Programación didáctica — Acceso a Datos (DAM 2º)

Enfoque: Java con Spring Boot

---

1. Identificación del módulo

- **Módulo:** Acceso a Datos
- **Código:** 0486
- **Ciclo formativo:** Técnico Superior en Desarrollo de Aplicaciones Multiplataforma
- **Duración:** 120 horas (9 ECTS)
- **Curso:** 2º DAM

2. Resultados de aprendizaje (RA) y criterios de evaluación (CE)

RA1. Maneja ficheros y flujos en Java.
RA2. Desarrolla aplicaciones con acceso a bases de datos relacionales mediante JDBC y frameworks.
RA3. Utiliza herramientas ORM (Hibernate / Spring Data JPA).
RA4. Accede a bases de datos no relacionales (MongoDB).
RA5. Maneja XML y bases de datos nativas XML.
RA6. Desarrolla componentes de acceso a datos e integra la persistencia en aplicaciones.

3. Temporalización y unidades didácticas (UD)

UD 1. Acceso a ficheros y flujos (≈ 20 h)

- Manejo de ficheros en Java (IO y NIO).
- Serialización y deserialización de objetos.
- JSON con Jackson / Gson.
- XML: DOM, SAX, StAX.
- Validación con DTD/XSD.
  (RA1, RA5)

UD 2. Acceso a bases de datos relacionales (≈ 20 h)

- Conexión JDBC a MySQL/PostgreSQL.
- CRUD desde Java.
- Transacciones, bloqueos y seguridad.
- Spring JDBC Template.
- Miniinterfaz web con **Thymeleaf** o **Vaadin** (formulario CRUD).
  (RA2, RA6)

UD 3. Herramientas de mapeo objeto-relacional (ORM) (≈ 20 h)

- Introducción a JPA.
- Hibernate + Spring Boot.
- Anotaciones (@Entity, @Id, @OneToMany...).
- Repositorios Spring Data JPA.
- Consultas con JPQL y Query Methods.
  (RA3, RA6)

UD 4. Bases de datos NoSQL: MongoDB (≈ 20 h)

- Instalación y configuración de MongoDB.
- Operaciones CRUD desde Spring Boot.
- Consultas con repositorios MongoRepository.
- Comparativa SQL vs NoSQL.
  (RA4, RA6)

UD 5. Bases de datos nativas XML y orientadas a objetos (≈ 20 h)

- Almacén XML nativo (existDB, BaseX).
- Consultas con XPath y XQuery.
- Bases de datos orientadas a objetos.
- Bases de datos objeto-relacionales.
  (RA5)

UD 6. Programación de componentes de acceso a datos (≈ 20 h)

- Diseño de una capa DAO/Repository.
- Integración en arquitectura por capas.
- Seguridad y control de transacciones.

- Pruebas unitarias con JUnit y MockMVC.

- Proyecto integrador: API REST con Spring Boot + MySQL/MongoDB.
  (RA6)

4. Metodología didáctica

- **ABP (Aprendizaje Basado en Proyectos):** cada UD cierra con un mini-proyecto práctico.

- **Metodologías activas:** pair programming, uso de GitHub Classroom, aprendizaje cooperativo.

- **Aprendizaje significativo:** conexión de los contenidos con proyectos reales (apps web empresariales).

- **Evaluación continua:** entregables parciales, participación, prácticas evaluables.

Recursos

- **Lenguaje:** Java 17+.
- **Frameworks:** Spring Boot, Spring Data JPA, Spring Web.
- **BBDD:** MySQL/PostgreSQL, H2 (tests), MongoDB, existDB/BaseX.
- **Herramientas:** IntelliJ IDEA, Docker, Postman/Insomnia, GitHub.
- **Librerías:** Jackson/Gson, Hibernate ORM, JUnit5.

UD 1. Acceso a ficheros, flujos, serialización, JSON y XML

Conceptos técnicos:

- **Java IO y NIO2:**
    - Clases `File`, `FileReader`, `FileWriter`, `BufferedReader`, `BufferedWriter`.
    - `InputStream` y `OutputStream` para ficheros binarios.
    - Uso de `Files` y `Paths` (API NIO2).
- **Serialización de objetos:**
    - Interfaces `Serializable` y `Externalizable`.
    - Métodos `ObjectOutputStream` y `ObjectInputStream`.
    - Riesgos de seguridad en la serialización.
- **Ficheros JSON:**
    - Librerías Jackson (`ObjectMapper`) y Gson.
    - Conversión POJO ↔ JSON.
    - Validación contra esquemas JSON.
- **Ficheros XML:**
    - APIs DOM (`DocumentBuilder`), SAX (`XMLReader`), StAX (streaming).
    - Validación con DTD y XSD.
    - Transformación con XSLT (`Transformer`).
    - Marshalling/unmarshalling con JAXB.

