UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

# Modelo de Programação Linear Inteira para o Problema de Alocação de Salas: Estudo de caso em uma Instituição de Ensino Superior

Luciano Fernandes Acioli Cabral e Silva

João Pessoa
2019

Luciano Fernandes Acioli Cabral e Silva

# Modelo de Programação Linear Inteira para o Problema de Alocação de Salas: Estudo de caso em uma Instituição de Ensino Superior

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Produçãoda Universidade Federal da Paraíba, como requisito parcial para obtenção de Mestre em Engenharia de Produção

Orientador: Lucídio dos Anjos Formiga Cabral

João Pessoa
2019

**Catalogação na publicação**
**Seção de Catalogação e Classificação**

S586m Silva, Luciano Fernandes Acioli Cabral e.
Modelo de programação linear inteira para o Problema de Alocação de Salas: Estudo de caso em uma Instituição de Ensino Superior / Luciano Fernandes Acioli Cabral e Silva. - João Pessoa, 2019.
51 f.

Orientação: Lucídio dos Anjos Formiga Cabral.
Dissertação (Mestrado) - UFPB/CT.

1. Problema de Alocação de Salas. 2. Otimização Combinatória. 3. Pesquisa Operacional. I. Cabral, Lucídio dos Anjos Formiga. II. Título.

UFPB/BC

Luciano Fernandes Acioli Cabral e Silva

# Modelo de Programação Linear Inteira para o Problema de Alocação de Salas: Estudo de caso em uma Instituição de Ensino Superior

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Produção da Universidade Federal da Paraíba como parte dos requisitos parciais para obtenção do título de mestre em Engenharia de Produção.

Aprovada em: 19/08/2019

BANCA EXAMINADORA

Prof. Dr. Lucídio dos Anjos Formiga Cabral (Orientador)
Universidade Federal da Paraíba

Prof. Dr. Roberto Quirino do Nascimento
Universidade Federal da Paraíba

Prof. Dr. Sandro Marden Torres
Universidade Federal da Paraíba

João Pessoa
2019

# AGRADECIMENTOS

Agradeço a todos os amigos e familiares que sempre me incentivaram em meus projetos acadêmicos e pessoais, e também estiveram sempre ao meu lado nos momentos mais difíceis dessa caminhada.

À minha esposa Cybelle, amiga e companheira de todas as horas e que sempre esteve ao meu lado me apoiando, me aturando, me incentivando e fazendo com que eu não desistisse de seguir em frente.

Aos meus pais Irlan e Elvia, que são meus alicerces. Tenho eterna gratidão pelo que vocês fizeram e fazem por mim. Tenho vocês como modelo de pessoas responsáveis e de bom caráter que devo sempre tentar seguir em minha vida.

Ao me orientador Lucídio, professor responsável e atencioso que sempre atendeu aos meus chamados, auxiliou em minhas dúvidas e entendeu minhas dificuldades e ausências.

Ao meu amigo Luciano Costa, que mesmo sempre muito ocupado com sua pesquisa de doutorado, nunca se recusou em me ajudar quando me encontrava perdido.

Ao amigo André Lucena, outro amigo, quase irmão, que sempre me incentivou a prosseguir nessa jornada acadêmica e que sempre tinha palavras para me apoiar quando mais precisei.

Aos meus colegas de trabalho do IFPE que assumiram minhas aulas enquanto estive afastado das minhas atividades como docente para a realização deste mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) que disponibilizou bolsa de estudos para que esta pesquisa pudesse ser realizada de forma mais eficaz.

# RESUMO

Alocar turmas às salas de aula é um problema vivenciado a cada semestre em instituições de ensino. Esse problema é conhecido como *Classroom Assignment Problem*, ou Problema de Alocação de Salas (PAS), e consiste na alocação eficiente das turmas, com dias e horários preestabelecidos, às salas de aulas, levando em consideração as necessidades dos docentes, dos discentes, das disciplinas e respeitando a capacidade das salas. Geralmente, este problema é resolvido pelas instituições de forma manual, no entanto, por se tratar de um problema combinatório, manualmente, esse processo pode levar vários dias e ainda assim não obter resultados satisfatórios. Devido à sua importância, dificuldade de resolução e características próprias que dificultam uma solução generalizada, este problema vem sendo amplamente estudado. Nesta pesquisa, o Centro de Informática da Universidade Federal da Paraíba foi tomado como caso de estudo para o PAS. Foi desenvolvido um modelo de programação linear inteira e um sistema informatizado que faz uso desse modelo para resolver este problema e que poderá ser utilizado na prática pelos responsáveis pela alocação de salas no Centro de Informática da UFPB.

**Palavras-chave:** Problema de Alocação de Salas, Otimização Combinatória, Pesquisa Operacional.

# ABSTRACT

*Allocating classes to classrooms is a problem experienced every semester in all educational institutions. This problem is known as the Classroom Assignment Problem and consists of the efficient allocation of classes, with pre-established days and times, to classrooms, taking into account the needs of teachers, students, classes and respecting the capacity of the classrooms. Generally, this problem is solved by the institutions manually, however, because it is a combinatorial problem, manually, this process can take several days and still not get satisfactory results. Due to its importance, difficulty of solving and its own characteristics that make difficult a generalized solution, this problem has been widely studied. In this research, the Informatics Center of the Federal University of Paraíba was taken as a case study for the Classroom Assignment Problem. An integer linear programming model was developed and a computerized system that makes use of this model to solve this problem and can be used in practice by those responsible for room allocation at the UFPB Informatics Center.*



**Key-words:** Classroom Assignment Problem, Combinatorial Optimization, Operational Research.

# LISTA DE FIGURAS

Figura 1: Subdivisões do *Timetabling* 17
Figura 2: Cursos de Graduação e Pós-graduação no CI 25
Figura 3: Planilha para inserção dos horários das disciplinas 37
Figura 4: Planilha para inserção das preferências das disciplinas à cada sala 37
Figura 5: Planilha para inserção dos horários das disciplinas 38
Figura 6: Resultado geral da alocação das salas 39
Figura 7: Resultado da alocação apresentando o horário de cada sala 39

# LISTA DE QUADROS

Quadro 1: Principais métodos heurísticos e seus autores 19
Quadro 2: Métricas de qualidade para o PAS encontradas na literatura 22
Quadro 3: Características das salas de aula, laboratórios e auditório do CI 26
Quadro 4: Quantidade de turmas ofertadas em cada horário e dia da semana 35
Quadro 5: Comparação dos resultados da alocação real com a alocação gerada pelo modelo 40
Quadro 6: Dados das Instâncias utilizadas 42

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| CAPES | Coordenação de Aperfeiçoamento de Pessoal de Nível Superior |
| CC | Ciência da Computação |
| CCEN | Centro de Ciências Exatas e da Natureza |
| CI | Centro de Informática |
| CTDR | Centro de Tecnologia e Desenvolvimento Regional |
| EC | Engenharia da Computação |
| FO | Função Objetivo |
| IOS | Índice de Ocupação de Salas |
| LIEPE | Laboratório Interdisciplinar de Ensino, Pesquisa e Extensão |
| MC | Matemática Computacional |
| NP | Não Polinomial |
| P | Polinomial |
| PAAS | Problema de Alocação de Aulas a Salas |
| PAS | Problema de Alocação de Salas |
| PGC | Pós-graduação em Computação |
| PGI | Pós-graduação em Informática |
| PGMMC | Pós-graduação em Modelagem Matemática Computacional |
| PO | Pesquisa Operacional |
| SIGAA | Sistema Integrado de Gestão de Atividades Acadêmicas |
| UFPB | Universidade Federal da Paraíba |
| VBA | *Visual Basic for Applications* |

## Sumário


**1 INTRODUÇÃO** **13**

1.1 Considerações iniciais . . . . . 13

1.1.1 Objetivo geral . . . . . 14

1.1.2 Objetivos específicos . . . . . 14

1.1.3 Justificativa . . . . . 15

**2 FUNDAMENTAÇÃO TEÓRICA** **16**

2.1 *Timetabling* . . . . . 16

2.2 Otimização Combinatória e Programação Linear . . . . . 17

2.3 Heurísticas e Metaheurísticas . . . . . 18

2.4 Matheurística . . . . . 20

2.5 Problema de Alocação de Aulas . . . . . 20

2.5.1 Definição e variações do PAS . . . . . 20

2.5.2 Complexidade Computacional . . . . . 22

2.5.3 Métodos de resolução do problema . . . . . 23

**3 DESCRIÇÃO DO PROBLEMA** **25**

3.1 Histórico do Centro de Informática (CI) . . . . . 25

3.2 O Problema de Alocação de Salas no Centro de Informática . . . . . 26

3.3 Escopo do PAS no CI . . . . . 27

3.4 Formulação matemática inicial . . . . . 29

**4 PROCEDIMENTOS METODOLÓGICOS** **31**

**5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS** **33**

5.1 Modelo proposto . . . . . 33

5.2 Caracterização da instância inicial utilizada . . . . . 35

5.3 Execução do modelo . . . . . 35

5.4 Utilização do sistema proposto . . . . . 36

5.5 Análise e discussão dos resultados . . . . . 40

5.5.1 Instância inicial 2019.1 . . . . . 40
5.5.2 Novas instâncias . . . . . 41

**6 CONCLUSÕES** **43**

**REFERÊNCIAS** **44**

**APÊNDICE 1 - Alocação para a instância 2019.1** **48**

**APÊNDICE 2 - Alocação para a instância 2019.1+15** **49**

**APÊNDICE 3 - Alocação para a instância 2019.1+30** **50**

**APÊNDICE 4 - Alocação para a instância 2019.1+47** **51**

# 1 INTRODUÇÃO

## 1.1 Considerações iniciais

Cada vez mais, os problemas reais vêm sendo resolvidos com o uso de técnicas de otimização. Isto se dá porque soluções mais rápidas e economicamente viáveis são cada vez mais necessárias para o gerenciamento de problemas com recursos restritos. Independentemente do resultado almejado, tais como, minimização de custos ou tempo, ou mesmo maximização da eficiência, o uso de técnicas adequadas possibilita a obtenção não somente de uma solução viável, mas também de resultados que visem alcançar algum objetivo pré-determinado levando em consideração as condições e restrições existentes para cada situação.

A cada semestre, as instituições de ensino superior necessitam determinar em qual sala de aula cada turma das disciplinas ofertadas no período irá ser alocada. Esta escolha deverá obedecer a uma série de restrições, como por exemplo, todas as aulas de turmas de um mesmo curso deverão estar em uma mesma sala (PRADO, 2014).

Como apresentado por Sales (2015b), de forma geral, grande parte das instituições de ensino continuam resolvendo esse problema manualmente, o que torna o processo demorado, podendo levar vários dias até ser finalizado. Para Silva (2005), como este é um problema altamente combinatório, com o aumento de turmas nas instituições de ensino a resolução manual desse problema pode demandar muito tempo e esforço e ainda não garantir que a solução obtida alcance o valor ótimo para o objetivo almejado.

O Problema de Alocação de Salas (PAS), também chamado de Problema de Alocação de Aulas a Salas (PAAS), pode ser considerado um caso especial do problema de *educational timetabling*, que envolve todos problemas no qual há a necessidade de alocar eficientemente os recursos educacionais em determinados horários. Nesse contexto, o PAS consiste em alocar aulas de disciplinas àquelas salas de aula que possam receber estas turmas, respeitando suas capacidades e choques de horário. Estas aulas já possuem horário de início e término previamente determinados (COSTA, 2013).

Na literatura, o PAS é apresentado como um problema pertencente à classe NP-Difícil, e para essa classe de problemas, obter uma solução ótima, em um período de tempo aceitável, não é uma tarefa trivial nem mesmo para sistemas computacionais de grande porte (SUBRAMANIAN *et al.*, 2006).

Apesar da complexidade desse tipo de problema, vários modelos matemáticos vêm sendo desenvolvidos e utilizados para a sua resolução. Phillips *et al.* (2015), mostra que existem muitas formulações matemáticas que podem considerar diversos critérios de avaliação da solução, como por exemplo: maximização do número de aulas alocadas; maximização do número de horas/aula cursadas pelos alunos; maximização da utilização

das salas; maximização das preferências associadas à alocação de um dado curso a uma sala; minimização do número de aulas de um mesmo curso alocadas em salas diferentes, entre outras.

Conforme apresentado por Prado (2014), o PAS é um problema de difícil generalização, pois ele pode variar de acordo com os regimes educacionais, que variam de região para região, bem como devido à características particulares de cada instituição. Desta forma, geralmente, os sistemas computacionais desenvolvidos para a solução desse problema visam atender a uma instituição específica.

Outro problema que leva a difícil generalização do problema é que, a depender das restrições de cada local, a sua resolução pode ser obtida através de algoritmos de programação linear, com tempo polinomial, ou então algoritmos baseados em técnicas heurísticas devido ao seu elevado custo computacional.

Desta forma, esta pesquisa se propõe a desenvolver um modelo matemático para resolução do PAS no Centro de Informática (CI) da Universidade Federal da Paraíba (UFPB), bem como criar um sistema informatizado no qual poderá ser utilizado pelos responsáveis por esta atividade para que eles possam resolver o problema de alocação de salas de forma mais rápida e eficiente.

### 1.1.1 Objetivo geral

Este trabalho tem como objetivo geral o desenvolvimento de um sistema informatizado que utilize uma abordagem de programação linear inteira para otimizar o processo de alocação de salas de aulas no Centro de Informática da UFPB.

