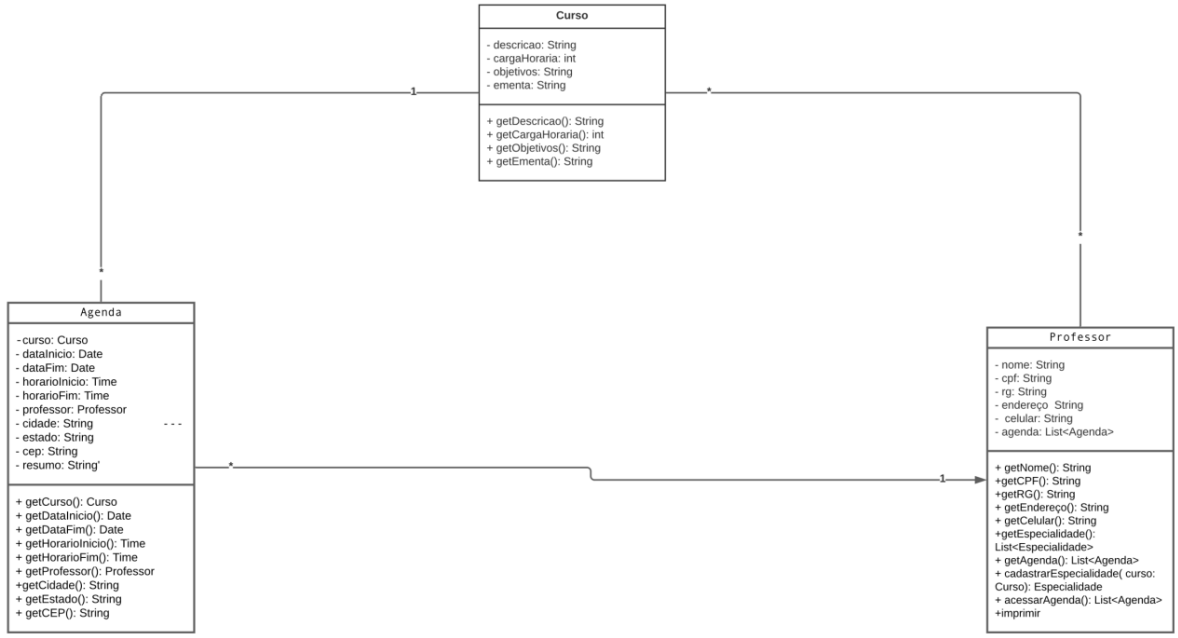
Diagrama de Classes



Integrantes:

Leonardo Pelegrin - 223538

Luis Felipe - 223438

Wesley - 223658

Maria Eduarda - 222901

Classe AgendaController.Java

```
package com.example.ac2f.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import com.example.ac2f.dtos.AgendaDTO;
import com.example.ac2f.services.AgendaServiceImpl;



import java.util.List;

@RestController
@RequestMapping("/api/agenda")
public class AgendaController {

 @Autowired
 AgendaServiceImpl service;

 @PostMapping
 public void inserir(@RequestBody AgendaDTO agenda) {
  service.inserir(agenda);
 }

 @GetMapping
 public List<AgendaDTO> obterTodos() {
```

```
  return service.obterTodos();
 }
}
```

Classe CursoController.Java

```
package com.example.ac2f.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.ac2f.dtos.CursoDTO;
import com.example.ac2f.services.CursoServiceImpl;

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

@RestController
@RequestMapping("/api/curso")
public class CursoController {

 @Autowired
 CursoServiceImpl service;

 @PostMapping()
 public void inserir(@RequestBody CursoDTO curso) {
  service.inserir(curso);
 }
```

```
 @GetMapping()
 public List<CursoDTO> listarTodas() {
  return service.obterTodos();
 }
}
```

Classe ProfessorController.Java

```
package com.example.ac2f.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.ac2f.dtos.ProfessorDTO;
import com.example.ac2f.services.ProfessorServiceImpl;

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

@RestController
@RequestMapping("/api/professor")
public class ProfessorController {

 @Autowired
 ProfessorServiceImpl service;

 @PostMapping()
 public void inserir(@RequestBody ProfessorDTO professor) {
  service.inserir(professor);
 }
```

```
 @GetMapping()
 public List<ProfessorDTO> listarTodas() {
  return service.obterTodos();
 }
}
```

