

**Universidade Federal dos Vales do Jequitinhonha e Mucuri**

**Faculdade de Ciências Exatas e Tecnológicas**

**Departamento de Computação**

**Curso de Sistemas de Informação**

**Desenvolvimento de um sistema especialista para  
o gerenciamento de correção de redações**

**Cristian Fernandes Andrade**

Diamantina

2019

**Universidade Federal dos Vales do Jequitinhonha e Mucuri**

**Faculdade de Ciências Exatas e Tecnológicas**

**Departamento de Computação**

**Curso de Sistemas de Informação**

**Desenvolvimento de um sistema especialista para  
o gerenciamento de correção de redações**

**Cristian Fernandes Andrade**

Trabalho de conclusão de curso de graduação  
apresentado à Universidade Federal dos Vales do  
Jequitinhonha e Mucuri como requisito parcial  
para obtenção do título de Bacharel em Sistemas  
de Informação.

Orientador: Prof. Me. Marcelo Ferreira Rego

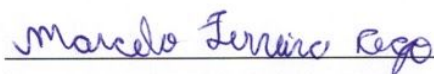
Diamantina

2019



Monografia de projeto final de graduação sob o título “Desenvolvimento de um sistema especialista para o gerenciamento de correção de redações”, defendida por Cristian Fernandes Andrade e aprovada em 18 de julho de 2019, em Diamantina, Minas Gerais.

**Banca Examinadora:**



---

Prof. Msc. Marcelo Ferreira Rego  
Orientador



---

Prof.ª Dr.ª Caroline Queiroz Santos  
Integrante da Banca Examinadora



---

Prof. Msc. Áthila Rocha Trindade  
Integrante da Banca Examinadora

Diamantina

2019

*Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, autor de meu destino, meu guia, socorro presente na hora da angústia, ao meu pai, minha mãe e aos meus irmãos.*

# AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom e por ter me proporcionado chegar até aqui.

Agradeço, aos meus pais, João Alcebíades e Maria do Espírito, aos meus irmãos, Rochelly Fernandes, Crisley Jordania e Rejane Fernandes, por não medirem esforços para que eu pudesse levar meus estudos adiante. Agradeço, em especial, ao professor orientador Marcelo Ferreira, pela paciência e incentivo que tornaram possível a conclusão desta monografia.

Agradeço a todos os professores do curso de Sistemas de Informação da UFVJM, que dividiram seus conhecimentos, que auxiliaram meu crescimento profissional e pessoal durante minha graduação.

Agradeço a Brenda Orlandi, por sempre me animar em momentos difíceis.

Agradeço aos meus colegas, Breno Caldeira, Brian Azevedo, Luiz Felipe, Luciano Silva, Marcus Paulo, Matheus Guedes, Réggis Dias, Rafael Pelli, Pedro Henrique e Thomaz Souto e aos demais amigos, por confiarem em mim e estarem do meu lado em todos os momentos da vida.

*“Se não existe esforço, não existe progresso.”*  
(Frederick Douglass)

# RESUMO

A Coordenação de Processos Seletivos (COPESE) é o setor da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) responsável por gerenciar os processos para o ingresso nos cursos de graduação (vestibulares) da instituição. Em alguns processos seletivos organizados pela COPESE, exige-se que o candidato escreva uma redação que é avaliada por uma banca de corretores. A avaliação da redação é baseada no método de "correção cega", na qual os avaliadores recebem as provas sem a identificação dos candidatos. Atualmente, cada corretor recebe as provas de redação e escreve na folha de resposta. Desse modo, quando o corretor seguinte recebe essa folha, ele tem acesso as notas e as observações feitas pelo corretor anterior. Acredita-se que o método de correção de redações atualmente adotado pela COPESE, pode estar causando uma influência na avaliação do segundo corretor, uma vez que ele fará sua avaliação tendo conhecimento da avaliação do corretor anterior. Desta forma, o presente trabalho apresenta uma solução que forneça maior sigilo, apoiando e facilitando o trabalho dos servidores e corretores que são responsáveis por participar destes processos.

**Palavras-chave:** Desenvolvimento. Sistemas de Informação. Sistema Especialista. Sistema web. Modelo Cascata.



# ABSTRACT

The Coordination of Selective Processes (COPESE) is the sector of the Federal University of Jequitinhonha and Mucuri Valleys (UFVJM) responsible for managing the processes for the entrance in the institution's undergraduate (entrance exam) courses. In some selection processes organized by COPESE, the candidate is required to write an essay that is evaluated by a brokerage board. The essay evaluation is based on the "blind correction" method, in which the evaluators receive the tests without identifying the candidates. Each broker currently receives proofreading and writes on the answer sheet. That way, when the next broker receives this sheet, he has access to the notes and observations made by the previous broker. It is believed that the wording correction method currently adopted by COPESE may be having an influence on the assessment of the second broker, as it will make its assessment having knowledge of the assessment of the previous broker. Thus, the present work presents a solution that provides greater confidentiality, supporting and facilitating the work of the servers and brokers who are responsible for participating in these processes.

**Keywords:** Development, Information Systems, Specialist System, Web System, waterfall model,

# LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura Cliente Servidor . . . . .	9
Figura 2 – Representação do Modelo MVC . . . . .	10
Figura 3 – Esquema de funcionamento dos controles de versão . . . . .	11
Figura 4 – Exemplo de utilização de tags HTML . . . . .	13
Figura 5 – Exemplo de texto sem estilo e com estilo . . . . .	14
Figura 6 – Requisição Síncrona e Assíncrona . . . . .	15
Figura 7 – Requisição Síncrona e Assíncrona . . . . .	16
Figura 8 – Histórico de commits do GitKraken . . . . .	18
Figura 9 – Descrição do Modelo em Cascata . . . . .	19
Figura 10 – Descrição do Modelo em Cascata Modificada . . . . .	20
Figura 11 – Especificação de Software . . . . .	21
Figura 12 – Diagrama de Caso de Uso . . . . .	26
Figura 13 – Cadastrar Processo . . . . .	39
Figura 14 – Consultar Processo . . . . .	40
Figura 15 – Visualização Redação - Admin . . . . .	40
Figura 16 – Tela inicial corretor . . . . .	41
Figura 17 – Avaliação de Redação . . . . .	41
Figura 18 – Teste de Shapiro-Wilk da amostras Data_s . . . . .	43
Figura 19 – Teste de Shapiro-Wilk da amostras Data_c . . . . .	43
Figura 20 – Resultado do teste de Mann-Whitney . . . . .	44
Figura 21 – Login . . . . .	51
Figura 22 – Tela Principal Administrador . . . . .	52
Figura 23 – Consulta Processo - Número Edital . . . . .	52
Figura 24 – Cadastro Usuário . . . . .	53
Figura 25 – Consulta Usuário . . . . .	53
Figura 26 – Editar Usuário . . . . .	54
Figura 27 – Arquivar processo . . . . .	54
Figura 28 – Download do Processo . . . . .	55

# LISTA DE TABELAS

Tabela 1 – Tags HTML . . . . .	12
Tabela 2 – Etapas do Modelo de Desenvolvimento de Software em Cascata . . . . .	19
Tabela 3 – Requisito Funcional 1 . . . . .	22
Tabela 4 – Requisito Funcional 2 . . . . .	22
Tabela 5 – Requisito Funcional 3 . . . . .	23
Tabela 6 – Requisito Funcional 4 . . . . .	23
Tabela 7 – Requisito Funcional 5 . . . . .	23
Tabela 8 – Requisito Funcional 6 . . . . .	23
Tabela 9 – Requisito Funcional 7 . . . . .	24
Tabela 10 – Requisito Funcional 8 . . . . .	24
Tabela 11 – Requisito Não Funcional 1 . . . . .	24
Tabela 12 – Requisito Não Funcional 2 . . . . .	25
Tabela 13 – Caso de Uso 1 . . . . .	27
Tabela 14 – Caso de Uso 2 . . . . .	28
Tabela 15 – Caso de Uso 3 . . . . .	29
Tabela 16 – Caso de Uso 4 . . . . .	29
Tabela 17 – Caso de Uso 5 . . . . .	30
Tabela 18 – Caso de Uso 6 . . . . .	30
Tabela 19 – Caso de Uso 7 . . . . .	31
Tabela 20 – Caso de Uso 8 . . . . .	32
Tabela 21 – Caso de Uso 9 . . . . .	33
Tabela 22 – Caso de Uso 10 . . . . .	34
Tabela 23 – Caso de Uso 11 . . . . .	35
Tabela 24 – Caso de Uso 12 . . . . .	35
Tabela 25 – Caso de Uso 13 . . . . .	36
Tabela 26 – Caso de Uso 14 . . . . .	37
Tabela 27 – Visão Geral das Amostras . . . . .	42

# LISTA DE ABREVIATURAS E SIGLAS

UFVJM	<i>Universidade Federal dos Vales do Jequitinhonha e Mucuri</i>
COPESE	<i>Coordenação de Processos Seletivos</i>
TI	<i>Tecnologia da Informação</i>
SGBD	<i>Sistema de Gerenciamento de Banco de dados</i>
POO	<i>Programação Orientada a Objetos</i>
LM	<i>Linguagem de Marcação</i>
MVC	<i>Model-View-Controller</i>
HTML	<i>Hypertext Markup Language</i>
RF	<i>Requisitos Funcionais</i>
RNF	<i>Requisitos Não-Funcionais</i>
LEC	<i>Licenciatura em Educação do Campo</i>
EAD	<i>Educação a Distância</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>2</b>
1.1	Descrição do Problema	2
1.2	Objetivos Gerais	3
1.3	Objetivos específicos	3
1.4	Justificativa	3
1.5	Organização da Monografia	4
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>5</b>
2.1	Tecnologia da Informação	5
2.2	Sistema de Informação	5
2.3	Sistema Especialista	6
2.4	World Wide Web	6
2.5	Banco de dados	7
2.5.1	Gerenciador de Banco de Dados	7
2.6	Programação Orientada a Objetos	8
2.7	Linguagem de Programação	8
2.8	Linguagem Compilada e Linguagem Interpretada	8
2.9	Linguagem de Marcação	9
2.10	Arquitetura cliente/servidor	9
2.11	Framework de Software	9
2.12	Padrão MVC	10
2.13	Sistema de Controle de Versão	10
<b>3</b>	<b>TECNOLOGIAS UTILIZADAS</b>	<b>12</b>
3.1	HTML	12
3.2	CSS	13
3.3	JavaScript	15
3.4	Laravel	16
3.5	PostgreSQL	16
3.6	Apache	17
3.7	Git	17
<b>4</b>	<b>METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE</b>	<b>19</b>
4.1	Requisitos	21

---

4.1.1	Riscos . . . . .	22
4.1.2	Requisitos Funcionais . . . . .	22
4.1.3	Requisitos Não Funcionais . . . . .	24
4.2	Projeto . . . . .	25
4.2.1	Escopo . . . . .	25
4.2.2	Perspectiva do Produto . . . . .	25
4.2.3	Características dos Usuários . . . . .	25
4.2.4	Funções do Produto . . . . .	26
4.3	Implementação . . . . .	27
4.3.1	Casos de uso do ator Administrador . . . . .	27
4.3.2	Corretor . . . . .	36
4.4	Testes . . . . .	37
4.5	Implantação . . . . .	38
<b>5</b>	<b>APRESENTAÇÃO DO SISTEMA . . . . .</b>	<b>39</b>
5.1	Telas do Administrador . . . . .	39
5.2	Telas do Corretor . . . . .	40
<b>6</b>	<b>ANÁLISE DOS RESULTADOS . . . . .</b>	<b>42</b>
6.1	Normalidade . . . . .	43
6.2	Teste de Tendência Central . . . . .	43
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>45</b>
	<b>Referências . . . . .</b>	<b>46</b>
	<b>Anexos . . . . .</b>	<b>50</b>
	<b>ANEXO A Telas do sistema . . . . .</b>	<b>51</b>

# 1 INTRODUÇÃO

A Coordenação de Processos Seletivos (COPESE) é o setor da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) responsável por gerenciar os processos para o ingresso nos cursos de graduação (vestibulares) da instituição. As principais responsabilidades desse setor são: a formação e a supervisão das equipes de trabalho envolvidas em cada processo seletivo (elaboradores de provas, aplicadores de provas, corretores de provas, dentre outros); gerenciamento do processo de inscrição dos candidatos; realização da distribuição dos candidatos de acordo com os locais de provas; coordenação da logística de aplicação das provas; divulgação dos gabaritos oficiais e do resultado final.

## 1.1 Descrição do Problema

Em alguns processos seletivos organizados pela COPESE, exige-se que o candidato escreva uma redação que é avaliada por uma banca de corretores. A avaliação da redação é baseada no método de "correção cega", na qual os avaliadores recebem as provas sem a identificação dos candidatos. Além disso, cada redação deve ser corrigida por, no mínimo, dois corretores diferentes e uma terceira correção acontece caso haja uma grande divergência entre as notas atribuídas nas duas primeiras correções, a nota final é calculada a partir da média das correções. Uma divergência entre as notas atribuídas por dois corretores é classificada da seguinte maneira:

Considerando que:

- **Divergência** é a diferença em módulo entre as notas atribuídas por dois avaliadores para uma redação de um candidato. Representado por  $D$
- **Tolerância** é igual à 20% da nota máxima possível para uma redação. Representado por  $T$ .

$$\text{Classificação da divergência} = \begin{cases} \text{pequena} & , \text{ se } D \leq T \\ \text{grande} & , \text{ se } D > T \end{cases}$$

Atualmente, cada corretor recebe as provas de redação e escreve na folha de resposta suas observações e a nota referente a sua correção. Desse modo, quando o corretor seguinte recebe essa folha, ele tem acesso as notas e as observações feitas pelo corretor anterior.

