



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Desenvolvimento de um Sistema Web  
para Controle de Faltas da  
Universidade Federal de Ouro Preto**

**Bárbara Chesman Almeida**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:

Fernando Bernardes de Oliveira

COORIENTAÇÃO:

João Pedro Santos de Moura

**Fevereiro, 2018  
João Monlevade–MG**

**Bárbara Chesman Almeida**

**Desenvolvimento de um Sistema Web para  
Controle de Faltas da Universidade Federal de  
Ouro Preto**

Orientador: Fernando Bernardes de Oliveira

Coorientador: João Pedro Santos de Moura

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Fevereiro de 2018**

A447d Almeida, Bárbara Chesman.  
Desenvolvimento de um sistema web para controle de faltas da  
Universidade Federal de Ouro Preto [manuscrito] / Bárbara Chesman  
Almeida. - 2018.

52f.: il.: color; grafs; tabs.

Orientador: Prof. Dr. Fernando Bernardes de Oliveira.  
Coorientador: Prof. MSc. João Pedro Santos de Moura.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de  
Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de  
Informação.

1. Sistemas de informação. 2. Software - Desenvolvimento. 3. Sistemas de  
recuperação da informação. I. Oliveira, Fernando Bernardes de. II. Moura, João  
Pedro Santos de. III. Universidade Federal de Ouro Preto. IV. Título.

# FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

## Desenvolvimento de um Sistema Web para Controle de Faltas da Universidade Federal de Ouro Preto

Bárbara Chesman Almeida

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 - Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



---

Fernando Bernardes de Oliveira  
Prof Dr.  
DECSI - UFOP



---

João Pedro Santos de Moura  
Bacharel  
ICMC - USP



---

George Henrique Godim da Fonseca  
Prof. Dr.  
Examinador  
DECSI - UFOP



---

Gilda Aparecida de Assis  
Profª Drª.  
Examinador  
DECSI - UFOP

João Monlevade, 8 de fevereiro de 2018

## ATA DE DEFESA

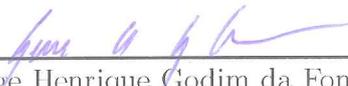
No dia 8 do mês de Fevereiro de 2018, às 17:15 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Bárbara Chesman Almeida**, sendo a Comissão Examinadora constituída pelos professores: Fernando Bernardes de Oliveira, João Pedro Santos de Moura, George Henrique Godim da Fonseca, Gilda Aparecida de Assis. O(a) candidato(a) apresentou a monografia intitulada: “Desenvolvimento de um Sistema Web para Controle de Faltas da Universidade Federal de Ouro Preto”. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,8 ( nove pontos e oito décimos), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).



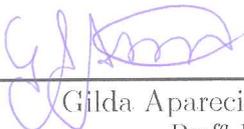
Fernando Bernardes de Oliveira  
Prof Dr.  
DECSI – UFOP



João Pedro Santos de Moura  
Bacharel  
ICMC – USP



George Henrique Godim da Fonseca  
Prof. Dr.  
Examinador(a)  
DECSI – UFOP



Gilda Aparecida de Assis  
Prof<sup>a</sup> Dr<sup>a</sup>.  
Examinador(a)  
DECSI – UFOP



Bárbara Chesman Almeida

## TERMO DE RESPONSABILIDADE

Eu, Bárbara Chesman Almeida declaro que o texto do trabalho de conclusão de curso intitulado "*Desenvolvimento de um Sistema Web para Controle de Faltas da Universidade Federal de Ouro Preto*" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 8 de fevereiro de 2018.



---

Bárbara Chesman Almeida

*Aos meus pais Ivete e Karil, minha irmã Luíza e a toda minha família que com muito carinho e apoio me incentivaram à enfrentar todos os obstáculos de minha vida.*

# Agradecimentos

A todos os professores do curso, amigos do Diretório Acadêmico, do Laboratório iMobilis, da INCOP, da ATLMARJOM e da salinha de estudos, obrigada por participarem dessa etapa de crescimento pessoal e acadêmico. Em especial ao Bruno Lacerda, parceiro durante todo o curso e que encerra essa conquista ao meu lado.

À república Mandala, que juntas demonstraram que amizades e companheirismo, mesmo que dotados de muitas diferenças pessoais, são os elementos essenciais para o convívio em grupo, além da irmandade que nos uniu. Especialmente, agradeço a amiga Pâmela Freitas por ter me acolhido e por todos os abraços e insultos carinhosos que trocamos ao longo dessa jornada.

Ao Rafael Bortoline, por seu exemplo, amizade, conselhos, cuidados e amor.

Agradeço aos meus orientadores Fernando e João Pedro, por terem me acompanhado nessa etapa, acreditarem na minha capacidade para desenvolver esse projeto e terem me guiado durante o decorrer deste trabalho.

Por fim, a todos que contribuíram para meu crescimento pessoal e profissional, muito obrigada!

*“Determinação coragem e autoconfiança são fatores decisivos para o sucesso.  
Se estamos possuídos por uma inabalável determinação conseguiremos superá-los.  
Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos  
de orgulho.”*

— Dalai Lama (2001),  
*in: O Livro de Dias.*

# Resumo

O uso de ferramentas de *software* para a melhoria de processos permite a desenvolvedores projetar e construir sistemas de informação para gerenciar e controlar processos. As metodologias utilizadas na UFOP para controle atualmente não permitem a divulgação do número de faltas por parte dos docentes de maneira apropriada e quando um grande número de discentes são matriculados em determinada turma, o processo de controle de frequência se torna custoso. Portanto, objetivo pretendido nesse trabalho é o desenvolvimento de um sistema *web* para informatizar e agilizar o processo, além de melhorar a metodologia de controle de assiduidade da UFOP. Durante o desenvolvimento foram utilizadas técnicas da engenharia de software que serviram de auxílio durante todo o processo. O sistema foi desenvolvido utilizando a linguagem PHP e o *framework* Laravel juntamente com o banco de dados MySQL. Com base nos resultados obtidos em testes após a conclusão do projeto, o sistema atende o processo de controle de faltas da UFOP e é capaz de substituir as metodologias utilizadas atualmente.

**Palavras-chaves:** Controle de faltas. Desenvolvimento de *Software*. *Web*.

# Abstract

The use of software tools for process improvement allows developers to design and build information systems to manage and control processes, in order to streamline tasks and improve existing methodologies in UFOP. The methodologies used for class frequency control currently do not allow the disclosure of the number of absences by the teachers in an appropriate manner, another issue observed is that, when a large number of students are enrolled in a certain class, the process of frequency control becomes costly. Therefore, this work proposes the design and development of a web control system for absences, which allows, through a list of attendance, to present to the students and teachers the list of absences. From the observation of the difficulties faced by students and professors in the control of attendance, it was verified the necessity of the production of this work. During development, software engineering techniques were used to aided the entire process. The system was developed using the PHP language and the Laravel framework together with the MySQL database. This work also addresses the main software tests by which the proposed tool was submitted as a form of validation and verification, resulting in an increase of the quality of the final product. Based on the obtained results in tests after project completion, the system complies with the UFOP's class attendance control process and is able to replace the currently used methodologies.

**Keywords:** Frequency control. *Software Development. Web.*

# Lista de ilustrações

Figura 1 – Diagrama de Casos de Uso . . . . .	26
Figura 2 – Modelo Entidade Relacionamento - UFOP Faltas . . . . .	27
Figura 3 – Importação de Turmas - UFOP Faltas . . . . .	28
Figura 4 – Tela de <i>Login</i> - UFOP Faltas . . . . .	29
Figura 5 – Tela Dados da Turma . . . . .	30
Figura 6 – Tela de Importação de Turma . . . . .	30
Figura 7 – Tela de Seleção Data . . . . .	30
Figura 8 – Tela de Lançamento de Faltas . . . . .	31
Figura 9 – Diário de Classe Semestral . . . . .	32
Figura 10 – Dedicção relativa em horas de reparo . . . . .	35
Figura 11 – Qual metodologia que você utiliza para registrar faltas? . . . . .	36
Figura 12 – Qual o nível de conhecimento e relacionamento com sistemas web? . . . . .	37
Figura 13 – Sua metodologia considera a presença por aula ou por dia? . . . . .	37
Figura 14 – Você substituiria sua metodologia por um sistema web que faz o que sua metodologia faz, inclusive geração de relatório? . . . . .	38
Figura 15 – Tela - Tutorial de Utilização do UFOP Faltas . . . . .	47

# Lista de tabelas

Tabela 1 – Funcionalidades - Administrador . . . . .	24
Tabela 2 – Funcionalidades - docente . . . . .	25
Tabela 3 – Funcionalidades - discente . . . . .	25
Tabela 4 – Problemas reportados pelos usuários . . . . .	34

# Lista de abreviaturas e siglas

**AD** *Active Directory*

**CSV** *Comma-Separated Values*

**DIT** *Directory Information Tree*

**MER** *Modelo Entidade Relacionamento*

**ICEA** Instituto de Ciências Exatas e Aplicadas

**LDAP** *Lightweight Directory Access Protocol*

**MVC** *Model-View-Controller*

**NTI** Núcleo de Tecnologia da Informação

**PMBOK** *Project Management Body of Knowledge*

**SGBD** Sistema de Gerenciamento de Banco de Dados

**SIG** Sistema Integrado de Gestão

**UFOP** Universidade Federal de Ouro Preto

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Problema</b>	<b>16</b>
<b>1.2</b>	<b>Objetivos</b>	<b>17</b>
1.2.1	Objetivos Específicos	17
<b>1.3</b>	<b>Justificativa</b>	<b>17</b>
<b>1.4</b>	<b>Estrutura do trabalho</b>	<b>18</b>
<b>2</b>	<b>TECNOLOGIAS E TRABALHOS CORRELATOS</b>	<b>19</b>
<b>2.1</b>	<b>Materiais e Métodos</b>	<b>19</b>
2.1.1	MySQL	19
2.1.2	Plataforma de desenvolvimento	19
2.1.3	Serviços de Diretório e LDAP	20
<b>2.2</b>	<b>Sistemas correlatos</b>	<b>21</b>
2.2.1	Formei UFOP	21
2.2.2	Presente!	21
2.2.3	Plataforma minhaUFMG e Moodle UFOP	21
<b>2.3</b>	<b>Considerações finais</b>	<b>22</b>
<b>3</b>	<b>MODELAGEM DO SISTEMA</b>	<b>23</b>
<b>3.1</b>	<b>BOPE</b>	<b>23</b>
<b>3.2</b>	<b>Análise de Requisitos</b>	<b>23</b>
3.2.1	Usuários do Sistema	24
3.2.2	Funcionalidades	24
<b>3.3</b>	<b>Casos de Uso</b>	<b>25</b>
3.3.1	Diagrama de Caso de Uso	25
<b>3.4</b>	<b>Banco de Dados</b>	<b>26</b>
3.4.1	Modelagem Conceitual	26
3.4.2	Importação de planilhas	27
<b>3.5</b>	<b>Considerações finais</b>	<b>28</b>
<b>4</b>	<b>APRESENTAÇÃO DA APLICAÇÃO DESENVOLVIDA</b>	<b>29</b>
<b>4.1</b>	<b>Autenticação</b>	<b>29</b>
4.1.1	Área do docente	30
4.1.2	Área do discente	31
<b>4.2</b>	<b>Teste de <i>software</i></b>	<b>32</b>
4.2.1	Testes Unitários	33

