



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**INSTITUTO UNIVERSIDADE VIRTUAL**  
**CURSO DE GRADUAÇÃO EM SISTEMAS E MÍDIAS DIGITAIS**

**PEDRO FELLIPE FREITAS LIMA**

**DESENVOLVIMENTO DE SISTEMA PARA AUTOMATIZAÇÃO DE ALOCAÇÃO  
DE SALAS NO CENTRO DE TECNOLOGIA DA UNIVERSIDADE FEDERAL DO  
CEARÁ**

**FORTALEZA**

**2020**

PEDRO FELLIPE FREITAS LIMA

DESENVOLVIMENTO DE SISTEMA PARA AUTOMATIZAÇÃO DE ALOCAÇÃO DE SALAS NO CENTRO DE TECNOLOGIA DA UNIVERSIDADE FEDERAL DO CEARÁ

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Dr. Leonardo Oliveira Moreira

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- L71d Lima, Pedro Fellipe Freitas.  
Desenvolvimento de sistema para automatização de alocação de salas no Centro de Tecnologia da Universidade Federal do Ceará / Pedro Fellipe Freitas Lima. – 2020.  
44 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2020.  
Orientação: Prof. Dr. Leonardo Oliveira Moreira.
1. Sistemas Web. 2. Aplicações para Universidades. 3. Processo de Alocação de Salas . 4. Processo de Desenvolvimento de Aplicações. I. Título.

CDD 302.23

---

PEDRO FELLIPE FREITAS LIMA

DESENVOLVIMENTO DE SISTEMA PARA AUTOMATIZAÇÃO DE ALOCAÇÃO DE  
SALAS NO CENTRO DE TECNOLOGIA DA UNIVERSIDADE FEDERAL DO CEARÁ

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Sistemas e Mídias  
Digitais do da Universidade Federal do Ceará,  
como requisito parcial à obtenção do grau de  
Bacharel em Sistemas e Mídias Digitais.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Leonardo Oliveira Moreira (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Gabriel Antoine Louis Paillard  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Emanuel Ferreira Coutinho  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Agradeço primeiramente à meus pais Sebastião e Esmeraldinha, por todo o apoio que precisei nesta jornada dura de aprendizado, e meus irmãos Gustavo e Gabriel, que proveram inspiração e encorajamento.

Agradeço aos meus amigos e colegas do Sistemas e Mídias Digitais que por sempre contribuem com apoio moral e sentimental. É através da amizade que aprendo coisas novas irrelevantes e cresço mentalmente. É vendo o sucesso dos meus próximos que vejo meu futuro e como posso melhorar.

Ao Prof. Leonardo Moreira por me orientar neste trabalho em meio a tempos difíceis. Um ombro atencioso que nunca deixou de falhar e ensinar.

Aos parceiros de desenvolvimento Hyan Santos, Franklyn Seabra e Lucas Esteves, durante a produção do produto deste trabalho, que por pouco tempo, mas marcante, fomos bons amigos.

## RESUMO

Recorrentemente o processo de alocação de salas em universidades se prova uma atividade complexa, alocar turmas às salas de aulas seguindo as devidas restrições de forma a comportar o corpo estudantil. Este problema é conhecido como *Classroom Assignment Problem* ou Problema de Alocação de Salas (PAS). Geralmente este processo é executado pelas instituições de ensino de forma manual e pode levar diversos dias para ser concluído. Tal situação descreve o estado atual enfrentado pelo Centro de Tecnologia da Universidade Federal do Ceará (UFC), que precisa lidar com a crescente demanda. O Centro conta com um total de 6 prédios, que comportam um total de 61 salas no momento da pesquisa, com capacidades que variam de 20 a 70 alunos. Este estudo se propôs a desenvolver uma forma de tornar mais eficiente o processo de alocação de salas do Centro de Tecnologia da UFC, eliminando a necessidade do esforço individual interno. Para tanto foi necessário tornar digital o processo, de forma que seja completamente efetivo, através de um sistema web. Foram utilizados como referência os dados das disciplinas ofertado a todos os semestres em 2019.1. De modo geral, os resultados obtidos, por meio do sistema desenvolvido, mostraram que o espaço disponível consegue comportar a demanda de forma eficiente, assim o esforço pelo Centro de Tecnologia da UFC tornou-se diluído.

**Palavras-chave:** Sistemas Web. Aplicações para Universidades. Processo de Alocação de Salas. Processo de Desenvolvimento de Aplicações.

## ABSTRACT

Recurrently, the process of allocating classrooms in universities proves to be complex. Allocating classes throughout classrooms following given restrictions to bear the student body. This problem is known as Classroom Assignment Problem. Generally this process is done by the educational institutions by manual means and can take days to be finished. This situation describes the current state faced by the Technology Center of the Federal University of Ceará, which has to deal with the increasing demand. The Center counts with a total of 9 graduation courses allocated through 6 buildings, which house a total of 61 rooms at the time of the research, with capacities that range from 20 to 70 students. This research proposes the development of a way to optimize the process of allocating classrooms in the Technology Center of Federal University of Ceará, eliminating the need of internal individual effort. To do so was necessary to digitalize the process, to be completely standalone, through a Web Software. The reference used in this research were the courses offered in all semesters of 2019.1. In general, the results obtained, through the developed system, showed that the available space can handle the demand efficiently, so the effort by the UFC Technology Center became reduced.

**Keywords:** Web Systems. Applications for Universities. Classroom Allocation Process. Application Development Process.

## LISTA DE FIGURAS

Figura 1 – Dados referentes aos blocos e suas respectivas salas . . . . .	24
Figura 2 – Diagrama Relacional . . . . .	27
Figura 3 – Página Inicial . . . . .	28
Figura 4 – Versões do Menu por Tipo de Usuário . . . . .	29
Figura 5 – Página de Consulta de Turmas . . . . .	30
Figura 6 – Representação da Localização do Bloco . . . . .	31
Figura 7 – Página de Reserva de Sala . . . . .	31
Figura 8 – Modal de Reserva de Sala . . . . .	32
Figura 9 – Página de Solicitação de Turma . . . . .	33
Figura 10 – Página de Gerência de Disciplinas . . . . .	34
Figura 11 – Modal de Gerência de Disciplinas . . . . .	34
Figura 12 – Página de Gerência de Turmas . . . . .	35
Figura 13 – Modal de Gerência de Turmas e Reservas . . . . .	35
Figura 14 – Página de Gerência de Salas . . . . .	36
Figura 15 – Modal de Gerência de Salas . . . . .	37
Figura 16 – Página de Alocação Automática de Salas . . . . .	37
Figura 17 – Página de Alteração de Data da Alocação de Salas . . . . .	38



## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CRUD	<i>Create, Read, Update e Delete</i>
CSS	<i>Cascading Style Sheets</i>
CTUFC	Centro de Tecnologia da UFC
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
JSF	<i>JavaServer Faces</i>
LINQ	<i>Language Integrated Query</i>
MVC	<i>Model-View-Controller</i>
ORM	<i>Object-Relational Mapping</i>
PAS	Problema de Alocação de Salas
PNE	Portador de Necessidade Especial
SALAS	Sistema de Alocação Automática de Salas
SGBD	Sistema Gerenciador de Bancos de Dados
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SQL	<i>Standard Query Language</i>
UFC	Universidade Federal do Ceará
URL	<i>Uniform Resource Locator</i>
XML	eXtensible Markup Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Contextualização e Motivação</b>	<b>11</b>
<b>1.2</b>	<b>Definição do Problema</b>	<b>12</b>
<b>1.3</b>	<b>Objetivos</b>	<b>13</b>
<i>1.3.1</i>	<i>Objetivo Geral</i>	<i>13</i>
<i>1.3.2</i>	<i>Objetivos Específicos:</i>	<i>13</i>
<b>1.4</b>	<b>Estrutura do Documento</b>	<b>13</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>14</b>
<b>2.1</b>	<b>Gerenciamento de Banco de Dados Relacional</b>	<b>14</b>
<b>2.2</b>	<b>Aplicação Web</b>	<b>15</b>
<b>2.3</b>	<b>Padrão Model-View-Controller (<i>Model-View-Controller</i> (MVC))</b>	<b>16</b>
<b>2.4</b>	<b>ASP.NET</b>	<b>17</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>19</b>
<b>3.1</b>	<b>Aspectos Metodológicos</b>	<b>19</b>
<b>3.2</b>	<b>Etapas Importantes do Processo Metodológico</b>	<b>19</b>
<i>3.2.1</i>	<i>Levantamento Completo dos Requisitos</i>	<i>19</i>
<i>3.2.2</i>	<i>Coleta de Dados</i>	<i>20</i>
<i>3.2.3</i>	<i>Tecnologias</i>	<i>20</i>
<i>3.2.4</i>	<i>Implementação</i>	<i>21</i>
<i>3.2.5</i>	<i>Tratamento de Limitações das Tecnologias Escolhidas</i>	<i>21</i>
<b>4</b>	<b>ASPECTOS TÉCNICOS E DE IMPLEMENTAÇÃO DO PRODUTO</b>	
	<b>MULTIMÍDIA</b>	<b>23</b>
<b>4.1</b>	<b>Contextualização e Requisitos</b>	<b>23</b>
<i>4.1.1</i>	<i>Centro de Tecnologia</i>	<i>23</i>
<i>4.1.2</i>	<i>Requisitos do Alocação de Salas</i>	<i>23</i>
<b>4.2</b>	<b>Projeto Lógico do Banco de Dados</b>	<b>26</b>
<b>4.3</b>	<b>Aspectos Funcionais do Produto</b>	<b>28</b>
<i>4.3.1</i>	<i>Página Inicial</i>	<i>28</i>
<i>4.3.2</i>	<i>Hierarquia</i>	<i>28</i>
<i>4.3.3</i>	<i>Consulta</i>	<i>30</i>

4.3.3.1	<i>Localização</i>	30
4.3.4	<i>Reserva de Sala</i>	30
4.3.5	<i>Solicitação de Turmas</i>	32
4.3.6	<i>Gerência</i>	32
4.3.6.1	<i>Gerência de Disciplinas</i>	33
4.3.6.2	<i>Gerência de Turmas</i>	33
4.3.6.3	<i>Gerência de Salas</i>	36
4.3.7	<i>Alocação Automática de Salas</i>	36
4.3.8	<i>Data da Alocação</i>	37
4.3.9	<i>Algoritmo de Alocação</i>	38
4.3.10	<i>Testes</i>	39
5	<b>CONCLUSÃO</b>	40
5.1	<b>Considerações Finais</b>	40
5.2	<b>Trabalhos Futuros</b>	40
	<b>REFERÊNCIAS</b>	42
	<b>APÊNDICES</b>	44
	<b>ANEXOS</b>	44
	<b>ANEXO A – Declaração de Autorização</b>	44

# 1 INTRODUÇÃO

Este capítulo apresenta a contextualização e motivação com intuito de proporcionar um entendimento sobre o tema tratado e o que incentivou a realização deste trabalho. Além disso, este capítulo, destaca o problema de pesquisa, os objetivos e a estrutura de capítulos contida no restante deste documento.