Acredita-se que o método de correção de redações atualmente adotado pela COPESE pode estar causando uma influência na avaliação do segundo corretor uma vez que ele fará sua avaliação tendo conhecimento da avaliação do corretor anterior.

## 1.2 Objetivos Gerais

Tendo em vista como é realizado o processo de avaliações de redações da UFVJM. O presente trabalho apresenta uma solução que forneça maior sigilo e apoie e facilite o trabalho dos servidores e corretores que são responsáveis por participar destes processos.

## 1.3 Objetivos específicos

Dentre os objetivos específicos do trabalho, destacam-se:

- Realizar o levantamento dos requisitos funcionais e não-funcionais para o desenvolvimento do sistema.
- Estudar as ferramentas computacionais para o desenvolvimento do sistema.
- Definir uma metodologia de desenvolvimento de software.
- Desenvolver um sistema especialista para gerenciamento do processo de avaliação de redações.
- Desenvolver um sistema com interface amigável.
- Desenvolver um sistema que armazene as informações das redações em um ambiente seguro.
- Desenvolver um sistema que forneça agilidade no gerenciamento do processo de avaliação de redação.
- Realizar teste do sistema junto com usuário.
- Verificar se o método de avaliação de redações utilizando o sistema proposto gerou alguma diferença nas notas atribuídas pelo corretores.

## 1.4 Justificativa

O trabalho proposto é desenvolver um sistema especialista, que cumpra a necessidade de fornecer maior sigilo das informações sobre o processo de correção de redações dos vestibulares da UFVJM. Para isso, apenas o administrador do sistema, que gerencia o processo, poderá acompanhar todas as etapas e todas as observações de cada correção. Além disso, a verificação



de divergências nas notas será automatizada a fim de verificar a necessidade de uma terceira correção. Ao final do processo de correção, o sistema disponibilizará um relatório contendo as notas finais e observações dos corretores.

## **1.5 Organização da Monografia**

Este trabalho está organizado da seguinte forma: o [Capítulo 1](#) trata-se da introdução, o [Capítulo 2](#) sobre a fundamentação teórica, o [Capítulo 3](#) apresenta as tecnologias utilizadas para o desenvolvimento do sistema, o [Capítulo 4](#) descreve sobre a metodologia utilizada para o desenvolvimento do sistema, [Capítulo 5](#) apresenta as principais telas do sistema, o [Capítulo 6](#) apresenta análise dos resultados e o [Capítulo 7](#) trata-se da conclusão.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica dos conceitos utilizados ao longo deste trabalho.

### 2.1 Tecnologia da Informação

De acordo com [Laudon and Laudon \(2007\)](#) Tecnologia da Informação (TI) deve ser compreendida não apenas pelo uso de computadores em si, mas também como toda solução de software e hardware que uma empresa necessita para alcançar seus objetivos organizacionais. Já [Rezende \(2011\)](#) descreve que a Tecnologia da Informação compreende os recursos tecnológicos e computacionais para o armazenamento, desenvolvimento, uso da informação e do conhecimento onde está fundamentada nos seguintes componentes: hardware e seus dispositivos periféricos; recursos de softwares; sistemas de telecomunicações e gestão de informação.

Os autores Albertin e Moura relatam sobre a importância da Tecnologia da informação nas organizações no seguinte trecho:

A importância da TI (Tecnologia da Informação) para as organizações é inegável, tanto na sua operacionalização quanto na sua estratégia competitiva. Atualmente, a utilização da TI como arma competitiva tem recebido grande atenção por parte dos executivos das organizações. ([ALBERTIN AND MOURA, 1995, p. 23](#)).

Segundo [Albertin \(2001\)](#) a Tecnologia da Informação pode ser decisiva para o sucesso ou fracasso de uma organização, de forma que contribui para que a ela seja ágil, flexível e forte. Contemplando com a mesma linha de pensamento, [Laudon and Laudon \(2007\)](#) afirma que a infra-estrutura de TI provê a base para qual uma empresa seja capaz de montar seus sistemas de informação.

### 2.2 Sistema de Informação

Conforme [Laudon and Laudon \(2007\)](#) um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informações, com o propósito de apoiar a tomada de decisões. Este conjunto é composto por três atividades que produzem informações para as organizações: entrada, o processamento e a saída. Sobre essa mesma perspectiva, [Gonçalves \(2009\)](#) descreve que todos os sistemas que manipulam dados e geram informações, usando ou não recursos tecnológicos, como sistemas de informação.

Segundo [Laudon and Laudon \(2007\)](#), a utilização de sistemas de informação proporcionam vantagens competitivas para uma organização, como por exemplo, melhoria na tomada de decisões, aumento do controle de operações, eficiência na análise de problemas e possibilidade de criação de novos produtos ou serviços.

## 2.3 Sistema Especialista

Um sistema especialista é construído capturando a expertise humana em uma área específica de conhecimento e transformando ela em um conjunto de regras para um sistema de software que pode ser usado por outras pessoas da organização. Esses sistemas normalmente executam um número limitado de tarefas que são realizadas por profissionais em alguns minutos ou horas. Tais sistemas são úteis em situações de tomada de decisão em que a expertise é cara ou escassa. [Laudon and Laudon \(2007\)](#).

Mendes destaca um aspecto importante para a compreensão de um sistemas especialistas:

Nem todos os problemas devem ser resolvidos por meio de sistemas especialistas. Existem características que indicam se determinado problema deve ou não ser instrumentalizado por esta tecnologia. A análise do problema, então, constitui-se no primeiro estágio do ciclo de desenvolvimento dos sistemas especialistas, contribuindo fortemente para o sucesso da implementação do sistema. Buscando facilitar o processo de análise do problema, distinguimos dentre outras, algumas condições, que, se observadas, poderão contribuir para a identificação do nível de adequação do uso da tecnologia de sistemas especialistas para a resolução do mesmo. ([MENDES, 1997](#), p. 1).

Para complementar [Mendes \(1997\)](#) cita que os sistemas especialistas podem trazer diversos benefícios para vida cotidiana de forma que possa auxiliar na tomada de decisões como por exemplo:

- Capacidade de melhorar a produtividade e desempenho de seus usuários.
- Possibilidade de ser distribuído para diversas pessoas, desta forma possível estender a facilidade na tomada de decisão.

## 2.4 World Wide Web

De acordo com [Martins \(2008\)](#) a ideia de World Wide Web surgiu em 1980, na Suíça e o precursor foi o britânico Tim Berners-Lee. Em 1990, foi usado um computador NeXTcube como seu primeiro servidor web e também utilizado para implementar o primeiro navegador, o World Wide Web. No ano seguinte, Tim Berners-Lee postou um resumo sobre todas as suas

ideias e projetos no grupo de notícias de nome alt.hypertext. Esta foi a marca de estreia oficial da Web como um serviço publicado na Internet.

[Marques and Marin \(2002\)](#) descrevem a World Wide Web ou apenas Web, como um aglomerado de documentos baseados na linguagem de hipertexto, que é utilizada por milhares de computadores que podem ser acessados através da Internet. Já para [Forouzan \(2009, p. 851\)](#) define WWW como “[...] um repositório de informação interligadas por diversos pontos espalhado ao redor do mundo.”. A Web apresenta como vantagem ser capaz de armazenar, recuperar, formatar e mostrar informação em formato de página da Internet, desta forma é possível servir de base para novos tipo de sistemas. [Laudon and Laudon \(2007\)](#).

## 2.5 Banco de dados

Conforme [Elmasri and Navathe \(2011, p.3\)](#) “[...] banco de dados é uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significados implícitos.”. Os bancos de dados são capazes de reunir informações de maneira organizada, consistente, protegida e acessível em tempo hábil. Desta forma é possível atender desde sistemas mais simples aos mais complexos. [Alecrim \(2018\)](#).

Segundo [Gomes \(2019\)](#), os bancos de dados são classificados baseando-se em suas estruturas lógicas, sendo assim os principais bancos são os relacionais e não-relacionais. Desta forma, [Laudon e Laudon](#) descreve a seguir, como o é organizado o banco de dados relacional:

Bancos de dados relacionais organizam os dados em tabelas bidimensionais (denominadas relações) com colunas em linhas. Cada tabela contém dados referentes a um entidade e seus atributos. Na maioria dos casos monta-se uma tabela para cada entidade do negócio. ([LAUDON AND LAUDON, 2007, p. 10](#)).

O banco de dados não-relacional desprende-se da existência de colunas pré definidas como no modelo relacional. Este modelo pode ser classificados em 4 tipos, sendo eles: orientado a documento, chave-valor, orientados a colunas e grafos. Cada tipo de banco de dados não-relacional possui características e requerimentos que os diferenciam dos bancos de dados relacionais. [Groffe \(2016\)](#).

### 2.5.1 Gerenciador de Banco de Dados

[Laudon and Laudon \(2007\)](#) descreve que um sistema de Gerenciamento de Banco de dados (SGBD) é um tipo de software usado para criar, armazenar, organizar e acessar dados a partir de um banco de dados. Segundo [Gomes \(2019\)](#) os mais populares SGBDs do modelo relacionais são os MySQL e PostgreSQL, já para o modelo não-relacional temos o MongoDB e o Redis.

## 2.6 Programação Orientada a Objetos

Gasparotto (2014) afirma que existem diversas linguagens de programação no mercado que seguem diferentes paradigmas. A Programação Orientada a Objetos (POO) é um paradigma que consiste em um padrão de desenvolvimento voltado para a segurança e reaproveitamento de código. Gudwin (1997) afirma que no passado os paradigmas eram baseados em módulos procedimentais que eram alimentados por dados, fornecendo assim, novos dados. Para POO essa concepção teve uma mudança, dando a origem a objetos como módulos que se comunicam por mensagens, encapsulando ao mesmo tempo dados e funções.

## 2.7 Linguagem de Programação

Segundo Medina and Ferting (2006) a linguagem de programação que um computador é capaz de compreender é composta apenas de números. Desta forma desenvolver algoritmos nesta linguagem se torna extremamente complexo para nós, seres humanos. Por isso, para facilitar a programação, foi necessário o desenvolvimento de um código mais fácil para o ser humano que relacionasse com a linguagem de máquina. Assim surgiram as linguagens de programação como JAVA, C, Assembly e entre outras.

Para Medina and Ferting (2006) as linguagens de programação são classificadas dada a sua proximidade com a linguagem de máquina. Desta forma as linguagens de programação de alto nível são linguagens distantes da linguagem máquina e próxima da linguagem natural. Para Gudwin (1997) linguagens de programação de baixo nível são linguagens cujas instruções correspondem quase que diretamente ao código de máquina que será enviado ao processador para execução.

## 2.8 Linguagem Compilada e Linguagem Interpretada

A linguagem compilada, tem como pré requisito executar um programa para traduzir os códigos fontes em código executável. Os programas compilados, só podem rodar no tipo de computador para o qual foram compilados. Estes programas apresentam altas performance devido que o funcionamento foi desenvolvido para um tipo específico de processador. Indrusiak (1996).

Segundo Indrusiak (1996) as linguagens interpretadas só existem no código fonte, de forma que quando estão em execução, é chamado o interpretador que é um programa que toma o código fonte e executa as ações indicadas pelos comandos. Para Bastos (2008) a linguagem interpretada é apenas uma linguagem intermediária durante a execução do programa, que consiste na tradução dos comandos do código fonte para a linguagem de máquina.

## 2.9 Linguagem de Marcação

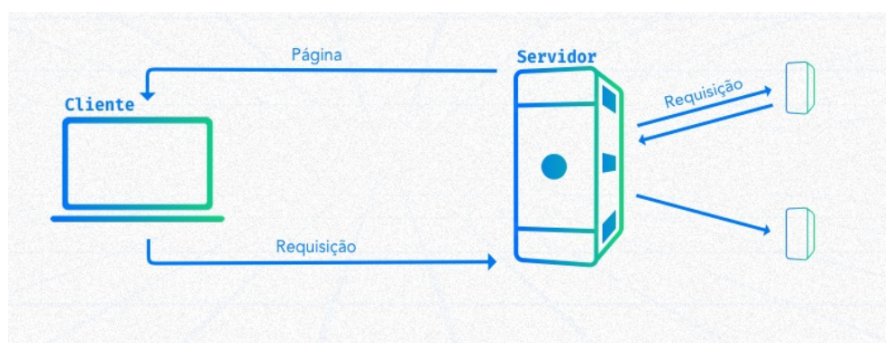
Existem diversos tipos de Linguagem de Marcação (LM), cada uma contém sua forma de utilização. As LM tem como característica principal criar marcas (tags) para delimitar um texto. Com a evolução das linguagens, tornou-se possível usar marcas para atribuir significado ao texto. [Furgeri \(2006\)](#).

Cada tag possui uma função específica, seja simplesmente para marcar um texto, seja para definir uma estrutura mais complexa como, por exemplo, uma tabela, um formulário, etc. Ao se delimitar um texto por marcas. É possível estabelecer um conteúdo semântico que pode ser tratado e manipulado por programas de computador. ([FURGERI, 2006](#), p. 226).

## 2.10 Arquitetura cliente/servidor

Conforme [Winckler and Pimenta \(2002\)](#) a arquitetura cliente/servidor é desenvolvido sobre uma rede de computadores. Da parte do cliente está um programa chamado browser ou navegador que intermedia a solicitação de informações ao servidor e as apresenta para o usuário. O servidor atende os diferentes clientes bem como outros servidores indistintamente. Este modelo pode ser observado na figura 1.

