

**PROJETO DE GRADUAÇÃO**

**Proposta de Mapeamento e Alocação de Salas do  
ENE via WEB e ANDROID**

Por,

**Luana Daher Lopes**

**Thamara Guimarães Silva**

**Brasília, 5 Julho de 2016**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Departamento de Engenharia Elétrica

## PROJETO DE GRADUAÇÃO

# Proposta de Mapeamento e Alocação de Salas do ENE via WEB e ANDROID

POR,

**Luana Daher Lopes**

**Thamara Guimarães Silva**

Relatório submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Engenharia de Redes de Comunicação.

Banca Examinadora

Prof. Georges Daniel Amvame Nze, Dr., ENE/UnB

*Orientador*

Prof. Alexandre Ricardo Soares Romariz, Dr., ENE/UnB

*Examinador Interno*

Prof. Rafael Timóteo de Sousa Jr., Dr., ENE/UnB

*Examinador Interno*

## **Dedicatória (s)**

*Dedico esse projeto à Deus, pois sem ele nada somos, à minha mãe que dedicou e dedica toda sua vida para que pudesse me tornar alguém, com a melhor educação, formação acadêmica e construção de ser humano possível e, logicamente, à Anna Carolina, pois do princípio ao fim esse projeto foi pensado para ela.*

*Thamara Guimarães Silva*

*Dedico esse projeto aos alunos e aos trabalhadores do departamento de Engenharia Elétrica da UnB.*

*Luana Daher Lopes*

## **Agradecimentos**

*Agradeço primeiro à minha família por todo o apoio, incentivo carinho e paciência durante esse processo, a todos os meus amigos que sempre estiveram ao meu lado e me ajudaram a trilhar meu caminho em especial a Thamara Guimarães que esteve ao meu lado nesse projeto e ao Dione Eleoterio pela paciência e apoio incondicional. Agradeço também aos professores que me permitiram chegar até essa etapa, em especial ao nosso orientador Georges Daniel pelo ânimo e dedicação ao nosso projeto. Muito obrigada a todos, o apoio de vocês foi essencial para minha vida e para esse projeto.*

*Luana Daher Lopes*

*Agradeço primeiramente à Deus por ter me dado saúde e força para superar todos os desafios dessa jornada. Ao professor e orientador Georges Daniel por todo empenho, paciência e dedicação de sempre. À minha mãe, que é minha inspiração, heroína e exemplo de ser humano, que sempre me deu todo apoio necessário e acreditou no meu potencial mesmo quando eu não acreditava. A todos os meus amigos que, de alguma forma, fizeram parte dessa conquista e principalmente ao meu namorado Alysson que soube me compreender com todo amor nos piores e melhores momentos desse processo. Por fim, agradeço à minha amiga e companheira de curso Luana Daher por me conceder a honra de trabalharmos juntas no desenvolvimento desse projeto que conclui uma das etapas mais importantes de nossas vidas. MUITÍSSIMO obrigada. Nada disso aconteceria se não fossem vocês.*

*Thamara Guimarães Silva*

---

## RESUMO

Este trabalho tem como objetivo a elaboração de um sistema web para facilitar e otimizar atividades desenvolvidas pelo departamento de Engenharia Elétrica (ENE) da Universidade de Brasília (UnB). O sistema intenta realizar atividades básicas do departamento para amenizar a sobrecarga dos funcionários e reduzir a utilização de insumos que podem contribuir para a degradação ambiental, uma vez que a página web desenvolvida pode ser utilizada como substituta para atividades realizadas manualmente e em papel, como a reserva de salas e as inscrições para monitorias.

Todo o processo terá por base a lista de oferta das disciplinas do departamento, otimizando as demandas dos usuários através do acesso de informações básicas e fundamentais. Um exemplo seria a disponibilidade de tabelas com os horários das disciplinas, dos professores, das salas e dos laboratórios, que facilmente poderão ser acessados pelos usuários através de uma conexão à internet independentemente do local que estejam e a qualquer momento do dia, possibilitando uma melhor organização das atividades exercidas.

Esse sistema foi desenvolvido utilizando linguagens voltadas para web com o objetivo de obter um melhor desempenho. Todas as etapas do seu desenvolvimento, tais como diagramas de casos de uso, UML e algoritmos principais, assim como os seus resultados e análises foram feitos com base em testes da funcionalidade do sistema para as atividades propostas. Serão apresentadas ao decorrer deste trabalho as etapas de implementação do projeto desde o levantamento das necessidades do departamento e pesquisas de tecnologias que possibilitariam a realização do sistema até o desenvolvimento da página web e aplicativo Android.

A implementação do sistema proposto foi feita através da construção de diversos algoritmos baseados nas linguagens de programação HTML, PHP e JavaScript utilizado para desenvolver códigos de lógicas, CSS para layout das páginas e MySQL para banco de dados.

O principal algoritmo do sistema é o que monta os horários dinamicamente através dos dados armazenados no banco, conforme a pesquisa realizada pelo usuário. Observando que os dados do banco também são dinâmicos e podem ser atualizados com auxílio de um dos principais algoritmos do sistema, que dá permissão ao administrador para atualizá-lo através de um arquivo em CSV.

Os resultados obtidos foram uma página web que atende de forma otimizada a necessidade do departamento como um todo, englobando alunos, funcionários e corpo docente; e um aplicativo Android interativo, leve e que cumpre com seu objetivo principal de facilitar o acesso dos usuários de dispositivos móveis às informações dessa plataforma web.

Como trabalhos futuros, deseja-se melhorar as funções existentes agregando maiores restrições para seu melhor funcionamento e conseqüentemente aumentar o nível de segurança das informações contidas nele, bem como realizar a implementação de novas funções para a página web e aplicativo Android como, por exemplo, reserva de salas.

Palavras-chave: Sistema web, Automatização, Horários, Departamento, Android.

---

## ABSTRACT

This work aims at the development of a web system to facilitate and optimize activities developed by the Electrical Engineering Department (ENE) at the University of Brasilia (UnB). The system performs basic activities of the department in order to ease the burden of the staff and to reduce the use of inputs that can contribute to environmental degradation as the developed web page can be used as a substitute for activities performed manually and on paper, as booking rooms and registration for tutoring.

The whole process will be based on the list of offer of the disciplines of the department, optimizing the demands of users through the basic and fundamental information access. An example would be the availability of weekly timetables of disciplines, teachers, rooms and laboratories, which, through the Internet, can be easily accessed by users, regardless of where they are and at any time, allowing for planning .

This system was developed using languages focused on web, in order to obtain better performance, all stages of its development, such as use case diagram, UML and main algorithms, as well as its results and analyzes were made based on system functionality tests for the proposed activities. It will be presented to the course of this work the project implementation steps since the lifting of the department's needs and research technologies that enable the implementation of the system to the web page development and Android application.

The implementation of the proposed system was made by constructing various algorithms based on the HTML programming language, PHP, JavaScript and used to develop logical codes, CSS layout for the page and to MySQL database.

The main system algorithm is riding schedules dynamically using data from the database, according to research done by the user. Noting that the database is also dynamic and can also be upgraded with the help of a key system algorithms, which allows the system administrator to update it via a CSV file.

The results were a web page that serves optimally the needs of the department as a whole, comprising students, staff and faculty, and an interactive Android application, lightweight and meets with its main objective to facilitate user access devices mobile web platform to this information.

As future work, we want to improve existing functions by adding more restrictions to better your improve its operation by increasing the information security level and also carry out the implementation of new features to the website and Android app, for example, the reservation rooms.

# SUMÁRIO

<b>CAPÍTULO 1</b> .....	<b>1</b>
<b>INTRODUÇÃO</b> .....	<b>1</b>
1.1    MOTIVAÇÃO .....	1
1.2    OBJETIVOS .....	2
1.3    ESTRUTURA DO TEXTO .....	2
<b>CAPÍTULO 2</b> .....	<b>4</b>
<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>4</b>
2.1    SISTEMA WEB .....	4
2.2    INFRAESTRUTURA INSTALADA - SERVIDOR LAMP .....	5
2.2.1    Linux .....	5
2.2.2    Apache .....	5
2.2.3    MySQL .....	6
2.2.4    PHP .....	7
2.2.5    phpMyAdmin .....	7
2.3    LINGUAGENS UTILIZADAS .....	7
2.3.1    HTML .....	8
2.3.2    PHP .....	8
2.3.3    CSS .....	9
2.3.4    JavaScript .....	9
2.4    ANDROID .....	10
2.5    ANDROID STUDIO .....	11
2.6    UML .....	11
2.6.1    Diagrama de Caso de Uso .....	12
2.6.2    Diagrama de Classe .....	12
<b>CAPÍTULO 3</b> .....	<b>13</b>
DESENVOLVIMENTOS RECENTES .....	13
<b>CAPÍTULO 4</b> .....	<b>15</b>
<b>ETAPAS DO PROJETO</b> .....	<b>15</b>
4.1    DELIMITAÇÃO DO TEMA .....	15
4.1.1    Pesquisa das Necessidades do Departamento .....	17
4.1.2    Pesquisa de Linguagens de Programação .....	19
4.1.3    Preparar Máquina .....	19
4.1.4    Criar Bancos de Dados .....	22
4.1.5    Inserção de dados no Banco de Dados .....	24
4.1.6    Desenvolvimento dos Algoritmos do Sistema .....	26
4.1.7    Desenvolvimento da Plataforma Web .....	32
4.1.8    Desenvolvimento do Aplicativo Android .....	37
<b>CAPÍTULO 5</b> .....	<b>41</b>
<b>RESULTADOS</b> .....	<b>41</b>
5.1    PLATAFORMA DEPON .....	41
5.1.1    Login .....	48
5.2    BANCO DE DADOS .....	51
5.3    APLICATIVO ANDROID .....	52
<b>CAPÍTULO 6</b> .....	<b>59</b>
<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>59</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>61</b>
<b>APÊNDICE A</b> .....	<b>63</b>
<b>A1 - INSTALAÇÃO DA PILHA LAMP</b> .....	<b>63</b>

# LISTA DE FIGURAS

- FIGURA 2. 1 – PORCENTAGEM DE UTILIZAÇÃO DE SERVIDORES WEB AO REDOR DO MUNDO, PESQUISA DA W3TECHS.COM DE JUNHO DE 2014
- FIGURA 2. 2 - RANKING DOS SGBDS MAIS POPULARES SEGUNDO DB-ENGINES EM JUNHO DE 2016
- FIGURA 4. 1 – ETAPAS DO PROJETO
- FIGURA 4. 2 - TELA DE VERIFICAÇÃO DO FUNCIONAMENTO DO SERVIDOR APACHE2.
- FIGURA 4. 3 - TELA DA VERIFICAÇÃO DO FUNCIONAMENTO DO PHP.
- FIGURA 4. 4 - TELA DA VERIFICAÇÃO DO FUNCIONAMENTO O PHPMYADMIN.
- FIGURA 4. 5 – TELA DE CRIAÇÃO DE BANCO DE DADOS
- FIGURA 4. 6 – TELA DE CRIAÇÃO DE TABELAS NO PHPMYADMIN.
- FIGURA 4. 7 – DIAGRAMA DAS TABELAS CRIADAS PARA O PROJETO.
- FIGURA 4. 8 – TRECHO DO CÓDIGO RESPONSÁVEL PELA INSERÇÃO DE DADOS EM UMA TABELA A PARTIR DA LEITURA DE UM ARQUIVO.
- FIGURA 4. 9 – TRECHO DO CÓDIGO RESPONSÁVEL PELA INSERÇÃO DE DADOS DIGITADOS PELO USUÁRIO NA TABELA DO BANCO DE DADOS.
- FIGURA 4. 10 – FLUXOGRAMA DO PROCESSO DE ALTERAÇÃO DA TABELA OFERTA.
- FIGURA 4. 11 – DIAGRAMA DE CLASSES DA OFERTA.
- FIGURA 4. 12 – DIAGRAMA DE CASOS DE USO DO PROCESSO DE MONITORIA.
- FIGURA 4. 13 – DIAGRAMA DE CLASSES REFERENTE AOS HORÁRIOS.
- FIGURA 4. 14– FLUXOGRAMA DA MONTAGEM DE HORÁRIOS.
- FIGURA 4. 15– DIAGRAMA DE CLASSES DO CADASTRO.
- FIGURA 4. 16– PRINT DO TERMINAL COM AS CLASSES UTILIZADAS NO PROJETO.
- FIGURA 4. 17– DIAGRAMA DE CASOS DE USO PARA USUÁRIOS GERAIS DO SISTEMA.
- FIGURA 4. 18– DIAGRAMA DE CASOS DE USO PARA USUÁRIOS DO TIPO ALUNO.
- FIGURA 4. 19– DIAGRAMA DE CASOS DE USO PARA USUÁRIO DO TIPO PROFESSOR.
- FIGURA 4. 20– DIAGRAMA DE CASOS DE USO PARA USUÁRIO DO TIPO ADMINISTRADOR.
- FIGURA 4. 21 – DIAGRAMA DE CASOS DE USO DO APLICATIVO ANDROID.
- FIGURA 4. 22- DEFINIÇÃO DO TAMANHO DO WEBVIEW PARA A APLICAÇÃO ATRAVÉS DO ANDROID STUDIO.
- FIGURA 4. 23– CÓDIGO PARCIAL REFERENTE A IMPLEMENTAÇÃO DO BOTÃO OFERTA DO MENU, COM UTILIZAÇÃO DO WEBVIEW.
- FIGURA 5. 1– PÁGINA INICIAL.
- FIGURA 5. 2 – PÁGINA OFERTA.

FIGURA 5. 3 – PÁGINA FORMULÁRIOS.

FIGURA 5. 4 – PÁGINA MAPA DE SALAS COM EXEMPLO DE HORÁRIO REFERENTE A SALA.

FIGURA 5. 5– PÁGINA CONTATO.

FIGURA 5. 6– PÁGINA LOGIN.

FIGURA 5. 7– PÁGINA QUEM SOMOS.

FIGURA 5. 8– PÁGINA CADASTRO.

FIGURA 5. 9– PÁGINA HORÁRIOS.

FIGURA 5. 10– HORÁRIO REFERENTE A DISCIPLINA CIRCUITOS ELÉTRICOS 1.

FIGURA 5. 11– PÁGINA CORPO DOCENTE.

FIGURA 5. 12– PÁGINA ADMINISTRADOR.

FIGURA 5. 13– PÁGINA ALUNO

FIGURA 5. 14– PÁGINA PROFESSOR.

FIGURA 5. 15– BANCO DE DADOS E AS TABELAS QUE O COMPÕEM CRIADAS ATRAVÉS DO GERENCIADOR DE BD PHPMYADMIN.

FIGURA 5. 16– TELA INICIAL DO APP E A APRESENTAÇÃO DE SEU MENU.

FIGURA 5. 17– TELA DO APP AO SELECIONAR A OPÇÃO MAPA DE SALAS E TELA REFERENTE À SALA AUDITÓRIO.

FIGURA 5. 18– TELA DO APP AO SELECIONAR A OPÇÃO PROFESSORES HORÁRIOS E TELA REFERENTE AO PROFESSOR GEORGES.

FIGURA 5. 19 – TELA DO APP AO SELECIONAR A OPÇÃO DISCIPLINAS HORÁRIOS E TELA REFERENTE À DISCIPLINA FUNDAMENTOS DE REDES.

FIGURA 5. 20 – TELA DO APP AO SELECIONAR A OPÇÃO CORPO DOCENTE.

FIGURA 5. 21 – TELA DO APP AO SELECIONAR A OPÇÃO CONTATOS.

FIGURA 5. 22– TELA DO APP AO SELECIONAR A OPÇÃO QUEM SOMOS.

# LISTA DE SÍMBOLOS

## Siglas

ABNT	Associação Brasileira de Normas Técnicas
LAMP	Linux Apache MySQL PHP
PHP	Hypertext Preprocessor
SQL	Structured Query Language
PDF	Portable Document Format
SO	Sistema Operacional
GNU	Gnu's Not UNIX
HTTP	Hypertext Transfer Protocol
SGBD	Sistema de Gerenciamento de Banco de Dados
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	Java Script
IDE	Ambiente de Desenvolvimento Integrado
API	Application Programming Interface
UML	Unified Modeling Language
APP	Aplicativo
CSV	Separados por Virgulas
ENE	Engenharia Elétrica
UnB	Universidade de Brasília
CSV	Separados por Virgulas
GPL	General Public License
ASF	Apache Software Foundation
CERN	European Council for Nuclear Research
iOS	Sistema Operacional iPhone
TV	Televisão
IDE	Ambiente de Desenvolvimento Integrado
SDK	Software Development Kit
FT	Faculdade de Tecnologia
SG11	Serviços Gerais 11
UAC	Universidade dos Açores

# Capítulo 1

## Introdução

*Este capítulo apresenta a contextualização do trabalho, os objetivos e, ao final, a disposição dos assuntos abordados ao longo deste trabalho de graduação.*

### 1.1 Motivação

Os departamentos são responsáveis pela esfera administrativa dos cursos oferecidos pela universidade. Cada curso possui seu respectivo departamento que é responsável por atender às demandas dos cursos de graduação e pós-graduação por professores, salas de aula, grades horárias, vagas em disciplinas, etc [1]. Uma disciplina de um departamento não está restrita apenas ao seu curso de origem, ou seja, também pode servir para outros cursos, algumas disciplinas são obrigatórias para vários cursos, o que acarreta um desafio maior ao departamento, o qual precisa levar em consideração os horários de outras disciplinas obrigatórias para os estudantes a fim de viabilizar que esse possa cursar as matérias de acordo com o fluxo[3].

Cada departamento possui Chefes e Vice Chefes eleitos pela comunidade acadêmica a cada dois anos e com direito à reeleição. Além desses cargos principais existem mais funções que variam de acordo com os departamentos para serem realizadas da melhor maneira possível. Basicamente os departamentos de universidades são responsáveis por estabelecer a relação aluno – universidade, enfatizando o compromisso social da universidade enquanto instituição pública sempre sob a supervisão geral da Reitoria[2][3].

O Departamento de Engenharia Elétrica da Universidade de Brasília é um exemplo de departamento que não se restringe apenas a seus cursos nativos e é responsável atualmente por quatro engenharias (Engenharia Elétrica, Engenharia de Redes de Comunicações, Engenharia da Computação e Engenharia Mecatrônica)[4] atendendo aos alunos dessas engenharias que precisam cursar disciplinas obrigatórias as quais pertencem a ele. Além disso também há demanda de alunos de toda a universidade que têm a opção de cursar determinadas disciplinas, desde que preencham o pré-requisito, se houver.

Dada a quantidade de alunos que o departamento necessita atender foi constatado, ao longo desses anos de graduação, a necessidade de viabilizar o acesso a diversos serviços de forma otimizada, visando valorizar tempo, dinheiro e principalmente o meio ambiente já que o projeto em questão poderá diminuir consideravelmente a quantidade de papel desperdiçado.

Além disso, as alunas desejavam realizar como projeto final de graduação algo que realmente pudesse ser útil, viável e implementado, ou seja, que não se limitasse a apenas simulações, mas que facilitasse o cotidiano dos alunos e servidores da instituição.

## **1.2 Objetivos**

Esse projeto possui como principal objetivo o desenvolvimento de uma plataforma web capaz de realizar diversas atividades exercidas pelo ENE, de forma a amenizar a sobrecarga de seus funcionários e reduzir a utilização de insumos que contribuem para a degradação ambiental visto que a página web desenvolvida pode ser utilizada como substituta para atividades que atualmente são realizadas manualmente e em papel.

Nessa plataforma, o aluno poderá inscrever-se em monitorias, ter acesso a informações pertinentes da universidade, checar horários de disciplinas, verificar o mapeamento de salas e conferir se foi aceito como monitor. O professor poderá verificar e selecionar os alunos que se candidataram à monitoria, além de acessar informações específicas aos docentes. O administrador pode inserir informações para cada tipo de usuário, verificar monitores selecionados, realizar cadastros de outros administradores, professores e também atualizar a lista de oferta. Além disso todos que visitarem a plataforma podem verificar as disciplinas alocadas em cada sala, através de uma busca, ou seja, digita-se a sala desejada e faz a verificação das disciplinas que lá serão ministradas assim como os seus horários, professores ou disciplinas; podem, também, visualizar informações básicas do departamento.

Por fim, há o desenvolvimento de um aplicativo Android, o qual tem o objetivo de auxiliar, principalmente os alunos, a visualizarem as alocações de salas do departamento, obterem informações básicas sobre professores e acerca da oferta de disciplinas, a fim de potencializar o acesso às funções básicas da plataforma.

## **1.3 Estrutura do Texto**

Este trabalho está apresentado nos seguintes capítulos:

- Capítulo 2: Fundamentação Teórica.

Neste capítulo são definidos e analisados conceitos essenciais para a compreensão do projeto como um todo.

- Capítulo 3: Desenvolvimentos Recentes

Neste capítulo são mostrados outros trabalhos desenvolvidos com temáticas semelhantes ao projeto executado.

- Capítulo 4: Etapas do Projeto

Neste capítulo são apresentadas as etapas para a realização do projeto, bem como as decisões tomadas no que desrespeita a escolha de softwares.

- Capítulo 5: Resultados

Aqui são expostos os resultados obtidos com o desenvolvimento do projeto, tanto da plataforma web quanto do aplicativo.

- Capítulo 6: Conclusão e Trabalhos Futuros

É exposta a validade das soluções propostas e os possíveis trabalhos futuros.

