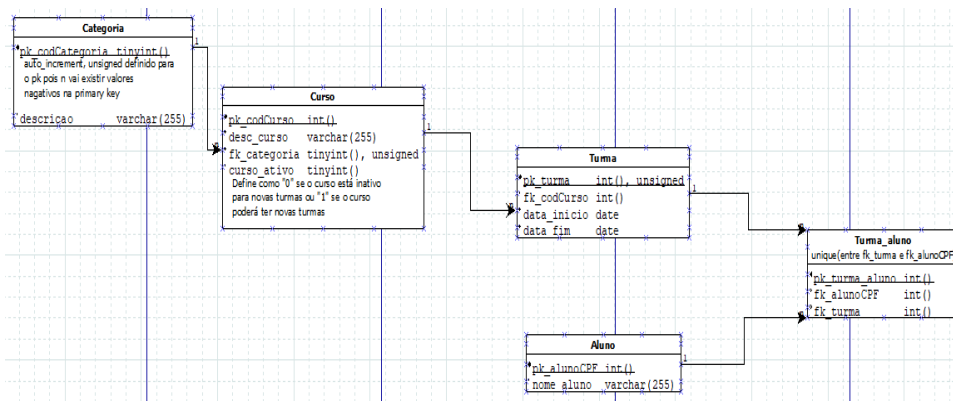


1- Modelagem do banco de Dados



- Pensei em uma modelagem diferente da que me foi passada, pois na minha visão vejo que “Curso” seria uma entidade e para poder ter um histórico das informações seria melhor somente alterar o seu status e não excluí-lo toda vez do banco e perder informações que podem ser julgadas como importantes, sendo assim foi criado as tabelas subsequentes onde vão ficar as informações dos alunos e turmas existentes, podendo somente ser criado novas turmas e se matricular na mesma se o curso estiver como status “ativo”

2- Criação da tela de cadastro de cursos

A interface de usuário para o cadastro de cursos é exibida no navegador em `localhost:8080/curso/cadastro`. Ela contém os seguintes elementos:

- Um menu suspenso rotulado "Selecione a Categoria".
- Um campo de texto rotulado "Preencha o nome do curso".
- Dois botões: "Cadastrar" e "Procure pelo Curso".
- Uma seção de cabeçalho com o texto "Curso Status Qtd Turmas".

```

<body>
  <form th:object="${curso}" id="id_desc" action="#"
        th:actions="@{/curso/cadastro}" method="post">

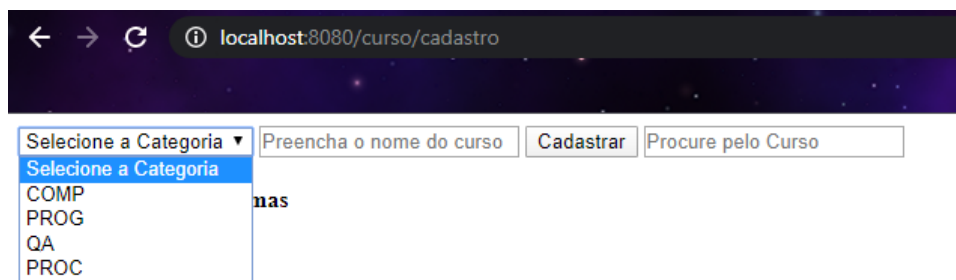
    <select th:field="${categoria}" id="listCategorias"></select>
    <input type="text" th:field="${descricao}" placeholder="Preencha o nome do curso" required="required">

    <button type="submit">Cadastrar</button>
    <input type="text" name="filtrar" id="search" placeholder="Procure pelo Curso">
    <input type="hidden" th:field="${ativo}">

  </form>
  <table id="id_Lista_curso">
    <thead>
      <tr>
        <th>Curso</th>
        <th>Status</th>
        <th>Qtd Turmas</th>
      </tr>
    </thead>
    <tbody id="lista_curso_id">
      <tr>
        <td id="id_teste"></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </tbody>
  </table>
</body>

```

- Função que traz as categorias em que os cursos podem ser cadastrados



```

<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Cadastro de Cursos</title>
<script th:src="@{/js/jquery-3.4.1.min.js}" type="text/javascript"></script>
<link rel="stylesheet" href="src/main/resources/static/css/tr.css" type="text/css" />
<script type="text/javascript">
  $(document).ready(function() {
    $('#listCategorias').find('option').remove();
    $.ajax({
      url : "../rest-categoria", // getContextPath()???
      type: "get",
      success: function(response) {
        $('#listCategorias').append($('', {
          value: 0,
          text: 'Selecione a Categoria'
        }));
        var d = response._embedded.categorias;
        for (item in d) {
          id = d[item].id;
          desc = d[item].categoria;
          $('#listCategorias').append($('

