

# Ejercicio: Sistema de Gestión de Prácticas Académicas

## Descripción General

Desarrollar una aplicación web para la gestión integral de prácticas académicas utilizando Java, MySQL y Eclipse/Spring Tools Suite. El sistema implementará el patrón Modelo-Vista-Controlador (MVC) y gestionará cuatro tipos de usuarios con diferentes niveles de acceso y funcionalidades.

## Objetivos de Aprendizaje

- Implementar el patrón MVC en Java
- Gestionar bases de datos relacionales con MySQL
- Implementar sistemas de autenticación y autorización
- Crear interfaces de usuario intuitivas
- Desarrollar funcionalidades CRUD completas
- Generar reportes y estadísticas
- Aplicar buenas prácticas de programación

## Tecnologías Requeridas

- **Lenguaje:** Java 11 o superior
- **Framework:** Spring Boot + Spring MVC + Spring Security
- **Base de Datos:** MySQL 8.0
- **ORM:** JPA/Hibernate
- **IDE:** Eclipse o Spring Tools Suite
- **Frontend:** Thymeleaf + Bootstrap 5
- **Build Tool:** Maven

## Especificación de Perfiles de Usuario

### 1. Administrador

- **Permisos:** Acceso completo al sistema
- **Funcionalidades:**
  - CRUD completo de todos los usuarios
  - Gestión de cursos y empresas
  - Configuración del sistema
  - Acceso a todas las estadísticas
  - Gestión de criterios de evaluación globales

## 2. Tutor del Curso

- **Permisos:** Gestión de sus cursos asignados
- **Funcionalidades:**
  - Ver alumnos de sus cursos
  - Consultar empresas asociadas a sus alumnos
  - Ver tutores de prácticas de sus alumnos
  - Evaluar alumnos y tutores
  - Generar reportes por curso/empresa/alumno
  - Acceder a estadísticas de sus cursos

## 3. Tutor de Prácticas (Empresa)

- **Permisos:** Gestión de sus alumnos asignados
- **Funcionalidades:**
  - Ver solo sus alumnos asignados
  - Registrar observaciones diarias
  - Evaluar prácticas de sus alumnos
  - Gestionar criterios de evaluación
  - Registrar incidencias
  - Actualizar horarios y actividades

## 4. Alumno

- **Permisos:** Gestión de su información personal
- **Funcionalidades:**
  - Ver y actualizar datos personales
  - Registrar observaciones diarias propias
  - Ver evaluaciones recibidas
  - Consultar horarios y actividades
  - Ver feedback de tutores

# Modelo de Datos

## Entidad: Alumno

- id (Long, PK, Auto-increment)
- nombre (String, 50, NOT NULL)
- apellidos (String, 100, NOT NULL)
- dni (String, 9, UNIQUE, NOT NULL)
- fechaNacimiento (Date, NOT NULL)
- email (String, 100, UNIQUE, NOT NULL)
- telefono (String, 15)
- cursold (Long, FK)
- empresald (Long, FK)

- tutorPracticasId (Long, FK)
- duracionPracticas (Integer) // en días
- horario (String, 200)
- fechalinicio (Date)
- fechaFin (Date)
- activo (Boolean, DEFAULT true)
- fechaCreacion (Timestamp)
- fechaActualizacion (Timestamp)

## **Entidad: TutorPracticas**

- id (Long, PK, Auto-increment)
- nombre (String, 50, NOT NULL)
- apellidos (String, 100, NOT NULL)
- dni (String, 9, UNIQUE, NOT NULL)
- email (String, 100, UNIQUE, NOT NULL)
- telefono (String, 15)
- empresald (Long, FK)
- cargo (String, 100)
- horario (String, 200)
- activo (Boolean, DEFAULT true)
- fechaCreacion (Timestamp)
- fechaActualizacion (Timestamp)

## **Entidad: TutorCurso**

- id (Long, PK, Auto-increment)
- nombre (String, 50, NOT NULL)
- apellidos (String, 100, NOT NULL)
- dni (String, 9, UNIQUE, NOT NULL)
- email (String, 100, UNIQUE, NOT NULL)
- telefono (String, 15)
- especialidad (String, 100)
- activo (Boolean, DEFAULT true)
- fechaCreacion (Timestamp)
- fechaActualizacion (Timestamp)

