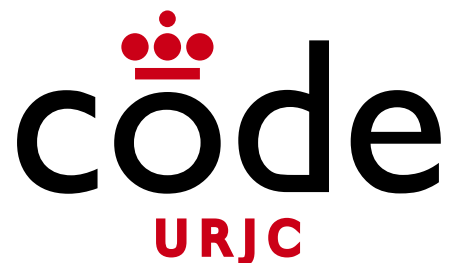


## Desarrollo Web

### Tecnologías de servidor web

# Tema 2.5: Gestión del esquema y Arquitectura



©2025

Micael Gallego, Francisco Gortázar, Michel Maes, Óscar Soto, Iván Chicano

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Gestión del esquema y Arquitectura

- Gestión del esquema
- Arquitectura

# Gestión del esquema

- En las **Bases de datos SQL** es necesario **crear el esquema** antes de insertar los datos (crear tablas, relaciones, etc...)
- En las bases de **datos en memoria** (H2, Derby, HSQL...), el esquema siempre se construye de forma **automática** al iniciar la aplicación
- Cuando se usa una base de datos en producción los datos tienen que guardarse en disco y gestionar adecuadamente la **creación y actualización del esquema**

- Instalación en ubuntu
- Servidor

```
$ sudo apt-get install mysql-server
```

- Herramienta interactiva

```
$ sudo apt-get install mysql-workbench
```



- Docker

```
$ docker run --rm -e MYSQL_ROOT_PASSWORD=password \  
-e MYSQL_DATABASE=posts -p 3306:3306 -d mysql:8.0.33
```

<http://dev.mysql.com/downloads/>

# MySQL

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar contains a 'MANAGEMENT' section with links to Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore. Below this is an 'INSTANCE' section with Startup / Shutdown, Server Logs, and Options File. The 'PERFORMANCE' section includes Dashboard, Performance Reports, and Performance Schema Setup. The 'SCHEMAS' section has a search bar and a tree view showing the 'libros' database with its tables, columns, and indexes. The main query editor displays a query: `SELECT titulo, precio FROM Libros WHERE precio > 2`. The 'Action Output' pane at the bottom shows a list of actions and their results. The first action, 'SELECT titulo, precio FROM Libros WHERE precio > 2 ...', failed with Error Code 1046: 'No database selected. Select the default DB to be used by double-click...'. Subsequent actions to apply changes and limit queries succeeded. The status bar at the bottom indicates 'Query interrupted'.

MySQL Workbench

127.0.0.1 x 127.0.0.1 x Localhost x

File Edit View Query Database Server Tools Scripting Help

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

libros

- Tables
- libros
  - Columns
  - Indexes

Object Info Session

Table: libros

Columns:

- idLibro int(11) PK
- titulo text
- precio decimal(10,0)

Query 1 x new\_schema - Schema x libros - Table x libros x libros x libros.libros x

Limit to 1000 rows

```
1 • SELECT titulo, precio
2 FROM Libros
3 WHERE precio > 2
4
```

Action Output

	#	Time	Action	Message	Dura
✗	1	23:56:17	SELECT titulo, precio FROM Libros WHERE precio > 2 ...	Error Code: 1046. No database selected Select the default DB to be used by double-click...	0,00
✓	2	23:56:31	Apply changes to libros	Changes applied	
✓	3	23:58:30	Apply changes to libros	Changes applied	
✓	4	23:58:53	SELECT * FROM libros.libros LIMIT 0, 1000	0 row(s) returned	0,00
✓	5	00:00:44	SELECT * FROM libros.libros LIMIT 0, 1000	0 row(s) returned	0,00
✗	6	00:01:06	Refresh Recordset	There are pending changes. Please commit or r...	
✓	7	00:01:58	SELECT * FROM libros.libros LIMIT 0, 1000	0 row(s) returned	0,00
✗	8	00:02:26	SELECT titulo, precio FROM Libros WHERE precio > 2	Error Code: 1146. Table 'libros.libros' doesn't e...	0,00

Query interrupted

# MySQL

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' panel lists 'optsicom' and 'test'. The main area displays the 'new\_schema - Schema' dialog box. The 'Name' field is set to 'new\_schema'. The 'Default Collation' is set to 'utf8mb4\_0900\_ai\_ci'. The 'Comments' field is empty. The 'Action Output' panel at the bottom is empty. A green box labeled 'Crear esquema' points to the 'new\_schema' button. Another green box labeled 'posts' points to the 'Name' field.

