

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>3</b>
1.1	Contextualização	3
1.2	Objetivos	3
1.2.1	Objetivos Gerais	4
1.2.2	Objetivos Específicos	4
1.3	Justificativa	4
1.4	Organização do Trabalho	4
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>6</b>
2.1	TimeTable	6
2.2	Heurística	7
2.2.1	Busca Tabu	8
2.2.2	Recozimento Simulado	8
2.2.3	Algoritmos genéticos	9
<b>3</b>	<b>METODOLOGIA</b>	<b>11</b>
3.1	Ferramentas Utilizadas	11
3.1.1	Sistema de Gerenciamento de Banco de Dados	11
3.1.2	Ferramentas Back-end	11
3.1.3	Ferramentas Front-end	12
3.1.4	IDE	13
<b>4</b>	<b>SISTEMA DESENVOLVIDO</b>	<b>14</b>
4.1	Modelo Tratado	14
4.2	Proposta de Solução	14
4.3	O Sistema Desenvolvido	14
4.3.1	Ambiente de Desenvolvimento	14
4.3.2	Modelagem do Sistema	14
4.3.3	Diagrama de Caso de Uso: Sistema	15
4.3.4	Funcionalidades	16
4.3.5	Dados de Entrada	17
4.3.6	Alocação	17
4.3.7	Relatórios	17
<b>5</b>	<b>RESULTADOS OBTIDOS</b>	<b>18</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>19</b>
6.1	Trabalhos futuros	19
	<b>REFERÊNCIAS</b>	<b>20</b>

## **1 INTRODUÇÃO**

O trabalho consiste na distribuição das disciplinas dos diversos cursos de graduação e pós-graduação apresentados pelos colegiados, em salas, através do sistema que será criado todo início de semestre assim que todos os colegiados já tenham enviado suas necessidades para aquele semestre.

Para realização da alocação, faz-se necessário o conhecimento das salas existentes para atender a demanda, além de conhecer sua capacidade e suas características. A partir deste passo, faz-se necessário o levantamento das solicitações enviadas pelos colegiados, para alocar as disciplinas em salas adequadas de forma a atender todas as solicitações de forma otimizada.

O principal problema atualmente é a falta de salas adequadas para a distribuição das disciplinas, uma vez que as turmas atuais estão com um número de alunos matriculados acima da capacidade das salas, devido a este problema o sistema deve propor um relatório de disciplinas que não atenderam a capacidade do prédio para que o responsável pela alocação possa negociar em prédios de outros cursos a alocação das disciplinas que não poderão ser alocadas no prédio que o sistema executou a alocação.

### **1.1 Contextualização**

explicar o problema passo a passo  
falar das restrições

### **1.2 Objetivos**

O tratamento do problema de geração de horários em escolas carece de bons trabalhos na literatura. Apesar de se encontrar ferramentas disponíveis, poucas tratam de maneira eficiente as restrições reais existentes em escolas. Com este trabalho objetiva-se:

### *1.2.1 Objetivos Gerais*

Este trabalho envolve conhecimentos de análise de sistemas e desenvolvimento almejando um sistema capaz tratar e otimizar a execução do problema de alocação de salas de uma universidade. Este trabalho possui um grande valor uma vez que pode facilitar a vida do responsável pela alocação das salas, por se tratar de um trabalho manual, trabalhoso e que para a execução são necessárias em torno de trinta horas que poderiam estar sendo utilizadas para uma tarefa mais importante.

### *1.2.2 Objetivos Específicos*

Objetivos específicos:

- Desenvolvimento do sistema.
- Apresentação de um modelo matemático acerca do problema; - Implementação de um algoritmo que solucione o modelo apresentado considerando as restrições mais comuns para a geração de um horário de qualidade. - Otimizar o tempo do gestor.
- Eficiência na geração dos relatórios.

## **1.3 Justificativa**

## **1.4 Organização do Trabalho**

Este trabalho está definido da seguinte forma, foi dividido em cinco capítulos, sendo este capítulo 1 e mais quatro outros.

O capítulo 2 apresenta o referencial teórico do trabalho, descrevendo os conceitos utilizados para o desenvolvimento do projeto proposto: conceitos de \_\_\_\_\_

No capítulo 3 é apresentada a metodologia do sistema e as tecnologias adotadas para desenvolvimento da solução.

No capítulo 4 iremos descrever e citar detalhadamente as características e propostas de desenvolvimento do sistema ———, proposto para este trabalho.

