

PRACTICA 2

SISTEMAS DE INFORMACIÓN

Diseño, desarrollo e instalación de un sistema de
información Web diseño e implementación de la capa de
persistencia de datos

Diego Santolaya Martínez

Abel Chils Trabanco

Alexandru Oarga Hategan

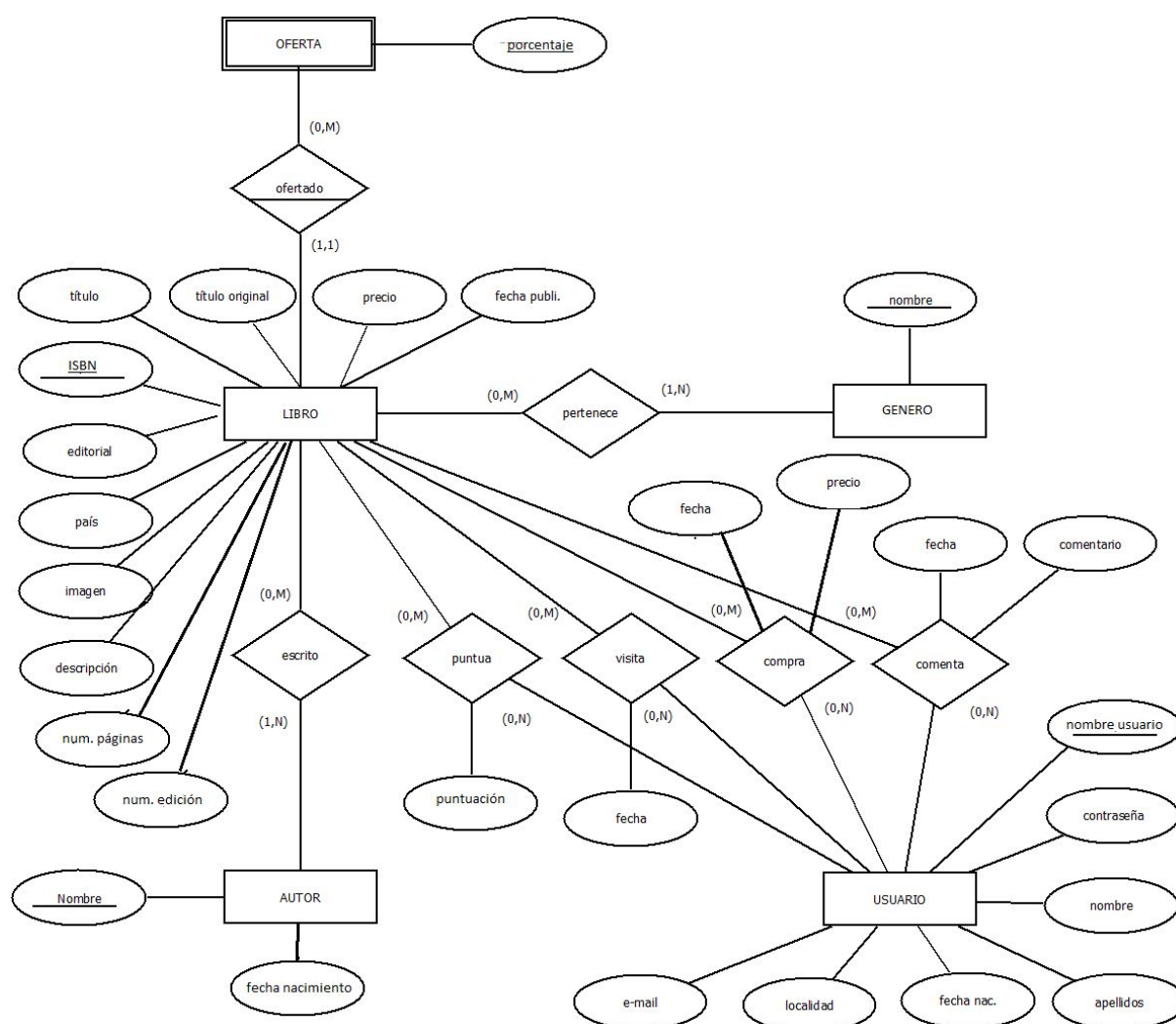
ÍNDICE

1. Introducción	3
2. Diseño del modelo Entidad-Relación	3
3. Modelo relacional	4
4. Implementación en SGBD	4
4.1 Creación de tablas	5
5. Capa de acceso a datos	5
5.1 Patrón VO	5
5.2 Patrón DAO	6
5.2.1 AutorVAO	6
5.2.2 GeneroVAO	6
5.2.3 LibroVAO	6
5.2.4 UsuarioVAO	6
5.3 Gestor de Conexiones	6
6. Estructura del proyecto	7
 Anexo	 8
Metodología	
Horas de trabajo	
 Bibliografía	

1. INTRODUCCIÓN

La persistencia de datos que se va a desarrollar tendrá como finalidad almacenar la información a la que accederá el sistema que se ha estado implementando en las prácticas anteriores. El contexto del sistema de información son las sugerencias de libros y la se almacenará la información referente a los libros(autor, categoría,etc),los usuarios y las acciones que realizan con el sistema (comentarios, compras, etc).

