

## Capítulo 1

# Acceso a bases de datos

### 1.1. Conceptos básicos



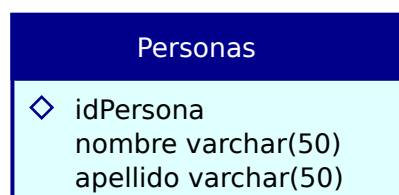
#### Teórico

- JDBC. Drivers.
- Carga del driver. Clase DriverManager.
- Conexiones. Cadenas de conexión. Clase Connection.
- Ejecución de sentencias. Clases Statement y ResultSet.

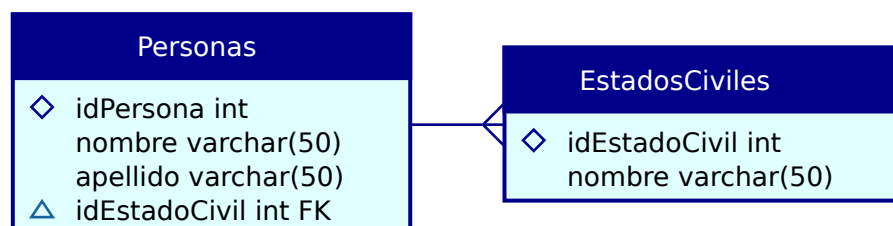


#### Práctico

1. Desarrollar un programa de consola que establezca una conexión a la base de datos Clase1, luego consulte la tabla Personas y muestre un listado del contenido de las columnas idPersona, nombre y apellido.



2. Desarrollar un programa con una ventana de carga de personas. El estado civil debe ser ingresado mediante un control de selección (JComboBox o JList).



## 1.2. Desarrollo en capas



### Teórico

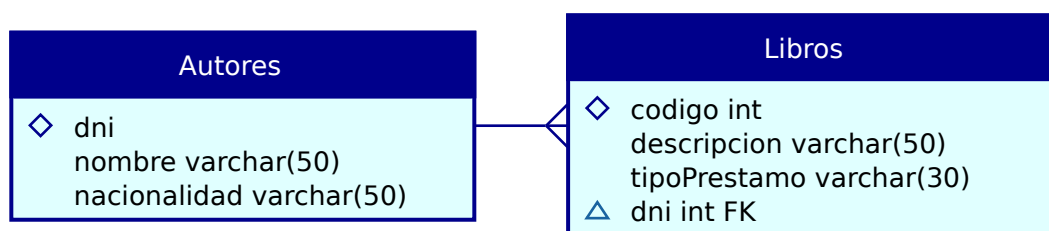
- Desarrollo en capas
- Patrón básico. Paquetes modelo, controlador e interfaz.