### 1.1.2 Objetivos específicos

Para que o objetivo geral seja alcançado, foram determinados os seguintes objetivos específicos:

- Minimizar o tempo atualmente gasto para se resolver este problema no Centro de Informática da UFPB.
- Buscar utilizar restrições reais para que o modelo desenvolvido possa se aproximar da realidade local.
- Procurar gerar soluções que resolvam o atual problema as vezes necessita alocar turmas à salas com capacidade inferior à demanda da turma.
- Facilitar a criação de diversos cenários de alocação para facilitar à gestão do centro na identificação de e solução dos problemas.

### 1.1.3 Justificativa

Como já apresentado, visto que o Problema de Alocação de Salas é um problema combinatório pertencente à classe de problemas NP-Difícil, uma solução manual demanda muito esforço e tempo, inviabilizando sua solução sem a utilização de alguma ferramenta computacional.

Outro fator inerente a este problema é a dificuldade de generalização, visto que cada instituição de ensino possui suas especificidades. Esta característica faz com que haja a necessidade de que os pesquisadores que estudam este tema desenvolvam modelos e sistemas específicos para cada instituição de ensino.

Segundo Cirino (2016), existem certas características que se repetem em vários estudos, são elas: Redução de Deslocamento (de docentes e de discentes); Estabilidade de Sala, onde tenta-se manter em determinadas salas um conjunto previamente especificado de aulas; Restrição de Utilização de Salas, minimizando a subutilização dos recursos das salas ou alocando turmas apenas àquelas salas que possuem os recursos necessários; Melhor Utilização da Capacidade da Sala, visando a redução do número de assentos livres nas salas; e Atendimento de Preferências por Salas, que podem incluir tanto as preferências de docentes por salas quanto preferências de discentes.

Ainda segundo Cirino (2016), as principais abordagens de solução para o problema são: uso da meta-heurística Recozimento Simulado (*Simulated Annealing*) e da meta-heurística Busca Tabu (*Tabu Search*). Na literatura também são encontradas abordagens menos usuais, tais como: Coloração de Grafos; heurísticas construtivas e técnicas de geração de colunas.

Devido às características e necessidades reais do Centro de Informártica da UFPB, foi possível utilizar uma abordagem de programação linear inteira para modelar e resolver este problema em um tempo computacional viável.

# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1 *Timetabling*

Schaerf (1999) define o problema de *Timetabling* como um problema de agendamento de uma sequência de aulas entre professores e alunos em um período de tempo prefixado e que satisfaçam uma série de diferentes restrições.

Devido à sua importância e aplicabilidade em instituições de ensino tanto superior quanto básico, esse problema começou a ser estudado por Gotlieb em 1963 e desde essa época novas formas de resolução vem sendo pesquisadas (SCHAERF, 1999).

Na literatura da Pesquisa Operacional, o problema de *Timetabling*, consiste na alocação eficiente de recursos em determinados horários e por diferentes pessoas. Para Qu *et. al.* (2009), esse tipo de problema pode aparecer em várias formas, tais quais: *educational timetabling*, *nurse scheduling*, *sports timetabling*, *transportation timetabling*, entre outros.

Genericamente, o problema de *Timetabling* é usualmente definido como um problema que visa organizar certo número de eventos (exames, cursos, encontros) em um limitado número de períodos de tempo, satisfazendo o maior número de restrições requeridas pela realidade (POULSEN, 2012).

Saviniec (2017) divide as restrições do problema de *Timetabling* em dois grupos: restrições *hard* (obrigatórias) e restrições *soft* (desejáveis). Um resultado que satisfaça todas as restrições *hard* é denominada como uma solução viável. Enquanto isso, uma solução que, ao mesmo tempo que ela satisfaça todas as restrições *hard*, também minimize o número de restrições *soft*, será chamada de solução ótima.

De forma geral, Corne (1994) define o problema de *Timetabling* utilizando quatro conjuntos finitos:

- $E = \{e_1, e_2, ..., e_v\}$ eventos (por exemplo, provas, aulas, encontros)
- $T = \{t_1, t_2, ..., t_s\}$ possíveis horários de início para esses eventos
- $P = \{p_1, p_2, ..., p_n\}$ locais onde esses eventos podem ser realizados
- $A = \{a_1, a_2, ..., a_m\}$ pessoas que possuem atribuições para desempenhar determinados eventos (por exemplo, professores, tutores, monitores)

Cada evento $e_\mathrm{i}$ pode ser considerado como um par ordenado $e_i = (e_i^l, e_i^s)$, onde $e_i^l$ é a duração do evento $e_i$ (ex. em minutos), e $e_i^s$ é o seu tamanho (ex. se $e_i$ é uma avaliação, $e_i^s$ deve ser o número de estudantes que irão realizar essa avaliação). Além disso, cada

local pode ser considerado como um par ordenado $p_i = (p_i^e, p_i^s)$, onde $p_i^e$ é o número de eventos diferentes que podem ocorrer simultaneamente neste local, e $p_i^s$ é o seu tamanho (por exemplo: o número total de alunos que pode suportar). Há também uma matriz $(D)$ de tamanho $n \times n$ dos tempos de viagem entre cada par de lugares.

Desta forma, uma alocação é uma tupla $(a, b, c, d)$, onde $a \in E$, $b \in T$, $c \in P$ e $d \in A$. Podendo ser interpretada como "um evento $a$, que se inicia no tempo $b$, no local $c$ e é realizado pela pessoa $d$".

O foco deste trabalho será o Problema de Alocação de Salas (*Classroom Assignment Problem*) que pode ser apresentado como um subproblema do *Course Timetabling* e consiste no agendamento de disciplinas a determinadas salas dado um horário já estabelecido. Este problema será abordado em detalhes na seção 3. A Figura 1 apresenta as subdivisões do problema de *Timetabling* apresentadas por Sales (2015a).

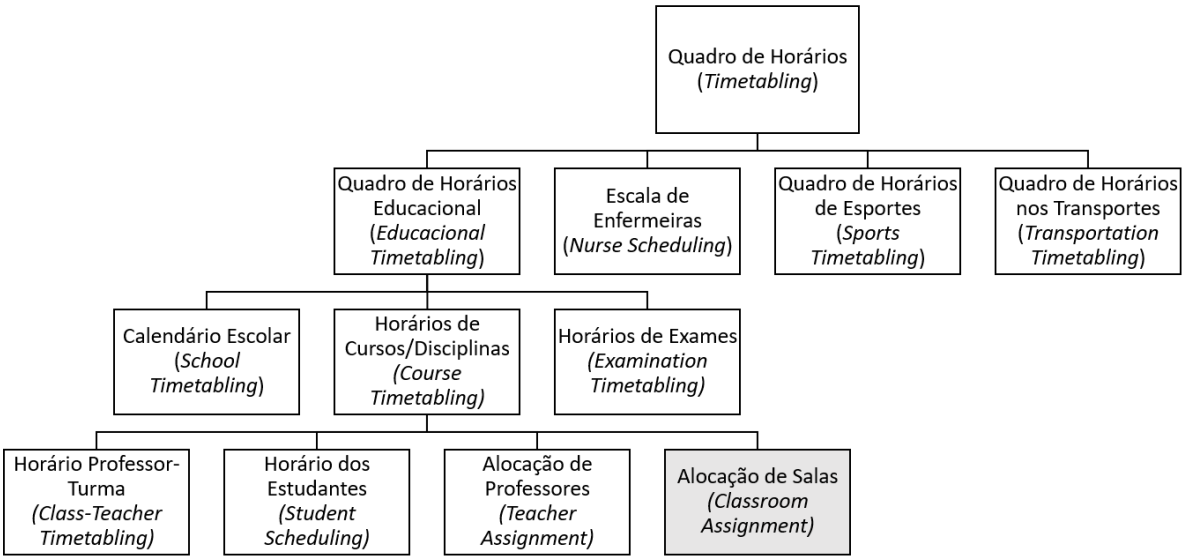


**Figura 1: Subdivisões do *Timetabling***
***Fonte: Sales (2015a)***

## 2.2 Otimização Combinatória e Programação Linear

Os problemas de *timetabling*, podem ser resolvidos utilizando técnicas de otimização combinatória e de programação linear. Entende-se por otimização, o processo de encontrar a melhor solução, também chamada de solução ótima, para um determinado problema.

Nos problemas de otimização combinatória a meta é encontrar soluções que representem o menor (para problemas de minimização) ou o maior (para problemas de maximização) valor possível em uma região viável (respeitando as restrições do problema), alcançando assim o chamado ótimo global. Entretanto, na prática, o que geralmente

ocorre é a aparição de muitos ótimos locais no espaço de busca. Estes pontos podem atrapalhar a busca, podendo até impedir que a solução ótima global possa ser obtida (Barbosa, 2011).

Um ótimo global é aquele ponto no qual o melhor valor da função objetivo, dentre todos os locais existentes no espaço de busca, é encontrado. Nos problemas de otimização combinatória, ótimos locais são comumente encontrados, desta forma, a busca de um ótimo global é dificultada e muitas vezes, são necessárias técnicas para escapar de ótimos locais e encontrar o ótimo global.

Para se resolver problemas de otimização combinatória podem ser usados métodos exatos, métodos heurísticos ou uma combinação dos dois métodos que é conhecido como *matheuristic*. Segundo Souza (2005), problemas de otimização combinatória são classificados como NP-Difícil, desta forma, não existe ainda um algoritmo que consiga provar a otimalidade de uma solução em tempo polinomial.

Devido a este fato, casos práticos desses problemas são resolvidos normalmente por meio de métodos heurísticos. Segundo Blum *et. al.* (2010), essa escolha ocorre visto que esses métodos produzem, em geral, soluções de boa qualidade em tempo computacional viável, sendo essas soluções consideradas suficientes para a maioria das situações reais.

## 2.3 Heurísticas e Metaheurísticas

Devido às suas características e complexidade computacional, os Problemas de *timetabling* dificilmente são resolvidos com métodos exatos, desta forma, uma das possibilidades de resolução se dá por meio de métodos heurísticos e metaheurísticos.

Métodos heurísticos são muito úteis para dar um resultado a problemas difíceis e que não podem ser resolvidos apropriadamente, em tempo aceitável, de forma exata. Como é o caso dos problemas combinatórios, que possuem diversas restrições em termo de complexidade computacional, e geralmente são resolvidos por meio de técnicas heurísticas (COLIN, 2007).

Para Hillier & Lieberman (2013), método heurístico é um procedimento que certamente conseguirá encontrar uma excelente solução viável para determinado problema, no entanto, essa solução não será necessariamente uma solução ótima. Mesmo sem garantir a otimalidade das soluções, as heurísticas podem gerar soluções que ao menos estejam próximas da solução ótima.

A principal vantagem das heurísticas é o fato dela conseguir obter boas soluções viáveis em um curto espaço de tempo e com um baixo custo computacional.

Uma metaheurística é geralmente aplicada para encontrar solução em problemas onde se tem pouca informação, por exemplo: situações onde a aparência da solução ótima

é desconhecida; onde há pouca informação heurística; e nas situações onde a força-bruta é desconsiderada devido ao tamanho do espaço de soluções (LUZIA, 2009).

Originalmente, de acordo com Glover (2003), metaheurísticas são definidas como métodos de solução que envolvem procedimentos de buscas locais com outras estratégias a fim de que seja possível escapar de mínimos locais para conseguir realizar uma busca robusta no espaço de soluções de um determinado problema.

Para Luke (2013), esta é uma área considerada como um subcampo primário da área de otimização estocástica, que engloba algoritmos e técnicas que necessitam de algum grau de aleatoriadade para que seja possível encontrar soluções tão ótimas quanto possível para diversos problemas reconhecidamente difíceis.

Luzia (2009) mostra que com o passar do tempo, a definição de metaheurística passou a abranger quaisquer procedimentos utilizados para escapar de mínimos locais em espaços de soluções de buscas locais. Sendo assim, esses métodos visam a obtenção de resultados satisfatórios de um problema, mas sem garantir a otimalidade destes resultados.

Atualmente, diversas metaheurísticas podem ser encontradas na literatura, dentre elas, as principais podem ser visualizadas no Quadro 1.

| **Metaheurística** | **Autor** |
|---|---|
| Algoritmos Genéticos | Goldberg, 1989 |
| Algoritmos Meméticos | Moscato, 1989 |
| *Ant Colony Systems* (Colônia de Formigas) | Dorigo, 1992 |
| *Simulated Annealing* | Kirkpatric, Gellati & Vecchi, 1983 |
| *Tabu Search* (Busca Tabu) | Glover, 1986 |
| *GRASP (Greedy Randomized Adptative Procedure)* | Feo & Resende, 1994 |
| *Variable Neighborhood Search* (VNS) e *Variable Neighborhood Descent* (VND) | Mladenovic e Hansen, 1997, 1999 |

**Quadro 1: Principais métodos heurísticos e seus autores.**

Fonte: Sales (2015)

## 2.4 Matheurística

As matheurísticas são algoritmos híbridos que combinam metaheurísticas e técnicas de programação matemática. A matheurística combina as vantagens de se tirar proveito das estruturas específicas dos problemas e combiná-las com o uso de *solvers* matemáticos de alto desempenho (QUITO, 2018).

Com o uso da matheurística, um problema pode ser divido em partes. Partindo de uma solução gerada por métodos heurísticos, pode-se isolar parte desta solução e de forma exata resolver essa parcela do problema. Desta forma, essa nova solução pode ser incorporada à solução anteriormente gerada de forma heurística gerando assim novas soluções.

Molencrush (2017) afirma que apesar de possuir excelentes resultados, as matheurísticas ainda são relativamente escassas na literatura.