Figura 1: Arquitetura Cliente Servidor



Fonte: [Schneider \(2016\)](#)

## 2.11 Framework de Software

Para [Lima \(2011\)](#) um framework de software é um conjunto de classes que incorpora um projeto. Um framework é reusável e seu código é envolvido por uma interface muito bem definida, através da qual as aplicações são construídas. Os frameworks estão disponíveis para auxiliar a construção de aplicações de software.

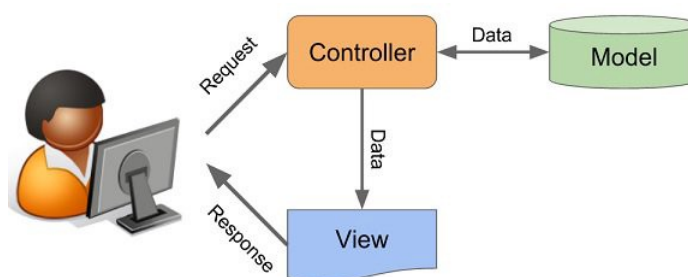
Um framework pode ser compreendido com uma conjunto de códigos, desenvolvido por uma linguagem de programação que busca disponibilizar funcionalidades ao desenvolvedor de

software. Estas funcionalidades são testadas e prontas para serem utilizadas na implementação de software, poupando muito tempo e trabalho na elaboração de operações como acesso a banco de dados, sistema de templates e mapeamentos de rotas. Jaques (2016).

## 2.12 Padrão MVC

Para da Silva (2012) o padrão MVC (Model-View-Controller) propõe uma arquitetura de software dividida em componentes, de modo que, o código fique organizado e enxuto, e posteriormente possa ser reutilizado. Já LUCIANO and ALVES (2017) o padrão MVC separa as camadas para aumentar a capacidade de reutilização e manutenção do projeto. A sua dinâmica é compreensível, todas as requisições da aplicação são direcionadas para a camada Controller, que acessa a camada Model para processar a tal requisição, e por fim exibe o resultado da camada View. A representação do modelo MVC pode ser observado na figura 2.

Figura 2: Representação do Modelo MVC



Fonte: Vilarinho (2018)

## 2.13 Sistema de Controle de Versão

Um sistema de controle de versão tem como função gerenciar diferentes versões do mesmo projeto. É possível obter históricos das modificações, permitindo que vários programadores trabalhem no mesmo projeto, e disponibiliza um comparativo entre várias versões do projeto. Ramos (2017).

De acordo com Lacerda (2012) um sistema de controle de versão é capaz de armazenar os arquivos de um projeto em repositório juntamente com o histórico de suas versões. Desta forma os desenvolvedores são capazes de acessar e recuperar as versões, além de fazer uma cópia local que pode ser alterada. A cada alteração é possível enviar ao servidor e atualizar a versão a partir de outras feitas pelo demais desenvolvedores. Os sistemas mais conhecidos são o Git e o Mercurial. A figura 3 representa o esquema de funcionamento de um sistema de controle de versão.

Figura 3: Esquema de funcionamento dos controles de versão



Fonte: [Lacerda \(2012\)](#)



## 3 TECNOLOGIAS UTILIZADAS

Este capítulo apresenta as tecnologias utilizadas para implementar o sistema proposto.

### 3.1 HTML

O HTML ( **H**ypertext **M**arkup **L**anguage) é uma **linguagem de marcação** que é muito utilizada no desenvolvimento de páginas na **Web**. Esta linguagem utiliza de marcações para indicar ao navegador o que é cada informação, como por exemplo: o que é um título, paragrafo ou até mesmo um botão. A tabela 1 apresenta exemplos dessas tags.

Tag	Descrição
<html>... </html>	Define um documento HTML;
<head>... </head>	Define uma informação sobre o documento;
<body>... </body>	Define o corpo da página;
<button>... </button>	Define um botão de comando;
<title>... </title>	Define um título ao documento;
<i>... </i>	Define um texto em itálico;
<b>... </b>	Define um texto em negrito;
<p>... </p>	Define um parágrafo;

Tabela 1: Tags HTML

Para melhor compreensão desta linguagem, segue em amostra um código escrito na linguagem de marcação. A figura 4 apresenta o resultado desse código após ser interpretado pelo navegador.

```

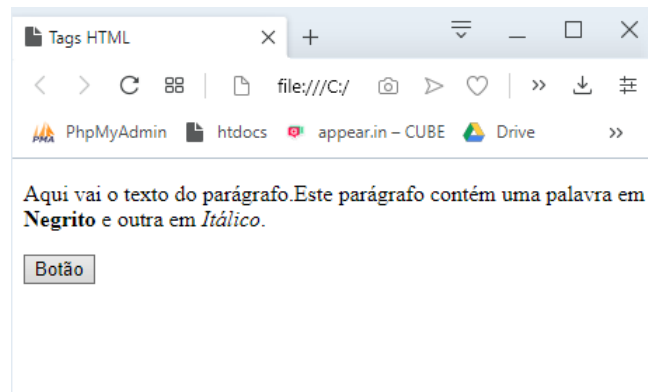
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>Tags HTML</title>
6 </head>
7 <body>
8 <p>Aqui contem o texto do par grafo .Este texto cont m uma palavra em <b
   >Negrito</b> e outra em <i>It lico</i>. </p>
9 <button>Bot o</button>
10 </body>
11 </html>

```

Código Fonte 3.1: Exemplo de Tags HTML

Optou por utilizar esta linguagem devido a sua vasta documentação, popularidade, facilidade de uso e sua fácil integração com CSS, Bootstrap e JavaScript.

Figura 4: Exemplo de utilização de tags HTML



Fonte: Dados do Autor 2018

## 3.2 CSS

O CSS é uma linguagem usada para adicionar estilos a um documento HTML. Este mecanismo é utilizado para estilizar elementos escritos por uma [linguagem de marcação](#). O código fonte a seguir mostra um exemplo de aplicação simples de um estilo em um código HTML.

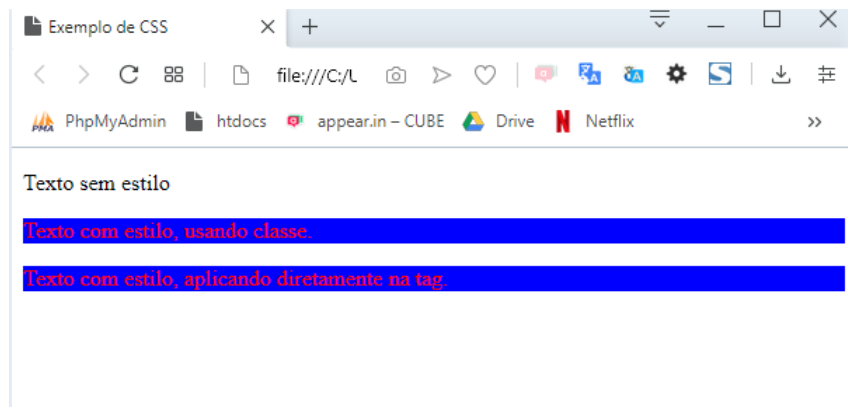
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Exemplo de CSS</title>
5   <style type="text/css">
6     .exemplo{
7       background: blue;
8       color: red;
9     }
10  </style>
11 </head>
12 <body>
13   <p>Texto sem estilo </p>
14   <p class="exemplo">Texto com estilo , usando classe. </p>
15   <p style="background:blue; color: red ">Texto com estilo , aplicando
16     diretamente na tag. </p>
17 </body>
18 </html>
```

Código Fonte 3.2: Exemplo de aplicação de CSS no HTML

A figura 5 apresenta o resultado desse código após ser interpretado pelo navegador.

O Bootstrap é um [framework](#) de código-fonte aberto para o desenvolvimento de componentes de interface para sites e aplicações Web. Ele utiliza o CSS e o JavaScript para criar uma interface que melhore a experiência do usuário e que o site fique visualmente amigável e

Figura 5: Exemplo de texto sem estilo e com estilo



Fonte: Dados do Autor 2018

responsivo. Este **framework** possui diversas componentes desenvolvidos em CSS e JavaScript, que podem ser reutilizadas em outros projetos possibilitando um aumento de produtividade. A documentação do Bootstrap é bem simples e intuitiva, dessa forma, até quem não tem experiência em programação pode apreendê-la. O código a seguir mostra um exemplo simples de sua utilização:

