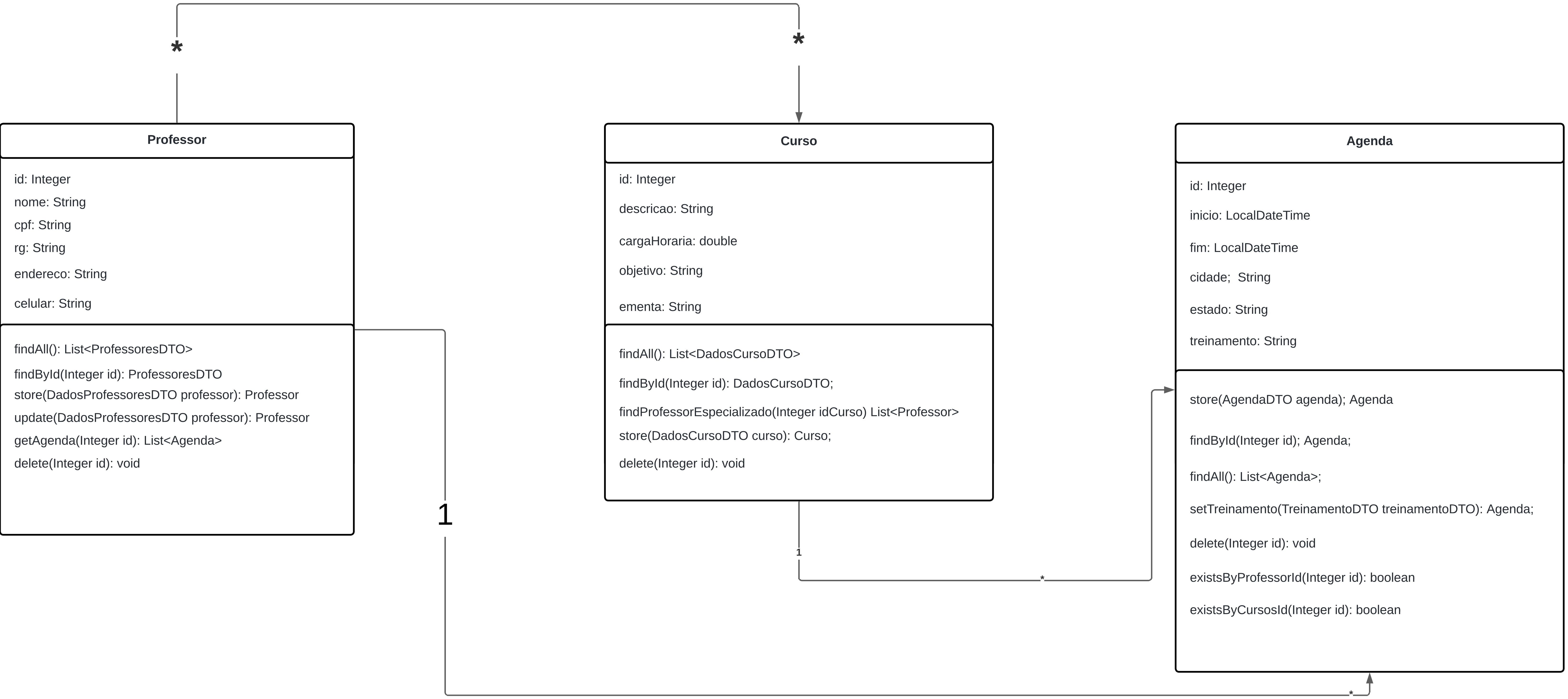
https://github.com/flima21/AC2_ArqWeb.git



Professor

id: Integer
nome: String
cpf: String
rg: String
endereco: String
celular: String

findAll(): List<ProfessoresDTO>
findById(Integer id): ProfessoresDTO
store(DadosProfessoresDTO professor): Professor
update(DadosProfessoresDTO professor): Professor
getAgenda(Integer id): List<Agenda>
delete(Integer id): void