### Práctico

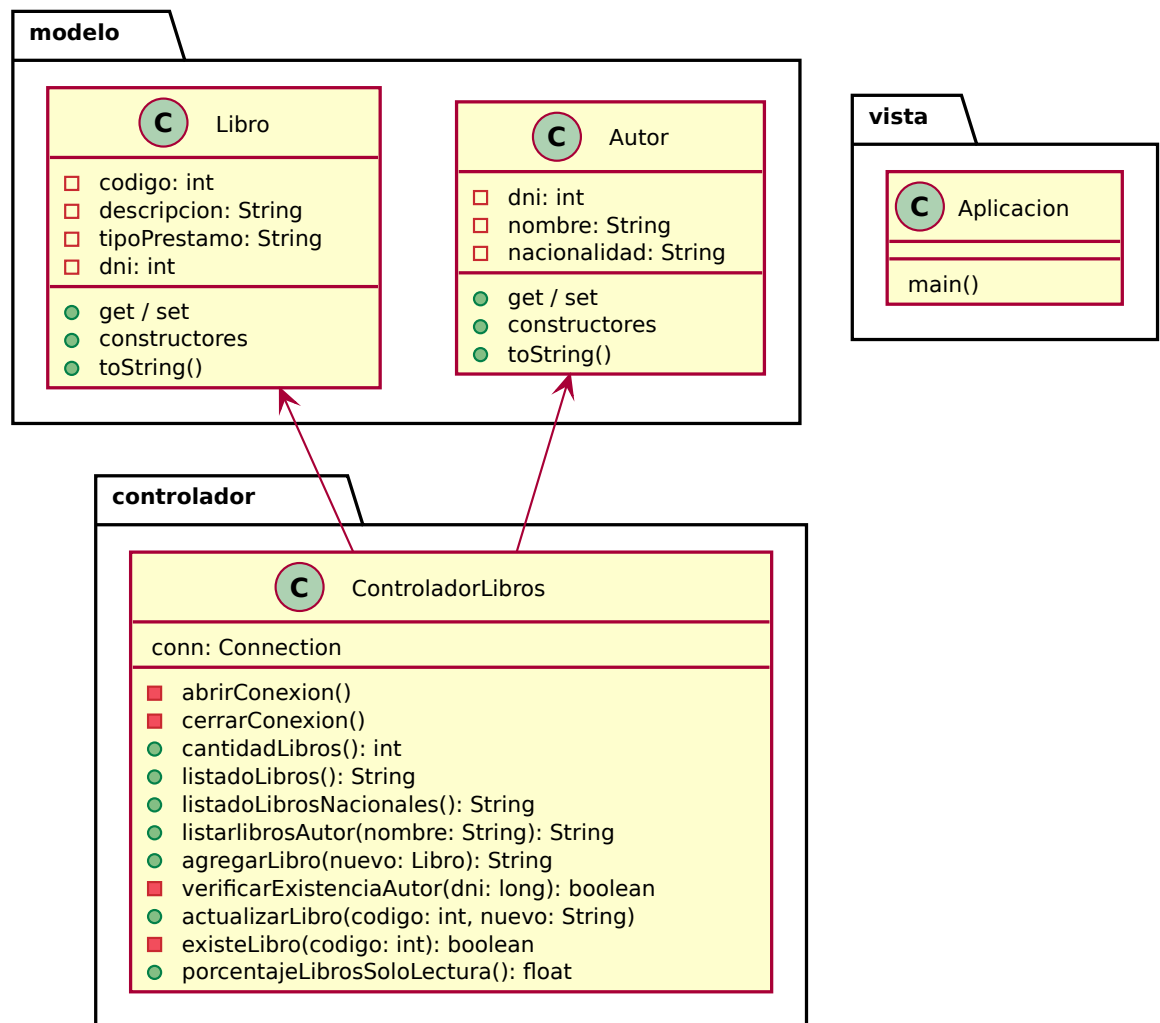
1. Desarrollar una aplicación Java que permita efectuar las siguientes las siguientes consultas:
  - a) Cantidad de libros registrados.
  - b) Informar un listado de libros con los siguientes datos: descripción del libro, nombre y nacionalidad del autor.
  - c) Informar un listado de libros cuyos autores sean de origen nacional. Los datos a mostrar código y nombre del libro y nombre del autor.
  - d) Para un determinado Nombre del autor liste sus datos y los libros de su autoría.
  - e) Método que permita la inserción de libros. Para esto previamente debe verificar que el DNI del autor exista en la tabla autores, en el caso de no existir no permitir la inserción del nuevo libro.
  - f) Método que permita la modificación de la descripción de un determinado libro. El número de libro debe ser ingresado por teclado.
  - g) Porcentaje de libros de la biblioteca que son de sólo lectura.

La Base de Datos que se usará para realizar la práctica se llama: Biblioteca



Implementar el siguiente esquema en el proyecto, respetando la siguiente estructura de paquetes:

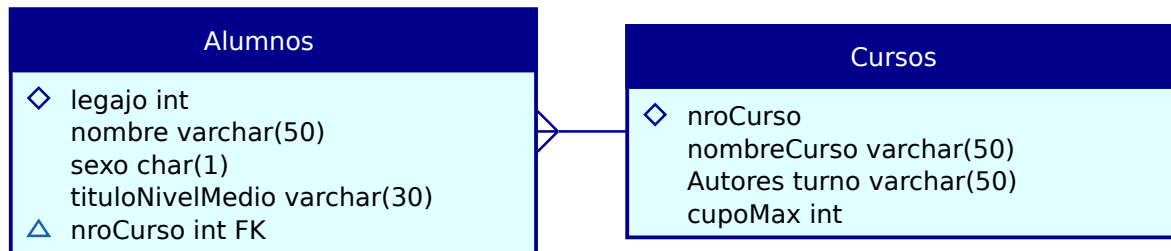
- Modelo: contiene las clases Autor y Libro.
- Controlador: contendrá la clase ControladorLibros, en la que se implementará las operaciones relacionados con el acceso a la base de datos.
- Vista: contendrá la clase Aplicación, que contiene el método main(). En esta clase debe invocar los métodos implementados en la clase ConexionJDBC.