```

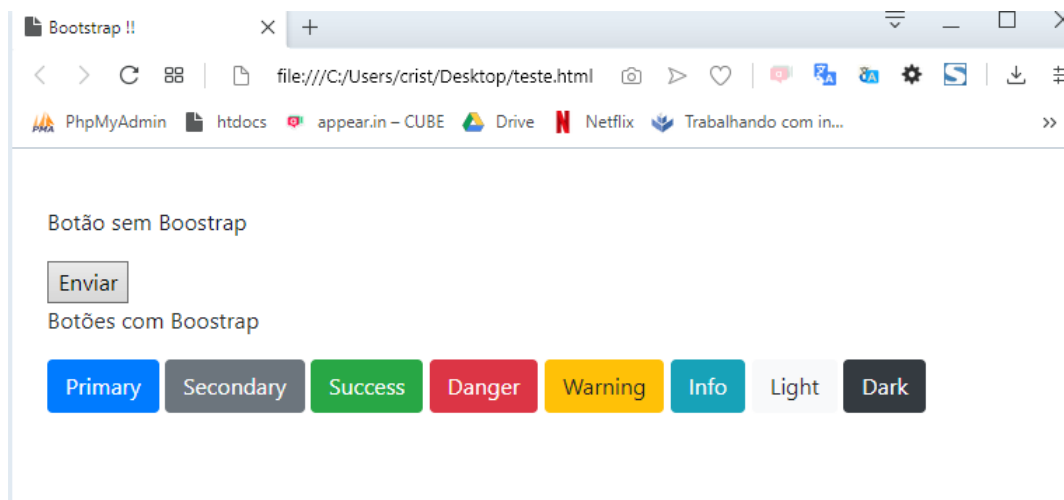
1  <div class="container">
2    <div class="row" style="margin-top: 40px">
3      <p>Bot o sem Bootstrap</p>
4    </div>
5    <div class="row">
6      <button>Enviar</button><br>
7    </div>
8
9    <div class="row" >
10     <p>Bot es com Bootstrap</p>
11   </div>
12   <div class="row">
13     <div class="form-group">
14       <button type="button" class="btn btn-primary">Primary</button>
15       <button type="button" class="btn btn-secondary">Secondary</button>
16       <button type="button" class="btn btn-success">Success</button>
17       <button type="button" class="btn btn-danger">Danger</button>
18       <button type="button" class="btn btn-warning">Warning</button>
19       <button type="button" class="btn btn-info">Info</button>
20       <button type="button" class="btn btn-light">Light</button>
21       <button type="button" class="btn btn-dark">Dark</button>
22     </div>
23   </div>
24 </div>

```

Código Fonte 3.3: Exemplo de aplicação de Bootstrap no HTML

O navegador irá interpretar o código HTML acima, e apresentará o seguinte resultado:

Figura 6: Requisição Síncrona e Assíncrona



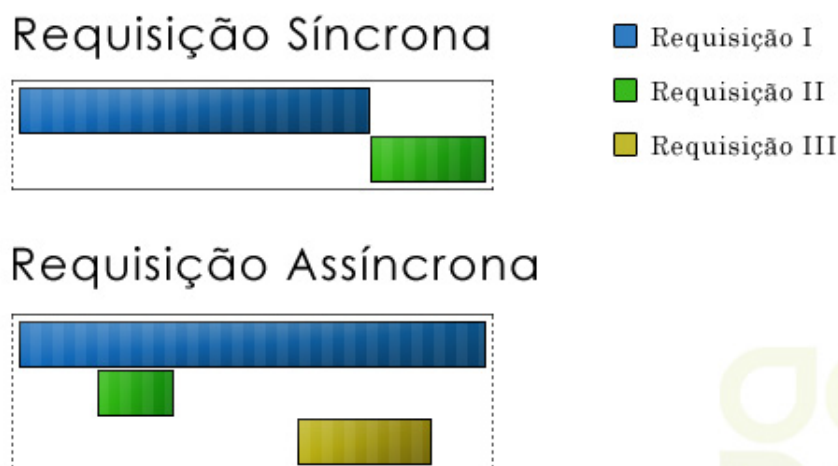
Fonte:Dados do Autor(2019)

### 3.3 JavaScript

JavaScript é uma linguagem de programação interpretada de alto nível. Esta linguagem permite que páginas Web se tornem interativas e dinâmicas, tonando possível controlar multimídias, imagens animadas e entre várias outras coisas. JavaScript tem como vantagem possuir maior usabilidade por ser dinâmica em comparação com o HTML que é estática. Esta linguagem é capaz de ser aplicada a certos elementos de páginas de internet para executar eventos específicos como cliques e rolagens de mouse personalizada.

O Ajax é uma tecnologia que utiliza o JavaScript, XML e HTML. Esta tecnologia não é uma linguagem de programação, mas uma nova forma de utilizar os padrões existentes. O Ajax permite efetuar requisições assíncrona no servidor, de modo que não necessita o recarregamento da página Web para atualizar seus dados. As requisição assíncronas não possuem sincronismo entre suas requisições, sendo assim possível enviar várias requisições em paralelo, onde cada resposta retorna quando estiver pronta. A figura 7 ilustra o conceito de requisições síncrona e assíncrona.

Figura 7: Requisição Síncrona e Assíncrona



Fonte: [Matheus \(2013\)](#)

### 3.4 Laravel

A linguagem de programação PHP é uma linguagem interpretada que pode ser embutida no HTML. O código é interpretado do lado do servidor pelo módulo do PHP, que gera uma página web do lado do cliente. Esta linguagem é muito popular devido a facilidade em apreendê-la, manuseá-la, além de ser compatível com quase todos os sistemas operacionais existentes.

Laravel é um framework de PHP e open-source que é utilizado para desenvolvimento de sistemas web que utiliza o padrão MVC. Composer é utilizado pelo Laravel para gerenciar as dependências, permitindo a instalação e utilização de pacotes de terceiros em sua aplicação. Um ponto interessante é a sua documentação, que é bastante intuitiva e de fácil de compreensão. Este framework possui uma interface de linha de comando denominada Artisan, que fornece vários comandos para facilitar o desenvolvimento da aplicação. Para o mapeamento de URLs são utilizadas rotas que permite o direcionamento para alguma ação dentro da aplicação. Com uma configuração simples e pequena e com pouco código podemos configurar uma conexão com banco de dados. Devido ao conjunto de facilidades descritas, optou por utilizar este framework.

### 3.5 PostgreSQL

O pgAdmin é um software gráfico para a administração do SGBD PostgreSQL. O PostgreSQL é um SGBD de código aberto do tipo relacional que suporta o padrão SQL e oferece umas funções modernas, como o uso de triggers, Views e functions.

Segundo [Dionisio \(2015\)](#) este SGBD tem como sua principal função armazenar dados de forma segura, permitindo a recuperação dos dados a pedidos de outras aplicações de software. O PostgreSQL pode lidar com volume de dados que vão desde pequenas aplicações a aplicações de grande porte voltadas para a Internet.

Segundo [Alecgrim \(2006\)](#) o PostgreSQL possui diversas vantagens as principais são:

- Compatibilidade multi-plataforma;
- Compatibilidade com várias linguagens, entre elas, Java, PHP, Python, Ruby, e C/C++;
- Base de dados de tamanho ilimitado;
- Tabelas com tamanho de até 32 TB;
- Otimizado para aplicações que envolvem grande volume de dados ou que tratam de de informações críticas;

## 3.6 Apache

É um servidor web de código aberto que é amplamente utilizada pelos sites hospedados pela internet que implementa a arquitetura [cliente/servidor](#) e ele pode ser altamente personalizável devido a sua estrutura baseada em módulos, onde o usuário pode instalar novos recursos bem como ativa-las ou desativa-las livremente.

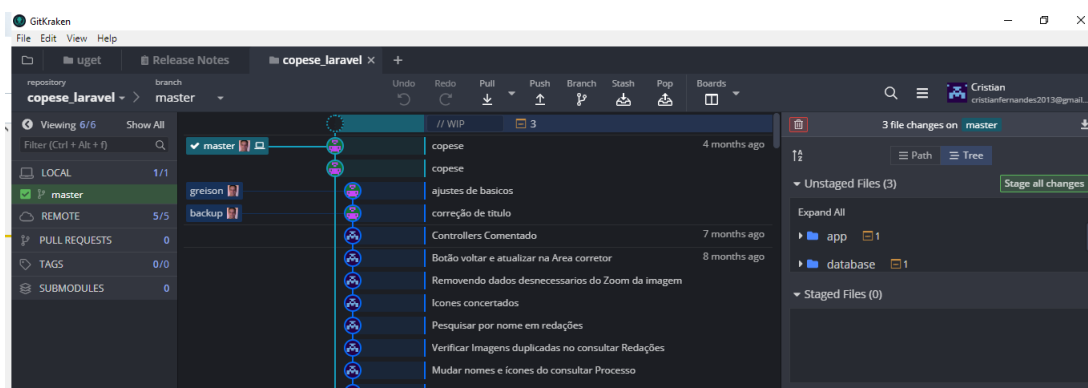
## 3.7 Git

O Git é um sistema de controle de versões de arquivos, usado principalmente no desenvolvimento de software. Este sistema possibilita o desenvolvimento de projetos com diversas pessoas permitindo a contribuição simultânea, de forma que o arquivo não corra risco que suas alterações serem sobrescritas.

O GitHub é um serviço web de hospedagem de código fonte que utiliza o Git. Desta forma é possível a acessar o repositório do projeto remotamente ou atualizar a versão local com maior facilidade.

O GitKraken é um cliente gráfico multiplataforma para o Git. Este software possui um visão gráfica muito bem feita da rede Git que permite entender o status atual do repositórios em termos de *commits* e *branches* como pode ser observado na figura 8. Outra vantagem que o GitKraken possui um integração com a conta do Github permitindo acessar e clonar seus repositórios remotos ou fazer um *fork* de um repositório existente. Outro ponto interessante deste software é a tela de conflitos de arquivos entre commits, que mostra de forma bem clara e específica o conflito para que possa manter ou sobrescrever o arquivo. A figura 8 apresenta a tela principal do GitKraken que mostra a rede de commits.

Figura 8: Histórico de commits do GitKraken

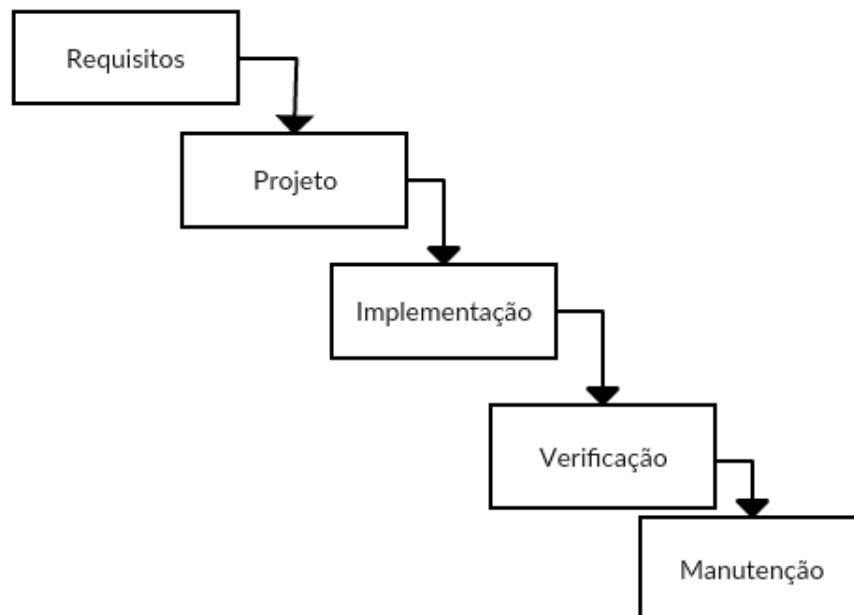


Fonte: Dados do Autor 2018

## 4 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

A metodologia do desenvolvimento do sistema proposto foi baseada no modelo de desenvolvimento de software em cascata, este modelo implementa etapas sequenciais, onde uma tarefa só inicia quando a tarefa anterior for completamente finalizada, conforme representa a figura 9.

Figura 9: Descrição do Modelo em Cascata



Fonte: [Silva \(2016\)](#) (Adaptada)

Segundo [Thomaz \(2017\)](#) cada etapa do modelo desenvolvimento de software em cascata, pode ser descrita da seguinte forma:

Etapa	Descrição
<b>Requisito</b>	Definição dos requisitos do produto e objetivos esperados.
<b>Projeto</b>	Idealização do desenvolvimento do produto ( estrutura e arquitetura de software).
<b>Codificação</b>	Desenvolvimento do software.
<b>Verificação</b>	Realização de testes no sistema.
<b>Manutenção</b>	Correção de problemas identificados.

Tabela 2: Etapas do Modelo de Desenvolvimento de Software em Cascata

A seguir são citadas vantagens e desvantagens dessa metodologia. [Builder \(2017\)](#).



Vantagens:

- É possível prever diversos cenários devido o planejamento feito previamente.
- Definir prazos mais previsíveis.
- Fácil gerenciamento do projetos, e etapas bem organizadas.

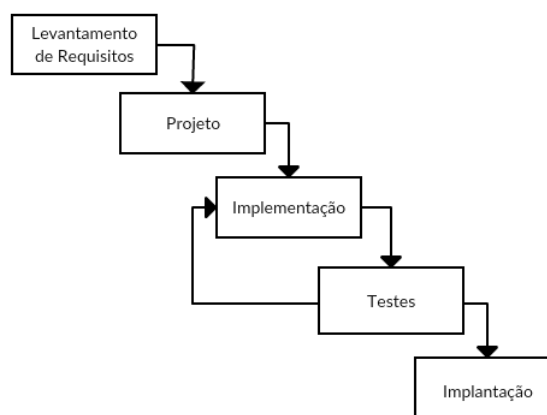
Desvantagens:

- O cliente irá visualizar algum resultado somente após a sua conclusão.
- Impõe grande rigidez durante a execução do projeto.
- O modelo cascata exige que o cliente estabeleça todos os requisitos no início do projeto.

O modelo em cascata visa documentar diversas de forma ampla, desde os requisitos à arquitetura de software, desta forma este modelo não é muito adaptável a mudanças. Por este motivo optou por realizar modificações neste modelo, para que seja possível realizar alterações ao longo do desenvolvimento.

As modificações desta metodologia consistem principalmente em simplificar a documentação, com objetivo de ser mais ágil e criar iteração entre as fases de Implementação e Verificação, ou seja, após cada atividade desenvolvida dentro da fase de Implementação são realizados os testes para que se torne um projeto mais flexível. A figura 10 representa essa modificação.

Figura 10: Descrição do Modelo em Cascata Modificada



Fonte: Próprio autor

Desta maneira as etapas dessa abordagem podem ser descritas da seguinte forma:

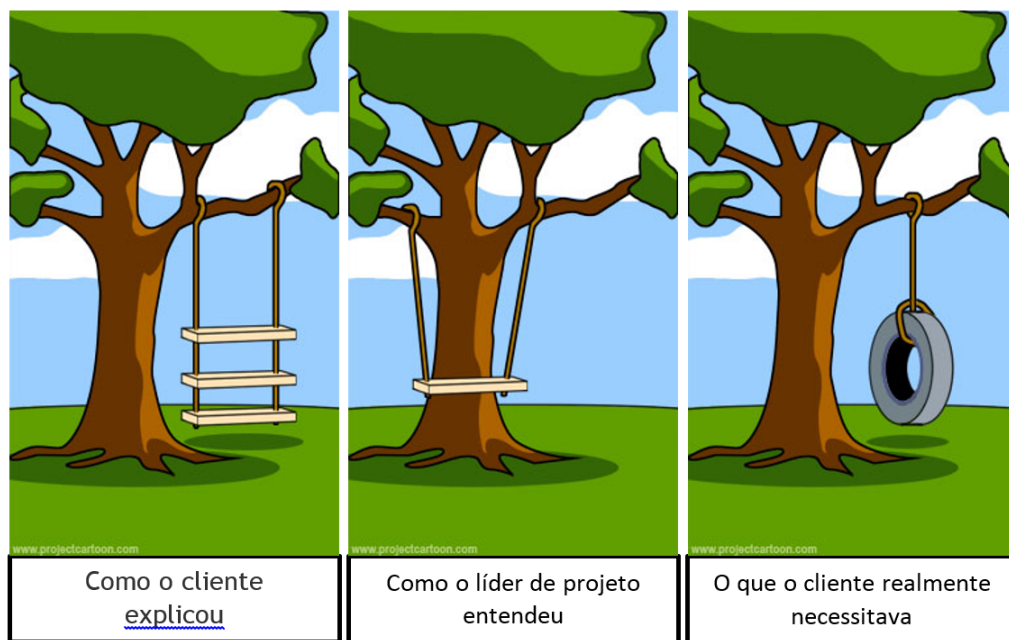
Etapa	Descrição
Requisito	Definição dos requisitos funcionais e não-funcionais.
Projeto	Visão geral sobre o sistema.
Implementação	Desenvolvimento do software baseado em módulos.
Testes	Realização de testes após cada módulo implementado.
Implantação	Implantação do sistema e treinamento de usuários.

As etapas dessa metodologia é descrita nas seções deste capítulo.

## 4.1 Requisitos

Os fracassos de um software geralmente estão atrelados na falha em compreender os requisitos. Dentre as principais razões para essas falhas destacam-se os requisitos mal organizados, mal expressos, e requisitos desnecessários para o cliente. [Medeiros \(2013\)](#). A figura 11 demonstra como o mal entendimento de requisitos pode ocasionar problemas na especificação do software.

Figura 11: Especificação de Software



Fonte: [Neponuceno \(2015\)](#)

Desta forma, esta seção descreve os Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF) coletados para a implementação do sistema .

### 4.1.1 Riscos

Para melhor compreensão dos requisitos, utilizou-se a seguinte tabela para descrever os graus de risco baseado no impacto que pode ocasionar ao sistema.

<b>Categoria</b>	<b>Definição</b>
C (Crítico):	Se o evento de risco ocorrer, o programa falhará.
S (Sério):	Se o evento de risco ocorrer, o programa encontrará grandes custos, aumento no cronograma e com chances remotas de recuperação.
Mo (Moderado):	Se o evento de risco ocorrer, o programa encontrará custos aumento no cronograma e com possibilidade de recuperação.
Mi (Menor):	Se o evento de risco ocorrer, o programa encontrará custo, aumento no cronograma. Esses impactos serão mínimos nos objetivos.
N (Negligenciável):	Se o evento de risco ocorrer, ele não terá efeito sobre o programa.

Fonte: [Engert and Lansdowne \(1999\)](#)(adaptada)

### 4.1.2 Requisitos Funcionais

Os requisitos funcionais podem ser compreendidos como tudo aquilo que o sistema deve fazer, ou seja, as funções que o sistema deve fornecer para o cliente e como o sistema se comportará em determinadas situações. [Neponuceno \(2015\)](#). As tabelas seguintes descrevem os requisitos funcionais do sistema.

<b>ID do requisito</b>	RF1
<b>Nome</b>	Autenticação (Login)
<b>Descrição</b>	O sistema deve fornecer serviços de autenticação para permitir que os Administradores possam criar e consultar/editar processos, consultar usuário, gerar relatórios, em relação ao Corretores o sistema deve direcionar apenas os processo cujo estiver cadastrado. Se o login falhar, o usuário deve entrar em contato com o administrador do sistema.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 3: Requisito Funcional 1

<b>ID do requisito</b>	RF2
<b>Nome</b>	Gerenciar Usuários
<b>Descrição</b>	O sistema deve fornecer as opções de: cadastrar, consultar, listar e editar usuários.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 4: Requisito Funcional 2

<b>ID do requisito</b>	RF3
<b>Nome</b>	Upload de arquivos
<b>Descrição</b>	O sistema deve fornecer a opção ao criar o processo selecionar as redações relacionada ao processo.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 5: Requisito Funcional 3

<b>ID do requisito</b>	RF4
<b>Nome</b>	Cadastro de Corretores.
<b>Descrição</b>	Para cadastrar um processo o sistema deve exigir que informe no mínimo de 3 corretores.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 6: Requisito Funcional 4

<b>ID do requisito</b>	RF5
<b>Nome</b>	Distribuir redações.
<b>Descrição</b>	O sistema distribui a redações entre os corretores balanceando a quantidade de redação para cada corretor de forma que nenhum corretor fique sobrecarregado.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 7: Requisito Funcional 5

<b>ID do requisito</b>	RF6
<b>Nome</b>	Correção Redação.
<b>Descrição</b>	O sistema deve permitir que o corretor acesse as redações que estão atribuídas a ele. O corretor deve informar uma nota e uma observação (Opcional). Havendo uma divergência muito alta, a redação deve ser encaminhada para uma terceira correção. Para saber se a divergência entre as notas é alta, é feita da seguinte forma: Se a diferença entre as duas primeiras correções for maior que 20% da nota máxima, a redação irá para a terceira correção. Caso contrário não.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 8: Requisito Funcional 6

<b>ID do requisito</b>	RF7
<b>Nome</b>	Gerenciar Processos.
<b>Descrição</b>	O Sistema deve fornecer as opções de: cadastrar, editar, consultar e listar processos, além de, disponibilizar a transição de etapas.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 9: Requisito Funcional 7