4.2.2	Testes de Integração . . . . .	33
4.2.3	Testes de sistema . . . . .	34
<b>4.3</b>	<b>Entrevista com os usuários . . . . .</b>	<b>35</b>
<b>4.4</b>	<b>Considerações finais . . . . .</b>	<b>38</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>39</b>
<b>5.1</b>	<b>Considerações Finais . . . . .</b>	<b>39</b>
<b>5.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>39</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>41</b>
	<b>APÊNDICES . . . . .</b>	<b>42</b>
	<b>APÊNDICE A – ESTÓRIAS DE USUÁRIO . . . . .</b>	<b>43</b>
<b>A.1</b>	<b>Critérios de aceitação sobre os requisitos do projeto UFOP Faltas . . . . .</b>	<b>43</b>
A.1.1	Serviço de Autenticação . . . . .	43
A.1.2	Serviço de Visualização de Informações . . . . .	43
	<b>APÊNDICE B – CENÁRIOS DE TESTE . . . . .</b>	<b>45</b>
B.0.1	Serviço de Autenticação . . . . .	45
B.0.2	Serviço de Visualização de Informações . . . . .	45
	<b>APÊNDICE C – QUESTIONÁRIO DE VALIDAÇÃO DO SISTEMA . . . . .</b>	<b>46</b>
	<b>APÊNDICE D – TUTORIAL . . . . .</b>	<b>47</b>
	<b>APÊNDICE E – DIÁRIO DE CLASSE UFOP FALTAS . . . . .</b>	<b>48</b>
	<b>APÊNDICE F – CONFIGURAÇÃO DO AMBIENTE . . . . .</b>	<b>50</b>
<b>F.1</b>	<b>Pré-requisitos . . . . .</b>	<b>50</b>
F.1.1	Pré-requisitos do Laravel, temos: . . . . .	50
<b>F.2</b>	<b>Instalação . . . . .</b>	<b>50</b>
F.2.1	Configuração . . . . .	51

# 1 Introdução

O gerenciamento de faltas é fundamental para auxiliar docentes e discentes, já que o número de faltas pode ser considerado como um dos critérios de avaliação para a aprovação nas disciplinas. O processo atual de gerenciar faltas da Universidade Federal de Ouro Preto (UFOP) é por meio do diário impresso em cursos presenciais. Porém, esse acompanhamento pode ser feito, por exemplo, em *softwares* ou em planilhas eletrônicas. Ao final do semestre, o número total de faltas deve ser lançado na plataforma *Minha UFOP*, e um diário de classe impresso deve ser entregue ao chefe de departamento de filiação do docente.

A importância do controle de faltas no ambiente acadêmico se deve à frequência mínima exigida para a aprovação dos estudantes, dada a relevância de que o aluno esteja no ambiente de aprendizado e que a presença seja contabilizada.

Por isso, é importante ter um controle adequado de assiduidade dos discentes. Para que o gerenciamento seja prático, o melhor meio é informatizando o processo para torná-lo uma metodologia aperfeiçoada, o que não ocorre nesse ambiente, pelo fato dos docentes terem metodologias de controle diferentes.

## 1.1 Problema

Quando um grande número de discentes são matriculados em determinada turma, o processo de controle de frequência pode se tornar custoso. Os docentes precisam fazer a chamada em cada aula. Ao final do semestre, eles precisam contabilizar os lançamentos e informar na plataforma *Minha UFOP*.

As metodologias utilizadas para controle não permitem a divulgação do número de faltas por parte dos docentes de maneira apropriada. Assim, essa situação deixa os discentes sem essa informação.

Como solução para atender a esse tipo de demanda surgiu a necessidade de um Sistema Integrado de Gestão (SIG), que fornecerá uma estrutura para registrar as faltas ao longo do semestre, podendo assim gerar relatórios e estudos para a análise do número de faltas. Portanto, com a criação da aplicação proposta neste trabalho, intitulada *UFOP Faltas*, os docentes poderão controlar de maneira apropriada a frequência dos discentes, bem como permitir que eles controlem e verifiquem suas faltas.

## 1.2 Objetivos

O objetivo principal deste trabalho é criar um ambiente que permita o armazenamento das faltas e que realize todo o gerenciamento de assiduidade, atendendo a demanda problematizada na seção anterior. Devido ao fato de não existir um portal para gerenciar faltas e o processo não ser padronizado, o objetivo proposto é controlar a assiduidade através de um sistema *Web*. A partir desse ambiente, além do usuário visualizar os dados armazenados, terá a opção de gerar relatórios de controle.

Para a realização desse projeto é necessário o desenvolvimento de uma aplicação que utiliza as mesmas etapas do gerenciamento de assiduidade da **UFOP** para realizar o controle de faltas. É necessário apenas que o docente envie o arquivo de turma, tendo este, discentes matriculados na turma que está sendo cadastrada. Dessa maneira, a aplicação permitirá o controle e lançamento da assiduidade pelos docentes e o monitoramento por parte dos discentes.

A seguir, os objetivos específicos do trabalho são apresentados.

### 1.2.1 Objetivos Específicos

Este trabalho possui aos seguintes objetivos específicos:

1. Prover por meio do *software* proposto uma plataforma centralizada que possibilite o gerenciamento do diário de classe.
2. Desenvolver um sistema responsivo e padronizado a fim de substituir o processo de gerenciar frequência manual.
3. Permitir que os usuários acessem o sistema *Web* a partir dos mesmos dados de autenticação de acesso à *Minha UFOP*
4. Realizar testes durante o desenvolvimento e com os usuários de forma a permitir a descobertas de erros e sua respectiva correção de forma a aumentar a qualidade do produto final.

## 1.3 Justificativa

Com o objetivo de desenvolver uma plataforma de faltas comum para os cursos presenciais, sua construção se faz importante para a instituição, principalmente por agilizar o gerenciamento de frequência para os docentes e acompanhamento dos discentes. Levando em consideração que a plataforma utilizará um **SIG** é esperado que a implantação da aplicação contribua positivamente para a melhoria do processo de controle de frequência, além de beneficiar docentes e discentes.

Atualmente, a UFOP não possui nenhum sistema de controle de assiduidade oficial, seja desenvolvido pelo Núcleo de Tecnologia da Informação (NTI) ou por terceiros. Desse modo, os docentes são livres para escolher a maneira de gerenciar. Com os dados acessíveis em um sistema *web*, além de auxiliar os docentes no controle de frequência, permite que os discentes tenham acesso direto à elas.

Não obstante, o NTI foi contactado para que a proposta desse projeto fosse validada. Até o momento, nenhum sistema tem sido desenvolvido para gerenciar faltas nos cursos presenciais.

Assim, com o desenvolvimento da aplicação, tanto o docente quanto o discente podem ter acesso à todas essas informações de maneira a fazer o acompanhamento de maneira eficiente.

## 1.4 Estrutura do trabalho

O restante deste trabalho é organizado como segue. No Capítulo 2 é apresentada a fundamentação teórica do trabalho desenvolvido, as ferramentas, as tecnologias e os sistemas relacionados. O Capítulo 3 descreve os requisitos e a modelagem do sistema. No Capítulo 4 são apresentados os detalhes da implementação, funcionalidades e metodologia utilizada no desenvolvimento do projeto, além dos resultados e testes do projeto realizados a fim de garantir a validação do sistema. Por fim, são apresentadas as considerações finais e propostas de trabalhos futuros no Capítulo 5.

## 2 Tecnologias e trabalhos correlatos

Este capítulo aborda a fundamentação das tecnologias visando uma melhor compreensão dos mesmos. Conceitos, tecnologias e informações que foram utilizados no desenvolvimento deste projeto serão apresentados. A Seção 2.1 discute as ferramentas e tecnologias utilizadas para o desenvolvimento e na Seção 2.2 são apresentados os sistemas correlatos e uma breve comparação com o sistema Web proposto.

### 2.1 Materiais e Métodos

As principais tecnologias de desenvolvimento do sistema são as mesmas que estão sendo utilizadas pelo NTI do Instituto de Ciências Exatas e Aplicadas (ICEA), de modo que a integração, a manutenção e a escalabilidade do projeto sejam realizadas de maneira apropriada. As ferramentas e as tecnologias utilizadas no ambiente de desenvolvimento do *Ufop Faltas* são:

- **Plataforma:** *Web*;
- **Servidor:** *Apache 2.4*;
- **Linguagens:** *PHP 7.0, HTML5, Javascript, JQuery, CSS, Node.js 6*;
- **Framework:** *Laravel 5.3*;
- **Banco de Dados:** *MySQL 5.7.57*;
- **API Autenticação:** *LD(AP)I*.

#### 2.1.1 MySQL

O armazenamento dos dados é realizado por meio do [MySQL \(2017\)](#). O segundo Sistema de Gerenciamento de Banco de Dados (SGBD) mais popular em uso de acordo com [Engines \(2017\)](#), conhecido pela sua riqueza de funções e da grande quantidade de informações, além de possuir várias funções de segurança já integradas. Neste trabalho foram utilizados os dados dos usuários da UFOP, por meio da API de autenticação para obter os dados de acesso do usuário na *MinhaUFOP*.

#### 2.1.2 Plataforma de desenvolvimento

Segundo [Otwell \(2017\)](#), o Laravel é um *framework* PHP utilizado para desenvolvimento web, que utiliza o padrão estrutural *Model-View-Controller* (MVC). Sua principal

característica é o desenvolvimento de aplicações seguras e com boa performance de modo rápido, com código simples e limpo, já que o Laravel incentiva o uso de boas práticas de programação.

