

INSTITUTO FEDERAL DE MINAS GERAIS

CAMPUS SÃO JOÃO EVANGELISTA

DAYANE COSTA DOS SANTOS; VANDER CARDOSO DE MACEDO

**PROPOSIÇÃO DE UMA PLATAFORMA WEB PARA CORREÇÃO AUTOMÁTICA
DE PROVAS DO MICROSOFT WORD 2010 COM BASE NAS EMENTAS DE
INFORMÁTICA BÁSICA DO IFMG-SJE.**

**SÃO JOÃO EVANGELISTA
2019**

DAYANE COSTA DOS SANTOS; VANDER CARDOSO DE MACEDO

**PROPOSIÇÃO DE UMA PLATAFORMA WEB PARA CORREÇÃO AUTOMÁTICA
DE PROVAS DO MICROSOFT WORD 2010 COM BASE NAS EMENTAS DE
INFORMÁTICA BÁSICA DO IFMG-SJE.**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais - *Campus* São João Evangelista como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Rosinei Soares de Figueiredo

**SÃO JOÃO EVANGELISTA
2019**

FICHA CATALOGRÁFICA

S194p Santos, Dayane Costa dos; Macedo, Vander Cardoso de.
2019

Proposição de uma plataforma web para correção automática de provas do Microsoft Word 2010 com base nas ementas de Informática básica do IFMG-SJE. / Dayane Costa dos Santos; Vander Cardoso de Macedo. – 2019.
57fl; il.

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista, 2019.

Orientador: Me. Rosinei Soares de Figueiredo.

1. Automatização. 2. Correção. 3. Informática Básica. I. Santos, Dayane Costa dos. II. Macedo, Vander Cardoso de. III. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista. IV. Título.

CDD 670.42755

Elaborada pela Biblioteca Professor Pedro Valério
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais
Campus São João Evangelista
Bibliotecária Responsável: Rejane Valéria Santos – CRB-6/2907

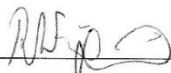
DAYANE COSTA DOS SANTOS; VANDER CARDOSO DE MACEDO

**PROPOSIÇÃO DE UMA PLATAFORMA WEB PARA CORREÇÃO AUTOMÁTICA
DE PROVAS DO MICROSOFT WORD 2010 COM BASE NAS EMENTAS DE
INFORMÁTICA BÁSICA DO IFMG-SJE**

Trabalho de conclusão de curso apresentado ao
Instituto Federal de Minas Gerais - Campus
São João Evangelista como exigência parcial
para obtenção do título de Bacharel em
Sistemas de Informação.

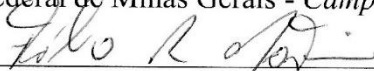
Aprovado em: 28/11/2019

BANCA EXAMINADORA



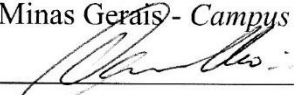
Orientador Prof. Me. Rosinei Soares de Figueiredo

Instituto Federal de Minas Gerais - *Campus* São João Evangelista



Convidado: Prof. Me. Fábio Rodrigues Martins

Instituto Federal de Minas Gerais - *Campus* São João Evangelista



Convidado: Prof. Me. Dênis Rocha de Carvalho

Instituto Federal de Minas Gerais - *Campus* São João Evangelista

**SÃO JOÃO EVANGELISTA
2019**

AGRADECIMENTOS

Agradecemos primeiramente a Deus, por ter nos agraciados com saúde e forças para superar os momentos difíceis que deparamos ao longo da nossa graduação. Aos nossos familiares e amigos, em especial aos nossos pais, que nos apoiaram muito no decorrer da nossa jornada e que mesmo à distância, estiveram presentes em todos os momentos.

Aos nossos professores, pela dedicação em nos ensinar, pela paciência, pelos bons conselhos, instruções e esclarecimentos, por sempre buscarem nos ajudar e incentivar durante nosso percurso.