## 1.1 Contextualização e Motivação

O aumento constante de exigências de espaço físico em uma universidade, ao longo do tempo, é decorrente do ininterrupto ingresso de alunos posto em comparação ao não constante número de alunos egressos. Dado o fato que um universitário não é obrigado a encerrar sua graduação no tempo estipulado, seja em quatro, cinco, ou quaisquer outras variações de anos, a cada semestre novas turmas completas são iniciadas em cada curso. Com isso, é introduzido um problema referente ao gerenciamento das salas, disciplinas e turmas neste ambiente.

O problema conhecido como Problema de Alocação de Salas (PAS), do inglês *Classroom Assignment Problem*, e consiste no processo de alocar aulas, com horários de início e término previamente programados, a um número fixo de salas (CARTER; LAPORTE, 1996) (SCHAERF, 1999) (SOUZA *et al.*, 2002). O PAS estabelece existem horários de início e término de aulas em determinadas salas físicas. O problema trabalha a alocação de turmas em suas respectivas salas de forma que todos os requisitos sejam acatados de forma viável (SILVA; SILVA, 2010). Para Souza *et al.* (2002) a alocação de salas é tratada como parte integrante do problema de programação de cursos (*course timetabling*) ou como um problema derivado dele (*classroom assignment*) (BARDADY, 1996).

É plausível ressaltar que esta problema, o PAS, é recorrente em diversos contextos, e por tal motivo a comunidade acadêmica vem investigando e buscando soluções automatizadas para o processo de alocação de recursos sob restrições, tais quais: eventos esportivos (SCHÖNBERGER *et al.*, 2004) (MINE *et al.*, 2006), transporte (SEMET; SCHOENAUER, 2005), escalonamento de enfermeiras em hospitais (CROCE; SALASSA, 2014) e horários educacionais (AUBIN; FERLAND, 1989) (PETROVIC; BURKE, 2004) (BURKE *et al.*, 2007) (ACHÁ; NIEUWENHUIS, 2010).

A importância de lidar com o PAS é motivado pelo fato de ser um processo extenso e independente do escopo de onde é aplicado, em certos casos envolvendo múltiplas entidades

e, obrigatoriamente, recorrente durante anos. Neste contexto, a Universidade Federal do Ceará (UFC), nas funções de alocação de turmas, ainda faz uso de um processo manual para alocação de salas a cada semestre. O processo é dividido em diversas etapas, pois o processamento é realizado durante períodos estabelecidos no calendário acadêmico e a reajustes constantes. Além disso, esse processo envolve diversas pessoas, tais como: coordenadores de curso, chefes de departamentos, coordenadores acadêmicos etc.

A UFC disponibiliza bolsas de apoio ao desenvolvimento de aplicações diversas, dentre estas, projetos que envolvem automação de atividades organizacionais que antes eram manuais. Neste âmbito, o Centro de Tecnologia da UFC (CTUFC), ciente do peso recorrente que a alocação de salas acarreta no início de todo semestre, o qual envolve diretamente nove coordenações diferentes e vários docentes para abrigar inúmeras turmas de diversas disciplinas, em um processo que tem duração em dias, quando feito manualmente. Mesmo utilizando ferramentas como editores de planilhas, que tornam a atividade menos exigente em formalização, o esforço desta atividade se torna insistente.

## **1.2 Definição do Problema**

A solução proposta pela diretoria do CTUFC foi projetar um sistema que torne possível a organização e automatização do processo de alocação das salas na jurisdição do CTUFC, visando acelerar e melhorar tal processo. Como parte para a solução do problema, que motivou o projeto do sistema, almejou-se a utilização de técnicas no âmbito do PAS. O produto multimídia ou sistema, discutido neste trabalho em questão, deve ser capaz de unificar o processo de gerenciamento de salas, disciplinas e turmas entre as diversas partes envolvidas, tais quais coordenadores e docentes.

Além disso, a principal necessidade que levou a criação do projeto, foi eliminar por completo a etapa inicial de alocação de salas, restando apenas os ajustes com base nas necessidades decorrentes nos períodos seguintes definidos no calendário acadêmico da UFC. A partir deste ponto, o CTUFC tornou público o processo seletivo referente à contratação da equipe que seria responsável pelo desenvolvimento da aplicação, que projetaria e desenvolveria à aplicação ou sistema web chamada de Sistema de Alocação Automática de Salas (SALAS).

A pergunta problema que permeia este trabalho consiste em: como reduzir o esforço no processo de alocação de salas em turmas de disciplinas vinculadas ao CTUFC?

Hipótese: *Algoritmos inspirados no PAS podem ser implementadas em um sistema*

*para sugerir um processo de alocação de salas em turmas de disciplinas vinculadas ao CTUFC, reduzindo o esforço dos envolvidos neste processo?*

### **1.3 Objetivos**

#### ***1.3.1 Objetivo Geral***

Desenvolver uma aplicação para realizar o processo de alocação de salas referente à jurisdição do Centro de Tecnologia da UFC.

#### ***1.3.2 Objetivos Específicos:***

- a) coletar os requisitos e as tecnologias a serem utilizadas na aplicação;
- b) projetar uma arquitetura computacional para suportar os requisitos e as tecnologias almejadas na aplicação;
- c) mostrar os aspectos de modelagem e funcionais da aplicação.

### **1.4 Estrutura do Documento**

O Capítulo 2 reúne todo arcabouço teórico necessário para o entendimento do trabalho desenvolvido neste documento. Já o Capítulo 3 apresenta a metodologia utilizada e suas respectivas as etapas que foram seguidas para a realização deste trabalho. Os aspectos técnicos e de implementação do produto, solução para o problema, são explicados no Capítulo 4. Por fim, o Capítulo 5 apresenta as considerações finais e uma discussão sobre possíveis trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este capítulo apresenta todo arcabouço teórico, envolvendo os conceitos de banco de dados relacional, aplicações web, padrão de arquitetura para projeto de aplicações e algumas tecnologias para o desenvolvimento de aplicações web que fazem uso de banco de dados como mecanismo de persistência.

### 2.1 Gerenciamento de Banco de Dados Relacional

Os bancos de dados surgiram como uma alternativa mais robusta e eficiente para gerenciamento de dados, que antes persistiam em sistemas de arquivos, por parte das aplicações orientada a dados (DATE, 2004) (ELMASRI; NAVATHE, 2005). No desenvolvimento de aplicações com persistência de dados em sistemas de arquivos, o programador tinha que se preocupar com os aspectos das regras de negócio das aplicações, codificação de caracteres e a falta de padronização de *Application Programming Interface* (API), *drivers* ou bibliotecas das linguagens de programação para manipulação de arquivos. Além disso, a abordagem de persistência em sistemas de arquivos apresenta problemas que envolvem baixo grau de concorrência, disponibilidade, integridade dos dados etc. (ELMASRI; NAVATHE, 2005).

Na alternativa que utiliza bancos de dados são reduzidos, em grande parte, esses problemas, reduzindo o esforço do programador que pode ter mais tempo para forçar nas regras de negócio da aplicação e não nas complexidades que envolvem os aspectos de persistência dos dados em sistemas de arquivos. Date (2004) comenta que um banco de dados pode ser delineado como um repositório para uma coleção de arquivos de dados organizados em uma estrutura que facilita os aspectos de recuperação e armazenamento dos dados em tais arquivos. Elmasri e Navathe (2005) apresentam características facilitadoras que a alternativa de banco de dados possui em oposição à alternativa de persistência em sistemas de arquivos, tais como: isolamento entre os programas aplicativos, compartilhamento de dados, recuperação após falhas, linguagem de consulta etc.

As características facilitadoras comentadas são implementadas por um Sistema Gerenciador de Bancos de Dados (SGBD). Segundo Elmasri e Navathe (2005) um SGBD é composto por um conjunto de programas que permite, aos usuários, criar e manter os bancos de dados. Bancos de dados que implementam o modelo de dados relacional são chamados de banco de dados relacional. Um banco de dados relacional pode ser representado como uma coleção de

relações (ELMASRI; NAVATHE, 2005), onde uma relação é similar a uma tabela, uma linha na tabela é similar a uma tupla e uma coluna na tabela é chamado de atributo. Elmasri e Navathe (2005) comenta que a linguagem de consulta *Standard Query Language* (SQL) é considerada um dos maiores motivos para o sucesso dos bancos de dados relacional. Essa linguagem é utilizada tanto para a definição do esquema do banco de dados relacional quanto para a manipulação dos dados em um banco de dados relacional.

## 2.2 Aplicação Web

Uma aplicação web (do inglês, *web application*) pode ser definida como um software aplicativo que utiliza a web como ambiente de execução (CONALLEN, 2003). Já um ambiente web, de uma forma sucinta, é composto por uma arquitetura, um protocolo de comunicação, um sistema de endereçamento e linguagens. A arquitetura cliente/servidor (KUROSE; ROSS, 2013) onde o processamento dos dados é dividido em módulos ou processos distintos, denominados cliente e servidor. O cliente é responsável por iniciar e terminar a interação com o servidor, requisitando algo. Já o servidor fica em execução contínua e aguardando por requisições dos clientes, recebendo e respondendo a solicitações dos clientes.