Classe Agenda.Java

```
package com.example.ac2f.models;

import java.time.LocalDate;
import java.time.LocalTime;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import jakarta.persistence.GenerationType;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Entity
@Table(name = "tb_agenda")
```

```
public class Agenda {
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 private long id;

 @Column(nullable = false)
 LocalDate dataInicial;

 @Column(nullable = false)
 LocalDate dataFinal;

 @Column(nullable = false)
 LocalTime horarioInicial;

 @Column(nullable = false)
 LocalTime horarioFinal;

 @ManyToOne
 @JoinColumn(name = "professor_id")
 private Professor professor;

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

 @Column(nullable = false)
 String cidade;

 @Column(nullable = false)
 String estado;

 @Column(nullable = false)
```

```
String cep;

@Column(nullable = false)
String resumo;

public long getId() {
 return id;
}

public void setId(long id) {
 this.id = id;
}

public LocalDate getDataInicial() {
 return dataInicial;
}

public void setDataInicial(LocalDate dataInicial) {
 this.dataInicial = dataInicial;
}

public LocalDate getDataFinal() {
 return dataFinal;
}

public void setDataFinal(LocalDate dataFinal) {
 this.dataFinal = dataFinal;
}

public LocalTime getHorarioInicial() {
 return horarioInicial;
}
```

```
public void setHorarioInicial(LocalTime horarioInicial) {
 this.horarioInicial = horarioInicial;
}

public LocalTime getHorarioFinal() {
 return horarioFinal;
}

public void setHorarioFinal(LocalTime horarioFinal) {
 this.horarioFinal = horarioFinal;
}

public Professor getProfessor() {
 return professor;
}

public void setProfessor(Professor professor) {
 this.professor = professor;
}

public Curso getCurso() {
 return curso;
}

public void setCurso(Curso curso) {
 this.curso = curso;
}

public String getCidade() {
 return cidade;
}
```

```
 public void setCidade(String cidade) {
  this.cidade = cidade;
 }

 public String getEstado() {
  return estado;
 }

 public void setEstado(String estado) {
  this.estado = estado;
 }

 public String getCep() {
  return cep;
 }

 public void setCep(String cep) {
  this.cep = cep;
 }

 public String getResumo() {
  return resumo;
 }

 public void setResumo(String resumo) {
  this.resumo = resumo;
 }
}
```

Classe Curso.Java

```
package com.example.ac2f.models;

import java.util.List;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.ManyToMany;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Entity
public class Curso {
 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 private long id;
```

```
@ManyToMany(mappedBy = "cursos")
List<Professor> professores;

@ManyToMany(mappedBy = "curso")
List<Agenda> agendas;

@Column(nullable = false)
String descricao;

@Column(nullable = false)
String cargaHoraria;

@Column(nullable = false)
String objetivo;

@Column(nullable = false)
String ementa;

public long getId() {
 return id;
}

public void setId(long id) {
 this.id = id;
}

public List<Professor> getProfessores() {
 return professores;
}

public void setProfessores(List<Professor> professores) {
```

```
 this.professores = professores;
}

public List<Agenda> getAgendas() {
 return agendas;
}

public void setAgendas(List<Agenda> agendas) {
 this.agendas = agendas;
}

public String getDescricao() {
 return descricao;
}

public void setDescricao(String descricao) {
 this.descricao = descricao;
}

public String getCargaHoraria() {
 return cargaHoraria;
}

public void setCargaHoraria(String cargaHoraria) {
 this.cargaHoraria = cargaHoraria;
}

public String getObjetivo() {
 return objetivo;
}

public void setObjetivo(String objetivo) {
```

```
    this.objetivo = objetivo;
  }

  public String getEmenta() {
    return ementa;
  }

  public void setEmenta(String ementa) {
    this.ementa = ementa;
  }
}
```

Classe Professor.Java

```java
package com.example.ac2f.models;

import java.util.List;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinTable;
import jakarta.persistence.ManyToMany;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import jakarta.persistence.JoinColumn;


@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Entity
@Table(name = "tb_professor")
public class Professor {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private long id;

  @ManyToMany
  @JoinTable(name = "tb_professor_curso",
  joinColumns = @JoinColumn(name = "professor_id"),
  inverseJoinColumns = @JoinColumn(name = "curso_id"))
  @Column(nullable = false)
  List<Curso> cursos;

  @ManyToMany(mappedBy = "professor")
  List<Agenda> agendas;
```

```java
@Column(nullable = false)
String nome;

@Column(nullable = false)
String cpf;

@Column(nullable = false)
String endereco;

@Column(nullable = false)
String telefone;

public long getId() {
  return id;
}

public void setId(long id) {
  this.id = id;
}

public List<Agenda> getAgendas() {
  return agendas;
}

public void setAgendas(List<Agenda> agendas) {
  this.agendas = agendas;
}

public List<Curso> getCursos() {
  return cursos;
}

public void setCursos(List<Curso> cursos) {
  this.cursos = cursos;
}

public String getNome() {
  return nome;
}

public void setNome(String nome) {
  this.nome = nome;
}

public String getCpf() {
  return cpf;
}
```

```java
  public void setCpf(String cpf) {
    this.cpf = cpf;
  }

  public String getEndereco() {
    return endereco;
  }

  public void setEndereco(String endereco) {
    this.endereco = endereco;
  }

  public String getTelefone() {
    return telefone;
  }

  public void setTelefone(String telefone) {
    this.telefone = telefone;
  }
}
```