O *framework* Laravel foi utilizado em busca da estruturação organizada do projeto e do aumento da produtividade e reaproveitamento de código. Na camada de visualização dos dados, onde a lógica de apresentação da interface do usuário é realizada, foi utilizado o Admin LTE como *template*. O Admin LTE é um *dashboard template* totalmente responsivo e gratuito que é disponibilizado no site: <<https://almsaeedstudio.com>>, e é baseado no uso do framework Bootstrap 3. Quanto à comunicação com o banco de dados para autenticação no sistema é utilizada uma implementação chamada *boilerplate*, que traz funcionalidades que facilitam a autenticação por causa das informações do usuário e grupos de acesso. A utilização do *boilerplate* permite acessar os dados de autenticação da *Minha UFOP* por meio da **LD(AP)I**.

Logo, com uma configuração simples e pequena é possível se conectar ao banco de dados da **UFOP** e controlar o acesso de sua aplicação com os mesmos dados. Essa autenticação é feita pelo protocolo de acesso a diretórios **LD(AP)I**<sup>1</sup>, desenvolvida por Plínio Roque de Almeida e João Pedro Santos de Moura. A aplicação busca facilitar o processo de autenticação de aplicações que usam autenticação baseadas em servidores *Active Directory (AD)* e a recuperação de dados de tais servidores por meio de pesquisas com sintaxe amigável.

### 2.1.3 Serviços de Diretório e LDAP

De acordo com Pereira (2009), o *Lightweight Directory Access Protocol (LDAP)* é um protocolo para fazer acesso a serviços de diretórios distribuídos. Os dados se encontram em formato de uma árvore de nós, sendo que cada nó consiste de um conjunto de atributos com seus respectivos valores.

A **UFOP** utiliza o *OpenLDAP 2.4*, uma implementação *open-source* do **LDAP**, para armazenar diversas informações sobre a universidade. O *Directory Information Tree (DIT)* utilizado pela **UFOP** possui informações sobre três tipos: *grupos*, *maquinas* e *people*. As unidades de *grupos* e *people* foram utilizadas no *boilerplate* a fim de obter os seguintes dados: *cpf*, *nome*, *grupo pertencente*, *campus* e *e-mail* para validar a autenticação dos usuários, permitindo que com os mesmos dados de acesso ao portal *Minha UFOP*, qualquer sistema que o implemente também tenha acesso.

A seção a seguir apresenta aplicações correlatas ao projeto desenvolvido. Elas foram utilizadas para identificar possíveis funcionalidades e restrições para o sistema.

<sup>1</sup> <https://github.com/jpmoura/ldapi>

## 2.2 Sistemas correlatos

Nesta seção serão apresentados alguns sistemas correlatos à plataforma desenvolvida. Para a definição foram observados os seguintes fatores: funcionalidades, métodos empregados e aplicação.

### 2.2.1 Formei UFOP

Santos (2016) propôs o *Formei UFOP*, uma aplicação móvel desenvolvida para facilitar o gerenciamento de notas do *campus* ICEA. O aplicativo móvel é voltado para os estudantes da *UFOP* acompanharem seu progresso acadêmico, desenvolvido com o principal objetivo de planejar e gerenciar o progresso dos discentes. O aplicativo possui as seguintes funcionalidades: avaliação do desempenho escolar; definição de prioridades de disciplinas a cada semestre e previsão de conclusão do curso. Esse trabalho tem como um de seus objetivos futuros, a criação do controle de faltas.

### 2.2.2 Presente!

O *Presente!*<sup>2</sup> é um aplicativo móvel que auxilia os discentes no controle das faltas em cada disciplina matriculada. Com o aplicativo, os próprios usuários cadastram as matérias que estão cursando no período corrente e os horários das aulas. Após o cadastro de horários das disciplinas que estão sendo cursadas, o aplicativo faz o controle de presença, solicitando ao usuário que confirme se esteve presença na aula ou não. O aplicativo é gratuito e *open-source*.

Apesar de todos os pontos positivos mencionados, utilizar o *Presente!* para gerenciar faltas não é eficiente visto que ele depende do aluno para registrar suas faltas. Sendo assim, o controle do aluno pode contradizer com o controle dos docentes, já que existe a possibilidade do aluno esquecer de registrar presença no aplicativo e perder o controle no número total de faltas. Sendo assim, o *UFOP Faltas* permite que os docentes controlem e os discentes monitorem as faltas, diminuindo os riscos de lançamentos errados. Outro diferencial em relação ao *Presente!* é que o *UFOP Faltas* é um sistema Web totalmente voltado para o processo da *UFOP*, na qual o controle de assiduidade é totalmente gerenciado pelos docentes a partir do diário fornecido pela Universidade Federal de Ouro Preto (*UFOP*).

### 2.2.3 Plataforma minhaUFMG e Moodle UFOP

O *minhaUFMG* é um portal *online* para atender a todos os professores, técnicos e alunos da instituição, fornecendo um conjunto de ferramentas de trabalho colaborativo,

<sup>2</sup> <http://www.ufla.br/ascom/2016/04/26/volta-as-aulas-estudante-da-ufla-cria-aplicativo-para-auxiliar-no-controle-de-faltas/>

como: correio eletrônico, agenda corporativa, controle de tarefas, notas e faltas pelo professor. A plataforma *Moodle UFOP*, que também fornece as funcionalidades do *minhaUFMG*, não possui com o controle de frequência para os cursos presenciais como a *minhaUFMG*. Sendo assim, o diferencial entre as plataformas *minhaUFMG* e *Moodle UFOP* é que o *UFOP Faltas* irá atender um serviço não fornecido pela plataforma *Moodle UFOP*.

Similarmente ao Moodle UFOP, a aplicação proposta neste trabalho é uma plataforma *Web* desenvolvida especificamente para a *UFOP* e que tem como objetivo informatizar e aperfeiçoar o processo de assiduidade.

## 2.3 Considerações finais

Neste capítulo foram apresentadas as tecnologias utilizadas no projeto e os sistemas relacionados. Foram levantados os principais recursos utilizados pelo NTI, bem como descrito todo o arcabouço de desenvolvimento. Os sistemas correlatos foram importantes para o levantamento de requisitos e a definição do domínio da aplicação.

No Capítulo 4 serão descritos os requisitos da plataforma e a metodologia utilizada para implementar as funcionalidades propostas.

## 3 Modelagem do sistema

Neste capítulo é detalhada a fase inicial do desenvolvimento do projeto. Na Seção 3.1 é descrita a metodologia utilizada para o desenvolvimento do *software*, as atividades referentes ao levantamento de requisitos são apresentadas na Seção 3.2. Na Seção 3.3 os casos de uso, e por fim na Seção 3.4 a arquitetura do banco de dados são apresentadas.

### 3.1 BOPE

Segundo Pereira (2014), o BOPE é um processo híbrido das metodologias ágeis do Scrum, do XP, do RUP e de um conjunto de diretrizes do *Project Management Body of Knowledge* (PMBOK). O ciclo de vida do processo determina que uma série de regras sejam definidas, tais como: tarefas de planejamento, execução e testes. É um processo dirigido por artefatos, em que todas as tarefas realizadas geram um entregável, como: histórias de usuário, os cenários e casos de teste. Esses artefatos permitem a definição das tarefas e atividades a serem concluídas durante o desenvolvimento, os quais serão apresentados no Capítulo 4.

Como este trabalho foi desenvolvido seguindo os artefatos do BOPE, alguns deles foram gerados de modo a garantir a entrega e a qualidade do sistema. Pode-se destacar a criação de histórias de usuários que guiaram todo o desenvolvimento e ajudaram a entender o escopo do sistema como um todo, suas limitações, vantagens e desvantagens.

Além das histórias de usuário também foram elaborados cenários de teste. Desta maneira, cada funcionalidade pode ser testada e conseqüentemente ter seu funcionamento validado. As histórias de usuário criadas para certas partes do sistema, assim como os cenários de teste podem ser vistos completamente nos Apêndices A e B.

### 3.2 Análise de Requisitos

Diante da aplicação proposta foram discutidos e levantados requisitos essenciais para a aplicação e quais funcionalidades o sistema contemplaria. “Os requisitos de um sistema são descrições de funcionalidades e restrições das mesmas, refletindo as necessidades daqueles que o utilizarão” (SOMMERVILLE, 2010, p. 83, tradução nossa).

O levantamento de requisitos é fundamental para facilitar a identificação e definição de requisitos. Dessa maneira, foram realizadas as seguintes atividades: identificação dos envolvidos e desenvolvimento de casos de uso das principais funcionalidades do sistema.

Os requisitos do usuário, os requisitos de sistema e a especificação de projeto de

*software* foram levantados por meio de levantamento com docentes e discentes e podem ser definidos como se segue:

Os principais requisitos levantados neste trabalho foram:

1. Importar planilha da turma para fazer chamada;
2. Registrar falta diária ou por intervalo de dias;
3. Acompanhamento do número de faltas por parte dos discentes e docentes.
4. Gerar diário de classe semestral.

### 3.2.1 Usuários do Sistema

Sommerville (2010) apresenta que os *stakeholders* são aqueles que são beneficiados pelo sistema tanto direta quanto indiretamente. Os principais *stakeholders* do *UFOP Faltas* são: *discentes, docentes, chefes de departamento, responsáveis pelo desenvolvimento do sistema, responsáveis pela manutenção do sistema*. Neste caso, os chefes de departamento são responsáveis por assinar o diário de classe a ser encaminhado ao colegiado. Assim, foi determinado quais seriam os diferentes tipos de usuário do sistema. Os tipos definidos são: *Administrador do Sistema, docente e discente* e são descritos a seguir:

- **Administrador:** gerenciam novas disciplinas.
- **Docente:** gerenciam turmas e realizam cadastro de faltas.
- **Discentes:** acompanham o lançamento de faltas e as gerenciam.

### 3.2.2 Funcionalidades

As funcionalidades que o sistema provê foram definidas e serão descritas na subseções que se seguem. Elas foram agrupadas pelo tipo de usuário.

A Tabela 1 apresenta a função que pode ser executada pelo usuário do tipo *Administrador*:

Tabela 1 – Funcionalidades - Administrador

Funcionalidade	Descrição
Cadastrar Disciplinas	Obter arquivo de disciplinas na secretária de ensino da instituição e importar ao <i>UFOP Faltas</i>

Fonte: produzido pelo autor

Devido o sistema só ter cadastrado em sua base de dados as disciplinas dos cursos do ICEA, se utilizado para outros departamentos, é necessário que o usuário administrador faça o cadastro das disciplinas através do *upload* da planilha de disciplinas.

