

# Validaciones

## Autenticación (Auth)

### 1. **CreateAuthDto** (src/auth/dto/create-auth.dto.ts):

- Email: debe ser un string y un email válido
- Password: string, mínimo 6 caracteres, máximo 50 caracteres
- FullName: string, mínimo 1 carácter, máximo 100 caracteres

### 1. **LoginUserDto** (src/auth/dto/Login-user.dto.ts):

- Email: debe ser un string y un email válido
- Password: string, mínimo 6 caracteres, máximo 50 caracteres

### 1. **JWT Strategy** (src/auth/strategies/jwt.strategy.ts):

- Valida que el token sea válido
- Valida que el usuario exista
- Valida que el usuario esté activo
- Valida que el token no esté en la lista negra

## Estudiantes

### 1. **CreateStudentDto** (src/students/dto/create-student.dto.ts):

- name: string requerido
- age: número opcional, debe ser positivo
- email: string, debe ser un email válido
- subjects: array de strings
- gender: debe ser uno de: 'Male', 'Female', 'Other'
- nickname: string opcional
- grades: array opcional de calificaciones

## Cursos

### 1. **CreateCourseDto** (src/courses/dto/create-course.dto.ts):

- name: string requerido
- description: string requerido
- code: string requerido
- teacherId: debe ser un UUID válido
- status: enum opcional (CourseStatus)
- startDate: debe ser una fecha válida
- endDate: debe ser una fecha válida

## Entregas (Submissions)

### 1. **CreateSubmissionDto** (src/submissions/dto/create-submission.dto.ts):

- courseId: debe ser un UUID válido
- studentId: debe ser un UUID válido
- fileUrl: debe ser una URL válida
- comments: string opcional

### 1. **GradeSubmissionDto** (src/submissions/dto/grade-submission.dto.ts):

- grade: número entre 0 y 5
- comments: string opcional, no vacío

### **Paginación**

#### 1. **PaginationDto** (src/commons/dto/pagination.dto.ts):

- limit: número opcional, debe ser positivo
- offset: número opcional, mínimo 0

### **Validaciones Globales**

#### 1. **ValidationPipe Global** (src/main.ts):

- whitelist: elimina propiedades no definidas en los DTOs
- forbidNonWhitelisted: lanza error si se envían propiedades no definidas
- transform: transforma automáticamente los tipos de datos

### **Roles y Permisos**

#### 1. **UserRoleGuard** (src/auth/guards/user-role/user-role.guard.ts):

- Valida que el usuario tenga los roles requeridos para acceder a los endpoints
- Roles válidos definidos en ValidRoles enum:
  - admin
  - super-user
  - teacher
  - student