



UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO



## **Abordagem do Problema de Alocação de Professores à Disciplinas e da Alocação de Disciplinas à Horários via PSR**

Relatório de Inteligência Artificial

Adisiel Alemão Santos Silveira, Thiago Joaquim Lima Ferreira e  
Flávio Rodolfo Nunes Santos

São Cristóvão – Sergipe

2020

UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO

Adisiel Alemão Santos Silveira, Thiago Joaquim Lima Ferreira e  
Flávio Rodolfo Nunes Santos

**Abordagem do Problema de Alocação de Professores à  
Disciplinas e da Alocação de Disciplinas à Horários via PSR**

Relatório correspondente à Unidade 3, submetido ao  
Dr. Carlos Alberto Estombelo como requisito total  
para a obtenção da nota da Unidade 3 em Inteligência  
Artificial.

São Cristóvão – Sergipe

2020

# Resumo

A atribuição de professores à disciplinas e de disciplinas à horários (a integração destes três componentes gera uma turma) é uma tarefa que ocorre periodicamente em instituições de ensino. Devido a imensa quantidade de possíveis atribuições distintas de professores à disciplinas, de disciplinas à horários e restrições impostas que devem ser atendidas, esse processo de alocação é uma tarefa complexa e que, se realizada de forma manual, pode demandar muito tempo e esforço de quem a realiza. Este trabalho apresenta uma proposta de solução para a problemática exposta acima, levando em consideração parte do processo ao qual ocorre no Departamento de Computação da Universidade Federal de Sergipe. São consideradas as preferências de professores por disciplinas além de outras restrições que serão descritas neste relatório. Para resolver o problema, foi formulado e implementado um modelo de PSR (Problema de Satisfação de Restrições) que atenda as restrições adotadas pela instituição.

**Palavras-chave:** Problemas de Satisfação de Restrição, Alocação de Professores à Disciplinas, Alocação de Disciplinas à Horários.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Desenvolvimento</b>	<b>5</b>
2.1	Modelagem do Problema - Variáveis	5
2.2	Modelagem do Problema - Domínio	5
2.3	Modelagem do Problema - Restrições	6
2.4	Algoritmos Utilizados - Busca com Retrocesso	7
2.5	Classes Principais - Domínio	8
2.6	Classes Principais - Variáveis	10
2.7	Classes Principais - Restrições	11
<b>3</b>	<b>Conclusão</b>	<b>13</b>
3.1	Resultados Obtidos	13
3.2	Manual de Instalação	14
	<b>Referências</b>	<b>15</b>

# 1

## Introdução

Grande parte das instituições de ensino se deparam constantemente com duas problemáticas: o problema de alocação de disciplinas à horários e o de alocação de professores à disciplinas, dada uma série de restrições em ambos os problemas. Além disso, sabe-se que os professores podem ter preferências por determinadas disciplinas e dias da semana para ministrar suas aulas. Portanto, devem ser bem gerenciados para que não haja conflitos no que tange a ambas as alocações. É notória a exponencialidade do problema abordado, uma vez que, para  $k$  professores e  $j$  disciplinas (por exemplo), tem-se  $k^j$  possibilidades, sendo possível notar o crescimento exponencial à medida que cresce o tamanho da instância. Diante de tantas possibilidades de atribuições, uma solução para o problema deve suprir a todas as restrições de viabilidade adotadas pela instituição (em nosso caso, o departamento de computação). Essa atividade, quando realizada de forma manual, demanda muito tempo (extremamente dispendiosa e sujeita a falhas) e trabalho de seus realizadores, tornando-se inapropriada em função do crescimento das instituições de ensino. Por sua vez, soluções computacionais tem ganhos na qualidade de resposta, além de reduzir o número de profissionais envolvidos e tempo gasto no processo manual. Para resolução deste problema, modelaremos o problema via CSP (Constraint Satisfaction Problems).

# 2

## Desenvolvimento

### 2.1 Modelagem do Problema - Variáveis

A primeira variável presente no modelo apresentado pelo grupo são as disciplinas convencionais, onde:

$D$  é o conjunto de disciplinas convencionais.

A segunda variável presente no modelo apresentado pelo grupo são as disciplinas obrigatórias, onde:

$D_{obr}$  é o conjunto de disciplinas obrigatórias (segundo a Universidade).

### 2.2 Modelagem do Problema - Domínio

Nesse contexto, definimos um slot (horário): um espaço de tempo de duas horas de um turno e dia da semana, conforme mostrado na tabela 1 abaixo. Por exemplo, o slot 00 (S00) representa a primeira aula de um turno na segunda-feira. Além disso, os professores também fazem parte do domínio. Assim, temos o domínio é formado pelo produto cartesiano de professores e slots (horários).