A Tabela 2 apresenta as funcionalidades do *docente*:

Tabela 2 – Funcionalidades - docente

Funcionalidade	Descrição
Cadastrar turma	Importar o arquivo CSV gerado a partir da <i>MinhaUFOP</i> , e submetido ao sistema <i>UFOP Faltas</i>
Gerenciar Faltas	Realizar a chamada e registrar faltas
Gerar Relatórios	Visualização do relatório final de faltas
Visualizar discentes	Busca e visualização sobre os discentes que estão matriculados na turma que o docente está vinculado
Visualizar Turmas	Visualizar turmas em que o docente está lecionando

Fonte: produzido pelo autor

A Tabela 3 apresenta as funções que podem ser executadas pelo usuário do tipo *discente*:

Tabela 3 – Funcionalidades - discente

Funcionalidade	Descrição
Visualizar Turmas	Lista das turmas que o discente está matriculado
Visualizar Faltas	Visualização de suas faltas para cada turma em que o discente está matriculado

Fonte: produzido pelo autor

### 3.3 Casos de Uso

Casos de uso são comumente utilizados a fim de fazer identificação requisitos de sistemas. Pressman (2010) apresenta duas versões dos casos de uso. Eles podem ser simples (pequenas histórias sobre como o usuário interage com o sistema) ou completos (semi-estruturado e com alto nível de detalhamento).

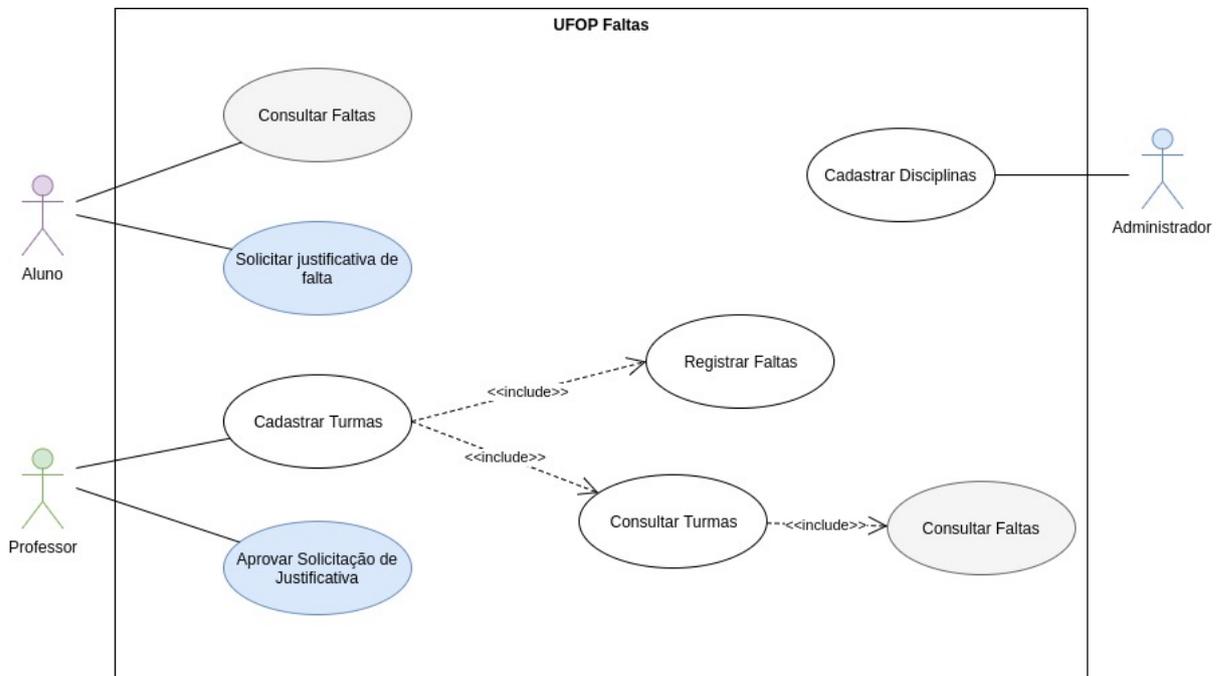
Foram identificados diferentes tipos de usuários, seus papéis e definidas as principais funcionalidades que eles poderiam executar dentro do sistema. Após a identificação foram criadas versões completas de casos de uso para cada grupo destas funcionalidade.

#### 3.3.1 Diagrama de Caso de Uso

Os casos de uso contribuíram para um melhor entendimento do fluxo de uso do sistema em geral. Além de contribuir na identificação de exceções e problemas que poderiam

ocorrer durante a utilização do *UFOP Faltas*. O diagrama de casos de uso é apresentado na Figura 1, na qual todas funcionalidades só podem ser operadas após autenticação no sistema.

Figura 1 – Diagrama de Casos de Uso



Fonte: produzido pelo autor

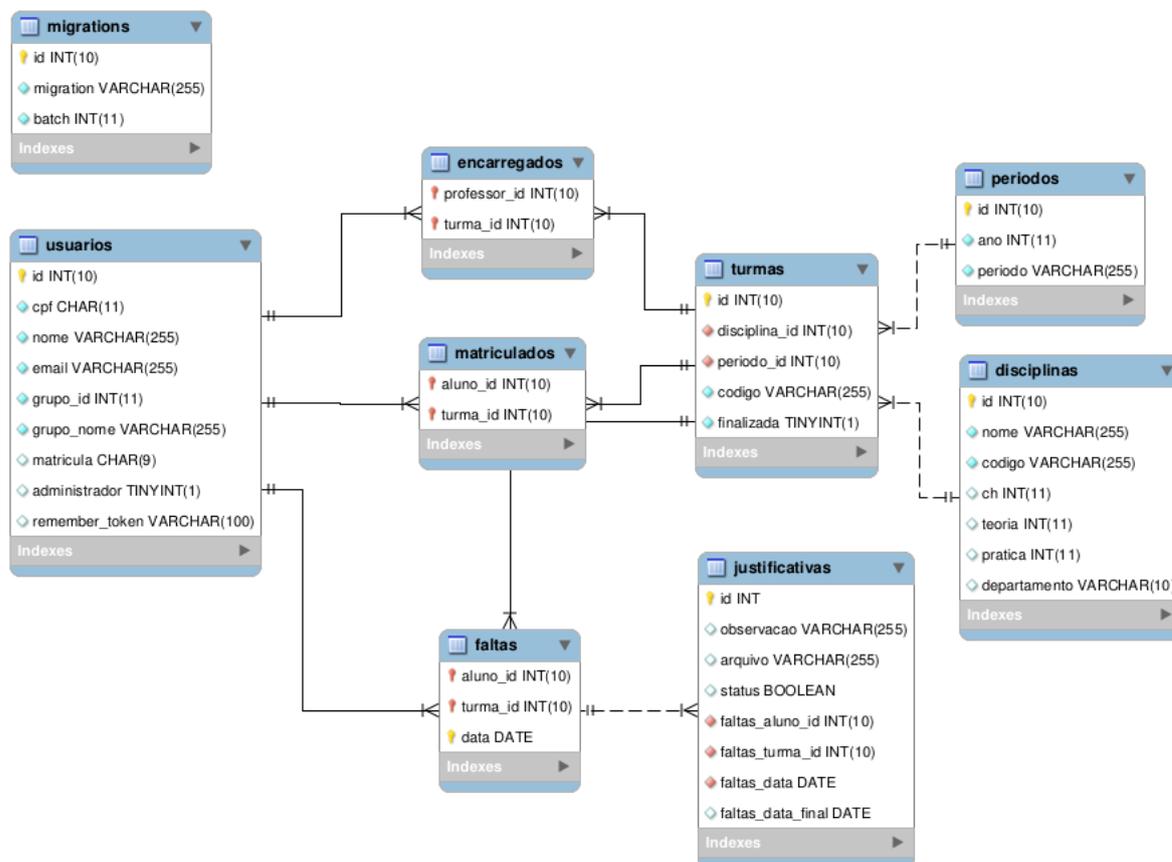
## 3.4 Banco de Dados

A modelagem do banco de dados é uma parte crucial para esse projeto, pois os dados enviados pelos usuários serão armazenados no sistema. Com todos os requisitos coletados e os casos de uso escritos, iniciou-se a modelagem, tendo essa uma abordagem clássica com banco de dados relacional.

### 3.4.1 Modelagem Conceitual

A técnica de modelagem de bancos de dados aplicada denomina-se *Modelo Entidade Relacionamento (MER)*. Segundo Heuser (2009) modelo de dados é uma descrição dos objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). A Figura 2 consiste em um diagrama de entidade relacional que ilustra os aspectos do sistema.

Figura 2 – Modelo Entidade Relacionamento - UFOP Faltas



Fonte: produzido pelo autor

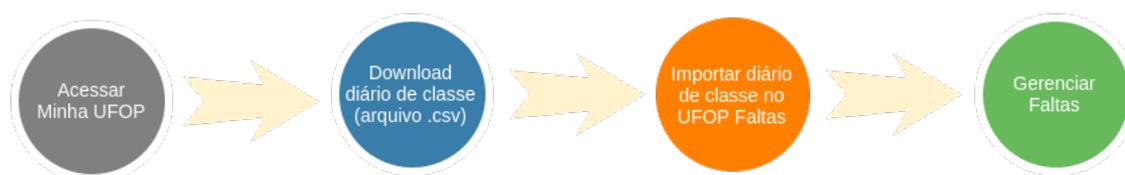
### 3.4.2 Importação de planilhas

No início de cada semestre os docentes acessam o portal *Minha UFOP* e fazem o *download* do diário de classe, e a partir dele iniciam o gerenciamento de faltas. O passo a passo para realizar a importação e iniciar a utilização do sistema Web está descrito a seguir e ilustrado na Figura 3.

1. Acesse o *Minha UFOP*;
2. Em Graduação, selecione a opção Diário de Classe;
3. Localize a turma que deseja gerenciar;
4. Clique em Imprimir ou Gerar o arquivo *Comma-Separated Values (CSV)*;
5. Em seguida acesse o *UFOP Faltas* com os mesmos dados de acesso a *Minha UFOP*;
6. Em “Minhas Turmas”, Criar (primeira opção);
7. Em “Escolha o arquivo (extensão .csv)” selecione o diário e clique em Importar;

8. Dessa forma é possível listar as turmas e Visualizar o diário;
9. Clique em Gerenciar Faltas;
10. Inicie a chamada e registre faltas para discentes ausentes;
11. Por fim, clique em “Aplicar”.