## Capítulo 2

### Fundamentação Teórica

*Este capítulo aborda temas relevantes ao entendimento das tecnologias utilizadas nesse projeto*

#### 2.1 Sistema Web

O sistema web é um sistema que está disponível na rede, ou seja, pode ser acessado de qualquer local e a qualquer momento, necessitando apenas de acesso à Internet. Devido a esses benefícios, as utilizações desse tipo de sistema estão cada vez mais inseridas na realidade das pessoas, pois ao serem comparados a ambientes físicos que provêm o mesmo tipo de serviço, trazem maiores facilidades para administradores e clientes.

É projetado especificadamente para realizar uma função ou atender as necessidades de uma empresa ou instituição, de modo que os usuários tenham mais simplicidade ao utilizar seus serviços, até porque os erros com maior incidência são levados em consideração no projeto para que não ocorram.

A mobilidade sem dúvida é a maior vantagem desses sistemas, pois está alocado em um servidor, podendo ser acessado pelos usuários de qualquer local e horário mediante a utilização de aparelhos com acesso à internet: como *tablets*, celulares, *notebooks* e computadores. Alguns sistemas possuem versões *mobile* com exatamente as mesmas informações, porém com resolução apropriada para visualização em dispositivos móveis com tamanho de tela reduzido.

Esses sistemas oferecem também mais segurança aos usuários, pois funcionam de forma transparente, ou seja, o usuário não sabe de que forma ou local seus dados estão sendo processados. Essa transparência auxilia na questão da segurança, porém não é o único método utilizado, há restrições para usuários, criptografia, entre outros métodos que podem ser implantados dependendo da utilização do sistema.

Esse tipo de sistema não é estagnado, ele pode sofrer acréscimos e modificações de acordo com suas necessidades. O administrador não precisa criar um novo sistema em caso de mudanças, ao contrário, ele apenas as verifica como adequá-las ao sistema, de tal forma que atenda às novas necessidades.

Diante do cenário atual em que a tecnologia é um grande facilitador, os sistemas web estão cada vez mais presentes, pois se mostram eficientes para substituir diversas funções que antes exigiam muito trabalho físico, além disso, também traz mais facilidades para os que usufruem desse serviço.

## 2.2 Infraestrutura Instalada - Servidor LAMP

Para implementar o que foi proposto optamos pela utilização de um servidor LAMP que consiste em um conjunto de softwares muito utilizados no desenvolvimento de aplicações web, são esses:

- a. Linux – sistema operacional
- b. Apache – servidor web
- c. MySQL – banco de dados
- d. PHP – linguagem de programação para web

Quando se trata de aplicações web abertas e gratuitas, o conjunto de aplicações mais usado é o LAMP, um acrônimo para Linux, Apache, MySQL e Perl/PHP/Python. Esse conjunto de aplicações é composto por, respectivamente, um sistema operacional, um servidor web, um sistema gerenciador de banco de dados e uma linguagem de programação.

### 2.2.1 Linux

O Linux é um Sistema Operacional (SO) licenciado pela fundação GNU (*General Public License – GPL*) com código fonte aberto, ou seja, há colaborações de voluntários para o seu desenvolvimento e aprimoramento. Frequentemente são geradas melhorias do sistema, contribuindo para a existência de várias versões e distribuições do Linux[5].

Atualmente o Ubuntu é uma das distribuições Linux mais conhecidas e utilizadas, isso ocorre devido a uma característica crucial, que é a preocupação com a maneira visual com que o sistema é apresentado ao usuário final[5].

Por isso fizemos uso da distribuição Ubuntu versão 14.04, o qual foi configurado por nós do início ao fim com todas as ferramentas necessárias para o bom funcionamento do sistema a ser desenvolvido.

### 2.2.2 Apache

O Apache é um servidor web HTTP, desenvolvido pela fundação ASF, mais conhecida como Apache Group. O Apache possui código aberto e funciona em qualquer sistema operacional. Tem como objetivo disponibilizar um servidor seguro, eficiente e extensível que fornece serviços HTTP constantemente atualizados para os padrões atuais[6].

Atualmente é o servidor web mais utilizado, ocupando cerca de 60% das instalações do mundo, como mostra a pesquisa feita pela W3Techs cujos resultados estão mostrados na Figura 2.1. Esse fator enfatiza sua eficiência em implementação de sistemas não só como a que está sendo desenvolvida nesse projeto, mas também para sistemas mais robustos[7].

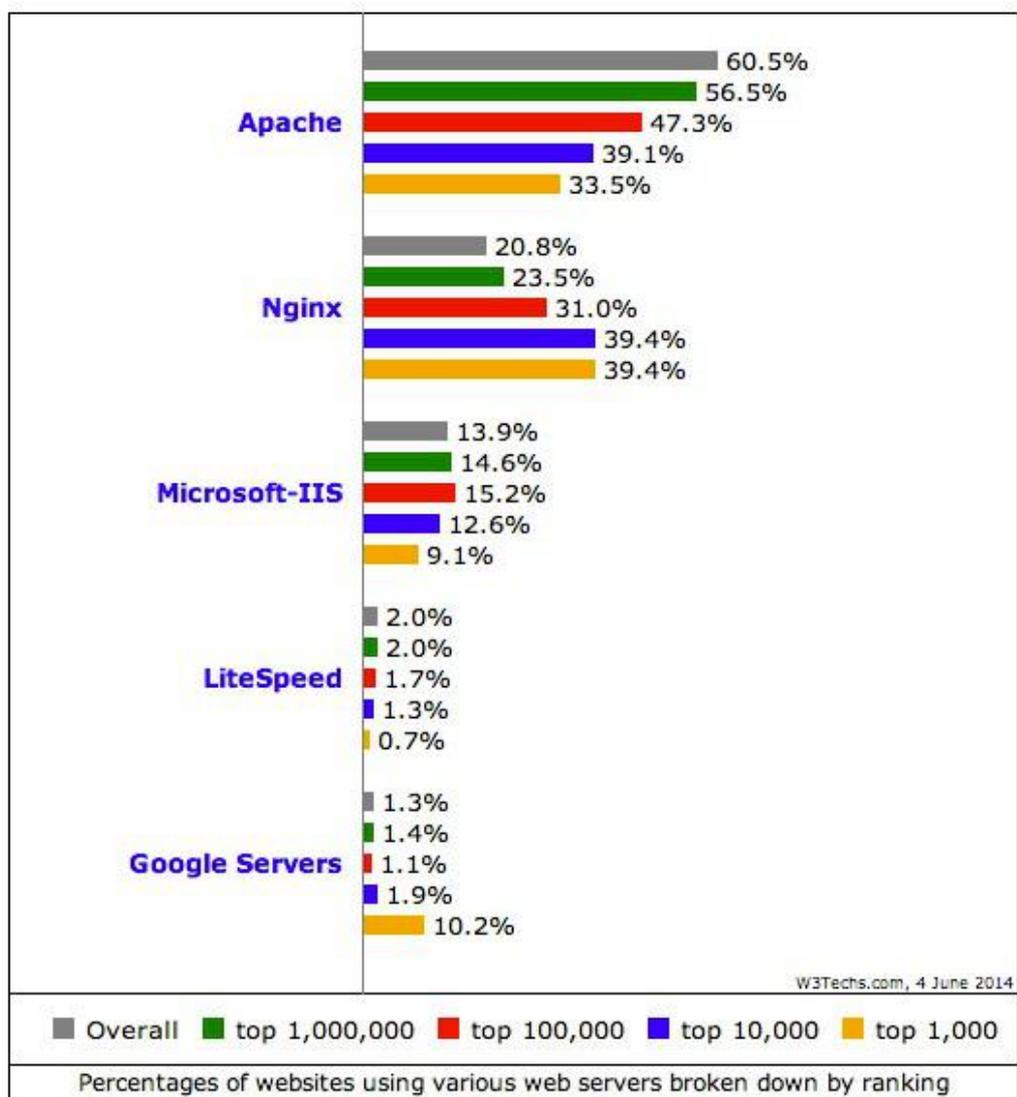


Figura 2. 1– Porcentagem de Utilização de Servidores Web ao redor do mundo, pesquisa da W3Techs.com de junho de 2014

### 2.2.3 MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) de código aberto, da Oracle. Utiliza a linguagem SQL (*Structured Query Language* - Linguagem de Consulta Estruturada em Português) em sua interface[8].

Esse é o software de bancos de dados mais popular, com mais de 10 milhões de instalações pelo mundo, muito utilizado para aplicações web, por ser um sistema portátil e interoperável com excelente desempenho[8][9].

A Figura 2.2 mostra a utilização do Sistemas Banco de Dados ao redor do mundo de acordo com a sua popularidade. Nela vemos que o MySQL é o segundo mais popular, pouco abaixo do SGBD da Oracle. Essa imagem foi obtida pelo site do DB-Engines[9].

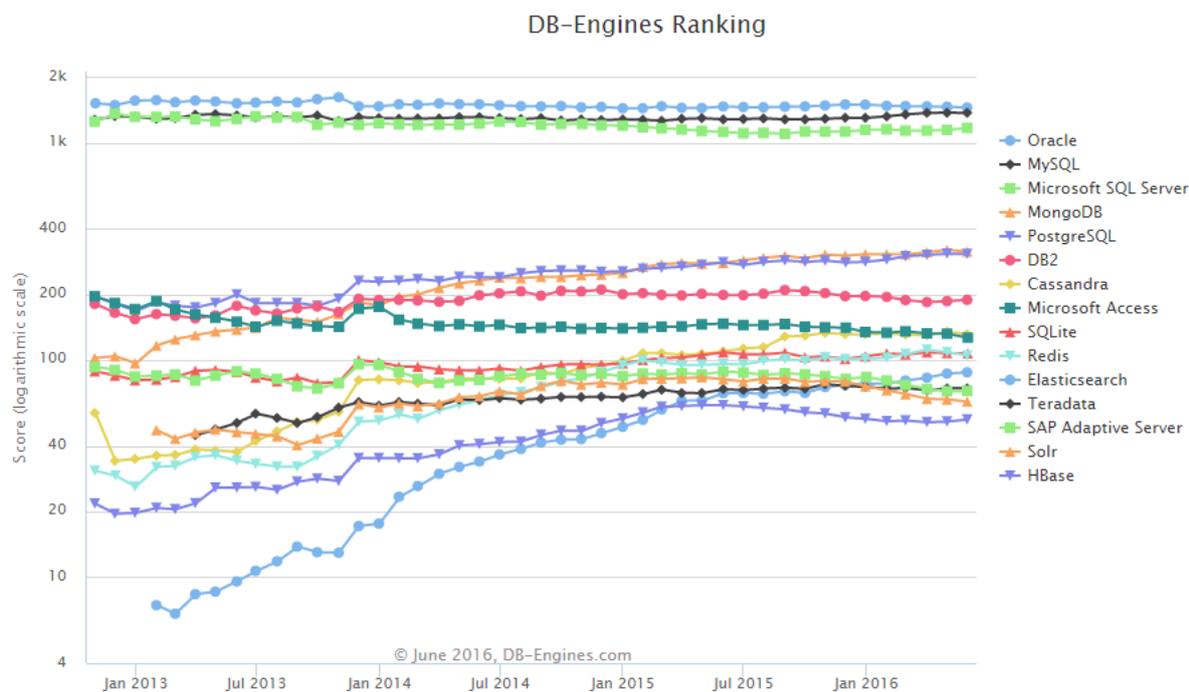


Figura 2. 2 - Ranking dos SGBDs mais populares segundo DB-Engines em junho de 2016

### 2.2.4 PHP

PHP (Acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de programação com código aberto voltada para o desenvolvimento de aplicações web. Possui várias características como a portabilidade, roda em qualquer sistema operacional e é compatível com outras linguagens, como HTML[10].

O maior benefício dessa linguagem é sua inteligibilidade. O PHP é uma linguagem simples, boa para um iniciante e que também oferece uma funcionalidade de diversos recursos para um programador profissional.

### 2.2.5 phpMyAdmin

Hoje em dia, a maior parte dos sites faz uso de bases de dados MySQL para guardar informações. O phpMyAdmin nada mais é que uma aplicação web que permite gerenciar essas bases de dados de forma muito simples e intuitiva, através de uma interface web. É um software livre, sendo seu gerenciamento desenvolvido com linguagem PHP de forma a possibilitar criação, remoção e manipulação de informações de bancos de dados e tabelas, como por exemplo: inserir, remover e editar campos[11].

## 2.3 Linguagens Utilizadas

A criação de sistemas web é desenvolvida mediante utilização de linguagens de

programação. Para esse projeto optamos por linguagens que se complementassem, facilitando o modo de utilizar o sistema para o usuário final, assim como para o desenvolvedor. As linguagens escolhidas são empregadas para desenvolvimento de aplicações web. São elas:

- a) HTML;
- b) PHP;
- c) CSS;
- d) JavaScript.

Cada linguagem possui suas próprias características e peculiaridades, sendo utilizadas de formas distintas em determinadas ações, ou seja, a forma como resolvemos um problema em uma linguagem pode ser completamente diferente em outra. Por isso, estudar para entender com clareza o modo de funcionamento de cada uma possibilita que o desenvolvedor consiga extrair ao máximo o potencial da linguagem escolhida para trabalhar.

### **2.3.1 HTML**

O HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto em Português) foi criado em 1991, por Tim Berners-Lee, no CERN (*European Council for Nuclear Research*) localizado na Suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas e compartilhar documentos com facilidade[12][13]. Foi rapidamente ganhando cada vez mais espaço. Como essa linguagem funciona em conjunto com o HTTP, quando sua API foi liberada, ele ganhou alcance global.

É a linguagem padrão para construção de páginas web. Essa linguagem faz uso de *tags* para estruturação da página e são elas que definem características desejadas como, por exemplo: o tipo de letra, tamanho, cor, espaçamento, e outros aspectos da página. As *tags* são interpretadas pelo *browser* e apresenta para o usuário a página editada.

Mesmo sendo uma linguagem totalmente voltada para páginas web, o HTML possui limitações, códigos mais complexos, os quais não podem ser escritos somente com essa linguagem, para essas funções mais complexas são utilizadas linguagens em servidor como Java, PHP, C# e Python.

### **2.3.2 PHP**

O PHP possui código fonte disponível para qualquer pessoa, sem custos, ou seja, sua licença de uso e edição é *Open Source*[10]. Isso quer dizer que ninguém pode comercializar qualquer versão do PHP, e qualquer modificação necessariamente deve continuar com o código fonte aberto para os usuários explorarem e modificarem caso queiram. Muitas empresas apoiam os desenvolvedores do PHP por diversos motivos, e um dos mais fortes é que essa linguagem é

muito usada no Linux e MySQL[14]. Além disso foi uma das primeiras linguagens passíveis de inserção no HTML, através da utilização de uma *tag* específica que diversificou ainda mais as funcionalidades para páginas web, visto que o PHP é uma linguagem de programação bem mais ampla que o HTML.

É necessário um servidor HTTP para que essa linguagem funcione, e por isso o Apache é essencial, pois quando necessário, o PHP faz a requisição ao servidor que retorna o que foi pedido, possibilitando funções de manipulação de bancos de dados e validações de segurança até processamentos mais complexos[14].

### **2.3.3 CSS**

O CSS (*Cascading Style Sheets*) é uma linguagem de estilo associada ao HTML. Possibilita mudança de fontes, cores, formatação entre outros. O maior benefício do CSS é que essas modificações não estão associadas ao conteúdo da página HTML, ou seja, é possível modificar o estilo sem alterar o conteúdo e vice-versa[15].

Outro benefício do CSS é a possibilidade da criação de um layout padrão para a aplicação de forma simples, dando uniformidade à informação, sem a necessidade da repetição de códigos. Essa linguagem também permite que as mesmas marcações de uma página sejam apresentadas em diferentes estilos. A maioria dos menus em cascata, estilos de cabeçalho e rodapé de páginas da internet atualmente são desenvolvidos em CSS. Tudo isso foi uma revolução para o desenvolvimento *web*, pois essa ferramenta nos permite controlar a interface de diferentes documentos em um único arquivo e diferentes dispositivos, nos permite uma melhor precisão a fim de manter uma mesma interface para diferentes navegadores, melhorias na acessibilidade com a possibilidade de “esconder” elementos da tela dos usuários sem problemas de visão, mas manter os mesmos elementos acessíveis aos leitores dessa tela, menor consumo de banda para usuário e servidor e também inúmeras técnicas dinâmicas que não poderiam ser utilizadas em tabelas[15].

### **2.3.4 JavaScript**

O JavaScript (JS) é uma linguagem de programação que auxilia o HTML e o CSS, controlando alguns comportamentos da página, como a aparição de botões, mudança de cores com cliques e envio automático de formulários, entre outras funções.

Assim temos três camadas bem definidas na programação. O HTML é responsável pelo conteúdo, o CSS pela formatação e o JS pelo comportamento. O JS manipula as duas camadas anteriores, pois um novo comportamento pode alterar tanto o conteúdo como a formatação.

Assim como na inserção de códigos PHP, existe uma *tag* específica para mostrar que está sendo usado JS dentro do código HTML e dessa forma o sistema consegue interpretar e saber como ler as informações adequadamente[16].

## 2.4 Android

Android é um Sistema Operacional *Mobile* baseado no Kernel Linux, desenvolvido pela *Open Handset Alliance*, uma aliança entre várias empresas, com participação da Google[17], esse SO foi desenvolvido para operar principalmente em dispositivos móveis, como celulares (*smartphones*) e *tablets*. Atualmente esse, está se expandindo e já se encontram versões do Android para TV, relógios, carros e câmeras fotográficas.

O SO tem as mesmas funcionalidades dos Sistemas Operacionais mais conhecidos, como Ubuntu, MacOS e Windows, ou seja, sua função principal é gerenciar os processos dos aplicativos e do *hardware* do dispositivo para garantir um bom funcionamento. A diferença é que o Android foi alavancado pela Google para ser introduzido em seus próprios dispositivos móveis e, assim, conseguir concorrer diretamente com outros sistemas operacionais dominantes como o Symbian (dispositivos Nokia), iOS (dispositivos Apple, como iPhone) e *Blackberry* OS. Foi desenvolvido com adaptações para um bom funcionamento em telas sensíveis ao toque, para a melhor utilização de teclados virtuais e manipulação de objetos virtuais pelo usuário.

Uma das principais vantagens do Android é, sem dúvida, sua integração com os serviços Google. Basta ter uma conta Google que o usuário consegue interligar todos esses serviços em um mesmo aparelho. Outra grande vantagem é o *Android Market*, a loja oficial dos aplicativos, que oferece uma infinidade de aplicativos gratuitos.

Esse possui código aberto, fazendo com que muitos programadores desenvolvam novas funções e recursos para o SO. Fazendo atualizações constantemente de forma a atender as necessidades mais recentes. Porém a utilização de código aberto juntamente com a grande quantidade de usuários leva a inúmeros ataques diários a esse sistema feito por *hackers*.

Atualmente o Android é o Sistema Operacional móvel mais utilizado no mundo, pois consiste em um *software* pronto, de baixo custo e personalizável, que suporta a alta tecnologia. Por isso é o escolhido para ser utilizado por muitas empresas que mexem com esse tipo de tecnologia[18].

Muitas aplicações utilizam o sistema Android, pois ele permite a visualização da aplicação de forma mais fácil para o usuário móvel, que vem consumindo cada vez mais, devido ao grande aumento da funcionalidade de *smartphones* e *tablets*, bem como a facilidade que a utilização desses dispositivos traz para o usuário.

## 2.5 Android Studio

Em 2013 a Google I/O anunciou o Android Studio, seu ambiente de desenvolvimento integrado (IDE) gratuito, o qual veio para facilitar o trabalho de desenvolvedores de aplicativos para dispositivos móveis.

Dentre as funções do *software* estão inclusas: edição inteligente de códigos, recursos para design de interface de usuário e análise de performance. A Google recomendava que os desenvolvedores utilizassem o IDE Eclipse para fazer aplicativos para o Android, mas desde o surgimento dessa plataforma, a Google oferece instruções para que os usuários migrem para o Android Studio[19].

O kit do Android Studio conta com:

- IDE do Android Studio;
- Ferramentas do Android SDK;
- Plataforma do Android 6.0 (Marshmallow);
- Imagem do sistema do emulador do Android 6.0 com APIs da Google.

É possível desenvolver aplicativos para celulares Android, *tablets*, Android Wear, Android TV, Android Auto e até mesmo *Google Glass*.

## 2.6 UML

O UML (*Unified Modeling Language*) é uma linguagem de modelagem usada para documentação de um software de forma padronizada[20]. Diferentemente das linguagens de desenvolvimento essa não faz parte da execução do projeto propriamente dito, ela auxilia o desenvolvedor, mostrando em forma de diagramas um comportamento do sistema[21].

Existem treze tipos de diagramas UML, de acordo com sua utilização, divididos em três categorias[21]:

- Diagramas de Estruturas: diagrama de classe, diagrama de objetos, diagrama de componentes, diagrama de estrutura composta, diagrama de pacote, e diagrama de implantação.
- Diagramas de Comportamento: diagrama de caso de uso, diagrama de atividades e diagrama de estado de máquina.
- Diagramas de Interação: diagrama de sequência, diagrama de comunicação, diagrama de tempo e diagrama da visão geral da interação.

Para o projeto em questão foram utilizados dois desses treze diagramas, os que mais se enquadram as necessidades do sistema que são: (1)Diagrama de Caso de Uso; (2)Diagrama de Classe.

### **2.6.1 Diagrama de Caso de Uso**

Um diagrama voltado para o que o sistema faz, do ponto de vista dos usuários, ou seja, descreve as funcionalidades do sistema de acordo com as interações dos usuários. Sem um aprofundamento técnico do que cada uma das funções representa.