A conclusão deste trabalho e planos futuros são mostrados no Capítulo 5.

Se tiver anexo explicar cada anexo.

## 2 REFERENCIAL TEÓRICO

### 2.1 TimeTable

(??) Muitas variantes do problema têm sido propostas na literatura, e diferem umas das outras pelo tipo de instituição de ensino envolvida, universidades ou escolas médias, e pelo tipo de restrições impostas ao problema. (??) cita três classes de problemas:

*Examination Timetabling*: seqüenciamento de exames de um conjunto de cursos em uma universidade, evitando exames simultâneos de cursos com estudantes em comum, e espalhando os exames o máximo possível. Segundo (??), apesar da similaridade com o course timetabling, eles se distinguem, sobretudo pela natureza das restrições envolvidas. Entre as restrições típicas deste tipo de problemas, destaca-se: nenhum estudante pode fazer mais do que certo número de exames por dia, exames de certas disciplinas não podem preceder a exames de determinadas disciplinas, alguns exames têm que ser realizados em um mesmo horário.

*School Timetabling*: seqüenciamento semanal das aulas de uma escola, evitando que professores e alunos tenham mais de uma aula simultaneamente. Basicamente, existe um conjunto de turmas, um conjunto de professores e um conjunto de horários reservados para a realização das aulas.

*Course Timetabling*: diz respeito à alocação de aulas de uma universidade típica. Basicamente há um conjunto de disciplinas (Cálculo I, Engenharia de Software, Genética, etc.) e para cada disciplina um número de aulas. Há, também, um conjunto de cursos (Engenharia de Alimentos, Ciência da Computação, Agronomia, etc). Cada curso envolve um conjunto de disciplinas. Os alunos matriculam-se em turmas das disciplinas de seu curso. Uma turma de uma disciplina pode ter estudantes de cursos diferentes. Há, por último, um conjunto de horários disponibilizados para a realização das aulas. O problema, então, trata do seqüenciamento semanal das aulas evitando a simultaneidade de disciplinas e respeitando os horários disponibilizados.

O problema de alocação de salas denotado por PAS é um problema NP-Difícil((??) e (??)).

Segundo (??) diversas instituições universitárias se deparam com o PAS durante o início de cada semestre letivo. Boa parte destas ainda resolve tal problema manualmente, o que torna o processo árduo e demorado, podendo levar vários dias para ser concluído. Ainda segundo (??)

a elaboração de um quadro de horários por esta via pode demandar duas semanas de trabalho em uma escola secundária ou até um mês em uma universidade e esta solução obtida pode ser insatisfatória com respeito a diversos aspectos.

(??) e (??) acreditam que problemas de geração de horários não podem ser completamente automatizados. Há duas justificativas para isso: por um lado, há razões que não podem ser facilmente expressas em um sistema automatizado, que tornam um quadro de horário melhor que o outro. Por outro, uma vez que o espaço de soluções é vasto, a intervenção humana pode conduzir a busca em direção a regiões promissoras, nas quais o sistema, por si só, dificilmente teria condições de chegar.

Uma vez que não é possível encontrar a solução ótima do PAS em tempo razoável, esse problema é normalmente tratado através de técnicas heurísticas e algoritmos aproximativos que apesar de não garantirem encontrar a solução ótima do problema são capazes de retornar uma solução de qualidade em um tempo adequado para as necessidades da aplicação.

## 2.2 Heurística

Segundo (??), as heurísticas são quaisquer métodos de aproximação sem uma garantia formal de seu desempenho. As heurísticas, apesar de não garantirem encontrar a solução ótima para um problema, procuram por soluções consideradas de boa qualidade em um tempo computacional razoável.

De acordo com (??), as heurísticas são quaisquer métodos de aproximação sem uma garantia formal de seu desempenho. Sendo necessárias para implementação de problemas NP Difícil , caso deseje-se resolver tais problemas em um tempo computacional razoável (????).

O termo heurística é derivado do grego heuriskein, que significa descobrir ou achar. Mas o significado da palavra em pesquisa operacional vai um pouco além da raiz etimológica. De um modo geral, o sentido dado ao termo heurística, refere-se a um método de busca de soluções em que não existe qualquer garantia de sucesso.

(??)Em Problemas de Otimização Combinatória, cujo universo de dados é grande, existe um número muito extenso de combinações, tornando inviável a análise de todas possíveis soluções, visto que o tempo computacional para uma enumeração completa seria demasiadamente longo. Neste sentido, têm-se as heurísticas, também conhecidas como algoritmos

heurísticos, que são métodos que compõem uma gama relativamente nova de soluções para Problemas de Otimização Combinatória.