## 2.5 Problema de Alocação de Aulas

### 2.5.1 Definição e variações do PAS

O Problema de Alocação de Salas é um problema que é observado em diversas instituições de ensino superior em todo o mundo. Pillay (2014) o aborda como parte integrante do Problema de Alocação de Cursos Universitários. Este último problema envolve a alocação de professores às turmas e as turmas às salas de aula. Como na maioria dos casos, são departamentos ou coordenações diferentes que realizam cada uma dessas alocações, o foco dessa pesquisa foi na alocação das turmas, com horários preestabelecidos, às salas de aula.

Com o crescente aumento no número de disciplinas ofertadas, cursos e alunos matriculados nas instituições de ensino, esse processo de distribuição de turmas à salas vem se tornando cada vez mais complexo. O fato de que muitas vezes a infraestrutura das instituições não acompanham esse crescimento, leva esses locais a terem cada vez mais problemas para realizar esta alocação de modo eficiente.

Embora estejamos vivenciando um grande avanço computacional nos últimos anos, muitas instituições ainda realizam a alocação de salas de forma manual, ocasionando demora, elevado esforço pessoal e além disso sem a garantia da viabilidade das soluções encontradas (MARTÍNEZ, 1998; SOUZA *et. al.*, 2002; SUBRAMANIAN *et. al.*, 2006; SUBRAMANIAN *et. al.*, 2011).

Conforme pode ser visto no Quadro 2, Cirino (2016) apresentou as principais publicações que tratam do PAS sob diferentes métricas de qualidade até o ano de 2015. Essas métricas podem ser definidas da seguinte forma:

- Redução de deslocamento ($M_1$)
  - Nestes casos, busca-se a minimização dos deslocamentos. Nestes deslocamentos podem ser considerados tanto os deslocamentos dos alunos quanto os deslocamentos dos professores.
- Estabilidade de salas ($M_2$)
  - Aqui o objetivo é tentar manter em um conjunto predeterminado de salas, um conjunto específico de aulas. Por exemplo, todas as aulas de cálculo e física devem estar em um conjunto específico de salas.
- Restrição de utilização de salas ($M_3$)
  - Usado quando se pretende minimizar a subutilização dos recursos das salas, ou em situações onde a sala de aula é compartilhada com outros cursos ou atividades, inviabilizando o uso das mesmas em determinados horários.
- Melhor utilização da capacidade das salas ($M_4$)
  - Métrica que visa a minimização da quantidade de assentos livres, evitando assim que salas demasiadamente grandes sejam utilizadas em turmas com poucos alunos.
- Atendimento de preferências por salas ($M_5$)
  - Nestes modelos, tanto alunos quanto professores podem ter preferências por aulas em determinadas salas. Essas preferências podem ser de acordo com critérios, como acessibilidade, disponibilidade de determinados recursos multimídia, entre outros.

| Publicação | Métricas | | | | |
|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ |
| Mulvey *et. al.* (1982) | | | | • | |
| Waterer (1995) | | | | • | • |
| Martinez-Alfaro e Flores-Teran (1998) | • | | | • | |
| Souza *et. al.* (2002a) | | • | • | • | |
| Souza *et. al.* (2002b) | • | • | • | • | |
| Nascimento *et. al.* (2005) | | • | • | | |
| Silva e Silva (2009) | | • | • | • | |
| Constantino *et. al.* (2009) | • | • | | | • |
| Kripka e Kripka (2010) | • | | | | |
| Subramanian *et. al.* (2010) | | • | • | | • |
| Cirino *et. al.* (2013) | | • | | • | |
| Prado e Souza (2014) | • | • | • | • | • |
| Souza e Scarpin (2014) | | • | • | • | • |
| Elloumi *et. al.* (2014) | | | | • | |
| Phillips *et. al.* (2015) | | • | • | | • |
| Cirino (2016) | • | • | • | • | • |

**Quadro 2: Métricas de qualidade para o PAS encontradas na literatura**

Fonte: Adaptado de Cirino (2016)

### 2.5.2 Complexidade Computacional

Segundo Carter (1992), o Problema de Alocação de Salas é considerado um problema clássico de otimização combinatória contida na classe de problemas NP-Difícil. Para estes tipos de problemas, obter uma solução ótima, em um tempo aceitável, pode ser uma atividade complexa.

Para Colin (2007), a complexidade computacional de uma solução é o parâmetro que define o grau de dificuldade com que um problema é resolvido. A complexidade computacional pode ser definida como uma medida de tempo de solução, de memória necessária para se utilizar um determinado algoritmo, essas medidas de análise devem ser independentes da tecnologia de *hardware* e *software* utilizadas.

De acordo com Garey (1979), conforme citado por Sales (2015a), segundo a teoria da complexidade computacional, os problemas de otimização podem ser classificados em P, NP e NP-Difícil. A mesma autora ressalta ainda que os problemas que estão na classe P, são considerados "tratáveis computacionalmente, já os demais são considerados intratáveis", isso quer dizer que "o número de computações executadas no melhor algoritmo conhecido cresce exponencialmente em função do tamanho da instância".

Goodrich (2008) e Colin (2007) definem essas classes de problemas da seguinte forma:

- Problemas da Classe P (*Polinomial Time*):
  - Nessa classe se encontram os problemas em que há algum algoritmo conhecido que forneça soluções por meio de uma função polinomial. Esse problemas são considerados de fácil solução.
- Problemas da Classe NP (*NonDeterministic Polinomial Time*):
  - Aqui se encontram os problemas que podem ser resolvidos por algoritmos de tempo polinomial em uma máquina não determinística. Nestes problemas, os algoritmos de solução são baseados em enumeração, e de maneira geral, o número de possibilidades é significativamente grande, fazendo com que esses tipos de algoritmos não consigam resolver grandes instâncias em um tempo hábil.
- Problemas NP-Completos
  - NP-completo é um subconjunto de NP, cujas soluções podem ser verificadas em tempo polinomial, porém esses problemas de decisão só podem ser resolvidos de forma ótima apenas por algoritmos de tempo exponencial.
- Problemas NP-Difíceis
  - São os problemas de otimização cujos problemas de decisão correspondentes são NP-Completos

Desta forma, Colin (2007) afirma que quando um problema é classificado como NP-Difícil, certamente a solução ótima não será encontrada em tempo hábil. Como exemplo, os problemas do caixeiro viajante, o problema geral de roteamento de veículos, o problema de particionamento, o problema de número cromático em grafos, o problema da mochila e o Problema de Alocação de Salas, entre outros, são conhecidamente classificados como NP-Difícil (FILHO, 2008).

### 2.5.3 Métodos de resolução do problema

Na literatura, as abordagens são as mais diversas possíveis, Constantino *et. al.* (2010) destaca os seguintes usos: meta-heurística Recozimento Simulado (*Simulated Annealing* ), Meta-heurística Busca Tabu (*Tabu Search*); Coloração de Grafos; Heurísticas construtivas; e Técnicas de Geração de Colunas para solução de modelo inteiro.

Além desses, outros métodos de resolução vêm sendo adotados para a resolução do PAS. Cirino (2015) e Theodoro (2016) fizeram uso de Algoritmos Genéticos, enquanto que Alarcão (2015) modelou o problema como um grafo bipartido valorado e utilizou a ideia do algoritmo Húngaro para solucionar o PAS e DIAS (2018) resolveu o problema por meio de programação linear modelado como problema de designação.

Pode-se observar as mais diversas formas de se resolver o Problema de Alocação de Salas, a melhor forma vai depender de cada situação. Um estudo realizado por Carter (1992), mostra a complexidade de se resolver o PAS de acordo com o objetivo pretendido. Por exemplo, quando o modelo busca apenas resultados viáveis, ou satisfatórios para as restrições locais, então possivelmente o problema poderá ser resolvido por meio de programação linear e facilmente poderá ser resolvido. Por outro lado, se o objetivo a ser alcançado é de otimização, com uma função objetivo de minimização de custos, e cada aula possua um custo associado ao ser alocada em uma sala, a resolução deste problema torna-se mais complexa, computacionalmente custosa e possivelmente serão necessários métodos heurísticos para a sua solução.

No caso do presente estudo de caso, um modelo de programação linear inteira foi utilizado para auxiliar o Centro de Informática a resolver o problema de alocação de suas aulas.

# 3 DESCRIÇÃO DO PROBLEMA

## 3.1 Histórico do Centro de Informática (CI)

O Centro de Informática teve início em dezembro de 2011. Este centro foi criado a partir do Departamento de Informática que fazia parte do Centro de Ciências Exatas e da Natureza (CCEN) objetivando a criação de novos cursos visando atender as metas definidas para o Programa de Expansão Universitária - REUNI/UFPB.

Atualmente o CI abriga três departamentos, quatro cursos de graduação e três cursos de pós-graduação. Os Departamentos são: Departamento de Informática, Departamento de Sistemas de Computação e Departamento de Computação Científica. Os cursos de graduação são: Ciência da Computação (CC), Engenharia da Computação (EC), Matemática Computacional (MC) e Licenciatura em Computação (à distância). Os cursos de pós graduação são: Pós-graduação em Informática (PGI), Pós-graduação em Modelagem Matemática Computacional (PGMMC) e Pós-graduação em Computação, Comunicação e Artes (PGCCA). Os cursos podem ser melhor visualizados na Figura 2.

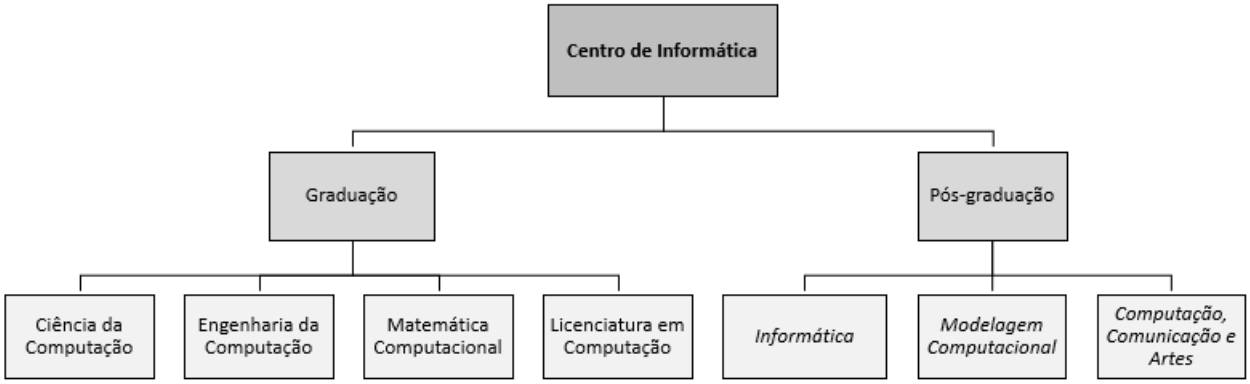


**Figura 2: Cursos de Graduação e Pós-graduação no CI**
**Fonte: O autor (2019)**

Atualmente o CI conta com uma infraestrutura de um prédio com quatro (4) pavimentos. Para as aulas, há dezessete (17) salas, sendo um (1) auditório, quatro (4) laboratórios e doze (12) salas de aula. As características de cada sala podem ser observadas no Quadro 3.

Foi observado, que estas salas não estão sendo suficientes para alocar todas as disciplinas do CI. Nos períodos de 2017.2 e 2018.1, houve a necessidade de fazer uso de mais quatro (4) e seis (6) salas, respectivamente, de outros prédios, como o LIEP (Laboratório Interdisciplinar de Ensino, Pesquisa e Extensão) e o CTDR (Centro de Tecnologia e Desenvolvimento Regional).

| Nome | Tipo | Qtd. de carteiras | Qtd. de cadeiras acolchoadas | Mesas | Qtd. de computadores | Projetor | Ar condicionado | Quadro de vidro |
|---|---|---|---|---|---|---|---|---|
| T-05 | auditório | 60 | 4 | - | - | Não | 1 | Sim |
| CI-101 | laboratório | 18 | 21 | 19 | 18 | Não | 2 | Sim |
| CI-105 | laboratório | 13 | 20 | 6 | 18 | Sim | - | Sim |
| CI-106 | laboratório | 31 | 24 | 4 | 24 | Sim | 2 | Sim |
| T-07 | laboratório | 6 | 59 | 13 | 57 | Não | 3 | Sim |
| CI-102 | sala de aula | 54 | 1 | - | - | Sim | 2 | Sim |
| CI-103 | sala de aula | 58 | 2 | - | - | Sim | 2 | Não |
| CI-104 | sala de aula | 26 | - | - | - | Sim | 1 | Sim |
| CI-107 | sala de aula | 28 | 1 | - | - | Sim | - | Lousa |
| CI-108 | sala de aula | 28 | 1 | - | - | Não | 2 | Lousa |
| CI-109 | sala de aula | 35 | - | - | - | Sim | 1 | Lousa |
| CI-304 | sala de aula | 66 | - | - | - | Sim | 2 | Lousa |
| CI-306 | sala de aula | 27 | - | - | - | Sim | 1 | Lousa |
| CI-307 | sala de aula | 27 | 1 | - | - | Sim | 1 | Não |
| CI-308 | sala de aula | 28 | - | - | - | Sim | 1 | Sim |
| CI-311 | sala de aula | 30 | - | - | - | Sim | 1 | Sim |
| SB-01 | sala de aula | 55 | - | - | - | Sim | 1 | Sim |

**Quadro 3: Características das salas de aula, laboratórios e auditório do CI**

Fonte: O autor (2019)

Para este estudo, serão considerados apenas os cursos de graduação, uma vez que os cursos de pós graduação possuem duas salas específicas para suas aulas. No entanto, ocorre que devido aos cursos de pós graduação não utilizarem essas salas em tempo integral, em algumas ocasiões essas salas também são utilizadas para os cursos de graduação. Sendo assim, essas duas salas poderão aparecer no modelo com restrições de uso em determinados horários.