É composto por quatro partes:

- Cenário: Eventos que acontecem a cada interação;
- Ator: Usuário do sistema;
- Caso de Uso: É uma tarefa realizada pelo ator;
- Comunicação: Faz a ligação do caso de uso com o ator.

### **2.6.2 Diagrama de Classe**

Esse diagrama tem como função descrever os objetos do sistema e como esses relacionam entre si. Esse diagrama possui quatro elementos:

- Nome;
- Atributos: tipo dos dados;
- Métodos: funções e parâmetros;
- Associação: tipo de comunicação entre as classes.

## Capítulo 3

### Desenvolvimentos Recentes

*Este capítulo apresenta projetos com temas próximos ao do projeto que está sendo desenvolvido.*

Para execução do trabalho foi necessária uma investigação para averiguar a existência de projeto semelhante no âmbito da UnB, em fase de implementação ou já executados, uma vez a universidade apresenta uma grande carência nessa área. Averiguou-se a existência de projetos com objetivos similares, ou seja, que visam auxiliar, de alguma forma, a gestão da instituição, a economia de recursos e a otimização dos serviços.

Nessa busca identificamos trabalhos que apresentam um mote similar em outras instituições de ensino superior, contudo com execução diferenciada, mas com objetivos em comum: o de auxiliar os serviços da universidade, com a utilização de sistemas *web*.

O artigo *Integration of university information systems using Web Services*[29] consiste na criação de um protótipo de uma *web service* para integrar os sistemas já utilizados pela Universidade dos Açores (UAC) em Portugal, de forma a ter mais agilidade, segurança, rapidez e praticidade nas prestações de serviço e trocas de informações internamente e externamente a UAC.

O artigo *Assessment system design for university web-based learning* [30], consiste no desenvolvimento de um sistema de gerenciamento feito para Universidade Técnica de Graz na China, para auxiliar na interação entre estudantes e professores, utilizando uma interface gerenciável e bancos de dados, implementado, portanto, para o portal de *e-learning* da universidade.

No artigo *Design and development of web-assisted instruction system for university physics course*[35], vemos uma implementação próxima ao sistema desenvolvido de fato para o curso de Física da Universidade Jiao Tong em Shangai – China.

Também foram encontrados artigos com temas próximos feitos em instituições brasileiras, como o artigo sistema *web* para gerenciamento de bancas de trabalhos acadêmicos[31], o qual tem por objetivo automatizar o sistema de bancas de avaliação, pois assim como no caso de estudo aplicado ao presente projeto, o departamento acadêmico dessa universidade possui a composição das bancas avaliativas de trabalhos e a ata de relatório como atividades realizadas de forma manual, bem como as atividades anteriormente citadas, as quais são desenvolvidas pelo ENE. O resultado do trabalho do estudante foi a implementação de um aplicativo que permite a automatização dessas tarefas. E como no projeto em questão, o sistema

foi disponibilizado na *web*. Assim como nós, ele fez uso de linguagens como CSS, HTML e SQL, porém também utilizou um ambiente de desenvolvimento chamado Eclipse Juno.

Um outro trabalho desenvolvido recentemente foi um sistema de controle de alunos em estágio curricular da Escola Politécnica de Pernambuco[32], que propõem o desenvolvimento de uma aplicação web de gerenciamento que tornasse mais fácil as tarefas exercidas pelo coordenador acerca desse assunto, evitando assim, matrículas de alunos com baixa qualidade e consequentemente aumenta o controle sobre os estagiários, bem como a devida participação de cada um na disciplina. Para isso ele utilizou tecnologias como: C#, ASP.NET e SQL Server.

Após encontrar diversas propostas de projetos aplicados à implementação de sistemas, que auxiliam em tarefas de extrema importância do cotidiano de uma universidade, chegamos à conclusão de que não só o Departamento de Engenharia Elétrica da Universidade de Brasília possui carências na automatização de determinadas tarefas, como também foi possível perceber que, em geral, há deficiência da área aplicada à gestão de departamentos no âmbito universitário. Assim vemos que projetos como o que está sendo implementado facilita a gestão, trazendo benefícios para toda comunidade acadêmica.

Diferentemente dos trabalhos já encontrados, o projeto em questão foca na criação de uma aplicação voltada ao auxílio do departamento como um todo, tendo uma larga abordagem e podendo ser ampliado para toda a instituição, bem como integrar os demais sistemas utilizados.

## Capítulo 4

### Etapas do Projeto

*Este capítulo aborda a metodologia utilizada para o desenvolvimento do sistema em questão.*

#### 4.1 Delimitação do Tema

O presente trabalho tem como proposta desenvolver um sistema *web* e um aplicativo Android para realizar algumas funções essenciais do Departamento de Engenharia Elétrica da Universidade de Brasília. Esse sistema faz a maioria de suas ações baseados em uma tabela *csv* (separado por vírgulas) que contém a lista de oferta.

Além de atividades com base nesta lista, como tabelas de horários, o sistema também possibilita algumas ações para o usuário como um cadastro em monitoria, disponibilização de formulários básicos para impressão, minimizando assim as interações físicas com o departamento.

A principal função do sistema seria aprimorar as atividades do departamento, colocando algumas de suas funcionalidades online, de forma a facilitar as interações entre professores, alunos e departamento.

O fluxograma da Figura (4.1) mostra as etapas realizadas para o desenvolvimento desse projeto.

**Etapas 1 – Pesquisa das Necessidades do Departamento:** definição do escopo do projeto, quais as funcionalidades essenciais para a implantação do projeto, e a viabilidade de implementação dessas.

**Etapas 2 – Pesquisa de Linguagens de Programação:** definição das linguagens a serem utilizadas para o melhor desempenho do sistema.

**Etapas 3 – Preparar Máquina:** fazer instalações necessárias na máquina para realização do projeto.

**Etapas 4 – Criar Bancos de Dados:** gerar um Banco de Dados no MySQL para suportar o desenvolvimento do sistema.

**Etapas 5 – Inserção de dados no Banco de Dados:** mostra como foi feita a inserção dos dados no Banco de Dados, de acordo com as necessidades do sistema.

**Etapas 6 – Desenvolvimento dos Algoritmos do Sistema:** apresentar o processo para o desenvolvimento dos principais algoritmos do sistema.



**Figura 4. 1– Etapas do Projeto**

**Etapa 7 – Desenvolvimento da Plataforma Web:** mostra as ligações feitas entre os algoritmos para o bom funcionamento do sistema *web* como um todo.

**Etapa 8 – Desenvolvimento do aplicativo Android:** apresenta quais ferramentas e quais funcionalidades foram implementadas para a construção do aplicativo Android.

#### 4.1.1 Pesquisa das Necessidades do Departamento

Atualmente, o Departamento de Engenharia Elétrica da Universidade de Brasília possui uma carência enorme de praticidade e otimização de alguns processos importantes realizados rigorosamente a cada semestre. Inicialmente sentamos com a responsável por essas tarefas a fim de definir as necessidades geradas em relação ao mapeamento das salas de aula. Descobrimos que a cada início de semestre é feito de forma manual a distribuição de todas as disciplinas ofertadas pelo ENE e para isso existem diversos quesitos que não podem ser ignorados, por exemplo, 4 engenharias diferentes cursam disciplinas do Departamento de Engenharia Elétrica, e para distribuí-las de forma que não entre em conflito com o fluxo de cada engenharia, isso gera um desgaste enorme e desnecessário, o qual poderia ser feito com base em uma lógica computacional já pré-definida.

Outro dilema é o processo de monitoria que é feito manualmente em papel, gerando outro desgaste, pois esse processo exige muito de todas as partes, necessitando de muito gerenciamento e análise feita de forma manual. A ideia do projeto é otimizar todo esse processo, transformando-o em algo virtual e mais prático, sem necessidade de gerenciamento.

Observando as necessidades do departamento em questão chegamos às funcionalidades básicas que precisam ser implementadas nesse sistema, essas funcionalidades estão listadas abaixo:

- **Automatização do processo de monitoria:** O processo de monitoria é bem trabalhoso para o departamento, pois envolve alunos, professores e administração. No sistema esse processo encontra-se totalmente automatizado de acordo com as necessidades citadas pelo departamento. Via sistema é possível o aluno se cadastrar para monitoria e são apresentados em uma tabela para que os professores possam aceitá-los ou não. Aqueles que são selecionados aparecem para o administrador para que ele possa finalizar o processo. O resultado do processo também é mostrado para o aluno para que ele tenha um retorno rápido. Assim, o sistema faz com que o processo, que atualmente é feito totalmente de forma manual com a utilização de formulários impressos pelo departamento, seja efetuado de forma mais rápida e prática para as três partes envolvidas, alunos, professores e administradores.

- **Mapeamento de Salas:** Existe sempre a dúvida de onde fica determinada sala, por isso ter um mapeamento, que mostra de forma visual, é importante, assim os alunos podem

acessar e visualizar para onde devem ir. Esse mapeamento apresentou dois desafios, o primeiro foi que a nomenclatura das salas utilizada pelo sistema de matrículas da UnB (matrícula web) e o utilizado internamente pelo departamento são diferentes. O segundo foi conseguir um mapa eficiente das instalações onde as aulas são ministradas. No mapeamento em questão foi escolhido fazer uso da nomenclatura utilizada pelo matrícula web visto que no horário disponível para os alunos é usada essa nomenclatura, assim os alunos poderão ver em seu horário a sala e procurar por ela facilmente no mapa disponível no sistema. O departamento não possui a planta da faculdade de tecnologia (FT) e do prédio de Serviços Gerais 11 (SG-11), onde são ministradas as aulas, por esse motivo não foi possível realizar um mapeamento eficiente, porém uma versão provisória foi feita com um mapa adaptado pelas desenvolvedoras para dar uma noção de como ficaria esse serviço. O mapeamento será aprimorado com a planta quando esta estiver disponível.

- **Disponibilização de Horários:** Todo início de semestre ocorrem dúvidas quanto à alocação de salas. Por vezes o matrícula web não apresenta mostra a localização das salas de forma eficiente, várias disciplinas são ofertadas com local a designar, assim o departamento precisa disponibilizar para os alunos essas informações. Normalmente montam-se as disposições das disciplinas por salas, imprimem-se os horários de cada uma delas e então são disponibilizados para os alunos em um mural. Esse método além de gerar mais gastos para o departamento com papel, também é um pouco difícil para os alunos que precisam procurar a sua matéria dentre todas as salas. No sistema foi implementado um algoritmo que monta os horários dinamicamente de acordo com a busca do aluno, os horários são gerados com base em um banco de dados que contém todas as informações da lista de oferta, conforme o anexo 2. Essa disponibilização de horários facilita muito o processo, tanto para os alunos quanto para o departamento, pois este não precisará mais gerar tabelas e imprimir para apresentá-las para o aluno uma vez que é possível acessá-las a qualquer momento, tanto pelo sistema como pelo aplicativo.

- **Disponibilização de Formulários:** O departamento possui vários formulários para diversas atividades, como: inscrição em projeto final, inscrição em monitoria e requisição de documentos. Esses formulários ficam disponíveis em uma pasta para serem retirados no departamento. Para facilitar esse processo, foram disponibilizados em PDF no sistema os formulários mais utilizados, assim os alunos não precisam ir ao departamento pegá-los, eles podem acessar o sistema e imprimi-los para entregar no local necessário já preenchido.

- **Reserva de Salas:** As reservas de salas são feitas manualmente, assim como o processo de monitoria. O objetivo é fazer um algoritmo para automatizar esse processo, porém esse algoritmo ainda não foi implementado no sistema. Para facilitar o processo de reserva de

salas, o formulário de reserva estará disponível no sistema, podendo ser impresso e preenchido com o auxílio dos horários disponibilizados na página web, onde é possível visualizar em quais momentos uma determinada sala está desocupada para então fazer a solicitação entregando esse formulário no departamento.

#### **4.1.2 Pesquisa de Linguagens de Programação**

Uma vez definidas as necessidades do projeto, foi feita uma pesquisa de linguagens de programação que atendessem os requisitos desse projeto. Entendido como funciona o desenvolvimento de uma página web, bem como cada linguagem e ferramenta necessária para tal, foram feitos diversos testes práticos a fim de fixar o aprendizado.

Assim foram escolhidas para esse projeto uma junção das linguagens HTML, PHP, CSS e JavaScript, que apresentam um bom funcionamento para o desenvolvimento dos recursos necessários para parte da plataforma web. Para o desenvolvimento do aplicativo Android foi usado Java.

#### **4.1.3 Preparar Máquina**

Ao iniciar a parte prática do projeto, ou seja, a criação da plataforma web para o departamento de Engenharia Elétrica, foi necessário, primeiramente, a criação de um ambiente na máquina. Nesse projeto utilizamos uma máquina virtual - “Virtual Machine” VirtualBox da Oracle – rodando o Ubuntu 14.04 client.

Para utilização dessa máquina precisamos instalar o “LAMP”. Como estamos utilizando nessa o Ubuntu que é um SO baseado em Linux, foi necessário a instalar o servidor Apache, o Banco de Dados MySQL e o PHP. Para facilitar a administração do Banco de Dados, instalamos também o phpMyAdmin.

Cada instalação foi feita através do terminal do Linux e, em seguida, foram feitas as verificações necessárias para garantir que tudo ocorreu conforme o planejado.

- **Apache:** os comandos utilizados na instalação do servidor Apache podem ser visualizados no Apêndice A1. A Fig. (4.2) mostra a verificação da instalação do servidor web, mostrada acessando o endereço de *localhost* da máquina.

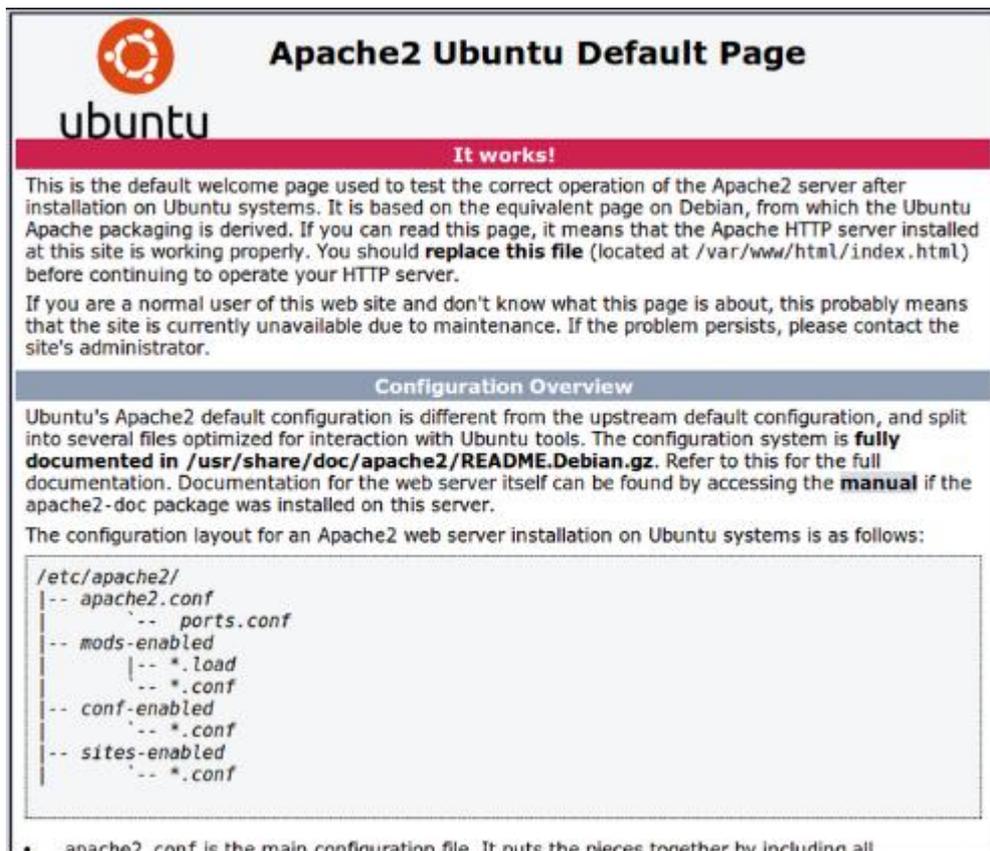


Figura 4. 2- Tela de verificação do funcionamento do Servidor Apache2.

- **MySQL:** os comandos utilizados na instalação do MySQL e a configuração do seu usuário também podem ser vistos no Apêndice A1.
- **PHP:** os comandos utilizados para a instalação do PHP estão disponíveis no Apêndice A1. Além disso, é possível verificar se o PHP foi instalado com sucesso acessando <http://localhost/info.php> e obtendo a página mostrada na Fig. (4.3) que mostra que esse foi instalado com sucesso.

PHP Version 5.5.9-1ubuntu4.5



<b>System</b>	Linux ubuntu12.04-LTS-VirtualBox 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:12 UTC 2014 i686
<b>Build Date</b>	Oct 29 2014 11:55:53
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/apache2
<b>Loaded Configuration File</b>	/etc/php5/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php5/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-imagick.ini, /etc/php5/apache2/conf.d/20-ison.ini, /etc/php5/apache2/conf.d/20-mcrypt.ini, /etc/php5/apache2

Figura 4. 3- Tela da verificação do funcionamento do PHP.

- **PhpMyAdmin:** os passos para a instalação do phpMyAdmin se encontram no Apêndice A1. A fim de verificar se tudo ocorreu como desejado na instalação, acessamos a URL <http://localhost/phpmyadmin> a verificação é mostrada na imagem da Fig. (4.4) que corresponde a página inicial de acesso ao phpMyAdmin.

phpMyAdmin

Bemvindo ao phpMyAdmin

Lingua - Language

Português - Portuguese

Entrada

Utilizador :

LuanaeThamara

Palavra-passe :

Executar

Figura 4. 4- Tela da verificação do funcionamento o phpMyAdmin.

#### 4.1.4 Criar Bancos de Dados

O MySQL foi o BD escolhido para o armazenamento dos dados e o phpMyAdmin foi o aplicativo utilizado com o objetivo de facilitar a gerência do BD. Para o projeto foi gerado um BD chamado formulário. Todas as tabelas do sistema foram criadas dentro desse BD. Na Figura (4.5) vemos um exemplo da criação de um banco de dados pelo gerenciador.

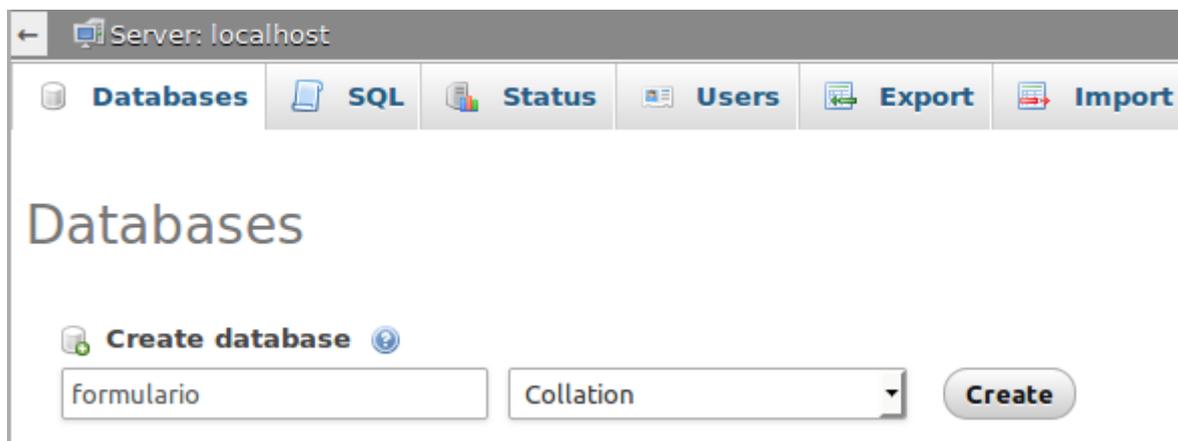


Figura 4. 5 – Tela de criação de banco de dados

O phpMyAdmin possibilita que a criação de tabelas de um jeito fácil e intuitivo, através da sua interface gráfica, ilustrado na Figura (4.6). Para o bom funcionamento do sistema foram criadas 8 tabelas, cada uma com sua respectiva função que será explicada detalhadamente no tópico 5.2.