Ressalta-se que dentre as heurísticas merecem especial atenção as chamadas meta-heurísticas que adotam técnicas para amenizar a dificuldade que os métodos heurísticos têm de escapar dos chamados ótimos locais. As meta-heurísticas podem partir em busca de regiões mais promissoras no espaço de soluções. As meta-heurísticas possuem grande abrangência, podendo ser aplicada à maioria dos problemas de otimização combinatória.

Podem-se citar como exemplo as meta-heurísticas: Busca Tabu (*Tabu Search*), Otimização por Colônias de Formigas (*Ant Colony Optimization*), Recozimento Simulado (*Simulated Annealing*) e Algoritmo Genético (*Genetic Algorithm*). Uma heurística é a instanciação de uma meta-heurística, ou seja, a aplicação da mesma em um problema específico de otimização.

### 2.2.1 Busca Tabu

A metaheurística BT foi inicialmente desenvolvida por (??) como uma proposta de solução para problemas de programação inteira. A partir de então, o autor formalizou esta técnica e publicou uma série de trabalhos contendo diversas aplicações da mesma. A metaheurística BT utiliza uma lista contendo o histórico da evolução do processo de busca, de modo a evitar ciclagem; incorpora uma estratégia de balanceamento entre os movimentos aceitos, rejeitados e aspirados; e adota procedimentos de diversificação e intensificação para o processo de busca.

(??) e (??) ressaltam a existência de um mecanismo, relacionado com a Lista Tabu, que anula o status tabu de um movimento, denominado função de aspiração. Se um movimento pode proporcionar uma melhora considerável da função objetivo, então o status tabu é abandonado e a solução resultante é aceita como potencial vizinho.

### 2.2.2 Recozimento Simulado

Técnica de busca local probabilística, proposta originalmente por (??), que se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de

átomos aquecidos. Isto é, conforme (??) em analogia a física da matéria: levando um cristal a sua temperatura de fusão, as moléculas estão desordenadas e se agitam livremente. Ao resfriar-se a amostra de maneira infinitamente lenta, as moléculas vão adquirir a estrutura cristalina estável que tem um nível de energia mais baixo possível. Conforme (??) a analogia com a otimização (combinatória ou não) é bastante direta. Os estados da matéria são as soluções realizáveis, a quantidade objetiva substitui a energia, os estados metaestáveis da matéria sendo ótimos locais e a estrutura cristalina corresponde ao ótimo global. Segundo (??), a temperatura  $T$  assume, inicialmente, um valor elevado  $T_0$  e o procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da função objetivo é mais aceita, isto é, quando o sistema está estável.

Mais informações em (??) e (??).

### 2.2.3 Algoritmos genéticos

Conforme cita (??), os algoritmos genéticos foram introduzidos por (??), com intuito de aplicar a teoria da evolução das espécies elaborada por (??) utilizando os conceitos da evolução biológica como genes, cromossomos, cruzamento, mutação e seleção na computação procurando explicar rigorosamente processos de adaptação em sistemas naturais e desenvolver sistemas artificiais (simulados em computador) que mantenham os mecanismos originais, encontrados em sistemas naturais.

Segundo (??), o processo de evolução executado por um algoritmo genético corresponde a um procedimento de busca no espaço de soluções potenciais para o problema e, como enfatiza (??), esta busca requer um equilíbrio entre dois objetivos aparentemente conflitantes: a procura das melhores soluções na região que se apresenta promissora ou fase de intensificação e a procura de outra região ou exploração do espaço de busca, também conhecida como diversificação.

Ainda segundo (??), os algoritmos genéticos têm se mostrado ferramentas poderosas para resolver problemas onde o espaço de busca é muito grande e os métodos convencionais se mostraram ineficientes.

Mitchel (??) cita que a terminologia biológica é muito importante para a compreensão do funcionamento dos algoritmos genéticos. Eis os principais termos:

- Cromossomo: estrutura que representa uma determinada característica da solução ou



a própria solução;

- Gene: característica particular de um cromossomo. O cromossomo é composto por um ou mais genes.