## 3.2 O Problema de Alocação de Salas no Centro de Informática

Atualmente a alocação de salas é feita duas vezes ao ano, geralmente inicia-se com antecedência de duas semanas antes do início de cada semestre letivo. O processo todo é realizado de forma manual, sem auxílio de nenhum ferramental computacional ou matemático. Os chefes dos três departamentos são os responsáveis por realizar esta alocação. Inicialmente eles pegam a lista de disciplinas a serem ofertadas no semestre e tentam designar cada turma a uma sala apropriada, sem que haja mais de uma aula em

uma mesma sala no mesmo horário e tentando atender às necessidades específicas de cada aula.

Um problema que acontece é que essa alocação é feita por cada departamento, então o primeiro irá realizar a alocação das suas turmas segundo suas necessidades, o próximo irá tentar alocar suas turmas nas salas disponíveis e por fim o terceiro departamento irá tentar alocar suas turmas nas salas restantes. Durante este processo os responsáveis conversam entre si para trocar salas para que se chegue em um consenso para se obter uma solução viável. O que geralmente ocorre é que a solução gerada quase sempre possui restrições que não são atendidas e que precisam ser corrigidas ao longo do período já em andamento.

Isso comprova o que foi dito por Schaerf (1999) quando afirmou que uma solução manual para esse tipo de problema requer vários dias de trabalho, e que a solução encontrada, por mais experientes que sejam os responsáveis, possivelmente será insatisfatória em algum aspecto.

Como pode ser visto, do modo que é realizado atualmente, este processo não consegue gerar uma solução ótima para o problema enfrentado semestralmente, por vezes não encontram nem mesmo uma solução viável. Atualmente 06 salas de outros prédios são utilizadas para aulas do CI, acredita-se isso aconteça devido a uma má distribuição das aulas nas salas do prédio do CI, já que no período letivo de 2017.2 as salas de aulas do CI estavam com uma taxa de ocupação de aproximadamente 70%.

## 3.3 Escopo do PAS no CI

Já foi dito anteriormente que o PAS possui diversas variações e que cada local necessita de restrições e objetivos específicos. Para se determinar quais salas são adequadas a determinadas disciplinas há uma conversa entre os professores das disciplinas e os chefes de cada departamento para que estes últimos atribuam uma sala apropriada para cada turma.

Foi observado que para se alocar as turmas às salas, existem dois tipos de necessidades, que poderão ser definidas e divididas em dois tipos de restrições: Restrições Essenciais e Restrições Não Essenciais. O primeiro tipo são restrições que ao serem violadas causam inviabilidade na solução, enquanto que o segundo tipo são restrições que representam necessidades desejáveis, mas que não causariam inviabilidade na alocação.

Desta forma, a partir de conversas com diferentes funcionários do CI para determinar as necessidades e restrições locais do centro, ficou definido que o modelo desenvolvido nesta pesquisa para o Centro de Informática da UFPB buscasse atender os seguintes requisitos:

- Requisitos Essenciais

    1. Cada turma deverá estar alocada em uma única sala de aula;
    2. Cada sala de aula só poderá ser ocupada por apenas uma turma em cada horário;
    3. Disciplinas que necessitem de laboratórios específicos deverão estar alocadas nos seus devidos locais (por exemplo: laboratório de programação, laboratório de microeletrônica);

- Requisitos Não Essenciais

    1. Professores que solicitem salas de aula normais não devem ser alocados em laboratórios ou no auditório;
    2. O auditório não deverá ser utilizado como sala de aula;
    3. Em casos de aulas em mais de um dia na semana, que estas aulas sejam na mesma sala;
    4. As aulas devem ser alocadas em salas com capacidade maior que o número de alunos matriculados;

Observando esses requisitos, vale salientar o quarto requisito não essencial. De acordo com essa restrição, há a possibilidade de que uma disciplina seja alocada a uma sala com menos carteiras que o número de alunos matriculados, isto será adotado para que se saiba onde possam estar ocorrendo inviabilidades. A modelagem desse conjunto de restrições foi realizada de modo que há uma penalização na funcão objetivo. Quando houver penalização da FO devido a essa restrição, caberá ao operador decidir se haverá a possibilidade de se colocar carteiras extras na sala ou se será necessário gerar uma nova solução que não viole esse requisito.

Embora a análise feita por Carter (1992), que mostra que um modelo que atenda a esses requisitos seria enquadrado na classe de problemas NP-Difícil, foi observado que o modelo desenvolvido foi capaz de resolver as instâncias reais do CI de forma exata. Desta forma, a modelagem de programação linear inteira utilizada nessa pesquisa será apresentada em detalhes a seguir.

## 3.4 Formulação matemática inicial

A modelagem matemática inicial foi baseada na modelagem proposta por Costa (2013) que visa a estabilidade das salas. Ela tem como função objetivo a minimização da quantidade de aulas de uma mesma turma alocadas em salas diferentes.

Esta abordagem visa a obtenção uma solução viável para o problema, sem que haja a necessidade de que essa solução seja uma solução ótima. Nestas circunstâncias, o PAS pode ser resolvido de modo exato, por meio de uma programação linear inteira.

Nesta abordagem, o modelo precisa atender a três grupos de restrições que viabilizam a solução encontrada, são elas:

- Todas as aulas de uma turma devem ser alocadas em uma única sala.
- A quantidade de alunos matriculados não pode ser maior que a quantidade de carteiras disponíveis na sala.
- Duas aulas não podem ser alocadas em uma mesma sala, no mesmo dia e horário.

Este modelo pode ser representado como um modelo de programação linear inteira conforme pode ser visto a seguir:

**Notação:**

*Conjuntos*

| | |
|---|---|
| $T$ | Conjunto das turmas $t$ oferecidas em cada período |
| $S$ | Conjunto das salas $j$ disponíveis em cada período |
| $H_t$ | Conjunto das aulas (horários) $i$ de cada turma $t$ |
| $TChH_i$ | Conjunto das aulas que possuem choque de horário com $i$ |

*Dados de Entrada*

| | |
|---|---|
| $Q_i$ | Capacidade da sala $j$ |
| $D_t$ | Demanda da turma $t$ |

*Variáveis de Decisão*

$$x_{ij} = \begin{cases} 1, \text{ se a aula } i \text{ da turma } t \text{ for alocada na sala } j \\ 0, \text{ caso contrário} \end{cases}$$

$$y_t = \begin{cases} 1, \text{ se as aulas da turma } t \text{ não forem alocadas na mesma sala} \\ 0, \text{ caso contrário} \end{cases}$$

**Modelo:**

$$F.O. \quad \min \sum_{t \in T} y_t \tag{1}$$

$$\text{S.A} \quad \sum_{j \in S} x_{ij} = 1 \qquad \forall i \in H_t | t \in T \tag{2}$$

$$x_{ij} + x_{kj} \leq 1 \qquad \forall j \in S, \forall i \in H_t | t \in T, \forall k \in TChH_i \tag{3}$$

$$x_{ij} D_t \leq Q_j \qquad \forall j \in S, \forall i \in H_t \,|\, t \in T \tag{4}$$

$$y_t \geq |x_{ij} - x_{kj}| \qquad \forall t \in T \,|\, \forall i, k \in H_t, \forall j \in S \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in P, \forall j \in T_i \tag{6}$$

$$y_t \in \{0, 1\} \qquad \forall t \in T \tag{7}$$

Neste modelo, a Função Objetivo (FO) (1) tem o intuito de minimizar a quantidade de aulas de uma mesma turma alocadas em salas diferentes. As restrições (2) garantem que cada uma das aulas oferecidas no período estarão alocadas em apenas uma sala. O conjunto de restrições (3) evita os choques de horário, impedindo que duas ou mais aulas sejam alocadas em uma mesma sala, no mesmo dia e horário. Por meio das restrições (4), uma aula só poderá ser alocada a salas que tenham capacidade superior ou igual à necessidade demandada pela aula. As restrições (5) são necessárias para relacionar as variáveis de decisão utilizadas no modelo. Por fim, os dois últimos grupos de restrições (6 e 7) estão para definir a natureza binária das variáveis.

# 4 PROCEDIMENTOS METODOLÓGICOS

Para dar início à execução deste trabalho, foi realizada uma pesquisa exploratória. Este tipo de pesquisa teve como objetivo o aprimoramento de ideias e descobertas de informações. Conceitos relacionados diretos e indiretamente ao Problema de Alocação de Salas, seus subproblemas, generalizações, formas de resolução e sistemas atualmente existentes para a resolução do PAS em diversas situações foram estudadas para que ideias e conceitos pudessem ser utilizados e adaptados na segunda fase do projeto.

Devido às características do PAS, o método de estudo abordado teve como fundamento os preceitos desenvolvidos na Pesquisa Operacional (PO). De acordo com Goldbarg (2005), os modelos de PO seguem uma estrutura lógica que são auxiliados por um ferramental matemático de representação, com o intuito de apresentar melhores condições de funcionamento para os sistemas a serem estudados. Em sua maioria, os modelos de PO são denominados de Programação Matemática, que são uma das principais variedades dos modelos quantitativos.

Durante a segunda fase da pesquisa, houve a coleta de dados junto ao CI para criação da instância inicial a ser resolvida. Através do SIGAA (Sistema Integrado de Gestão de Atividades Acadêmicas), que é o sistema oficial de gerenciamento de atividades utilizado pela UFPB, foram coletados dados referentes às alocações de salas realizadas nos períodos letivos de 2017.2, 2018.1 e 2019.1. Dentre os dados coletados constam: lista de disciplinas ofertadas pelos cursos de graduação em Matemática Computacional (MC), Ciências da Computação (CC) e Engenharia da Computação (EC); lista de disciplinas ofertadas pelos cursos de Pós-graduação em Informática (PGI), Pós-graduação em Modelagem Matemática Computacional (PGMMC) e Pós-graduação em Computação, Comunicação e Artes (PGCCA); quantidade de vagas ofertadas e de alunos matriculados em cada turma; horários das aulas; e locais de realização de cada turma.

Esses dados puderam ser utilizados como base para comparação entre a alocação realizada manualmente com os resultados obtidos a através do sistema desenvolvido nesta pesquisa.

O objetivo fundamental dessa fase é a escolha das principais modelagens e formas de resolução para o problema, buscando escolher aquela que consegue obter os melhores resultados para as instâncias estudadas. Parâmetros como tempo computacional para gerar uma solução, percentual de utilização das salas e quantidade de restrições não violadas foram analisados como indicadores na escolha da melhor modelagem.

É válido ressaltar que modelos matemáticos se assemelham à realidade, mas não são cópias fidedignas. Os modelos buscam descrever, relatar e resolver o problema real da melhor forma possível, mas nem sempre é possível que exista aferições que garantam

a certeza absoluta aos resultados. O que pode ser garantido é a obtenção de soluções viáveis ao problema modelado.

Durante a fase seguinte, houve o desenvolvimento e implementação dos modelos e técnicas de resolução escolhidos. Para garantir que os modelos estivessem operando da forma esperada, houve uma fase de validação dos modelos. Nesta etapa, o objetivo foi a verificação da adequação das modelagens adotadas em relação aos problemas reais da instituição. Os testes puderam ser realizados fazendo uso de instâncias pequenas e em seguida analisando os resultados obtidos. Os casos que não obtiveram resultados satisfatórios foram analisados para resolução do problema, bem como para verificar a sua eficácia, sendo decidido então a sua continuidade ou descarte, em casos que outras formas de resolução estiverem obtendo resultados significativamente melhores.

Após isto ser feito, durante a última fase do projeto, o sistema foi desenvolvido e testado visando a adoção de uma interface simples e de fácil utilização pelo usuário para que pudesse haver pouca resistência ao uso do sistema proposto nesta pesquisa. Foi utilizado uma planilha de Excel que possibilita uma uma fácil inserção e alteração dos dados das disciplinas e salas, e permitindo que haja um pré-processamento dos dados antes que o modelo resolva o problema.

# 5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

## 5.1 Modelo proposto

A modelagem desenvolvida nesta pesquisa buscou atender às principais necessidades observadas no Centro de Informática da Universidade Federal da Paraíba para auxiliar o processo de alocação semestral das disciplinas às suas respectivas salas de aula. Foram tomados como objetivos a estabilidade de uma disciplina com aulas em mais de um dia na semana ser alocada em uma única sala de aula e foi levado em consideração às preferencias de determinadas disciplinas estarem alocadas em um conjunto específico de salas. Outro ponto a ser salientado, é o fato de que foi pedido que o modelo pudesse retornar soluções que violassem a restrição de capacidade das salas, para que o tomador de decisões observasse a viabilidade da insersão de algumas carteiras a mais nessas salas.