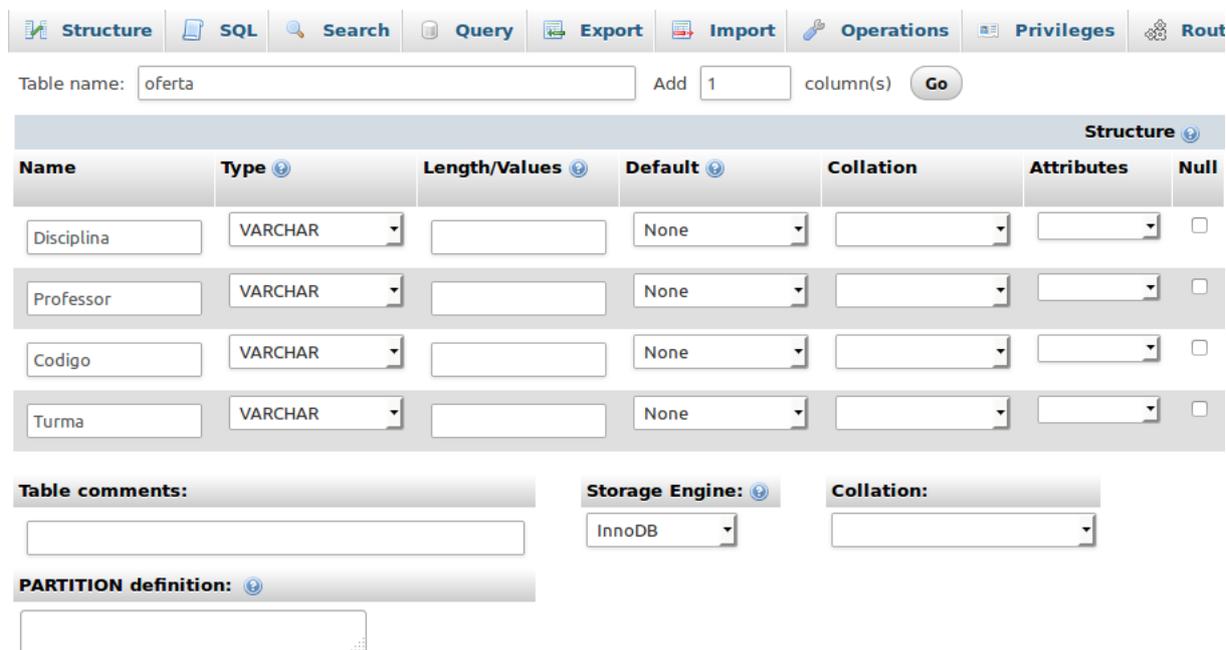
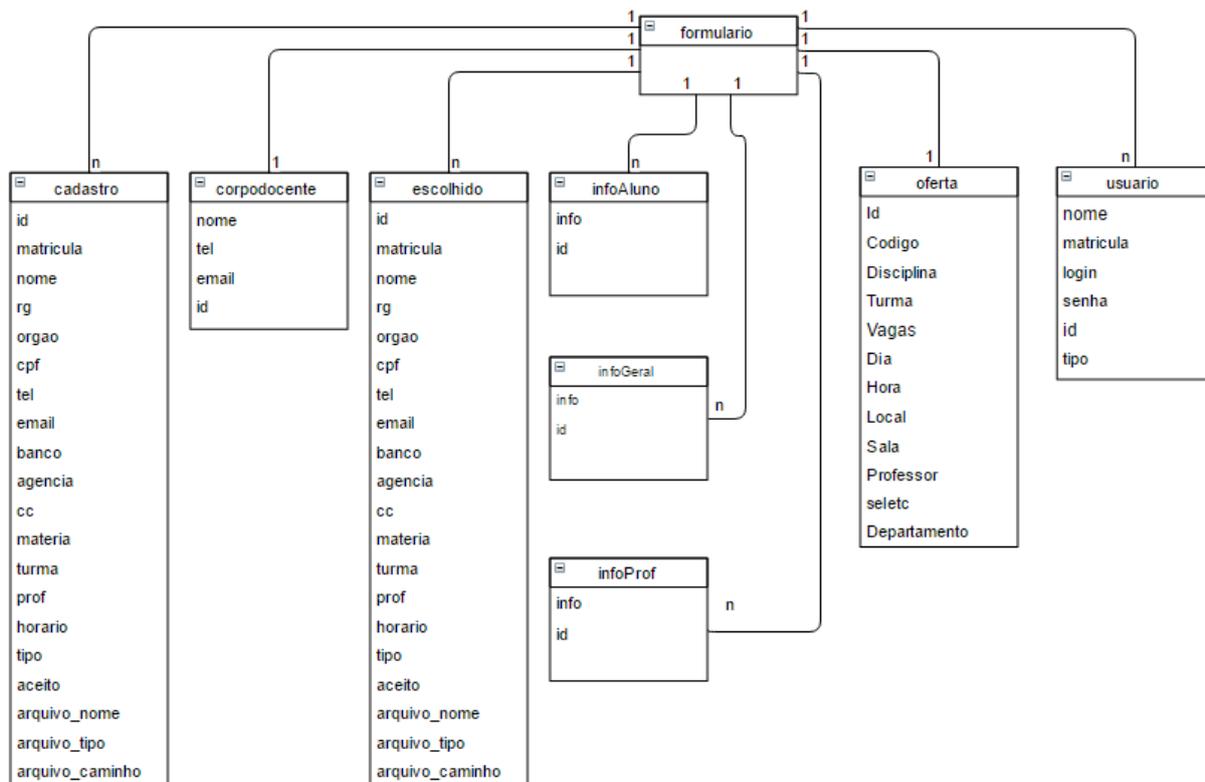


Figura 4. 6– Tela de criação de tabelas no phpMyAdmin.

Na Figura (4.6) temos um exemplo de criação de uma tabela para o banco de dados pelo phpMyAdmin. Para gerar essa tabela, o preenchimento de pelo menos três primeiros campos devem estar completos, os demais não são obrigatórios. O primeiro (*Name*) corresponde ao nome da coluna, usado como referência na inserção de dados. O segundo (*Type*) é o tipo de dado que será inserido naquele campo. Os tipos de campos mais utilizados nesse projeto foram: *varchar* para dados em texto e *int* para valores numéricos. É importante observar que se o campo

for do tipo *int* e for inserido um valor em caracteres, esse não aceitará o valor. O terceiro e último campo obrigatório é o tamanho (*length*). Neste, definimos o limite máximo de dígitos que pode ter determinado campo, por exemplo o referente à turma não precisa de mais de um caractere.

Para a realização do sistema proposto foi necessário criar um banco de dados chamado formulário composto por oito tabelas: cadastro, corpodocente, escolhido, infoAluno, infoProf, infoGeral, oferta e usuario, cada uma com sua função e variáveis específicas. A configuração das tabelas do banco é mostrada na Figura (4.7) abaixo:



**Figura 4. 7 – Diagrama das tabelas criadas para o projeto.**

A tabela cadastro guarda as informações dos usuários que se candidatam para a monitoria, podendo haver “n” cadastros efetuados. A corpodocente armazena as informações dos professores que compõem o departamento, havendo apenas a possibilidade de um corpo docente. A escolhido guarda os dados dos alunos que foram selecionados pelo professor no processo de monitoria, podendo haver mais de um monitor. As tabelas infoAluno, infoProf e infoGeral guardam as informações disponibilizadas para seus respectivos usuários. A oferta é a tabela mais importante do projeto, pois é nela que encontramos a lista de oferta do semestre. E por fim, na usuario encontramos as informações referentes aos usuários cadastrados no sistema.

#### 4.1.5 Inserção de dados no Banco de Dados

Diante das aplicações do sistema, observou-se a necessidade de se inserir dados nas

tabelas por dois métodos. O primeiro método consiste em enviar um arquivo que será lido e seus dados irão para o banco. Essa aplicação foi feita tendo em vista que a lista de oferta costuma ser uma tabela no *excel*, que pode ser facilmente transferida para o formato *csv* (separado por vírgulas), e é colocada no SGBD para que seus dados possam ser utilizados. O trecho de código responsável por essa aplicação encontra-se na Figura (4.8).

```
<?php
include "conecta_mysql.inc";
```

Porém na leitura de arquivo esse fator é mais difícil de ser controlado visto que este pode não ter necessariamente a mesma sequência que a utilizada na tabela. Uma sequência diferente pode gerar diversos erros.

Nesse projeto utilizamos um arquivo disponibilizado pelo ENE para a criação da tabela. Esse artigo obedece a um padrão que deve ser sempre seguido para o bom funcionamento do projeto. Caso o modelo desse arquivo seja alterado, serão necessárias mudanças tanto no banco de dados quanto no código responsável por essa tarefa.

#### **4.1.6 Desenvolvimento dos Algoritmos do Sistema**

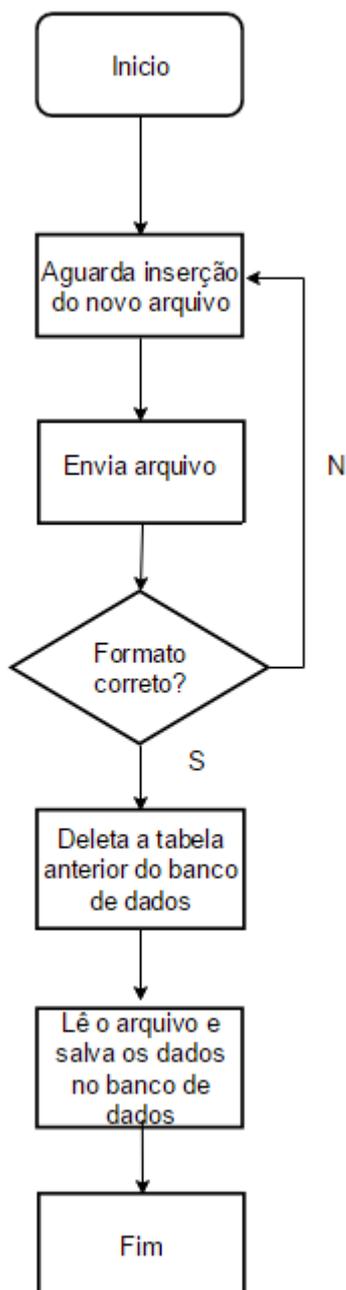
Antes da criação de uma plataforma completa, foi necessária a criação dos algoritmos responsáveis pelas funções básicas do sistema, como: criação de horários através da leitura de dados do banco, formulários para monitoria e cadastro; e conexão do sistema com o BD.

Para criação desses algoritmos utilizamos PHP, HTML e JavaScript. Com essas linguagens, criamos os algoritmos principais e também fizemos os testes de maior importância.

Os algoritmos são baseados em leituras e inserção de dados no SGBD, como visto no tópico 3.1.4. Esse projeto utiliza apenas um banco de dados, chamado formulário. Para facilitar o acesso a esse banco, foi feita uma função chamada `conecta_mysql.php` que é responsável por todas as configurações necessárias para acesso a esse banco específico, como: nome, host, login e senha. Essa função é importante para o sistema, pois evita repetição do código, basta chamar a função ao invés disso.

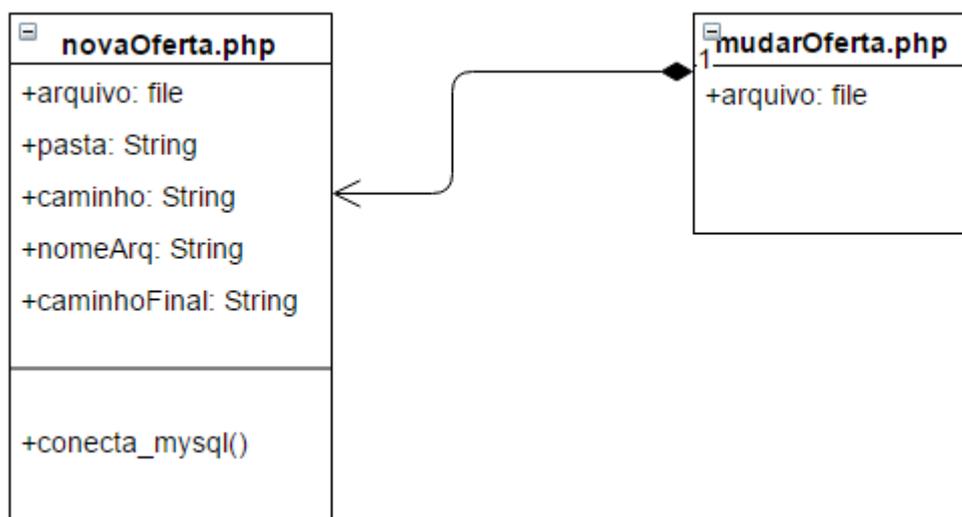
A seguir vamos explicar detalhadamente o funcionamento dos principais algoritmos do sistema, os quais são os responsáveis pela oferta, processo de monitoria, criação de horários e cadastro no sistema.

A tabela de oferta é a base para a construção do sistema, por isso é importante que essa seja sempre mantida atualizada. Visando facilitar esse processo, foi desenvolvido um algoritmo que permite a mudança da tabela no banco de dados por meio de uma interface gráfica simples, que solicita ao administrador do sistema um novo arquivo *csv* com a tabela a ser inserida no sistema. Assim que o usuário realiza *upload* de um arquivo válido no sistema, a tabela anterior é substituída pela nova no banco de dados. Esse processo está descrito no fluxograma da Figura (4.10).



**Figura 4. 10 – Fluxograma do processo de alteração da tabela oferta.**

O diagrama que mostra as classes envolvidas nesse processo encontra-se na Figura (4.11), onde vemos duas classes, em que uma depende totalmente da outra. A primeira mudarOferta recebe o arquivo enviado pelo o usuário, em seguida chama a classe novaOferta, que faz a verificação de se o arquivo está ou não no formato correto, caso o arquivo recebido seja válido, é chamada a função *conecta\_mysql*, e quando tem acesso ao banco, substitui os dados antigos pelos novos lidos do arquivo, caso contrário ela retorna para classe mudarOferta e espera um novo arquivo, como mostrado no fluxograma acima, Figura(4.10).

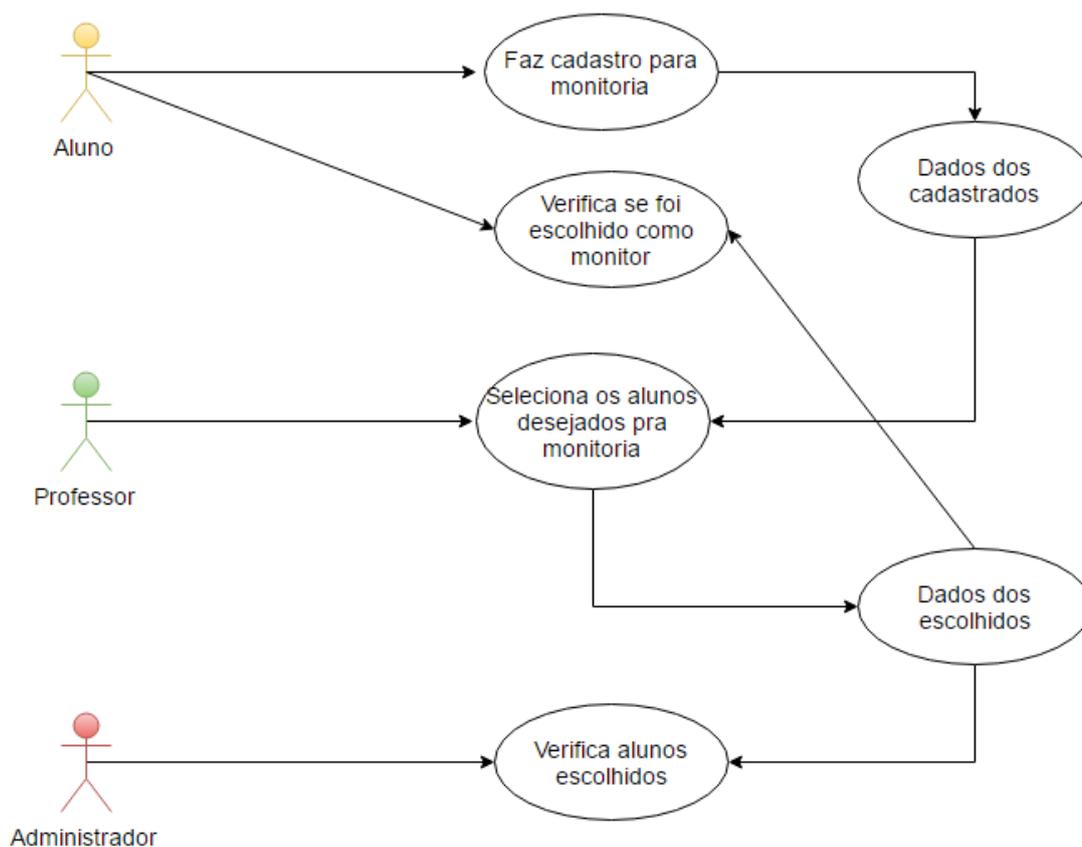


**Figura 4. 11 – Diagrama de classes da oferta.**

O processo de monitoria é complexo, pois envolve os três tipos de usuários do sistema: alunos, professores e administrador. Para seu bom funcionamento, ele foi feito em partes. Primeiro, um formulário de inscrição a ser preenchido pelos alunos, alguns dados desse formulário são retirados da tabela oferta, como por exemplo as disciplinas ofertadas visto que não faria sentido se cadastrar para monitoria em uma disciplina que não será ofertada. Uma vez que os alunos se inscrevam, é gerada uma tabela com todos os inscritos. Os professores têm acesso a essa tabela (somente dos alunos que se inscreveram para monitoria em matérias ministradas por eles), e escolhem os alunos que desejam ter como monitores. Os candidatos escolhidos são enviados para outra tabela.

Essa última tabela contém todos os alunos escolhidos como monitores. O aluno pode ver se foi ou não escolhido como monitor através de um acesso a essa tabela. É importante ressaltar que o aluno não visualiza toda a tabela, ele tem acesso filtrado pelo seu nome de usuário, logo ele só visualiza se foi ou não selecionado. Já o administrador tem acesso à tabela com todas as informações dos monitores selecionados pelos professores para realizar as ações necessárias fora do sistema.

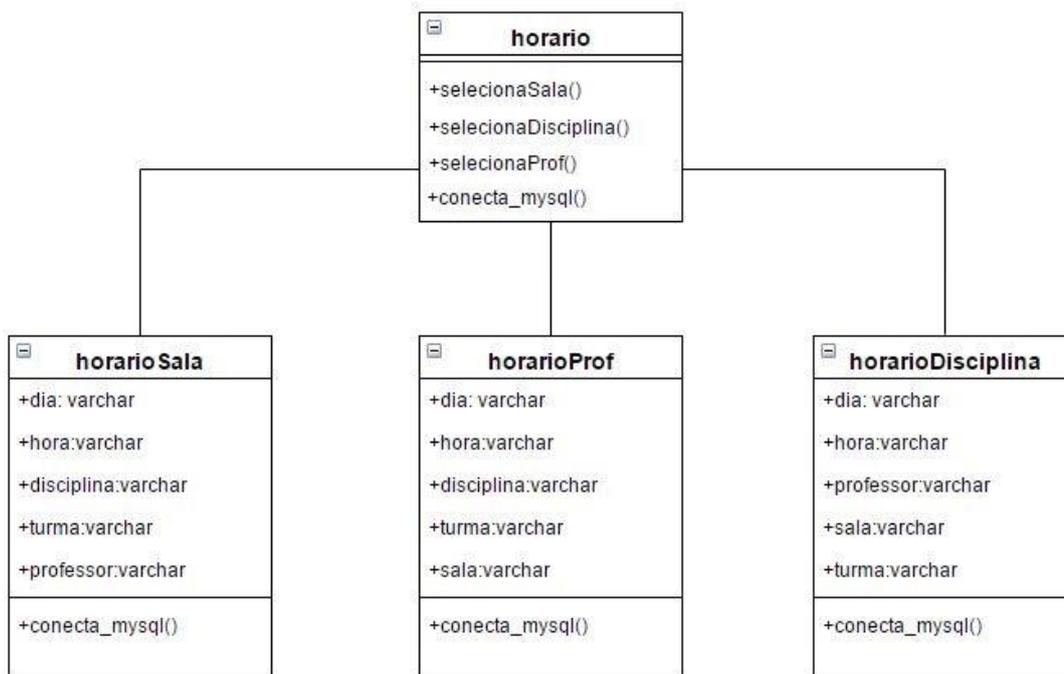
O passo a passo descrito acima está representado em forma de diagrama de caso de uso na Figura(4.12), de forma a se visualizar melhor como ocorre esse processo, e o que cada um dos usuários podem fazer.



**Figura 4. 12– Diagrama de casos de uso do processo de monitoria.**

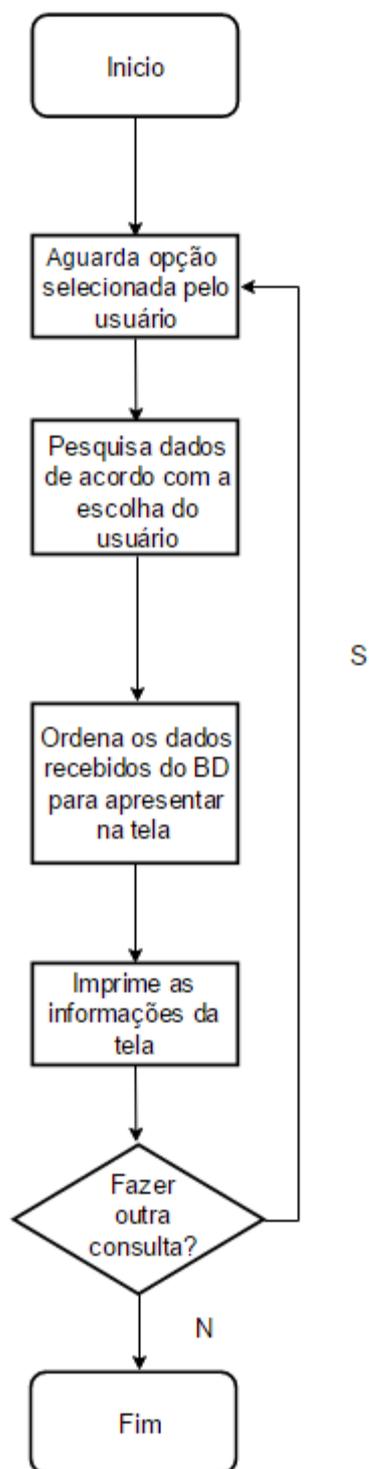
A montagem dos horários é de grande importância nesse sistema. Eles podem ser acessados de três formas, de acordo com a necessidade do usuário: por disciplinas, por salas e por professores. A lógica de montagem dos horários é a mesma para os três tipos de acesso, o que muda é a ordem da forma pela qual as informações são apresentadas, como podemos ver no diagrama de classes da Figura (4.13).