Figura 3 – Importação de Turmas - UFOP Faltas



Fonte: produzido pelo autor

### 3.5 Considerações finais

Nesse capítulo foram discutidas as metodologias de desenvolvimento da aplicação, a arquitetura e modelagem da base dados construída e a criação dos artefatos do projeto. O próximo capítulo apresentará o projeto, os testes do sistema Web com os usuários, no qual foram aplicados os questionários e realizadas as entrevistas a fim de validar a proposta e afirmar a importância da construção da plataforma.

## 4 Apresentação da aplicação desenvolvida

O presente trabalho demonstra a utilização do sistema. Nas seções a seguir serão descritos o contexto de utilização e as técnicas utilizadas no desenvolvimento dos módulos. Esta seção apresenta a autenticação para acessar todo o sistema a fim de se ter uma visão geral das funcionalidades do módulo de discente e docente. Em cada seção será apresentado brevemente o funcionamento do sistema bem como suas respectivas telas.

### 4.1 Autenticação

Cada usuário do sistema, pode efetuar seu login informando o CPF e senha da *Minha UFOP* como ilustrado na Figura 4. Conforme o tipo do usuário, ele será direcionado para a área do docente ou para a área do discente.

Figura 4 – Tela de *Login* - UFOP Faltas



A imagem mostra a interface de login do sistema UFOP Faltas. No topo, há um ícone de graduação e o texto "UFOP Faltas". Abaixo, há um formulário com dois campos de entrada: "CPF do Minha UFOP" (com ícone de usuário) e "Senha do Minha UFOP" (com ícone de cadeado). Abaixo dos campos, há uma caixa de seleção "Lembre-se de mim". Um botão vermelho com o texto "Entrar" e um ícone de seta para a direita está centralizado. Na base do formulário, há o texto "Use o mesmo CPF e a mesma senha do Minha UFOP" com um ícone de casa.

### 4.1.1 Área do docente

Após o *login*, o docente é redirecionado para a página Minhas Turmas, que exibe as informações sobre as turmas criadas, como apresentado na Figura 5.

Figura 5 – Tela Dados da Turma



Minhas Turmas Essa é a lista com todas as turmas as quais você está lecionando. 🏠 Início

Mostrar 10 registros por página Procurar:

Código da Disciplina	Disciplina	Turma	Ano	Período	Status	Ação
CSI030	PROGRAMACAO DE COMPUTADORES I	43	2017	1	Ativa	<a href="#">🔍 Detalhes</a> <a href="#">🔄 Atualizar</a> <a href="#">📅 Visualizar diário</a>
CSI030	PROGRAMACAO DE COMPUTADORES I	42	2017	1	Ativa	<a href="#">🔍 Detalhes</a> <a href="#">🔄 Atualizar</a> <a href="#">📅 Visualizar diário</a>
CSI477	SISTEMAS WEB I	11	2017	1	Ativa	<a href="#">🔍 Detalhes</a> <a href="#">🔄 Atualizar</a> <a href="#">📅 Visualizar diário</a>

Mostrando página 1 de 1 Anterior 1 Próximo

Fonte: produzido pelo autor

Quando não existe nenhuma turma criada, o docente deverá ir no menu **Minhas Turmas** e **Criar** turma importando a planilha em CSV, como mostrado na Figura 6.

Figura 6 – Tela de Importação de Turma



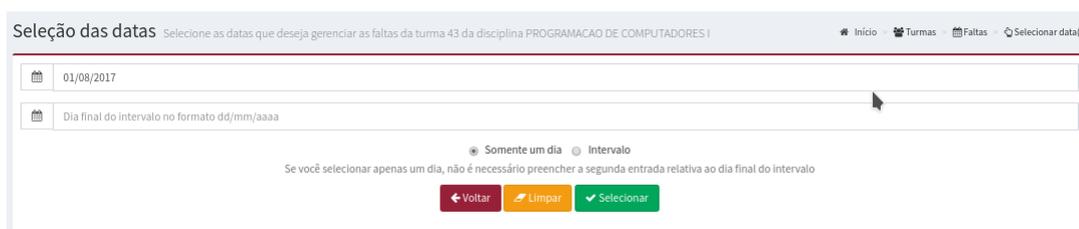
Criar Turma Escolha o arquivo CSV que contenha os dados da turma. 🏠 Início

Selecione o diário da turma que deseja criar  No file chosen

Fonte: produzido pelo autor

A partir da lista de turmas criadas anteriormente, o docente possui ações para cada turma, no caso de realizar a chamada, o docente irá em **Visualizar Diário** -> **Gerenciar Faltas** e indicará a data ou o intervalo que fará os lançamentos, apresentado na Figura 7.

Figura 7 – Tela de Seleção Data



Seleção das datas Selecione as datas que deseja gerenciar as faltas da turma 43 da disciplina PROGRAMACAO DE COMPUTADORES I 🏠 Início 📅 Turmas 📅 Faltas 🔍 Selecionar data(s)

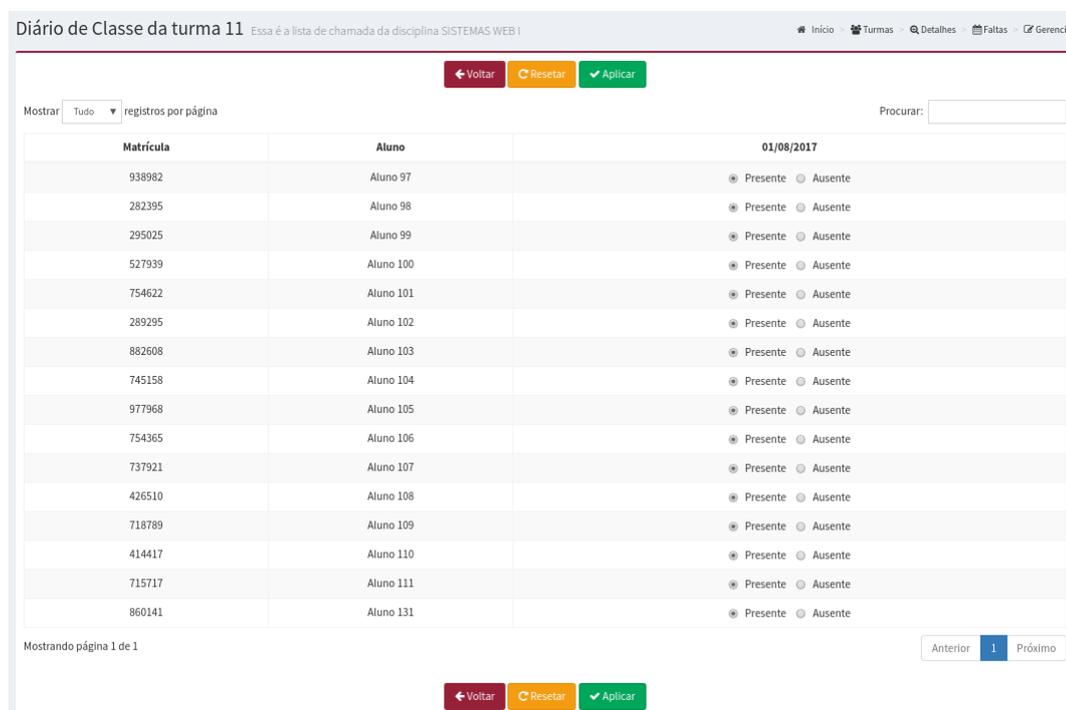
Somente um dia  Intervalo

Se você selecionar apenas um dia, não é necessário preencher a segunda entrada relativa ao dia final do intervalo

Fonte: produzido pelo autor

Seguida da seleção o docente fará a chamada verbal e lançará falta para discentes ausentes, como ilustrado na Figura 8.

Figura 8 – Tela de Lançamento de Faltas



The screenshot displays the 'Diário de Classe da turma 11' interface. At the top, there are navigation links: 'Início', 'Turmas', 'Detalhes', 'Faltas', and 'Gerenciar'. Below the header, there are three buttons: 'Voltar' (red), 'Resetar' (orange), and 'Aplicar' (green). A search bar labeled 'Procurar:' is on the right. The main content is a table with columns for 'Matrícula', 'Aluno', and '01/08/2017'. The table lists 15 students, each with a radio button for 'Presente' and 'Ausente'. At the bottom, there are pagination controls: 'Mostrando página 1 de 1', 'Anterior', '1', and 'Próximo', along with the same three action buttons.

Matrícula	Aluno	01/08/2017
938982	Aluno 97	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
282395	Aluno 98	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
295025	Aluno 99	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
527939	Aluno 100	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
754622	Aluno 101	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
289295	Aluno 102	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
882608	Aluno 103	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
745158	Aluno 104	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
977968	Aluno 105	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
754365	Aluno 106	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
737921	Aluno 107	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
426510	Aluno 108	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
718789	Aluno 109	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
414417	Aluno 110	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
715717	Aluno 111	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente
860141	Aluno 131	<input checked="" type="radio"/> Presente <input type="radio"/> Ausente

Fonte: produzido pelo autor

Ao fim do semestre o docente pode gerar o relatório contendo a assiduidade da turma. Um exemplo de relatório gerado com dados de teste está no Apêndice E.

#### 4.1.2 Área do discente

Após a autenticação do discente, em **Minhas Turmas** é exibida as Turmas e o discente poderá visualizar sua assiduidade, conforme apresentado na Figura 4.

Figura 9 – Diário de Classe Semestral

UFOP Faltas da turma 43 Essa é a lista de chamada da disciplina PROGRAMACAO DE COMPUTADORES I

Início Turmas Detalhes Faltas

Mostrar 10 registros por página Procurar:

Matrícula	Aluno	31/07/2017	09/08/2017	10/08/2017	11/08/2017	16/08/2017	17/08/2017	18/08/2017	24/08/2017	Número de faltas
954563	Aluno 3	✓ Presente	✗ Ausente	✓ Presente	✗ Ausente	✓ Presente	✓ Presente	✓ Presente	✓ Presente	2
697648	Aluno 4	✓ Presente	✓ Presente	✓ Presente	✗ Ausente	✓ Presente	✓ Presente	✓ Presente	✓ Presente	1
524551	Aluno 5	✓ Presente	✗ Ausente	1						
429811	Aluno 6	✓ Presente	✗ Ausente	✓ Presente	✓ Presente	1				
475402	Aluno 7	✓ Presente	✗ Ausente	✗ Ausente	✗ Ausente	✗ Ausente	✓ Presente	✓ Presente	✓ Presente	4
987580	Aluno 8	✓ Presente	✗ Ausente	✗ Ausente	✓ Presente	✓ Presente	✓ Presente	✓ Presente	✗ Ausente	3
711694	Aluno 9	✓ Presente	✗ Ausente	✓ Presente	1					
495729	Aluno 10	✓ Presente	✗ Ausente	✓ Presente	1					
243566	Aluno 11	✓ Presente	✗ Ausente	✓ Presente	✓ Presente	1				
530642	Aluno 12	✓ Presente	0							

Mostrando página 1 de 5

Anterior 1 2 3 4 5 Próximo

Relatório Voltar Gerenciar faltas

Fonte: produzido pelo autor

Nessa seção foram apresentadas as telas da aplicação e seu funcionamento. Na seção seguinte 4.2 serão descritos os testes realizados na plataforma e a metodologia utilizada para implementar e avaliar o resultado obtido nos testes.