Desta forma, o problema foi modelado matematicamente como um modelo de programação linear inteira conforme pode ser visto a seguir:

**Notação:**

*Conjuntos*

| | |
|---|---|
| $T$ | Conjunto das turmas $t$ oferecidas em cada período |
| $S$ | Conjunto das salas $j$ disponíveis em cada período |
| $H_t$ | Conjunto das aulas (horários) $i$ de cada turma $t$ |
| $TChH_i$ | Conjunto das aulas que possuem choque de horário com $i$ |

*Dados de Entrada*

| | |
|---|---|
| $Q_j$ | Capacidade da sala $j$ |
| $D_t$ | Demanda da turma $t$ |
| $C_{ij}$ | Custo da alocação da aula $i$ na sala $j$ |
| $P$ | Penalização da FO para cada restrição de capacidade que for violada |

*Variáveis de Decisão*

$$x_{ij} = \begin{cases} 1, \text{ se a aula } i \text{ da turma } t \text{ for alocada na sala } j \\ 0, \text{ caso contrário} \end{cases}$$

$$y_t = \begin{cases} 1, \text{ se as aulas da turma } t \text{ não forem alocadas na mesma sala} \\ 0, \text{ caso contrário} \end{cases}$$

$$z_{ij} = \begin{cases} 1, \text{ se a aula } i \text{ da turma } t \text{ for alocada na sala } j \text{ , caso } D_i > Q_j \\ 0, \text{ caso contrário} \end{cases}$$

**Modelo:**

$$F.O. \quad \min \quad \sum_{t \in T} y_t \quad + \quad \sum_{i \in H_t | t \in T} \sum_{j \in S} C_{ij} x_{ij} + z_{ij} P \tag{8}$$

$$\text{s.a} \quad \sum_{j \in S} x_{ij} = 1 \qquad \forall i \in H_t | t \in T \tag{9}$$

$$x_{ij} + x_{kj} \leq 1 \qquad \forall j \in S, \forall i \in H_t | t \in T, \forall k \in TChH_i \tag{10}$$

$$y_t \geq |x_{ij} - x_{kj}| \qquad \forall t \in T \,|\, \forall i, k \in H_t, \forall j \in S \tag{11}$$

$$x_{ij} D_t - z_{ij} M \leq Q_j \qquad \forall j \in S, \forall i \in H_t \,|\, t \in T \tag{12}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in P, \forall j \in T_i \tag{13}$$

$$z_{ij} \in \{0, 1\} \qquad \forall i \in P, \forall j \in T_i \tag{14}$$

$$y_t \in \{0, 1\} \qquad \forall t \in T \tag{15}$$

Neste modelo, a primeira parte da função objetivo (8) visa minimizar a quantidade de aulas de uma mesma turma alocadas em salas diferentes. Na segunda parte, para toda aula $i$ alocada na sala $j$ há um fator de custo ($C$) que dependerá da preferência desta aula estar alocada nesta sala, e nos casos em que houver excesso de demanda para a sala ($Z_{ij} = 1$), haverá um fator $P$ que irá punir a FO para que esses casos sejam evitados sempre que possível.

O conjunto de restrições (9) garante que cada uma das aulas ofertadas estarão alocadas em uma única sala. As restrições (10) garantem que não haja choques de horário, proibindo que mais de uma aula seja alocada em uma mesma sala, no mesmo dia e horário. As restrições (11) são necessárias para relacionar as variáveis de decisão utilizadas no modelo que visam a estabilidade de sala para as aulas de uma mesma turma. O grupo de restrições (12) tem como finalidade penalizar a FO sempre que houver alguma turma alocada a uma sala com capacidade inferior à demanda desta turma. Por fim, os três últimos grupos de restrições (13), (14) e (15) são necessários para definir a natureza binária das variáveis.

## 5.2 Caracterização da instância inicial utilizada

A instância utilizada foi o caso real das disciplinas ofertadas no período letivo de 2019.1 nos três cursos de graduação presenciais existentes no Centro de Informática da UFPB. No total houve a oferta de 103 disciplinas que precisaram ser alocadas nas 17 salas disponíveis no bloco do CI.

Como pode ser visualizado no Quadro 4, certos horários como as terças e quintas-feiras possuem 16 aulas sendo ministradas simultaneamente, enquanto que na sexta feira há no máximo 6 aulas sendo ministradas em um mesmo horário. Esta falta de homogeneidade nos horários das disciplinas dificulta a utilização mais eficiente das salas de aula do CI, uma vez que se houvesse a troca de horário de algumas aulas das terças ou quintas para as sextas feiras haveria a possibilidade de alocar todas as disciplinas ofertadas pelo CI em salas do próprio CI, sem a necessidade da utilização de salas de aula do LIEPE.

| | segunda | terça | quarta | quinta | sexta |
|---|---|---|---|---|---|
| 08:00 / 08:50 | 9 | 12 | 9 | 11 | 4 |
| 08:50 / 09:40 | 9 | 12 | 9 | 12 | 6 |
| 09:50 / 10:40 | 12 | 16 | 11 | 16 | 6 |
| 10:40 / 11:30 | 13 | 16 | 12 | 16 | 5 |
| 11:30 / 12:20 | 1 | 0 | 1 | 0 | 0 |
| 13:10 / 14:00 | 1 | 0 | 0 | 1 | 1 |
| 14:00 / 14:50 | 11 | 14 | 10 | 15 | 3 |
| 14:50 / 15:40 | 11 | 14 | 10 | 15 | 3 |
| 15:50 / 16:40 | 8 | 9 | 6 | 10 | 2 |
| 16:40 / 17:30 | 8 | 9 | 6 | 9 | 1 |

**Quadro 4: Quantidade de turmas ofertadas em cada horário e dia da semana.**

Outro fato a ser observado é que há 31 disciplinas com uma demanda maior ou igual a 50 alunos, e no CI só há 6 salas com essa capacidade, dentre essas 6 salas, apenas uma é laboratório de informática. Este fato é algo que deve ser levado em consideração ao se determinar o horário dessas disciplinas, de modo que elas possam ter menos choques de horário entre si para que possam ser melhor alocadas.

## 5.3 Execução do modelo

Para implementação do modelo e para facilitar o uso dos operadores finais do sistema, foram desenvolvidas planilhas de Excel para inserção e pré-processamentos dos dados. Por meio de programação em *Visual Basic for Applications* (VBA) foi possível

utilizar a biblioteca de funções UFFLP versão 3.02 para implementar o modelo de programação linear inteira proposto e que posteriormente faz uso do Coin-Cbc 2.7 como resolvedor para se obter os resultados das alocações das salas de aula.

Para execução foi utilizado um computador portátil equipado com um processador Intel (R) Core (TM) i5-2450M CPU @ 2.50GHz 2.50GHz, 4,00GB de memória RAM e Sistema operacional de 64bits.

O tempo médio de execução para se obter a solução inteira ótima das instâncias testadas foi de 300s.

## 5.4 Utilização do sistema proposto

O algoritmo desenvolvido segue os seguintes passos:

1. uso do Excel para entrada dos dados das disciplinas ofertadas (horário de aula, demanda de alunos e preferência de salas) e das salas disponíveis (tipo de sala e capacidade); preenchimento automático da matriz de choques de horário;
2. utilização do UFFLP para criação do arquivo .lp contendo a modelagem do problema;
3. resolução do modelo pelo Coin-Cbc;
4. retorno dos resultados que são processados e apresentados em formato de planilhas do Excel.

Para que isto aconteça, o operador precisará alimentar a planilha com os dados de entrada das disciplinas ofertadas e das salas disponíveis.

Na planilha o operador pode inserir, editar ou excluir as discipilnas que serão ofertadas no período letivo. É necessário que seja inserido o nome da disciplina, horário das aulas, demanda esperada de alunos (Figura 3) e quais as preferências por salas que cada disciplina demanda (Figura 4).

| Disciplinas Ofertadas | Créditos | Dia | Hor | Dia | Hor | Dia | Hor |
|---|---|---|---|---|---|---|---|
| Álgebra Linear Computacional | 4 | TERÇA | 09:50 / 10:40 | TERÇA | 10:40 / 11:30 | QUINTA | 09:50 / 10:40 |
| Banco de Dados I | 4 | TERÇA | 08:00 / 08:50 | TERÇA | 08:50 / 09:40 | QUINTA | 08:00 / 08:50 |
| Cálculo das Probabilidades I | 4 | TERÇA | 09:50 / 10:40 | TERÇA | 10:40 / 11:30 | QUINTA | 09:50 / 10:40 |
| Cálculo Diferencial e Integral I | 4 | SEGUNDA | 08:00 / 08:50 | SEGUNDA | 08:50 / 09:40 | QUARTA | 08:00 / 08:50 |
| Cálculo Diferencial e Integral II | 4 | SEGUNDA | 15:50 / 16:40 | SEGUNDA | 16:40 / 17:30 | QUARTA | 15:50 / 16:40 |
| Cálculo Diferencial e Integral III (turma 1) | 4 | SEGUNDA | 15:50 / 16:40 | SEGUNDA | 16:40 / 17:30 | QUARTA | 15:50 / 16:40 |
| Cálculo Diferencial e Integral III (turma 2) | 4 | SEGUNDA | 15:50 / 16:40 | SEGUNDA | 16:40 / 17:30 | QUARTA | 15:50 / 16:40 |
| Cálculo Numérico | 4 | TERÇA | 09:50 / 10:40 | TERÇA | 10:40 / 11:30 | QUINTA | 09:50 / 10:40 |
| Cálculo Vetorial e Geomettria Analítica | 4 | TERÇA | 08:00 / 08:50 | TERÇA | 08:50 / 09:40 | QUINTA | 08:00 / 08:50 |
| Computadores e Sociedade | 2 | QUINTA | 08:00 / 08:50 | QUINTA | 08:50 / 09:40 | - | - |
| Elementos dos Cálculos das Variações | 4 | TERÇA | 14:00 / 14:50 | TERÇA | 14:50 / 15:40 | QUINTA | 14:00 / 14:50 |
| Equações Diferenciais Ordinárias Aplicadas | 4 | SEGUNDA | 09:50 / 10:40 | SEGUNDA | 10:40 / 11:30 | QUARTA | 09:50 / 10:40 |
| Equações Diferenciais Parciais Aplicadas | 4 | TERÇA | 08:00 / 08:50 | TERÇA | 08:50 / 09:40 | QUINTA | 08:00 / 08:50 |
| Estrutura de Dados | 4 | SEGUNDA | 14:00 / 14:50 | SEGUNDA | 14:50 / 15:40 | QUARTA | 14:00 / 14:50 |
| Física Aplicada a Computação I | 4 | SEGUNDA | 09:50 / 10:40 | SEGUNDA | 10:40 / 11:30 | QUARTA | 09:50 / 10:40 |
| Física Aplicada a Computação II | 4 | SEGUNDA | 14:00 / 14:50 | SEGUNDA | 14:50 / 15:40 | QUARTA | 14:00 / 14:50 |
| Inteligência Computacional | 4 | TERÇA | 09:50 / 10:40 | TERÇA | 10:40 / 11:30 | QUINTA | 09:50 / 10:40 |
| Introdução a Algebra Linear | 4 | TERÇA | 09:50 / 10:40 | TERÇA | 10:40 / 11:30 | QUINTA | 09:50 / 10:40 |
| Introdução a Análise Real | 6 | SEGUNDA | 08:00 / 08:50 | SEGUNDA | 08:50 / 09:40 | QUARTA | 08:00 / 08:50 |
| Introdução à Computação Científica | 4 | SEGUNDA | 09:50 / 10:40 | SEGUNDA | 10:40 / 11:30 | QUARTA | 09:50 / 10:40 |
| Introdução à Computação Gráfica | 4 | SEGUNDA | 08:00 / 08:50 | SEGUNDA | 08:50 / 09:40 | QUARTA | 08:00 / 08:50 |

**Figura 3: Planilha para inserção dos horários das disciplinas.**
**Fonte: O Autor (2019)**

| | LAB INF | | | | LAB ELET | | AUD | | SALAS | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | CI101 | CI105 | CI106 | T07 | CI309 | CI310 | CI311 | T05 | CI102 | CI103 | CI104 | CI304 | CI107 | CI108 | CI109 |
| Álgebra Linear Computacional | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Banco de Dados I | 1 | 1 | 1 | 1 | | | | | | | | | | | |
| Cálculo das Probabilidades I | | | | | | | | | | | | 1 | | | |
| Cálculo Diferencial e Integral I | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cálculo Diferencial e Integral II | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cálculo Diferencial e Integral III (turma 1) | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cálculo Diferencial e Integral III (turma 2) | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cálculo Numérico | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Cálculo Vetorial e Geomettria Analítica | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Computadores e Sociedade | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Elementos dos Cálculos das Variações | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Equações Diferenciais Ordinárias Aplicadas | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Equações Diferenciais Parciais Aplicadas | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Estrutura de Dados | | | | | | | | | | | | | | | |
| Física Aplicada a Computação I | | | | | | | | | | | | | | | |
| Física Aplicada a Computação II | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Inteligência Computacional | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Introdução a Algebra Linear | | | | | | | 1 | 100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figura 4: Planilha para inserção das preferências das disciplinas à cada sala.**
**Fonte: O Autor (2019)**

Na aba das salas, o operador precisará informar quais as salas disponíveis para serem utilizadas para a alocação das turmas, quais as suas capacidades e qual o tipo de sala (Figura 5).

| Salas | Capacidade | |
|---|---|---|
| CI101 | 40 | laboratório de informática |
| CI105 | 40 | |
| CI106 | 50 | |
| T07 | 70 | |
| CI309 | 25 | laboratório de eletricidade e eletrônica |
| CI310 | 25 | |
| CI311 | 30 | Lousa Digital |
| T05 | 100 | Auditório |
| CI102 | 60 | Salas de Aula Normais |
| CI103 | 60 | |
| CI104 | 25 | |
| CI304 | 66 | |
| CI107 | 35 | |
| CI108 | 30 | |
| CI109 | 35 | |
| CI306 | 30 | |
| CI307 | 30 | |
| CI308 | 30 | |
| SB-01 | 60 | |

**Figura 5: Planilha para inserção das salas disponíveis, seu tipo e capacidade. Fonte: O Autor (2019)**

Antes de resolver o problema, há um pré-processamento dos dados, que pode determinar se a solução é inviável observando a quantidade de turmas em cada horário. Se esta quantidade for maior que a quantidade de salas disponíveis, o sistema já acusa a inviabilidade para que o resolvedor não tente buscar uma solução inteira ótima quando esta não existe.