Nesse diagrama vemos a classe principal horários, onde o usuário seleciona o dado que ele deseja; pode ser uma sala, uma disciplina ou um professor. Conforme a escolha do usuário os dados são lidos do banco de dados, ordenados e apresentados em tela. Assim que o usuário faz a sua escolha, o horário é montado utilizando o dado escolhido, por exemplo: se for uma sala, a função utilizada será a selecionaSala, quando esta chamará a classe horarioSala, a qual montará o horário, conforme será apresentado para o usuário. Vemos no diagrama da Figura (4.13) que serão apresentadas as informações: dia, hora, disciplina, turma e professor para essa requisição.



**Figura 4. 13 – Diagrama de classes referente aos horários.**

Para montar os horários, é feita uma varredura dos dados da tabela oferta do banco de dados. Vale ressaltar que o horário é construído dinamicamente de acordo com a leitura das informações no banco de dados, cada vez que é feita uma requisição, o horário é montado instantaneamente, não ficando salvo após ter sido feito anteriormente. O processo dessa montagem pode ser melhor compreendido através do fluxograma da Figura (4.14).



**Figura 4. 14– Fluxograma da montagem de horários.**

Para um bom funcionamento, são necessárias algumas restrições, pois não é qualquer usuário que pode mudar a oferta no sistema, ou aceitar os monitores, por isso foi necessária a criação de um sistema de cadastro para que pudesse limitar o que cada um pode ou não acessar.

O cadastro é feito por meio do preenchimento de um formulário que vai para o banco de dados. Após o cadastro, o usuário vai precisar fazer o login para ter acesso às partes mais

específicas do sistema.

As classes envolvidas no processo de cadastro estão representadas no diagrama de classes da Figura (4.15), onde vemos a classe cadastro que é responsável por salvar os dados digitados pelo usuário e a vericadastro que verifica a validade desses dados, conforme algumas restrições básicas, como por exemplo, todos os campos devem estar preenchidos para que o cadastro seja validado. Caso a verificação seja bem-sucedida, é chamada a função `cadastra` que insere os dados no banco de dados.



Figura 4. 15– Diagrama de classes do cadastro.

#### 4.1.7 Desenvolvimento da Plataforma Web

A plataforma web foi desenvolvida para funcionar como facilitador da interação dos usuários com o sistema. Através dessa, é possível fazer remotamente todas as ações com uma interface gráfica organizada. Tem como principal função fazer a junção de todos os algoritmos de forma a facilitar a utilização do sistema pelos usuários. Essa foi desenvolvida utilizando as linguagens PHP, HTML, CSS e JavaScript.

A Figura (4.16) mostra a lista com todos os programas criados para o bom funcionamento do sistema como um todo. As principais classes são essas que foram explicadas no tópico acima, porém, existem mais classes no projeto, essas são responsáveis pela ligação entre as classes por meio de interface gráfica, que permite os usuários acessarem e utilizarem as funcionalidades do sistema sem problemas.

As classes que tem App no nome são as utilizadas no aplicativo. O princípio de funcionamento é o mesmo utilizado nas do sistema, porém essas estão com uma interface gráfica diferenciada para melhor visualização no aplicativo.

Vemos também que a maior parte dos códigos utilizam extensão *.php*, o PHP é utilizado dentro do HTML, assim como pequenos *scripts* JavaScript. Os artigos utilizados para parte

gráfica .css se encontram dentro do diretório *css* que pode ser visto na Figura (4.16). O diretório *imagens* contém as Figuras e gifs utilizadas no sistema.

Os demais diretórios *oferta*, *pdf* e *QuadroResumo* contêm dados arquivados para utilização no sistema. O *pdf* armazena os formulários que podem ser acessados pelo usuário em formato pdf, o *oferta* guarda o arquivo *csv* com a oferta que foi inserida no sistema pelo administrador e o *QuadroResumo*, o qual arquiva os quadros resumos incluídos pelos estudantes durante o cadastro na monitoria.

```
alteraCadastro.php      imagens
altera.inc              indexApp.php
alunosSelecionados.php index.php
cadastra.inc            infoAlunos.php
cadastro2.php           infoGeral.php
cadastro.php            infoProf.php
conecta_mysql.inc       insereBancoAluno.php
contato.php             insereBancoGeral.php
contatosApp.php         insereBancoProf.php
controle.php            insere.inc
corpoDocente.php        insereInfoAluno.php
css                     insereInfoGeral.php
formulario.php          insereInfoProf.php
formularios.php         insereMonitor.php
horarioDisciplinasApp.php login.php
horarioDisciplinas.php monitoresEscolha.php
horarioProfApp.php     mudaCadastro.php
horarioProf.php         mudarOferta.php
horarioSalasApp.php    novaOferta.php
horarioSalas.php        oferta
horarios.php            paginaAdmin.php
                        paginaAluno.php
                        paginaOferta.php
                        paginaProfessor.php
                        pdf
                        QuadroResumo
                        quemSomosApp.php
                        quemSomos.php
                        reserva.php
                        selecionaDisciplinaApp.php
                        selecionaDisciplina.php
                        selecionado.php
                        selecionaMonitor.php
                        selecionaProfApp.php
                        selecionaProf.php
                        SeleccionaSalaApp.php
                        selecionaSala.php
                        tabelaOferta.php
                        validaLogin.php
                        vericadastro.php
                        verifica.php
```

Figura 4. 16– Print do terminal com as classes utilizadas no projeto.

Essa plataforma possui quatro divisões, de acordo com o tipo de usuário que está acessando o sistema, para que algumas funções possam ser realizadas apenas por tipos pré-determinados de usuários. Dessa forma um aluno não terá acesso a nada do que somente um administrador pode acessar, assim algumas funções mais específicas estão protegidas pelo sistema de login.

A primeira parte, presente na página inicial da plataforma, é a mais geral disponível para os usuários e contém informações gerais sobre o sistema, como: a lista de oferta, o corpo docente, formulários para download, mapa de salas, horários e informações sobre o departamento. Um diagrama com todas as atividades disponíveis para os usuários do sistema está disponível na Figura (4.17).



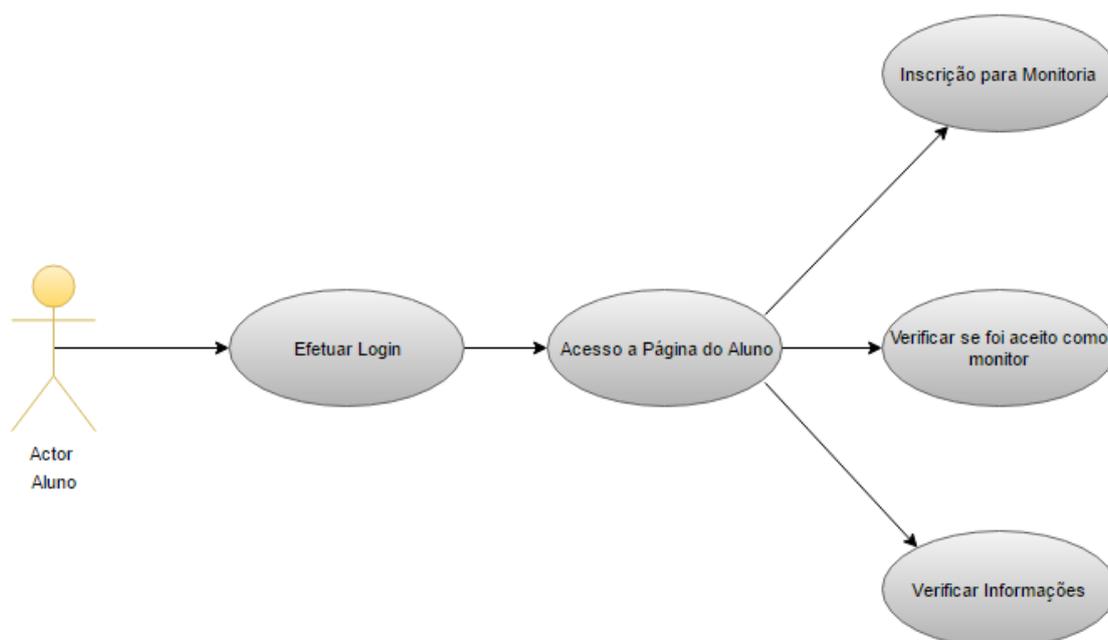
**Figura 4. 17– Diagrama de casos de uso para usuários gerais do sistema.**

As outras três divisões estão protegidas pelo login, por serem atividades que dizem respeito a um tipo específico de usuário. O sistema possui três tipos de usuários: alunos, professores e administradores, cada um com suas funções específicas, por isso a partir do login o usuário é encaminhado para uma das três partes subsequentes do sistema, de acordo com seu tipo.

Ao realizar o login, é feita uma checagem através da função validaLogin.php, de se o

usuário está cadastrado no sistema, e se sim qual o seu tipo de usuário. Assim e feito o encaminhamento automaticamente para a página com as funções específicas para ele.

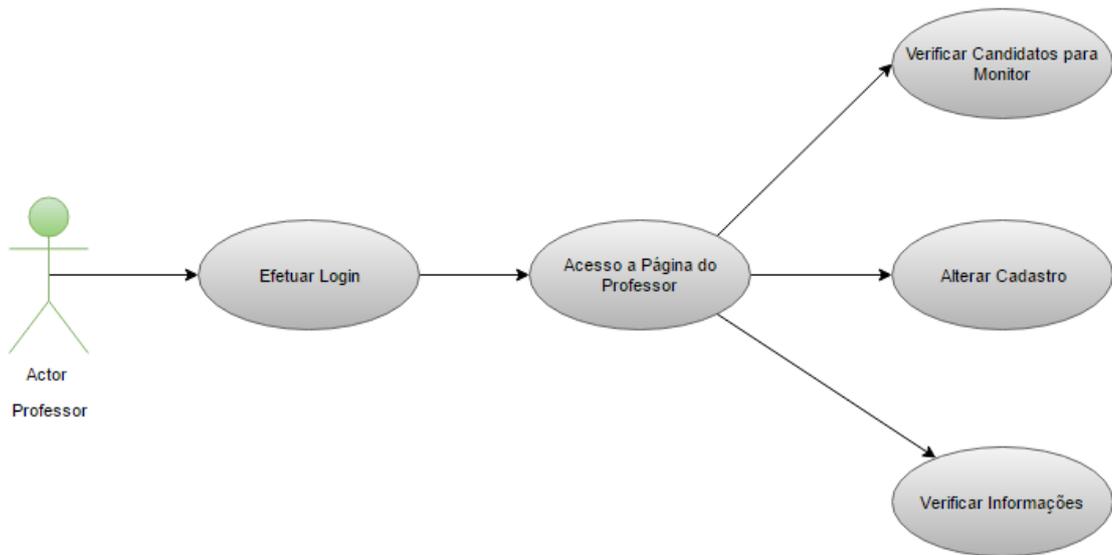
O aluno ao realizar login é encaminhado durante a validação do seu usuário diretamente para a paginaAluno.php. Nessa página estão disponíveis as funções para os alunos, como o cadastro para monitoria, a verificação para averiguar se foi escolhido ou não como monitor, e também informações específicas. É importante observar que cada aluno que acessar o sistema terá seus dados associados à página do aluno, assim cada um apenas poderá acessar seus próprios dados. O diagrama que mostra as ações possíveis para os alunos está na Figura (4.18).



**Figura 4. 18– Diagrama de Casos de Uso para Usuários do tipo aluno.**

A mesma validação ocorre no login de professores ao verificar que o tipo de usuário que efetuou login é Professor. Esse é encaminhado para paginaProfessor.php, nessa estão disponíveis as funções restritas a professores, como: verificar os inscritos para monitores, e aceitá-los ou não, fazer alterações e fazer alterações no seu cadastro visto que por motivos de bom funcionamento do sistema, todos os professores foram previamente cadastrados. Caso tenham informações pertinentes aos professores, essas também estarão disponíveis nessa página. Cada professor vê somente seus dados, não tendo acesso a informações referentes a outros usuários. Essa limitação também é feita no processo de validação do usuário.

A Figura (4.19) contém o diagrama UML de casos de uso referente a usuários do tipo professor, mostrando as ações que podem ser efetuadas por ele.



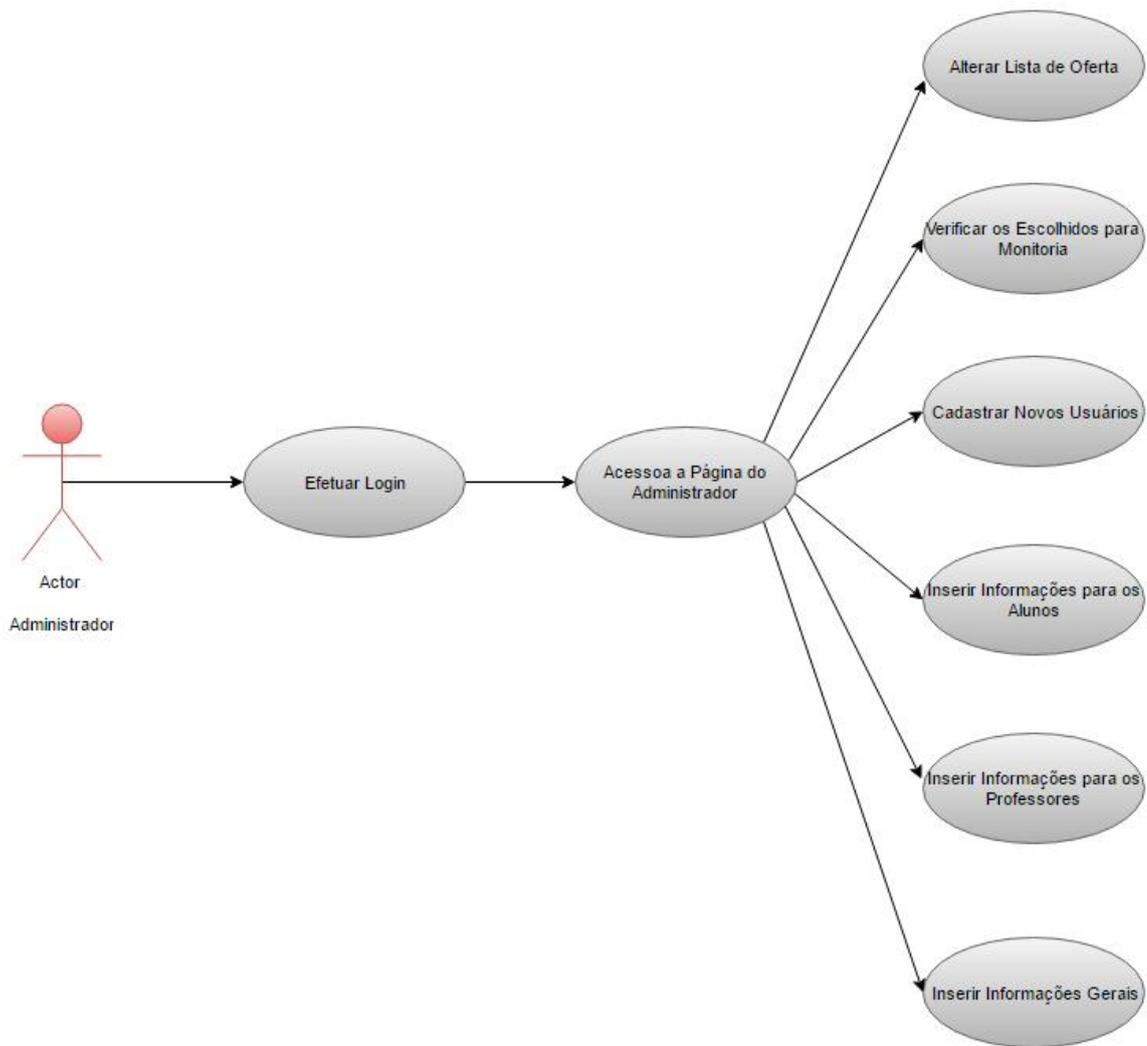
**Figura 4. 19– Diagrama de Casos de Uso para Usuário do tipo Professor.**

Para o bom funcionamento do sistema, é importante a presença de um administrador, que tem como função manter o sistema atualizado e integrado com outros sistemas pré-existentes. Como não podemos deixar qualquer usuário fazer as funções de administrador, esse também faz login para ter acesso às funções específicas.

Ao passar pela validação do login, esse usuário é encaminhado para [paginaAdmin.php](#) que possui as funções a serem executadas pelo administrador, que são:

- Atualização da lista de oferta através da substituição da lista antiga pela mais recente, utilizando o algoritmo de oferta previamente explicado;
- Verificar quais os monitores foram escolhidos. É importante frisar que esses dados precisam ser adicionados em outra plataforma da universidade, os quais foram disponibilizados em uma tabela contendo os dados necessários para serem copiados para o outro sistema;
- Cadastrar novos usuários: alunos podem efetuar seu próprio cadastro, porém por motivos de segurança somente administradores já logados no sistema podem cadastrar outros administradores e professores;
- Inserção de informações: essas podem ser específicas para um determinado tipo de usuário (aluno ou professor), que estarão disponíveis apenas na página referente a esse tipo de usuário, ou podem ser informações gerais, mostradas na página inicial do sistema.

Usuários do tipo administrador não têm uma distinção durante a validação, ou seja, as informações para qualquer administrador são as mesmas. O diagrama de casos de uso, mostrando o que pode ser feito por administradores está na Figura (4.20).

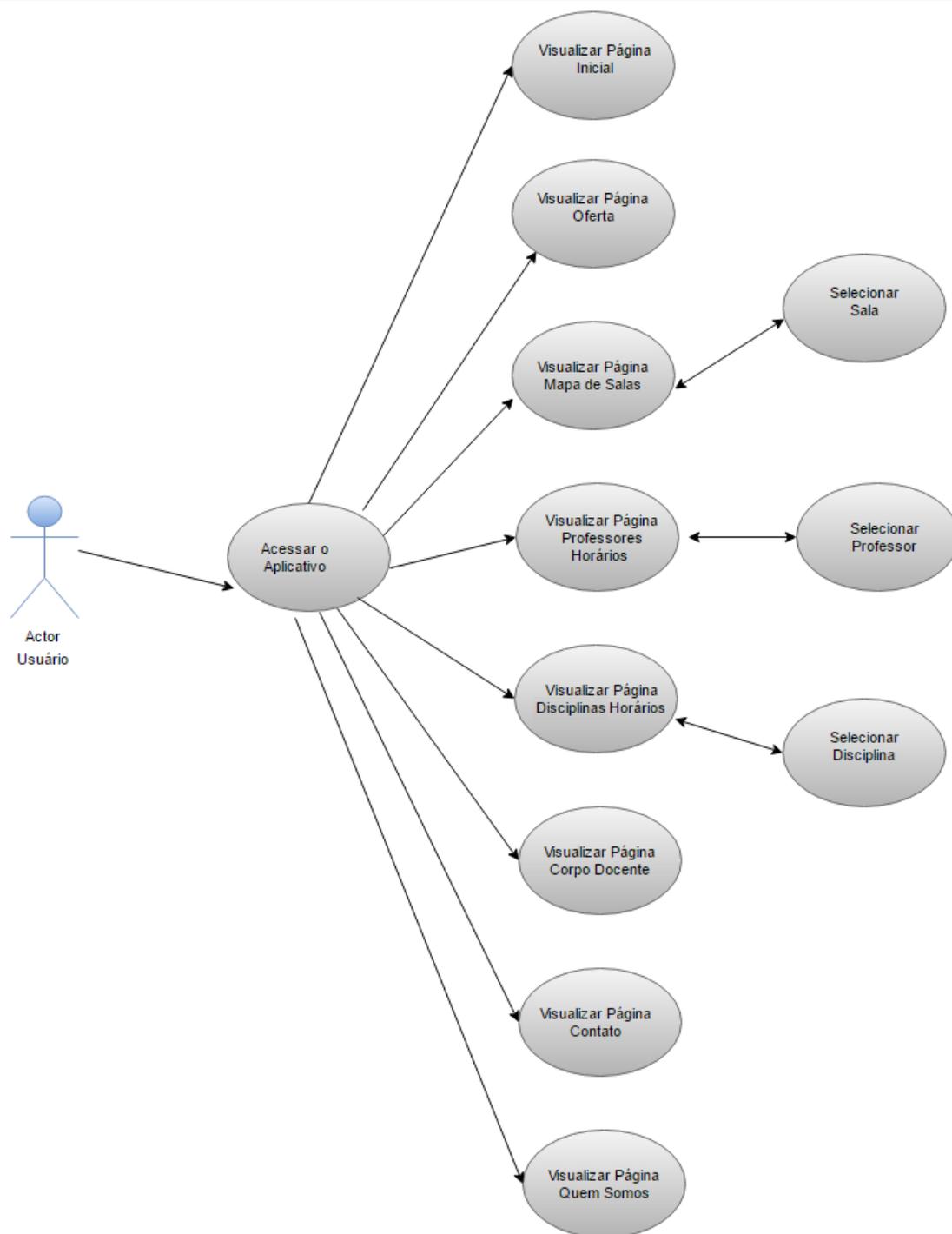


**Figura 4. 20– Diagrama de Casos de Uso para Usuário do tipo Administrador.**

Observando os diagramas de casos de usos temos uma noção geral de como os usuários podem interagir com o sistema, assim temos sabemos do que é necessário fazer para obter essas interações nos resultados.