## 4.2 Teste de *software*

Teste de *software* é o processo de execução de um produto para averiguar se ele atingiu suas especificações e funciona corretamente em seu ambiente alvo (NETO, 2007). De acordo com Luo (2001), um bom teste é o que possui uma alta probabilidade de encontrar um erro ainda não descoberto. Um teste bem sucedido é o que de fato descobre erros desconhecidos.

Desenvolvimento e testes sempre existiram pelo mesmo propósito: entregar *software* de qualidade. Sendo assim, os testes são indicados para a garantia de qualidade, visto que são realizados do início ao final do processo de desenvolvimento. O teste de *software* cumpre um papel fundamental, pois pode contribuir para melhorar não só a qualidade, mas também para diminuir os custos, evitar o retrabalho, aumentar a segurança e a confiabilidade, antecipar a descoberta de falhas e incompatibilidades.

Para avaliar a eficácia e consistência da solução implementada, foram propostos estórias de usuários e cenários de testes, que abrangessem os usos típicos do sistema. Foram utilizadas rodadas de testes incrementais até que se conseguisse obter resultados satisfatórios, isto é, até que não fossem encontrados mais erros. Os testes de unidade, de

integração e de sistema serão apresentados nas seções 4.2.1, 4.2.2 e 4.3 respectivamente.

### 4.2.1 Testes Unitários

Teste unitário possui como objetivo verificar a existência de defeitos em cada módulo do projeto, tendo como alvo os métodos desenvolvidos ou pequenos trechos específicos de código (NETO, 2007).

Esse tipo de teste é realizado durante o desenvolvimento pelo próprio desenvolvedor, pois testa a unidade básica de *software*, por sua vez chamado de unidade Luo (2001).

Segundo Maldonado et al. (2007), um dos objetivos do teste unitário é identificar erros de lógica e de implementação. Logo, alguns erros na lógica foram encontrados e corrigidos:

1. Durante a importação do arquivo *Comma-Separated Values (CSV)* de turmas, quando o delimitador entre os campos não fosse ponto e vírgula, era reportado erro de leitura no arquivo.
2. Nomes de alunos iniciados por espaço em branco alterava a ordem alfabética na listagem de alunos matriculados.
3. Ao acessar urls sem autenticação, erros do Laravel são reportados.

No Apêndice A consta a relação das estórias de usuário criadas e os cenários de testes no Apêndice B.

### 4.2.2 Testes de Integração

Teste de integração tem como objetivo averiguar a existência de falhas relacionadas a interface do *software* entre seus diferentes módulos quando estes são integrados (NETO, 2007). É realizado quando uma estrutura maior é formada (devido a integração de dois ou mais módulos), sendo que os módulos possuem suas especificações individuais testadas, porém olhando-se para o conjunto (LUO, 2001). A medida que essas estruturas vão sendo testadas, a estrutura de programa que foi determinada pelo projeto vai sendo construída (MALDONADO et al., 2007).

Após a produção do sistema ter atingido o seu estágio final, foi agendada uma reunião com um grupo de professores e alunos para apresentação do projeto desenvolvido e obtenção de *feedbacks* e sugestões, foi feita hospedagem do sistema em um servidor online a fim de serem executados os testes em ambiente de produção. Foi possível criar turmas a partir das planilhas de um grupo de cinco professores e seis alunos acessaram o sistema para consultar as faltas.

A Tabela 4 lista todas as exceções de *software* lançadas durante o teste, foi formulada com as características que os usuários tiveram algum problema durante os testes. Nesta é apresentada uma breve descrição do problema, a solução para o problema em nível de implementação de *software* e o status atual da implementação do *software*.

Tabela 4 – Problemas reportados pelos usuários

Descrição	Solução	Status
Erro no serviço de autenticação do usuário ao tentar acessar o portal sem cadastro no MinhaUFOP	Exibir mensagem: "O Usuário não está autorizado, as credenciais são inválidas"	Corrigido
Lista de turmas exibida para o discente sem filtro	Exibir somente a lista de turmas ao qual o discente está matriculado	Corrigido
Ao registrar uma falta é necessário marcar os presentes e ausentes	Ao registrar é necessário marcar somente os ausentes	Corrigido
Não é possível filtrar as faltas por data no diário	Filtro de data adicionado	Corrigido
Cursor do CPF vai para o meio do campo ao digitar (Mobile)	Utilizar o <i>plugin</i> de máscara do JavaScript	Não adicionado nessa versão

O teste de integração obteve uma coleta satisfatória de resultados que possibilitaram correções no *software*.

### 4.2.3 Testes de sistema

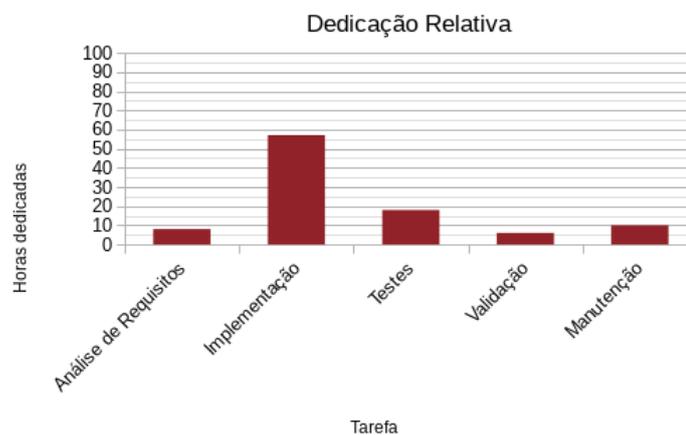
O teste de desenvolvimento é o tipo de teste realizado durante a fase de produção do sistema pelos desenvolvedores com o objetivo de identificar *bugs*, erros e não-conformidades. *Bug* pode ser entendido como uma ação inconsistente, ocasionada por uma tentativa de entender uma determinada informação, erro é a diferença entre o valor obtido e o valor esperado e falha é o comportamento do *software* diferente do esperado pelo usuário (SOMMERVILLE, 2010, p. 210).

Nessa fase, o programa é testado como um todo, diferente do teste de integração que testas os componentes individuais. É um teste bem mais completo, uma vez que inclui todos os componentes do projeto simulando o usuário final no ambiente.

De acordo com Filho (2003), os testes de regressão verificam novamente esses segmentos, para checar se eles continuam funcionando corretamente. Sendo assim, sempre que uma nova funcionalidade foi desenvolvida, independentemente da complexidade da mesma, a funcionalidade foi testada individualmente no ambiente de desenvolvimento.

O índice de *bugs* foi analisado a partir das horas dedicadas para manutenção, e de acordo com a métrica indicada, o índice crítico foi baixo como apresentado na Figura 10, onde apenas 10% da dedicação foi destinada a correção de erros. Após atestado que ela teve o comportamento esperado, foram executados testes de regressão a fim de garantir que não surgiram novos defeitos em componentes já analisados. Foi possível identificar que o sistema esta de acordo com as especificações determinadas e que funciona corretamente no qual foi projetado.

Figura 10 – Dedicção relativa em horas de reparo



Fonte: produzido pelo autor

A métrica de cobertura dos testes utilizada baseiam-se na cobertura dos requisitos de *software* e do código-fonte. O teste de cobertura tem a finalidade de verificar se o roteiro de teste executado, tanto nos testes de unidade quanto nos testes de integração, estão abrangendo a completude total do código implementado. De acordo com os requisitos levantado na Seção 3.2, os testes aplicados cobriram 100% dos requisitos. Os requisitos foram verificados para determinar consistência e completude a partir da análise manual sistemática e dos cenários de teste.

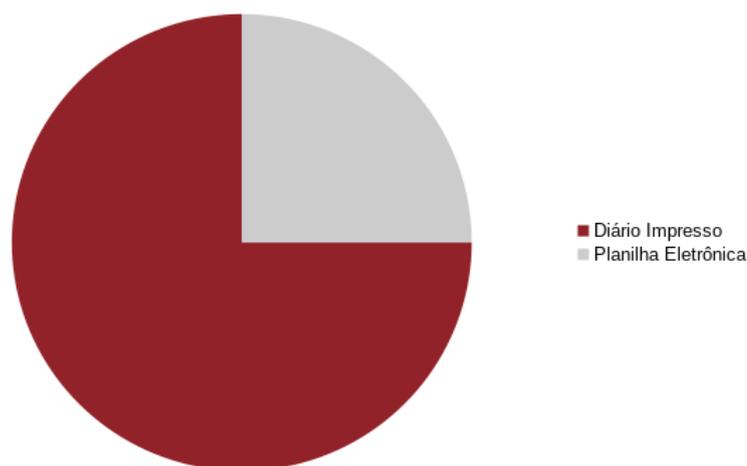
### 4.3 Entrevista com os usuários

Para justificar a aplicação proposta, foi definido que a métrica de validação seria em forma de questionários respondidos de acordo com a ferramenta analisada e a atual metodologia utilizada pelo docente no gerenciamento de assiduidade. Para cada resposta, teve-se a necessidade de avaliar e validar a proposta do *UFOP Faltas*.

Primeiramente, o projeto foi apresentado para cada professor, suas motivações e como ele poderia beneficiar a comunidade acadêmica, informatizando e padronizando o processo. A pesquisa foi feita envolvendo 16 docentes do Instituto de Ciências Exatas e Aplicadas (ICEA), em que cada um desses docentes respondeu ao questionário de pesquisa presente no Apêndice C. Nesta fase os discentes e docentes testaram o *UFOP Faltas*

De acordo com o resultado da Figura 11, 75% dos entrevistados utilizam diário impresso e os demais, planilhas eletrônicas em formato proprietário. Diante da entrevista pode-se observar que a maioria dos docentes preferem utilizar o diário impresso por não ter outra metodologia mais adequada e que ainda utilizam o pelo fato de ser necessária a entrega deste ao fim do semestre para ao departamento do docente. Sendo assim, o *UFOP Faltas* gera o relatório apontando os registros de assiduidade e pode facilmente substituir o diário impresso.