O protocolo *HyperText Transfer Protocol* (HTTP) é o protocolo que possibilita o transporte de dados, informações ou arquivos na web. O sistema de endereçamento, denominado *Uniform Resource Locator* (URL), é utilizado para identificar objetos na web, onde um objeto pode ser uma página web, imagens, documentos, vídeos etc. (KUROSE; ROSS, 2013). Já as linguagens podem ser divididas entre linguagens da camada de apresentação da aplicação web e linguagens de programação para camada de processamento das regras de negócio das aplicações web. Dentre as linguagens da camada de apresentação pode-se destacar o *HyperText Markup Language* (HTML) que é responsável pela estrutura de uma página web, o *Cascading Style Sheets* (CSS) que realiza a parte de apresentação de uma página web e, por fim, o JavaScript que é quem se permite realizar comportamentos dinâmicos em uma página web, por exemplo validação de formulários em uma página web. Já linguagens de programação para camada de processamento das regras de negócio, são linguagens que são utilizadas para escreverem códigos, que são mantidos em um servidor, no intuito de processar as regras de negócio na aplicação web a partir da requisição de um cliente.

Em uma típica aplicação web, em uma arquitetura cliente/servidor, o cliente é um navegador web e o servidor são softwares que fazem a gestão dos componentes de processamento



da lógica de negócio da aplicação, tais como: conteúdos estáticos e de linguagens de programação que fazem a geração de conteúdos dinâmicos, banco de dados etc. Um arquivo estático é um arquivo que não tem seu conteúdo modificado por qualquer requisição. Assim, esses arquivos estáticos podem ser gerenciados por um Servidor HTTP (TANENBAUM; STEEN, 2007), onde esse servidor apenas identifica o arquivo por meio da URL e responde, ao cliente, com o conteúdo do arquivo requisitado. Já para os conteúdos dinâmicos, o cliente faz uma requisição à URL do recurso dinâmico e um Servidor de Aplicação web (TANENBAUM; STEEN, 2007) faz a gestão desta requisição. Ao receber a requisição, o Servidor de Aplicação web identifica o código a ser executado, escrito em uma linguagem de programação, e realiza a execução deste código, opcionalmente com parâmetros fornecidos pelo cliente na requisição, e responde com o conteúdo que foi gerado pelo código em resposta ao cliente.

### 2.3 Padrão Model-View-Controller (MVC)

O padrão de arquitetura de software MVC foi desenvolvido no Smalltalk-80 e está sendo amplamente adotado nos projetos de software (GUANGCHUN *et al.*, 2003). Segundo Ning *et al.* (2008) o padrão MVC separa o processamento, controle de entrada e saída do programa e a representação dos resultados. O padrão arquitetural MVC, quanto adotadas em projeto de softwares, possuem três camadas separadas por diferentes responsabilidades. Sendo assim, Charoenporn (2019), comenta as responsabilidades estas três camadas que são: modelo (do inglês, *model*), visão (do inglês, *view*) e controle (do inglês, *controller*). O padrão MVC realiza um desacoplamento entre as camadas para aumentar a flexibilidade e favorecer ao reuso (GUANGCHUN *et al.*, 2003).

O modelo que encapsula as regras de negócio relativa ao processamento ou lógica de negócio da aplicação. Geralmente, em aplicações persistentes, o modelo, considerados objetos da aplicação, possuem comunicação com os dados armazenados e as operações de *Create, Read, Update e Delete* (CRUD), que são operações básicas para inserir, ler, atualizar e remover dados. A camada de visão representa os elementos de interface com o usuário, por exemplo caixas de texto e botões. A visão é considerada a camada de apresentação da aplicação, onde o usuário final visualiza o resultado de um processamento. Por fim, a camada de controle tem como função orquestrar todo o fluxo da aplicação, recebendo dados de entrada da camada de visão, por meio de uma solicitação, e identificando a camada de modelo que pode resolver essa solicitação.

De um modo geral, as interações entre as três camadas do MVC dar-se-ão da seguinte

forma: a visão, por meio das ações do usuário, envia uma solicitação a um controle; o controle intercepta a ação do usuário e mapeia essa ação que é enviada ao modelo, fluxo de ida, ou a uma visão, fluxo de volta, para realizar a alteração apropriada; o modelo, ao ser solicitado, realiza computações sobre as regras de negócio, produzindo um novo estado; e a visão, após o processamento, apresenta as informações que foram processadas ou atualizadas pelo modelo. É válido ressaltar que a visão não tem informações sobre o que a aplicação está executando, se limitando a receber instruções do controle e informações oriundas do modelo para fins de exibição. Diversos *frameworks*<sup>1</sup> MVC foram propostos para auxiliar o desenvolvimento de aplicações seguindo o padrão de arquitetura MVC, são exemplos de alguns destes *frameworks*: *JavaServer Faces* (JSF), Apache Struts, Spring Web MVC, ASP.NET etc.

## 2.4 ASP.NET

O *framework* ASP.NET é uma tecnologia para desenvolvimento web, da Microsoft<sup>2</sup>, que usa linguagens de programação orientada a objetos (GUPTA *et al.*, 2012). O ASP.NET foi criado para auxiliar desenvolvedores usando linguagens, ferramentas e bibliotecas desenvolvidas em Visual Basic, C# e F#. Existem dois componentes do *framework* ASP.NET, chamados Core e MVC, dedicados a dois objetivos distintos, para aplicações *desktop* e web respectivamente. Segundo Gupta *et al.* (2012) o componente ASP.NET MVC que gerencia a complexidade usando o padrão de arquitetura MVC, fornecendo funcionalidades que implementam o comportamento comum de uma aplicação web. Assim, pode ser considerada uma tecnologia que disponibiliza um conjunto de códigos reutilizáveis com o intuito de estabelecer agilidade e praticidade de uso.

Dentre os recursos da plataforma .NET, pode-se destacar o recurso denominado *Language Integrated Query* (LINQ). Dietrich e Chaudhari (2009) apresentam a LINQ como uma linguagem de consulta inovadora que preenche a lacuna entre bancos de dados e linguagens de programação orientadas a objetos. LINQ é considerada uma linguagem declarativa, fortemente tipada, baseada em programação funcional e expressões lambda (DIETRICH; CHAUDHARI, 2009). Segundo Dietrich e Chaudhari (2011) LINQ é mais do que uma abordagem *Object-Relational Mapping* (ORM), já que LINQ pode consultar dados armazenados como relações, objetos e eXtensible Markup Language (XML). ORM é uma técnica ou recurso que realiza o mapeamento ente objetos e relações, permitindo a manipulação de instâncias de objetos e que

<sup>1</sup> Um *framework* possui um conjunto de códigos ou componentes que resolvem problemas comuns por meio de uma abordagem genérica e padronizada, favorecendo aspectos de reuso e desenvolvimento rápido.

<sup>2</sup> Microsoft. Disponível em: <<https://www.microsoft.com/pt-br>>. Acesso em: 13 de outubro de 2020.

os estados destes objetos são traduzidas e persistidas em bancos de dados relacionais. Assim, neste cenário, LINQ inspeciona os objetos persistente e faz o mapeamento automático para SQL. Assim, LINQ permite praticidade de uso, legibilidade e verificação de segurança, diminuindo os possíveis erros sintáticos de SQL e a presença de instruções maliciosas.

### **3 METODOLOGIA**

Este capítulo apresenta a metodologia utilizada para elaborar este trabalho, destacando quatro etapas importantes, onde são destacados o levantamento de requisitos, a coleta de dados, tecnologias e o tratamento de limitações das tecnologias escolhidas. Assim, ressaltando o processo de desenvolvimento e as limitações do projeto.

#### **3.1 Aspectos Metodológicos**

Segundo Oliveira e Seabra (2015) as metodologias de desenvolvimento de software foram concebidas para utilização orientada de métodos, ferramentas e procedimentos, com foco na elaboração de um produto de software. Tais metodologias visam o aprimoramento, reduzir custos, tempo de execução e a melhoria da qualidade do produto final (OLIVEIRA; SEABRA, 2015).

O modelo incremental permite que o usuário ou cliente do produto priorize quais funcionalidades serão entregues primeiro (SOMMERVILLE, 2011). Assim, a cada entrega de um conjunto de funcionalidades que engloba uma iteração, esse conjunto de funcionalidades podem ser agregados ao produto em produção, evoluindo sempre em direção ao produto almejado pelo usuário ou cliente (OLIVEIRA; SEABRA, 2015).

#### **3.2 Etapas Importantes do Processo Metodológico**

Nas seguintes subseções são destacadas as etapas mais importantes embutidas no processo metodológico utilizado para a concepção do SALAS.

##### ***3.2.1 Levantamento Completo dos Requisitos***

A primeira parte da pesquisa teve caráter exploratório (WAZLAWICK, 2009), com o intuito de esclarecer as necessidades, reunir os conceitos necessários para a implementação do projeto e firmar o escopo do desenvolvimento. Esse processo ocorreu em duas etapas; na primeira foi organizar o antigo processo manual de distribuição de salas entre os cursos em um fluxograma, ordenando cada etapa de forma a se obter um escopo linear, de onde se possa obter informações, inserir novos fatos, e dado o formato do processo, recomeçar tudo de forma simples no próximo semestre letivo.

Sendo assim, após o levantamento as ações esperadas da aplicação incluem:

1. cadastro e atualização de usuários com hierarquia de administrador, coordenador e professor;
2. função login e edição dos dados pessoais de perfil;
3. cadastro e atualização de disciplinas e turmas, onde cada turma pertence a uma disciplina, em uma relação N para 1, respectivamente. Cada turma também é ministrada por um professor;
4. cadastro e atualização de salas;
5. solicitação de turmas por usuários do tipo coordenador, pois estes são responsáveis por injetar esses dados no sistema no determinado período de tempo;
6. escolha de sala para cada turma ministrada para usuários do tipo professor;
7. automatização de distribuição das turmas restantes em salas disponíveis nos determinados horários;
8. pesquisa das turmas alocadas no sistema, com representação visual do mapa do Campus do PICI e indicação de qual bloco a turma pertence; e
9. remoção completa e automatizada das turmas do dado semestre.