Classe AgendaRepository.Java

```java
package com.example.ac2f.repositories;

import com.example.ac2f.models.Agenda;
import com.example.ac2f.models.Professor;

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 java.time.LocalDate;
import java.time.LocalTime;
import java.util.List;
import java.util.Optional;

@Repository
public interface AgendaRepository extends JpaRepository<Agenda, Long> {

    // verifica data (disponibilidade) do professor

    @Query("select a from Agenda a where a.dataInicial <= :data and
a.dataFinal >= :data and a.professor.id = :id")
    List<Agenda> agendasProfessorPorData(@Param("id") Long id,
            @Param("data") LocalDate data);

    Optional<Professor> findByDataInicialBeforeEqual(LocalTime
localtime);

}
```

Classe CursoRepository.Java

```java
package com.example.ac2f.repositories;

import com.example.ac2f.models.Curso;
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 java.util.Optional;

public interface CursoRepository extends JpaRepository<Curso, Long> {

    @Query("select c from Curso c left join fetch c.professores p where
c.id = :id ")
    Optional<Curso> findCursoProfessorFetchCursos(@Param("id") Long id);

}
```

Classe ProfessorRepository.Java

```java
package com.example.ac2f.repositories;

import com.example.ac2f.models.Agenda;
import com.example.ac2f.models.Professor;

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;

@Repository
public interface ProfessorRepository extends JpaRepository<Professor,
Long> {

    // professor que detem a especialização do curso
    @Query("select a from Agenda a where a.professor.id = :id")
    List<Agenda> agendasDoProfessor(@Param("id") Long id);

}
```

Classe AgendaService.Java

```java
package com.example.ac2f.services;

import com.example.ac2f.dtos.AgendaDTO;
import java.util.List;
import java.time.LocalDate;
import java.time.LocalTime;

public interface AgendaService {
  void inserir(AgendaDTO agendaDto);
  List<AgendaDTO> obterTodos();
  AgendaDTO obterPorId(Long id);
  List<AgendaDTO> obterPorData(LocalDate localtime, long idProf);
  void editar(AgendaDTO agendaDto);
}
```