Ao nosso orientador, Professor Me. Rosinei Soares Figueiredo, por ter proposto esse desafio a nós, confiando que seríamos capazes, por todo suporte, apoio, dedicação, compreensão e paciência que teve conosco durante este trabalho.

Por fim, agradecemos a todos que contribuíram de alguma maneira para nossa formação e para execução desse trabalho. Agradecemos a instituição por proporcionar a oportunidade de obtenção do título de Bacharel em Sistemas de Informação.

RESUMO

O referido trabalho trata-se de uma proposição de uma plataforma *Web* para correção de avaliações da ferramenta *Microsoft Word* 2010, para o seu desenvolvimento foi analisado os planos de aulas da disciplina de informática básica do IFMG-SJE. Observou-se a necessidade da automatização do processo de correção realizado pelos professores, exigindo um grande tempo que poderia ser empregado no aperfeiçoamento das aulas e demais atividades. Portanto, tornou-se viável estudar, validar e implementar um sistema que facilitasse e agilizasse essas correções, sendo este estudo uma implementação inicial que irá oferecer futuro apoio aos professores no processo de correção de avaliações. A metodologia utilizada neste estudo detém caráter qualitativo, não tendo como finalidade a quantificação dos dados, e sim, o levantamento, compreensão e qualificação dos mesmos. Para a implementação do modelo inicial de correção, foi utilizado o *framework CodeIgniter* para agilizar o processo de implementação, os editores de texto *Sublime Text* e *Atom*, a Linguagem de Programação *Hypertext Processor* (PHP) para codificação *Web*, a Linguagem de Marcação de Hipertexto (HTML), Folhas de Estilo em Cascata (CSS) para estilização, *Sourcetree* para controle de versionamento e o MySQL, sendo este o banco de dados da plataforma. Com o uso dessas tecnologias, o objetivo deste estudo foi alcançado, desenvolver a implementação inicial de uma plataforma *Web* para correção automática de provas do *Microsoft Word*.

Palavras-chaves: Automatização. Correção. Informática Básica.

ABSTRACT

The said work is about a proposition of a web platform for correcting evaluations of the Microsoft Word 2010 tool, for its development, the lesson plans of the IFMG-SJE basic computer discipline were analyzed. The need for of automation of the correction process performed by the teachers was observed, requiring a large amount of time that could be spent on improving classes and other activities. Therefore, it became feasible to study, validate and implement a system that would facilitate and speed up these corrections, this study being an initial implementation that will offer future support to teachers in the assessment correction process. The methodology used in this study has a qualitative character, not having as purpose the quantification of the data, but the survey, understanding and qualification of the same. To implement the initial correction model, the CodeIgniter framework was used to streamline the implementation process, the Sublime Text and Atom text editors, the Hypertext Processor (PHP) programming language for Web coding, the Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) for styling, Sourcetree for versioning control and MySQL, this being the platform's database. Using these technologies, the objective of this study was achieved, to develop the initial implementation of a web platform for automatic correction of Microsoft Word evaluations.

Keywords: Automation. Correction. Basic Informatic.

LISTA DE FIGURAS

Figura 1 - Diagrama de Caso de Uso.....	33
Figura 2 - Diagrama de Classe	33
Figura 3 - Arquivo .docx	37
Figura 4 - Configuração de parágrafo aplicada	38
Figura 5 - Representação XML do arquivo Word.....	39
Figura 6 - Logomarca	40
Figura 7 - Tela inicial de configurações	40
Figura 8 - Tela de configurações do título.....	41
Figura 9 - Configurações do título registradas com sucesso	42
Figura 10 - Tela de configurações do texto	42
Figura 11 - Configurações do texto registradas com sucesso.....	43
Figura 12 - Avançar para próxima página	44
Figura 13 - Tela de Upload de arquivos	44
Figura 14 - Enviar arquivos para correção	45
Figura 15 - Efeito ao passar o mouse em cima.....	45
Figura 16 - Relatório	46