Curso

id: Integer
descricao: String
cargaHoraria: double
objetivo: String
ementa: String

findAll(): List<DadosCursoDTO>
findById(Integer id): DadosCursoDTO;
findProfessorEspecializado(Integer idCurso) List<Professor>
store(DadosCursoDTO curso): Curso;
delete(Integer id): void

Agenda

id: Integer
inicio: LocalDateTime
fim: LocalDateTime
cidade; String
estado: String
treinamento: String

store(AgendaDTO agenda); Agenda
findById(Integer id); Agenda;
findAll(): List<Agenda>;
setTreinamento(TreinamentoDTO treinamentoDTO): Agenda;
delete(Integer id): void
existsByProfessorId(Integer id): boolean
existsByCursosId(Integer id): boolean

```java
package com.example.ac2.models;

import java.util.ArrayList;
import java.util.List;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.annotation.Generated;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.FetchType;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.ManyToMany;
import jakarta.persistence.OneToMany;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Professor {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    @Column(nullable = false)
    private String nome;

    @Column(nullable = false)
    private String cpf;

    @Column(nullable = false)
    private String rg;

    @Column(nullable = false)
    private String endereco;

    @Column(nullable = false)
    private String celular;

    @JsonIgnore
    @OneToMany(mappedBy = "professores")
    private List<Agenda> agenda = new ArrayList<>();;

    // Alterar para many to many
    @JsonIgnore
    @ManyToMany(mappedBy = "professores")
    private List<Curso> cursos = new ArrayList<>();

}
```

```java
package com.example.ac2.models;

import java.util.ArrayList;
import java.util.List;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.FetchType;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.JoinTable;
import jakarta.persistence.ManyToMany;
import jakarta.persistence.OneToMany;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Curso {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    @Column(nullable = false)
    private String descricao;

    @Column(nullable = false)
    private double cargaHoraria;

    @Column(nullable = false)
    private String objetivo;

    @Column(nullable = false)
    private String ementa;

    @JsonIgnore
    @OneToMany(mappedBy = "cursos")
    private List<Agenda> agenda;

    @JsonIgnore
    @ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    @JoinTable(name = "cursos_professores", joinColumns = {@JoinColumn(name = "professores_id")},inverseJoinColumns = {@JoinColumn(name = "cursos_id")})
    private List<Professor> professores = new ArrayList<>();

}
```

```java
package com.example.ac2.models;

import java.time.LocalDateTime;
import java.util.Date;


import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.OneToOne;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Agenda {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    @Column(nullable = false)
    private LocalDateTime inicio;

    @Column(nullable = false)
    private LocalDateTime fim;

    @Column(nullable = false,length = 80)
    private String cidade;

    @Column(length = 80)
    private String estado;

    @Column(nullable = false,length = 80)
    private String cep;

    @Column(length = 80)
    private String treinamento;

    @ManyToOne
    @JoinColumn(name = "curso_id")
    private Curso cursos;

    @ManyToOne
    @JoinColumn(name = "professor_id")
    private Professor professores;
}
```

```java
package com.example.ac2.repository;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.example.ac2.models.Agenda;

@Repository
public interface AgendaRepository extends JpaRepository<Agenda, Integer> {
    @Query("SELECT ag FROM Agenda ag LEFT JOIN FETCH ag.professores WHERE ag.professores.id = :id")