S00	S10	S20	S30	S40
S01	S11	S21	S31	S41
S02	S12	S22	S32	S42

Tabela 1 – Slots - (Horários)

Utilizando o formalismo matemático, o domínio para as disciplinas convencionais exceto as disciplinas obrigatórias seria representado por:

Seja  $P$  o conjunto de Professores e seja  $S$  o conjunto de Slots (Horários),

$$P \times S = \{(P, S) \mid p \in P \wedge s \in S\} \quad (2.1)$$

Existem também disciplinas que já são tidas como obrigatórias (Departamento de Matemática, Departamento de Elétrica, Departamento de Física) e estas já possuem seu espaço em slots, dado a escolha inicial do usuário para reservar estes espaços. Seu domínio pode ser representado por: Seja  $P$  o conjunto de Professores e seja  $D_{obr}$  o conjunto de disciplinas obrigatórias (segundo a Universidade) onde  $\forall d \in D_{obr}$ , esta possui um slot  $s \in S_{pred}$ , onde  $S_{pred}$  é um conjunto que contém um slot predefinido relacionado a  $d$ ,

$$P \times S_{pred} = \{(P, S_{pred}) \mid p \in P \wedge s \in S_{pred}\} \quad (2.2)$$

## 2.3 Modelagem do Problema - Restrições

Abaixo, serão listadas as restrições que foram devidamente implementadas:

- (i) Um professor tem uma quantidade máxima de horas que deve trabalhar semanalmente [Equação 2.3](#).
- (ii) Cada professor somente pode lecionar apenas disciplinas as quais está apto (inclusive as de sua preferência) [Equação 2.4](#).
- (iii) Um professor não pode ser alocado em duas disciplinas que estão em um mesmo slot [Equação 2.5](#).
- (iv) Se uma disciplina tem um ou mais professores que possuem preferência pela mesma mas estes já estão com suas respectivas cargas horárias preenchidas, então esta disciplina recebe a tag "Professor a Definir".

Utilizando o formalismo matemático, podemos representar estas restrições da seguinte maneira:

Sejam consideradas as seguintes variáveis:

- (a)  $x_{pd} \rightarrow$  Tem como valor 1 se o professor  $p$  está associado a disciplina  $d$ , senão 0.
- (b)  $x_{ij} \rightarrow$  Tem como valor 1 se as disciplinas  $i$  e  $j$  compartilham slot, senão 0.
- (c)  $x_{ps} \rightarrow$  Tem como valor 1 se professor  $p$  atribuído a alguma disciplina no slot  $s$ , senão 0.
- (d)  $D_a \rightarrow$  Conjunto de disciplinas atribuídas ao professor  $p$ .

Sejam considerados os seguintes parâmetros:

- (a)  $max_p \rightarrow$  Número máximo de créditos do professor  $p$ ;
- (b)  $crd_i \rightarrow$  Número de créditos da disciplina  $i$ .

Assim temos postulado matematicamente as seguintes restrições:

•

$$\sum_{d \in D} crd_d * x_{pi} \leq max_p, \forall p \in P. \quad (2.3)$$

•

$$d_{prof} : P \mapsto \varphi(D) \mid d_{prof}(p) = D_p, \quad (2.4)$$

onde  $D_p$  é o conjunto de disciplinas que podem ser atribuídas ao professor.

•

$$x_{pi} + x_{pj} + x_{ij} \leq 2, \forall i, j \in D, \forall p \in P. \quad (2.5)$$

•

$$\forall p \in P \mid \sum_{d \in D_a} crd_d \geq max_p \wedge d \in D_p, \quad (2.6)$$

$d$  recebe a tag "Professor a Definir".

## 2.4 Algoritmos Utilizados - Busca com Retrocesso

A expressão busca com retrocesso é utilizada para indicar uma busca em profundidade que escolhe valores para uma variável de cada vez e que efetua o retrocesso quando uma variável não tem valores válidos restantes a serem atribuídos. Ele escolhe repetidamente uma variável não atribuída e depois experimenta todos os valores no domínio da variável, por vez, tentando encontrar uma solução. Se for detectada uma inconsistência, o RETROCESSO-RECURSSIVO retorna falha, fazendo com que a chamada anterior tente outro valor. O algoritmo é modelado usufruindo-se da busca recursiva em profundidade. Pela variação das funções SELECIONAR-VARIÁVEL-NÃO-ATRIBUÍDA e VALORES-DE-ORDEM-NO-DOMÍNIO podemos implementar as heurísticas de uso geral. A função INFERÊNCIA pode ser utilizada opcionalmente para impor arco-consistência, caminho-consistência ou k-consistência, conforme desejado. Se uma escolha de valor levar ao fracasso (por INFERÊNCIA ou por RETROCESSO-RECURSSIVO), as atribuições de valor (incluindo as realizadas por INFERÊNCIA) são removidas da atribuição e um novo valor será experimentado. O algoritmo é mostrado logo abaixo:



**Algoritmo 1:** função RETROCESSO-RECURSSIVO**Dados:** atribuição, PSR.**Resultado:** Uma solução ou falha.**if** atribuição é completa **then**    | **retornar** atribuiçãovar ← SELECIONAR-VARIÁVEL-NÃO-ATRIBUÍDA(psr); **for** valor emVALORES-DE-ORDEM-NO-DOMÍNIO(var, atribuição, psr) **do**    **if** valor é consistente com atribuição **then**

| adicionar (var = valor) a atribuição;

| inferência ← INFERÊNCIA(psr, atribuição, var);

**if** inferência ≠ falhar **then**

| adicione inferência para atribuição;

| resultado ← RETROCESSO-RECURSIVO(atribuição, psr);

**if** se resultado ≠ falha **then**                | **retornar** resultado

| remover (var = valor) de atribuição;

    | **retornar** falha;

## 2.5 Classes Principais - Domínio

Figura 1 – Diagrama de Classe do Domínio

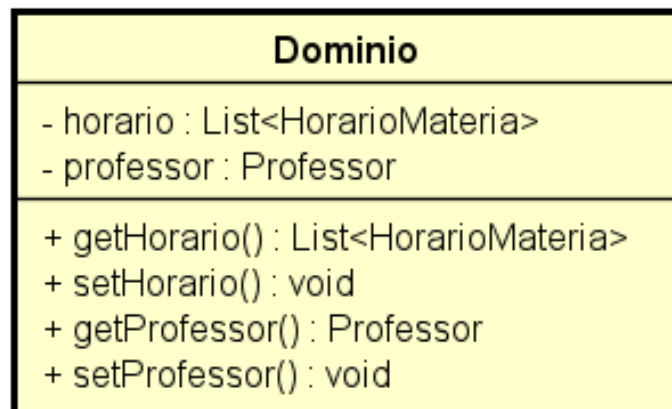


Figura 2 – Diagrama de Classe do Professor

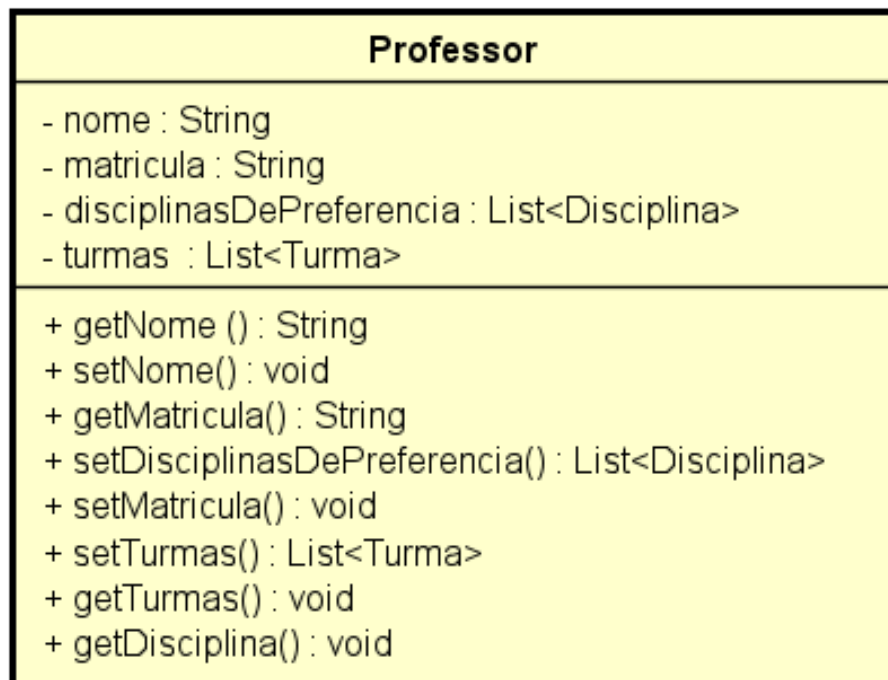
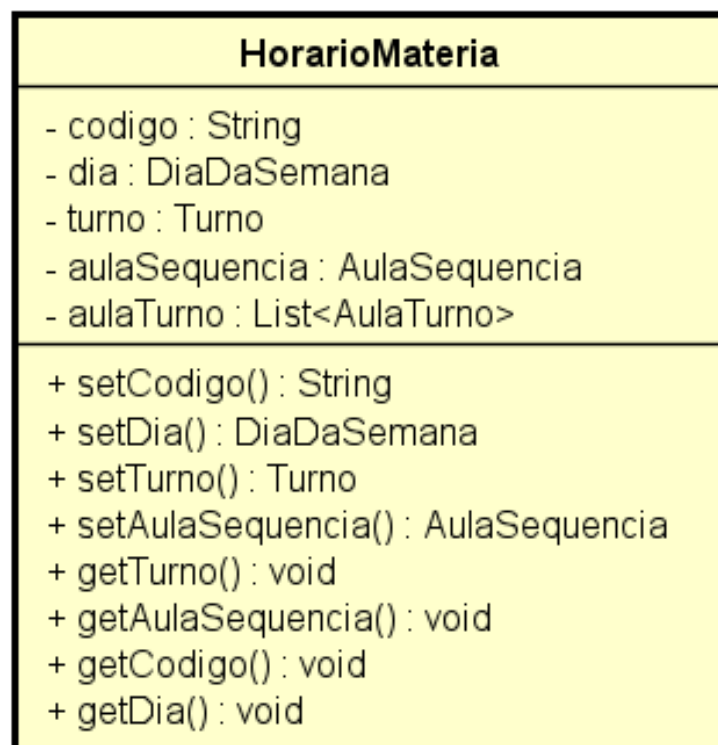