LISTA DE SIGLAS

UML - *Unified Modeling Language*

XP - *Extreme Programming*

HTML - *HyperText Markup Language*

XML - *Extensible Markup Language*

HTTP - *Hypertext Transfer Protocol*

CSS - *Cascading Style Sheet*

PHP - *Hypertext Preprocessor*

SGBD - Sistema Gerenciador de Banco de Dados

SQL - *Structure Query Language*

IDE - *Integrated Development Environment*

AJAX - *Asynchronous JavaScript and XML*

OBI - Olimpíada Brasileira de Informática

W3C - *World Wide Web Consortium*

MVC - *Model-View-Controller*

CRUD - *Create, Read, Update and Delete*

IFMG - SJE - Instituto Federal de Minas Gerais - *Campus São João Evangelista*

GNU - *General Public Licence*

SGML - *Standard Generalized Markup Language*

SUMÁRIO

1. INTRODUÇÃO	10
2. REFERENCIAL TEÓRICO	12
2.1 ENGENHARIA DE SOFTWARE	12
2.2 REQUISITOS DE SOFTWARE	13
2.3 VERSIONAMENTO DE SOFTWARE	14
2.4 DESENVOLVIMENTO DO SOFTWARE.....	15
2.5 PROCESSO DE SOFTWARE	15
2.6 MODELAGEM DE SOFTWARE.....	16
2.7 MÉTODOS ÁGEIS	16
2.7.1 Scrum	17
2.7.2 Extreme programming (XP)	18
2.8 TECNOLOGIAS E FERRAMENTAS UTILIZADAS	19
2.8.1 HTML	20
2.8.2 XML	20
2.8.3 Javascript	21
2.8.4 CSS	21
2.8.5 Bootstrap	21
2.8.6 Php	22
2.8.7 Sistema de gerenciamento de banco de dados	22
2.8.8 Ide's utilizadas	24
2.8.9 Ajax e jquery	24
2.8.10 Codeigniter	25
2.8.11 XAMPP	25
2.9 MICROSOFT WORD	26
2.10 OPEN XML	26
2.11 OFFICE OPEN XML	27
2.12 TRABALHOS CORRELATOS	27
3. METODOLOGIA	29
3.1 NATUREZA DA PESQUISA	29
3.2 POPULAÇÃO E AMOSTRA	30
3.3 INSTRUMENTOS UTILIZADOS	30
3.4 MÉTODOS E PROCEDIMENTOS	31

4. RESULTADOS E DISCUSSÕES	36
4.1 VERSÃO INICIAL.....	39
5. CONSIDERAÇÕES FINAIS	47
REFERÊNCIAS	48
APÊNDICES	54
APÊNDICE A – Exemplo de um arquivo document.xml.	54

1. INTRODUÇÃO

Os conhecimentos de informática básica são de grande importância na preparação de profissionais qualificados para o mercado de trabalho, que está cada vez mais exigente. Ao longo dos anos o avanço tecnológico tem provocado transformações, modificando os papéis de diversos profissionais, dos quais são exigidas novas habilidades e competências para atuar na sociedade (OLIVEIRA, 2007).

Sendo assim, as escolas e instituições de ensino procuram oferecer disciplinas relacionadas à informática básica aos seus alunos, com a finalidade de auxiliar e impulsionar o desenvolvimento profissional e pessoal dos estudantes.

Na disciplina de informática básica, um dos diversos conteúdos abordados é o pacote *Office*, sendo este muito utilizado nas áreas acadêmicas como, por exemplo, para elaboração de projetos, trabalhos, artigos científicos, entre outros.

Dentre as ferramentas mais utilizadas do pacote *Office* está o *Word*. Este é um processador de textos desenvolvido pela *Microsoft*. Entre as suas diversas funcionalidades estão: escrita e edição de textos, inserção de tabelas, imagens, entre muitos outros recursos (GIORDANE, 2014).

Nos diversos cursos em que a disciplina de informática básica é contemplada, é comum a realização de provas específicas da ferramenta *Word*, nestas são cobrados conhecimentos de formatação, edição e normas gerais seguidas por determinadas instituições.

