

Prueba Técnica para Desarrollador Mid Sistema de Registro Escolar

Objetivo

Desarrollar un sistema CRUD para el registro de personas en una escuela, implementando tanto el backend (Java) como el frontend (Angular).

Requisitos Técnicos

Backend

- Java 11 o superior
- Spring Boot 2.7 o superior
- Spring Data JPA
- Base de datos H2 (para desarrollo) / MySQL o PostgreSQL (opcional para producción)
- Maven como gestor de dependencias
- Arquitectura en capas (controladores, servicios, repositorios)
- APIs RESTful con CRUD completo
- Validaciones básicas (campos obligatorios, email válido, teléfono numérico, fechas correctas)
- Manejo de excepciones con respuestas JSON consistentes
- Manejo de logs
- Implementar DTOs con ModelMapper

Opcional (Bonus):

- Paginación en listados
- Autenticación con JWT
- Documentación con Swagger

Frontend

- Angular 13 o superior
- Angular Material o PrimeNG
- TypeScript
- Diseño responsive
- Módulo para gestionar personas (estudiantes, profesores y administrativos)
- Formulario de creación/edición con validaciones
- Listado con paginación y opción de eliminar

Descripción del Proyecto

El sistema debe permitir gestionar la información de los diferentes actores de una escuela (estudiantes, profesores y personal administrativo), así como sus relaciones y actividades.

Modelo de Datos

Entidades y Relaciones

1. **Persona** (Entidad base)

- id_persona (PK)
- nombre
- apellido
- fecha_nacimiento
- email
- telefono

2. **Estudiante** (Hereda de Persona)

- id_persona (PK, FK)
- numero_matricula
- grado

3. **Profesor** (Hereda de Persona)

- id_persona (PK, FK)
- especialidad
- fecha_contratacion

(Opcional - Bonus)

Agregar entidad Administrativo, Curso y Inscripción como relación entre Estudiante y Curso

4. **Administrativo** (Hereda de Persona)

- id_persona (PK, FK)
- cargo
- departamento

5. **Curso**

- id_curso (PK)
- nombre
- descripcion
- creditos
- id_profesor (FK)

6. **Inscripcion** (Relación entre Estudiante y Curso)

- id_inscripcion (PK)
- id_estudiante (FK)
- id_curso (FK)
- fecha_inscripcion

Funcionalidades Requeridas

Backend

1. APIs RESTful para todas las entidades:

- Implementar CRUD completo (GET, POST, PUT, DELETE)
- Implementar filtros básicos

2. Validaciones:

- Email válido
- Teléfono válido (formato numérico)
- Campos obligatorios no nulos
- Fechas válidas (no futuras para nacimiento)
- Relaciones válidas (no se puede inscribir a un curso inexistente)
- Validación de unicidad (email, número de matrícula)

3. Manejo de excepciones:

- Crear excepciones personalizadas
- Implementar controlador global de excepciones
- Respuestas de error consistentes

Opcional (Bonus):

4. Agregar filtros y paginación en listados
5. Implementar autenticación con JWT
6. Casos de pruebas unitarias JUNIT

Frontend

1. Módulos Angular:

- Módulo de personas
- Módulo de estudiantes
- Módulo de profesores
- Módulo de administrativos
- Módulo de cursos
- Módulo de inscripciones

2. Componentes mínimos:

- Lista de cada entidad con paginación
- Formulario de creación/edición

- Diálogo de confirmación para eliminación

Validaciones en formularios:

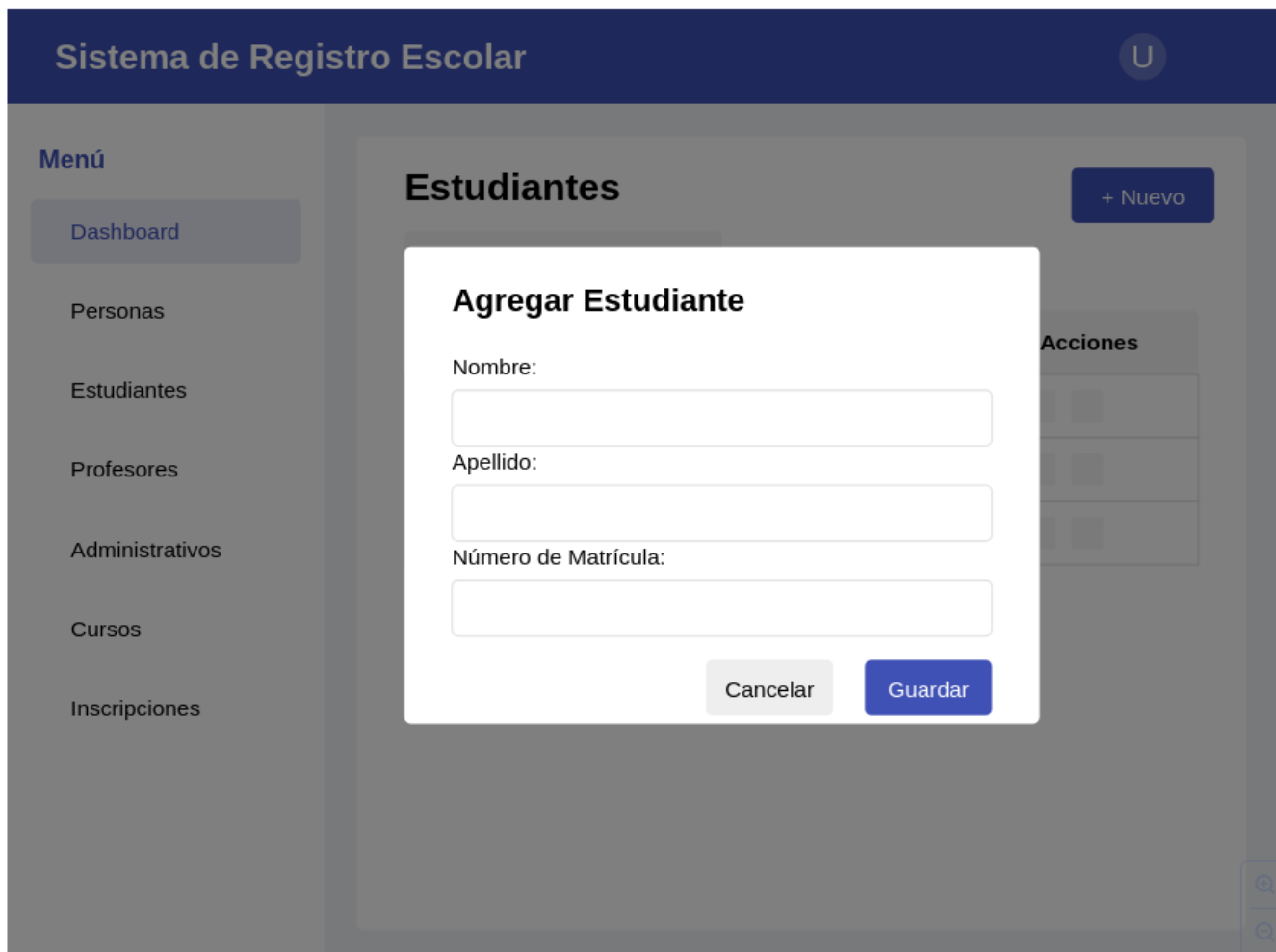
- Validaciones de campos requeridos
- Validación de formato de email
- Validación de formato de teléfono
- Validación de fechas

3. Servicios:

- Implementar servicios para comunicación con el backend
- Manejo de errores HTTP
- Interceptores para token (opcional)

4. Diseño:

- Usar componentes de Angular Material o PrimeNG
- Implementar tema personalizado
- Diseño responsive



Entregables

1. Código fuente:

- Proyecto Java Spring Boot (backend)
- Proyecto Angular (frontend)
- Script SQL para creación de la base de datos

2. Documentación:

- README con instrucciones de instalación y ejecución
- Diagrama de la base de datos
- Manual de usuario básico

Opcional (Bonus):

1. Postman Collection
2. Dockerfile para levantar backend y base de datos
3. Despliegue cloud (AWS, Azure, entre otros).
4. Casos de prueba backend y frontend (JUnit, Mock)

Desarrollo

Se recomienda seguir los siguientes pasos:

1. Diseñar y crear la base de datos
2. Implementar las entidades y repositorios JPA
3. Crear los servicios y controladores REST
4. Implementar el proyecto Angular con las rutas
5. Desarrollar los servicios para consumir las APIs
6. Crear los componentes de interfaz usuario
7. Implementar las validaciones y manejo de errores
8. Realizar pruebas y corregir errores
9. Documentar el código y generar documentación

Tiempo Estimado

- Duración esperada: 4 días

Entrega

El candidato debe entregar el código fuente a través de un repositorio Git (GitHub, GitLab, Bitbucket) y proporcionar instrucciones claras sobre cómo ejecutar el proyecto.