**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

Sistema de gestão BOX QRG

Caio Augustinho do Nascimento

Belo Horizonte

Julho/2023

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Arquitetura da Solução 3

2. Padrão Arquitetural 4

3. C4 model - Diagrama de Contexto 4 Estrutura Base do Front End 5

4. Modelo Relacional ou Projeto de Banco de Dados NoSQL 6

5. Plano de Testes 16

6. Apropriação de Horas no Projeto 17

7. Código da Aplicação 19

8. Avaliação Retrospectiva 19

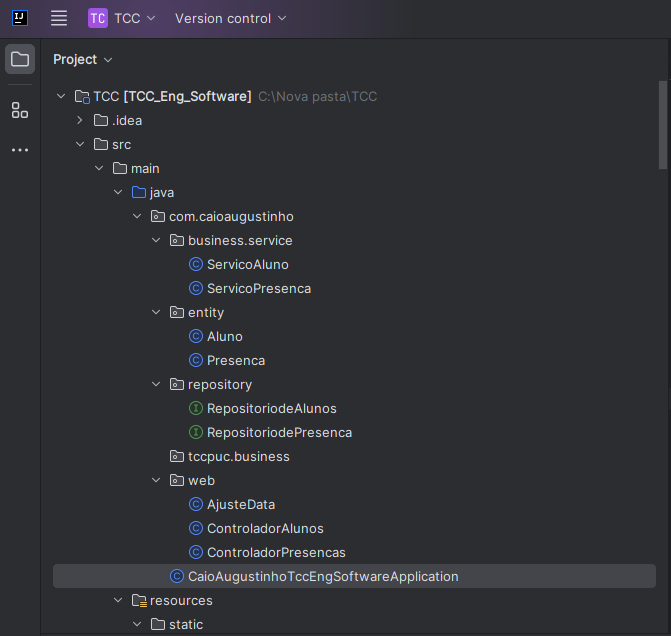
9. Objetivos Estimados 20

10. Objetivos Alcançados 20

11 Lições aprendidas 20

## Arquitetura da Solução

## Padrão Arquitetural

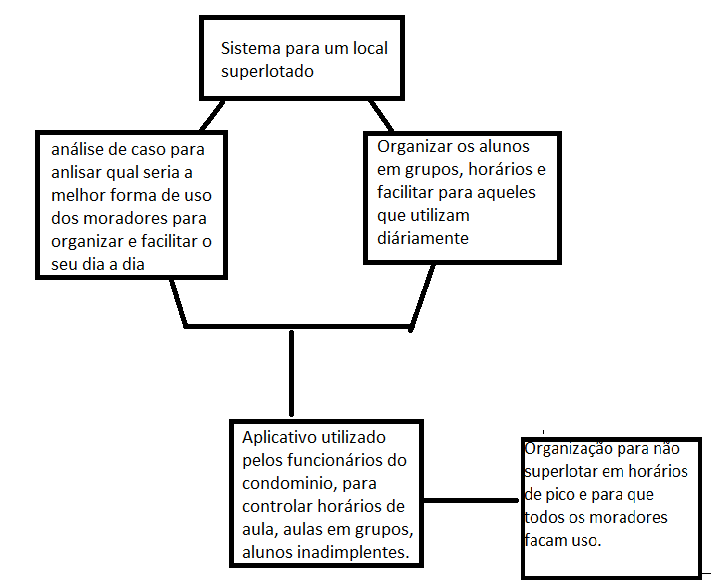


## C4 model - Diagrama de Contexto

A primeiro momento foi necessário entender qual era a maior necessidade do condiminio, de forma que o sistema fosse criado para de fato atender sua necessidade.

Após entender o que acontecia, foi criado o software para que os funcionários controlassem horários de aula, aula em grupo, alunos inadimplentes, e fácil liberação das aulas.

Após a criação, o sistema atendeu de fato a organização, facilitando o uso pelos funcionários que são trocados trimestralmente.