No dia 06/08/2019 foi informado pela secretária de registros escolares do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus São João Evangelista (IFMG-SJE) que é ofertada no Instituto, a disciplina de Informática Básica para os cursos Técnicos de: Agropecuária, Nutrição e Dietética, Informática e Agrimensura, também são ofertadas para os Bacharelados em: Administração, Agronomia e Engenharia Florestal, assim como para as Licenciaturas em Ciências Biológicas e Matemática. No segundo semestre de 2019 há o total de 227 alunos matriculados na disciplina de Informática básica, realizando-se uma média de duas provas por semestre, totalizando 454 provas a serem corrigidas.

As correções dessas provas são realizadas manualmente pelo professor, sendo que este poderá ter várias turmas distintas que estão matriculadas na disciplina de informática básica.

Sendo assim, o volume de provas a serem corrigidas é grande. Por consequência, o professor tem uma defasagem de tempo para com a preparação de novos conteúdos e

ministração das aulas, já que o tempo que poderia ser investido para o aperfeiçoamento da disciplina é utilizado para correções repetitivas de provas relacionadas à ferramenta *Word*.

Neste contexto, a proposta do presente trabalho é aplicar recursos tecnológicos existentes para tornar o processo mais eficiente, agregando valor tanto para o professor, quanto para o aluno, ambos serão beneficiados pela agilidade na correção das provas.

O objetivo geral do projeto é elaborar e validar uma proposta de uma plataforma *Web (software)* para realizar a correção automática de avaliações realizadas no *Microsoft Word*.

Para alcançar o objetivo geral, definem-se os seguintes objetivos específicos:

- a. compreender o processo de correção manual praticado pelos professores;
- b. elaborar e propor um processo para correção automática;
- c. levantar e documentar os requisitos do sistema;
- d. compreender a estruturação dos documentos do *Word*;
- e. implementar uma versão inicial do sistema para validação;
- f. validar o sistema que será desenvolvido.

O desenvolvimento dessa proposta proporcionará benefícios aos docentes ministrantes da disciplina de informática básica, sendo esses: redução do tempo destinado à correção das avaliações, permitindo maior agilidade e possibilitando a correção de maiores quantidades de avaliações em um determinado período de tempo, sendo esse relativamente menor se comparado às correções manuais, maior padronização nas correções ao retirar a subjetividade do processo.

2. REFERENCIAL TEÓRICO

Essa seção apresenta as pesquisas relacionadas e conceitos sobre os temas que serão abordados no desenvolvimento do estudo. Serão apresentados alguns autores que irão prover suporte e embasamento teórico, contribuindo para a compreensão do problema a ser tratado.

2.1 ENGENHARIA DE SOFTWARE

A engenharia de *software* é uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software (PRESSMAN, 2016, p. 15). “A engenharia de *software* analisa questões práticas de custo, prazo e confiança, assim como as necessidades dos clientes e produtores de *software*” (SOMMERVILLE, 2011).

Pressman define que:

Software de computador é o produto que profissionais de *software* desenvolvem e ao qual dão suporte a longo prazo. Abrange programas executáveis em um computador de qualquer porte ou arquitetura, conteúdos (apresentados à medida que os programas são executados), informações descritivas tanto na forma impressa (*hard copy*) como na virtual, abrangendo praticamente qualquer mídia eletrônica (PRESSMAN, 2011).

Segundo Sommerville (2011), a engenharia de *software* não foca apenas nos processos técnicos do desenvolvimento de *software*. Está relacionada a todos os aspectos da produção de *software*, como atividades de gerenciamento de projetos de *software*, desenvolvimento de ferramentas, métodos e teorias para apoiar a produção de *software*.

Conforme afirma Pressman (2011), o *software* é importante porque afeta quase todos os aspectos da vida das pessoas e das organizações, ele incorporou-se no comércio, na cultura e nas atividades cotidianas das pessoas, portanto o *software* deve apresentar qualidade elevada, pois indivíduos, negócios e governos dependem cada vez mais de *softwares* para decisões estratégicas e táticas, assim como para controle e para operações cotidianas. Ele também afirma que qualquer abordagem de engenharia (inclusive engenharia de *software*) deve estar fundamentada em comprometimento organizacional com a qualidade e que a pedra fundamental que sustenta a engenharia de *software* é o seu foco na qualidade.