#### **4.1.8 Desenvolvimento do Aplicativo Android**

Foi desenvolvido um aplicativo Android para que o usuário tenha ainda mais praticidade no acesso ao sistema. Esse aplicativo possui apenas funções gerais do sistema, de forma que os usuários possam ter sempre em mãos essas informações para sanar eventuais dúvidas. As funções implementadas por esse *app* são parecidas com as funções disponíveis para usuários gerais na plataforma web, como ver oferta, horários, corpo docente e informações gerais sobre o departamento e sobre o sistema, como mostrado detalhadamente no diagrama de casos de usos do aplicativo, Figura (4.21).

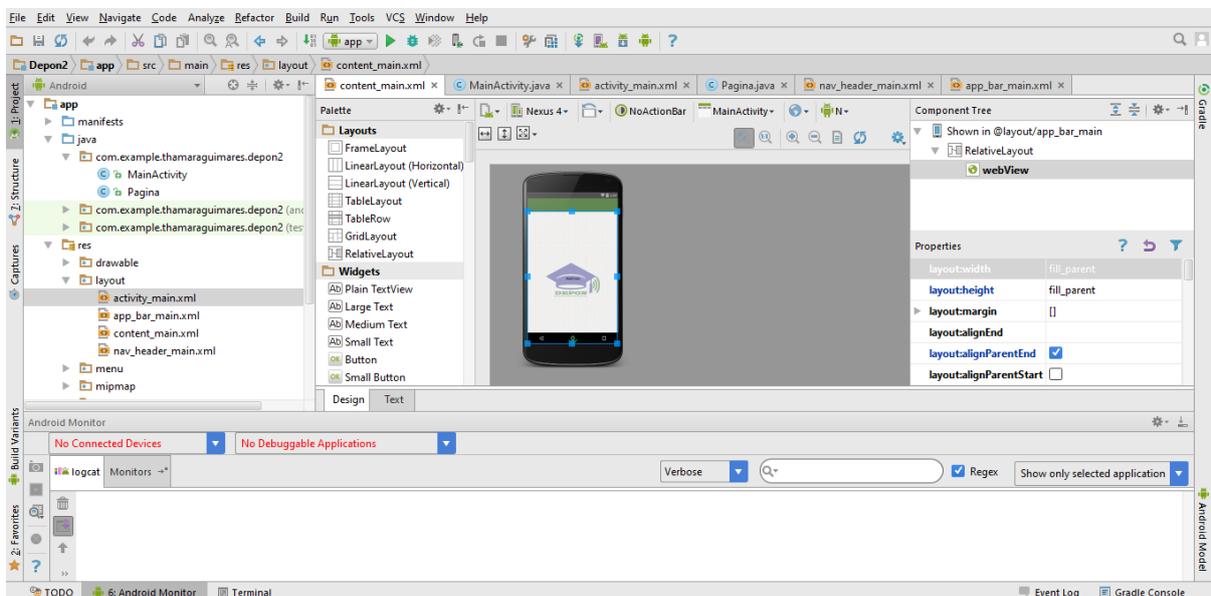


**Figura 4. 21 – Diagrama de Casos de Uso do Aplicativo Android.**

A ferramenta utilizada para implementar o *app* para dispositivos móveis foi o Android Studio. Esse *software* é gratuito e possui inúmeras funcionalidades para desenvolver diversos tipos de aplicações, inclusive uma exclusiva para web chamada WebView.

O Android WebView é um componente que possui tecnologia do Google Chrome (navegador web da Google) que permite a exibição de um conteúdo *da web* dentro da aplicação Android. Essa ferramenta vem pré-instalada nos dispositivos e precisa ser atualizada sempre

que necessário para evitar qualquer tipo de *bugs*. No aplicativo criado para o projeto o WebView foi utilizado. Definimos que toda a tela do smartphone receberia a imagem do WebView de forma a facilitar a visualização dos dados pelo usuário. Essas definições, assim como a aparência do software utilizado, estão mostradas na Figura (4.22).



**Figura 4. 22- Definição do tamanho do WebView para a aplicação através do Android Studio.**

O menu do aplicativo foi feito utilizando um dos *layouts* prontos que o Android Studio fornece chamado: *Drawer Layout*. Esse *layout* já vem com um tipo de menu chamado Navigation Drawer. Basicamente quando o usuário aperta o botão referente ao menu, uma barra com todas as opções do aplicativo escorrega da esquerda para a direita na tela do dispositivo.

A lógica para cada botão do menu é a mesma, faz-se uma chamada utilizando o WebView, o que difere é a página web ele vai apresentar na tela do aplicativo. A Figura (4.23) abaixo mostra a lógica implementada para o botão oferta.

```
ontent_main.xml x MainActivity.java x activity_main.xml x Pagina.java x nav_header_main.xml x activity_main_drawer.xml x main.xml x
} else if (id == R.id.nav_oferta) {
    //Mostra o que foi selecionado
    Toast.makeText(getApplicationContext(), "Oferta", Toast.LENGTH_LONG).show();

    WebView view = (WebView) this.findViewById(R.id.webView);
    view.getSettings().setJavaScriptEnabled(true);
    view.setWebViewClient(new Pagina());

    //Deixa a WebView Transparente
    view.setBackgroundColor(0);
    view.setLayerType(View.LAYER_TYPE_SOFTWARE, null);

    //Carrega a url
    view.loadUrl("http://164.41.222.101/tcc/paginaOferta.php");

    return true;
} else if (id == R.id.nav_mapadesalas) {
    //Mostra o que foi selecionado
    Toast.makeText(getApplicationContext(), "Mapa de Salas", Toast.LENGTH_LONG).show();

    WebView view = (WebView) this.findViewById(R.id.webView);
    view.getSettings().setJavaScriptEnabled(true);
    view.setWebViewClient(new Pagina());
}
```

**Figura 4. 23– Código parcial referente a implementação do botão oferta do menu, com utilização do WebView.**

Quando o usuário clicar em oferta irá abrir, na área delimitada anteriormente com a WebView, o conteúdo que está na url <http://164.41.222.101/paginaOferta.php>. Para que o conteúdo ficasse adaptado à tela do dispositivo móvel, foram criadas páginas específicas para o *app*. Elas possuem a mesma lógica de programação das páginas web originais, porém retiramos todo o designer gráfico que havia, com exceção das tabelas. Isso foi feito para que o fundo do aplicativo estivesse presente em todas as apresentações de conteúdo das páginas, e assim a visualização do conteúdo fosse melhor para o usuário.

É importante frisar que para melhor visualização do *app* seria mais adequado que a lista de oferta fosse feita sem a utilização de caracteres especiais, pois o aplicativo não os reconhece, e acaba gerando dificuldades na leitura do conteúdo apresentado que provém dela.

## Capítulo 5

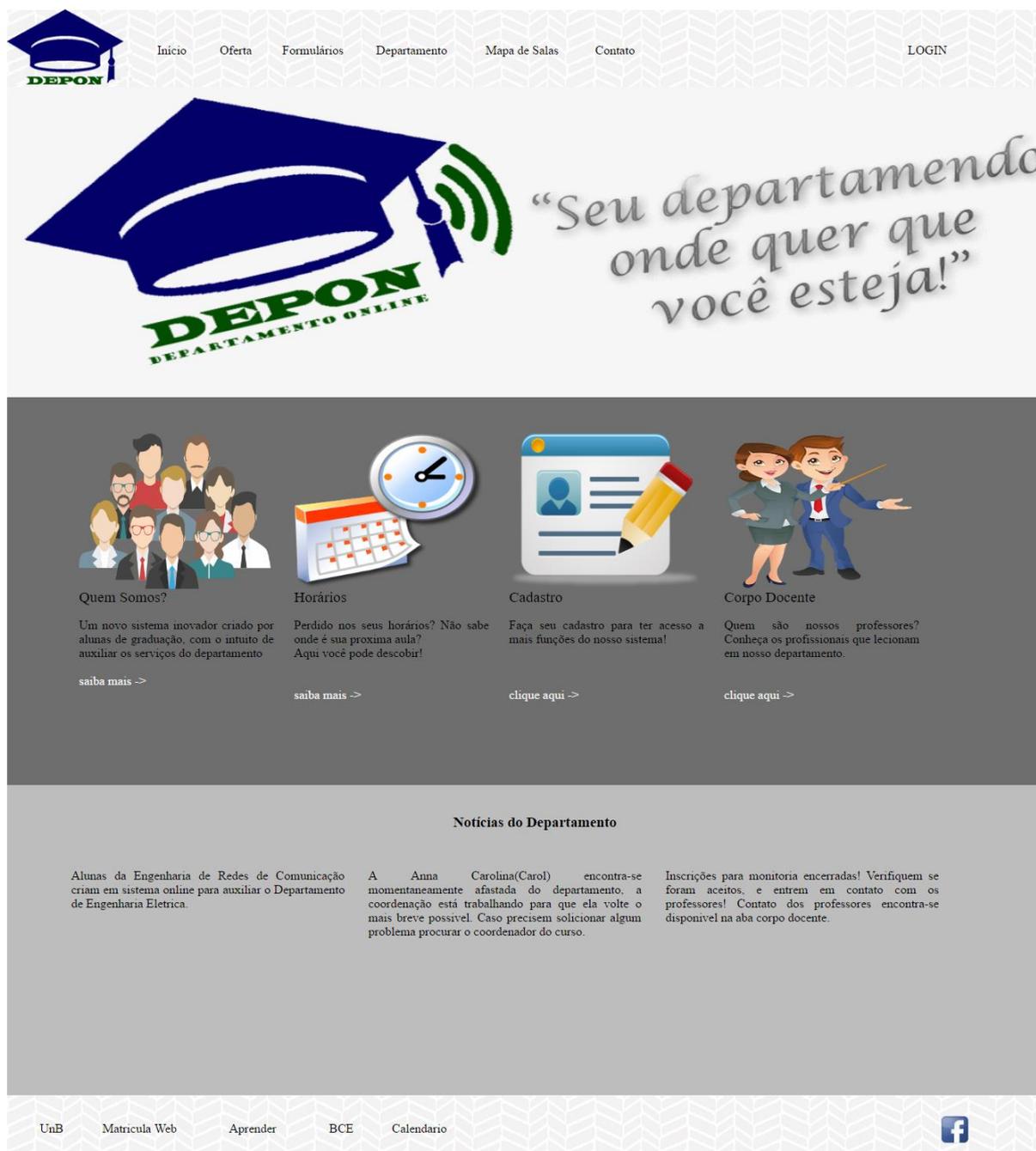
### Resultados

*Este capítulo mostra os resultados obtidos após realizar todos os processos de desenvolvimento necessários e apresentados até o momento.*

#### 5.1 Plataforma DEPON

O Depon foi desenvolvido utilizando as linguagens PHP, HTML e JavaScript para realização das funções de lógicas, CSS para designer das páginas e MySQL para banco de dados. Está hospedado no endereço <http://164.41.222.101> que pertence ao lab redes e ficará disponível no servidor por tempo limitado, posteriormente será movido para um domínio permanente. Toda a plataforma está também em um arquivo do tipo .ova que foi criada para sua execução.

A página inicial possui em seu topo um menu com links para as páginas início, oferta, formulários, departamento (página oficial do ENE), mapa de salas, contato e *login*. Em seu conteúdo o usuário tem acesso às páginas quem somos, horários, cadastro e corpo docente. Logo abaixo há um espaço destinado às notícias, e no rodapé temos alguns *links* considerados úteis, como: a página da UnB, plataforma matrícula web, *moodle*, página da biblioteca central e, por fim, a página do departamento no *facebook*. O *layout* geral dessa página está apresentado abaixo, na Figura (5.1)



**Figura 5. 1– Página Inicial.**

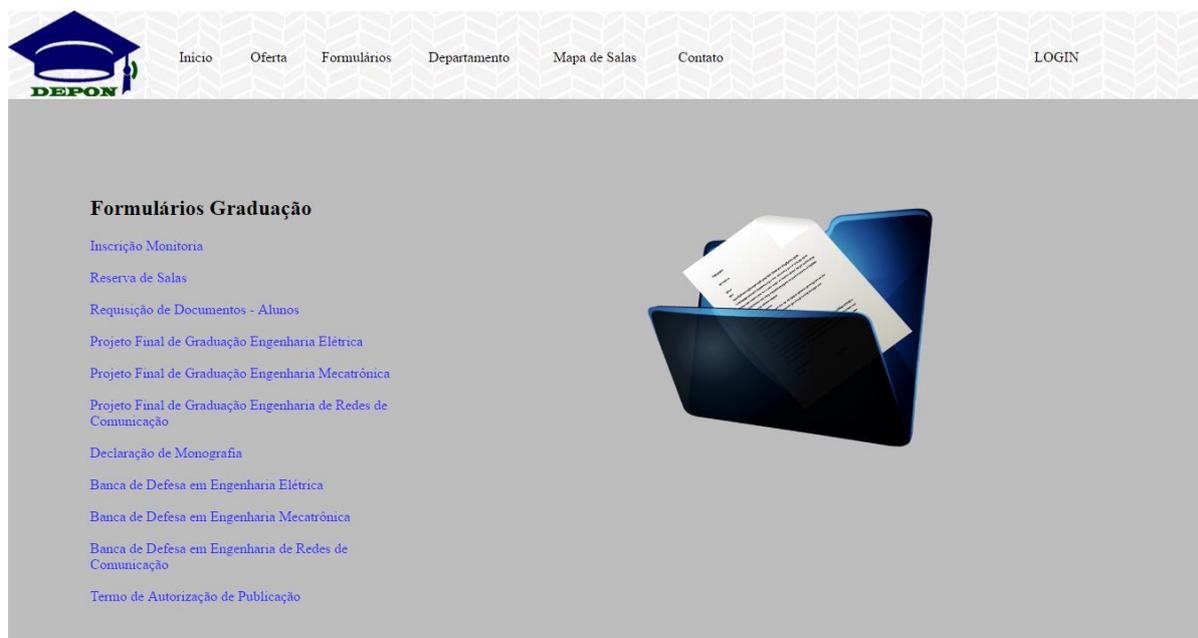
A página oferta irá apresentar uma tabela com a lista de oferta que está sendo utilizada no momento, essa pode ser atualizada pelo administrador do sistema, conforme for alterada, a mudança será efetuada automaticamente na página, levando a lista mostrada na página como sempre a última adicionada ao sistema.

Para que o usuário possa visualizar melhor a tabela em questão, optamos por não colocar menu inicial nessa página a fim de que o usuário possa abrir a lista e ainda consiga continuar navegando no sistema, sem precisar ficar utilizando o retorno no *browser*. Essa página abre automaticamente em uma nova aba. Na Figura (5.2) é possível observar esse comportamento, assim como visualizar o layout da lista de oferta.

Codigo	Disciplina	Turma	Vagas	Dias	Hora	Local	Sala	Professor	Departamento
Intr. a Engenharia Elétrica		A	40	Sex	10:00-11:50		AT-17	Edil	Elétrica
Circuitos Elétricos 1		A	60	Ter	10:00-11:50		Auditório	Franklin	Ele/Redes
Circuitos Elétricos 1		A	0	Qui	10:00-11:50		Auditório	Franklin	
Circuitos Elétricos 1		B	60	Ter	10:00-11:50		Aud SG	Alex	Ele/Redes
Circuitos Elétricos 1		B	0	Qui	10:00-11:50		Aud SG	Alex	
Circuitos Elétricos 1		C	60	Seg	16:00-17:50		Auditório	Franklin	Mec/Com
Circuitos Elétricos 1		C	0	Qua	16:00-17:50		Auditório	Franklin	
Circuitos Elétricos 1		D	60	Seg	16:00-17:50		Aud SG	Edil	Mec/Comp
Circuitos Elétricos 1		D	0	Qua	16:00-17:50		Aud SG	Edil	
Lab de Circuitos 1		A	20	Ter	10:00-11:50		Lab 2	Vargas	Ele
Lab de Circuitos 1		B	20	Qua	08:00-09:50		Lab 2	Alex	Redes
Lab de Circuitos 1		C	20	Qui	08:00-09:50		Lab 2	Vargas	Redes
Lab de Circuitos 1		D	20	Sex	10:00-11:50		Lab 2		Mec
Lab de Circuitos 1		E	20	Seg	10:00-11:50		Lab 2		Comp
Lab de Circuitos 1		F	20	Qua	10:00-11:50		Lab 2	Alex	Comp
Lab de Circuitos 1		G	20	Sex	08:00-09:50		Lab 2		Mec
Lab de Circuitos 1		H	20	Seg	08:00-09:50		Lab 2		Ele
Circuitos Elétricos 2		A	40	Seg	14:00-15:50		Auditório	Assis	Ele/Mec
Circuitos Elétricos 2		A	0	Qua	14:00-15:50		Auditório	Assis	Ele/mec

**Figura 5. 2 – Página Oferta.**

A página formulários disponibiliza os principais formulários no formato PDF para que os usuários possam realizar *downloads* quando necessário. Atualmente para ter acesso a esses formulários, é necessário fazer um pedido e retirar diretamente no departamento. Essa página tem como intuito facilitar o acesso a esses formulários, sendo também uma redução no uso de papéis pelo departamento. A estruturação dessa página está mostrada na Figura (5.3).



**Figura 5. 3 – Página Formulários.**

A página mapa de salas disponibiliza para o usuário o mapeamento do corredor do departamento, de forma a possibilitar a apresentação de onde está localizada cada sala, e de forma interativa selecionar uma delas, bem como visualizar o horário das disciplinas que serão

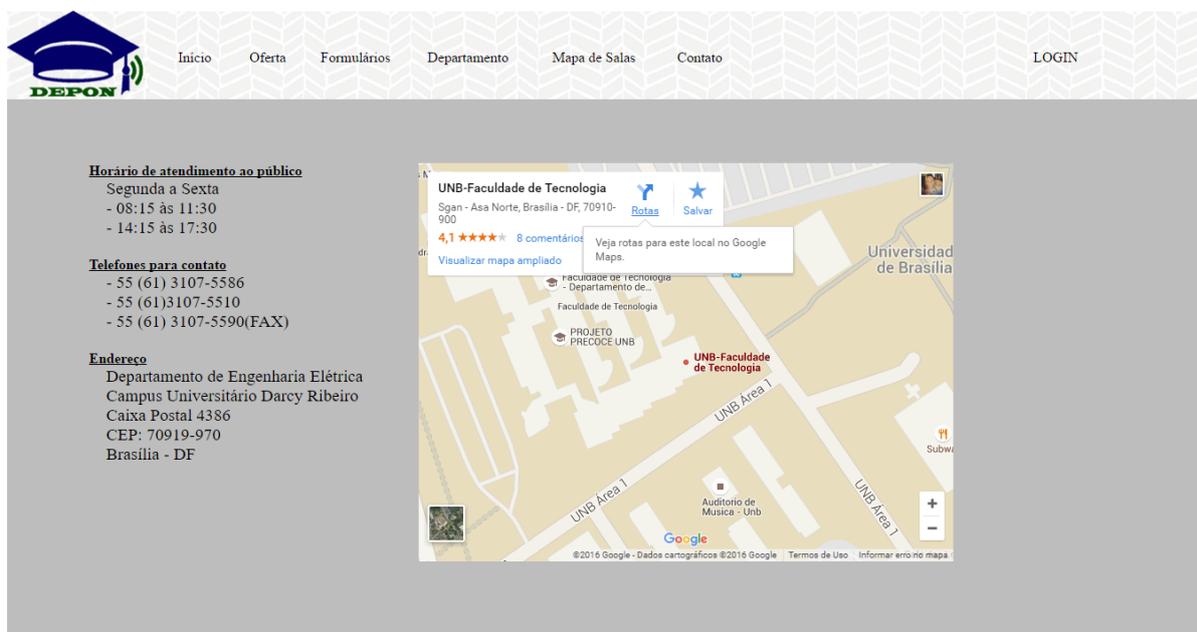
ministradas nela. Esse horário será aberto em uma janela pop-up.

Esse mapeamento utiliza a nomenclatura de salas utilizada pelo sistema matrícula web, ou seja, o aluno poderá procurar sua sala com base nos dados de sua grade horária e, assim, confirmar se houve ou não alterações na sala em que a disciplina será ministrada. Nessa página foi feita também uma equivalência entre a nomenclatura utilizada pelo matrícula web e a pelo departamento. No mapa aparece a nomenclatura utilizada pelo matrícula web, porém na janela com o horário é utilizada a nomenclatura do departamento, como pode ser visto na Figura (5.4) que mostra o exemplo da sala BT 11/15 15/13 que corresponde ao auditório.

Horario referente a Sala: Auditório		Consultar outra sala				
Dias/Horarios	Segunda	Terca	Quarta	Quinta	Sexta	Sabado
08:00-09:50		Eletrônica 1 Turma: A Prof: Stefan	Análise Dinâmica Linear Turma: A Prof: Renato	Eletrônica 1 Turma: A Prof: Stefan	Análise Dinâmica Linear Turma: A Prof: Renato	
10:00-11:50	Mat. Elétricos e Magnéticos Turma: A Prof: Artemis	Circuitos Elétricos 1 Turma: A Prof: Franklin	Mat. Elétricos e Magnéticos Turma: A Prof: Artemis	Circuitos Elétricos 1 Turma: A Prof: Franklin		
12:00-13:50						
14:00-15:50	Circuitos Elétricos 2 Turma: C Prof: Assis		Circuitos Elétricos 2 Turma: C Prof: Assis			
16:00-17:50	Circuitos Elétricos 1 Turma: C Prof: Franklin		Circuitos Elétricos 1 Turma: C Prof: Franklin			

Figura 5. 4 – Página Mapa de Salas com exemplo de horário referente a sala.

O [link Contato](#) traz informações básicas sobre o departamento, como: horário de atendimento ao público, telefones para contato e localização. A localização está mostrada de duas formas em um mapa obtido através do *google maps*, que mostra não só o departamento, mas também a FT, prédio onde se encontra do departamento de engenharia elétrica da UnB, e seu endereço, como pode ser visto na Figura (5.5) abaixo.