UD 2. Acceso a bases de datos relacionales (locales y remotas) + mini interfaz web

Conceptos técnicos:

- **Conexión JDBC:**
    - Drivers (MySQL Connector/J, PostgreSQL Driver).
    - Clases `DriverManager`, `Connection`, `Statement`, `PreparedStatement`, `ResultSet`.
    - Control de transacciones: `commit()`, `rollback()`, `setAutoCommit()`.
- **Spring JDBC Template:**
    - Configuración de `DataSource`.
    - Ejecución de consultas con `queryForObject`, `query`, `update`.
    - RowMappers para mapear ResultSet → objetos.
- **Diseño de consultas SQL:**
    - CRUD básico y consultas JOIN.
    - Procedimientos almacenados y llamadas desde JDBC.
- **Miniinterfaz web con Spring Boot:**
    - **Thymeleaf**: plantillas, formularios, binding de objetos.
    - **Vaadin**: creación de vistas con componentes Java.
    - Controladores con Spring MVC (`@Controller`, `@GetMapping`, `@PostMapping`).

UD 3. Herramientas ORM: Hibernate + Spring Data JPA

Conceptos técnicos:

- **Configuración básica JPA:**
    - `application.properties` para conexión.
    - Entidades con anotaciones: `@Entity`, `@Table`, `@Id`, `@GeneratedValue`.
- **Mapeo de relaciones:**

    - `@OneToOne`, `@OneToMany`, `@ManyToMany`.
    - Estrategias de carga: *lazy* vs *eager*.
    - Estrategias de herencia (`@Inheritance`).
- **Consultas:**
    - JPQL (`@Query`).
    - Derived Queries (findBy, countBy, existsBy…).
    - Native Queries.
- **Spring Data JPA:**
    - Interfaces `JpaRepository`, `CrudRepository`.
    - Paginación y ordenación (`Pageable`, `Sort`).
- **Gestión de transacciones:**
    - `@Transactional`.
    - Aislamiento y propagación.
    - Optimistic vs Pessimistic Locking.

Segunda Evaluación

UD 4. Bases de datos NoSQL: MongoDB

Conceptos técnicos:

- **Conceptos básicos:**
    - Colecciones, documentos, campos.
    - Tipos de datos BSON.
- **Spring Boot + MongoDB:**

    - Dependencia `spring-boot-starter-data-mongodb`.
    - Entidades con `@Document`.
    - Repositorios `MongoRepository`.
    - Consultas con `@Query`.
- **Operaciones CRUD:**
    - Inserción, actualización (`save`), eliminación (`delete`).
    - Consultas por atributos.
    - Índices.
- **Comparativa SQL vs NoSQL:**
    - Flexibilidad de esquema.
    - Escalabilidad horizontal.
    - Uso en Big Data y microservicios.

UD 5. Bases de datos nativas XML, orientadas a objetos y objeto-relacionales

Conceptos técnicos:

- **Bases de datos XML:**
    - Motores: eXistDB, BaseX.
    - Lenguajes de consulta: XPath, XQuery.
    - Validación integrada con XSD.
- **Bases de datos orientadas a objetos:**
    - Concepto de persistencia transparente.
    - Motores como db4o (obsoleto) o ObjectDB.

    - Almacenamiento directo de objetos Java.
- **Bases de datos objeto-relacionales:**
    - Extensión de SQL con tipos complejos.
    - Soporte en PostgreSQL (tipos JSON, arrays, objetos compuestos).
    - Integración con Hibernate/JPA.

UD 6. Programación de componentes de acceso a datos

Conceptos técnicos:

- **Arquitectura de acceso a datos:**
    - Patrones DAO, Repository, Service.
    - Inyección de dependencias con Spring (`@Autowired`).
- **Buenas prácticas:**
    - Separación de capas (MVC).
    - Uso de DTO y mapeadores (MapStruct).
- **Seguridad en datos:**
    - Roles y permisos en Spring Security.
    - Control de inyecciones SQL/NoSQL.
- **Pruebas:**
    - JUnit5 + Mockito.
    - Pruebas de integración con `@DataJpaTest`.
    - TestContainers (Docker para pruebas con BBDD reales).
- **Proyecto:**
    - API REST con Spring Boot + MySQL/MongoDB.

- Endpoints CRUD.
- Documentación con Swagger/OpenAPI.