    List<Agenda> findAgendaByIdProfessoresFetchProfessores(@Param("id") Integer professores_id);

    @Query("SELECT ag FROM Agenda ag inner JOIN FETCH ag.professores WHERE (ag.inicio >= :dataInicio and ag.fim <= :dataFinal) and (ag.professores.id = :id)")
    Agenda findAgendaFetchDataInicioProfessores(@Param("dataInicio") LocalDateTime dataInicio, @Param("dataFinal") LocalDateTime dataFinal, @Param("id") Integer id);

    boolean existsByProfessoresId(Integer id);
    boolean existsByCursosId(Integer id);

}
```

```
package com.example.ac2.repository;

import java.util.List;
import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.example.ac2.dtos.DadosProfessoresDTO;
import com.example.ac2.models.Curso;
import com.example.ac2.models.Professor;

@Repository
public interface CursoRepository extends JpaRepository<Curso, Integer> {
    @Query("select p from Curso cc left join cc.professores p where cc.id = :id")
    List<Professor> findCursoByIdFetchProfessores(@Param("id") Integer id);

}
```

```
package com.example.ac2.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import com.example.ac2.models.Professor;

@Repository
public interface ProfessorRepository extends JpaRepository<Professor, Integer> {
    @Query("SELECT p FROM Professor p RIGHT JOIN FETCH p.agenda WHERE p.id = :id")
    List<Professor> findProfessoresFetchAgenda(@Param("id") Integer id);

}
```

```java
package com.example.ac2.services;

import java.util.List;

import com.example.ac2.dtos.AgendaDTO;
import com.example.ac2.dtos.TreinamentoDTO;
import com.example.ac2.models.Agenda;

public interface AgendaService {
  Agenda store(AgendaDTO agenda);
  Agenda findById(Integer id);
  List<Agenda> findAll();

  Agenda setTreinamento(TreinamentoDTO treinamentoDTO);

  List<Agenda> findAgendaByProfessor(int professorId);

  void delete(int agendaId);

  boolean existsByProfessoresId(Integer professor_id);
  boolean existsByCursosId(Integer cursos_id);
}
```

```java
package com.example.ac2.services;

import java.util.List;

import com.example.ac2.dtos.CursosDTO;
import com.example.ac2.dtos.DadosCursosDTO;
import com.example.ac2.dtos.DadosProfessoresDTO;
import com.example.ac2.models.Curso;
import com.example.ac2.models.Professor;

public interface CursoService {
  List<DadosCursosDTO> findAll();
  DadosCursosDTO findById(Integer id);
  List<Professor> findProfessorEspecializado(Integer idCurso);

  Curso store(DadosCursosDTO curso);
  Curso update(DadosCursosDTO professor);

  void delete(Integer id);

}
```

```java
package com.example.ac2.services;

import com.example.ac2.dtos.AgendaDTO;
import com.example.ac2.dtos.DadosProfessoresDTO;
import com.example.ac2.dtos.ProfessoresDTO;
import com.example.ac2.models.Agenda;
import com.example.ac2.models.Professor;
import java.util.List;

public interface ProfessorService {
  List<ProfessoresDTO> findAll();

  ProfessoresDTO findById(Integer id);
  Professor store(DadosProfessoresDTO professor);
  Professor update(DadosProfessoresDTO professor);

  List<Agenda> getAgenda(Integer id);

  void delete(Integer id);
}
```

```java
package com.example.ac2.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PatchMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import com.example.ac2.dtos.AgendaDTO;
import com.example.ac2.dtos.DadosAgendaDTO;
import com.example.ac2.dtos.TreinamentoDTO;
import com.example.ac2.models.Agenda;
import com.example.ac2.services.AgendaService;

import jakarta.transaction.Transactional;
import lombok.AllArgsConstructor;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

@RestController
@RequestMapping("/agenda")
@AllArgsConstructor
public class AgendaController {
  private AgendaService agendaImpl;

  @GetMapping
  @ResponseStatus(HttpStatus.OK)
  public List<Agenda> findAll() {
    return this.agendaImpl.findAll();
  }

  @GetMapping("/{id}")
  @ResponseStatus(HttpStatus.OK)