Na área da engenharia de *software* existem duas vertentes para o processo de desenvolvimento de *software*, sendo essas as metodologias tradicionais e ágeis. As

metodologias tradicionais são consideradas “pesadas” ou dirigidas por documentação, apresentam como aspecto a sua divisão em etapas e/ou fases, sendo estas bem definidas. A metodologia tradicional compreende atividades para planejamento detalhado e minucioso do sistema, envolvendo atividades como: análise, modelagem, desenvolvimento e testes. Normalmente cada fase finalizada gera uma documentação do *software*. Já as metodologias ágeis possuem caráter adaptativo e direcionado a pessoas (MAINART; SANTOS, 2010).

Segundo Lima (2015), Metodologias de desenvolvimento de Software é um conjunto de atividades, ordenadas ou não, com a finalidade de produzir um software de qualidade. Entre as atividades associadas a Engenharia de *software* temos: Requisitos de *software*, versionamento de *software*, desenvolvimento do *software*, processo de *software* e métodos ágeis. Sendo estes contemplados nos tópicos seguintes.

2.2 REQUISITOS DE SOFTWARE

Para Sommerville (2011), os requisitos de um sistema são as funcionalidades que devem estar presentes no sistema, os serviços que este oferece e as suas limitações de funcionamento. Os requisitos são um reflexo das necessidades dos clientes para um sistema que serve a uma finalidade específica. O processo de descobrir, analisar, documentar e verificar esses serviços e limitações é denominado engenharia de requisitos. Os requisitos de *software* são classificados como:

- a. requisitos funcionais: são os serviços e funcionalidades que possuem valor para o cliente e que o sistema deve contemplar;
- b. requisitos não funcionais: são limitações aos serviços ou funções que são usuais em um sistema.

A implementação de um conjunto específico de requisitos de *software*, funcionais ou não funcionais, compõe uma versão de *software*. O tópico seguinte irá abordar de forma mais coesa e detalhada o versionamento de *software*.

2.3 VERSIONAMENTO DE SOFTWARE

De acordo com Sommerville (2011), gerenciamento de versões “é o processo de acompanhamento de diferentes versões de componentes de *software* ou itens de configuração e os sistemas em que esses componentes são usados”. O versionamento gerencia as inúmeras versões de um *software*, proporcionando uma agilidade maior para o desenvolvedor, havendo um maior controle sobre as mudanças realizadas no *software*. Os sistemas de gerenciamento de versões oferecem uma diversidade de recursos, sendo estes:

- a. identificação de versão e *release*: as versões gerenciadas recebem identificadores ao serem submetidas ao sistema;
- b. gerenciamento de armazenamento: com a intenção de reduzir o espaço necessário para armazenar as várias versões de componentes que possuem poucas alterações, os sistemas de gerenciamento de versões fornecem recursos de gerenciamento de armazenamento;
- c. registro de histórico de mudanças: todas as mudanças que ocorrem no código de um sistema ou componente são registradas;
- d. desenvolvimento independente: é possível que desenvolvedores distintos trabalhem no mesmo componente ao mesmo tempo;
- e. suporte a projetos: um sistema de gerenciamento de versões é capaz de apoiar o desenvolvimento de vários projetos, sendo que estes compartilham os mesmos componentes.

Há uma grande variedade de *softwares* utilizados quando relacionados ao versionamento de código, alguns mais comuns são: Git, *Bitbucket* e o *Sourcetree*.

O Git é um sistema para o controle de versão distribuído, tem código aberto e sua utilização é gratuita, foi projetado para atender as demandas de projetos de diversos portes, apresentando velocidade e eficiência, propiciando ainda uma fácil aprendizagem (GIT, 2019).