MySQL Workbench

Local x Local x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

SCHEMAS

Filter objects

- optsicom
- test

Object Info Session

No object selected

new\_schema - Schema x SQL File 5 x

Name: new\_schema

Rename References

Default Collation: utf8mb4\_0900\_ai\_ci

Comments:

The name of the schema. It is recommended to use lowercase letters and numbers. Spaces should be avoided.

Refactor model, changing all references, triggers, stored procedures and functions to the new schema name.

Specifies which charset/collation to use if they do not have an explicit set.

Common choices are Latin1 or UTF8.

Schema

Action Output

Added new script editor

- pom.xml

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.33</version>
</dependency>
```

Dependencia a **Spring Data** para BBDD relacionales

Dependencia al driver **MySQL**



- Datos de conexión a la BBDD

Este fichero tiene que estar en el fichero **src/main/resources**

**Host** en el que está alojado el servidor MySQL

Nombre del **esquema**. Este esquema tiene que ser creado manualmente por el desarrollador usando herramientas de MySQL

**application.properties**

```
spring.datasource.url=jdbc:mysql://localhost/posts
spring.datasource.username=root
spring.datasource.password=password
spring.jpa.hibernate.ddl-auto=...
```

**Usuario y contraseña.**  
En algunos SO estos datos se configuran al instalar MySQL. La contraseña debe tener al menos 8 caracteres.

# Gestión del esquema

ejem18

- **Propiedad de generación del esquema**

