden Ring (RPG) - Rating: 9.5 - Éxito: 63.3
3. Cyberpunk 2077 (Acción) - Rating: 7.8 - Éxito: 35.0
4. Call of Duty: MW (Acción) - Rating: 8.2 - Éxito: 39.4
5. Hades (Indie) - Rating: 9.0 - Éxito: 16.2
6. Hollow Knight (Indie) - Rating: 8.7 - Éxito: 15.7

--- MENÚ PRINCIPAL ---
1. Ver biblioteca completa
2. Análisis de éxito (Async)
3. Filtrar por año
4. Estadísticas de la biblioteca
5. Salir
Opción: 2

Iniciando análisis de éxito...
Estado: Iniciando análisis...
Estado: Analizando algoritmos de éxito...
Estado: Procesando datos de servidor...
ANÁLISIS COMPLETADO

Análisis completado: The Witcher 3 es el más exitoso
JUEGO MÁS EXITOSO: The Witcher 3 (2015)
Puntuación de Éxito: 148.80

--- MENÚ PRINCIPAL ---
1. Ver biblioteca completa
2. Análisis de éxito (Async)
3. Filtrar por año
4. Estadísticas de la biblioteca
5. Salir
Opción: 5

;Gracias por usar GameVerse!
```