Figura 3 – Diagrama de Classe do HorarioMateria



## 2.6 Classes Principais - Variáveis

É de fundamental importância ressaltar uma classe que tem correlação com disciplina (TurmaVariavel), onde esta foi criada para facilitar a agregação no código, dado que uma turma possui apenas uma única disciplina. O construtor de TurmaVariavel possui o código de uma disciplina.

Figura 4 – Diagrama de Classe de TurmaVariavel

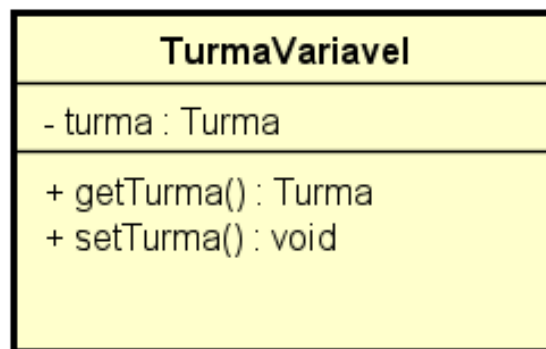
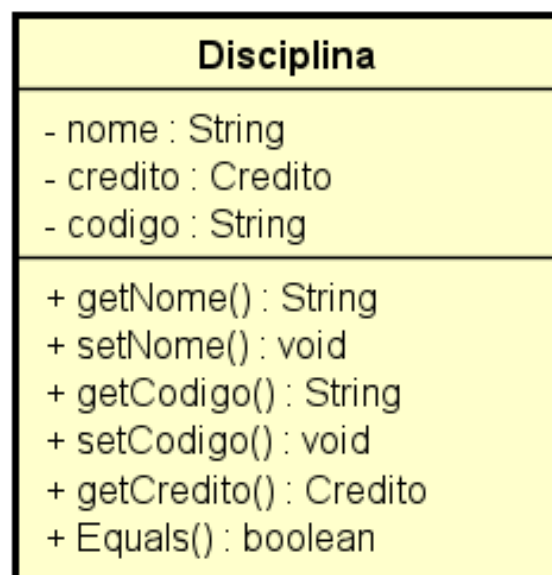


Figura 5 – Diagrama de Classe de Disciplina



## 2.7 Classes Principais - Restrições

Figura 6 – Diagrama de Classe das Restrições (i)

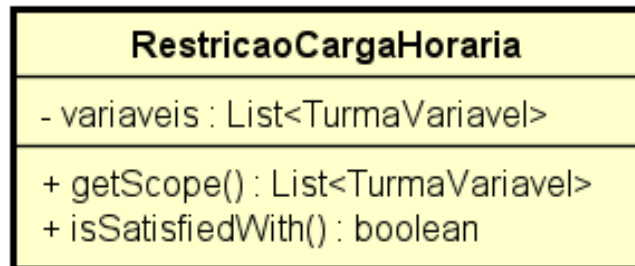


Figura 7 – Diagrama de Classe da Restrição (ii)