- Alelo: valor de determinado gene;

- Locus: determinada posição do gene no cromossomo;

- Genótipo: estrutura que codifica uma solução. Um genótipo pode ser formado por um ou mais cromossomos;

- Fenótipo: decodificação ou o significado da estrutura;

- Fitness: significa aptidão. O quanto o indivíduo é apto para determinado ambiente;

As principais características que diferenciam os algoritmos genéticos de métodos tradicionais são (??):

- Parâmetros: os algoritmos genéticos trabalham com a codificação dos parâmetros e não com os parâmetros propriamente;

- Número de soluções: os algoritmos genéticos trabalham com uma população de indivíduos (representando um conjunto de soluções) e não com uma única solução;

- Avaliação das soluções: os algoritmos genéticos utilizam informações de custo ou recompensa penalizando ou premiando determinadas características das soluções;

- Regras: os algoritmos genéticos utilizam regras probabilísticas e não determinísticas;

O algoritmo genético é uma forma da estratégia gerar-e-testar realizando os testes baseados nos parâmetros da evolução biológica. Uma desvantagem notável é a variação dos operadores genéticos do algoritmo em cada problema. Dessa forma, para resolução de determinado problema, torna-se necessário um estudo particular a respeito do mesmo.

O algoritmo genético atua sobre uma população fazendo com que esta evolua de acordo com uma função de avaliação. O funcionamento é iterativo iniciando com a geração de uma população inicial que pode ser aleatória ou não, seguida do processo de avaliação, seleção, cruzamento e mutação, que ocorre a cada iteração até que seja atingido algum critério de parada. Os passos gerais de um algoritmo genético são ilustrados na figura Figura XXXX. Cada passo pode ser realizado de várias maneiras e pode variar de problema para problema (óteo 2002++IMPLóteo 2002++YEAR, ).

Figura Etapas de um Algoritmo Genético Básico

### 3 METODOLOGIA

A resolução deste trabalho

Algoritmos e porque Função Objetivo do algoritmo genetico

Levantamento de requisitos

arquitetura

Dcumentação com o tadeu

Implementação

passo a passo

criou-se o crude a interface e testo o algoritmo de alocação se der tempo teste de unidade.

#### 3.1 Ferramentas Utilizadas

Este trabalho conta com a utilização de tecnologias próprias para o desenvolvimento de sistemas web, foram utilizadas as seguintes ferramentas: Para SGBD foi o utilizado PostgreSQL; No back-end foi utilizado Java e o *framework Play!*; No front-end as tecnologias utilizadas foram HTML, CSS, JavaScript e *framework AngularJS* e a IDE utilizada *Eclipse*.

##### 3.1.1 Sistema de Gerenciamento de Banco de Dados

:(??)

##### 3.1.2 Ferramentas Back-end

Foi escolhida uma linguagem de programação Java por ser orientada a objeto. Também foi escolhido o *Play! framework*, para que o desenvolvimento aconteça de forma mais rápida, fácil e eficiente.

Java foi criada pela Sun Microsystems para desenvolver inovações tecnológicas em

1992, time liderado por James Gosling. O Java utiliza do conceito de máquina virtual, onde existe, entre o sistema operacional e a aplicação, uma camada extra responsável por "traduzir" mas não apenas isso - o que sua aplicação deseja fazer para as respectivas chamadas do sistema operacional, onde ela está rodando no momento. Sua aplicação roda sem nenhum envolvimento com o sistema operacional, sempre conversando apenas com a JVM - Java Virtual Machine (CAELUM, 2013).

Em 2009 a Oracle comprou a Sun, fortalecendo a marca. A Oracle sempre foi, junto com a IBM, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java. Em 2011 surge a versão Java 7 com algumas pequenas mudanças na linguagem (CAELUM, 2013).

!/: Framework is a modern Java (and Scala) web application open-source framework that provides a clean alternative to bloated Enterprise Java stacks. Play has two version Play 1.x (Java Scala) and Play 2.x (Java Scala). Play is a high-productivity Java and Scala web application framework that integrates the components and APIs you need for modern web application development. Play is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications thanks to its reactive model, based on Iteratee IO.(??)

### 3.1.3 Ferramentas Front-end

: é uma linguagem de programação interpretada<sup>2</sup>. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido. É atualmente a principal linguagem para programação client-side em navegadores web. Foi concebida para ser uma linguagem script com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe. Possui suporte à programação funcional e apresenta recursos como fechamentos e funções de alta ordem comumente indisponíveis em linguagens populares como Java e C++. É baseada em ECMAScript padronizada pela Ecma international nas especificações ECMA-262 e ISO/IEC 16262.(??)