### **3.2.2 Coleta de Dados**

A segunda parte da pesquisa continua o caráter exploratório, onde a equipe catalogou o espaço físico da jurisdição do CTUFC. Cada sala disponível foi registrada em um documento, juntamente com localização (dividida em bloco e andar), identificação por número, capacidade, acesso a Internet, disponibilidade de lousa quadriculada (requisito exigido por professores) e acessibilidade. Por acessibilidade definiu-se que a sala encaixava-se em dois parâmetros, que eram estar no andar térreo do seu respectivo bloco ou ter acesso a rampas ou elevadores. Quaisquer sala que cumpria uma das exigências foi declarada como válida para acessibilidade.

### **3.2.3 Tecnologias**

A terceira parte da pesquisa teve caráter descritivo, onde foi estabelecido por sugestão do orientador que fosse utilizado a linguagem C# para desenvolvimento do *back-end* da aplicação e ReactJS<sup>1</sup> para o desenvolvimento do *front-end*. Ainda durante o início do processo, enquanto

<sup>1</sup> React. React: Uma biblioteca JavaScript para criar interfaces de usuário. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 13 de outubro de 2020.

a equipe se adequava ao formato estabelecido, notou-se incompatibilidade com a tecnologia ASP.NET e as funções do ReactJS, pois ambos tentavam assumir funções similares.

Deste ponto em diante o ReactJS foi abandonado, pois acreditava-se que não seria necessitado. A versatilidade do ASP.NET, no entanto, acaba ficando restrita ao domínio da Microsoft, pois seu ambiente de desenvolvimento nativo é o Visual Studio<sup>2</sup>, uma *Integrated Development Environment* (IDE) poderosa, mas infame por ser exigente. Apesar de tudo, o ambiente de desenvolvimento escolhido traz consigo todas as ferramentas para que o projeto funcione, fazendo uso do servidor IIS Express<sup>3</sup> e tendo comunicação nativa com o SGBD PostgreSQL<sup>4</sup>.

### 3.2.4 Implementação

A quarta parte do trabalho caracteriza a implementação da pesquisa descrita na produção de um produto palpável. As etapas de execução descrevem o processo de elaboração das arquiteturas e algoritmos que farão parte da estrutura do projeto. A partir das tecnologias escolhidas, pode-se fazer uso de bancos de dados relacionais, juntamente a processamentos feitos na linguagem de programação pertencente ao *back-end* da aplicação, para realizar gerência e alocações de salas de forma digital, obtendo resultados semelhantes ao mesmo processo manual, mas ao mesmo tempo reduzindo o esforço e tempo dedicados a tal.

### 3.2.5 Tratamento de Limitações das Tecnologias Escolhidas

A última parte do trabalho ficou dedicada a lidar com as limitações do projeto. Esta categoria se divide em duas funções específicas: a primeira é a incapacidade do sistema de ser auto reciclável, no sentido que inicialmente se imaginou que seria possível disparar as ações que executariam *scripts* cruciais para o projeto de forma automatizada. No entanto, esta possibilidade não foi executável, então o processo tornou-se acionável manualmente.

A segunda função foi automatizar a reciclagem de disciplinas no sistema. Dada a irregularidade das matrizes curriculares em cada semestre, um banco de dados produzido não será sempre válido, então para lidar com as incertezas, foi produzido uma ferramenta externa

<sup>2</sup> Microsoft. Visual Studio. Disponível em: <<https://visualstudio.microsoft.com/pt-br/>>. Acesso em: 13 de outubro de 2020.

<sup>3</sup> Microsoft. IIS Express Overview. Disponível em: <<https://docs.microsoft.com/en-us/iis/extensions/introduction-to-iis-express/iis-express-overview>>. Acesso em: 13 de outubro de 2020.

<sup>4</sup> PostgreSQL. PostgreSQL: The World's Most Advanced Open Source Relational Database. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 13 de outubro de 2020.

ao sistema, que coleta uma base nos dados disponíveis no Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), e a partir disso introduz todos os novos casos de disciplinas e professores que não existiam previamente no sistema de forma automática.

## 4 ASPECTOS TÉCNICOS E DE IMPLEMENTAÇÃO DO PRODUTO MULTIMÍDIA

Este capítulo apresenta os aspectos técnicos e de implementação do SALAS, incluindo contextualização, ambiente, requisitos, modelagem, projeto, implementação e aspectos funcionais do sistema que tem como objetivo surgir alocação de salas para turmas de disciplinas do CTUFC.

### 4.1 Contextualização e Requisitos

#### 4.1.1 *Centro de Tecnologia*

O Centro de Tecnologia da UFC conta com um total de quatorze coordenações dedicadas à graduação<sup>1</sup>, que correspondem aos cursos de Arquitetura e Urbanismo, Design, Engenharia Ambiental, Engenharia Civil, Engenharia de Computação, Engenharia de Energias Renováveis, Engenharia de Petróleo, Engenharia de Produção Mecânica, Engenharia de Telecomunicações, Engenharia de Teleinformática, Engenharia Elétrica, Engenharia Mecânica, Engenharia Metalúrgica e Engenharia Química.

Durante a realização deste trabalho, em termos de espaço físico, foi considerado o total de seis Blocos dentro do Campus do PICI: Bloco 707, Bloco 708, Bloco 711, Bloco 717, Bloco 726 e Bloco 727, cada um com diferentes tipos de acomodações, recursos e funcionalidades. De modo geral, o Bloco 707 agrupa a grande maioria da carga horária das aulas no conjunto, contando com o maior aglomerado de salas e melhor localização, sendo também o único bloco ativo durante o período noturno de aulas, até o presente momento. Além disso, o Bloco 711 é o único equipado com mesas de desenho exigidas para certas disciplinas aplicadas entre alguns dos cursos supracitados. A Figura 1 apresenta, de forma tabular, o bloco, suas salas, capacidade de lugares, existência de quadro e aspectos de acessibilidade.

#### 4.1.2 *Requisitos do Alocação de Salas*

Conforme os requisitos previamente citados, o processo de alocação de salas deve apresentar uma série de regras para que seja considerado efetivo. Considerar necessidade de acessibilidade e atender às necessidades de vagas das turmas são prioridades básicas. Este trabalho tem como base o conceito apresentado por Souza *et al.* (2002), que caracteriza requisitos

<sup>1</sup> UFC. Centro de Tecnologia: Coordenações dos Cursos de Graduação. Disponível em: <<https://ct.ufc.br/pt/graduacao/>>. Acesso em: 13 de outubro de 2020.



Figura 1 – Dados referentes aos blocos e suas respectivas salas

Bloco	Sala	Lugares	Quadro	Acessibilidade	Bloco	Sala	Lugares	Quadro	Acessibilidade
707	11	70	DQ	Sim	708	11	90	Q	Térreo
707	12	70	DQ	Sim	708	21	90	DC	Não
707	13	70	D	Sim	708	22	40		Não
707	14	70	DQ	Sim	708	23	70		Não
707	15	90	DQ	Sim	708	24	70	DQ	Não
707	21	70	DQ	Sim	708	25	30		Não
707	22	70	DQ	Sim	708	26	90	DQC	Não
707	23	70	DQ	Sim	711	11	Sala Desativada		Térreo
707	24	80	D	Sim	711	12	Sala Desativada		Térreo
707	25	90	DQ	Sim	711	13	Sala Desativada		Térreo
707	31	40	D	Sim	711	14	Sala Desativada		Térreo
707	32	50	D	Sim	711	15	20	DC	Térreo
707	33	60	D	Sim	711	16	23	DC	Térreo
707	34	50	D	Sim	717	11	60	DQ	Térreo
707	35	50	D	Sim	717	12	60	DQ	Térreo
707	36	50	D	Sim	726	11	20		Térreo
707	37	50	D	Sim	726	12	20		Térreo
707	41	40		Sim	726	13	20	Q	Térreo
707	42	50		Sim	726	14	20		Térreo
707	43	60		Sim	726	15	20		Térreo
707	44	50	D	Sim	726	16	20		Térreo
707	45	50		Sim	726	17	20		Térreo
707	46	50		Sim	726	18	20		Térreo
707	47	50		Sim	726	19	70	Q	Térreo
707	51	Sala Desativada		Sim	727	11	70	DQ	Térreo
707	52	50		Sim	727	12	70	Q	Térreo
707	53	60		Sim	727	13	70	D	Térreo
707	54	50		Sim	727	14	70	D	Térreo
707	55	Sala Desativada		Sim	727	21	40	D	Sim
707	56	50		Sim	727	22	Sala Desativada		Sim
					727	23	40		Sim

Fonte: Elaborado pelo Autor.

essenciais e não essenciais. Podemos considerar como essenciais aqueles requisitos que são obrigatórios para a validade do processo e como não essenciais aqueles que são desejáveis, mas não obrigatórios.

Desta forma, o processo de alocação já executado pelo CTUFC contém os seguintes requisitos:

**Essenciais:**

1. as salas escolhidas devem comportar ou, se possível, ser maior que o valor exigido pela turma alocada;

2. aulas do período noturno devem ser obrigatoriamente isoladas no Bloco 707; e
3. dada necessidade de material, aulas das disciplinas de desenho devem ser alocadas nas salas do Bloco 711.

**Não Essenciais:**

1. priorizar salas com fator acessibilidade para usuários caracterizados como Portador de Necessidade Especial (PNE); e
2. quando solicitado, priorizar salas com quadro quadriculado.

Além dos requisitos supracitados, existem, no entanto, outros fatores que podem ser considerados informais. Destes pode-se citar os desejos de docentes de não permitir que suas turmas sejam realocadas, ou seja, sempre estar na mesma localização independente do semestre ou que turmas de primeiro semestre estejam sempre localizadas nas mesmas salas.