Após a inserção desses dados de entrada o sistema já estará pronto para alocar as turmas às suas determinadas salas.

Uma vez feito isto, há duas abas diferentes para apresentar os resultados obtidos. Há uma aba que apresenta de forma geral quais turmas foram alocadas a cada sala (Figura 6). Há também uma segunda forma de visualização do resultado, onde pode ser escolhida uma sala por vez e o sistema mostrará os horários de cada turma que foi alocada a esta sala (Figura 7).

Solve Problem

**Alocação de Aulas às Salas**

| Salas | Disciplinas | | | | | |
|---|---|---|---|---|---|---|
| CI101 | Banco de Dados I | Metodologia do Trabalho Científico | Introdução à Programação para Engenharia de Computação | Linguagem e Programação III | Paradigmas de Ling de Programação | |
| CI105 | Matemática Discreta | Teoria do Controle Ótimo | Avaliação e Desempenho de Sistemas Operacionais | Física Experimental para Computação | | |
| CI106 | Introdução à Programação | Eletrônica Aplicada I | Introdução à Mecânica dos Fluídos | Materiais para Micro e Nano Tecnologia | Materiais para Micro e Nano Tecnologia | Redes de Computadores I |
| T07 | Linguagem de Programação I | Pesquisa Operacional | Linguagem de Programação I | Língua Inglesa I | Microcontroladores | Microcontroladores |
| CI309 | | | | | | |
| CI310 | | | | | | |
| CI311 | Programação Matemática | Circuitos Lógicos II (Turma 2) | Sistemas Embarcados I | | | |
| T05 | Introdução a Algebra Linear | Pesquisa Aplicada a Computação | | | | |
| CI102 | Computadores e Sociedade | Física Aplicada a Computação II | Introdução à Teoria dos Grafos | Programação Não Linear | Simulação Computacional | Cálculo Diferencial e Integral II |
| CI103 | Inteligência Computacional | Introdução à Matemática Computacional | Análise e Projeto de Algoritmos (turma 2) | Circuitos Lógicos I (turma 2) | Circuitos Lógicos II (Turma 1) | Eletricidade e Circuitos para Computação II (turma 2) |
| CI104 | Elementos dos Cálculos das Variações | Análise e Projeto de Algoritmos (turma 1) | Eletricidade e Circuitos para Computação I (turma 2) | | | |
| CI304 | Cálculo das Probabilidades I | Cálculo Diferencial e Integral II | Introdução à Computação Científica | Computadores e Sociedade | Introdução à Computação Gráfica | Introdução à Microeletrônica |
| CI107 | Equações Diferenciais Ordinárias Aplicadas | Arquitetura de Computadores | Laboratório de Introdução a Programação | Redes sem Fio | Séries e Equações Diferenciais Ordinárias (turma 1) | Sistemas Operacionais I |

**Figura 6: Resultado geral da alocação das salas**
**Fonte: O Autor (2019)**

**UNIVERSIDADE FEDERAL DA PARAÍBA**
Campus V - Mangabeira
Centro de Informática

| SALA: | CI304 | | | | | |
|---|---|---|---|---|---|---|
| | Segunda-Feira | Terça-Feira | Quarta-Feira | Quinta-Feira | Sexta-Feira | Sábado |
| 07:10 / 08:00 | | | | | | |
| 08:00 / 08:50 | Computadores e Sociedade | Lógica Aplicada à Computação | Introdução à Microeletrônica | | Tópicos Especiais em Interação Homem-Máquina | |
| 08:50 / 09:40 | Computadores e Sociedade | Lógica Aplicada à Computação | Introdução à Microeletrônica | | | |
| 09:50 / 10:40 | Introdução à Computação Científica | Cálculo das Probabilidades I | Introdução à Computação Científica | Cálculo das Probabilidades I | Métodos de Projeto de Software | |
| 10:40 / 11:30 | Introdução à Computação Científica | Cálculo das Probabilidades I | Introdução à Computação Científica | Cálculo das Probabilidades I | Métodos de Projeto de Software | |
| 11:30 / 12:20 | | | | | | |
| 13:10 / 14:00 | | | | | | |
| 14:00 / 14:50 | Introdução à Computação Gráfica | Introdução à Microeletrônica | Tópicos Especiais em Interação Homem-Máquina | Lógica Aplicada à Computação | | |
| 14:50 / 15:40 | Introdução à Computação Gráfica | Introdução à Microeletrônica | Tópicos Especiais em Interação Homem-Máquina | Lógica Aplicada à Computação | | |
| 15:50 / 16:40 | Cálculo Diferencial e Integral II | Métodos de Projeto de Software | Cálculo Diferencial e Integral II | Introdução à Computação Gráfica | | |
| 16:40 / 17:30 | Cálculo Diferencial e Integral II | Métodos de Projeto de Software | Cálculo Diferencial e Integral II | Introdução à Computação Gráfica | | |

**Figura 7: Resultado da alocação apresentando o horário de cada sala**
**Fonte: O Autor (2019)**

Em uma última aba, pode ser observado se houveram inviabilidades de capacidade na solução. Esta aba é importante pois pode acontecer que essa inviabilidade aconteça devido à demanda ser um pouco maior que a capacidade da sala, cabendo ao operador decidir se há a possibilidade de se colocar mais carteiras na sala de aula, ou então alterar preferências das aulas e gerar novas soluções.

## 5.5 Análise e discussão dos resultados

### 5.5.1 Instância inicial 2019.1

Como pode ser observado no Quadro 5, a alocação realizada de modo manual para o período letivo de 2019.1 precisou fazer uso de 4 salas do LIEPE para alocar 22 turmas.

Na coluna "Resultado 01"consta a alocação inicial realizada pelo modelo, no qual foi necessário o uso de apenas 1 (uma) sala do LIEPE para alocar 3 turmas. Para este resultado, os parâmetros utilizados para resolver o problema visavam a menor utilização das salas dos outros prédios e tentava utilizar o auditório o mínimo possível.

Embora o modelo não tenha conseguido alocar todas as disciplinas nas salas do CI, pode-se observar uma diminuição expressiva na quantidade de turmas que poderiam deixar de utilizar salas de outro prédio e poderiam ser alocadas no próprio CI.

| | | Quantidade de Turmas | | |
|---|---|---|---|---|
| | | Situação real | Resultado 01 | Resultado 02 |
| Auditório | T05 | 5 | 2 | - |
| Salas de aula | CI102 | 6 | 9 | 8 |
| | CI103 | 5 | 7 | 8 |
| | CI104 | 3 | 3 | 4 |
| | CI107 | 3 | 6 | 6 |
| | CI108 | 2 | 5 | 3 |
| | CI109 | 2 | 5 | 4 |
| | CI304 | 7 | 9 | 6 |
| | CI306 | 6 | 5 | 2 |
| | CI307 | 5 | 5 | - |
| | CI308 | 3 | 5 | - |
| | CI311 | 5 | 3 | 6 |
| | SB-01 | 1 | 10 | 8 |
| Lab. Inf. | CI101 | 5 | 5 | 7 |
| | CI105 | 4 | 5 | 5 |
| | CI106 | 5 | 5 | 4 |
| | T07 | 10 | 6 | 4 |
| LIEPE | LIEPE 21 | 4 | 3 | 6 |
| | LIEPE 22 | 6 | - | 5 |
| | LIEPE 23 | 6 | - | 5 |
| | LIEPE 27 | 6 | - | 5 |

**Quadro 5: Comparação dos resultados da alocação real com a alocação gerada pelo modelo.**

Também é possível observar nesse quadro que o uso do Auditório do CI diminuiu de 5 para 2 turmas. Essas 2 turmas foram alocadas no Auditório, pois a penalização de se utilizar o Auditório para essas duas turmas é menor do que precisar fazer uso de uma outra sala do prédio do LIEPE.

Outro aspecto a ser observado é que no resultado obtido, houve grande estabilidade de salas para as turmas com aulas em mais de um dia na semana. Das 103 disciplinas ofertadas, apenas 2 tiveram suas aulas alocadas a salas diferentes.

A pedido da gestão, um outro resultado foi gerado utilizando parâmetros que objetivassem um maior uso das salas do LIEPE e a exclusão do Auditório como sala de aula. Neste caso, como pode ser observado na coluna "Resultado 02"no Quadro 5 é possível observar que houve uma melhor distribuição das disciplinas em todas as salas utilizadas, bem como a exclusão do Auditório como sala de aula para que o mesmo possa ser utilizado apenas para outras finalidades, como reuniões, eventos, palestras e defesas.

Como pode ser observado, vários resultados podem ser obtidos alterando apenas alguns parâmetros no sistema para a resolução do modelo. Isto pode ser feito alterando os dados de entrada do sistema em relação às punições ($P$), para os casos de restrições de capacidade violadas, e com o custo ($C_{ij}$) de se alocar uma aula à determinada sala. Desta forma, a gestão poderá criar facilmente vários cenários nos quais poderão atender e auxiliar na alocação final das turmas às salas de aula.

O tomador de decisão tem, com essa ferramenta, a facilidade de simular diversas situações nas quais novas salas de aula poderão ser utilizadas, turmas que necessitem ficar em alguma sala específica estejam nela, salas que forem compartilhadas podem ter disponibilidade apenas em determinados horários, entre outros.

### 5.5.2 Novas instâncias

Como testes adicionais, foram criadas outras 3 instâncias acrescentando novas turmas fictícias à instância 2019.1. Podendo então analisar os resultados do modelo em uma situação no qual novas turmas fossem criadas no caso de uma possível abertura de um outro curso.

A criação dessas novas turmas visou homogeneizar a demanda de turmas em cada horário disponível. A partir dos dados apresentados no Quadro 4 apresentado anteriormente, que possilitava a visualização de uma concentração de turmas em determinados dias e horários da semana, foi possível a criação de novas disciplinas nos horários em que a demanda por turmas eram menores.

Sendo assim as instâncias *2019.1+15, 2019.1+30* e *2019.1+47* foram criadas adicionando respectivamente 15, 30 e 47 turmas a mais para completar a instância inicial

2019.1 priorizando os horários nos quais havia pouca demanda por turmas.

Desta forma, é possível mostrar que, em relação ao espaço físico, há possibilidade da criação de novas disciplinas sem que a capacidade de salas disponíveis seja ultrapassada. O Quadro 6 apresenta as características e tempo de processamento para a resolução das instâncias utilizadas.

| Instância | Quantidade de turmas | Índice de ocupação das salas | Tempo de processamento |
|---|---|---|---|
| 2019.1 | 103 | 49% | 3min 7s |
| 2019.1+15 | 118 | 56% | 5min 12s |
| 2019.1+30 | 133 | 63% | 9min 6s |
| 2019.1+47 | 150 | 70% | 33min 40s |

**Quadro 6: Dados das instâncias utilizadas.**

Para a determinação do índice de ocupação das salas (IOS) foi considerado que em 1 semana há 40 janelas de horários disponíveis para que as disciplinas sejam alocadas. Utilizando as 20 salas disponíveis, este índice foi determinado para cada instância utilizando a equação 16.

$$IOS = \frac{\text{demanda de horários}}{(\text{horários disponíveis } x \text{ quantidade de salas})} \tag{16}$$

Os resultados das alocações das instâncias *2019.1, 2019.1+15, 2019.1+30* e *2019.1+47* podem ser observadas, respectivamente, nos Apêndices 1, 2, 3 e 4.

Conforme dito anteriormente, existe uma concentração de aulas ministradas nas terças e quintas feiras, e poucas aulas sendo ministradas nas sextas feiras. Se houvesse uma melhor distribuição das disciplinas nos dias da semana, facilmente todas as turmas ofertadas pelo CI poderiam ser alocadas no prédio do próprio CI. Inslcusive, caso haja a necessidade de criação de um novo curso, este aspecto deverá ser levado em condiseração no momento da determinação dos horários de cada nova disciplina.

Em testes realizados utilizando a criação das mesmas 47 turmas adicionais sendo alocadas em horários aleatórios, o modelo passou mais de 2h processando e não obteve uma solução viável. No entanto, quando o teste foi realizado com a mesma quantidade de turmas adicionais, mas com elas sendo direcionadas aos dias e horários de menor demanda, o resultado foi obtido de forma exata com um tempo computacional muito inferior.

# 6 CONCLUSÕES

Esta pesquisa atingiu seu objetivo inicial, uma vez que houve o desenvolvimento de um sistema informatizado que poderá ser utilizado para futuras alocações de sala para o Centro de Informática da UFPB.

O sistema foi desenvolvido de modo que atende às principais exigências identificadas pelos gestores do CI, tais quais: permitir que aulas teóricas possam ser alocadas a laborátórios, mas que o contrário não seja possível; em caso se solução inviável, alocar aulas em salas com capacidade inferior à demanda para que o gestor possa tomar a decisão de incluir mais carteiras à estas salas; e tentar utilizar o mínimo possível o auditório como sala de aula para que este possa ser utilizado livremente para outras atividades e eventos.