<b>ID do requisito</b>	RF8
<b>Nome</b>	Etapas do Processo
<b>Descrição</b>	O processo de avaliação de redações pode conter no mínimo 2 etapas e no máximo 3. Cada etapa consiste em um corretor avaliando uma nota de uma redação. Se as notas da primeira e segunda avaliação tiver uma divergência alta, o sistema deve disponibilizar ao administrador uma terceira etapa para que um corretor que não corrigiu a esta prova, avalie-a. Se a divergência entre as notas é alta, é feita da seguinte forma: Se a diferença entre as duas primeiras avaliações for maior que 20% da nota máxima, a redação irá para a terceira correção. Caso contrário não.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado.

Tabela 10: Requisito Funcional 8

### 4.1.3 Requisitos Não Funcionais

Os requisitos não-funcionais não estão relacionados diretamente às funcionalidades de um sistema, porém é de grande importância no desenvolvimento do sistema. Tratados geralmente como premissas e restrições técnicas de um projeto os requisitos não funcionais são praticamente todas as necessidades que não podem ser atendidas através de funcionalidades. [Codificar \(2017\)](#). Os requisitos não funcionais do sistema é descrito nas tabelas seguintes.

<b>ID do requisito</b>	RFN1
<b>Nome</b>	Disponibilidade.
<b>Descrição</b>	O sistema deve estar disponível 24 horas por dia, 7 dias por semana. Devido a possibilidade do corretor efetuar a correção fora do horário comercial da universidade.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Sério (S) - O sistema ainda está funcional, mas não poderá ser acessado

Tabela 11: Requisito Não Funcional 1

<b>ID do requisito</b>	RFN2
<b>Nome</b>	Segurança.
<b>Descrição</b>	O sistema deve conter um sistema de autenticação e autorização. As senhas devem ser criptografadas para que seja capaz de obter uma maior sigilo.
<b>Prioridade</b>	Prioridade 1 - Obrigatória
<b>Risco</b>	Crítico (C) - Ele quebrará a principal funcionalidade do sistema. O sistema não pode ser usado se este requisito não for implementado..

Tabela 12: Requisito Não Funcional 2

## 4.2 Projeto

### 4.2.1 Escopo

O escopo deste projeto limita-se a desenvolver um serviço off-line, onde os corretores possam agir de forma confidencial durante suas avaliações, pois atualmente o processo permite que um corretor tenha acesso a resultados de avaliações anteriores de outros corretores e isso não é desejável, visto que, a nota anterior pode influenciar na avaliação. O sistema deverá permitir correções das redações de forma privada, a fim de garantir a imparcialidade.

### 4.2.2 Perspectiva do Produto

Para obter níveis maiores de segurança, será implementado o sistema de Gates disponível no framework Laravel, que determina se um usuário possui autorizações de execuções de ações ou consulta à informações sigilosas. A verificação deste sistema será baseada principalmente no tipo de usuário, no qual será redirecionado somente para a telas que o mesmo esteja autorizado.

### 4.2.3 Características dos Usuários

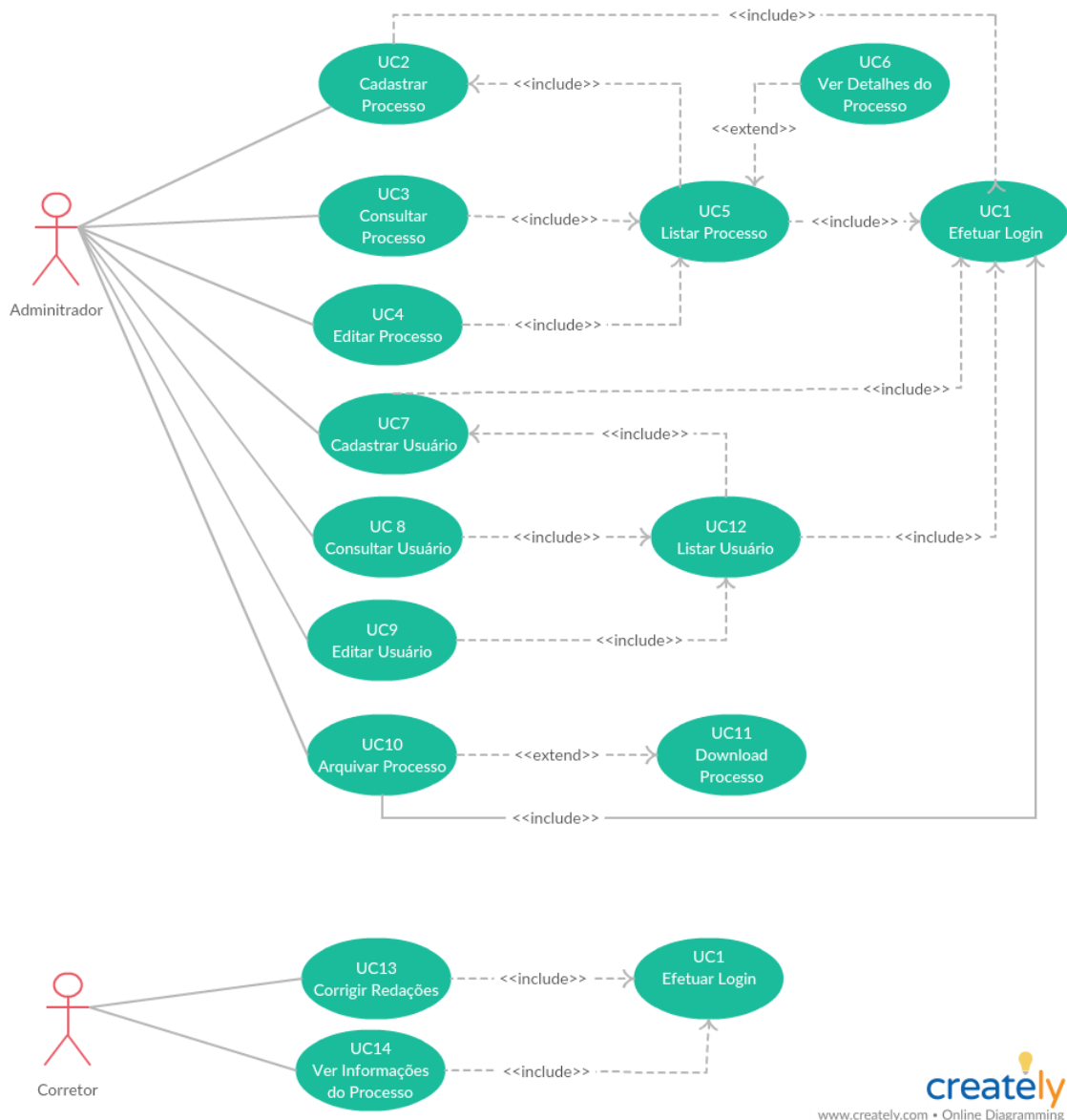
O sistema será desenvolvido visando atender os seguinte tipos de usuários:

- Usuário 1: Administrador. O sistema buscará facilitar a criação de um processo e também a sua consulta sobre o andamento das suas correções. O sistema possui níveis de acesso permitindo apenas o administrador poderá acessar as notas lançadas pelo corretores.
- Usuário 2: Corretores. O sistema facilitará a forma da correção mostrando de forma simples a redação que deverá ser corrigida. O sistema também mostrará quais processos que o corretor está cadastrado e também a quantidade de redações a serem avaliadas.

### 4.2.4 Funções do Produto

O sistema consiste em dois níveis de acesso: Administrador e Corretor. Para utilizar o sistema somente o administrador poderá efetuar o cadastro dos corretores e também de outros administradores. O usuário cadastrado como administrador tem acesso total dos dados no decorrer do processo. Já o corretor terá acesso apenas nos processos que foi cadastrado pelo administrador. As atividades que cada tipo de usuário podem realizar no sistema podem ser observado na figura 12.

Figura 12: Diagrama de Caso de Uso



Fonte: Dados do Autor 2018

## 4.3 Implementação

O desenvolvimento do sistema foi baseado em módulos. Cada módulos consiste em uma atividade que os usuários podem realizar no sistema. Dessa forma as tabelas a seguir, descrevem cada atividade desenvolvida baseado nos casos de uso descrito na figura 12.

### 4.3.1 Casos de uso do ator Administrador

<b>ID do Caso de Uso</b>	UC1
<b>Nome</b>	Efetuar Login no Sistema
<b>Descrição</b>	Detalha as etapas de login de um profissional que é necessário para obter acesso a outras funcionalidades do sistema. Este caso de uso busca atender o requisito funcional da tabela 3.
<b>Ator</b>	Administrador/ Corretor
<b>Precondições</b>	O Profissional deve possuir cadastro no sistema.
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta campos para e-mail e senha.</li> <li>2. O profissional insere seu e-mail.</li> <li>3. O profissional insere sua senha.</li> <li>4. O profissional clica no botão Entrar.</li> <li>5. Se as e-mail e senhas tiverem corretas o sistema apresenta a tela principal.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	<p>E-mail ou senha inválidos:</p> <ol style="list-style-type: none"> <li>a) Se o e-mail do profissional e/ou sua senha não puderem ser localizados, o sistema emite uma mensagem de erro.</li> <li>b) Para efetuar um cadastro apenas o administrador pode efetuar essa ação.</li> </ol>
<b>Pós-condições</b>	Usuário logado
<b>Notas e Problemas</b>	1. idioma padrão do sistema é o português.

Tabela 13: Caso de Uso 1



<b>ID do Caso de Uso</b>	UC 2
<b>Nome</b>	Cadastrar Processo
<b>Descrição</b>	Descreve como o administrador pode cadastrar um novo processo. Este caso de uso atende parcialmente o RF7 e atendendo completamente o RF3 e RF4.
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Administrador logado e autenticado.</li> <li>• Corretores já cadastrados.</li> </ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta os campos a serem preenchido: Nome do processo; Número do Edital; Nota máxima; Data de início do processo; Data de término do processo; Quantidade de corretores a participarem do processo; Selecionar quais corretores participaram do processo; Quantidade de redações a serem corrigidas; Upload de redações;</li> <li>2. O profissional clica no botão Cadastrar.</li> <li>3. Processo cadastrado com sucesso.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. Caso os seguintes critérios não forem atendidas os sistema notificar o administrador com um alerta. <ol style="list-style-type: none"> <li>a) Todos os campos são obrigatórios.</li> <li>b) Nome do processo com no mínimo 3 caracteres.</li> <li>c) Nome do processo com no mínimo 100 caracteres.</li> <li>d) Nota máxima deve ser do tipo numérico.</li> <li>e) Data fim deve ter um período posterior ao data Início.</li> <li>f) Quantidade de corretores deve ser do tipo numérico.</li> <li>g) A quantidade mínima de corretores é 3.</li> <li>h) Quantidade de Redações deve ser do tipo numérico.</li> </ol> </li> <li>2. Caso houver alguma divergência de dados como na quantidade de corretores selecionados for diferente da quantidade que foi preenchida no campo f, ou se as quantidade de redações se divergirem não será efetuado o cadastro do processo.</li> <li>3. Caso o sistema já houver um processo com mesmo número do edital o sistema não efetuará o cadastro do processo.</li> </ol>
<b>Pós-condições</b>	Processo cadastrado com sucesso
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 14: Caso de Uso 2

<b>ID do Caso de Uso</b>	UC3
<b>Nome</b>	Consultar processo.
<b>Descrição</b>	Somente o administrador tem autorização para visualizar os dados relacionado ao processo. Este caso de uso atende parcialmente o <a href="#">RF7</a> .
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Administrador logado e autenticado.</li> <li>• Haver pelo um processo já cadastrado.</li> </ul>
<b>Fluxo Normal</b>	O sistema informa ao usuários, os nomes dos corretores associado ao processo, e mostra quantas redações faltam a serem corrigidas de cada corretor.
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Não possui.
<b>Pós-condições</b>	Consulta do processo.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 15: Caso de Uso 3

<b>ID do Caso de Uso</b>	UC 4
<b>Nome</b>	Editar processo.
<b>Descrição</b>	Somente o administrador tem autorização para editar as informações do processo. Este caso de uso atende parcialmente o <a href="#">RF7</a> .
<b>Ator</b>	Administrador
<b>Precondições</b>	Administrador logado e autenticado.
<b>Fluxo Normal</b>	O sistema apresenta apenas o seguinte campo a ser editado: Data de término do processo;
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	O sistema não aceita data anterior a data de cadastro do processo.
<b>Pós-condições</b>	Consulta ou arquivamento do processo.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 16: Caso de Uso 4

<b>ID do Caso de Uso</b>	UC 5
<b>Nome</b>	Listar processo
<b>Descrição</b>	Somente o administrador tem autorização para visualizar as listagem de processos. Este caso de uso atende parcialmente o RF7.
<b>Ator</b>	Administrador
<b>Precondições</b>	O sistema disponibiliza a listagem de duas formas: <ul style="list-style-type: none"> <li>• Padrão: O sistema mostra a listagem de todos os processos em andamento.</li> <li>• Pesquisa: O sistema disponibiliza um campo para a filtragem desse dados.</li> </ul>
<b>Fluxo Normal</b>	A listagem pode ser feita através de um desses dois passos: <ol style="list-style-type: none"> <li>1. Ao acessar a página, o sistema já informa todos os processos abertos.</li> <li>2. Ao utilizar o campo de pesquisa, pode efetuar um filtro que é baseado no número do edital do processo.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Caso não houver dados referente ao dado pesquisado, o sistema mostrará que não possui processos com esse número de edital.
<b>Pós-condições</b>	Listagem dos processos.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 17: Caso de Uso 5

<b>ID do Caso de Uso</b>	UC 6
<b>Nome</b>	Detalhes do processo
<b>Descrição</b>	O sistema disponibiliza todas as informações sobre o processo. Inclusive os nome de cada redação, notas, observações e nomes dos corretores. Este caso de uso atende parcialmente o RF7.
<b>Ator</b>	Administrador
<b>Precondições</b>	Haver processos cadastrados.
<b>Fluxo Normal</b>	O sistema apresenta as seguintes informações do processo: Nome da redação; Nota 1; Nota 2; Nota 3; Média; Botão consultar; Ao clicar no botão consultar o sistema disponibilizará as observações de cada corretor bem como a redação em si.
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Nenhum
<b>Pós-condições</b>	Consulta de informações específicas de cada redações.
<b>Notas e Problemas</b>	Nenhum.

Tabela 18: Caso de Uso 6

<b>ID do Caso de Uso</b>	UC 7
<b>Nome</b>	Cadastro de usuário
<b>Descrição</b>	Somente o administrador tem autorização para realizar o cadastro de novos usuários. Este caso de uso atende parcialmente o RF2.
<b>Ator</b>	Administrador
<b>Precondições</b>	Administrador logado e autenticado.
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta os campos a serem preenchidos: Nome do usuário; E-mail do usuário; Celular; SIAP; Data de nascimento; Tipo de usuário; Senha; Confirmar senha;</li> <li>2. O administrador clica no botão de cadastro.</li> <li>3. Usuário cadastrado com sucesso.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. Caso os seguintes critérios não forem atendidas os sistema irá notificar o administrador com um alerta: <ol style="list-style-type: none"> <li>a) Todos os campos são obrigatórios..</li> <li>b) Nome com mínimo 3 caracteres.</li> <li>c) Nome no máximo 100 caracteres</li> <li>d) Senha com mínimo 6 caracteres.</li> <li>e) Senha com no máximo 20 caracteres.</li> <li>f) SIAP tem que ser do tipo numérico e com no mínimo de 3 caracteres e no máximo 6.</li> <li>g) Celular tem que ser do tipo numérico.</li> </ol> </li> </ol>
<b>Pós-condições</b>	Usuário cadastrado terá acesso ao sistema.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema. .

Tabela 19: Caso de Uso 7

<b>ID do Caso de Uso</b>	UC 8
<b>Nome</b>	Consulta de usuário
<b>Descrição</b>	Somente o administrador tem autorização para efetuar a consulta de informações sobre os usuários. Este caso de uso atende parcialmente o <a href="#">RF2</a> .
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Administrador logado e autenticado no sistema.</li> <li>• O usuário ter cadastro no sistema.</li> </ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta as seguintes informações dos usuários: Nome; SIAP; Email; Celular; Tipo de usuário; Botão editar;</li> <li>2. Acesso a informações sobre os usuários.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Não possui exceções
<b>Pós-condições</b>	Consulta de informações sobre os usuários.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 20: Caso de Uso 8

<b>ID do Caso de Uso</b>	UC 9
<b>Nome</b>	Editar usuário