Classe AgendaServiceImpl.Java

```java
package com.example.ac2f.services;

import java.time.LocalDate;
import java.time.LocalTime;
import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.ac2f.dtos.AgendaDTO;
import com.example.ac2f.models.Agenda;
import com.example.ac2f.models.Curso;
import com.example.ac2f.models.Professor;
import com.example.ac2f.repositories.AgendaRepository;
import com.example.ac2f.repositories.CursoRepository;
import com.example.ac2f.repositories.ProfessorRepository;


@Service
public class AgendaServiceImpl implements AgendaService {
  @Autowired
  private AgendaRepository agendaRepository;

  @Autowired
  private ProfessorRepository professorRepository;

  @Autowired
  private CursoRepository cursoRepository;

  public AgendaServiceImpl(AgendaRepository agendaRepository,
ProfessorRepository professorRepository,
      CursoRepository cursoRepository) {
    this.agendaRepository = agendaRepository;
    this.professorRepository = professorRepository;
    this.cursoRepository = cursoRepository;
```

```
  }

  @Override
  public void inserir(AgendaDTO agendaDto) {
    Agenda agenda = new Agenda();
    Professor professor =
professorRepository.findById(agendaDto.getProfessorId()).orElseThrow();
    Curso curso =
cursoRepository.findById(agendaDto.getCursoId()).orElseThrow();

    agenda.setDataFinal(agendaDto.getDataFinal());
    agenda.setDataInicial(agendaDto.getDataInicial());
    agenda.setHorarioFinal(agendaDto.getHorarioFinal());
    agenda.setHorarioInicial(agendaDto.getHorarioInicial());
    agenda.setCidade(agendaDto.getCidade());
    agenda.setEstado(agendaDto.getEstado());
    agenda.setCep(agendaDto.getCep());
    agenda.setResumo(agenda.getResumo());
    agenda.setCurso(curso);
    agenda.setProfessor(professor);

    agendaRepository.save(agenda);
  }

  @Override
  public List<AgendaDTO> obterTodos() {
    List<AgendaDTO> agendas = agendaRepository.findAll().stream().map(a -
> {
      return AgendaDTO.builder()
          .id(a.getId())
          .dataInicial(a.getDataInicial())
          .dataFinal(a.getDataFinal())
          .horarioInicial(a.getHorarioInicial())
          .horarioFinal(a.getHorarioFinal())
          .professorId(null)
          .cursoId(null)
          .cidade(a.getCidade())
          .cep(a.getCep())
          .estado(a.getEstado())
          .resumo(a.getResumo())
          .build();
    }).collect(Collectors.toList());

    return agendas;
  }

  @Override
  public AgendaDTO obterPorId(Long id) {
    Agenda a = agendaRepository.findById(id).orElseThrow();
```

```
    return AgendaDTO.builder()
        .id(a.getId())
        .dataInicial(a.getDataInicial())
        .dataFinal(a.getDataFinal())
        .horarioInicial(a.getHorarioInicial())
        .horarioFinal(a.getHorarioFinal())
        .professorId(a.getProfessor() == null ? null
:a.getProfessor().getId())
        .cursoId(a.getCurso() == null ? null :a.getCurso().getId())
        .cidade(a.getCidade())
        .cep(a.getCep())
        .estado(a.getEstado())
        .resumo(a.getResumo())
        .build();
  }
  @Override
  public List<AgendaDTO> obterPorData(LocalDate localtime, long idProf) {
      List<AgendaDTO> agendas =
agendaRepository.agendasProfessorPorData(idProf,localtime).stream().map(a
-> {
          return AgendaDTO.builder()
                  .id(a.getId())
                  .dataInicial(a.getDataInicial())
                  .dataFinal(a.getDataFinal())
                  .horarioInicial(a.getHorarioInicial())
                  .horarioFinal(a.getHorarioFinal())
                  .professorId(a.getProfessor() == null ? null :
a.getProfessor().getId())
                  .cursoId(a.getCurso() == null ? null :
a.getCurso().getId())
                  .cidade(a.getCidade())
                  .cep(a.getCep())
                  .estado(a.getEstado())
                  .resumo(a.getResumo())
                  .build();
      }).collect(Collectors.toList());

      return agendas;
  }

  @Override
  public void editar(AgendaDTO agendaDto) {
    Agenda agenda = new Agenda();
    Professor professor =
professorRepository.findById(agendaDto.getProfessorId()).orElseThrow();
    Curso curso =
cursoRepository.findById(agendaDto.getCursoId()).orElseThrow();
```

```
    agenda.setDataFinal(agendaDto.getDataFinal());
    agenda.setDataInicial(agendaDto.getDataInicial());
    agenda.setHorarioFinal(agendaDto.getHorarioFinal());
    agenda.setHorarioInicial(agendaDto.getHorarioInicial());
    agenda.setCidade(agendaDto.getCidade());
    agenda.setEstado(agendaDto.getEstado());
    agenda.setCep(agendaDto.getCep());
    agenda.setResumo(agenda.getResumo());
    agenda.setCurso(curso);
    agenda.setProfessor(professor);
    agenda.setId(agendaDto.getId());

    agendaRepository.save(agenda);
  }
}
```

Classe CursoService.Java

```java
package com.example.ac2f.services;


import java.util.List;
import com.example.ac2f.dtos.CursoDTO;


public interface CursoService {
  void inserir(CursoDTO cursoDto);
  List<CursoDTO> obterTodos();
  CursoDTO obterPorId(Long id);
}
```