A problemática, gerada por estes fatores não regulados, é que gera-se um empecilho constante no processo de alocação, pois o algoritmo não pode fluir, já que a aproximação matemática de um algoritmo de alocação de salas é feita em um ambiente livre. Dessa forma, pode-se expressar fielmente os dados para que se otimize complexidade e desempenho. Todavia, para viabilizar esta ideia, é necessário que não haja nenhuma forma de intervenção na forma como as variáveis disponíveis são obtidas e processadas.

Deste modo, uma das soluções adotadas pela direção do CTUFC foi eliminar os fatores externos durante o processo automático. Por fatores externos, considera-se o histórico de intervenção de docentes nas escolhas nomeadas pela instituição por motivos de conforto. A medida foi tomada pois é uma ferramenta que atua na etapa inicial do semestre letivo, onde será afetado pelos eventuais reajustes.

Adotou-se então uma solução que torna o modelo matemático menos crucial. Separou-se o processamento de vagas em duas fases com duas etapas cada. A primeira fase lida com os docentes cadastrados como PNE, a segunda lida com os docentes restantes. Ambas as etapas são similares em estrutura da seguinte forma: estipula-se um período onde serão permitidos aos docentes cadastrados executarem reservas manualmente para as disciplinas registradas sob seus nomes.

Portanto, a opção de reserva de sala permanece bloqueada em quaisquer outro período para todos os usuários não administradores. Após a conclusão desta etapa manual, automatiza-se o remanejamento de salas para os usuários pertencentes a fase em questão. Considera-se restante todas as turmas que permaneceram sem uma sala no período manual.

Vale ressaltar que o caráter dos dados não pode ser considerado mutável para os usuários, no sentido que, caso haja conflito de reservas, apenas a primeira será aceita, não havendo sobrescrita nos registros.

Assumindo um caráter híbrido de manual e automático, o modelo matemático que possa ser estabelecido torna-se ineficiente. Não ineficaz ou inútil, no entanto, mas menos crucial assumindo as capacidades das tecnologias utilizadas e o intuito de tornar esta decisão neutra. Assumindo o potencial tecnológico escolhido para ambos a construção da aplicação quanto manipulação de bancos de dados, têm-se ferramentas que tornam simples o processo de alocação automática.

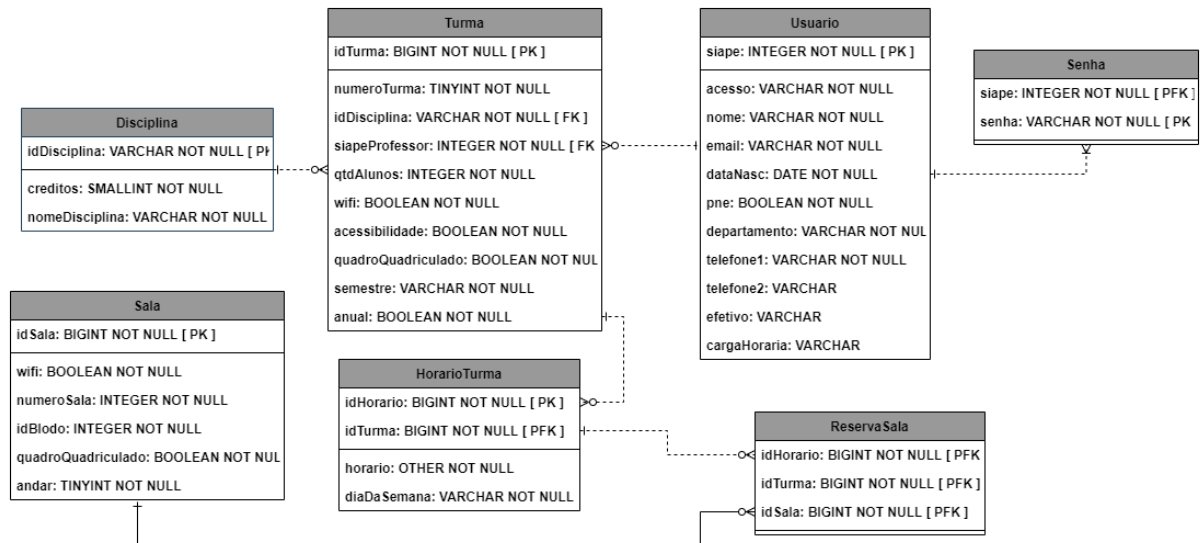
O processo automático é executado nas seguintes etapas:

- obtém-se todas as turmas cadastradas, conseqüentemente obtém-se as disciplinas vinculadas, as turmas são ordenadas por quantidade de alunos de forma decrescente;
- para cada turma, baseado na quantidade de alunos, obtém-se uma lista de salas não ocupadas no horário cadastrado que sejam capazes de comportá-la, caso não se obtenham resultados, estende-se o limite permitido em dez vagas para expandir o escopo da busca; e
- escolhe-se aleatoriamente uma sala da lista obtida;

## **4.2 Projeto Lógico do Banco de Dados**

Um projeto lógico de um banco de dados tem como objetivo descrever a estrutura ou esquema do banco de dados que está sendo projetado em modelo de dados, objetivando a avaliação se o modelo está correto, completo e se necessita passar por aspectos de avaliação de desempenho. A Figura 2 exhibe o Diagrama Relacional do banco de dados, no modelo relacional, utilizado pelo SALAS.

Figura 2 – Diagrama Relacional



Fonte: Elaborado pelo Autor.

Por meio da Figura 2, as relações estabelecidas entre as tabelas se dão da seguinte forma:

- um usuário possui informações pessoais e uma senha, senhas podem ser repetíveis entre usuários;
- uma turma faz parte de uma disciplina, podendo haver múltiplas turmas em uma mesma disciplina;
- uma turma possui horários específicos, relativo ao ciclo de repetição durante a semana; e
- uma turma pode ser vinculada a uma sala por meio de uma reserva. Esta reserva ocupa o horário específico no dia da semana e uma sala, sendo estes valores únicos, então uma mesma sala não pode ser ocupada por duas turmas ao mesmo tempo.

Esse modelo foi elaborado a partir de uma deliberação que facilita a identificação de relações e evita estresse no servidor. Isolar as entidades e fixar relações através de chaves estrangeiras torna simples as vinculações, enquanto mantém essas entidades modularizadas, de forma que alterações fora dessas relações não geram efeitos em cascata.

O desenvolvimento deste modelo exigiu cuidados nas atribuições de relacionamentos. As principais dificuldades se compuseram da necessidade de isolar as principais informações que serão tratadas como chaves estrangeiras de forma a manter os valores únicos referenciáveis. Estes valores devem ser coerentes com os estados das entidades relacionadas e rapidamente

mutáveis.

## 4.3 Aspectos Funcionais do Produto

### 4.3.1 Página Inicial

A página inicial, ilustrada na Figura 3 representa o padrão de cada tela. A barra lateral e o menu ao topo ficam expostos em todas as páginas remanescentes.

Figura 3 – Página Inicial



Fonte: Elaborado pelo Autor.

### 4.3.2 Hierarquia

Dado o requisito de hierarquia, um usuário possui um atributo “Acesso” em seu perfil. Este atributo determina qual nível de permissão no sistema este terá acesso. Sendo assim, o sistema possui também, diferentes versões para representar estes estados. A Figura 4 apresenta, da esquerda para direita, os níveis de acesso baseado em hierarquia e suas ações exclusivas.

Figura 4 – Versões do Menu por Tipo de Usuário

SALAS			
Início	Início	Início	Início
Consultar Horários	Consultar Horários	Consultar Horários	Consultar Horários
Tabela das Salas	Meu Horário	Meu Horário	Gerenciar Dados ▾
Sobre	Reserva de Sala	Reserva de Sala	Salas
Voltar ao CT	Tabela das Salas	Solicitar Sala	Usuários
	Sobre	Tabela das Salas	Disciplinas
	Voltar ao CT	Sobre	Turmas
		Voltar ao CT	Reservar Sala
			Alterar Reserva
			Tabela das Salas
			Alocação Automática
			Data da Alocação
			Sobre
			Voltar ao CT

Fonte: Elaborado pelo Autor.

Como anônimo um usuário somente tem acesso à consulta do calendário de turmas cadastradas. Um usuário do tipo “Professor” pode executar as funções específicas de solicitar reserva de sala para as turmas cadastradas sob seu nome e consultar o horário filtrado destas.

Um usuário do tipo “Coordenador” tem acesso às funções de um usuário do tipo professor, já que é possível que os cargos coincidam, adicionalmente da função específica de solicitar turmas para os docentes de seu departamento. Esta solicitação é o que permitirá os professores de executarem reservas para suas turmas posteriormente.

Por fim, o usuário do tipo “Administrador” possui acesso às funções de gerência do sistema. São eles que podem alterar os dados das salas, disciplinas e turmas manualmente. Também são os únicos autorizados a executar os programas de alocação automática de salas e limpeza de banco de dados.

### 4.3.3 Consulta

A consulta produz resultados baseados nos filtros aplicados. Uma busca vazia reproduz todas as turmas alocadas no sistema. Cada página da lista resultado contém um número estipulado de turmas. A separação por páginas foi aplicada para aliviar o estresse de organização durante a renderização da página. A Figura 5 apresenta a tela de consulta de turmas.

Figura 5 – Página de Consulta de Turmas

Consultar

Disciplina  
Ex: Cálculo I

Código  
Ex: CT123

Professor  
Ex: Carlos

Turma  
Ex: 1


Semestre  
Ex: 2018.1

Dia

Hora Inicio

Hora Fim

Consultar

Mapa	Código	Disciplina	Turma	Bloco	Sala	Professor	Horário
			5	707	32	Professor	Qua : 11:00 - 12:00 Seg : 10:00 - 12:00

Fonte: Elaborado pelo Autor.