<b>Descrição</b>	Somente o administrador tem autorização para efetuar a edição de dados referente ao usuários cadastrados. Este caso de uso atende parcialmente o RF2.
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"><li>• Administrador logado e autenticado no sistema.</li><li>• O usuário ter cadastro no sistema.</li></ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"><li>1. O sistema apresenta as seguintes informações dos usuários: Nome; SIAP; Email; Celular; Tipo de usuário; Botão editar;</li><li>2. Neste botão o administrador pode alterar qualquer informação exceto a senha.</li><li>3. Alterações de dados dos usuários.</li></ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Não possui exceções
<b>Pós-condições</b>	Edição de dados dos os usuários.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 21: Caso de Uso 9

<b>ID do Caso de Uso</b>	UC10
<b>Nome</b>	Arquivar processo
<b>Descrição</b>	Mostra os processos que já foram concluídos. Informando todas os dados sobre o processo. Inclusive os nome de cada redação, notas e nomes dos corretores.
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Administrador logado e autenticado no sistema.</li> <li>• Processo deve estar concluído.</li> </ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta as seguintes informações do processo: Número do Edital; Nome; Quantidade de redações que o processo possui; Etapa : Etapa atual; Data de início do processo; Data de término do processo; Botão detalhes; Botão download; Botão deletar;</li> <li>2. Ao clicar nesse botão detalhes irá para o caso de uso UC6.</li> <li>3. Ao clicar nesse botão download irá para o caso de uso UC11.</li> <li>4. Ao clicar neste botão deletar, os sistema irá deletar todas as informações sobre o processo.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Ao clicar no botão Deletar é necessário que o processo esteja concluído e deve digitar a senha do usuário logado novamente.
<b>Pós-condições</b>	1. Ao clicar no botão Deletar é necessário que o processo esteja concluído e será necessário digitar a senha do usuário logado.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 22: Caso de Uso 10

<b>ID do Caso de Uso</b>	UC11
<b>Nome</b>	Download do processo
<b>Descrição</b>	O sistema disponibiliza o download de todas as informações do processo em um arquivo de texto.
<b>Ator</b>	Administrador
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Administrador logado e autenticado no sistema.</li> <li>• Processo deve estar concluído.</li> </ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. O arquivo de download, consta as seguinte informações do processo: Nome do processo; Nota 1; Nota 2; Nota 3; Média; Data início; Data Término; Matrícula da redação.;</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Nenhum
<b>Pós-condições</b>	Arquivo com as informações sobre o processo é baixado.
<b>Notas e Problemas</b>	Nenhum

Tabela 23: Caso de Uso 11

<b>ID do Caso de Uso</b>	UC 12
<b>Nome</b>	Listar usuários
<b>Descrição</b>	Somente o administrador tem autorização para realizar visualização da lista de usuários. Este caso de uso atende parcialmente o <a href="#">RF2</a> .
<b>Ator</b>	Administrador
<b>Precondições</b>	O sistema disponibiliza a listagem de duas formas: <ul style="list-style-type: none"> <li>• Padrão: O sistema mostra a listagem de todos os usuários cadastrado.</li> <li>• Pesquisa: O sistema disponibiliza um campo para a filtragem desse dados.</li> </ul>
<b>Fluxo Normal</b>	A listagem pode ser feita através de um desses dois passos: <ol style="list-style-type: none"> <li>1. Ao acessar a página, o sistema já informa todos os usuários cadastrados.</li> <li>2. Ao utilizar o campo de pesquisa, pode efetuar um filtro que é baseado no número de matrícula do usuário.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	Caso não houver dados referente ao dado pesquisado, o sistema mostrará que não possui usuários com esse número de matrícula.
<b>Pós-condições</b>	Listagem dos usuários.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 24: Caso de Uso 12



### 4.3.2 Corretor

<b>ID do Caso de Uso</b>	UC 13
<b>Nome</b>	Corrigir Redações
<b>Descrição</b>	O sistema disponibiliza os campos para os corretores realizarem a avaliação das redações. Este caso de uso atende o <a href="#">RF6</a> .
<b>Ator</b>	Corretor
<b>Precondições</b>	<ul style="list-style-type: none"> <li>• Usuário tenha sido cadastrado em algum processo.</li> <li>• Exista redações que ainda não foram corrigidas.</li> </ul>
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. Mostra as seguintes campos a serem preenchidos: Observação :Justificativa do corretor para dar a nota; Nota;</li> <li>2. O sistema disponibiliza a redação a ser corrigida e oferece a opção de alteração do tamanho da imagem para melhorar a visualização do corretor.</li> </ol>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. Mostrará notificação caso o usuário não tenha sido cadastrado em nenhum processo.</li> <li>2. Caso a nota que o corretor der for maior que a nota máxima cadastrada ao criar o processo mostrará um alerta.</li> </ol>
<b>Pós-condições</b>	Redação avaliada.
<b>Notas e Problemas</b>	As inconsistências serão mostradas nos alertas do sistema.

Tabela 25: Caso de Uso 13

<b>ID do Caso de Uso</b>	UC 14
<b>Nome</b>	Informações do Processo
<b>Descrição</b>	O sistema disponibiliza ao corretor informações sobre os processos que está cadastrado.
<b>Ator</b>	Corretor
<b>Precondições</b>	Usuário tenha sido cadastrado em algum processo.
<b>Fluxo Normal</b>	<ol style="list-style-type: none"> <li>1. Mostra as seguintes informações do processo que o corretor está cadastrado: Nome do processo; Etapa do processo; Número do Edital; Data de Início do processo; Data de término do processo; Quantidade de redações que falta o corretor corrigir de cada processo; Botão Corrigir redações:</li> <li>2. Se o corretor clicar no botão Corrigir redações o usuário irá para o caso de uso UC13.</li> </ol> <p>1. Irá para o caso de uso UC10</p>
<b>Fluxo Alternativo</b>	Não possui fluxo alternativo.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. Mostrará notificação caso o usuário não tenha sido cadastrado em nenhum processo.</li> </ol>
<b>Pós-condições</b>	Consulta de informações do corretor.
<b>Notas e Problemas</b>	Nenhum.

Tabela 26: Caso de Uso 14

## 4.4 Testes

Os teste e correções compõem a parte iterativa desta metodologia. O teste unitário, integração e de usuário, foram realizados após o desenvolvimento de cada módulo, sendo aplicados da seguinte forma:

Primeiro é realizado o teste unitário, que consiste em validar os dados de entrada e saída. Depois é realizado o teste de integração, que visa em integrar os módulos desenvolvidos ao sistema. Após o teste de integração é realizado o teste com usuário, que nada mais é do que os próprios usuários utilizando o sistema, desta forma é possível obter feedback para seja feito as correções do sistema.

Durante o desenvolvimento foi necessário constante acompanhamento, executando vários teste, e a implementação de pequenos ajustes.

## **4.5 Implantação**

O sistema foi hospedado em um computador com sistema operacional Ubuntu na versão 18.04, e como servidor foi utilizado o Apache na versão 2.4 . A configuração do servidor local foi realizada, criando um arquivo de configurações de Virtual Host para que o sistema seja capaz de aceitar requisições na porta 80. Desta forma é possível acessar o sistema pelo IP do servidor.

Após o período de instalação do sistema, foi realizado o treinamento dos usuários, que consistia em realizar todas as atividades descritas na seção 4.3.

## 5 APRESENTAÇÃO DO SISTEMA

Nesta capítulo são apresentadas as principais telas do sistema. As telas estão divididas baseada nos tipos de usuários, sendo assim, administrador e corretor. As demais telas do sistema estarão contidas em anexo.

### 5.1 Telas do Administrador

Após ter feito o login através do navegador o usuário é capaz de efetuar o cadastro de um novo processo seletivo. Nesta tela apresenta alguns campos que devem ser preenchidos. O [caso de uso 2](#) apresenta mais informações sobre esta tela.

Figura 13: Cadastrar Processo

A imagem mostra a interface de usuário para a criação de um processo seletivo. O cabeçalho indica 'COPESE' e o perfil do usuário 'adm'. O título da página é 'Criar Processo'. Há links para 'Home' e 'Criar Processo'. Abaixo, há botões para 'Voltar' e 'Atualizar'. O formulário contém os seguintes campos:

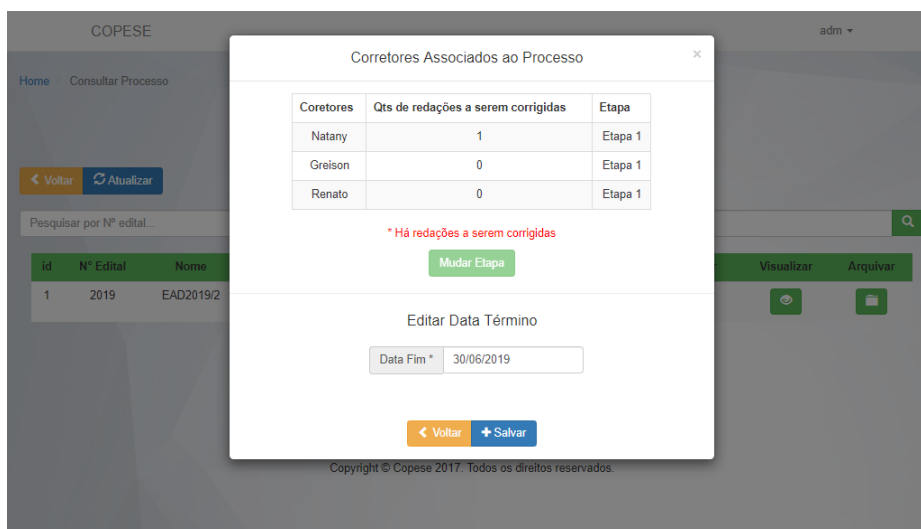
- Nome\* (Nome do processo)
- N° Edital\* (Número do edital)
- Nota máxima\* (Nota máxima)
- Data Início\* (dd/mm/aaaa)
- Data Término\* (dd/mm/aaaa)
- Corretores\* (Quantidade de corretores)
- Quantidade de Redações\* (Quant. de Re)
- Corretores\* (Nothing selected)
- Escolher Ficheiros (Nenhum ficheiro selecionado)

Um botão 'Gravar' está localizado na parte inferior direita do formulário.

Fonte: Dados do Autor 2018

Ao efetuar uma consulta de um processo, o sistema apresenta esta tela ao administrador informações sobre andamento do processo. É possível verificar quantas redações faltam a serem avaliadas de cada corretor como descrito no [caso de uso 3](#), bem como efetuar a edição da data de término do processo detalhado no [caso de uso 4](#).

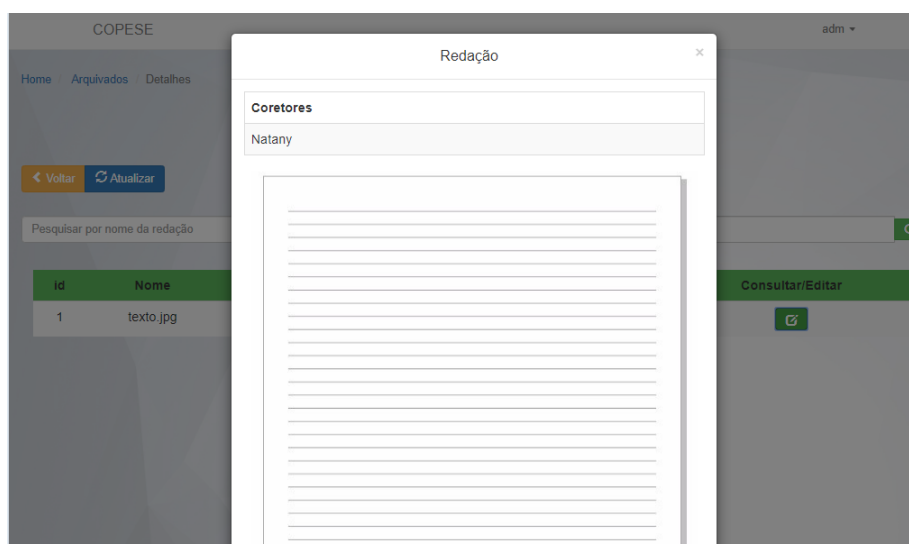
Figura 14: Consultar Processo



Fonte: Dados do Autor 2018

Após efetuar uma consulta de um processo, o administrador é capaz de optar por visualizar informações específicas sobre cada redação. O [caso de uso 6](#) descreve melhor sobre essas informações.

Figura 15: Visualização Redação - Admin



Fonte: Dados do Autor 2018

## 5.2 Telas do Corretor

Após o usuário do tipo corretor realizar o login no sistema, a seguinte tela será apresentada ao usuário. Esta tela apresenta ao usuário informações dos processo no qual foi cadastrado. O [caso de uso 14](#) especifica as informações sobre esta tela.

Figura 16: Tela inicial corretor



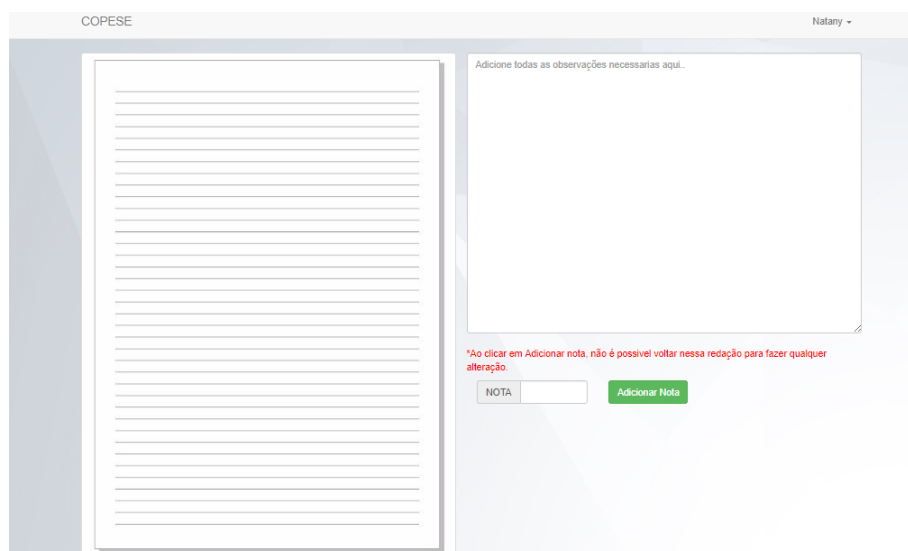
id	Nome	Etapa	Nº Edital	Data/Início	Data/Término	Qtd. Redações	Redações
1	EAD2019/2	Etapa 1	2019	2019-06-01	2019-06-30	1	

Copyright © Copese 2017. Todos os direitos reservados.

Fonte: Dados do Autor 2018

O usuário do tipo corretor é capaz de avaliar as redações. A seguinte figura apresenta a tela, cujos campos que devem ser preenchido bem como a redação em si. O [caso de uso 13](#) detalha melhor sobre essas informações.

Figura 17: Avaliação de Redação



Adicione todas as observações necessárias aqui.

\*Ao clicar em Adicionar nota, não é possível voltar nessa redação para fazer qualquer alteração

NOTA

Fonte: Dados do Autor 2018

## 6 ANÁLISE DOS RESULTADOS

Neste capítulo deseja-se verificar se após a utilização do sistema para o gerenciamento de avaliação das redações, houve alguma diferença significativa nas notas atribuídas pelos avaliadores. Para a realização desta análise foram utilizadas dois de grupos, sendo o primeiro grupo composto pelos processos: EAD 2014, EAD 2016, LEC 2015, LEC 2016, LEC 2017. Enquanto o segundo: EAD 2019 e LEC 2019. É importante destacar que os processos referentes ao primeiro grupo ocorreram antes do ano de 2019 e portanto não utilizaram o sistema desenvolvido, por outro lado as avaliações de redações do segundo grupo foram avaliadas utilizando o sistema proposto neste trabalho. Desta forma a análise dos dados será feita em duas amostras, descritas da seguinte maneira:

- Data\_s contém as diferença das notas das do primeiro grupo, para cada redação.
- Data\_c contém as diferença das notas das do segundo grupo, para cada redação.

Alguns cálculos realizados das amostras são demonstrados na tabela 27.

Amostra	Média	Desvio Padrão	1º Quartil	3º Quartil	Tamanho
Data_s	1.088014	0.927876	0	2	1798
Data_c	4.140069	3.161651	1.5	6	579

Tabela 27: Visão Geral das Amostras

Observando que:

$$\bar{x}_c - \bar{x}_s = 3,052055$$

Ao analisar apenas a diferença entre as médias das duas amostras, os resultados sugerem uma diferença entre os grupos. Porém verificar se os dados amostrais trazem evidências que apoiem a hipótese de existência de uma diferença significativa entre as médias, e assim verificar o efeito a utilização do sistema, propõe-se o seguinte teste de hipótese:

$$\begin{cases} H_0 : \mu_1 - \mu_2 = 0 \\ H_1 : \mu_1 - \mu_2 \neq 0 \end{cases}$$

Sendo assim, a hipótese nula ( $H_0$ ) indica que não existe diferença entre os resultados das avaliações realizadas pela metodologia antiga (manual) e pelo sistema proposto neste trabalho (automatizada), já hipótese alternativa ( $H_1$ ) indica que houve diferenças entre os resultados.

Para decidir qual teste de hipótese utilizar, é necessário verificar se as amostras possuem distribuição normal.

## 6.1 Normalidade

Para o teste de normalidade da amostra coletada, foi utilizado o teste de Shapiro-Wilk. A hipótese nula deste teste indica que a população possui distribuição normal. Portanto, um valor de  $p < 0,05$  indica que você rejeitou a hipótese nula, ou seja seus dados não possui distribuição normal, utilizando o  $\alpha = 0,05$  e o resultado obtido foi o seguinte:

Figura 18: Teste de Shapiro-Wilk da amostras Data\_s