2. Desarrollar una aplicación Java que permita efectuar las siguientes las siguientes operaciones y consultas:

- Insertar un curso.
- Insertar un alumno. Verificar antes de insertar si el cupo no ha sobrepasado su capacidad máxima.
- Modificar los datos de un alumno específico.
- Porcentaje de alumnos por género para un determinado curso.
- Informar para un determinado curso, los datos de todos los alumnos inscriptos.
- Informar los datos de los alumnos, para un determinado título alcanzado en nivel medio.
- Listar los cursos y alumnos que pertenecen al turno tarde.

La Base de Datos que se usará para realizar la práctica se llama: Académica



Implementar el siguiente esquema en el proyecto, respetando la siguiente estructura de paquetes:

- Modelo: contiene las clases Curso y Alumno.
- Controlador: contendrá la clase GestorCursos, en la que se implementará las operaciones relacionados con el acceso a la base de datos.(Que son las solicitadas al comienzo del enunciado)
- Vista: contendrá la clase Aplicación, que contiene el método main(). En esta clase debe invocar los métodos implementados en la clase ConexionJDBC.

### 1.3. ABM con interfaz de ventanas



#### Teórico

- Modelo MVC con interfaz Swing
- Control JComboBox. Manipulación de items. Clase DefaultComboBoxModel
- Control JTable. Clase DefaultTableModel.



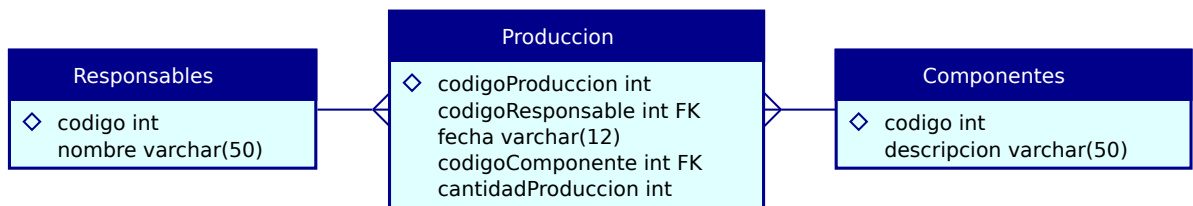
#### Práctico

1. Una empresa dedicada a la producción de componentes (Arandelas, Tornillos, Tuercas), nos ha solicitado la implementación de un programa que facilite la registración de las producciones de dichos componentes, como así también la gestión de información estadística relacionada con el rendimiento de producción de los responsables del área de Producción de esta empresa.

Diseñar un menú que habilite efectuar las siguientes operaciones:

- a) Registrar las órdenes de producción.
- b) Consultar las producciones realizadas para un determinado operario y totalizar la cantidad de producciones
- c) Permitir la modificación de los datos una determinada orden de producción. Los datos que pueden editarse son la fecha, y la cantidad de producción.

La Base de Datos que se usará para realizar la práctica se llama: Producciones



Implementar el siguiente esquema de paquetes en el proyecto:

- **Modelo:** contiene las siguientes clases:
  - **Produccion:** que tiene los siguientes atributos: codigoProduccion(int), codigoResponsable (int), fecha (String), codComponente (int), cantidadProduccion (int).
  - **Responsable:** código (int), nombreResponsable (String).
  - **Componente:** codComponente (int), descComponente (String)
- **Controlador:** contendrá la clase GestorProduccion, en la que se implementará los métodos relacionados con el acceso a la base de datos.
- **Vista:** se propone el siguiente esquema de interfaces:
  - **Menú:** pantalla permite el acceso a la interfaces que permiten la registración y la consulta de producción.
  - **Alta Producción:** pantalla permite el ingreso de todos los datos de la producción de cada operario para su posterior registración.
  - **Consulta Producción:** pantalla permite realizar el filtro de un determinado operario visualizando todas sus producciones y totalizando la cantidad total de componentes producidos.
  - **Reportes:**
    - Cantidad de producciones por cada responsable.
    - Cantidad total producida por cada componente.
    - Listar nombre del responsable, nombre del componente, fecha y cantidad producida de las producciones de más de 100 unidades.
    - Promedio general de cantidad producida.
    - Porcentaje de ordenes de producción con cantidad producida menor a 10 sobre el total del producciones.