#### 4.3.3.1 Localização

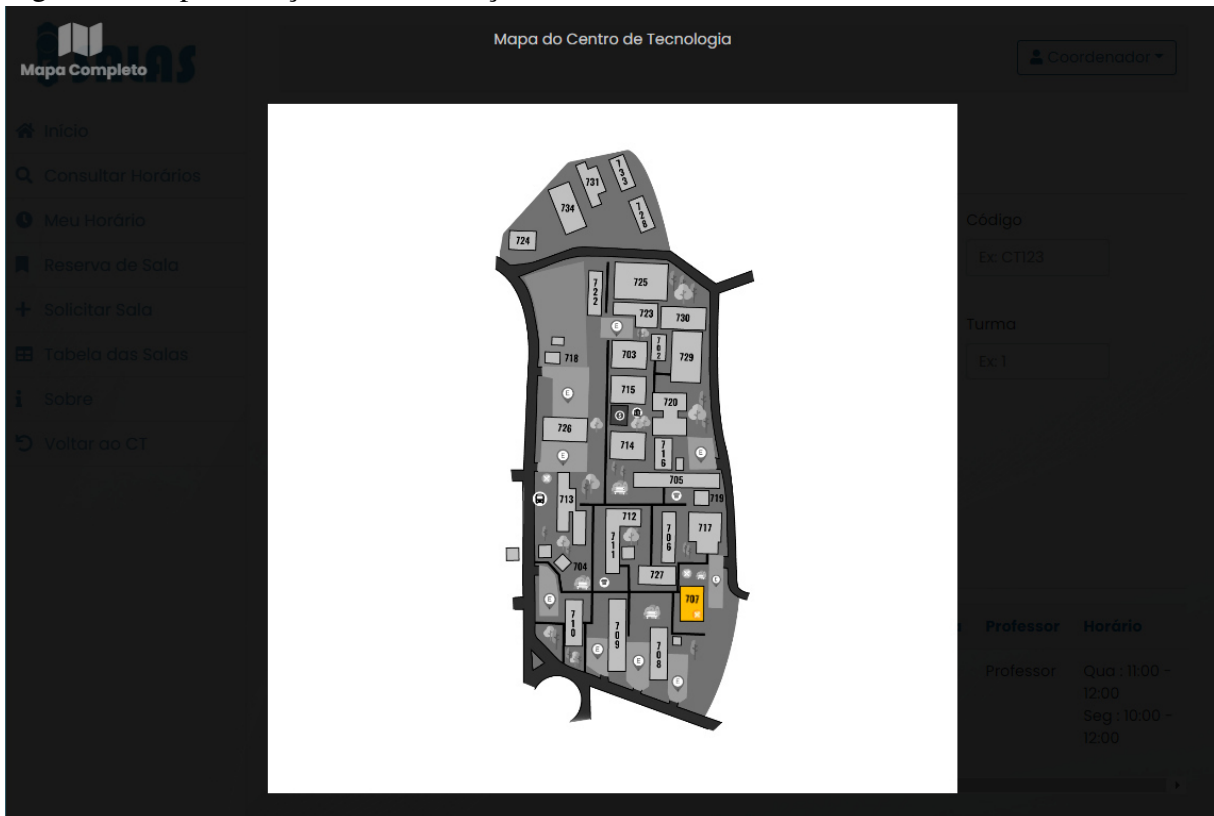
Cada turma resultante de uma consulta possui um atributo “Bloco”. A partir deste valor é produzido uma imagem do mapa da jurisdição do Centro de Tecnologia no Campus do PICI com foco no bloco ao qual a turma pertence, para fins de referência. A Figura 6 ilustra a tela que representa a localização do bloco.

### 4.3.4 Reserva de Sala

A página de reserva de sala, representada pela Figura 7, lista as turmas do docente cuja sessão está ativa. Para cada turma existe um modal, ilustrada pela Figura 8, independente que é então populado com as salas disponíveis consideradas compatíveis com a turma escolhida. Dados os requisitos de alocação, somente serão apresentadas salas livres e com lugares adequados.

Os dados exibidos são resultado de um processamento que segue os mesmos parâmetros da alocação automática. A lista salas disponíveis produzida para cada horário da turma

Figura 6 – Representação da Localização do Bloco



Fonte: Elaborado pelo Autor.

Figura 7 – Página de Reserva de Sala

### Lista de reservas

Código	Disciplina	Turma	
		1	<a href="#">Reservar</a>

Fonte: Elaborado pelo Autor.

escolhida deve, obrigatoriamente, conter apenas salas que comportem a quantidade de alunos para a turma e que não esteja ocupada no dado horário. Quando o usuário confirma sua escolha, a reserva torna-se permanente, só sendo possível ser alterada por um “administrador”.

Para a obtenção desta lista de salas foi utilizado um algoritmo que é carregado junto à página ou, como neste caso, para ser mais específico, quando o respectivo modal é aberto. A partir da arquitetura MVC pode-se estabelecer os dados que serão disponibilizados para o usuário através do “controle” dedicado. O algoritmo identifica, através do usuário identificado na sessão, quais turmas pertencem a sua identificação.

Tendo em mãos essas turmas, o algoritmo então executa uma busca no banco de dados, utilizando SQL, por salas disponíveis, ou seja, não alocadas em reserva no horário da



Figura 8 – Modal de Reserva de Sala

**Reservar**

**Turma 5**

Horário:  
 Qua : 11:00 - 12:00  
 Sala Atual: 32 / 707

Escolha uma nova Sala

Horário:  
 Seg : 10:00 - 12:00  
 Sala Atual: 32 / 707

Escolha uma nova Sala

Voltar Reservar

Fonte: Elaborado pelo Autor.

turma e que tenham capacidade para comportar o equivalente de vagas da respectiva turma. A lista obtida é então guardada no campo selecionável dentro do modal de sua respectiva turma em ordem crescente por bloco.

#### 4.3.5 *Solicitação de Turmas*

A solicitação de turma é feita por um “coordenador” em um período anterior ao destinado a reservas. Cada turma solicitada será listada como reserva pendente para o usuário cadastrado como docente. É uma etapa menos restrita. Cada coordenador pode solicitar diversas turmas para quaisquer disciplina, dado o caso que não há restrição entre os departamentos.

Uma turma pode ter até três horários em uma mesma semana, possivelmente no mesmo dia. O número de turmas não tem limite, mas a identificação não pode ser repetida e também não é sequencial, pois varia dependendo de qual departamento originou a solicitação. A Figura 9 apresenta a página de solicitação de turma.

#### 4.3.6 *Gerência*

As etapas de gerência são compostas das funções básicas de manipulação de dados. Cada uma é composta das funções básicas de adição, leitura, atualização e remoção, sendo limitadas pelo uso dos dados. Por exemplo, uma sala não pode deixar de existir enquanto ocupada, portanto deve-se realocar as respectivas turmas previamente.

Figura 9 – Página de Solicitação de Turma

## Solicitar Sala

The form 'Solicitar Sala' contains the following fields:

- Disciplina:** Dropdown menu with placeholder text "-- Seleccione uma Disciplina --".
- Código:** Dropdown menu with placeholder text "-----".
- Turma:** Text input with placeholder "Ex: 1".
- Nº de Alunos:** Text input with placeholder "Ex: 50".
- Semestre:** Text input with placeholder "Ex: 2020.1".
- Row 1:**
  - Dia:** Dropdown menu with "Segunda".
  - Hora Início:** Time picker with "--:--".
  - Hora Fim:** Time picker with "--:--".
  - Action:** Blue button with a "+" sign.
- Row 2:**
  - Dia:** Dropdown menu with "Segunda".
  - Hora Início:** Time picker with "--:--".
  - Hora Fim:** Time picker with "--:--".
- Row 3:**
  - Dia:** Dropdown menu with "Segunda".
  - Hora Início:** Time picker with "--:--".
  - Hora Fim:** Time picker with "--:--".
  - Action:** Red button with a "-" sign.
- Professor:** Dropdown menu with placeholder text "-- Seleccione um Professor --".
- SIAPE:** Dropdown menu with placeholder text "-----".
- Quadro Quadrado:** Dropdown menu with "Não".
- Wi-Fi:** Dropdown menu with "Não".
- Acessibilidade:** Dropdown menu with "Não".
- Confirmar:** Blue button at the bottom center.

Fonte: Elaborado pelo Autor.

### 4.3.6.1 Gerência de Disciplinas

Disciplinas possuem poucos atributos, pois sua principal função é identificação dentro de uma turma. Seu atributo mais importante é o código, pois é por ele que se isola disciplinas com nomes repetidos, já que é um atributo único. O código também representa a qual departamento a disciplina faz parte. A Figura 10 exibe a página e gerência de disciplinas.

O modal de edição, apresentado na Figura 11, não permite alteração do código, por ser um atributo único. No entanto, permite alteração do nome da disciplina e o valor de créditos. Por motivos de segurança, esta edição só é utilizável por usuários do nível "Administrador".

### 4.3.6.2 Gerência de Turmas

Turmas são o esqueleto das relações entre as tabelas ou entidades da aplicação. Uma turma possui uma identificação própria, mas também contém dados sobre disciplinas,

Figura 10 – Página de Gerência de Disciplinas

**Disciplinas**

Organizar por: Disciplina ▼      Pesquisar disciplina:  🔍      + Nova Disciplina

Código	Disciplina	Créditos		
1000	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1001	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1002	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1003	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1004	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1005	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>
1006	ALGEBRA LINEAR	6	<span>Editar</span>	<span>Deletar</span>

Fonte: Elaborado pelo Autor.

Figura 11 – Modal de Gerência de Disciplinas

**Editar**

Código

Disciplina

Créditos

Voltar      Confirmar

Fonte: Elaborado pelo Autor.

usuários, salas e horários. Uma turma pode ser criada, alterada ou excluída por usuários do nível “Administrador”, no entanto a responsabilidade de criação é dos usuários de nível “Coordenador”. A Figura 12 apresenta a página de gerência de turmas.