## **Entidad: Curso**

- id (Long, PK, Auto-increment)
- nombre (String, 100, NOT NULL)
- descripcion (String, 500)
- duracion (Integer) // en horas
- fechalinicio (Date)
- fechaFin (Date)
- tutorCursold (Long, FK)
- activo (Boolean, DEFAULT true)
- fechaCreacion (Timestamp)

## **Entidad: Empresa**

- id (Long, PK, Auto-increment)
- nombre (String, 100, NOT NULL)
- cif (String, 12, UNIQUE, NOT NULL)
- direccion (String, 200)
- telefono (String, 15)
- email (String, 100)
- personaContacto (String, 100)
- sector (String, 50)
- activo (Boolean, DEFAULT true)
- fechaCreacion (Timestamp)

## **Entidad: ObservacionDiaria**

- id (Long, PK, Auto-increment)
- alumnonId (Long, FK)
- fecha (Date, NOT NULL)
- actividades (TEXT)
- explicaciones (TEXT)
- observacionesAlumno (TEXT)
- observacionesTutor (TEXT)
- horasRealizadas (Integer)
- fechaCreacion (Timestamp)

## **Entidad: Incidencia**

- id (Long, PK, Auto-increment)
- alumnonId (Long, FK)
- tutorPracticasId (Long, FK)
- fecha (Date, NOT NULL)
- tipo (ENUM: FALTA, RETRASO, PROBLEMA\_ACTITUD, OTROS)
- descripcion (TEXT, NOT NULL)
- resolucion (TEXT)
- estado (ENUM: ABIERTA, EN\_PROCESO, RESUELTA)
- fechaCreacion (Timestamp)
- fechaResolucion (Timestamp)

## **Entidad: CriterioEvaluacion**

- id (Long, PK, Auto-increment)
- nombre (String, 100, NOT NULL)
- descripcion (TEXT)
- peso (DECIMAL, 5,2) // porcentaje sobre la nota final
- activo (Boolean, DEFAULT true)

## **Entidad: CapacidadEvaluacion**

- id (Long, PK, Auto-increment)
- criteriodId (Long, FK)
- nombre (String, 100, NOT NULL)
- descripcion (TEXT)
- puntuacionMaxima (Integer, DEFAULT 10)

- activo (Boolean, DEFAULT true)

## Entidad: Evaluacion

- id (Long, PK, Auto-increment)
- alumnoId (Long, FK)
- tutorPracticasId (Long, FK)
- capacidadId (Long, FK)
- puntuacion (DECIMAL, 4,2)
- observaciones (TEXT)
- fecha (Date, NOT NULL)
- fechaCreacion (Timestamp)

## Entidad: EvaluacionTutor

- id (Long, PK, Auto-increment)
- tutorPracticasId (Long, FK)
- tutorCursold (Long, FK)
- puntuacion (DECIMAL, 4,2)
- observaciones (TEXT)
- aspectosPositivos (TEXT)
- aspectosMejorar (TEXT)
- fecha (Date, NOT NULL)

## Entidad: Usuario (Para autenticación)

- id (Long, PK, Auto-increment)
- email (String, 100, UNIQUE, NOT NULL)
- password (String, 255, NOT NULL) // encriptada
- rol (ENUM: ADMIN, TUTOR\_CURSO, TUTOR\_PRACTICAS, ALUMNO)
- referenciaId (Long) // ID de la tabla correspondiente según el rol
- activo (Boolean, DEFAULT true)
- ultimoAcceso (Timestamp)
- fechaCreacion (Timestamp)

# Funcionalidades Específicas

## Dashboard Principal (Todos los roles)

- Resumen de estadísticas personalizadas según el rol
- Notificaciones y alertas
- Acceso rápido a funciones principales
- Calendario de actividades

## Gestión de Observaciones Diarias

- Formulario para registro diario de actividades
- Vista calendario con observaciones

- Filtros por fecha, alumno, tutor
- Exportación a PDF/Excel

## Sistema de Evaluación

- Configuración de criterios y capacidades
- Formularios de evaluación intuitivos
- Cálculo automático de notas finales
- Historial de evaluaciones

## Sistema de Incidencias

- Registro de incidencias con clasificación
- Workflow de resolución
- Notificaciones automáticas
- Seguimiento de estados

