

# TRABAJO FINAL ANTEPROYECTO

## REQUISITOS CLIENTE

Se necesita un programa para gestionar una tienda de música.

## ESPECIFICACIONES

Existirán tres tipos de productos: discos, libros y merchandising.

El programa debe permitir:

- Añadir y eliminar productos del catálogo.
- Gestión tienda:
  - Añadir
  - Eliminar.
- Búsqueda:
  - Por autor
  - Por disponibilidad o no de libros.
- Mostrar el catálogo.
- Gestión de ficheros:
  - Nuevo
  - Abrir.
  - Guardar / GuardarComo

Cada categoría de producto tendrá una posibilidades diferentes en cuanto a la tienda y prestación de servicios al usuario.

Al no abordar para este proyecto la gestión de cartera de clientes, se crearán dos usuarios para su prueba. Uno de ellos tendrá un carácter estándar y otro premium.

1. Los discos se podrán vender, bien sea en formato CD o MP3, y un usuario estándar tendrá tres escuchas gratuitas al mes pasando a ser ilimitadas si se convierte en usuario premium.
2. Los libros solo se podrán alquilar, teniendo un plazo para su devolución.
3. El merchandising solo se podrá vender.

Los campos de los productos serán:

- Comunes: id, título, autor, categoría, soporte, fecha de lanzamiento, distribuidora, y precio.
- Ampliación para discos: género, numero de canciones, escuchas restantes y escuchable.
- Ampliación para libros: género, días restantes para devolución y alquilable .
- No existirá ampliación para merchandising, todos sus campos serán heredados de l clase padre, producto.

## TÉCNICAS A IMPLEMENTAR

Para el desarrollo de este programa se implementarán todas las técnicas aprendidas durante el curso:

- ✓ **ArrayList (estructuras de datos):** La biblioteca musical en sí misma.
- ✓ **Enumeraciones:** para los diferentes tipos categorías de producto, de soportes en los que se encuentra el álbum, y para los géneros musicales.
- ✓ **Herencia:** Se creará una clase padre llamada Producto y clases hijas como Disco, Libro, y Merchandising. Las clases hijas tendrán un comportamiento diferente entre si: la clase Álbum permitirá unas escuchas máximas mensuales gratuitas, tras superar este límite se impedirá su escucha hasta el mes siguiente, a no ser que se pague cuota. La clase Libro podrá prestarse con un plazo máximo de entrega antes de reclamarlo. Y la clase Merchandising solo se podrá comprar.
- ✓ **Interfaces:** implementaré la interfaz Comparable para ordenar los productos por fechas de lanzamiento. También Serializable para que se puedan tratarse los objetos en los flujos. Por último interfaces creadas como esVendible y esAlquilable, que obligarán a implementar métodos de comportamiento propio de cada producto (calcularPrecio para esVendible, y bajarDiasRestantes, libroDisponible, y calcularAlquiler ambos para esAlquilable).
- ✓ **Flujos de datos y ficheros:** La biblioteca se almacenará en una estructura de ficheros de la que se podrá recuperar la información para consultas.
- ✓ **Excepciones:** se controlarán los datos no válidos mediante excepciones (no coincidencias de la fecha introducida con la expresión regular creada, apertura de ficheros no existentes, consultas mediante titulo no existente, producto no existente, fecha no válida, etc.)
- ✓ **Expresiones regulares:** para las diferentes fechas de lanzamiento.
- ✓ **Fechas:** clase Calendar para manejar los datos referentes a las fechas de lanzamiento del álbum.
- ✓ **GUI:** Me apoyaré en el plug-in WindowBuilder para la creación de la interfaz gráfica.