## Frameworks de Trabalho

### ITIL

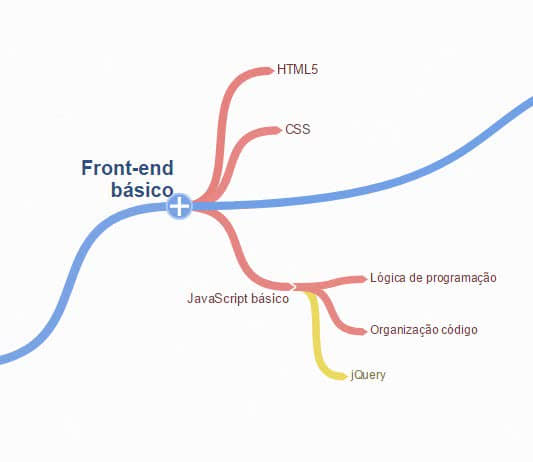
* estratégia de serviços;
* transição de serviços;
* desenho de serviços;
* operação de serviços;
* melhoria contínua de serviços.

### PMBOK

* iniciação;
* planejamento;
* execução;
* monitoramento;
* controle e encerramento.

Para a realização deste trabalho, Também foi utilizado um desktop com Windows 10, microssoft Office Word 7, IntelliJ IDEA Community Edition 2023.2, Google Chrome, ipad e banco de dados.

## Estrutura Base do Front End



PÁGINA INICIAL

<!DOCTYPE html>  
<html lang="pt\_br" xmlns:th="">  
 <head>  
 <meta charset="utf-8"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>  
 <link rel="stylesheet" href="/webjars/bootstrap/css/bootstrap.min.css">  
 <link rel="stylesheet" href="css/puc.css"/>  
 <script src="https://kit.fontawesome.com/59d868f5b6.js" crossorigin="anonymous"></script>   
   
 <title>TCC Caio Augustinho</title>  
   
 </head>  
  
 <body>  
  
 <div class="container-fluid">  
 <h1 class="text-center"><u>BoxQRG</u></h1>  
  
 <style>  
 body {  
 background-color: #999999  
 }  
 </style>  
 <div class="navegadorSuperior">  
 <a class="active"><i class="fas fa-home"></i> <u>Início</u></a>  
 <div class="navegadorSuperior-right">  
 <a href="/areaDoEmpregado"><i class="fas fa-user-cog"></i> Área do Empregado</a>  
 </a>  
 </div>  
 </div>  
 <br><br>  
 <footer>  
 <p class="text-muted"></p>  
 </footer>  
 </div>  
 </body>  
</html>

CADASTRO DO ALUNO

<!DOCTYPE html>  
<html lang="pt\_br" xmlns:th="http://www.thymeleaf.org">  
  
 <head>  
 <meta charset="utf-8"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>  
 <link rel="stylesheet" href="/webjars/bootstrap/css/bootstrap.min.css">  
 <link rel="stylesheet" href="css/puc.css"/>  
 <script src="https://kit.fontawesome.com/59d868f5b6.js" crossorigin="anonymous"></script>  
 <title>Cadastro de Aluno</title>  
 </head>  
 <body>  
 <div class="container-fluid">  
 <h1 class="text-center"><u>BoxQRG </u></h1>  
 <div class="navegadorSuperior">  
 <a class="active"><i class="fas fa-user-cog"></i> Área do Empregado: Cadastro de Aluno</a>  
 <div class="navegadorSuperior-right">  
 <a class="navegadorSuperior-right" href="/"><i class="fas fa-home"></i> Retornar ao Início</a>  
 </div>  
 </div>  
   
 <br>  
 <form action="#" th:action="@{/cadastrarAluno}" method="post"   
 th:object="${aluno}">  
 <div class="form-group">  
 <label for="nome">Nome</label>  
 <input type="text" class="form-control" th:field="\*{nome}" placeholder="Nome">  
 </div>  
 <div class="form-group">  
 <label for="nome">Matrícula</label>  
 <input type="number" class="form-control" th:field="\*{matricula}" placeholder="Matrícula">  
 </div>  
 <div class="form-group">  
 <label for="nome">CPF</label>  
 <input type="text" class="form-control" th:field="\*{cpf}" placeholder="000.000.000-00">  
 </div>  
 <div class="form-group">  
 <label for="nome">Endereço</label>  
 <input type="text" class="form-control" th:field="\*{endereco}" placeholder="Endereço">  
 </div>   
 <button class="btn btn-success" type="submit"> Enviar</button></p>  
 </form>  
 <footer>  
 <p class="text-muted"></p>  
 </footer>  
 </div>  
 </body>  
</html>