**Figura 5. 5– Página Contato.**

A última opção do menu principal é o link Login, o qual possibilita que usuários cadastrados no sistema possam acessar outras informações de acordo com seu tipo de usuário. Caso o usuário em questão ainda não tenha cadastro nessa página, há também um link para efetuar novos cadastros. O layout utilizado na página login encontra-se na Figura (5.6).



**Figura 5. 6– Página Login.**

No corpo da página inicial temos outras páginas com funções importantes (quem somos, cadastro, horários e corpo docente) que serão mostradas mais detalhadamente abaixo.

A página quem somos faz uma descrição dos objetivos do sistema, o que é a plataforma e de onde veio a proposta de realizar esse projeto, e mostra ao usuário quem foram as desenvolvedoras. A estruturação dessa página está na Figura (5.7).



**Figura 5. 7– Página Quem Somos.**

A página cadastro dá a opção do usuário se cadastrar no sistema para ter acesso a mais informações e outras aplicações do sistema. Para efetuar o cadastro algumas informações são necessárias (nome e matrícula do usuário), assim como a definição de usuário e senha para que possa ter acesso a tudo que o sistema tem a oferecer. Para esse cadastro, apenas usuários do tipo Aluno podem se cadastrar, o registro de outro tipo de usuário somente será efetuado pelo administrador do sistema por questões de segurança. Na Figura (5.8) vemos a página cadastro, e todas as informações necessárias para efetuá-lo conforme descrito.



**Figura 5. 8– Página Cadastro.**

Ao clicar no [link](#) para verificar os horários o usuário é levado à página que lhe dá a opção

de fazer essa consulta de acordo com a melhor forma de pesquisar o que deseja, ou seja, através do nome da sala, da disciplina ou do professor. Segue abaixo um exemplo de como é apresentado o mapeamento, Figura (5.9).

DEPON

Inicio Oferta Formulários Departamento Mapa de Salas Contato LOGIN

Horários por Salas

Horários por Disciplina

Horários por Professores

Não sabe onde é sua aula seleccione as salas e veja quais matérias estão sendo lecionadas em cada sala.

Sabe qual é a disciplina, verifique aqui informações sobre essa, como local onde está sendo lecionada.

Verifique aqui as disciplinas lecionadas pelos nossos professores.

Selecione a Sala:

Selecione a disciplina:

Selecione o Professor:

Selecione

Enviar

Selecione

Enviar

Selecione

Enviar

**Figura 5. 9– Página Horários.**

Os dados presentes nas caixas de seleção vêm diretamente de um acesso à tabela oferta do banco de dados. Ao selecionar uma das opções será mostrado o horário referente a escolha do usuário, conforme mostrado no exemplo da Figura (5.10). No exemplo o usuário utilizou a opção selecionar disciplina, e escolheu Circuitos Elétricos 1, e foi encaminhado para outra página que continha as informações sobre essa disciplina: professores que a ministrarão disciplina, sala onde a aula ocorrerá e turma. Esses dados então disponíveis em forma de uma grade horária. Caso o usuário deseje ver informações sobre outra disciplina, existe a opção na página. Caso contrário é possível continuar a navegar pelo site através do menu inicial.

DEPON

Inicio Oferta Formulários Departamento Mapa de Salas Contato LOGIN

Horario referente a disciplina: Circuitos Elétricos 1 | Consultar outra disciplina

Dias/Horarios	Segunda	Terca	Quarta	Quinta	Sexta	Sabado
08:00-09:50						
10:00-11:50		Alex Turma: B Sala: Aud SG		Alex Turma: B Sala: Aud SG		
12:00-13:50						
14:00-15:50						
16:00-17:50	Edil Turma: D Sala: Aud SG		Edil Turma: D Sala: Aud SG			
18:00-19:50						

**Figura 5. 10– Horário referente a disciplina Circuitos Elétricos 1.**

Existe também a opção de conhecer e ter acesso às informações dos professores que compõem o departamento. Basta clicar no link referente ao corpo docente, e as informações disponíveis serão mostradas em uma nova aba, conforme mostrado na Figura (5.11). Essas informações são lidas de uma tabela do banco de dados com o corpo docente e mostradas em forma de tabela.

Professor	Ramal	Email
Adolfo Bauchspiess	31075571	adolfo@unb.br
Adoniram Judson de Barros Braga	31075745	jbraga@ene.unb.br
Alcides Leandro Silva	31075579	alleand@unb.br
Alex da Rosa	31075572	alex@ene.unb.br
Alexandre Ricardo Soares Romariz	31075559	romariz@ene.unb.br
Anderson Clayton Alves Nascimento	0	andclay@ene.unb.br
André Noll Barreto	31075564	andrebarreto@ene.unb.br
Anésio de Leles Ferreira Filho	31075592	leles@ene.unb.br
Antonio José Martins Soares	31075561	martins@ene.unb.br
Antônio Padilha L. Bó	31071040	antonio.plb@lara.unb.br
Artemis Marti Ceschin	31075741	artemis@ene.unb.br
Cláudia Jacy Barenco Abbas	0	barenco@ene.unb.br
Daniel Guerreiro e Silva	31075647	danielgs@unb.br
Eduardo Peixoto Fernandes da Silva	31075647	eduardopeixoto@jeee.org
Eduardo Stockler Tognetti	31075556	estognetti@ene.unb.br
Felipe Vigolvinio Lopes	31075647	felipevlopes@unb.br
Flavia Maria Guerra de Sousa Aranha Oliveira	31075649	flavia@ene.unb.br
Flavio Elias de Deus	31075744	flavioelias@unb.br
Francisco Assis de Oliveira Nascimento	31075570	assis@ene.unb.br

**Figura 5. 11– Página Corpo Docente.**

### 5.1.1 Login

Existem 3 tipos de usuário: Administrador, Aluno e Professor. Cada um possui suas respectivas funcionalidades.

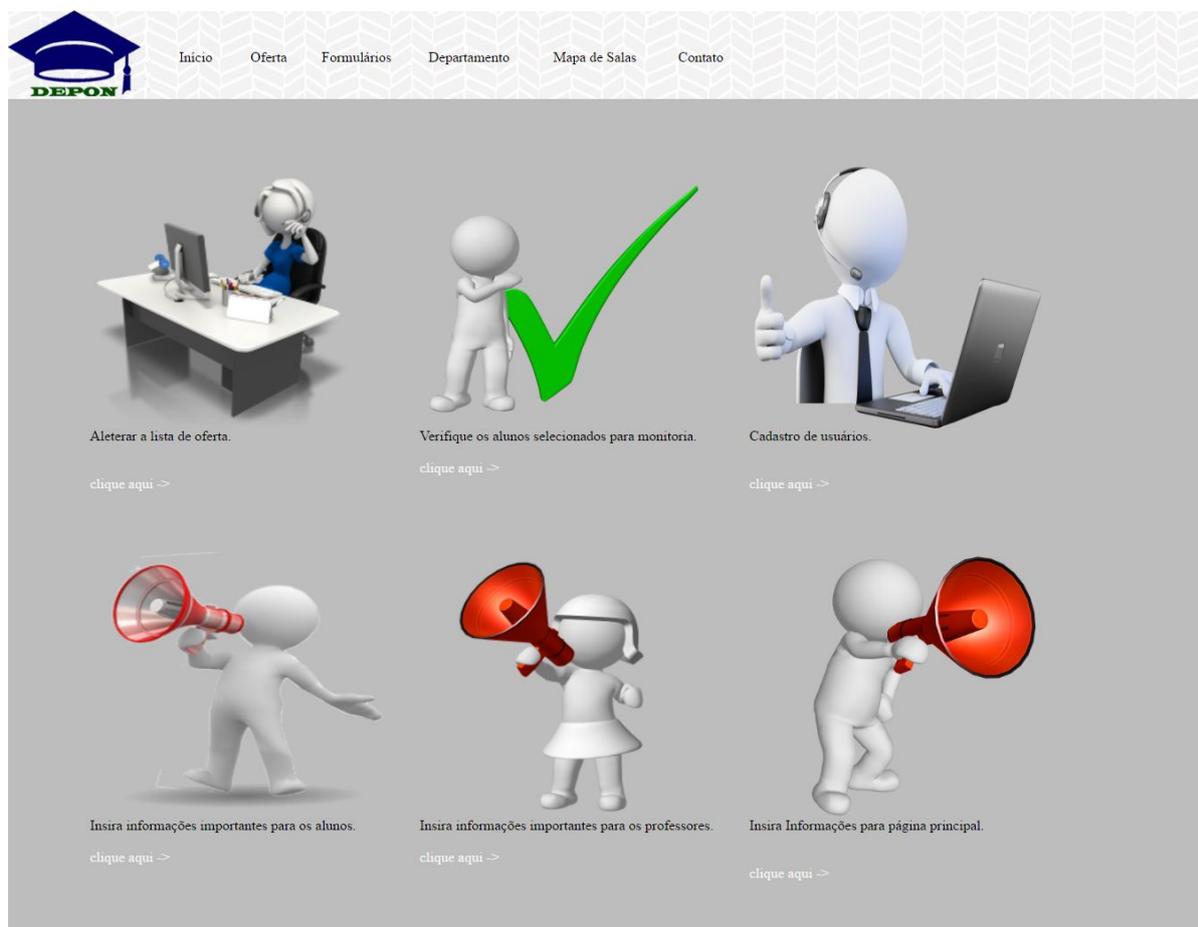
O administrador tem como principal função manter o sistema atualizado, ele pode alterar a lista de oferta facilmente de forma que o sistema esteja sempre com a lista vigente, pois muitas operações do sistema são dependentes da lista de oferta, por isso essa precisa estar sempre atualizada. Também com o intuito de manter o sistema atualizado o administrador pode inserir informações para as páginas início, professor e aluno.

O administrador também verifica alunos selecionados para monitoria e passa essa informação para o sistema SIGRA, um sistema já utilizado pela universidade para matrículas, para que os alunos sejam matriculados como monitores e possam receber as vantagens desse posto.

A última função do administrador é efetuar cadastro de novos usuários no sistema, como vimos anteriormente, alunos podem se cadastrar, porém professores e novos administradores precisam ser cadastrados pelo administrador, isso ocorre para garantir que nenhum usuário se cadastre com a categoria incorreta e assim tenha acesso a outras partes do sistema.

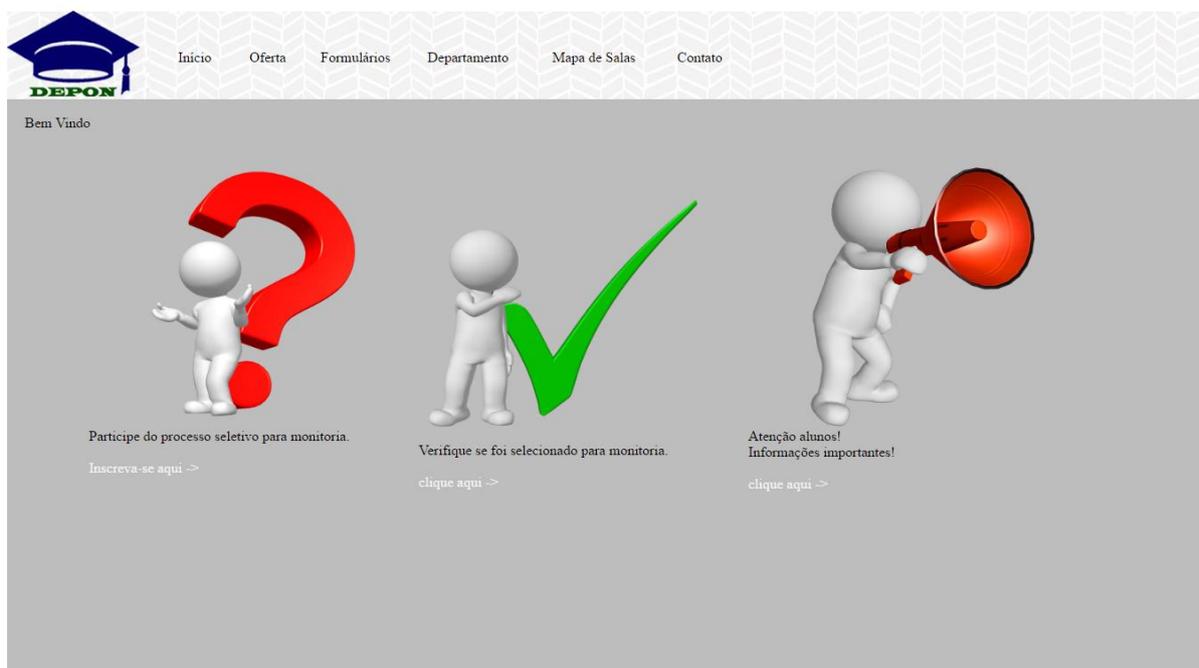
As divisões das tarefas na página do administrador foram feitas para que ele tenha

facilidade em fazer as alterações no sistema, o *layout* dessa página encontra-se na Figura (5.12), que possui *links* para todas as atividades descritas.



**Figura 5. 12– Página Administrador.**

Na página aluno, Figura (5.13) vemos as funções que podem ser executadas pelo usuário do tipo aluno, essas são: cadastro para o processo seletivo de monitoria e verificar se foi selecionado ou não como monitor. Além dessas duas funções o aluno também terá acesso a algumas informações mais específicas para esse tipo de usuário, como: faltas ou atrasos de professores, informações sobre estágios e projetos.



**Figura 5. 13 – Página Aluno**

A página professor, Figura (5.14), possui as opções que podem ser efetuadas pelos professores. Verificam os alunos que se inscreveram para serem seus monitores. Lembrando que cada professor verifica as informações referentes a si de forma individualizada, assim cada professor só tem acesso aos alunos que se inscreveram para serem monitores das disciplinas ministradas por ele. Tendo conhecimento dos inscritos, os professores os aceitam ou não como monitores. Além disso os professores podem fazer alterações em seus cadastros.

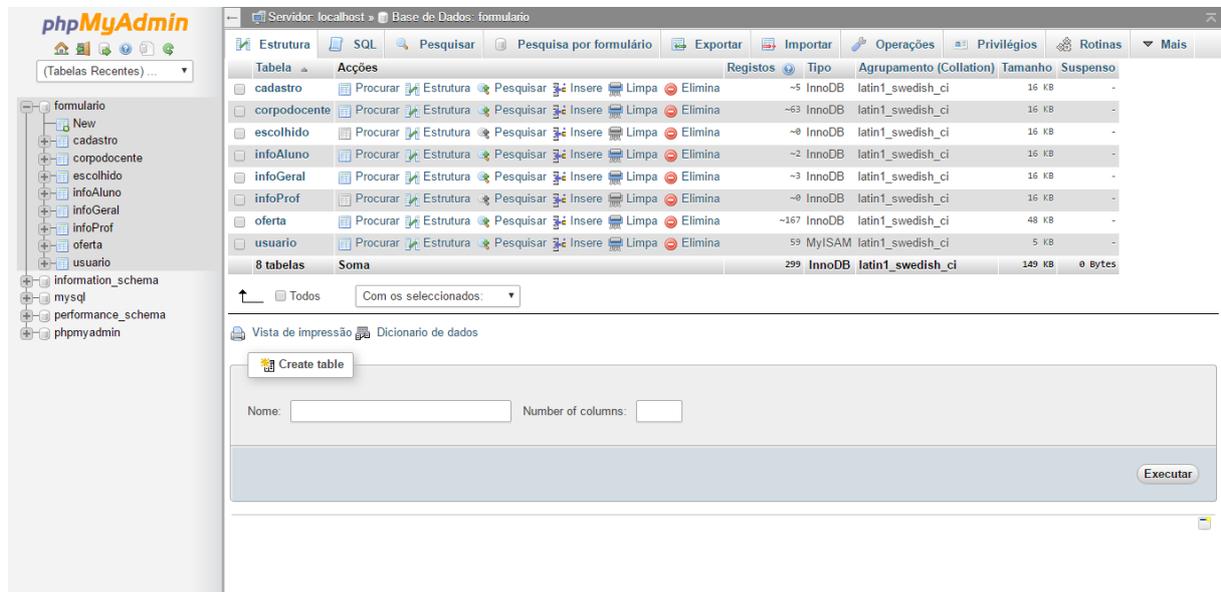
Os professores também tem um link de informações, onde ficam informações pertinentes a eles como: reuniões e comunicados do departamento.



**Figura 5. 14– Página Professor.**

## 5.2 Banco de Dados

Para armazenar todos os dados do projeto foi criado um BD MySQL chamado formulário composto por várias tabelas onde cada uma armazena dados específicos a alguma funcionalidade. É importante salientar que todas as tabelas possuem o campo de id para que possam ser enumerados os cadastros. Segue abaixo a Figura (5.15) que mostra a construção desse banco e suas tabelas.



**Figura 5. 15– Banco de dados e as tabelas que o compõem criadas através do gerenciador de BD phpmyadmin.**

- Cadastro: É responsável por armazenar os dados necessários de cada aluno que realizou inscrição para monitoria. Algumas dessas informações serão mostradas na página do professor para que ele escolha qual monitor ele selecionará;
- Corpo Docente: Guarda os dados nome, telefone e e-mail de todos os professores que compõem o corpo docente do departamento e essas informações são mostradas na página de corpo docente da página *web* e no aplicativo Android;
- Escolhido: É idêntica a tabela cadastro, porém ela guarda informações apenas dos alunos que foram selecionados pelos professores. Essas informações são apresentadas na página do administrador para que ele dê seguimento ao processo de monitoria, e também são acessadas pelo aluno para que ele saiba se foi ou não escolhido como monitor;
- InfoAluno, infoGeral e infoProf: Guardam as informações que o administrador irá inserir em cada página *web*, ou seja, a infoAluno irá guardar aquelas informações destinadas aos alunos, a infoGeral, as que serão apresentadas na página inicial e a infoProf, as que são destinadas aos professores. As composições das tabelas são idênticas;
- Oferta: Guarda as informações referentes à lista de ofertas. Esse banco de dados pode ser alterado pelo administrador na opção de alterar oferta através do upload de um novo arquivo

CSV que contém todas as informações de oferta de disciplinas, professores e salas. É dessa tabela que vem a montagem das tabelas de mapa de salas seja qual for o tipo de busca, pois ela contém todas as informações necessárias para montar as grades do departamento. Isso é válido tanto para a página web quanto para o aplicativo Android;

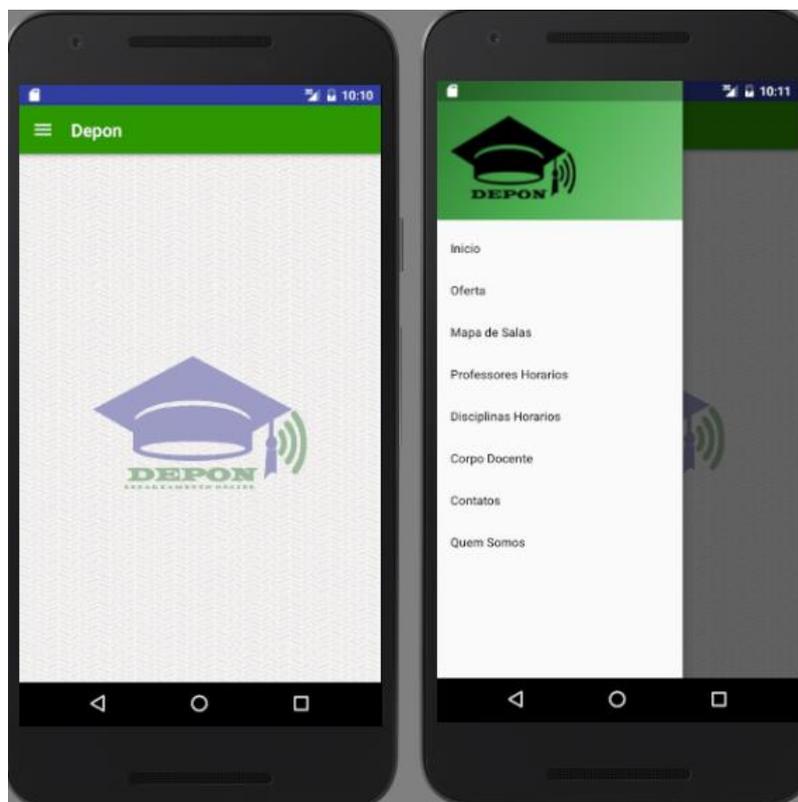
- Usuário: Guarda as informações dos usuários cadastrados para ter acesso ao sistema. Ao se cadastrar na página de cadastros são incluídos os novos dados do usuário. Quando o usuário preenche os dados da página de *login* é feito uma varredura nessa tabela do banco de dados comparando as informações cadastradas com as digitadas e caso sejam idênticas, é liberado o acesso, caso contrário é exibida a mensagem de *login* inválido.

O sistema foi desenvolvido para o bom funcionamento da plataforma como um todo, porem alguns problemas ainda persistiram e, devido a falta, de tempo não foi levado em conta o fator segurança da aplicação, o que leva a alguns problemas críticos.

Para garantir a integridade das informações, dados devem ser protegidos para que essas não possam ser alterados por usuários maliciosos. Essa implementação de segurança está como prioridade para trabalhos futuros. Outro problema de segurança é o quesito acesso; foi implementado um sistema de *login*, contudo ainda não foi implementada uma autenticação segura de usuários. Assim um usuário que souber o caminho de determinada página pode acessá-la e fazer modificações, fato esse que se apresenta como um significativo problema que demandará um trabalho que garanta essa autenticação segura para todas as páginas de usuário de forma a limitar o acesso, assegurando um melhor funcionamento da plataforma.