```
spring.jpa.hibernate.ddl-auto=...
```

- **create-drop:** Crea el esquema al iniciar la aplicación y le borra al finalizar (ideal para programar como si la BBDD estuviera en memoria)
- **create:** Crea el esquema al iniciar la aplicación
- **update:** Añade al esquema actual las tablas y atributos necesarios para hacer el esquema compatible con las clases Java (no borra ningún elemento). Si el esquema está vacío, se genera completo.
- **validate:** Verifica que el esquema de la BBDD es compatible con las entidades de la aplicación y si no lo es genera un error.
- **none:** No hace nada con el esquema y asume que es correcto.

# Gestión del esquema

- Estrategia de desarrollo con BBDD
  - **Fase 1:** Desarrollo
  - **Fase 2:** Despliegue
  - **Fase 3:** Ejecución
  - **Fase 4:** Actualización de versión

# Gestión del esquema

- **Estrategia de desarrollo con BBDD**
  - **Fase 1: Desarrollo: ddl-auto=create-drop**
    - Cualquier cambio en las entidades se refleja de forma automática en la BBDD al arrancar la aplicación
    - Se suele trabajar con datos iniciales de desarrollo para que reiniciar la aplicación no requiera la introducción manual de datos

# Gestión del esquema

- **Estrategia de desarrollo con BBDD**
  - **Fase 2: Despliegue: ddl-auto=update**
    - Al iniciar la aplicación por primera vez se genera el esquema.
    - Al iniciar la aplicación por segunda vez el esquema no se modifica (porque la app no ha cambiado)
    - Si se quiere evitar el tiempo extra que requiere comprobar el esquema, se puede configurar la gestión del esquema a "none"

# Gestión del esquema

- Estrategia de desarrollo con BBDD
  - Fase 3: Ejecución: ddl-auto=none
    - El esquema no debe modificarse en las siguientes ejecuciones de la aplicación

# Gestión del esquema

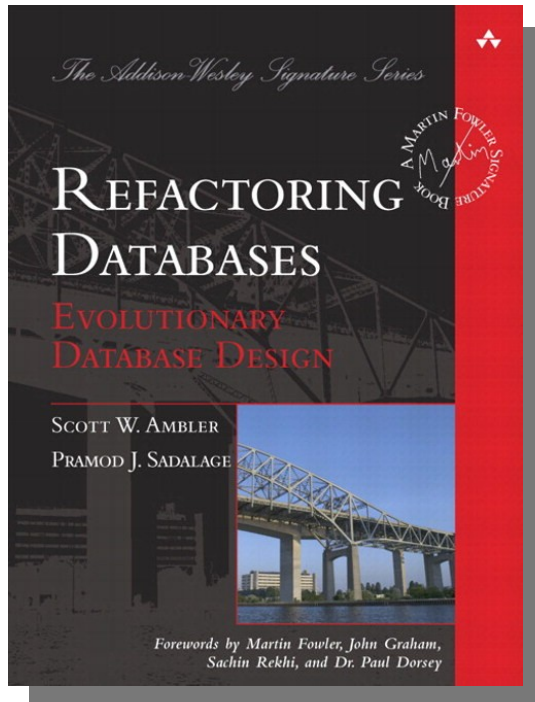
- Estrategia de desarrollo con BBDD
  - Fase 4: Actualización: Usar herramientas específicas
    - Si una nueva versión de la aplicación requiere cambios en las tablas es necesario **aplicar los cambios pero sin alterar los datos (migración del esquema)** antes de ejecutar la nueva versión