LISTA DE ALUNOS

<!DOCTYPE html>  
<html lang="pt\_br" xmlns:th="http://www.thymeleaf.org">  
 <head>  
 <meta charset="utf-8"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>  
 <link rel="stylesheet" href="/webjars/bootstrap/css/bootstrap.min.css">  
 <link rel="stylesheet" href="css/puc.css"/>  
 <script src="https://kit.fontawesome.com/59d868f5b6.js" crossorigin="anonymous"></script>   
 <title>TCC - Caio Augustinho</title>  
 </head>  
 <body>  
 <div class="container-fluid">  
 <h1 class="text-center"><u>BoxQRG</u></h1>  
 <div class="navegadorSuperior">  
 <a class="active"><i class="fas fa-user-cog"></i> <u>Área do Empregado(a) : Lista de Alunos</u></a>  
 <div class="navegadorSuperior-right">  
 <a href="/FormulariodeCadastroAluno"><i class="fas fa-user-edit"></i> Cadastrar Novo Aluno(a)</a>  
 <a href="/"><i class="fas fa-home"></i> Retornar ao Início</a>  
 </div>  
 </div>  
   
 <br>  
 <table class="table table-striped">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Nome</th>  
 <th>CPF</th>  
 <th>Endereco</th>  
 <th>Ações</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="aluno: ${listadeAlunos}">  
 <td th:text="${aluno.id}">ID</td>  
 <td th:text="${aluno.nome}">Nome</td>  
 <td th:text="${aluno.cpf}">CPF</td>  
 <td th:text="${aluno.endereco}">Endereco</td>  
 <td><a th:href="@{/listarPresencasAluno(id=${aluno.id})}"> Visualizar Estatisticas de Presença</a></td>  
 </tr>  
 </tbody>  
 </table>  
 <footer>  
 <p class="text-muted"></p>  
 </footer>  
 </div>   
 </body>  
</html>

LISTA PRESENÇA DE ALUNOS

<!DOCTYPE html>  
<html lang="pt\_br" xmlns:th="http://www.thymeleaf.org">  
 <head>  
 <meta charset="utf-8"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>  
 <link rel="stylesheet" href="/webjars/bootstrap/css/bootstrap.min.css">  
 <link rel="stylesheet" href="css/puc.css"/>  
 <title>Lista de Presencas</title>  
   
 <script src="/webjars/jquery/jquery.min.js"></script>  
 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>   
 <script src="https://kit.fontawesome.com/59d868f5b6.js" crossorigin="anonymous"></script>   
 </head>  
 <body>  
 <div class="container-fluid">  
 <h1 class="text-center"><u>BoxQRG</u></h1>  
 <div class="navegadorSuperior">  
 <a class="active"><i class="fas fa-user-cog"></i></a>   
 <a class="active"><u>Área do Empregado(a) - Lista de Presencas do(a) Aluno(a):</u></a>  
 <a class="active" th:text="${alunoX.nome}"><u><b></b></u></a>  
 <a class="active"><i class="fas fa-user"></i></a>  
 <div class="navegadorSuperior-right">  
 <a href="/listarAlunos"><i class="fas fa-users"></i> Listar Alunos</a>  
 <a href="/"><i class="fas fa-home"></i> Retornar ao Início</a>  
 </div>  
 </div>  
 <br>   
 <div class="row">  
  
   
 <div class="col-10">   
 <h3><b><u>Estatísticas da Presença</b></u></h3>  
 <div >  
 <div id="chart\_div" style="height: 500px;"></div>  
 <!-- <div id="piechart" style="width: 900px; height: 500px;"></div> -->  
 <div id="piechart" style="height: 500px;"></div>  
   
 </div>   
 <script th:inline="javascript">  
 var real\_data = /\*[[${chartData}]]\*/'noValue';  
   