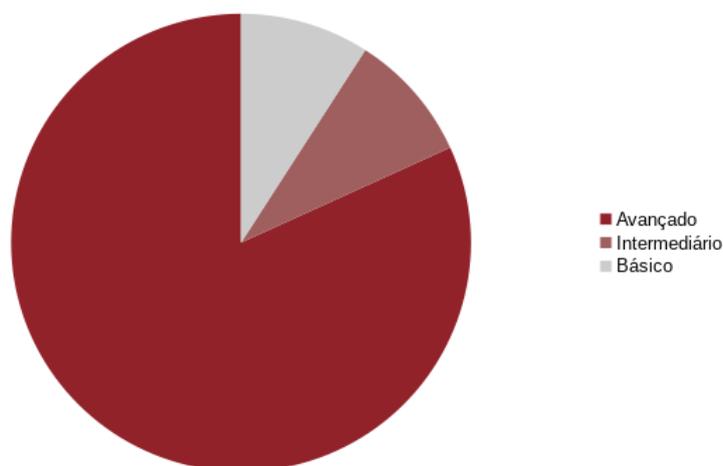
Figura 11 – Qual metodologia que você utiliza para registrar faltas?



Fonte: produzido pelo autor

Vale ressaltar a importância do conhecimento prévio para trabalhar no gerenciamento de um sistema Web. Por isso, a Figura 12 mostra que a maioria tem conhecimento para utilizar o sistema e que, a minoria que tem conhecimento básico. Portanto, durante a entrevista o sistema foi apresentado e considerado simples e fácil de utilizar.

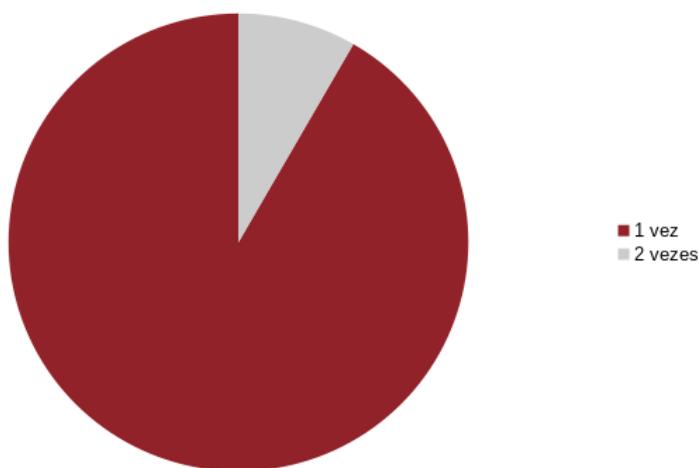
Figura 12 – Qual o nível de conhecimento e relacionamento com sistemas web?



Fonte: produzido pelo autor

Portanto, na figura 13 a questão se faz necessária dado que existem docentes que fazem as chamadas por aula, o que representa 92%, sendo duas por dia e não uma valendo para as duas aulas seguidas.

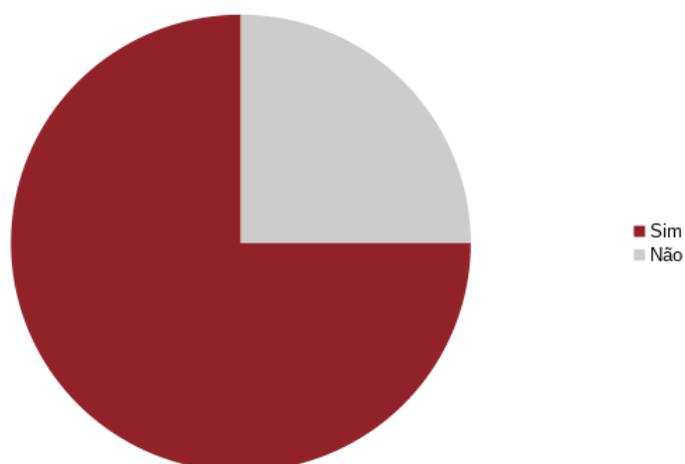
Figura 13 – Sua metodologia considera a presença por aula ou por dia?



Fonte: produzido pelo autor

No caso da inserção de uma nova metodologia, se faz necessário considerar o interesse dos possíveis usuários. Como mostra a Figura 14, 75% dos entrevistados substituiriam sua metodologia de controle pelo sistema Web proposto nesse trabalho, o que reforça a validação da proposta.

Figura 14 – Você substituiria sua metodologia por um sistema web que faz o que sua metodologia faz, inclusive geração de relatório?



Fonte: produzido pelo autor

#### 4.4 Considerações finais

Com o objetivo de validar a proposta e identificar falhas, os testes comprovaram a eficiência da aplicação, já que nesses foram encontrados erros, e posteriormente corrigidos. A partir disso, pode-se concluir que a metodologia de gerenciamento de assiduidade proposta tem condições de substituir a atual metodologia, além de ter uma ótima aceitação dos usuários que participaram dos testes. No capítulo que se segue serão apresentadas as considerações e conclusões do projeto.

## 5 Conclusão

Este trabalho apresentou a fundamentação teórica para o desenvolvimento do projeto, as tecnologias e os sistemas relacionados. Foram descritos os requisitos e a modelagem do sistema, além da apresentação da aplicação, funcionalidades e metodologias utilizadas. Por fim apresentados os resultados e testes a fim de garantir a validação do sistema *web*. A seguir são apresentadas as considerações finais e propostas de trabalhos futuros.

### 5.1 Considerações Finais

Com o sistema desenvolvido por este trabalho, espera-se que o mesmo quando implantado na [UFOP](#) consiga aperfeiçoar a maneira como é feito o gerenciamento de assiduidade.

O sistema proposto conseguiu cumprir com todos os objetivos e requisitos inicialmente especificados para o projeto. Embora algumas das suas funcionalidades ainda necessitem de melhorias, a ferramenta se mostrou eficiente, satisfatória e visivelmente agradável segundo dados coletados na entrevista com os usuários, o que se mostra um grande ponto positivo da ferramenta.

O software foi desenvolvido utilizando-se tecnologias e padrões de projeto para desenvolvimento Web visando facilitar a sua manutenção e a sua evolução. As tecnologias baseadas na Web permitira aos usuários a oportunidade de acompanhar suas faltas a partir de qualquer navegador Web.

No decorrer de todo o projeto foram realizadas reuniões semanais com os orientadores deste trabalho a fim de discutir sobre o andamento do mesmo, problemas enfrentados e definir tarefas a serem executadas e a prioridade de cada uma delas. Deste modo, foi possível garantir que o projeto evoluísse constantemente em todas as etapas. Durante as reuniões, novos requisitos foram definidos e alguns dos existentes adaptados. A utilização do *framework* MVC *Laravel* e da metodologia BOPE foram fatores primordiais que contribuíram para a diminuição da complexidade de adaptação do sistema aos novos requisitos e qualidade da ferramenta.

### 5.2 Trabalhos Futuros

Durante o desenvolvimento deste projeto, considerando algumas limitações e definições do escopo do trabalho, algumas propostas de continuidade foram identificadas. Os

testes identificaram os problemas adequadamente e foram implementadas as correções dos erros encontrados.

Novas funcionalidades irão complementar o sistema, como possíveis trabalhos futuros, pode-se apontar:

1. Módulo de lançamento e divulgação de notas
2. Comportar um maior número de departamentos, ou seja, importar dados de disciplinas ofertadas em outros departamentos da UFOP
3. Construção de um aplicação móvel que armazena os dados local e quando tiver acesso a rede *Wifi*, envie os dados para o servidor, possibilitando utilizar o sistema quando o docente não tiver acesso a internet e for realizar a chamada.

# Referências

- ENGINES, D. *DB Engines Ranking*. [S.l.]: Obtido de DB Engines: <http://dbengines.com/en/ranking>, 2017. Citado na página 19.
- FILHO, W. de P. P. *Engenharia de software*. [S.l.]: LTC, 2003. v. 2. Citado na página 35.
- HEUSER, C. A. *Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS*. [S.l.]: Bookman Editora, 2009. Citado na página 26.
- LUO, L. Software testing techniques. *Institute for software research international Carnegie mellon university Pittsburgh, PA*, v. 15232, n. 1-19, p. 19, 2001. Citado 2 vezes nas páginas 32 e 33.
- MALDONADO, J. C. et al. Introdução ao teste de software. *Rio de Janeiro: Campus*, p. 52, 2007. Citado na página 33.
- MYSQL, A. Mysql: the world's most popular open source database. <http://www.mysql.com/>, 2017. Citado na página 19.
- NETO, A. C. D. Introdução a teste de software. *Engenharia de Software Magazine*, v. 1, p. 22, 2007. Citado 2 vezes nas páginas 32 e 33.
- OTWELL, T. Laravel-the php framework for web artisans. *Laravel. com. Last accessed*, v. 30, 2017. Citado na página 19.
- PEREIRA, I. M. Desenvolvendo software inovador em universidades públicas: adaptando processos ágeis para a realidade de laboratórios de pesquisa e desenvolvimento. 2014. Citado na página 23.
- PEREIRA, M. H. Segurança de redes sem fio, uma proposta com serviços integrados de autenticação ldap e radius. *Pontifícia Universidade Católica da Paraná*, p. 9, 2009. Citado na página 20.
- PRESSMAN, R. S. Software engineering sevent edition. *America, McGraw Hill*, 2010. Citado na página 25.
- SANTOS, E. Formei ufop - um sistema de controle do progresso academico. 2016. Citado na página 21.
- SOMMERVILLE, I. *Software Engineering*. 9. ed. Harlow, England: Addison-Wesley, 2010. ISBN 978-0-13-703515-1. Citado 3 vezes nas páginas 23, 24 e 34.