### **5.3 Aplicativo Android**

Ao iniciarmos o aplicativo é mostrado ao usuário o fundo com a logo criada para o sistema e denominado como Depon juntamente com o botão de menu. Ao clicar nesse botão uma barra com as opções de funcionalidades do aplicativo escorrega da lateral esquerda do dispositivo móvel ocupando até a metade da tela, as opções do menu são: início, oferta, mapa de salas, professores horários, disciplinas horários, corpo docente, contato e corpo docente, o que cada uma dessas opções faz este descrito abaixo. A tela inicial e o menu estão mostrados na Figura (5.16).



**Figura 5. 16– Tela inicial do app e a apresentação de seu menu.**

O menu é composto por 7 botões:

- Início – Retorna para a tela inicial do aplicativo;
- Oferta – Apresenta a lista de oferta;
- Mapa de Sala – O usuário escolhe a sala e pode ver quais disciplinas, turmas e professores estão alocados em cada horário nela, o comportamento dessa função está exemplificado na Figura (5.17);

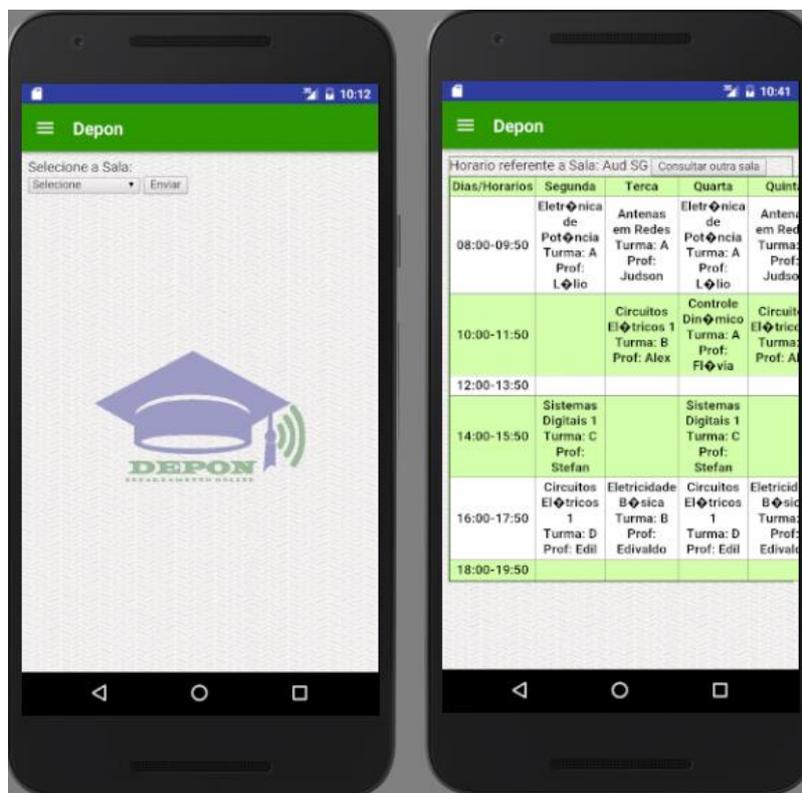


Figura 5. 17– Tela do app ao selecionar a opção mapa de salas e tela referente à sala auditório.

- Professores Horários – O usuário escolhe o professor e pode ver onde e quando ele estará lecionando determinada disciplina, conforme o exemplo, Figura(5.18)

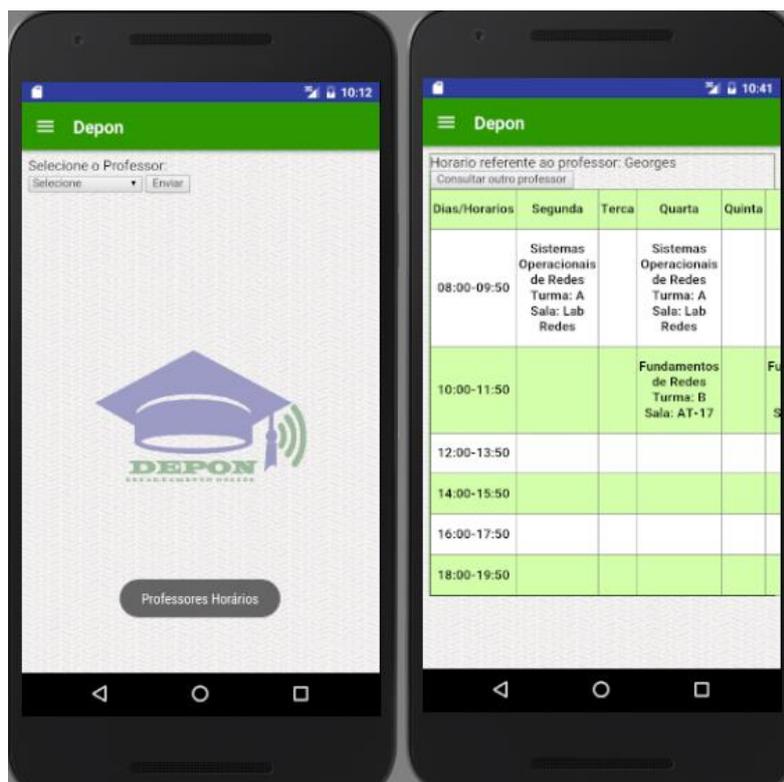
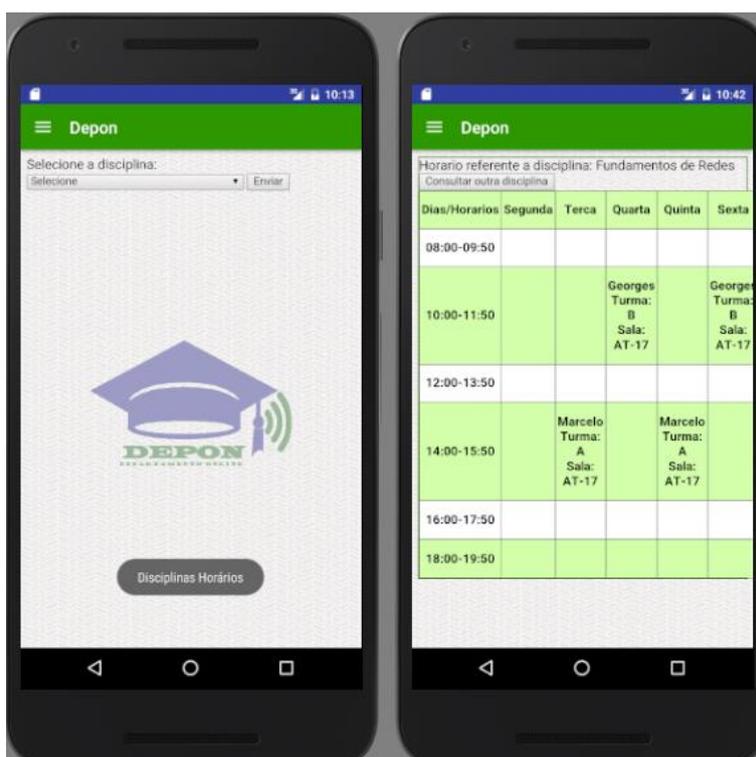


Figura 5. 18– Tela do app ao selecionar a opção professores horários e tela referente ao professor Georges.

- Disciplinas Horários – O usuário escolhe a disciplina e pode ver os horários,

professores e salas onde está acontecendo as aulas daquela matéria, o horário por disciplinas está exemplificado na Figura (5.19);



**Figura 5. 19 – Tela do app ao selecionar a opção disciplinas horários e tela referente à disciplina fundamentos de redes.**

- **Corpo Docente** – Mostra informações de todos os professores que compõem o corpo docente do ENE, lendo as informações do banco de dados e imprimindo-as em uma tabela, da mesma forma que é feita na página *web*. A apresentação dessa tabela no aplicativo está na Figura (5.20);

Professor	Ramal	Email
Adolfo Bauchspiess	31075571	adolfob@unb.br
Adoniram Judson de Barros Braga	31075745	jbraga@ene.unb.br
Alcides Leandro Silva	31075579	alleand@unb.br
Alex da Rosa	31075572	alex@ene.unb.br
Alexandre Ricardo Soares Romariz	31075559	romariz@ene.unb.br
Anderson Clayton Alves Nascimento	0	andclay@ene.unb.br
André Noll Barreto	31075564	andrebarreto@ene.unb.br
Anésio de Leles Ferreira Filho	31075592	leles@ene.unb.br

Figura 5. 20 – Tela do app ao selecionar a opção corpo docente.

- Contatos – Mostra informações para contato direto com o departamento (Horário de funcionamento, endereço, telefones, etc.) apresentado na tela conforme mostrado na Figura (5.21);

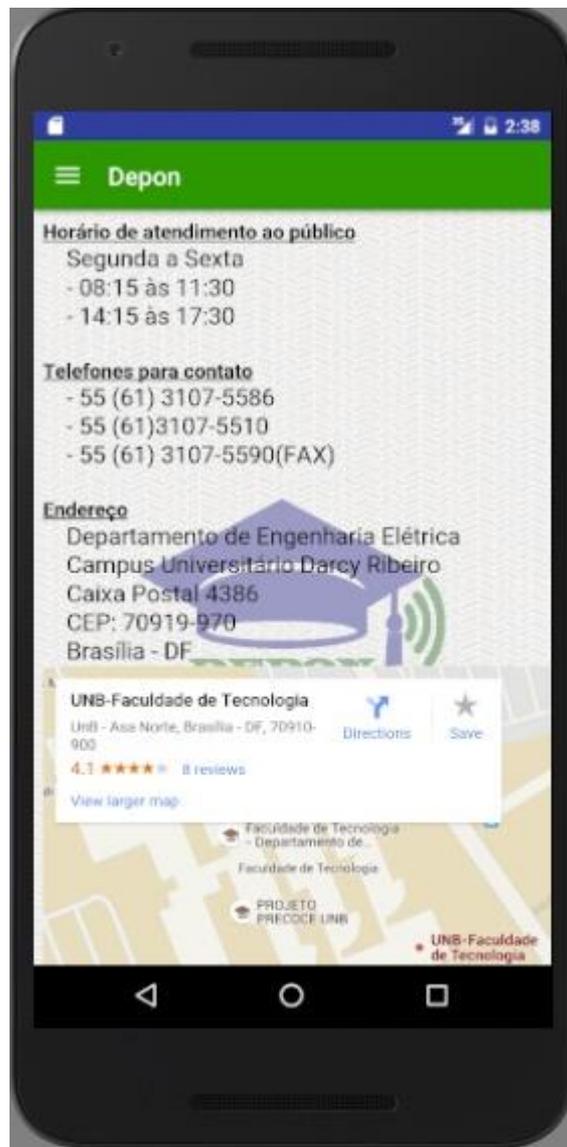


Figura 5. 21 – Tela do app ao selecionar a opção contatos.

- Quem somos – Informações sobre quem desenvolveu o aplicativo, como ele funciona e porque decidiram desenvolvê-lo, está ilustrado na Figura (5.22).



Figura 5. 22– Tela do app ao selecionar a opção quem somos.

## Capítulo 6

### Conclusão e Trabalhos Futuros

*Este capítulo apresenta a as conclusões do trabalho, e sugestões para trabalhos futuros.*

Este projeto foi implementado com o objetivo de desenvolver uma página web que fosse capaz de otimizar diversas atividades importantes e corriqueiras do Departamento de Engenharia Elétrica da Universidade de Brasília (ENE-UnB). Atividades essas que abrangem funcionários administrativos, professores e alunos. Com esse projeto cada tipo de usuário poderá ter acesso a respectivas funcionalidades. Alunos podem se candidatar e verificar o resultado de monitoria, professores têm acesso ao banco de dados de todos os alunos que se inscreveram para sua disciplina e assim será capaz de realizar sua escolha. Já o administrador pode realizar novos cadastros de qualquer tipo de usuário, atualizar a lista de oferta e inserir informações específicas e/ou gerais que aparecem em suas páginas logo após realizarem login. Também foi desenvolvido um aplicativo Android que mostra de maneira fácil e prática o mapeamento de salas, lista de oferta e informações do corpo docente.

A metodologia usada para atingirmos o objetivo do projeto foi baseada em pesquisas bibliográficas acerca de todas as tecnologias viáveis e anteriormente já citadas, implementadas em ambiente LINUX. Inicialmente o foco da pesquisa foi obter informações e conhecimentos sólidos que capacitassem a decisão de ferramentas e tecnologias que melhor se adaptassem ao propósito almejado.

É importante frisar que o projeto foi desenvolvido pensando em sanar necessidades específicas do departamento de engenharia elétrica da UnB, mas isso não exclui a possibilidade de ser usado por outros departamentos, porém é necessário fazer adaptações de acordo com a finalidade desejada. Essa plataforma pode ter um alcance ainda maior que o desenvolvido, a ideia é propagá-lo para outras instituições de educação que possuem as mesmas necessidades.

O objetivo do projeto de desenvolver uma plataforma *web* e aplicativo para auxiliar o departamento foi alcançado, porém ainda precisa de melhorias para ser amplamente utilizado.

Para trabalhos futuros foram levantados as seguintes possibilidades:

- Adicionar a possibilidade de reservar salas, diretamente pelo sistema;
- Permitir mostrar o problema de choque de horário de maneira versátil;
- Adicionar funções de usuário ao aplicativo;
- Aperfeiçoamento do designer da página web e aplicativo;
- Melhorar a parte visual da página mapa de salas e adicionar o SG-11;

- Otimizar as restrições e trazer mais segurança ao sistema.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1]Departamentos. Disponível em: <https://www.ufpr.br/portalufpr/a-universidade-institucional/departamentos/>. Acesso em: 10/05/2016
- [2]Unidades Acadêmicas. Disponível em: [http://www.unb.br/unidades\\_academicas](http://www.unb.br/unidades_academicas). Acesso em: 10/05/2016
- [3]Estatuto e Regimento Geral – UnB. Disponível online em: [http://www.unb.br/unb/transparencia/downloads/regimento\\_estatuto\\_unb.pdf](http://www.unb.br/unb/transparencia/downloads/regimento_estatuto_unb.pdf). Acesso em: 10/05/2016
- [4]Departamento de Engenharia Elétrica. Disponível em: <http://www.ene.unb.br/>. Acesso em: 10/05/2016
- [5]O que é GNU/Linux. Disponível em: <https://www.vivaolinux.com.br/linux/> Acesso em: 23/02/2016
- [6]The Apache Software Foundation. Disponível em: <http://www.apache.org/foundation/>. Acesso em: 23/02/2016
- [7]Conhecendo o Servidor Apache. Disponível em: <http://www.infowester.com/servapach.php>. Acesso em: 23/02/2016
- [8]MySQL. Disponível em: <http://www.mysql.com/> Acesso em: 23/02/2016
- [9]DB-Engines Ranking - Trend Popularity. Disponível em: [http://db-engines.com/en/ranking\\_trend](http://db-engines.com/en/ranking_trend). Acesso em 02/06/2016
- [10]Manual do PHP. Disponível em: [https://secure.php.net/manual/pt\\_BR/preface.php](https://secure.php.net/manual/pt_BR/preface.php). Acesso em: 23/02/2016
- [11]phpMyAdmin About. Disponível em: <https://www.phpmyadmin.net/>. Acesso em: 23/02/2016
- [12]HTML. Disponível em: <https://www.w3.org/html/>. Acesso em: 26/02/2016
- [13]A História do HTML. Disponível em: <http://www.frontendbrasil.com.br/artigos/a-historia-do-html/>. Acesso em: 26/02/2016
- [14]PHP do Jeito Certo. Disponível em: <http://br.phptherightway.com/>. Acesso em:26/02/2016
- [15]CSS Tutorial. Disponível em: <http://www.w3schools.com/css/>. Acesso em: 28/02/2016
- [16]JavaScript Tutorial. Disponível em: <http://www.w3schools.com/js/>. Acessado em: 28/02/2016
- [17>Welcome to the Android Open Source Project! Disponível em: <http://source.android.com/> Acessado em: 28/02/2016
- [18]Android Vs. iOS: What's The Most Popular Mobile Operating System In Your Country? Disponível em: <http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892>. Acessado em: 28/02/2016
- [19]Android Studio - The Official IDE for Android. Disponível em: <http://developer.android.com/intl/pt-br/sdk/index.html#Requirements>. Acesso em 14/04/2016
- [20]O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. Disponível em: <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 07/05/2016
- [21]Introduction to OMG's Unified Modeling Language. Disponível em: <http://www.uml.org/what-is-uml.htm>. Acesso em: 07/05/2016

- [22]MySQL Documentation. Disponível em: <http://dev.mysql.com/doc/>. Acesso em:13/11/2015
- [23] PHP Documentation. Disponível em: <http://php.net/docs.php>. Acesso em:13/11/2015
- [24]Fórum PHP. Disponível em: <http://forum.imasters.com.br/forum/3-php/>. Acesso em: 20/11/2015
- [25]Iniciantes ao PHP. Disponível em: <http://phpbrasil.com/phorum/>. Acesso em:13/04/2015
- [26]Nixon, R. Learning PHP, MySQL & JavaScript with jquery, css & html5. 4ª ed. Estados Unidos: O'Reilly, 2015.
- [27]Beighley, L.; Morrison, M. Use a Cabeça! PHP & MySQL: O Guia Amigo do Seu Cérebro. Rio de Janeiro: Atlas Books Editora, 2010.
- [28]MySQL Fórum. Disponível em: <http://forum.imasters.com.br/forum/67-mysql/>. Acesso em: 20/11/2015
- [29]Costa, C.; Fernandes; A.; Melo, A. C.; Gomes; L.M.; Guerra, H. Integration of university information systems using Web Services. Santiago de Compostela 2010. Disponível online em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5556691&queryText=institution%20web%20systems&newsearch=true>. Acesso em:16/06/2016
- [30]Bo, X.; Wang, C.; Yan, H.; Zhou, C. Assessment system design for university web-based learning. Chongqing, 2010. Disponível online em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5607653&queryText=web%20systems%20for%20universities&newsearch=true>. Acesso em: 16/06/2016
- [31]Pichetti, J. G. B. Sistema web para gerenciamento de bancas de trabalhos acadêmicos. Pato Branco 2013. Disponível online em: [http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/2020/1/PB\\_COADS\\_2013\\_2\\_06.pdf](http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/2020/1/PB_COADS_2013_2_06.pdf). Acesso em: 16/06/2016
- [32]Cerqueira, A. C. Sistema de controle de alunos em estágio curricular. Recife, 2008. Disponível online em: [http://tcc.ecomp.poli.br/20081/anthony\\_v3.0\\_FINAL.pdf](http://tcc.ecomp.poli.br/20081/anthony_v3.0_FINAL.pdf). Acesso em: 16/06/2016
- [33]Getting Started: WebView-based Applications for Web Developers. Disponível online em: <https://developer.chrome.com/multidevice/webview/gettingstarted>. Acesso em: 16/04/2016
- [34]Android - WebView Tutorial. Disponível online em: [http://www.tutorialspoint.com/android/android\\_webview\\_layout.htm](http://www.tutorialspoint.com/android/android_webview_layout.htm). Acesso em: 20/04/2016
- [35]Xiaoling, Z.; Qitu, H.; Xiao, D. Design and development of web-assisted instruction system for university physics cours. Busan – Coreia do Sul, 2011. Disponível online em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6016638&queryText=web%20systems%20for%20universities&newsearch=true>. Acesso em: 16/06/2016

## **Apêndice A**

### **A1 - Instalação da pilha LAMP**

#### **Apache**

A instalação do apache foi feita através do terminal do linux, usando os seguintes comandos:

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

O apt é um gerenciador de pacotes do Ubuntu, ele nos permite instalar uma grande variedade de softwares mantidos em um repositório.

#### **MySQL**

Para a instalação do banco de dados MySQL também utilizamos o repositório apt, de acordo com o seguinte comando.

```
sudo apt-get install mysql-server php5-mysql
```

Durante a instalação o servidor solicita a configuração de uma senha para o usuário *root* do MySQL. O usuário *root* é um super usuário, ele possui mais privilégios do que os demais usuários. Assim que a senha é digitada ele pede uma confirmação, uma vez que as duas sejam correspondentes a instalação segue.

#### **PHP**

A instalação do PHP na máquina se deu pelo comando a seguir:

```
sudo apt-get install php5 libapache2-mod-php5 php5-mysql
```

Como padrão o apache usa o index.html para que seja utilizado o index.php e necessário fazer essa configuração, para isso alteramos o nome no padrão diretamente no arquivo de configuração do apache em seguida reiniciamos o apache para que as novas configurações passassem a valer, usando o comando:

```
sudo /etc/init.d/apache2 restart
```

#### **Phpmyadmin**

Como as demais instalações o phpmyadmin também foi instalado utilizando o comando apt-get no terminal do Linux.

```
sudo apt-get install phpmyadmin
```

Por padrão o phpMyAdmin está em outro diretório que não o `var/www` que é o diretório do apache, mesmo que durante a configuração tenha sido definido o apache como o servidor utilizado, logo é necessário mudar o diretório, essa alteração foi feita usando o comando a seguir.

```
sudo ln -s /user/share/phpmyadmin /var/www/
```

Para finalizar a instalação é necessário fazer novamente um *restart* no servidor *web*, para garantir que todas as configurações feitas no último passo foram atualizadas no apache.

```
sudo /etc/init.d/apache2 restart
```

Assim todas as instalações foram feitas e a máquina está agora pronta para ser utilizada para o desenvolvimento do sistema.