 $(document).ready(function() {  
 google.charts.load('current', {  
 packages : [ 'corechart', 'bar' ]  
 });  
 google.charts.setOnLoadCallback(drawColumnChart);  
 google.charts.setOnLoadCallback(drawPieChart);  
 });  
   
 function drawColumnChart() {  
 var data = new google.visualization.DataTable();  
 data.addColumn('string', 'Year');  
 data.addColumn('number', 'Presenças');  
 Object.keys(real\_data).forEach(function(key) {  
 data.addRow([ key, real\_data[key] ]);  
 });  
 var options = {  
 title : 'Histórico Anual de Presenças do Aluno',  
 hAxis : {  
 title : 'Anos',  
 },  
 vAxis : {  
 title : 'Presenças',  
 minValue: 0  
 }  
 };  
 var chart = new google.visualization.ColumnChart(document  
 .getElementById('chart\_div'));  
 chart.draw(data, options);  
 }  
   
 function drawPieChart() {  
 var data = new google.visualization.DataTable();  
 data.addColumn('string', 'Year');  
 data.addColumn('number', 'Views');  
 Object.keys(real\_data).forEach(function(key) {  
 data.addRow([ key, real\_data[key] ]);  
 });  
 var options = {  
 title : 'Distribuição de Presenças do Aluno por Ano'  
 };  
 var chart = new google.visualization.PieChart(document  
 .getElementById('piechart'));  
 chart.draw(data, options);  
 }  
 </script>  
   
 </div>  
 <div class="col-2">  
 <h3><b><u>Tabela de Presenças</u></b></h3>  
 <table class="table table-striped table-sm">  
 <thead>  
 <tr>  
 <th class="text-center">Data da Presença</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="alunoX: ${alunoX.getListaPresencas()}">  
 <td class="text-center" th:text="${#strings.toUpperCase(#dates.format(alunoX.dataPresenca, 'dd.MMM.yyyy HH:mm'))}">Data da Presença</td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
 </div>  
 <footer>  
 <p class="text-muted"></p>  
 </footer>  
 </div>  
 </body>  
</html>

MENU ÁREA DE EMPREGADOS

<!DOCTYPE html>  
<html lang="pt\_br" xmlns:th="http://www.thymeleaf.org">  
 <head>  
 <meta charset="utf-8"/>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>  
 <link rel="stylesheet" href="/webjars/bootstrap/css/bootstrap.min.css">  
 <link rel="stylesheet" href="css/puc.css"/>  
 <script src="https://kit.fontawesome.com/59d868f5b6.js" crossorigin="anonymous"></script>  
 <title>TCC - Caio Augustinho</title>  
 </head>  
   
 <body>  
 <div class="container-fluid">  
 <h1 class="text-center"><u>BoxQRG</u></h1>  
 <div class="navegadorSuperior">  
 <a class="active"><i class="fas fa-user-cog"></i> <u>Área do Empregado(a)</u></a>  
 <div class="navegadorSuperior-right">  
 <a href="/listarAlunos"><i class="fas fa-users"></i> Visualizar Relatório de Alunos</a>   
 <a href="/FormulariodeCadastroAluno"><i class="fas fa-user-edit"></i> Cadastrar Novo Aluno(a)</a>  
 <a href="/"><i class="fas fa-home"></i> Retornar ao Início</a>  
 </div>  
 </div>  
   
 <br>  
   
  
 <footer>  
 <p class="text-muted"></p>  
 </footer>  
 </div>  
 </body>  
</html>

## Modelo Relacional

package com.caioaugustinho.business.service;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.Random;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import com.caioaugustinho.entity.Aluno;  
import com.caioaugustinho.entity.Presenca;  
import com.caioaugustinho.repository.RepositoriodeAlunos;  
import com.caioaugustinho.repository.RepositoriodePresenca;  
import com.caioaugustinho.web.AjusteData;  
@Service  
public class ServicoPresenca {  
  
  
 private RepositoriodePresenca repositorioDePresenca;  
 private RepositoriodeAlunos repositoriodeAlunos;  
  