```

- Função que cria uma tabela e vai adicionando os cursos

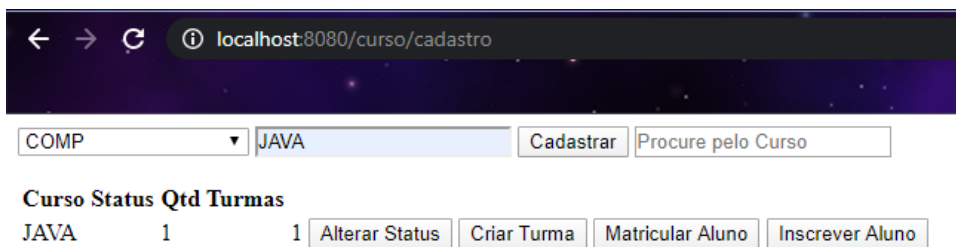
```

$('#listCategorias').change(function(v) {
    $('#lista_curso_id tr').remove();
    $.ajax({
        url: "../curso/lista?categoria=" + $('#listCategorias').val(), // getContextPath()???
        type: "get",
        success: function(response) {
            var c = response;

            //var x = c.filter(c => c.response._embedded.agCursos.categoria == 1);

            for (item in c) {
                console.log(c);
                //alert("estou aqui");
                descCurso = c[item].descricao;
                id = c[item].id;
                at = c[item].ativo;
                qt = c[item].quantidadeTurmas;
                $('#lista_curso_id').append("<tr value=" + id + "><td> " + descCurso + "</td><td></td><td> " + at + "</td> " +
                    "<td> " + qt + "</td>" +
                    "<td><button onclick='alterarStatus(" + id + ")' type='submit'>Alterar Status</button></td> " +
                    "<td><button onclick='redirecionar(" + id + ")' type='submit'>Criar Turma</button></td> " +
                    "<td><button onclick='redirecionarAluno(" + id + ")' type='submit'>Matricular Aluno</button></td> " +
                    "<td><button onclick='redirecionarAlunoTurma(" + id + ")' type='submit'>Inscrever Aluno</button></td>" +
                    "</tr>");
            }
        },
        error: function(e) {
            // ERROR
        }
    });
});

```



- Controller

```

@Controller
@RequestMapping("/curso")
public class CursoController {

    @Autowired
    private CursoRepository cursoRepo;

    @Autowired
    private TurmaRepository turmaRepository;

    @GetMapping("/cadastro")
    public String exibirFormulario(Model model) {
        model.addAttribute("curso", new Curso());
        return "form-cad-curso";
    }

    @GetMapping("/lista")
    public @ResponseBody List<Curso> listarCursos(@RequestParam("categoria") Categoria categoria) {
        List<Curso> cursos = new ArrayList<>();
        cursoRepo.findAll(QCurso.curso.categoria.eq(categoria)).forEach(cursos::add);

        for (Curso curso : cursos) {
            curso.setQuantidadeTurmas(new Long(turmaRepository.count(QTurma.turma.curso.eq(curso))).intValue());
        }

        return cursos;
    }

    @PostMapping("/cadastro")
    public String cadastrarCurso(
        @ModelAttribute("curso") Curso curso,
        Model model) {
        cursoRepo.save(curso);
    }
}

```

- Alteração do status para ativar ou inativar o curso

COMP Preencha o nome do curso Cadastrar Procure pelo Curso

Curso	Status	Qtd Turmas
JAVA	0	1

Alterar Status Criar Turma Matricular Aluno Inscrever Aluno

- Controller

```
@GetMapping("/status")
public String StatusCurso(
    @RequestParam("curso") Curso curso) {

    curso.setAtivo(curso.getAtivo().equals(1) ? 0 : 1);
    cursoRepo.save(curso);

    return "redirect:/curso/cadastro";
}
```

- Função que redireciona para a criação de turma

```
<script>
function alterarStatus(id) {
    window.location.href = "http://localhost:8080/curso/status?curso=" + id;
}
</script>
```

3- Criação da tela de cadastrar turma pelo id do curso

localhost:8080/turma/cadastro?curso=1

Data de Inicio: dd/mm/aaaa

Data de Fim: dd/mm/aaaa

Cadastrar

- Existem validações para que as turmas não possam ser criadas com datas inválidas, podemos criar várias turmas de um mesmo curso mas não podemos criar turmas dentro do mesmo período

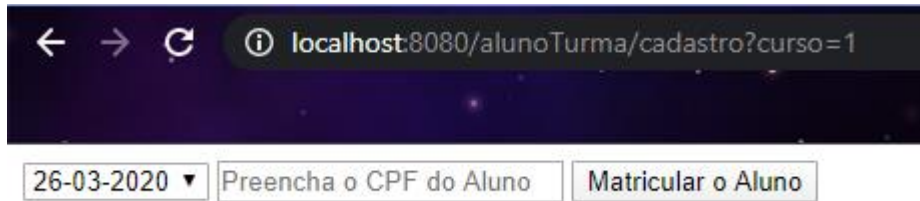
4- Matricular alunos

localhost:8080/aluno/cadastro?curso=1

Preencha o CPF do Aluno Preencha o nome do Aluno Cadastrar Aluno

- Aluno é matriculado e para essa matricula temos o “CPF” dele como uma primary key, para poupar o espaço do banco criando uma nova coluna só para criar um id

5- Inscrever Aluno



The screenshot shows a web browser window with the address bar displaying the URL `localhost:8080/alunoTurma/cadastro?curso=1`. Below the browser window, there is a form with three elements: a date dropdown menu showing `26-03-2020`, a text input field with the placeholder text `Preencha o CPF do Aluno`, and a button labeled `Matricular o Aluno`.

- Selecionamos uma turma que está sendo trazida no ComboBox usando o “id” do curso para trazer somente as datas que estão cadastradas naquele curso