Classe CursoServiceImpl.Java

```java
package com.example.ac2f.services;

import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.ac2f.dtos.CursoDTO;
import com.example.ac2f.models.Curso;
import com.example.ac2f.repositories.CursoRepository;
import com.example.ac2f.repositories.ProfessorRepository;


@Service
public class CursoServiceImpl implements CursoService {
  @Autowired
  private CursoRepository cursoRepository;
```

```java
  @Autowired
  ProfessorRepository professorRepository;

  public CursoServiceImpl(CursoRepository cursoRepository) {
    this.cursoRepository = cursoRepository;
  }

  @Override
  public void inserir(CursoDTO cursoDto) {
    Curso curso = new Curso();
    curso.setDescricao(cursoDto.getDescricao());
    curso.setCargaHoraria(cursoDto.getCargaHoraria());
    curso.setObjetivo(cursoDto.getObjetivo());
    curso.setEmenta(cursoDto.getEmenta());

    cursoRepository.save(curso);
  }

  @Override
  public List<CursoDTO> obterTodos() {
    List<CursoDTO> cursos = cursoRepository.findAll().stream().map(c -> {
      return CursoDTO.builder()
          .id(c.getId())
          .descricao(c.getDescricao())
          .cargaHoraria(c.getCargaHoraria())
          .objetivo(c.getObjetivo())
          .ementa(c.getEmenta())
          .professoresId(c.getProfessores() == null ? null
              : c.getProfessores().stream().map(p ->
p.getId()).collect(Collectors.toList()))
          .agendasId(
              c.getAgendas() == null ? null :
c.getAgendas().stream().map(a -> a.getId()).collect(Collectors.toList()))
          .build();
    }).collect(Collectors.toList());
    return cursos;
  }

  @Override
  public CursoDTO obterPorId(Long id) {
    Curso c = cursoRepository.findById(id).orElseThrow();

    return CursoDTO.builder()
        .id(c.getId())
        .descricao(c.getDescricao())
        .cargaHoraria(c.getCargaHoraria())
        .objetivo(c.getObjetivo())
        .ementa(c.getEmenta())
        .professoresId(c.getProfessores() == null ? null
```

```
            : c.getProfessores().stream().map(p ->
p.getId()).collect(Collectors.toList()))
        .agendasId(
            c.getAgendas() == null ? null : c.getAgendas().stream().map(a
-> a.getId()).collect(Collectors.toList()))
        .build();
  }

}
```

Classe ProfessorService.Java

```
package com.example.ac2f.services;

import java.util.List;

import com.example.ac2f.dtos.ProfessorDTO;


public interface ProfessorService {
  void inserir(ProfessorDTO dto);
  List<ProfessorDTO> obterTodos();
  ProfessorDTO getById(Long id);
}
```

```
package com.example.ac2f.services;


import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.ac2f.dtos.ProfessorDTO;
import com.example.ac2f.models.Curso;
import com.example.ac2f.models.Professor;
import com.example.ac2f.repositories.CursoRepository;
import com.example.ac2f.repositories.ProfessorRepository;



@Service
public class ProfessorServiceImpl implements ProfessorService {

  @Autowired
  private ProfessorRepository professorRepository;
```

```
  @Autowired
  CursoRepository cursoRepository;

  public ProfessorServiceImpl(ProfessorRepository professorRepository) {
    this.professorRepository = professorRepository;
  }

  @Override
  public void inserir(ProfessorDTO dto) {
    Professor professor = new Professor();
    List<Curso> cursos = dto.getCursosId().stream().map( c ->
cursoRepository.findById(c).orElseThrow()).collect(Collectors.toList());
    professor.setNome(dto.getNome());
    professor.setCpf(dto.getCpf());
    professor.setEndereco(dto.getEndereco());
    professor.setTelefone(dto.getTelefone());
    professorRepository.save(professor);
    professor.setCursos(cursos);
    professorRepository.save(professor);
  }

  @Override
  public List<ProfessorDTO> obterTodos() {
    List<ProfessorDTO> professor =
professorRepository.findAll().stream().map(p -> {
      return ProfessorDTO.builder()
          .id(p.getId())
          .nome(p.getNome())
          .cpf(p.getCpf())
          .endereco(p.getEndereco())
          .telefone(p.getTelefone())
          .cursosId(p.getCursos() == null ? null :
p.getCursos().stream().map(c -> c.getId()).collect(Collectors.toList()))
          .agendaId(p.getAgendas() == null ? null :
p.getAgendas().stream().map(a -> a.getId()).collect(Collectors.toList()))
          .build();
    }).collect(Collectors.toList());

    return professor;
  }

  @Override
  public ProfessorDTO getById(Long id) {
    Professor p = professorRepository.findById(id).orElseThrow();

    return ProfessorDTO.builder()
        .id(p.getId())
        .nome(p.getNome())
        .cpf(p.getCpf())
```

```
        .endereco(p.getEndereco())
        .telefone(p.getTelefone())
        .cursosId(p.getCursos() == null ? null :
p.getCursos().stream().map(c -> c.getId()).collect(Collectors.toList()))
        .agendaId(p.getAgendas() == null ? null :
p.getAgendas().stream().map(a -> a.getId()).collect(Collectors.toList()))
        .build();
  }
}
```

Link do Github:

https://github.com/LeonardoPelegrin/AC2F-Arquitetura