**Documentación de la Prueba Técnica - Sistema de Registro Escolar**

**Descripción General**

Este proyecto corresponde a una prueba técnica para un cargo de Desarrollador Mid. El objetivo fue desarrollar un sistema CRUD completo para la gestión de personas en una institución educativa, incluyendo estudiantes, profesores y administrativos, usando **Java (Spring Boot)** en el backend y **Angular con Angular Material** en el frontend.

**Tecnologías Utilizadas**

**Backend (Java Spring Boot)**

* Java 17
* Spring Boot 3
* Spring Web, Spring Data JPA, Validation
* ModelMapper para conversión DTO-Entidad
* Base de datos: MYSQL
* Maven como gestor de dependencias

**Frontend (Angular)**

* Angular 17
* Angular Material
* TypeScript
* SCSS
* Responsive Design

**Estructura del Backend**

**Entidades:**

* **Persona** (abstracta)
* **Profesor** (hereda de Persona)
* **Estudiante** (hereda de Persona)
* (Bonus: no implementado en esta fase: Administrativo, Curso, Inscripción)

**Paquetes principales:**

* controller: controladores REST para Profesor y Estudiante
* service: servicios de negocio con interfaces y clases implementadas
* repository: interfaces JPA para acceso a datos
* dto: clases DTO con validaciones
* mapper: convertidores DTO <-> entidad usando ModelMapper
* config: configuración general, incluyendo ModelMapper
* exception: manejo global de errores con @ControllerAdvice

**Funcionalidades principales:**

* CRUD completo para Profesor y Estudiante
* Validaciones:
  + Email válido
  + Teléfono numérico
  + Fechas de nacimiento no futuras
* Excepciones personalizadas con respuestas JSON

**Bonus implementado:**

* Uso de ModelMapper
* Arquitectura en capas

**Estructura del Frontend**

**Módulos creados:**

* NavbarComponent (menú de navegación)
* RegistroProfesorComponent, ListarProfesoresComponent
* RegistroEstudianteComponent, ListarEstudiantesComponent

**Características:**

* Formulario reutilizable para crear y editar profesores/estudiantes
* Listado en tabla con Angular Material (mat-table)
* Botones de acción para editar y eliminar
* Validaciones reactivas en los formularios
* Navegación con routerLink
* Estilos y componentes de Angular Material

**Servicios:**

* ProfesorService y EstudianteService para llamadas HTTP al backend

**Diagrama Entidad-Relación (ER)**

**Persona** (abstracta)

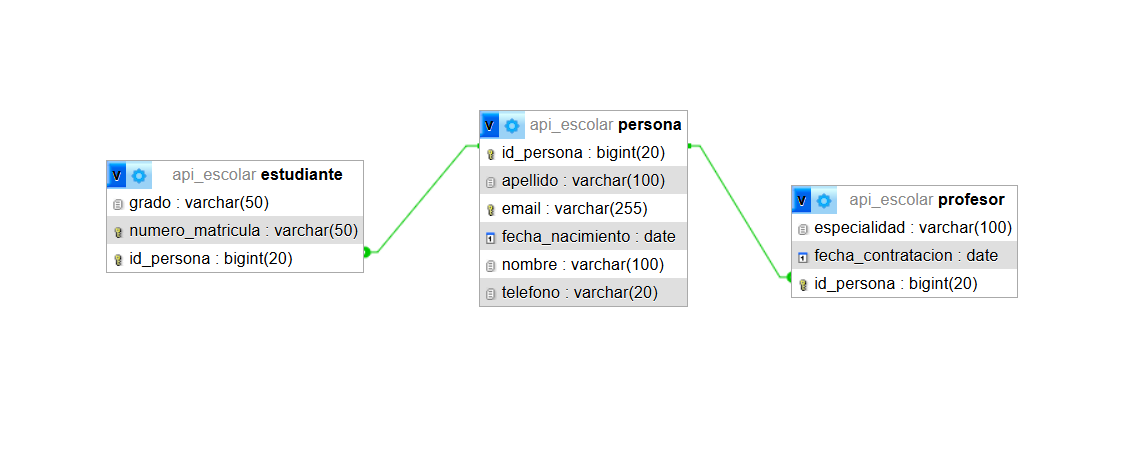
* id\_persona (PK)
* nombre
* apellido
* email
* teléfono
* fecha\_nacimiento

**Profesor**

* especialidad
* fecha\_contratacion

**Estudiante**

* numero\_matricula
* grado



**Instrucciones para Ejecutar**

**Backend**

cd backend

mvn spring-boot:run

La aplicación corre en: http://localhost:8080

**Frontend**

cd frontend

npm install

ng serve

La aplicación corre en: http://localhost:4200

**Endpoints REST**

**Profesores**

* GET /api/profesores
* GET /api/profesores/{id}
* POST /api/profesores
* PUT /api/profesores/{id}
* DELETE /api/profesores/{id}

**Estudiantes**

* GET /api/estudiantes
* GET /api/estudiantes/{id}
* POST /api/estudiantes
* PUT /api/estudiantes/{id}
* DELETE /api/estudiantes/{id}

**Validaciones Realizadas**

**Backend (DTOs)**

* @NotBlank, @Email, @Size, @PastOrPresent, @Pattern

**Frontend**

* Validación de campos requeridos
* ­­Validación de email y teléfono
* Validación de fechas

**Manejo de Errores**

* GlobalExceptionHandler con @ControllerAdvice
* Errores formateados como JSON

{

"timestamp": "2025-07-07T10:22:11",

"status": 400,

"errors": [

"El nombre es obligatorio",

"El email debe ser válido"

]

}

**Entregables**

* Proyecto completo en dos carpetas:
  + /backend (Java Spring Boot)
  + /frontend (Angular)
* Script SQL de referencia para estructura (solo si se usara MySQL)

**Implementado**

* Swagger :

Para visualizar el dashboard de swagger:

http://localhost:8080/swagger-ui/index.html#/

* Docker :  
    
  Para levantar el servicio con postgres: ejecutar el siguiente comando

docker-compose up -d

**Tiempo aproximado de desarrollo**

* **Total**: 3 días (intercalado con otras tareas)
* Backend: 1.5 días
* Frontend: 1.5 días

**Manual de Usuario Básico**

1. Entrar a http://localhost:4200
2. Usar el menú para navegar:
   * Registrar Profesor o Estudiante
   * Ver listado
   * Editar desde el listado
   * Eliminar
3. Validaciones en formularios evitarán errores comunes
4. Al guardar se muestra un mensaje de éxito

**Futuras Mejoras**

* Agregar autenticación (JWT)
* Agregar paginación en listados
* Implementar inscripciones y cursos
* Usar PostgreSQL en ambiente productivo
* Agregar pruebas unitarias

**Repositorio**

Proyecto realizado con enfoque en buenas prácticas, arquitectura limpia, y uso correcto de Angular Material y Spring Boot.