

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Bacharelado em Sistemas de Informação

Alexandre Gonzaga Mendes

**DEFINIR O NOME**  
**Subtítulo do Trabalho**

Belo Horizonte  
2013

Alexandre Gonzaga Mendes

**DEFINIR O NOME**  
**Subtítulo do Trabalho**

Monografia apresentada ao programa de Bacharelado em Sistemas de Informação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: João Caram Adriano

Belo Horizonte  
2013

Alexandre Gonzaga Mendes

**DEFINIR O NOME**  
**Subtítulo do Trabalho**

Monografia apresentada ao programa de Bacharelado em Sistemas de Informação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

---

João Caram Adriano

---

Nome do Avaliador 1

---

Nome do Avaliador 2

Belo Horizonte, DD de MM de 2013

*Dedicatória: Página onde o autor presta homenagem a uma ou mais pessoas. O layout desta página fica a critério do autor, mas o tipo e tamanho de letras são definidos pela ABNT.*

## **AGRADECIMENTOS**

Agradecimentos a pessoas que contribuíram para o desenvolvimento do trabalho. Agradecimentos a pessoas que contribuíram para o desenvolvimento do trabalho.

*Epígrafe: Pensamentos retirados de um livro, uma música, um poema, normalmente relacionados ao tema do trabalho. Deve ser elaborada conforme norma NBR 10520/2002. Apresentação de citações em documentos. Se desejar, a epígrafe pode ser grafada em itálico. Ao final do trabalho deve-se fazer a referência completa da publicação de onde a epígrafe foi retirada.*

## **RESUMO**

Apresentação concisa dos pontos relevantes do texto. Deve ressaltar o objetivo, o método, resultados e conclusões do trabalho. Deve-se utilizar o verbo na voz ativa ou terceira pessoa do singular. O resumo não deve conter citações ou indicações bibliográficas.

Palavras-chave: Ao final do resumo deve-se elaborar palavras-chave representativas do conteúdo do trabalho, separadas entre si por um ponto.

## **ABSTRACT**

Versão do resumo em idioma de divulgação internacional. Deve ser a tradução literal do resumo em português e apresentar palavras-chave no mesmo idioma, logo abaixo do texto, separadas entre si por um ponto.

**Keywords:** O resumo em língua estrangeira também deve conter palavras-chave representativas do conteúdo do trabalho, separadas entre si por um ponto.



## LISTA DE FIGURAS

FIGURA 1 – Diagrama de caso de uso . . . . .	19
FIGURA 2 – Modelagem Banco de Dados . . . . .	20

## **LISTA DE TABELAS**

## **LISTA DE SIGLAS**

S1 – Sigla 1

S2 – Sigla 2

S3 – Sigla 3

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Justificativas e Objetivos	13
1.2	Organização do Trabalho	14
<b>2</b>	<b>REVISÃO LITERARIA</b>	<b>15</b>
2.1	Descrição do problema	15
2.2	Complexidade do problema	15
2.3	Métodos	15
<b>3</b>	<b>METODOLOGIA</b>	<b>16</b>
3.1	Modelo Tratado	16
3.2	Proposta de Solução	16
3.3	O Sistema Desenvolvido	16
3.3.1	<i>Linguagens e Ferramentas Utilizadas</i>	16
3.3.2	<i>Modelagem do Sistema</i>	18
3.3.3	<i>Interface</i>	20
3.3.4	<i>Funcionalidades</i>	20
3.3.5	<i>Dados de Entrada</i>	21
3.3.6	<i>Alocação</i>	21
3.3.7	<i>Relatórios</i>	21
3.3.8	<i>Considerações Finais do Capítulo</i>	21
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>22</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>23</b>
5.1	Trabalhos futuros	23
	<b>REFERÊNCIAS</b>	<b>24</b>
<b>6</b>	<b>APÊNDICES</b>	<b>25</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 Justificativas e Objetivos**

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.

Com este trabalho objetiva-se:

- Desenvolvimento do sistema.
- Apresentação de um modelo de algoritmo que atenda o problema proposto.
- Otimizar o tempo do gestor.
- Eficiência na geração dos relatórios.

## 1.2 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.

## **2 REVISÃO LITERARIA**

### **2.1 Descrição do problema**

Escrever sobre os objetivos e as justificativas

### **2.2 Complexidade do problema**

Este trabalho está definido da seguinte forma. Capítulo 1 Capítulo 2 Capítulo 3 Capítulo

4

### **2.3 Métodos**

## **3 METODOLOGIA**

### **3.1 Modelo Tratado**

### **3.2 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.

### **3.3 O Sistema Desenvolvido**

Descrição sobre o Sistema

#### *3.3.1 Linguagens e Ferramentas Utilizadas*

Descrição sobre o que o que será abordada nesta subseção

##### **3.3.1.1 Linguagens de Programação e Frameworks**

JavaScript 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-2623 e ISO/IEC 16262.(??)

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.(??)

The Play! 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.

(??)

AngularJS is an open-source JavaScript framework, maintained by Google, that assists with running single-page applications. Its goal is to augment browser-based applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier. The library reads in HTML that contains additional custom tag attributes; it then obeys the directives in those custom attributes, and binds input or output parts of the page to a model represented by standard JavaScript variables. The values of those JavaScript variables can be manually set, or retrieved from static or dynamic JSON resources.

(??)

