Informe de Diseño del Sistema de Gestión de Tienda de Discos

INFORME DE DISEÑO DEL SISTEMA DE GESTIÓN DE TIENDA DE DISCOS

Base Conceptual: Modelo orientado a objetos siguiendo principios SOLID, DRY, y Domain-Driven Design (DDD).

1. NOMENCLATURA DE CLASES Y VARIABLES
- Store: Representa la entidad física de la tienda. Ej: name, address
- Employee: Modela a los trabajadores. Ej: employeeNumber
- MusicalProduct: Clase abstracta para productos como CD/DVD.
- RecordingService: Servicio de grabación con costos variables.
- MultimediaContent: Interfaz común para canciones y videos.

Justificación:
- Claridad semántica y coherencia con el lenguaje del dominio.

2. TIPOS DE DATOS Y VALIDACIONES
- employeeNumber: String con formato especial.
- mediaType: Enum (CD, DVD) para evitar errores.
- baseCost: double para cálculos monetarios.
- collaborators: Lista de artistas.

Ejemplo de validación: Regex y validación de formato YYMMDD#####

3. USO DE INTERFACES Y ABSTRACCIÓN
- MultimediaContent (interfaz)
- PricingStrategy (estrategia de precios)
- MusicalProduct (abstracta)

Beneficios: Extensibilidad y desacoplamiento.

4. ENUMERACIONES
MediaType: { CD, DVD }
Razones: Seguridad de tipos, eficiencia, mantenibilidad.

5. RELACIONES ENTRE CLASES
- Store -> Employee: Composición 1..*
- Disc -> MultimediaContent: Agregación 1..*
- Song -> Album: Asociación 1..1

6. DISEÑO DE REPORTES
- Tendencias de géneros: Media

Informe de Diseño del Sistema de Gestión de Tienda de Discos

- Distribución de servicios por soporte: Baja
- Impacto de colaboradores: Media-Alta
- Costo-beneficio: Alta

7. OPERACIONES CRUD

Estructura base con validaciones (ej: clase Utils).

8. PRINCIPIOS DE DISEÑO

- SOLID: SRP, OCP
- DRY: Reutilización
- Encapsulación: Atributos privados con getters

CONCLUSIÓN

Diseño alineado al dominio, robusto y escalable.

Recomendaciones: BD relacional, API REST, pruebas JUnit.