Angularjs é um *framework JavaScript* construído e mantido pelo grupo de engenheiros

do Google, ele usa o HTML como uma *template engine*, tudo isso no intuito de fornecer uma solução completa para o cliente-side de sua aplicação. Além disso tem total compatibilidade com as bibliotecas javascript mais utilizadas, como jQuery. É um novo conceito para desenvolvimento de web apps client-site.(MENDES, 2013)

### 3.1.4 IDE

O Eclipse é uma IDE (*integrated development environment*). Diferente de uma RAD(*Rapid Application Development*), onde o objetivo é desenvolver o mais rápido possível através do arrastar-e-soltar do mouse, onde montanhas de código são gerados em background, uma IDE te auxilia no desenvolvimento, evitando se intrometer e fazer muita mágica (CAELUM, 2013).

O Eclipse é a IDE líder de mercado. Formada por um consórcio liderado pela IBM, possui seu código livre. A última versão é a 4.3, mas com qualquer versão posterior a do 3.1 você terá suporte ao Java 5, 6 e 7 (CAELUM, 2013).

Esta IDE foi escolhida devido ao grande reconhecimento mundial, por sua eficiência ao se trabalhar com a linguagem de programação Java, por ser open-source e pela existência de várias ferramentas criadas pela comunidade, para o auxílio no desenvolvimento de softwares.

## **4 SISTEMA DESENVOLVIDO**

### **4.1 Modelo Tratado**

### **4.2 Proposta de Solução**

Será desenvolvido um sistema que otimiza a alocação das salas em até 90% facilitando a vida do gerente. Por se tratar de um problema específico fica difícil encontrar tecnologias disponíveis para a resolução do problema sendo assim necessário o atendimento de um sistema que atenda todas as necessidades exigidas.

### **4.3 O Sistema Desenvolvido**

Descrição sobre o Sistema

#### *4.3.1 Ambiente de Desenvolvimento*

IDE eclipse, sublimeText, Google Chrome, programa DIA para o desenvolvimento dos diagramas(??)

#### *4.3.2 Modelagem do Sistema*

Antes de tudo foi necessária a modelagem do sistema, para que todos os requisitos fossem atendidos de acordo com a necessidade.

Achar alguma referência sobre metodologias de modelagem de dados UML

(??)

Para a análise deste sistema foram desenvolvidos os seguintes diagramas:

(??)

Diagramas de Caso de Uso

Diagramas de classes

Diagramas de Sequência

Diagrama de Atividades

Diagrama de Estados

#### 4.3.3 Diagrama de Caso de Uso: Sistema

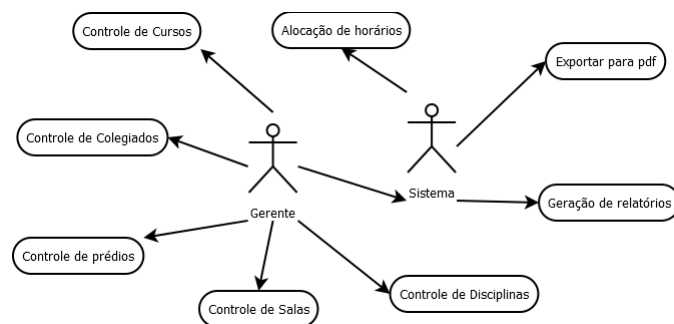
Criar o caso de uso que diz respeito a todas as funcionalidades que o sistema tem de cadastro e manutenção.

(??)

Caso de uso do sistema

(??)

**Figura 1 – Diagrama de caso de uso**



**Fonte: Autor**

##### 4.3.3.1 Diagrama de Atividade: Alocação

Descrever a rotina de atividades da alocação do sistema

##### 4.3.3.2 Diagrama de Classe das controllers

Achar alguma referencia de diagrama de classe.