    - Para ello se utilizan herramientas específicas

# Gestión del esquema

- **Migración del esquema de la BBDD**
  - Cuando una aplicación con BBDD se usa en **producción** se **guardan datos** en la misma que se deben mantener
  - Es posible que la aplicación **evolucione**:
    - Incluyendo nuevas entidades
    - Modificando las entidades existentes
  - Para **actualizar** la aplicación en **producción**, antes hay que **migrar** la BBDD al nuevo esquema



# Gestión del esquema



## Refactoring Databases: Evolutionary Database Design

by Scott W. Ambler and  
Pramod J. Sadalage

Addison Wesley Professional  
(2006)

<http://www.ambysoft.com/books/refactoringDatabases.html>

# Gestión del esquema

- Existen dos herramientas externas que facilitan la evolución del esquema:



<http://flywaydb.org/>

**Database Migrations Made Easy**

Gestiona la migración con sentencias SQL específicas de la BBDD



<http://www.liquibase.org/>

**Source Control for your Database**

Gestiona la migración con XML genérico independiente de la de la BBDD

# Gestión del esquema

- **Flyway y Liquibase** mantienen una **lista de cambios** que hay que hacer en el esquema para **evolucionar/migrar** de una versión a otra superior
- Cuando la aplicación **Spring** se **inicia**, se pueden ejecutar de forma **automática estos cambios** para convertir el esquema en la versión deseada

<https://www.baeldung.com/database-migrations-with-flyway>

<https://thorben-janssen.com/database-migration-with-liquibase-getting-started>

# Gestión del esquema y Arquitectura

- Gestión del esquema
- **Arquitectura**

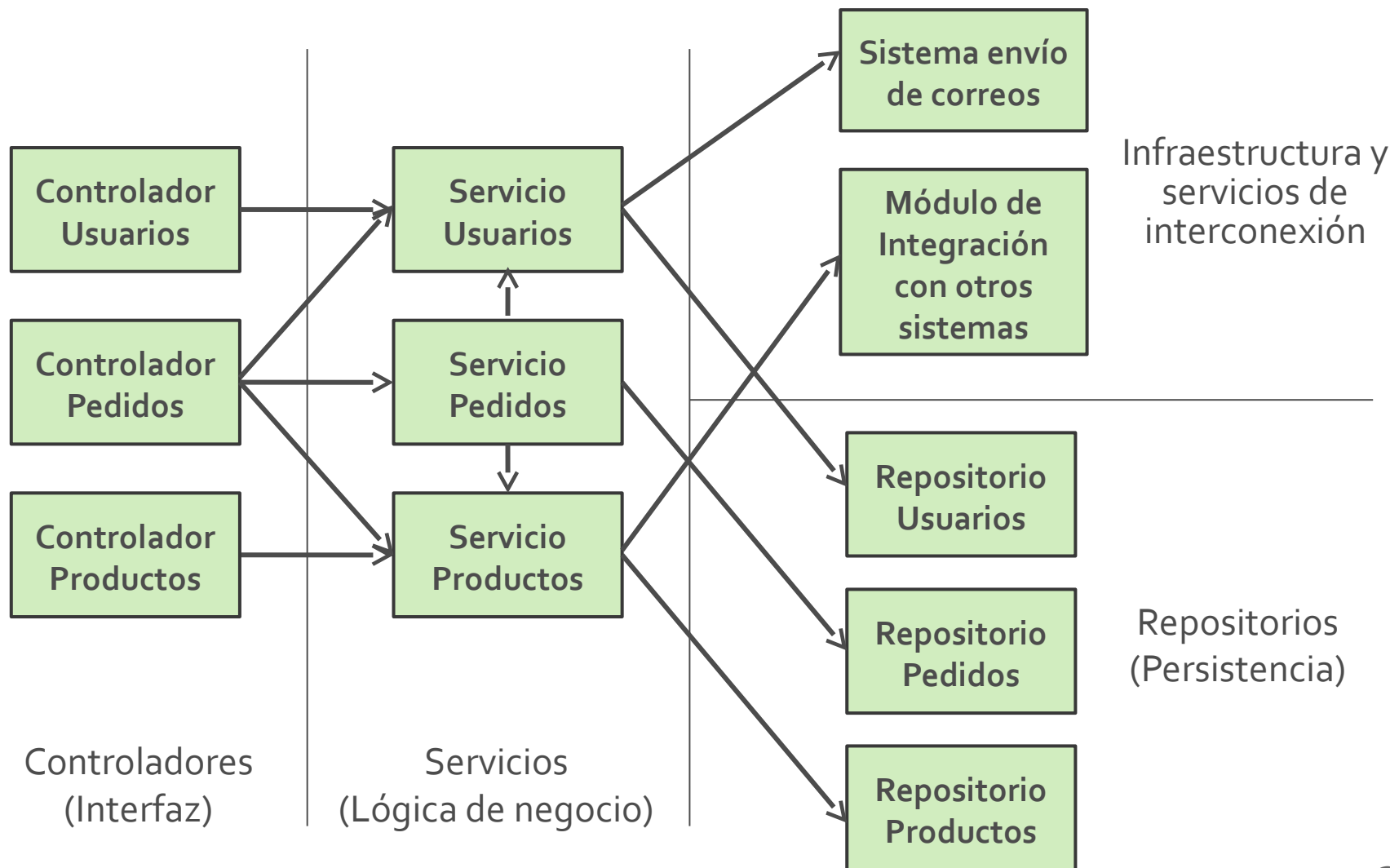
# Arquitectura

- La **arquitectura de software** define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.
- Existen diferentes **estilos arquitectónicos** que pueden ser adecuados para diferentes tipos de aplicaciones

# Arquitectura

- Las aplicaciones Spring con BBDD suelen usar una arquitectura en la que las clases se agrupan por responsabilidades:
  - **Controladores:** Gestión de consultas web o REST
  - **Entidades:** Elementos básicos de información que se persisten en la BD. Modelo.
  - **Repositorios:** Guardar y consultar entidades
  - **Servicios:** Lógica de negocio que gestiona entidades
  - **Infraestructura:** Envío de mails
  - **Servicios de interconexión**

# Arquitectura



# Arquitectura

- **Implementación**
  - **Controladores:** @Controller o @RestController
  - **Entidades:** @Entity
  - **Repositorios:** implements JpaRepository
  - **Servicios:** @Service
  - **Infraestructura:** @Service
  - **Servicios de interconexión:** @Service

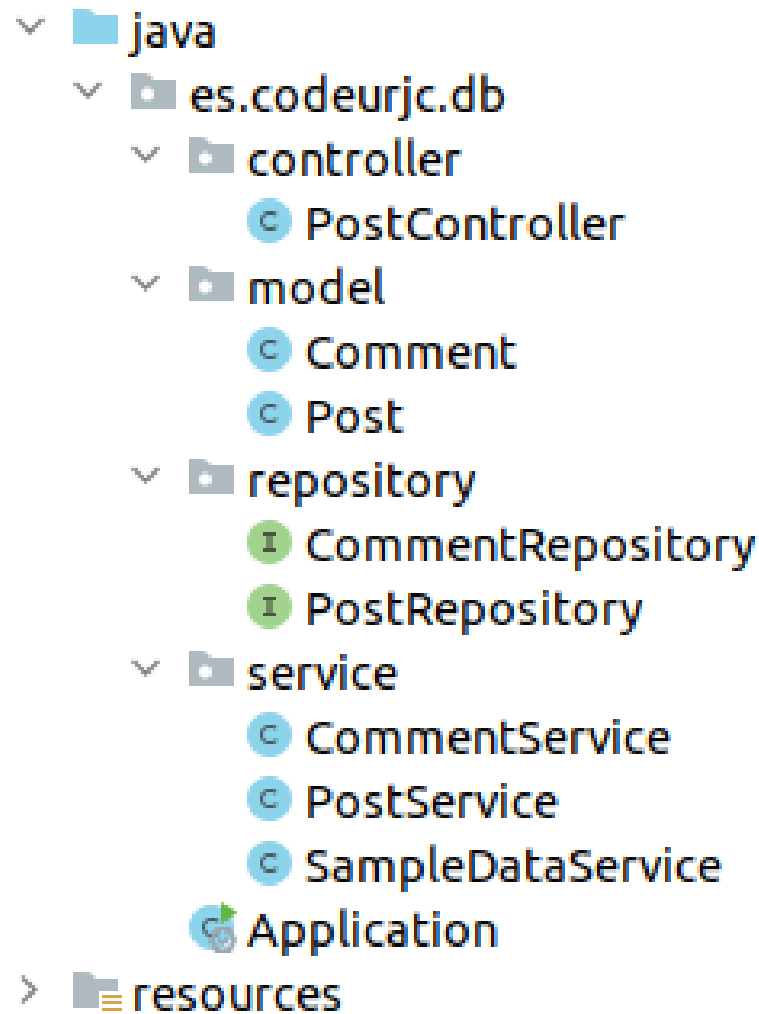


# Arquitectura

- **Ventajas de esta arquitectura:**
  - Los **servicios** pueden testearse de forma unitaria (con dobles de repositorios y otros servicios)
  - Los **controladores** pueden testearse de forma unitaria (con doble del servidor web y de los servicios)
  - La **lógica de negocio** se puede reutilizar entre varios tipos de controladores (web, REST, línea de comandos...)

# Arquitectura

ejem19



```
graph TD; java --> es_codeurjc_db[es.codeurjc.db]; java --> resources[resources]; es_codeurjc_db --> controller; es_codeurjc_db --> model; es_codeurjc_db --> repository; es_codeurjc_db --> service; controller --> PostController; model --> Comment; model --> Post; repository --> CommentRepository; repository --> PostRepository; service --> CommentService; service --> PostService; service --> SampleDataService; Application
```

The image shows a project structure in a code editor. The root is 'java', which contains 'es.codeurjc.db' and 'resources'. 'es.codeurjc.db' is expanded to show 'controller', 'model', 'repository', and 'service'. 'controller' contains 'PostController'. 'model' contains 'Comment' and 'Post'. 'repository' contains 'CommentRepository' and 'PostRepository'. 'service' contains 'CommentService', 'PostService', and 'SampleDataService'. There is also an 'Application' class at the bottom level of the 'java' package.

# Arquitectura