O *Bitbucket* é um servidor de hospedagem de códigos que apresenta o gerenciamento de códigos do Git, oportuniza às equipes de desenvolvimento um ambiente para planejar o projeto, colaborar com o desenvolvimento do código, testar e implementá-lo (BITBUCKET, 2019).

O *Sourcetree* facilita a interação com o repositório Git, é simples para desenvolvedores iniciantes, assegura uma facilidade no controle de versão e maior produtividade para usuários mais avançados (SOURCETREE, 2019).

2.4 DESENVOLVIMENTO DO SOFTWARE

Segundo Pressman e Maxin (2016), o desenvolvimento de *software* é um processo compreendido na área de engenharia de *software*. Pode-se dividir o desenvolvimento de *software* em:

- a. concepção: são feitas análises sobre o sistema, levantando a viabilidade do mesmo e como será a sua construção;
- b. levantamento de requisitos: é realizada a identificação das funcionalidades que o sistema deverá contemplar, são planejadas soluções que atendam aos requisitos e as necessidades do cliente;
- c. projeto: os dados obtidos pelo levantamento de requisitos são transformados em uma documentação que será interpretada pela equipe de desenvolvimento;
- d. implementação: é a fase de codificação dos módulos do sistema, traduzindo o projeto de design em um programa;
- e. testes: para garantir a qualidade e o funcionamento adequado do sistema.
- f. implantação: o sistema é implantado para o cliente, os usuários recebem instruções para a utilização do sistema e para auxiliar no gerenciamento de possíveis mudanças provocadas pela implantação do sistema.

2.5 PROCESSO DE SOFTWARE

Para Sommerville (2011), um processo de *software* é uma sequência de atividades com a finalidade de produção de *software*. Entre as diversas atividades, quatro são fundamentais, sendo elas:

- a. especificação do *software*: As funcionalidades são elicitadas pelos engenheiros do *software* com auxílio do cliente.
- b. desenvolvimento de *software*: Os requisitos são implementados de acordo com as especificações requeridas pelo cliente.
- c. validação de *software*: é verificado se as funcionalidades do *software*, atendem as especificações do cliente.

- d. evolução de *software*: o *software* é adaptado, de acordo com os requisitos do cliente e as demandas do mercado.

O processo de *software* é uma abordagem adaptável, que proporciona a equipe de desenvolvimento, trabalhar com a seleção e a escolha do conjunto mais adequado de ações e tarefas para o desenvolvimento de um *software*. Tendo como finalidade a entrega do produto dentro do prazo estimado, atendendo às necessidades dos patrocinadores e dos usuários (PRESSMAN, 2011).

2.6 MODELAGEM DE SOFTWARE

De acordo com Sommerville (2011, p. 82) “Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema”. Uma forma habitual de modelagem de *software* é através da *Unified Modeling Language* (UML), traduzida para o português significa Linguagem de modelagem unificada

Conforme Guedes (2014), a UML é uma linguagem visual, utilizada para a modelagem de sistemas computacionais por meio do paradigma de orientação a objetos. Essa linguagem foi adotada internacionalmente pela indústria de engenharia de *software* para modelagem.

Gudwin (2010) defende que a linguagem UML pode ser utilizada para a modelagem de todo o processo de desenvolvimento de *software* e produção dos artefatos de *software* necessários à documentação.

2.7 MÉTODOS ÁGEIS

Em relação aos métodos ágeis, Sommerville fala que:

Geralmente, os métodos ágeis contam com uma abordagem iterativa para especificação, desenvolvimento e entrega de *software*, e foram criados principalmente para apoiar o desenvolvimento de aplicações de negócios nas quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento. Eles destinam-se a entregar um *software* de trabalho rapidamente

aos clientes, que podem então propor novos requisitos e alterações a serem incluídos nas iterações posteriores do sistema (SOMMERVILLE, 2007).

Com intuito de encontrar formas mais eficientes de desenvolver sistemas complexos, foram criadas diversas iniciativas, a partir destas, originou-se algumas metodologias, que enfatizam maior valor nos fatores humanos e na satisfação do cliente, em vez de agregar maior valor na burocracia dos processos (SILVA, 2013), essas metodologias são conhecidas como métodos ágeis.