Via a tela de solicitação de turmas estes podem criar turmas com base nos dados apresentados. O formulário foi elaborado para unificar o formato dos dados de cada turma, visto

Figura 12 – Página de Gerência de Turmas

Disciplina	Turma	Professor	Alunos	Horário	Sala/Bloco		
...	5	Professor	50	Seg : 10:00 - 12:00	32/707	Editar	Deletar
...	5	Professor	50	Qua : 11:00 - 12:00	32/707	Editar	Deletar
...	1	...	20	Sex : 14:00 - 16:00	/	Editar	Deletar
...	1	...	20	Qua : 14:00 - 16:00	/	Editar	Deletar
...	1	...	40	Ter : 10:00 - 12:00	/	Editar	Deletar
...	1	...	40	Qui : 10:00 - 12:00	/	Editar	Deletar
...	1	...	40	Sex : 08:00 - 10:00	/	Editar	Deletar
...	1	...	40	Qua : 08:00 - 10:00	/	Editar	Deletar

Fonte: Elaborado pelo Autor.

que no modelo manual já existente, havia discrepância na forma como eram apresentados. Os dados de alteração de turma podem também incluir dados de reservas de sala, se existentes, permitindo remoção se necessário. A Figura 13 ilustra o modal de gerência de turmas e reservas.

Figura 13 – Modal de Gerência de Turmas e Reservas

**Editar**

Disciplina: [dropdown]

Professor: [dropdown] SIAPE: [input: 222222]

Semestre: [input: 2019.2] Turma: [input: 5] Nº de Alunos: [input: 50]

Dia: [dropdown: Segunda] Hora Início: [input: 10:00] Hora Fim: [input: 12:00]

Quadro Quadric.: [dropdown: Não] Wi-Fi: [dropdown: Não] Acessibilidade: [dropdown: Não]

Sala Reservada: 32/707 [Desalocar Sala]

[Voltar] [Confirmar]

Fonte: Elaborado pelo Autor.

#### 4.3.6.3 Gerência de Salas

Cada sala é uma representação do espaço físico real dentro do Campus do PICI. Para representar fielmente o estado de cada sala, possuem um atributo chamado “Ativo”. Este atributo permite que uma sala possa continuar existindo no banco de dados, assumindo que não está apta para uso, e ser eliminada do processo de alocação.

A necessidade de existência para este atributo se deu pelo fato que, durante o levantamento de dados, fomos apresentados com um andar em construção de um certo bloco, que estaria disponível para uso em um período próximo. A incerteza de disponibilidade foi resumida em um valor acessível por usuários do nível “Administrador”. A Figura 14 exibe a página de gerência de salas, enquanto a Figura 15 ilustra o modal de gerência de salas.

Figura 14 – Página de Gerência de Salas

Bloco	Andar	Sala	Vagas	Wi-Fi	Acessibilidade	Quadro Qadric.	
1	1	1	1	Não	Não	Não	<a href="#">Editar</a> <a href="#">Deletar</a>
707	1	11	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	1	12	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	1	13	70	Sim	Sim	Não	<a href="#">Editar</a> <a href="#">Deletar</a>
707	1	14	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	1	15	90	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	2	21	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	2	22	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	2	23	70	Sim	Sim	Sim	<a href="#">Editar</a> <a href="#">Deletar</a>
707	2	24	80	Sim	Sim	Não	<a href="#">Editar</a> <a href="#">Deletar</a>

Fonte: Elaborado pelo Autor.

#### 4.3.7 Alocação Automática de Salas

O menu de Alocação de Salas é de uso exclusivo para administradores. Sua função é unicamente ativar o *script* dedicado a alocar o restante das turmas em salas disponíveis. Por motivos de neutralidade, a escolha é feita aleatoriamente dentro do escopo estabelecido. Fez-se necessário o destaque na importância desta função via elementos visuais, tais quais o texto descritivo e o botão vermelho, pois executa um processo sem volta. A Figura 16 apresenta a página de alocação automática de salas.

Figura 15 – Modal de Gerência de Salas

**Editar**

Ativada <sup>?</sup>

Bloco:  Andar:

Sala:  Vagas:

Quadro Quadric.:  Wifi:  Acessibilidade:

[Voltar](#) [Confirmar](#)

70 Sim Sim

Fonte: Elaborado pelo Autor.

Figura 16 – Página de Alocação Automática de Salas

## Alocação Automática

### Informações

- Sobre a alocação**  
 A alocação automática distribuirá as turmas nas salas disponíveis considerando o número de alunos como a prioridade máxima na hora da escolha, assim evitando turmas pequenas de serem atribuídas à salas grandes. Além disso o sistema tomará em consideração as preferências dadas a cada turma no momento da solicitação. Todas as turmas do semestre atual que não tiveram salas reservadas serão automaticamente alocadas. Cuidado, pois este processo não pode ser desfeito.
- Ajustes manuais**  
 Caso exista necessidade de mudança manual, ajuste as reservas na opção [Alterar Reserva](#) do menu lateral.  
 Para desalocar uma turma de uma sala vá em [Gerenciar Dados > Turmas](#)  
 Selecione a opção Editar da turma desejada e clique em Desalocar Sala.

Atenção: Este botão executará a alocação automática de salas

[Confirmar](#)

Fonte: Elaborado pelo Autor.

#### 4.3.8 Data da Alocação

A data estipulada para a alocação automática é necessária para que os usuários do nível “Professor” tenham acesso a página de Reserva de Salas. Este atributo também contém uma variável Semestre, que será considerada pelo código de alocação automática como forma de filtro para quais turmas farão parte do processo. Este filtro se fez necessário para evitar possíveis

resquícios que possam existir durante o processamento. A Figura 17 apresenta a página de alteração de data da alocação de salas.

Figura 17 – Página de Alteração de Data da Alocação de Salas

**Alocação Automática**

Informações

- **Sobre a data**  
A alocação automática acontecerá primeiramente para os usuários marcados como PNE (Portadores de necessidades especiais) assim garantindo salas com acessibilidade. A data geral fará a alocação dos demais usuários. O semestre descrito aqui representará o semestre vigente para o sistema, qualquer turma que esteja com o semestre diferente deste não aparecerá como opção para reserva manual e não será considerada na reserva automática. Para alterar o semestre de uma turma vá em [Gerenciar Dados > Turmas](#)  
Selecione a opção Editar da turma desejada, modifique o campo Semestre e clique em Confirmar.

Data Geral	Data PNE	Semestre
29/05/2019 23:59:59	26/05/2019 23:59:59	2019.2

Atenção: Este botão mudará a data da alocação automática de Salas

**Confirmar**

Fonte: Elaborado pelo Autor.

#### 4.3.9 Algoritmo de Alocação

É plausível ressaltar que o algoritmo produzido é um fruto do desenvolvimento do projeto. Utilizando ferramentas oferecidas pela plataforma e manipulação de dados através de funções nativas da linguagem SQL, foi possível a produção de uma solução para o problema proposto. Para isso, reutilizou-se diversos blocos de códigos já exibidos anteriormente.

O fator central do algoritmo envolve encontrar as turmas cadastradas no semestre corrente. Para tal, reutiliza-se a função de listar turmas, da mesma forma que é apresentada na página de Gerência. No entanto, para o algoritmo não interessa turmas que já possuem reserva cadastrada, portanto, utilizando a função *LEFT JOIN* da linguagem SQL, pode-se unir os resultados referentes à Turmas aos dados atribuídos a tabela de Horários e Reservas. Utilizando a identificação primária da tabela de turma, que também encontra-se como chave estrangeira nas tabelas de horários e reservas, é possível reunir os resultados e tratá-los como uma informação só. Tendo os dados desejados, aplica-se um filtro referente a dados existentes a identificação de salas dentro dos valores obtidos da tabela de reservas.

Tendo as turmas não já reservadas, deve-se então buscar as salas disponíveis para cada uma. Para evitar possíveis casos de turmas pequenas ocuparem salas espaçosas, organiza-se

a lista de prioridades de turmas pelo número de alunos de forma decrescente. Uma sala pode ser considerada como disponível através da combinação de chaves guardadas pela tabela de reserva. A combinação de chaves estrangeiras identifica os horários aos quais as turmas podem ocupar em uma mesma sala, também significando que uma combinação não pode se repetir, pois identifica choque de horário. Utilizando a filtragem de uma pesquisa SQL, obtemos as salas disponíveis no horário da turma corrente, e utilizando uma segunda filtragem por quantidade de vagas, obtemos a lista de salas às quais a dada turma possa ocupar. Caso necessário, adiciona-se também os filtros de necessidade de acessibilidade, quadros quadriculados ou internet sem fio, de forma similar, mas como fatores opcionais, caso tornem a busca nula, são desligados.

Por fim, o algoritmo então seleciona, da lista de salas obtida, uma sala aleatória. Isso é produzido, na verdade, via limitação a um, dos resultados obtidos, em uma lista que foi ordenada de forma aleatória anteriormente. Como nenhum usuário irá observar o fluxo de dados, deve-se explicar o processo passo a passo, no entanto a execução de fato é feita de forma limitada desde a etapa de listagem de salas livres.

#### **4.3.10 Testes**

Durante o processo de desenvolvimento do produto, testes foram executados para garantir a funcionalidade do algoritmo. De forma não catalogada, no entanto, obteve-se resultados positivos repetidos, que podem afirmar que o espaço do Centro de Tecnologia é capaz de comportar de fato a demanda exigida no momento dado. É válido ressaltar que a demanda é crescente a cada semestre, assumindo que o fluxo de ingresso de discentes é maior que o fluxo de egressos, em termos de constância. Além disso, o espaço físico da jurisdição do CTUFC também é mutável, portanto há necessidade de manter observação sobre quais os fatores estão disponíveis para o sistema a qualquer dado momento. Dessa forma, espera-se que sempre seja possível obter resultados positivos nas condições estabelecidas.



## 5 CONCLUSÃO

Este último capítulo encerra o trabalho realizado com as considerações finais e as atividades futuras que podem ser realizadas como continuidade do trabalho.