 @Autowired  
 public ServicoPresenca(RepositoriodePresenca repositorioDePresenca, RepositoriodeAlunos repositorioDeAlunos) {  
 this.repositorioDePresenca = repositorioDePresenca;  
 this.repositoriodeAlunos = repositorioDeAlunos;  
 popularPresencasIniciais();  
 }  
  
 public void cadastrarPresencaNoBanco(Aluno aluno) {  
 Presenca presencaX = new Presenca(aluno, AjusteData.*createDateFromDateString*("31-12-2020"));  
 aluno.addPresenca(presencaX);  
 this.repositorioDePresenca.save(presencaX);  
 }  
  
 public List<Presenca> recuperarListadePresencadoBanco() {  
 return this.repositorioDePresenca.findAll();  
 }  
   
 public Map<String, Integer> recuperarEstatisticaPresenca(Aluno aluno) {  
 Map<String, Integer> map = new HashMap<>();  
   
 List<Presenca> listaPresencas = aluno.getListaPresencas();  
   
 for (Presenca presenca : listaPresencas) {  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.setTime(presenca.getDataPresenca());  
 String ano = String.*valueOf*(calendar.get(Calendar.*YEAR*));  
 if (map.containsKey(ano)) {  
 Integer quantidade = map.get(ano);  
 map.put(ano, quantidade+1);  
 } else {  
 map.put(ano, 1);  
 }  
 }  
   
 return map;  
 }  
  
  
 public void popularPresencasIniciais() {  
 List<Presenca> presencas = new ArrayList<>();  
 int quantidadeAlunos = 10;  
 int limiteDia = 28;  
 int limiteMes = 12;  
 int inicioAno = 2019;  
 int fimAno = 2023;  
 Random random = new Random();  
 for (int idAluno = 1; idAluno <= quantidadeAlunos; idAluno++) {  
 Aluno aluno = this.repositoriodeAlunos.findById(idAluno).get();  
 for (int ano = inicioAno; ano <= fimAno; ano++) {  
 for (int mes = 1; mes <= limiteMes; mes++) {  
 for (int dia = 1; dia <= limiteDia; dia++) {  
 if (random.nextBoolean()) {  
 presencas.add(new Presenca(aluno, AjusteData  
 .*createDateFromDateString*(String.*format*("%d-%d-%d %d:%d", dia, mes, ano, random.nextInt(24), random.nextInt(60)))));  
 }  
 }  
 }  
 }  
 }  
 this.repositorioDePresenca.saveAll(presencas);   
 }  
}

package com.caioaugustinho.business.service;  
  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import com.caioaugustinho.entity.Aluno;  
import com.caioaugustinho.repository.RepositoriodeAlunos;  
  
@Service  
public class ServicoAluno {  
  
 private RepositoriodeAlunos repositoriodeAlunos;  
  
 @Autowired  
 public ServicoAluno(RepositoriodeAlunos repositoriodeAlunos) {  
 this.repositoriodeAlunos = repositoriodeAlunos;  
 cadastrarAlunosnoBancoInicial();  
 }  
  
 public void cadastrarAlunoNoBanco(Aluno aluno) {  
 this.repositoriodeAlunos.save(aluno);  
 }  
  
 public Aluno recuperarAlunodoBanco(Integer id) {  
 Aluno aluno = this.repositoriodeAlunos.findById(id).get();   
 return aluno;  
 }  
  
 public List<Aluno> recuperarListadeAlunosdoBanco() {  
 List<Aluno> ListaAlunosRecuperados = this.repositoriodeAlunos.findAll();   
 //Impressão de Aluno Recuperados para necessidade de Debug  
 //ListaAlunosRecuperados.forEach(alunoRecuperado ->{System.out.println(alunoRecuperado.toString());});  
 return ListaAlunosRecuperados;  
 }  
   