# Apêndices

# APÊNDICE A – Estórias de Usuário

## A.1 Critérios de aceitação sobre os requisitos do projeto UFOP Faltas

### A.1.1 Serviço de Autenticação

COMO um usuário cadastrado no Minha UFOP

EU GOSTARIA DE logar no sistema mediante fornecimento do "cpf" e "senha"

PARA CONSEGUIR acesso às funcionalidades do sistema

### A.1.2 Serviço de Visualização de Informações

COMO um docente cadastrado

EU GOSTARIA DE criar turmas

PARA CONSEGUIR lançar faltas

COMO um docente cadastrado

EU GOSTARIA DE visualizar minhas turmas

PARA CONSEGUIR verificar a corretude dos dados

COMO um docente cadastrado

EU GOSTARIA DE visualizar o diário de uma turma

PARA CONSEGUIR realizar o lançamento de faltas

COMO um docente cadastrado

EU GOSTARIA DE visualizar minhas turmas

PARA CONSEGUIR atualizar os dados após ajuste de matrícula

COMO um docente cadastrado

EU GOSTARIA DE visualizar minhas turmas

PARA CONSEGUIR verificar a existência de alunos que atingiram número máximo

de faltas

COMO um discente cadastrado

EU GOSTARIA DE visualizar minhas turmas

PARA CONSEGUIR verificar minha assiduidade

## APÊNDICE B – Cenários de teste

### B.0.1 Serviço de Autenticação

DADO que o usuário forneça cpf ou senha inválida

QUANDO o botão "Entrar"for pressionado

ENTÃO (a) a aplicação deve exibir mensagem que informe que "O Usuário não está autorizado, as credenciais são inválidas"

### B.0.2 Serviço de Visualização de Informações

DADO que o docente crie a turma

QUANDO o botão "Importar"for pressionado

ENTÃO (a) a aplicação deve exibir mensagem que a importação está sendo realizada (b) disciplina da turma não existe na base de dados (c) Nenhum arquivo foi selecionado

DADO que o docente registre uma falta

QUANDO o botão "Aplicar"for pressionado

ENTÃO (a) a aplicação deve exibir mensagem que a falta foi registrada com sucesso

DADO que o docente atualize uma turma

QUANDO o botão "Importar"for pressionado

ENTÃO (a) a aplicação deve exibir mensagem que a turma foi atualizada

DADO que o docente registre uma falta

QUANDO o botão "Gerenciar Faltas"for pressionado

ENTÃO (a) a aplicação deve solicitar a data ou o intervalo de dias para realizar a chamada

DADO que o discente tenha acesso o sistema

QUANDO o botão "Minhas Turmas"for pressionado

ENTÃO (a) a lista de turmas que o aluno está matriculado, deve ser carregada

# APÊNDICE C – Questionário de Validação do Sistema

**1. Qual metodologia o professor utiliza para registrar faltas?**

- Diário Impresso
- Planilha Excel
- Lista de Presença - Coleta de assinatura
- Outro

**2. Sua metodologia considera a presença por aula ou por dia?**

- Por aula
- Por dia

**3. Você substituiria sua metodologia por um sistema Web que faz o que sua metodologia faz, inclusive geração de relatório:**

- Sim
- Não
- Talvez

**4. Qual o nível de conhecimento e relacionamento com sistemas web?**

- Avançado
- Intermediário
- Básico

# APÊNDICE D – Tutorial

Este tutorial fornece o passo a passo para a utilização do Diário. É relevante a leitura das instruções para que possa rapidamente familiarizar-se com a forma de manipulação dos recursos disponíveis. Primeiramente, a turma deverá ser criada para que os *Alunos* possam acessar a turma e conforme o *Professor* lançar faltas, o aluno acompanhá-las, conforme exemplificado pela Figura 15:

Figura 15 – Tela - Tutorial de Utilização do UFOP Faltas

**Tutorial** Início

Este tutorial fornece o passo a passo para a utilização do Diário. É relevante que você o leia e veja as instruções para que possa rapidamente familiarizar-se com a forma de manipulação dos recursos disponíveis.

**Acesso Professor - Recursos permitidos somente para docentes**

- + Criar Turma** Minhas Turmas > Criar Turma  
A partir da fase de ajuste de matrícula, os professores tem acesso ao diário das turmas que ministrarão. Portanto, com o diário de classe de extensão (.csv), o professor escolherá o arquivo no seu dispositivo e importará para a plataforma através do botão Importar. Assim, a turma estará criada e pronta para o gerenciamento de faltas.
- ☰ Listar Turmas** Minhas Turmas > Listar Turmas  
Cadastrada as turmas, o professor visualizará todas as turmas as quais ele é responsável. Podendo obter detalhes dos dados de cada uma delas ou visualizar o diário.
- 👤 Gerenciar Turmas** Minhas Turmas > Listar Turmas > Detalhes  
O gerenciamento das turmas será feito a partir da seleção de determinada turma, assim será apresentado os alunos matriculados e seus respectivos dados, permitindo que o professor desmatricule um aluno ou finalize a turma, caso as aulas do período se encerrem. Nesta tela será possível carregar um diário atualizado após diferentes ajustes de matrícula, e assim automaticamente alunos serão adicionados ou excluídos de determinada turma.
- 🕒 Gerenciar Faltas** Minhas Turmas > Listar Turmas > Visualizar Diário > Gerenciar Faltas  
O gerenciamento de faltas conta com o registro das faltas, podendo cadastrar faltas em um período de tempo ao mesmo tempo, ou cadastrando somente uma chamada. Por exemplo, registrar faltas de uma semana inteira, ou somente de uma aula.
- 📅 Visualizar Diário** Minhas Turmas > Listar Turmas > Visualizar Diário  
Na apresentação do diário, o professor o acessa conforme a turma escolhida. De modo que liste o diário de todos as aulas em que a chamada foi realizada. Em que cada linha representa a presença ou falta do aluno em determinado dia.

**Acesso Aluno - Recursos permitidos para discentes**

- ☰ Listar Turmas** Minhas Turmas > Listar Turmas  
Nesta página são listadas todas as turmas que o aluno logado está matriculado. A partir da criação das turmas, automaticamente o aluno é vinculado a turma e assim o aluno só precisa listar as turmas e consultar suas faltas.
- 📅 Visualizar Faltas** shared a video  
Para cada turma em que o aluno está matriculado é possível visualizar as faltas registradas para ele.

Fonte: produzido pelo autor

APÊNDICE E – Diário de Classe  
UFOP FALTAS

DIÁRIO DE CLASSE 2017/1

<b>MATRÍCULA</b>	<b>NOME ALUNO(A)</b>	<b>TOTAL</b>
11.1.8300	Aluno 97	2
10.2.8051	Aluno 98	0
12.1.8358	Aluno 99	1
13.1.8461	Aluno 100	0
14.2.8449	Aluno 101	0
13.2.8398	Aluno 102	1
13.1.8145	Aluno 103	0
13.2.8149	Aluno 104	1
14.2.8519	Aluno 105	1
12.2.8244	Aluno 106	0
13.1.8143	Aluno 107	0
14.1.8303	Aluno 108	0
14.1.8151	Aluno 109	0
14.2.8564	Aluno 110	0
12.1.8390	Aluno 111	0
14.1.8229	Aluno 131	0

PROFESSOR Aluno 2

DATA	ASS. PROFESSOR	DATA	ASS. CHEFE DEPTO	
------	----------------	------	------------------	--

LISTAGEM SUJEITA A ALTERAÇÕES DEVIDO A REQUERIMENTOS EM ANDAMENTO.  
INSTRUÇÕES PARA PREENCHIMENTO: A CHAMADA ORAL É OBRIGATÓRIA. IDENTIFIQUE O TIPO DE AULA COM T=TEÓRICA E P=PRÁTICA. CADA AULA CORRESPONDE A UMA MARCAÇÃO. A PRÓ-REITORIA DE GRADUAÇÃO SOMENTE RECONHECE COMO ALUNO MATRICULADO AQUELE CUJO NOME CONSTA NESTE DIÁRIO.

# APÊNDICE F – Configuração do ambiente

## F.1 Pré-requisitos

Antes de executar os comandos de instalação, é necessário que você tenha instalado no dispositivo no qual o sistema irá ser hospedado o Node.js entre as versões 0.12 e 6.9.4 e o gulp.js instalado globalmente. Você pode ver mais detalhes sobre a instalação do Node.js visitando o site da aplicação para ter mais detalhes sobre os procedimentos em cada plataforma.

No caso do gulp.js, você deve instalar previamente o Node.js e então executar o comando **npm install –global gulp-cli**. Caso você esteja usando um sistema UNIX como MacOS ou uma distribuição Linux, talvez seja necessário executar o comando com permissão de administrador e para isso você pode usar o comando `sudo` ou `su`, variando de acordo com o seu sistema operacional.

### F.1.1 Pré-requisitos do Laravel, temos:

- Versão do PHP igual ou superior a 5.6.4;
- Extensão OpenSSL do PHP ativada;
- Extensão PDO do banco de dados utilizado ativada;
- Extensão Mbstring do PHP ativada;
- Extensão Tokenizer do PHP ativada;
- Extensão XML do PHP ativada;

## F.2 Instalação

Para a instalação são necessários os seguintes comandos:

```
composer install
```

```
npm install
```

```
gulp –prod
```

```
php artisan migrate
```

O comando **composer install** é usado para realizar a instalação de todos os pacotes os quais o sistema é dependente para que o mesmo funcione.

O comando **npm install** é usado para instalar as bibliotecas necessárias para manipulação de assets ou recursos do sistema como arquivos CSS, JavaScript, etc. Caso você esteja usando um sistema Windows para hospedar a aplicação, talvez seja necessário executar o comando **npm install --no-bin-links** ao invés do **npm install** devido algumas restrições do sistema operacional.

O comando **gulp --prod** irá realizar todos os comandos definidos no arquivo `gulpfile.js`, onde estão definidos todos os procedimentos referentes aos arquivos de recursos do sistema. No caso desse boilerplate todos os arquivos CSS e JavaScript serão mesclados e minificados em um único arquivo, pois isso é uma boa prática para o desenvolvimento de sistemas WEB.

E por último, o comando **php artisan migrate** faz a migração da tabela que irá conter os usuários para o banco de dados.

### F.2.1 Configuração

Dentro da pasta `config` existe o arquivo `ldapi.php` que mostra que no arquivo de ambiente, o chamado arquivo `.env` devem existir as entradas:

LDAPAPI USER: usuário da API;

LDAPAPI PASSWORD: senha do usuário da API;

LDAPAPI REQUEST METHOD: Método HTTP da requisição de autenticação. Originalmente a API foi desenvolvida para usar somente método POST mas isso pode ser alterado facilmente, então é necessário verificar o método com o responsável pela API;

LDAPAPI AUTH URL: URL para qual a requisição de autenticação será enviada. Verifique também com o responsável pela API qual é o endereço que recebe as requisições de autenticação.

No arquivo `.env.example` é possível ver um exemplo preenchido. Para obter o seu usuário e senha da LDAPAPI é necessário entrar em contato com o suporte de informática do ICEA.

Após preenchido o usuário e senha do LDAPAPI no arquivo `.env` pode ser necessário executar o comando **php artisan config:cache** na raiz do diretório do sistema. Esse comando limpa a configuração anterior que foi salva em cache e força um novo carregamento do arquivo de configuração.