2. DISEÑO DEL MODELO ENTIDAD-RELACIÓN



* RESTRICCIONES:

- fechas no pueden ser nulas
- porcentaje < 100 y porcentaje > 0
- puntuación no puede ser nulo
- email no puede ser nulo

Se ha planteado la entidad 'Usuario' para contener los datos de registro que introduce el cliente que se registra en el sistema. Esto incluye nombre de usuario, nombre, apellidos, email, fecha de nacimiento, localidad, contraseña, teléfono y cuenta bancaria, siendo esta

última información necesaria la compra de un libro. La clave primaria que se utiliza es el nombre de usuario ya que no se podrán repetir nombres de usuario en el sistema. Para garantizar seguridad se almacenan las contraseñas cifradas mediante cifrado asimétrico. También por motivos de privacidad, los datos email, teléfono, dirección y número de cuenta se guardarán con cifrado simétrico.

La entidad 'Libro' contiene los atributos de todos los libros que tiene el sistema. Estos atributos son: ISBN, Título, Título original, Precio, Fecha de publicación, País de publicación, Editorial, Descripción, Número de páginas y Número de edición. La clave primaria será el ISBN por ser el método de identificación propia que utilizan los libros. No se ha considerado Autor como atributo de libro al ser una entidad aparte y no una propiedad del libro. La entidad autor tiene los atributos nombre y país de nacimiento siendo el primero la clave primaria. Autor se relaciona con Libro mediante la relación 'escrito_por' (pudiendo tener un libro más de un autor) y se utilizará para las consultas de libros de mismo autor.

La entidad usuario se relaciona con libro mediante 4 relaciones, una por cada forma de interacción del usuario con el sistema: visita, compra, puntuar y comentar. Estas relaciones se utilizarán para generar el sistema de información, y es por este motivo por lo que cada una tiene el atributo fecha, pues se va a dar más prioridad a las acciones más recientes. Los libros pertenecen también a un género por lo que se ha añadido esta entidad y una relación 'pertenece_al_género' con 'Libro'. Se utilizará para las consultas de libros de un mismo género.

Por último, se ha planteado una entidad oferta, entidad débil de libro, para almacenar ofertas temporales sobre libros determinados y para obtener todos los libros en oferta. La información que se guarda es el porcentaje sobre el precio de un libro y solo se mantendrá mientras el libro esté en oferta. Cómo es posible que el usuario compre libros en oferta, se ha añadido el atributo precio en la relación compra pues el precio puede haber variado según el momento de compra.

3. Modelo relacional

Se utiliza una tabla para usuario, libro, autor y categoría, cada una con sus atributos.

Las relaciones visita, compra, puntúa, comenta, escrito por y pertenece_al_género al ser de cardinalidad N-M se han planteado en una tabla cada una con los atributos indicados en el apartado anterior. Cada tabla tiene una relación con las claves primarias que relaciona.

LIBRO	ISBN	título	título original	precio	fecha publi.	editorial	país
	precio	num. Páginas	num. Edición	idioma	desc. Corta	descripción	
GENERO	nombre						
AUTOR	nombre	país nacimiento					
USUARIO	nombre_de_us.	contraseña	nombre	apellidos	fecha nacim.	localidad	e-mail
OFERTA	libro	porcentaje					
Refs:	LIBRO(ISBN)						
PERTENECE	libro	género					
Refs:	LIBRO(ISBN)	GENERO(nombre)					
ESCRITO	libro	autor					
Refs:	LIBRO(ISBN)	AUTÓR(nombre)					
COMPRA	libro	usuario	fecha	precio			
Refs:	LIBRO(ISBN)	USUARIO(nombre de)					
VISITA	libro	usuario	fecha				
Refs:	LIBRO(ISBN)	USUARIO(nombre de)					
COMENTA	libro	usuario	fecha	comentario			
Refs:	LIBRO(ISBN)	USUARIO(nombre de)					
PUNTUA	libro	usuario	puntuación				
Refs:	LIBRO(ISBN)	USUARIO(nombre de)					

4. IMPLEMENTACIÓN EN S.G.B.D.

El SGBD donde se ha implementado el modelo ha sido MySql.

4.1 CREACIÓN DE TABLAS

Se ha elaborado un script sql para la creación de las tablas anteriormente planteadas. Se utiliza el comando CREATE_TABLE. Algunas de las decisiones que cabe destacar son:

- Numero de paginas y numero de edicion de libro se definen como datos INTEGER, precio como REAL, ISBN como cadena de caracteres (VARCHAR), fecha de publicación como DATE y descripción y descripción corta como cadenas de caracteres de 1000 caracteres (VARCHAR(1000))
- La clave primaria de las relaciones de cardinalidad N-M son las claves primarias de las entidades que relaciona.
- En libro _en_ oferta, porcentaje_oferta se define como REAL NOT NULL. Además se comprueba que se cumpla la restricción “((porcentaje_rebaja > 0.0) AND (porcentaje_rebaja < 100.0))” con la etiqueta CHECK.
- En usuario la clave primaria es nombre de usuario pero para satisfacer el requisito de que no se repitan emails, se utiliza el comando “UNIQUE(email)”
- Los atributos fecha de visita, compra, puntua, comenta se definen como datos DATE que no pueden ser nulos (NOT NULL). El atributo precio (REAL) tampoco puede ser nulo.