```
Shapiro-Wilk normality test
data:  Data_s
W = 0.7909, p-value < 2.2e-16
```

Fonte: Dados do Autor 2018

Figura 19: Teste de Shapiro-Wilk da amostras Data\_c

```
Shapiro-Wilk normality test
data:  Data_c
W = 0.93273, p-value = 1.806e-15
```

Fonte: Dados do Autor 2018

Desta forma, ambas as populações não possuem população com distribuição normal, ou seja conseguimos rejeitar a hipótese nula das duas amostras.

## 6.2 Teste de Tendência Central

O Teste de Wilcoxon ou Mann-Whitney, é um teste que afirma se a mediana da variável estudada é igual nas duas amostras. A Hipótese Nula ( $H_0$ ) deste teste afirma que mediana da variável estudada é igual nas duas amostras, enquanto a Hipótese Alternativa ( $H_1$ ) indica que a mediana da variável é diferente entre as amostras. O resultado do teste aplicado é apresentado na figura 20.



Figura 20: Resultado do teste de Mann-Whitney

```
Wilcoxon rank sum test with continuity correction

data: diferenca by info
W = 188332, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Fonte: Próprio autor

Como  $\rho < 0,05$ , rejeitamos a hipótese nula. O teste fornece evidência de que existe uma diferença nas entre as notas atribuídas a uma redação pelos avaliadores.

## 7 CONCLUSÃO

Este trabalho relata o processo de desenvolvimento de um sistema para o gerenciamento de correção de redações da COPESE. O sistema resultante, apresenta maior sigilo além de auxiliar o trabalho dos corretores e servidores que são responsáveis por participar dos processos seletivos.

O sistema permite ao administrador tenha uma visão geral dos processos, sendo capaz de verificar o andamento da avaliações de cada corretor, ou até mesmo visualizar as redações específicas juntamente com as notas atribuídas pelos corretores. Os corretores, por sua vez, têm a possibilidade de consultar relatórios sobre quais processos seletivos estão cadastrados ou verificar quantas redações lhe faltam ser avaliadas, além disso é possível utilizar o sistema para efetuar a sua avaliação. Desta forma o sistema propõe maior sigilo devido que o segundo corretor não ser capaz de visualizar a nota do avaliador anterior.

Afim de verificar se houve diferenças significativas nas notas atribuídas pelos corretores, foi aplicado método estatístico Mann-whitney. O resultado deste teste indica que há diferenças nas notas atribuídas pelos corretores.

O sistema desenvolvido atendeu as necessidades da COPESE em relação ao sigilo no processo de correção das provas de redação. Futuramente, novos módulos podem ser adicionados, ou até mesmo modificações no cálculo das divergências das redações, afim de obter melhores resultados.

## REFERÊNCIAS

- Alberto Luiz Albertin. Valor estratégico dos projetos de tecnologia de informação. *Revista de Administração de Empresas*, 41(3):42–50, 2001. Citado na página 5.
- Alberto Luiz Albertin and Rosa Maria de Moura. Administração de informática e seus fatores críticos de sucesso no setor bancário privado nacional. *Revista de Administração de Empresas*, 35(5):22–29, 1995. Citado na página 5.
- Emerson Alecrim. Requisições síncronas e assíncronas, 2006. URL <https://www.infowester.com/postgremysql.php>. Acessado: 25 jun. 2019. Citado na página 17.
- Emerson Alecrim. Bancos de dados são mais importantes nas nossas vidas do que a gente imagina, 2018. URL <https://tecnoblog.net/245120/banco-de-dados-importancia/>. Acessado: 11 jul. 2019. Citado na página 7.
- Henrique Bastos. Diferenças entre linguagem compilada e linguagem interpretada, 2008. URL [https://henriquebastos.net/pt\\_br/diferencas-entre-linguagem-compilada-e-linguagem-interpretada/](https://henriquebastos.net/pt_br/diferencas-entre-linguagem-compilada-e-linguagem-interpretada/). Acessado: 03 jun. 2019. Citado na página 8.
- Project Builder. Guia da gestão de projetos: metodologia waterfall, 2017. URL <https://www.projectbuilder.com.br/blog/guia-da-gestao-de-projetos-metodologia-waterfall/>. Acessado: 10 jul. 2019. Citado na página 19.
- Codificar. O que são requisitos funcionais e requisitos não funcionais?, 2017. URL <https://codificar.com.br/blog/requisitos-funcionais-nao-funcionais/>. Acessado: 25 jun. 2019. Citado na página 24.
- Valéria Martins da Silva. Revisão sistemática da evolução mvc na base acm. 2012. Citado na página 10.
- Edson José Dionisio. Postgresql tutorial, 2015. URL <https://www.devmedia.com.br/space/edson-jose-dionisio>. Acessado: 25 jun. 2019. Citado na página 17.
- Ramez Elmasri and Shamkant B Navathe. Sistemas de banco de dados. 6a edição. *São Paulo: Person*. 837p, page 24, 2011. Citado na página 7.
- Pamela A Engert and Zachary F Lansdowne. Risk matrix user’s guide. *Bedford, MA: The MITRE Corporation*, 1999. Citado na página 22.