 //  
 public void cadastrarAlunosnoBancoInicial() {  
 this.repositoriodeAlunos.save(new Aluno(1,"Daniel Alfa","356.657.638.71","47.099.056-8", "Rua A"));  
 this.repositoriodeAlunos.save(new Aluno(2,"Ricardo Amaral","356.657.638.72","47.099.057-8", "Rua B"));  
 this.repositoriodeAlunos.save(new Aluno(3,"Bruna Bravo","356.657.638.73","47.099.058-8", "Rua C"));  
 this.repositoriodeAlunos.save(new Aluno(4,"Samuel Lopes","356.657.638.74","47.099.059-8", "Rua D"));  
 this.repositoriodeAlunos.save(new Aluno(5,"Sandra Gold","356.657.638.75","47.099.060-8", "Rua E"));  
 this.repositoriodeAlunos.save(new Aluno(6,"kiko Nascimento","356.657.638.76","47.099.061-8", "Rua F"));  
 this.repositoriodeAlunos.save(new Aluno(7,"Felipe Ferreira","356.657.638.77","47.099.062-8", "Rua G"));  
 this.repositoriodeAlunos.save(new Aluno(8,"Rogerio Mendes","356.657.638.78","47.099.063-8", "Rua H"));  
 this.repositoriodeAlunos.save(new Aluno(9,"Bruno Silva","356.657.638.79","47.099.064-8", "Rua I"));  
 this.repositoriodeAlunos.save(new Aluno(10,"Anderson Silva","356.657.638.80","47.099.065-8", "Rua J"));  
 this.repositoriodeAlunos.save(new Aluno(11,"Thiago Ricardo","356.657.638.81","47.099.066-8", "Rua K"));  
 this.repositoriodeAlunos.save(new Aluno(12,"Leandro Ossuna","356.657.638.82","47.099.067-8", "Rua L"));  
 this.repositoriodeAlunos.save(new Aluno(13,"Rogerio Renato","356.657.638.83","47.099.068-8", "Rua M"));  
 this.repositoriodeAlunos.save(new Aluno(14,"Rodrigues Alfa","356.657.638.84","47.099.069-8", "Rua N"));  
 this.repositoriodeAlunos.save(new Aluno(15,"Luiz Augusnto","356.657.638.85","47.099.070-8", "Rua O"));  
 this.repositoriodeAlunos.save(new Aluno(16,"Juliano Luidg","356.657.638.86","47.099.071-8", "Rua P"));  
 this.repositoriodeAlunos.save(new Aluno(17,"Aline Lins","356.657.638.87","47.099.072-8", "Rua Q"));  
 this.repositoriodeAlunos.save(new Aluno(18,"Daniel Roberto","356.657.638.88","47.099.073-8", "Rua R"));  
 this.repositoriodeAlunos.save(new Aluno(19,"Ferreira Augusto","356.657.638.89","47.099.074-8", "Rua S"));  
 this.repositoriodeAlunos.save(new Aluno(20,"Gabriel Lins","356.657.638.90","47.099.075-8", "Rua U"));  
 }  
}