### 3.3.1.2 Sistema Gerenciador de Banco de Dados

Achar alguma referencia sobre o postgresql

(??) Postgreesql

### 3.3.1.3 Ambiente de Desenvolvimento

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

### 3.3.2 Modelagem do Sistema

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

Achar alguma referencia sobre metodologias de modelagem de dados UML

(??)

Para a analise deste sistema foram desenvolvidos os seguintes diagramas:

(??)

Diagramas de Caso de Uso

Diagramas de classes

Diagramas de Seqüência

Diagrama de Atividades

Diagrama de Estados

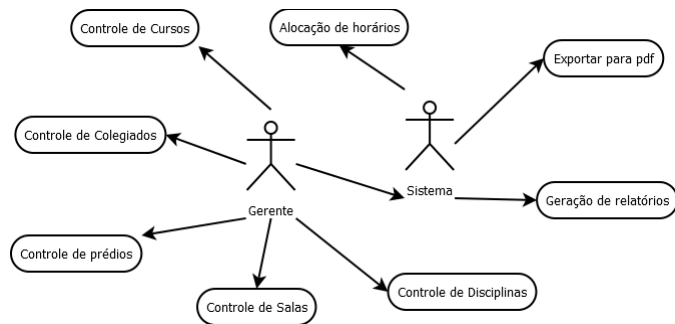
#### 3.3.2.1 Diagrama de Caso de Uso: Sistema

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****3.3.2.2 Diagrama de Atividade: Alocação**

Descrever a rotina de atividades da alocação do sistema

**3.3.2.3 Diagrama de Classe das controllers**

Achar alguma referencia de diagrama de classe.

Imagem do diagrama de classe das controllers

**3.3.2.4 Diagrama de Classe das models**

Imagem do diagrama de classe das models

**3.3.2.5 Diagrama de Classe das views**

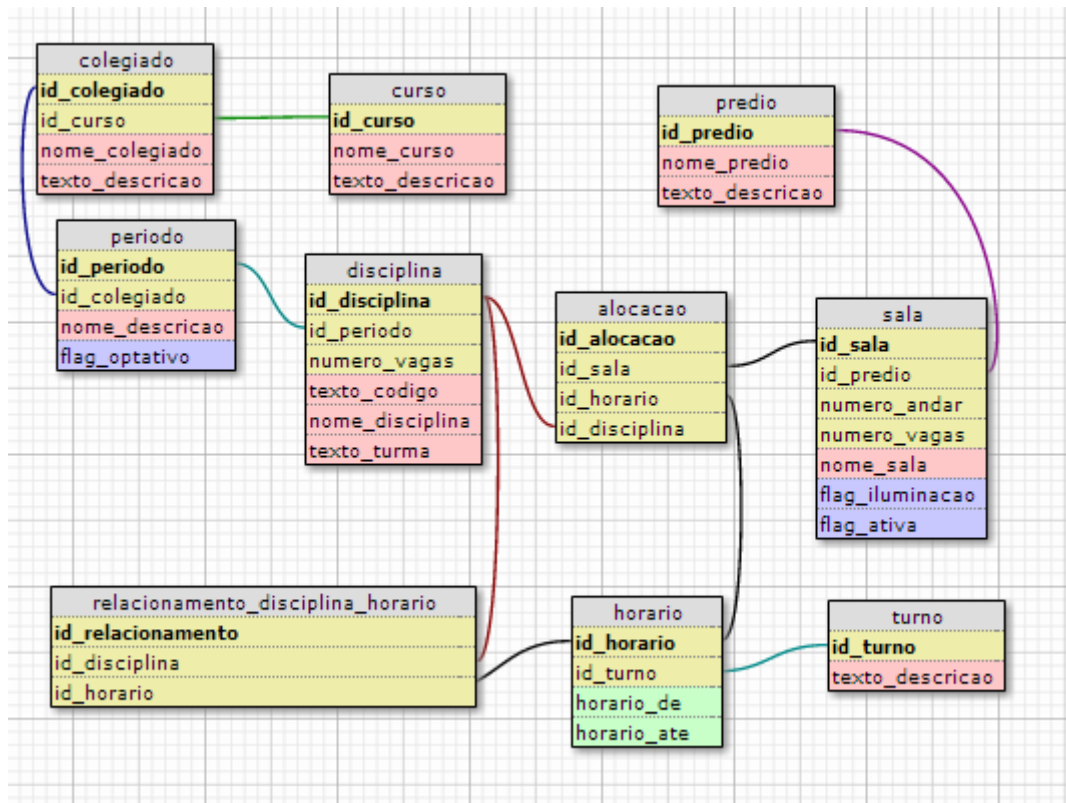
Imagem do diagrama de classe das views

**3.3.2.6 Modelagem de Dados**

Achar alguma referencia de modelagem de dados

Inserir imagem do modelo

**Figura 2 – Modelagem Banco de Dados**



Fonte: Autor

### 3.3.3 Interface

Achar alguma referencia sobre interface

Twitter bootstrap.

### 3.3.4 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

### *3.3.5 Dados de Entrada*

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

### *3.3.6 Alocação*

Processa os dados de alocação

### *3.3.7 Relatórios*

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

### *3.3.8 Considerações Finais do Capítulo*

Considerações finais do capitulo

## **4 RESULTADOS E DISCUSSÕES**

## **5 CONCLUSÃO**

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.

### **5.1 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



## **6 APÊNDICES**