Esse sistema poderá ser instalado nos computadores dos responsáveis por alocar as aulas do CI e poderá facilitar e muito a resolução dessa atividade que antes levava dias para ser concluída e agora, uma vez inseridos todos os dados de entrada, poderá ser resolvido em poucos minutos.

Essa agilidade na resolução do problema abre novas possibilidades para a gestão que irá fazer uso do mesmo, uma vez que estarão disponíveis para realizar outras atribuições de seu setor, bem como poderão criar novos cenários de alocações das aulas visando diferentes objetivos.

# REFERÊNCIAS


[1] ALARCÃO, D. T. A. **Melhorias para o problema de designação de salas de aula da PUC Goiás** 2015. 85 f. Dissertação (Mestrado em Engenharia) - Pontifícia Universidade Católica de Goiás, GOIÂNIA, 2015.

BARBOSA, S. H. D. **Contribuições para a resolução do problema de programação de cursos universitários baseada em currículos via metaheurísticas**. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional), Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte - MG, 2012.

BLUM, C. *et. al.* **A Brief Survey on Hybrid Metaheuristics**. 4th international conference on bioinspired optimization methods and their applications, pp 3–16, 2010.

CARTER, M. W., TOVEY, C. A. **When Is the Classroom Assignment Problem Hard?**. Operations Research. v. 40. n. 1, supplement 1, p. S28-S39, 1992.

CIRINO, R. B. Z., **Abordagens de solução para o problema de alocação de aulas a salas**. 119 f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional). Instituto de Ciências Matemáticas e de Computação ICMC-USP, São Carlos - SP, 2016.

CIRINO, R. B. Z., SANTOS, M. O., DELBEM, A.C.B. **Aplicação da metaheurística AGC para o problema de alocação de aulas à salas**. Anais do XLVII SBPO, 2015.

COLIN, E. C. **Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, *marketing* e vendas**. Rio de Janeiro: LTC, 2007.

CORNE D., ROSS P., FANG H. L. **Fast practical evolutionary timetabling**. In: Fogarty T.C. (eds) Evolutionary Computing. Lecture Notes in Computer Science, vol 865. Springer, Berlin, Heidelberg, 1994.

COSTA, L. C. A. **Otimização na gestão de atividades acadêmicas: aplicações na UFPB**. 78 f. Monografia (Curso de Graduação em Engenharia de Produção) - Universidade Federal da Paraíba, João Pessoa - PB, 2013.

FILHO, W. M. **Desenvolvimento e aplicação de algoritmos heurísticos ao poblema de alocação de espaço físico**. 65f. Dissertação (Mestrado em Ciência da Computação) Universidade Estadual de Maringá, Maringá, 2008.

GLOVER, F., KOCHENBERGER, G. A. **Handbook of Metaheuristics**. Kluwer Academic, Publishers, Boston, 2003.

GOLDBARG, M. C., LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2ª ed - Rio de Janeiro: Elsevier, 2005.

GOODRICH, M. T., TAMASSIA, R. **Projeto de algoritmos[recurso eletrônico]: fundamentos, análise e exemplos da internet**. Tradução Bernardo Copstein e João Batista Oliveira. Porto Alegre : Bookman, 2008.

LUKE, S. **Essentials of Metaheuristics**, Lulu, Segunda edição, 2013. Disponível em: http://cs.gmu.edu/∼ sean/book/metaheuristics/

LUZIA, L. F., RODRIGUES, M. C. **Introdução ao Escalonamento e Aplicações**: Estudo sobre as Metaheurísticas. Notas de aula, 2009.

MARTÍNEZ, A. H., FLORES, T. G. **Solving the classroom assignment problem with simulated annealing**. In: Ieee International Conference on Systems, Man & Cybernetics, San Diego, USA, v. 4, p. 37033708, 1998.

MOLENBRUHC, Y., BRAEKERS, K., CARIS, A. **Typology and literature review for dial-a-ride problems**. Annals of Operations Research, v. 259, n. 1-2, p.295-325, 2017.

PHILLIPS, A. E., *et. al.*. **Integer programming methods dor large-scale practical classroom assignment problems**. Computers & Operations research, Vol. 53, Jan. 2015, P. 42-53.

PILLAY, N. **A survey of school timetabling research**. Annals of Operations Research, v. 218, n. 1, p.261-293, 2014.

PRADO, A. S., SOUZA, S. R. **Problema de Alocação de Salas em cursos universitários: um estudo de caso**. XLVI Simpósio Brasileiro de Pesquisa Operacional, Salvador - BA, 2014.

POULSEN, C. J. B. **Desenvolvimento de um modelo para o *School Timetabling Problem* baseado na Meta-Heurística *Simulated Annealing***. 141 f. Dissertação (Curso de Mestrado em Administração) Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, 2012.

TRUJILLO, A. R. M., SARANGO, P. A. Z. **Altoritmos matheurísticos para un problema de planificación de rutas vehiculares y pedestres**. 118f. Trabajo de Titulación (Matemático e Ingeniero Matemático). Facultad de Ciencias. Escola Politécnica Nacional. Quito, 2018.

SALES, E. S. **Problema de Alocação de Salas e a Otimização dos Espaços no Centro de Tecnologia da UFSM**. 120 f. Dissertação (Mestrado em Administração), Universidade Federal de Santa Maria, Santa Maria - RS, 2015a.

SALES, E. S., MÜLLER, F. M., SIMONETTO, E. O. **Solução do Problema de Alocação de Salas utilizando um modelo matetmático multi-índice**. XLVII Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas, 2015b.

SAVINIEC, L. **Models and algorithms for high school timetabling problems**. Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2017.

SCHAERF, A. **A Survey of Automated Timetabling**. Artificial Intelligence Review. v. 13, n. 2, p. 87-127, 1999.

SILVA, A. S. N. **Estudo e Implementação, Mediante Recozimento Simulado, do Problema de Alocação de Salas**. 120 f. Monografia (Curso de Graduação em Ciência da Computação). Universidade Federal de Lavras, Lavras - MG, 2005.

SOUZA, M. J. F. **Ingeligência Computacional para Otimização**. Notas de aula, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto. Ouro Preto - MG, 2005.

SOUZA, M. J. F., MARTINS, A. X., ATAÚJO, C. R. **Experiências com a utilização de *Simulated Annealing* e Busca Tabu na resolução do Problema de Alocação de Salas**. In: Simpósio Brasileiro de Pesquisa Operacional, 34, 2002.

QU, R., *et. al.*. **A survey of search methodologies and automated system development for examination timetabling**. Journal of Scheduling, v. 12, n. 1, p. 55-89, 2009

SUBRAMANIAN, A., MEDEIROS, J.M.F., CABRAL, L.A.F.e SOUZA, M.J.F. **Aplicação da metaheurística Busca Tabu na resolução do Problema de Alocação de Salas do Centro de Tecnologia da UFPB**. XXVI Encontro Nacional de Engenharia de Produção, Fortaleza - CE, 2006.

SUBRAMANIAN, A., MEDEIROS, J. M. F., FORMIGA, L. A., SOUZA, M. J. F. **Aplicação da metaheurística busca tabu ao problema de alocação de aulas a salas em uma instituição universitária**. Revista Produção Online. v.11, n.1, mar. 2011.

THEODORO, G. P,. PERETTA, I. S., YAMANAKA, K. **Utilização de Algoritmos Genéticos para o Problema de Alocação de Salas da Universidade Federal de Uberlândia**. XII Encontro Nacional de Inteligência Artificial e Computacional, Recife - PE, 2016.

# APÊNDICE 1 - Alocação para a instância 2019.1

| Salas | Disciplinas | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CI101 | Banco de Dados I | Matemática Discreta | Adm para Engenharia | Análise e Projeto de Algoritmos (turma 1) | Introdução à Programação para Engenharia de Computação | Paradigmas de Ling de Programação | Tópicos Especiais em Arquiteturas de Computadores | | | |
| CI105 | Metodologia do Trabalho Científico | Teoria do Controle Ótimo | Linguagem de Programação II | Mecânica para Engenharia da Computação | Redes de Computadores I | | | | | |
| CI106 | Introdução à Programação | Introdução à Mecânica dos Fluídos | Materiais para Micro e Nano Tecnologia | Linguagens Formais e Autômatos | | | | | | |
| T07 | Linguagem de Programação I | Pesquisa Operacional | Linguagem de Programação I | Língua Inglesa I | | | | | | |
| CI309 | | | | | | | | | | |
| CI310 | | | | | | | | | | |
| CI311 | Computadores e Sociedade | Programação Matemática | Introd à Engenharia da Computação | Introdução aoo Processamento Digital de Imagens | Sistemas e Controle de Automação | Sistemas Embarcados I | | | | |
| T05 | | | | | | | | | | |
| CI102 | Introdução a Álgebra Linear | Soluções Numéricas em Equações Diferenciais II | Circuitos Lógicos I (turma 2) | Computadores e Sociedade | Eletrônica Aplicada I | Pesquisa Aplicada a Computação | Metodologia do Trabalho Científico | Métodos de Projeto de Software | | |
| CI103 | Cálculo Diferencial e Integral III (turma 1) | Cálculo Numérico | Cálculo Vetorial e Geometria Analítica | Introdução à Computação Científica | Avaliação e Desempenho de Sistemas Operacionais | Circuitos Lógicos II (Turma 2) | Laboratório de Introdução a Programação | Tópicos Especiais em Interação Homem-Máquina | | |
| CI104 | Elementos dos Cálculos das Variações | Estrutura de Dados | Eletricidade e Circuitos para Computação I (turma 1) | Redes sem Fio | | | | | | |
| CI304 | Cálculo das Probabilidades I | Cálculo Diferencial e Integral II | Linguagem de Programação II | Ordenação e Recuperação de Dados | Introdução à Inteligência Artificial | Sinais e Sistemas Dinâmicos | | | | |
| CI107 | Álgebra Linear Computacional | Engenharia de Software | Introdução à Álgebra Linear | Séries e Equações Diferenciais Ordinárias (turma 1) | Sistemas Operacionais I | Tópicos Especiais em Engenharia de Computação IV | | | | |
| CI108 | Introdução à Computação Gráfica | Eletricidade e Circuitos para Computação I (turma 2) | Eletrônica Aplicada II | | | | | | | |
| CI109 | Equações Diferenciais Parciais Aplicadas | Física Aplicada a Computação I | Linguagem e Programação III | Tópicos Especiais em Processamento Digital de Imagens | | | | | | |
| CI306 | Concepção Estruturada de Circuitos Integrados | Eletricidade e Circuitos para Computação II (turma 1) | | | | | | | | |
| CI307 | | | | | | | | | | |
| CI308 | | | | | | | | | | |
| SB-01 | Equações Diferenciais Ordinárias Aplicadas | Equações Diferenciais Ordinárias Aplicadas | Física Aplicada a Computação II | Simulação Computacional | Cálculo Diferencial e Integral II | Cálculo Vetorial e Geometria Analítica | Circuitos Lógicos I (turma 1) | Trabalho Individual em Arquitetura de Computadores | Teoria da Computação | |
| LIEPE 21 | Inteligência Computacional | Introdução à Matemática Computacional | Introdução à Computação Gráfica | Introdução à Microeletrônica | Microcontroladores | Séries e Equações Diferenciais Ordinárias (turma 1) | | | | |
| LIEPE 22 | Análise e Projeto de Algoritmos (turma 2) | Circuitos Lógicos II (Turma 1) | Eletricidade e Circuitos para Computação II (turma 2) | Introdução à Teoria da Informação (turma 2) | Introdução ao Computador | | | | | |
| LIEPE 23 | Cálculo Diferencial e Integral I | Introdução à Teoria dos Grafos | Programação Não Linear | Introdução à Teoria da Informação (turma 1) | Laboratório de Linguagem de Programação I | | | | | |
| LIEPE 27 | Cálculo Diferencial e Integral III (turma 2) | Arquitetura de Computadores | Física Experimental para Computação | Robotica | Lógica Aplicada à Computação | | | | | |