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Visualizar Relatório de Alunos | Validar que todos os alunos contidos no banco de dados estão sendo mostrados. | - Clicar em “Visualizar Relatório de Alunos”. | Lista de alunos deve ser exibida corretamente com todos os alunos cadastrados no banco e seus respectivos dados. |
|  | Visualizar Relatório de Alunos | Validar que no caso de não haver alunos cadastrados no banco de dados, a mensagem de erro foi exibida. | - Antes de adicionar algum aluno no sistema, clicar em “Visualizar Relatório de Alunos”. | Alerta de que não existem alunos cadastrados no banco de dados deve ser exibido na tela. |
|  | Cadastrar Aluno | Validar alertas de restrições dos campos de preenchimento. | - Abrir formulário de cadastro de aluno.  - Inserir além do limite permitido e deixar de inserir dados em alguns campos.  - Clicar no link “Enviar”. | Alerta de erro ou falta de dados deve ser exibido na tela. |
|  | Cadastrar Aluno | Validar se registro dos campos preenchidos foi feito corretamente no banco de dados. | - Abrir formulário de cadastro de aluno.  - Inserir além do limite permitido e deixar de inserir dados em alguns campos.  - Clicar no link “Enviar”. | Lista de alunos deve ser exibida com a inclusão correta do aluno adicionado e seus respectivos dados cadastrados. |
|  | Visualizar Estatística de Presença | Validar que todos as presenças de um determinado aluno contidas no banco de dados estão sendo mostrados. | - Clicar em “Visualizar Relatório de Alunos”.  - Clicar no link “Visualizar Estatística de Presenças” de um determinado aluno. | Lista de todas as presenças de um determinado aluno devem ser exibidas corretamente junto com as estatísticas de presença disponíveis. |
|  | Visualizar Estatística de Presença | Validar que no caso de não haver presenças de um determinado aluno cadastradas no banco de dados, a mensagem de erro foi exibida. | - Clicar em “Visualizar Relatório de Alunos”.  - Clicar no link “Visualizar Estatística de Presenças” de um determinado aluno recém-adicionado ao sistema e sem presenças cadastradas. | Alerta deve ser exibido na tela. Com a informação de que não existem presenças cadastradas no banco de dados referentes ao aluno selecionado. |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 15/06/2023 | aula de visão geral do tcc, leitura da documentação de roteiro e escopo do tcc, | 1 hora |
| 16/06/2023 | análise da documentação de escopo para identificação de atores, requisitos e demais informações necessárias | 3 horas |
| 17/06/2023 | Identificação da ferramenta e desenvolvimento do diagrama de caso de uso | 2 horas |
| 19/06/2023 | Identificação da ferramenta e desenvolvimento do diagrama de caso de uso | 1 hora |
| 20/06/2023 | Identificação da ferramenta e desenvolvimento do diagrama de caso de uso | 1 hora |
| 21/06/2023 | Identificação da ferramenta e desenvolvimento do diagrama de caso de uso | 2 hora |
| 22/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Classe. | 1h30 |
| 23/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Classe. | 1h30 |
| 24/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Classe. | 2 horas |
| 24/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Componentes. | 2 horas |
| 25/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Componentes. | 5 horas |
| 25/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Implantação | 1 hora |
| 26/06/2023 | Identificação de ferramenta e desenvolvimento de diagrama de Implantação | 6 horas |
| 26/06/2023 | Definição de 3 casos de uso para o escopo do trabalho | 2 horas |
| 27/06/2023 | Definição de 3 casos de uso para o escopo do trabalho | 4 horas |
| 28/06/2023 | Especificação das Interfaces dos 3 casos de uso definidos. | 6 horas |
| 29/06/2023 | Especificação das Interfaces dos 3 casos de uso definidos. | 2 horas |
| 30/06/2023 | Desenvolvimento da lista de pontos de função e preenchimento da planilha. | 1 hora |
| 04/07/2023 a 07/07/2023 | Definição e setup de ferramentas, IDEs e frameworks, Treinamento online (LinkedIn Learning) a respeito do frameworks Spring Boot, Treinamento online (LinkedIn Learning) a respeito da API JPA. | 10 horas diárias |
| 08/07/2023 | Implementação/Codificação Back-End: entidades, repositórios, serviços e controladores. | 6 horas |
| 09/07/2023 a 18/07/2023 | Implementação/Codificação Front-End: HTML, CSS e Bootstrap. | 5 horas diárias |
| 18/07/2023 | Armazenamento da implementação/codificação em repositório de código site repositório | 1h30 |
| 19/07/2023 a 24/07/2023 | Ajustes código e texto. | 2 horas ao dia |
| 25/07/2023 a 10/08/2023 | Parte escrita do trabalho e sua finalização | 1 hora ao dia |

## Código da Aplicação

<https://github.com/caioaugsutinho/TCC>

## Avaliação Retrospectiva

No presente processo de criação do software, o maior problema enfrentado foi organizar o sistema para que de fato resolvesse o problema do cliente, uma vez que se tratava de um condomínio de médio porte, contando com mais de 465 moradores.

A organização do local possibilitou organizar o sistema de forma fácil e organizada.

## Objetivos Estimados

Os objetivos do referido projeto era que o sistema funcionasse de forma simples, quais todos os funcionários pudessem aprender de forma rápida, além do funcionamento continuo e facilitado do software, de forma que todos os moradores pudessem utilizar o Box do condomínio sem ser prejudicado pela superlotação. Tal objetivo foi estimado com grande expectativa.

## 13.2 Objetivos Alcançados

O objetivo de criação do software foi facilmente alcançado, tendo em vista uma vez que sei a real intenção era organizar o Box, para que todos os integrantes pudessem utilizar sem problemas, evitando superlotação, organizar melhor os horários de aulas, bem como criando grupos de aula, para um melhor uso.

## Lições aprendidas

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Organização de ideias | positivo |
| 2 | Ordem do planejamento | positivo |
| 3 | Manter a calma em relação a criação do trabalho | negativo |
| 4 | Facilidade para enfrentar os obstaculos encontrados no caminho | positivo |
| 5 | Como colocar ideias do papel na pratica | positivo |