Imagem do diagrama de classe das controllers

#### **4.3.3.3 Diagrama de Classe das models**

Imagem do diagrama de classe das models

#### **4.3.3.4 Diagrama de Classe das views**

Imagem do diagrama de classe das views

#### **4.3.3.5 Modelagem de Dados**

Achar alguma referencia de modelagem de dados

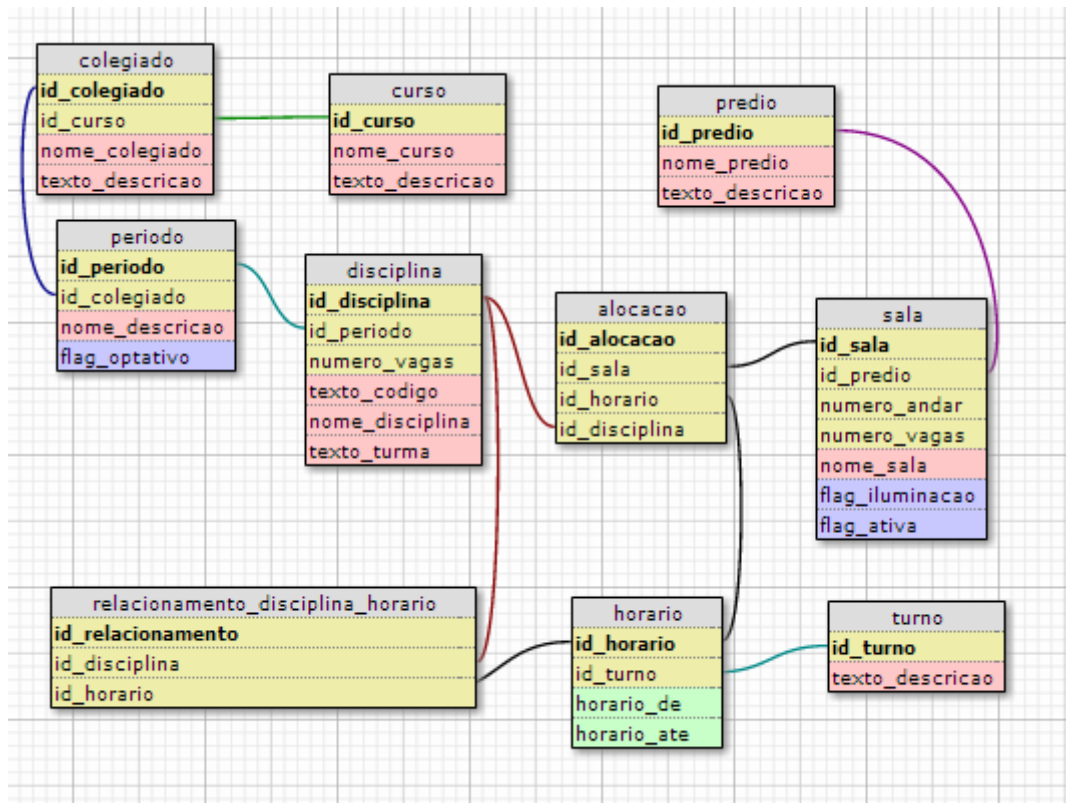
Inserir imagem do modelo

#### **4.3.4 Funcionalidades**

O sistema consiste nas seguintes funcionalidades.

1. Controle de cursos
2. Controle de colegiados
3. Controle de disciplinas
4. Controle de salas
5. Controle de prédios
6. Alocação de horários
7. Geração de relatórios
8. Exportar para pdf

**Figura 2 – Modelagem Banco de Dados**



Fonte: Autor

#### 4.3.5 Dados de Entrada

Como serão inseridas as informações, e quais são os dados de entrada Informados pelo gerente.

#### 4.3.6 Alocação

Processa os dados de alocação

#### 4.3.7 Relatórios

Geração dos relatorios determinados na analise do sistema, todos os relatorios podem ser exportados para pdf



## **5 RESULTADOS OBTIDOS** Depois do sistema implementado

Entrada processamento e saída

ajuste do algoritomo de alocação

## **6 CONSIDERAÇÕES FINAIS**

\*Relembra ro problema e comprar como o resultadoo bjetido.

Discussão dos resultados obtidos na pesquisa, onde se verificam as observações pessoais do autor. Poderá também apresentar sugestões de novas linhas de estudo. A conclusão deve estar de acordo com os objetivos do trabalho. A conclusão não deve apresentar citações ou interpretações de outros autores.

### **6.1 Trabalhos futuros**

Criar um DW para geração dos relatorios de acordo com a dimensão escolhida.

Utilização de outros algoritimos para a resolução do problema ex. algoritimos evolutivos formiga entre outros.

Pegar o feed back do usuario para melhoria na interface, e do algoritimo.

## REFERÊNCIAS

CAELUM. *Apostila do curso FJ-11 - Java e Orientação a Objetos*. 2013. Disponível em: <<http://www.caelum.com.br/apostila-java-orientacao-objetos/>>. Acesso em: 29 set. 2013.

MENDES, W. *AngularJS um framework para facilitar sua vida*. 2013. Disponível em: <<http://www.slideshare.net/WilsonMendes/angularjs-um-framework-para-facilitar-sua-vida/>>. Acesso em: 29 set. 2013.

[titletoc,toc,page]appendix