- Behrouz A Forouzan. *Comunicação de dados e redes de computadores*. AMGH Editora, 2009. Citado na página 7.
- Sérgio Furgeri. O papel das linguagens de marcação para a ciência da informação. *TransInformação*, 18(3), 2006. Citado na página 9.
- Henrique Machado Gasparotto. Os 4 pilares da programação orientada a objetos, 2014. URL <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>. Acessado: 11 jul. 2019. Citado na página 8.
- Pedro César Tebaldi Gomes Gomes. Quais os principais bancos de dados e quais suas diferenças?, 2019. URL <https://www.opservices.com.br/banco-de-dados/>. Acessado: 11 jul. 2019. Citado na página 7.
- Leandro Salenave Gonçalves. *Sistema de informação*. São Paulo: Atlas, 2009. Citado na página 5.
- Renato Groffe. Bancos de dados nosql: uma visão geral, 2016. URL <https://imasters.com.br/banco-de-dados/bancos-de-dados-nosql-uma-visao-geral>. Acessado: 25 jun. 2019. Citado na página 7.
- Ricardo R Gudwin. *Linguagens de programação*. Campinas: DCA/FEEC/UNICAMP, page 24, 1997. Citado na página 8.
- Leandro Soares Indrusiak. *Linguagem java*. Grupo JavaRS JUG Rio Grande do Sul, 1996. Citado na página 8.
- Rafael Jaques. O que é um framework? para que serve?, 2016. URL <http://www.phpit.com.br/artigos/o-que-e-um-framework.phpit>. Acessado: 25 jun. 2019. Citado na página 10.
- Eduardo Cavalcante Lacerda. *Sistemas de controle de versão*, 2012. URL <https://www.devmedia.com.br/sistemas-de-controle-de-versao/24574>. Acessado: 25 jun. 2019. Citado 2 vezes nas páginas 10 e 11.
- Kenneth Craig Laudon and Jane Price Laudon. *Sistemas de informação gerenciais*. Pearson Educación, 2007. Citado 3 vezes nas páginas 5, 6 e 7.
- Naisses Zoia Lima. *Desenvolvimento de um framework para métodos sem malha*. 2011. Citado na página 9.
- JOSUÉ LUCIANO and WALLISON JOEL BARBERÁ ALVES. *Padrão de arquitetura mvc: Model-viewcontroller*, 2017. Citado na página 10.

Isaac Rosa Marques and Heimar de Fátima Marin. Enfermagem na web: o processo de criação e validação de um web site sobre doença arterial coronariana. *Revista Latino-americana de Enfermagem*, 2002. Citado na página 7.

Elaine Martins. O que é world wide web?, 2008. URL <https://www.tecmundo.com.br/web/759-o-que-e-world-wide-web-.htm>. Acessado: 25 jun. 2019. Citado na página 6.

Diogo Matheus. Requisições síncronas e assíncronas, 2013. URL <http://www.diogomatheus.com.br/blog/php/requisicoes-sincronas-e-assincronas/>. Acessado: 25 jun. 2019. Citado na página 16.

Higor Medeiros. Introdução a requisitos de software, 2013. URL <https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>. Acessado: 25 jun. 2019. Citado na página 21.

Marco Medina and Cristina Ferting. *Algoritmos e programação: teoria e prática*. Novatec Editora, 2006. Citado na página 8.

Raquel Dias Mendes. Inteligência artificial: sistemas especialistas no gerenciamento da informação. *Ciência da Informação*, 26(1), 1997. Citado na página 6.

Benedito Germano Neponuceno. Modelos de processos de softwares, 2015. URL <http://bpm.bgnweb.com.br/2015/06/26/modelos-de-processos-de-sofwares/>. Acessado: 25 jun. 2019. Citado 2 vezes nas páginas 21 e 22.

Allan Ramos. Controle de versão, git, github e bitbucket- afinal, o que é tudo isso ?, 2017. URL <https://medium.com/trainingcenter/controle-de-vers~ao-git-github-e-bitbucket-afinal-o-que-é-tudo-isso-9fa13fc1330>. Acessado: 03 jun. 2019. Citado na página 10.

Denis Alcides Rezende. *Sistemas de informações organizacionais: Guia Prático para Planejar a Tecnologia da Informação Intefrada ao Planejamento Estratégico das Organizações*. 2011. Citado na página 5.

Adolfo Henrique Schneider. Desenvolvimento web com client side rendering: combinando single page application e serviços de backend. 2016. Citado na página 9.

Tomas Anderson Souza Silva. Modelos lineares de processo, 2016. URL <http://tassinfo.com.br/gestao-de-produto/modelos-lineares-de-processo/>. Acessado: 10 jul. 2019. Citado na página 19.

Alice Thomaz. Modelos lineares de processo, 2017. URL <https://uvagpclass.wordpress.com/2017/08/31/modelo-cascata-x-modelo-agil/>. Acessado: 10 jul. 2019. Citado na página 19.

Leonardo Vilarinho. Mvc - framework: usando a arquitetura sem código de terceiros, 2018. URL <https://medium.com/trainingcenter/mvc-framework-usando-a-arquitetura-sem-código-de-terceiros-bf95a744c66d>. Acessado: 25 jun. 2019. Citado na página 10.

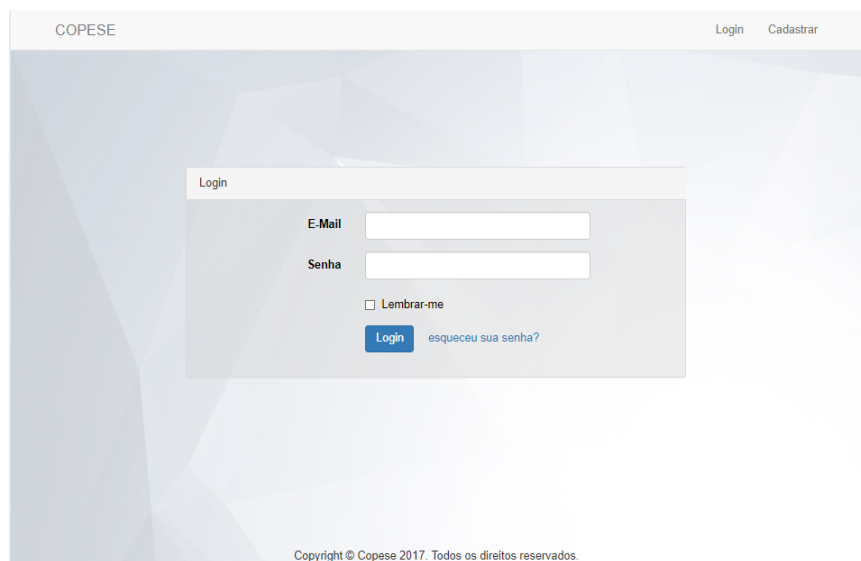
Marco Winckler and Marcelo Soares Pimenta. Avaliação de usabilidade de sites web. *ESCOLA REGIONAL de Informática. Porto Alegre: SBC*, pages 1–54, 2002. Citado na página 9.

# Anexos

# ANEXO A – TELAS DO SISTEMA

Tela de login, permite que os usuários cadastrado acessem o sistema. O caso de uso 1 apresenta mais informações sobre esta tela.

Figura 21: Login



A imagem mostra a interface de login do sistema COPESE. No topo da página, o nome 'COPESE' está à esquerda e os links 'Login' e 'Cadastrar' estão à direita. O formulário de login é centralizado e contém os seguintes elementos: um campo de entrada para 'E-Mail', um campo de entrada para 'Senha', uma caixa de seleção desativada para 'Lembrar-me', um botão azul 'Login' e um link azul 'esqueceu sua senha?'. No rodapé da página, há o texto 'Copyright © Copese 2017. Todos os direitos reservados.'

Fonte: Dados do Autor 2018

Tela principal do administrador, apresenta as principais atividades que o administrador pode realizar.



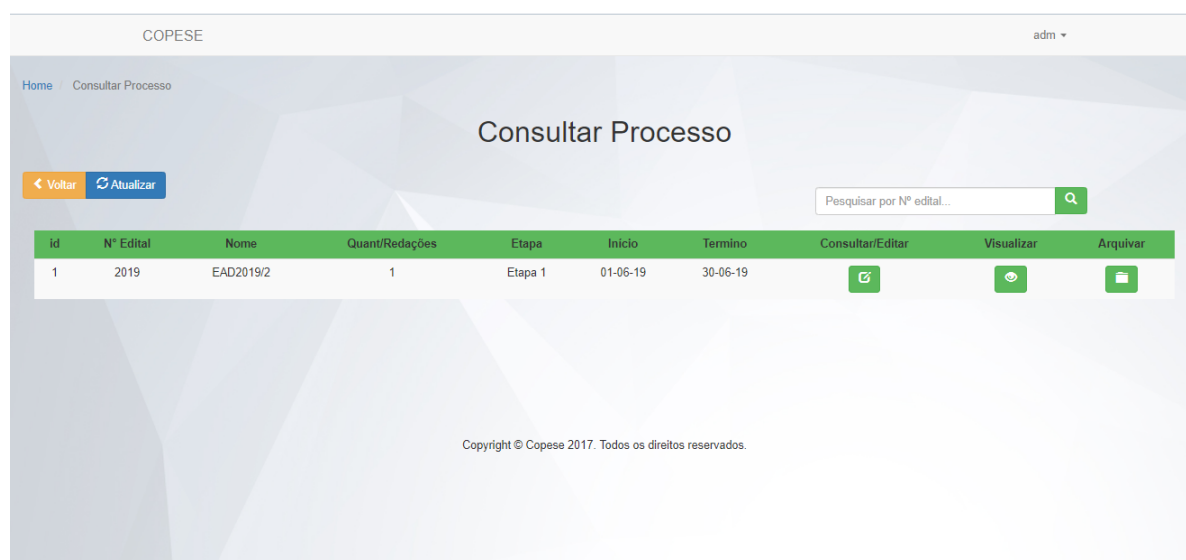
Figura 22: Tela Principal Administrador



Fonte: Dados do Autor 2018

Esta tela permite que administrador efetue uma consulta de processos, baseado por um campo de entrada, sendo ele, o número do edital.

Figura 23: Consulta Processo - Número Edital



Fonte: Dados do Autor 2018

As figuras 24, 25 e 26, permite ao administrador cadastrar, consultar e editar usuários respectivamente.

Figura 24: Cadastro Usuário

COPESE adm

Home Gestão de Usuários Cadastrar

### Cadastro de Usuário

Nome \* Nome do Usuário

email \* e-mail do Usuário Celular \* Celular do Usuário

Siap \* Nº de SIAP do Usuário Data de Nascimento \* dd/mm/aaaa

Tipo de Funcionário \* Seledone... Senha \* Senha de acesso. Mínimo 6 dígitos

Confirmar Senha \* Senha de acesso. Minir

< Voltar + Cadastrar

Fonte: Dados do Autor 2018

Figura 25: Consulta Usuário

### Consultar Usuários

< Voltar Atualizar

Pesquisar por Nº SIAP... 🔍

Nome	Siape	Email	Celular	Tipo	Editar	Mudar status
Greison	110	gleison@ufvjm.com	35321200	corretor	✎	ativo ▾
Renato	11	renato@ufvjm.com	35321200	corretor	✎	ativo ▾
adm	2018	copese@ufvjm.com	35321200	administrador	✎	ativo ▾
Natany	10	naty@ufvjm.com	35321200	corretor	✎	ativo ▾

Fonte: Dados do Autor 2018

Figura 26: Editar Usuário

Nome	ID	email	Celular	Função	Status
adm	2018	copese@ufvjm.com	35321200	administrador	ativo
Natany	10	naty@ufvjm.com	35321200	corretor	ativo

Fonte: Dados do Autor 2018

Esta tela permite ao administrador efetuar o arquivamento do processo após o término do mesmo.

Figura 27: Arquivar processo

Id	Nº Edital	Nome	Quant/Redações	Etapa	Início	Término	Consultar/Editar	Visualizar	Arquivar
1	2019	EAD2019/2	1	Etapa 1	01-06-19	30-06-19			

Copyright © Copese 2017. Todos os direitos reservados.

Fonte: Dados do Autor 2018

Esta tela permite ao administrador efetuar o download de todos os dados do processos.

Figura 28: Download do Processo

COPESE adm

Home / Arquivados

## Processos arquivados

[Voltar](#) [Atualizar](#)

Pesquisar por N° edital...

id	N° Edital	Nome	Quantidade	Etapa	Distribuido	Data/Inicio	Data/Termino	Detalhes	Download	Deletar
1	2019	EAD2019/2	1	Etapa 1	Sim	2019-06-01	2019-06-30			

Copyright © Copese 2017. Todos os direitos reservados.

Fonte: Dados do Autor 2018