5. CAPA DE ACCESO A DATOS

Para la capa de acceso de datos se utilizan los patrones VO y DAO y se implementan en Java.

5.1 VALUE OBJECT (VO)

A un nivel más bajo, para abstraer los datos del sistema de persistencia, se ha utilizado el patrón VO. Se ha hecho una clase para cada uno de los siguientes elementos a encapsular: Autor, Comentario, Compra, Genero, Libro, Usuario, Valoración y Visita. Los atributos de la clase se han declarado privado y se ha dotado a las clases de métodos get y set. De esta

forma se ha conseguido cierta independencia con respecto al sistema de almacenamiento y se proporciona un modelo abstracto para los datos.

5.2 DATA ACCESS OBJECT (DAO)

El patrón DAO se implementa para encapsular el acceso al sistema de almacenamiento. La capa DAO trabajará a su vez con los objetos declarados anteriormente.

Se han implementado en total 4 clases: AutorDAO, LibroDAO, GeneroDAO y UsuarioDAO. No se han implementado las clases comentario, compra, valoración y visita por ser acciones que se realizan entre el libro y el usuario, es decir, no se trabajará con ellas directamente sino que se utilizarán siempre acompañadas del usuario que las realiza.

En los siguientes apartados se comentan algunos detalles de implementación de cada clase.

5.2.1 AutorDAO

La clase está dotada de los métodos: insertarAutor, actualizarActor y eliminarActor. Estos métodos reciben como parámetro la información del autor en un objeto VO y un tipo de dato Connection (la conexión con el S.G.) para la ejecución de los query SQL. En estos métodos se utilizan los comandos SQL INSERT, UPDATE Y DELETE respectivamente.

El método encontrarDatosAutor con parámetro String devuelve en un dato AutorVO la información del autor indicado en la cadena si existiese y el mismo método solo con un parámetro Connection, devuelve una lista de estos objetos que contiene todos los autores almacenados. Para ambos métodos se utiliza una consulta SQL (SELECT).

Por último el método encontrarLibrosAutor devuelve una lista de objetos LibroVO escritos por el autor que se pasa por parámetro. Para ello, se ejecuta la consulta de los libros escritos por dicho autor y se hace un INNER JOIN con dichos libros para obtener la información de estos.

5.2.2 GeneroVAO

Al igual que la clase anterior está dotada de los métodos insertarGenero, eliminarGenero y encontrarGenero (devuelve los géneros existentes en una lista de objetos GeneroVO).

Se implementa también el método existeGenero que tiene como resultado un booleano que indica si la consulta sobre el nombre de un género devuelve algún valor.

5.2.3 LibrosVAO

Al igual que las clases anteriores tiene métodos que trabajan con la información de libros (con la clase LibroVO). La diferencia con las anteriores es que contiene también métodos para la inserción, modificación y eliminación de comentarios, valoraciones, comprar y puntuaciones (cada una trabaja con su clase VO). Las consultas realizadas en esta clase no tienen operaciones costosas como por ejemplo JOIN.

5.2.4 UsuarioVAO

De forma semejante a la clase LibrosVAO, además de los métodos para trabajar con los datos del usuario, la clase contiene métodos para trabajar con comentarios, valoraciones, compras y puntuaciones que se asignan a determinado usuario.

5.3 GESTOR DE CONEXIONES

La clase GestorDeConexiones se utiliza para obtener la clase Connect que utilizan las clases DAO para la ejecución de las instrucciones SQL. La clase se encarga de establecer la conexión con el S.G.B.D.

6. ESTRUCTURA DEL PROYECTO

La estructura de ficheros que se ha seguido en el proyecto es la siguiente:

```
/vista
    ...
/modelo
    TiendaFacade.java
/datos
    /VO
        AutorVO.java
        ComentarioVO.java
        CompraVO.java
        GeneroVO.java
        LibroVO .java
        UsuarioVO.java
        ValoracionVO.java
        VisitaVO.java
    /DAO
        AutorDAO.java
        LibroDAO.java
        GeneroDAO.java
        UsuarioDAO.java
/GestorDeConexiones
    GestorDeConexiones.java
```

ANEXO

METODOLOGÍA

- Alex Oarga: Redacción de la memoria
- Abel Chis: Creación de tablas SQL, Implementación de la capa de acceso de datos.
- Diego Santolaya: Diseño ER, Implementación de la capa de acceso de datos.

COSTES

- Alex Oarga: 2h de prácticas + 5h memoria
- Abel Chis: 2h de prácticas + 9h desarrollo
- Diego Santolaya: 2h de prácticas + 7h desarrollo