# APÊNDICE 2 - Alocação para a instância 2019.1+15

| Salas | Disciplinas | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI101 | Banco de Dados I | Análise e Projeto de Algoritmos (turma 1) | Arquitetura de Computadores | Avaliação e Desempenho de Sistemas Operacionais | Introdução à Mecânica dos Fluídos | Materiais para Micro e Nano Tecnologia | Mecânica para Engenharia da Computação | 4 | 10 | 14 | | | | |
| CI105 | Matemática Discreta | Teoria do Controle Ótimo | Introdução à Programação para Engenharia de Computação | Redes de Computadores I | Linguagens Formais e Autômatos | 13 | | | | | | | | |
| CI106 | Metodologia do Trabalho Científico | Pesquisa Operacional | Linguagem de Programação I | Linguagem de Programação II | Materiais para Micro e Nano Tecnologia | Microcontroladores | 2 | 15 | | | | | | |
| T07 | Introdução à Programação | Linguagem de Programação I | Língua Inglesa I | Tópicos Especiais em Processamento Digital de Imagens | 5 | 7 | 11 | 12 | | | | | | |
| CI309 | Estrutura de Dados | | | | | | | | | | | | | |
| CI310 | Física Aplicada a Computação I | | | | | | | | | | | | | |
| CI311 | Equações Diferenciais Ordinárias Aplicadas | Equações Diferenciais Parciais Aplicadas | Linguagem de Programação II | Programação Matemática | Adm para Engenharia | Eletricidade e Circuitos para Computação III (turma 1) | Laboratório de Introdução a Programação | | | | | | | |
| T05 | | | | | | | | | | | | | | |
| CI102 | Soluções Numéricas em Equações Diferenciais II | Cálculo Diferencial e Integral II | Cálculo Vetorial e Geometria Analítica | Computadores e Sociedade | Introdução à Computação Gráfica | Redes sem Fio | Tópicos Especiais em Interação Homem-Máquina | Trabalho Individual em Arquitetura de Computadores | Teoria da Computação | | | | | |
| CI103 | Cálculo Diferencial e Integral III (turma 1) | Computadores e Sociedade | Introdução à Teoria dos Grafos | Ordenação e Recuperação de Dados | Simulação Computacional | Eletrônica Aplicada I | Eletrônica Aplicada II | Laboratório de Linguagem de Programação I | Introdução ao Computador | Linguagem e Programação III | | | | |
| CI104 | Introdução à Computação Gráfica | Física Experimental para Computação | Séries e Equações Diferenciais Ordinárias (turma 1) | Sistemas Embarcados I | Métodos de Projeto de Software | | Séries e Equações Diferenciais Ordinárias (turma 1) | | | | | | | |
| CI304 | Cálculo das Probabilidades I | Cálculo Diferencial e Integral II | Física Aplicada a Computação II | Introdução à Computação Científica | Análise e Projeto de Algoritmos (turma 2) | Circuitos Lógicos II (Turma 1) | 8 | | | | | | | |
| CI107 | Eletricidade e Circuitos para Computação I (turma 1) | Eletricidade e Circuitos para Computação I (turma 2) | Introdução à Álgebra Linear | Sistemas Operacionais I | Tópicos Especiais em Engenharia de Computação IV | 3 | | | | | | | | |
| CI108 | Álgebra Linear Computacional | Elementos dos Cálculos das Variações | | | | | | | | | | | | |
| CI109 | Introdução ao Processamento Digital de Imagens | Sistemas e Controle de Automação | Metodologia do Trabalho Científico | | | | | | | | | | | |
| CI306 | Circuitos Lógicos II (Turma 2) | | | | | | | | | | | | | |
| CI307 | | | | | | | | | | | | | | |
| CI308 | Concepção Estruturada de Circuitos Integrados | Introd à Engenharia da Computação | | | | | | | | | | | | |
| SB-01 | Inteligência Computacional | Introdução à Matemática Computacional | Programação Não Linear | Circuitos Lógicos I (turma 1) | Circuitos Lógicos I (turma 2) | Sinais e Sistemas Dinâmicos | 6 | | | | | | | |
| LIEPE 21 | Cálculo Vetorial e Geometria Analítica | Introdução à Teoria da Informação (turma 2) | Pesquisa Aplicada a Computação | Paradigmas de Ling de Programação | 9 | | | | | | | | | |
| LIEPE 22 | Introdução a Álgebra Linear | Robotica | 1 | | | | | | | | | | | |
| LIEPE 23 | Cálculo Numérico | Engenharia de Software | Introdução à Microeletrônica | | | | | | | | | | | |
| LIEPE 27 | Cálculo Diferencial e Integral I | Cálculo Diferencial e Integral III (turma 2) | Introdução à Inteligência Artificial | Introdução à Teoria da Informação (turma 1) | Lógica Aplicada à Computação | Tópicos Especiais em Arquiteturas de Computadores | | | | | | | | |

# APÊNDICE 3 - Alocação para a instância 2019.1+30

| Salas | Disciplinas | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI101 | Linguagem de Programação I | Matemática Discreta | Introdução à Programação para Engenharia de Computação | Linguagem e Programação III | 19 | 21 | 42 | | | | | | | |
| CI105 | Teoria do Controle Ótimo | Adm para Engenharia | Linguagem de Programação II | 26 | | | | | | | | | | |
| CI106 | Metodologia do Trabalho Científico | Linguagem de Programação I | Linguagem de Programação II | Redes de Computadores I | Introdução à Mecânica dos Fluidos | Materiais para Micro e Nano Tecnologia | 18 | 20 | 43 | | | | | |
| T07 | Banco de Dados I | Introdução à Programação | Pesquisa Operacional | Lingua Inglesa I | Avaliação e Desempenho de Sistemas Operacionais | Métodos de Projeto de Software | 4 | 12 | 30 | 38 | | | | |
| CI309 | | | | | | | | | | | | | | |
| CI310 | | | | | | | | | | | | | | |
| CI311 | Álgebra Linear Computacional | Eletrônica Aplicada II | Laboratório de Introdução a Programação | Séries e Equações Diferenciais Ordinárias (turma 1) | Eletrônica Aplicada I | Sistemas Operacionais I | 33 | | | | | | | |
| T05 | | | | | | | | | | | | | | |
| CI102 | Equações Diferenciais Parciais Aplicadas | Estrutura de Dados | Cálculo Diferencial e Integral II | Eletricidade e Circuitos para Computação II (turma 1) | Microcontroladores | Séries e Equações Diferenciais Ordinárias (turma 1) | Pesquisa Aplicada a Computação | Paradigmas de Ling de Programação | 3 | 15 | | | | |
| CI103 | Cálculo Diferencial e Integral I | Introdução a Álgebra Linear | Introdução à Computação Gráfica | Programação Não Linear | Análise e Projeto de Algoritmos (turma 1) | Sistemas Embarcados I | 14 | 25 | | | | | | |
| CI104 | Elementos dos Cálculos das Variações | 7 | 23 | 39 | | | | | | | | | | |
| CI304 | Cálculo das Probabilidades I | Cálculo Diferencial e Integral II | Introdução à Computação Gráfica | Programação Não Linear | Análise e Projeto de Algoritmos (turma 1) | Sistemas Embarcados I | 5 | 6 | 24 | | | | | |
| CI107 | Introd à Engenharia da Computação | Introdução ao Processamento Digital de Imagens | Redes sem Fio | Sistemas e Controle de Automação | Sistemas Embarcados I | Circuitos Lógicos I (turma 1) | 14 | 25 | | | | | | |
| CI108 | Circuitos Lógicos II (Turma 2) | Concepção Estruturada de Circuitos Integrados | 22 | 29 | 35 | Introdução à Álgebra Linear | Introdução à Microeletrônica | Mecânica para Engenharia da Computação | 8 | 17 | | | | |
| CI109 | Engenharia de Software | Tópicos Especiais em Engenharia de Computação IV | Linguagens Formais e Autômatos | Teoria da Computação | 44 | | | | | | | | | |
| CI306 | Cálculo Diferencial e Integral III (turma 2) | Equações Diferenciais Ordinárias Aplicadas | 10 | | | | | | | | | | | |
| CI307 | Física Aplicada a Computação I | Linguagem de Programação II | Eletricidade e Circuitos para Computação I (turma 1) | Eletricidade e Circuitos para Computação I (turma 2) | 2 | Circuitos Lógicos I (turma 2) | 13 | Circuitos Lógicos II (Turma 1) | Eletricidade e Circuitos para Computação II (turma 2) | 28 | | | | |
| CI308 | Programação Matemática | Física Experimental para Computação | 1 | 9 | 40 | | | | | | | | | |
| SB-01 | Cálculo Diferencial e Integral III (turma 1) | Cálculo Numérico | Soluções Numéricas em Equações Diferenciais II | Análise e Projeto de Algoritmos (turma 2) | Circuitos Lógicos I (turma 2) | | | | | | | | | |
| LIEPE 21 | Computadores e Sociedade | Inteligência Computacional | Introdução à Computação Científica | Cálculo Vetorial e Geometria Analítica | Circuitos Lógicos I (turma 1) | Introdução à Álgebra Linear | Introdução à Microeletrônica | Tópicos Especiais em Interação Homem-Máquina | 27 | 37 | | | | |
| LIEPE 22 | Física Aplicada a Computação II | Introdução à Teoria dos Grafos | Arquitetura de Computadores | Computadores e Sociedade | Introdução à Álgebra Linear | | | | | | | | | |
| LIEPE 23 | Introdução à Matemática Computacional | Ordenação e Recuperação de Dados | Simulação Computacional | Introdução à Teoria da Informação (turma 1) | Laboratório de Linguagem de Programação I | Sinais e Sistemas Dinâmicos | Lógica Aplicada à Computação | Tópicos Especiais em Arquiteturas de Computadores | Tópicos Especiais em Processamento Digital de Imagens | | | | | |
| LIEPE 27 | Cálculo Vetorial e Geometria Analítica | Introdução à Inteligência Artificial | Introdução à Teoria da Informação (turma 2) | Robotica | Introdução ao Computador | 11 | 34 | 36 | 41 | 45 | | | | |

# APÊNDICE 4 - Alocação para a instância 2019.1+47

| Salas | Disciplinas | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI101 | Linguagem de Programação I | Matemática Discreta | Introdução à Programação para Engenharia de Computação | Linguagem e Programação III | 19 | 21 | 42 | | | | | | | |
| CI105 | Teoria do Controle Ótimo | Adm para Engenharia | Linguagem de Programação II | 26 | | | | | | | | | | |
| CI106 | Metodologia do Trabalho Científico | Linguagem de Programação I | Linguagem de Programação II | Língua Inglesa I | Redes de Computadores I | 16 | 18 | 20 | | | | | | |
| T07 | Banco de Dados I | Introdução à Programação | Pesquisa Operacional | Avaliação e Desempenho de Sistemas Operacionais | Introdução à Mecânica dos Fluidos | Materiais para Micro e Nano Tecnologia | 4 | 12 | 30 | 38 | | | | |
| CI309 | | | | | | | | | | | | | | |
| CI310 | | | | | | | | | | | | | | |
| CI311 | Álgebra Linear Computacional | Eletrônica Aplicada II | Laboratório de Introdução a Programação | Séries e Equações Diferenciais Ordinárias (turma 1) | Métodos de Projeto de Software | 33 | | | | | | | | |
| T05 | | | | | | | | | | | | | | |
| CI102 | Equações Diferenciais Parciais Aplicadas | Estrutura de Dados | Cálculo Diferencial e Integral II | Eletricidade e Circuitos para Computação II (turma 1) | Eletrônica Aplicada I | Pesquisa Aplicada a Computação | Paradigmas de Ling de Programação | 3 | 15 | | | | | |
| CI103 | Cálculo Diferencial e Integral I | Introdução a Álgebra Linear | Introdução à Computação Gráfica | Microcontroladores | Séries e Equações Diferenciais Ordinárias (turma 1) | Sistemas Operacionais I | Trabalho Individual em Arquitetura de Computadores | Metodologia do Trabalho Científico | 31 | 32 | | | | |
| CI104 | Elementos dos Cálculos das Variações | 7 | 23 | 39 | | | | | | | | | | |
| CI304 | Cálculo das Probabilidades I | Cálculo Diferencial e Integral II | Introdução à Computação Gráfica | Programação Não Linear | Análise e Projeto de Algoritmos (turma 1) | 5 | 6 | 24 | | | | | | |
| CI107 | Introd à Engenharia da Computação | Introdução aoo Processamento Digital de Imagens | Redes sem Fio | Sistemas e Controle de Automação | Sistemas Embarcados I | 14 | 25 | | | | | | | |
| CI108 | Circuitos Lógicos II (Turma 2) | Concepção Estruturada de Circuitos Integrados | 22 | 29 | | | | | | | | | | |
| CI109 | Engenharia de Software | Tópicos Especiais em Engenharia de Computação IV | Linguagens Formais e Autômatos | Teoria da Computação | 35 | 44 | | | | | | | | |
| CI306 | Cálculo Diferencial e Integral III (turma 2) | Equações Diferenciais Ordinárias Aplicadas | 10 | | | | | | | | | | | |
| CI307 | Física Aplicada a Computação I | Linguagem de Programação II | Eletricidade e Circuitos para Computação I (turma 1) | Eletricidade e Circuitos para Computação I (turma 2) | 2 | 13 | | | | | | | | |
| CI308 | Programação Matemática | Física Experimental para Computação | 1 | 9 | 40 | | | | | | | | | |
| SB-01 | Cálculo Diferencial e Integral III (turma 1) | Cálculo Numérico | Soluções Numéricas em Equações Diferenciais II | Análise e Projeto de Algoritmos (turma 2) | Circuitos Lógicos I (turma 2) | Circuitos Lógicos II (Turma 1) | Eletricidade e Circuitos para Computação II (turma 2) | 28 | | | | | | |
| LEPE 21 | Computadores e Sociedade | Inteligência Computacional | Introdução à Computação Científica | Cálculo Vetorial e Geometria Analítica | Circuitos Lógicos I (turma 1) | Tópicos Especiais em Interação Homem-Máquina | 27 | 37 | | | | | | |
| LEPE 22 | Física Aplicada a Computação II | Introdução à Teoria dos Grafos | Arquitetura de Computadores | Computadores e Sociedade | Introdução à Álgebra Linear | Introdução à Microeletrônica | Mecânica para Engenharia da Computação | 8 | 17 | | | | | |
| LEPE 23 | Introdução à Matemática Computacional | Ordenação e Recuperação de Dados | Simulação Computacional | Introdução à Teoria da Informação (turma 1) | Laboratório de Linguagem de Programação I | Sinais e Sistemas Dinâmicos | Lógica Aplicada à Computação | Tópicos Especiais em Arquiteturas de Computadores | Tópicos Especiais em Processamento Digital de Imagens | 45 | | | | |
| LEPE 27 | Cálculo Vetorial e Geometria Analítica | Introdução à Inteligência Artificial | Introdução à Teoria da Informação (turma 2) | Robotica | Introdução ao Computador | 11 | 34 | 36 | 41 | | | | | |