## Módulo de Estadísticas e Informes

- **Estadísticas Generales:**
  - Tasa de aprobados por curso
  - Notas medias por empresa/tutor
  - Distribución de calificaciones
  - Tiempo medio de realización de prácticas
- **Estadísticas de Empleabilidad:**
  - Porcentaje de alumnos contratados
  - Empresas que más contratan
  - Evolución temporal de contrataciones
  - Correlación notas-contratación
- **Informes Personalizados:**
  - Por curso, empresa, tutor o alumno
  - Comparativas temporales
  - Análisis de tendencias
  - Reportes ejecutivos

## Funcionalidades de Búsqueda y Filtrado

- Búsqueda avanzada por múltiples criterios
- Filtros dinámicos
- Ordenación personalizable
- Exportación de resultados

## Arquitectura del Sistema

## **Capa de Presentación (Views)**

```
src/main/resources/templates/  
└── admin/  
└── tutor-curso/  
└── tutor-practicas/  
└── alumno/  
└── common/  
└── auth/  
└── reports/
```

## **Capa de Controladores**

```
src/main/java/controllers/  
└── AdminController.java  
└── TutorCursoController.java  
└── TutorPracticasController.java  
└── AlumnoController.java  
└── AuthController.java  
└── ReportsController.java  
└── ApiController.java (para AJAX)
```

## **Capa de Servicios**

```
src/main/java/services/  
└── UserService.java  
└── AlumnoService.java  
└── TutorService.java  
└── EvaluacionService.java  
└── EstadisticasService.java  
└── ReportsService.java
```

## **Capa de Repositorios**

```
src/main/java/repositories/  
└── AlumnoRepository.java  
└── TutorPracticasRepository.java  
└── TutorCursoRepository.java  
└── EvaluacionRepository.java  
└── [resto de entidades]Repository.java
```

## **Modelos/Entidades**

```
src/main/java/models/  
└── Alumno.java  
└── TutorPracticas.java  
└── TutorCurso.java  
└── Evaluacion.java  
└── [resto de entidades].java
```

# **Requisitos de Implementación (Voluntarios, no obligatorios)**

## **Seguridad**

- Implementar Spring Security
- Encriptación de contraseñas con BCrypt
- Autorización basada en roles
- Prevención de ataques CSRF/XSS
- Validación de datos de entrada

## **Interfaz de Usuario**

- Responsive design con Bootstrap
- Navegación intuitiva
- Formularios con validación JavaScript
- Feedback visual de acciones
- Accesibilidad básica (WCAG 2.1)

## **Base de Datos**

- Diseño normalizado
- Índices optimizados
- Constraints de integridad
- Triggers para auditoría
- Procedimientos almacenados para estadísticas

## **Testing**

- Tests unitarios para servicios
- Tests de integración para repositorios
- Tests de controladores
- Cobertura mínima del 80%

# Criterios de Evaluación

## Funcionalidad (40%)

- Todas las funcionalidades implementadas correctamente
- Sistema de permisos funcionando
- CRUD completo para todas las entidades
- Cálculos y estadísticas precisos

## Código (30%)

- Aplicación correcta del patrón MVC
- Código limpio y bien documentado
- Manejo adecuado de excepciones
- Uso apropiado de Spring Framework

## Base de Datos (20%)

- Diseño normalizado y eficiente
- Consultas optimizadas
- Integridad referencial
- Manejo de transacciones

## Interfaz de Usuario (10%)

- Diseño intuitivo y funcional
- Responsive design
- Validaciones del lado cliente
- Experiencia de usuario fluida

## Entregables

1. **Código fuente completo** con estructura Maven
2. **Base de datos** con script de creación y datos de prueba
3. **Documentación técnica** incluyendo:
  - Manual de instalación
  - Diagramas de arquitectura
  - Documentación de API
  - Manual de usuario
4. **Video demostración** (5-10 minutos)

## Cronograma Sugerido

- **Semana 1-2:** Análisis y diseño de BD, setup del proyecto
- **Semana 3-4:** Implementación de entidades y repositorios
- **Semana 5-6:** Desarrollo de servicios y controladores
- **Semana 7:** Implementación de vistas y funcionalidades básicas
- **Semana 8:** Sistema de evaluación y estadísticas
- **Semana 9:** Testing, refinamiento y documentación

## **Extensiones Opcionales (Para nota)**

- API REST completa para integración externa
- Notificaciones por email automáticas
- Sistema de backup automático
- Integración con calendario externo
- Chat interno entre tutores y alumnos
- Sistema de archivos adjuntos
- Módulo de planificación de prácticas
- Dashboard con gráficos interactivos (Chart.js)