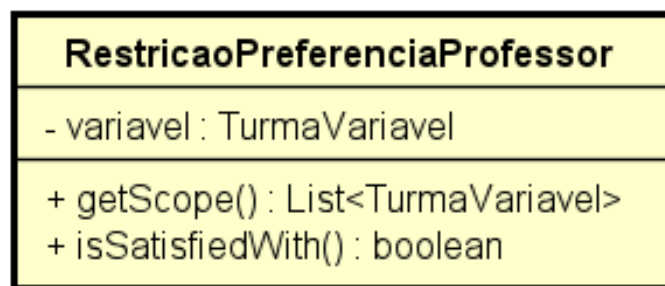


Figura 8 – Diagrama de Classe da Restrição (iii)

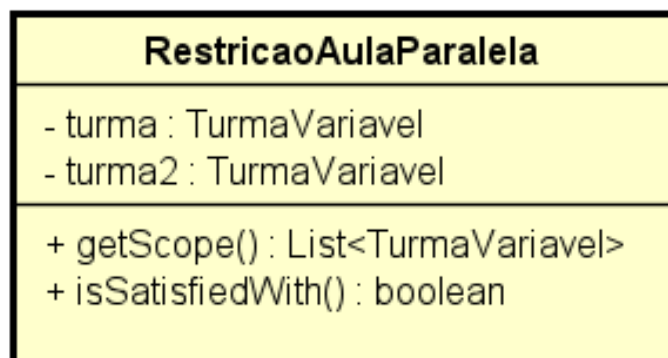
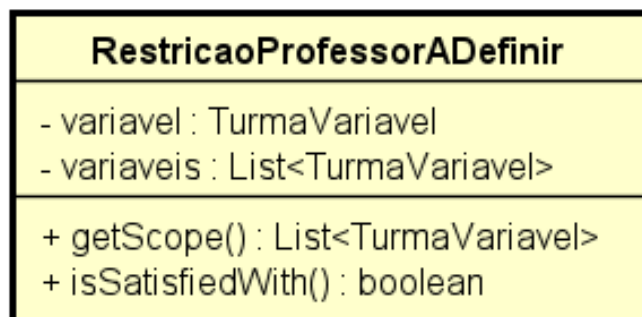


Figura 9 – Diagrama de Classe da Restrição (iv)



# 3

## Conclusão

### 3.1 Resultados Obtidos

Entrada de Disciplinas:

Figura 10 – Tabela de Disciplinas

Disciplinas Inseridas				
Nome	Horarios	Carga Horaria	Código	
COMP01		SEIS	ED1	<a href="#">✖ Excluir</a>
COMP02		SEIS	PAA	<a href="#">✖ Excluir</a>
COMP03		SEIS	IA	<a href="#">✖ Excluir</a>
COMP04		DOIS	BD	<a href="#">✖ Excluir</a>

Entrada de Professores:

Figura 11 – Tabela de Professores

Professores Inseridos		
Nome	Matricula	Disciplinas de preferência
Andre	20170001	[COMP01, COMP02, COMP03]
Marcos	20170002	[COMP02, COMP03, COMP04]
LAIS	20170003	[COMP01, COMP03, COMP04]
Joao	20170004	[COMP01, COMP02, COMP04]
Vitor	20170005	[COMP01, COMP02, COMP03, COMP04]

Resultado:

Figura 12 – Tabela com Resultado Final

Resultado		
Professor	Disciplina	Horário
Andre	COMP01	[2N12, 2N34, 3N12]
Andre	COMP02	[3N34, 4N12, 5N34]
Andre	COMP03	[4N34, 5N12, 6N12]
Marcos	COMP04	[6N34]

## 3.2 Manual de Instalação

Tutorial - Utilizando WildFly manualmente:

1. Faça o download do WildFly.
2. No diretório onde está presente o WildFly, copie o war `AlocadorGUI-0.0.1-SNAPSHOT.war` que está na pasta `AlocadorGUI` para o seguinte pasta: `standalone/deployments`
3. Na pasta do WildFly, abra a pasta `bin` no cmd
4. Execute o arquivo `standalone.bat`
5. Abra o navegador e acesse a seguinte URL: `http://localhost:8080/AlocadorGUI/`
6. Utilize a Aplicação.
7. Caso queira executar novamente a aplicação, basta voltar ao passo 3 e seguir normalmente até o passo 6.

# Referências

NORVIG, P.; RUSSELL, S. *Inteligência artificial: Tradução da 3a Edição*. Elsevier Brasil, 2014. ISBN 9788535251418. Disponível em: <<https://books.google.com.br/books?id=BsNeAwAAQBAJ>>. Nenhuma citação no texto.