### 5.1 Considerações Finais

Este trabalho apresentou o desenvolvimento de um sistema web, denominado SALAS, para gerenciamento e alocação de salas. O sistema deve ser capaz de suprir todo o processo de criação de salas, disciplinas e turmas entre as coordenações do CTUFC e executar, por meio da função de agendamento, a etapa de alocação de salas destinadas ao semestre corrente. Para tal, foi necessário um entendimento de todo o processo que envolve o objetivo do sistema, desde a administração, usuários, visitantes e dados do espaço físico do Campus do PICI. Destacou-se, também, os conceitos, tecnologias e logística para o funcionamento do processo automatizado, juntamente com a modelagem, projeto e implementação do SALAS.

Por meio do conteúdo apresentado no Capítulo 4, é possível perceber que a hipótese do trabalho pode ser confirmada, pois o SALAS faz sugestão de alocação de salas para turmas de disciplinas vinculadas ao CTUFC utilizando uma solução inspirada no PAS. Por fim, é válido destacar que o objetivo geral foi cumprido, pois o SALAS foi desenvolvido e apresentado no Capítulo 4. Além disso, ações definidas nos três objetivos específicos definidos podem ser visualizados no Capítulo 2, Capítulo 3 e Capítulo 4.

### 5.2 Trabalhos Futuros

Como trabalhos futuros almeja-se realizar atividades a serem feitas posteriormente, como continuidade deste trabalho, são as que estão na sequência.

- a) Expandir o sistema para englobar todos os campi pertencentes à UFC. Este fator foi um ponto considerado no início do desenvolvimento, mas não foi utilizado. No entanto, a flexibilidade para tal objetivo já existe em código, restando apenas a aplicação.
- b) Implementação de um canal de comunicação com os usuários Administradores, visto que imprevistos possam ocorrer. Ter um canal unificado facilita o gerenciamento.
- c) Implementação de uma ferramenta interna que permitirá comunicação com o

portal SIGAA, que facilitará o gerenciamento de novo conteúdo advindo dos novos semestres.

Da mesma forma, pode-se considerar a chance de revisão do produto já existente. Considera-se plausível a possibilidade de uso de modelos matemáticos para tratamento do PAS de forma real, dado o fato da solução atual ser proprietária, para melhor desempenho tanto em termos de tempo quanto em efetividade. Acredita-se que se aplicado de forma aceitável, pode ser tratada como uma ferramenta auto suficiente. Com o avanço tecnológico nos setores de adaptação de inteligência artificial com *Machine Learning*, pode-se considerar que é possível a execução de uma ferramenta que não só solucione o PAS de forma coerente, quanto se adapta a mudanças ao longo do tempo.

## REFERÊNCIAS

- ACHÁ, R. A.; NIEUWENHUIS, R. Curriculum-based course timetabling with sat and maxsat. In: **Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT)**. Belfast, Northern Ireland, UK: Queen's University Belfast, 2010. p. 42–56.
- AUBIN, J.; FERLAND, J. A. A large scale timetabling problem. **Computers & Operations Research**, Elsevier Science Ltd., v. 16, n. 1, p. 67–77, jan. 1989. ISSN 0305-0548.
- BARADYDYM, V. A. Computer-aided school and university timetabling: The new wave. In: BURKE, E. K.; ROSS, P. (Ed.). **Practice and Theory of Automated Timetabling**. Berlin, Heidelberg: Springer, 1996. (Lecture Notes in Computer Science, v. 1153), p. 22–45.
- BURKE, E. K.; MCCOLLUM, B.; MEISELS, A.; PETROVIC, S.; QU, R. A graph-based hyper-heuristic for educational timetabling problems. **European Journal of Operational Research**, v. 176, n. 1, p. 177–192, 2007. ISSN 0377-2217.
- CARTER, M. W.; LAPORTE, G. Recent developments in practical examination timetabling. In: BURKE, E.; ROSS, P. (Ed.). **Practice and Theory of Automated Timetabling**. Berlin, Heidelberg: Springer, 1996. (Lecture Notes in Computer Science, v. 1153), p. 1–21.
- CHAROENPORN, P. Designing web services with mvc for classification rice system. In: **Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning**. New York, NY, USA: Association for Computing Machinery, 2019. (IC4E '19), p. 411–415. ISBN 9781450366021.
- CONALLEN, J. **Building Web Applications with UML**. [S.l.]: Addison-Wesley, 2003. (Addison-Wesley Object Technology Series).
- CROCE, F. D.; SALASSA, F. A variable neighborhood search based metaheuristic for nurse rostering problems. **Annals of Operations Research**, v. 218, n. 1, p. 185–199, 2014. ISSN 1572-9338.
- DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. [S.l.]: Elsevier Editora, 2004.
- DIETRICH, S. W.; CHAUDHARI, M. The missing link between databases and object-oriented programming: Linq as an object query language for a database course. **Journal of Computing Sciences in Colleges**, Consortium for Computing Sciences in Colleges, Evansville, IN, USA, v. 24, n. 4, p. 282–288, abr. 2009. ISSN 1937-4771.
- DIETRICH, S. W.; CHAUDHARI, M. Linq rox! integrating linq into the database curriculum. In: **Proceedings of the 42nd ACM Technical Symposium on Computer Science Education**. New York, NY, USA: Association for Computing Machinery, 2011. (SIGCSE '11), p. 293–298.
- ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados**. [S.l.]: Pearson Brasil, 2005.
- GUANGCHUN, L.; LU, W.; HANHONG, X. A novel web application frame developed by mvc. **SIGSOFT Softw. Eng. Notes**, Association for Computing Machinery, New York, NY, USA, v. 28, n. 2, p. 7, mar. 2003. ISSN 0163-5948.
- GUPTA, P.; MATA-TOLEDO, R.; MONGER, M. Utilizing asp.net mvc in web development courses. **Journal of Computing Sciences in Colleges**, Consortium for Computing Sciences in Colleges, Evansville, IN, USA, v. 27, n. 3, p. 10–14, jan. 2012. ISSN 1937-4771.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet: uma abordagem top-down**. [S.l.]: Pearson Education do Brasil, 2013.

MINE, M. T.; SILVA, M. S. A.; SOUZA, M. J. F.; SILVA, G. P.; OCHI, L. S. Iterated local search aplicado à resolução do problema de programação de jogos de competições esportivas realizadas em turnos completos e espelhados: um estudo de caso. In: **Anais do XIII Congresso Latino-Iberoamericano de Investigación Operativa (CLAIO)**. Montevideo, Uruguay: IEEE, 2006. v. 1, p. 1–6.

NING, W.; LIMING, L.; YANZHANG, W.; YI-BING, W.; JING, W. Research on the web information system development platform based on mvc design pattern. In: **Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03**. USA: IEEE Computer Society, 2008. (WI-IAT '08), p. 203–206. ISBN 9780769534961.

OLIVEIRA, F. G.; SEABRA, J. M. P. Metodologias de desenvolvimento de software: Uma análise no desenvolvimento de sistemas na web. **Periódico Científico Tecnologias em Projeção**, v. 6, n. 1, p. 20–34, 2015. ISSN 2178-6267.

PETROVIC, S.; BURKE, E. University timetabling. In: **Handbook of Scheduling: Algorithms, Models, and Performance Analysis, chapter 45**. [S.l.]: Chapman Hall/CRC Press, 2004.

SCHAERF, A. A survey of automated timetabling. **Artificial Intelligence Review**, Kluwer Academic Publishers, USA, v. 13, n. 2, p. 87–127, abr. 1999. ISSN 0269-2821.

SCHÖNBERGER, J.; MATTFELD, D.; KOPFER, H. Memetic algorithm timetabling for non-commercial sport leagues. **European Journal of Operational Research**, Elsevier, v. 153, n. 1, p. 102–116, 2004. Timetabling and Rostering.

SEMET, Y.; SCHOENAUER, M. An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. In: **Proceedings of the 2005 IEEE Congress on Evolutionary Computation**. Edinburgh, Scotland, UK: IEEE, 2005. v. 3, p. 2752–2759.

SILVA, D. J.; SILVA, G. C. Heurísticas baseadas no algoritmo de coloração de grafos para o problema de alocação de salas em uma instituição de ensino superior. In: **Anais do XLII Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. Bento Gonçalves, RS, Brasil: [s.n.], 2010. p. 2839–2849.

SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Education do Brasil, 2011.

SOUZA, M. J. F.; MARTINS, A. X.; ARAÚJO, C. R. Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas. In: **Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. Rio de Janeiro, RJ, Brasil: [s.n.], 2002. p. 1100–1110.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas Distribuídos: Princípios e Paradigmas**. [S.l.]: Pearson Education do Brasil, 2007.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. [S.l.]: Elsevier Editora, 2009.

## **ANEXO A – DECLARAÇÃO DE AUTORIZAÇÃO**

Documento que autoriza o discente Pedro Fellipe Freitas Lima a relatar o desenvolvimento do sistema SALAS na elaboração do seu Trabalho de Conclusão de Curso.

Universidade Federal do Ceará  
Centro de Tecnologia  
Diretoria Adjunta de Ensino

## Autorização

Autorizo **PEDRO FELIPE FREITAS LIMA**, estudante curso de Sistemas e Mídias Digitais desta Universidade, relatar o desenvolvimento da aplicação SALAS – projeto aprovado e desenvolvido pela Diretoria Adjunta de Ensino do Centro de Tecnologia - DAECT, sob o nome de “Soluções Administrativas para o Centro de Tecnologia”, em Edital da Pró-Reitoria de Planejamento e Administração, projeto do qual o referido estudante foi bolsista e um dos desenvolvedores dentro da equipe vinculada a DAECT. Importante destacar que tal aplicação teve finalidade de otimizar o processo de alocação de turmas de disciplinas às salas de aula do Centro de Tecnologia – para fins de elaboração do seu Trabalho de Conclusão de Curso.

Fortaleza, 15 de outubro de 2020.



**Prof. Bruno Vieira Bertoncini**  
Diretor Adjunto de Ensino do Centro de Tecnologia