  public Agenda findById(@PathVariable("id") Integer id) {
    return this.agendaImpl.findById(id);
  }

  @PostMapping()
  @ResponseStatus(HttpStatus.CREATED)
  public Agenda registerAgenda(@RequestBody AgendaDTO agenda) {
    return this.agendaImpl.store(agenda);
  }

  @DeleteMapping("/{id}")
  @ResponseStatus(HttpStatus.GONE)
  public void delete(@PathVariable Integer id) {
    this.agendaImpl.delete(id);
  }

  @PatchMapping()
  @ResponseStatus(HttpStatus.OK)
  public Agenda setTreinamento(@RequestBody TreinamentoDTO treinamentoDTO) {
    return this.agendaImpl.setTreinamento(treinamentoDTO);
  }
}
```

```
package com.example.ac2.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import com.example.ac2.dtos.CursosDTO;
import com.example.ac2.dtos.DadosCursosDTO;
import com.example.ac2.dtos.DadosProfessoresDTO;
import com.example.ac2.models.Curso;
import com.example.ac2.models.Professor;
import com.example.ac2.services.CursoService;

import lombok.AllArgsConstructor;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

@RestController
@RequestMapping("/curso")
@AllArgsConstructor
public class CursoController {
  private CursoService cursoImpl;

  @GetMapping()
  @ResponseStatus(HttpStatus.OK)
  public List<DadosCursosDTO> listAllCursos() {
    return this.cursoImpl.findAll();
  }

  @GetMapping("/{id}")
  @ResponseStatus(HttpStatus.OK)
  public DadosCursosDTO listaPorId(@PathVariable("id") Integer id) {
    return this.cursoImpl.findById(id);
  }

  @GetMapping("/curso-professor/{id}")
  @ResponseStatus(HttpStatus.OK)
  public List<Professor> getEspecializacao(@PathVariable Integer id) {
    return this.cursoImpl.findProfessorEspecializado(id);
  }

  @PostMapping()
  @ResponseStatus(HttpStatus.CREATED)
  public Curso registerCourse(@RequestBody DadosCursosDTO curso) {
    return this.cursoImpl.store(curso);
  }

  @PutMapping()
  @ResponseStatus(HttpStatus.OK)
  public Curso update(@RequestBody DadosCursosDTO curso) {
    return this.cursoImpl.update(curso);
  }

  @DeleteMapping("/{id}")
  @ResponseStatus(HttpStatus.OK)
  public void delete(@PathVariable Integer id) {
    this.cursoImpl.delete(id);
  }



}
```

```java
package com.example.ac2.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import com.example.ac2.dtos.AgendaDTO;
import com.example.ac2.dtos.DadosProfessoresDTO;
import com.example.ac2.dtos.ProfessoresDTO;
import com.example.ac2.interfaces.ProfessorImpl;
import com.example.ac2.models.Agenda;
import com.example.ac2.models.Professor;
import com.example.ac2.services.ProfessorService;

import jakarta.transaction.Transactional;
import lombok.AllArgsConstructor;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

@RestController
@RequestMapping("/professor")
@AllArgsConstructor
public class ProfessorController {
  private ProfessorService professorImpl;

  @GetMapping()
  @ResponseStatus(HttpStatus.OK)
  public List<ProfessoresDTO> listAllProfessores() {
    return this.professorImpl.findAll();
  }

  @GetMapping("/agenda/{id}")
  @ResponseStatus(HttpStatus.OK)
  public List<Agenda> getAgenda(@PathVariable Integer id) {
    return this.professorImpl.getAgenda(id);
  }

  @GetMapping("/{id}")
  @ResponseStatus(HttpStatus.OK)
  public ProfessoresDTO professorById(@PathVariable Integer id) {
    return this.professorImpl.findById(id);
  }

  @PostMapping()
  @ResponseStatus(HttpStatus.CREATED)
  public Professor registerProfessor(@RequestBody DadosProfessoresDTO professor) {
    return this.professorImpl.store(professor);
  }

  @PutMapping()
  @ResponseStatus(HttpStatus.OK)
  public Professor update(@RequestBody DadosProfessoresDTO professor) {
    return this.professorImpl.update(professor);
  }

  @DeleteMapping("/{id}")
  @ResponseStatus(HttpStatus.OK)
  public void delete(@PathVariable Integer id) {
    this.professorImpl.delete(id);
  }
}
```