Para Sommerville (2007), entre os métodos ágeis mais conhecidos e utilizados, podem ser abordados: *Extreme Programming* (XP), *Scrum*, *Crystal*, Desenvolvimento Dirigido a Características (*Feature Driven Development*), Desenvolvimento de *Software* Adaptativo e Método Dinâmico de Desenvolvimento de *Software*.

Conforme Vidor *et al.* (2013), “os métodos *Scrum* e XP são dois métodos ágeis que focam nos clientes e processos facilitando a vida do programador com métodos que se adaptam às mudanças, falhas e entregas frequentes”. O *Scrum* é uma gestão ampla para projetos que não possuem dependência com a área de conhecimento, enquanto o XP possui uma aplicação restrita, focada no processo de desenvolvimento de *software*. Na utilização do *Scrum* para gerência e criação de *softwares*, as utilizações de práticas presentes no XP são de grande importância, como por exemplo, o uso da refatoração de código, para um maior ganho de qualidade e garantia que a manutenção futura seja simplificada (DEVMEDIA, 2019).

Nos tópicos seguintes serão abordados os dois métodos ágeis mais utilizados atualmente, sendo esses o *Scrum* e o XP.

2.7.1 Scrum

Segundo Schwaber e Sutherland (2017), *Scrum* é um *framework* para desenvolver, entregar e manter produtos complexos. O *Scrum* consiste de times *Scrum* associados a papéis, eventos, artefatos e regras. Cada componente dentro do *framework* serve a um propósito específico, sendo essencial para o uso e sucesso do *Scrum*. Ele foi desenvolvido por Ken Schwaber e Jeff Sutherland. Sabbagh (2013) define o *Scrum* como:

[...] *framework* ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. *Scrum* é baseado no empirismo e utiliza uma abordagem iterativa e incremental para entregar valor com frequência e, assim, reduzir os riscos do projeto (SABBAGH, op. cit.).

De acordo com Schwaber e Sutherland (2017), o time *Scrum* é composto por três papéis principais: *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. O *Product Owner* é o responsável por maximizar o valor do produto resultado do trabalho do Time de Desenvolvimento, ele é responsável por gerenciar o *Backlog* do Produto. O time de desenvolvimento é responsável pelo desenvolvimento do projeto, eles são auto organizados e multifuncionais. O *Scrum Master* é responsável por promover e suportar o *Scrum*, auxiliando a todos a entenderem a teoria, as práticas, as regras e os valores do *Scrum*, ele é um servolider para o time de desenvolvimento.

Para Bona (2002) *apud* Lima, Freire e Costa (2012) *Scrum* se baseia em práticas que são representadas por:

- a. reuniões diárias que são realizadas com todos os membros do time de desenvolvimento, tendo duração máxima de 15 minutos, nessas reuniões os integrantes do time apresenta o que fez no dia anterior, e o que estará fazendo no dia. Caso exista algum impedimento ele também é abordado nessa reunião.
- b. reuniões de planejamento de *Sprint* (ciclo de desenvolvimento do projeto de *software*, podendo ter sua duração de semanas a um mês);
- c. reunião de revisão de *Sprint*: É realizado uma retrospectiva da *Sprint*, debatendo os pontos positivos e negativos do decorrer do desenvolvimento;
- d. ordenação do *backlog*: lista dos requisitos do *software*, é realizado conforme a prioridade de cada funcionalidade. Sendo assim, as funcionalidades de maiores prioridades são implementadas antes que as de menores prioridades;
- e. apresentação das versões do *software*: Por meio desta apresentação o cliente pode acompanhar a implementação do sistema.

O projeto *Scrum* se inicia quando a visão do que irá se desenvolver é criada. Nessa visão são observadas as características que o cliente deseja que o projeto atenda ao final do seu desenvolvimento (ALBINO; SOUZA; PRADO, 2013).

2.7.2 Extreme programming (XP)

O *Extreme Programming* (XP) é possivelmente o mais conhecido e utilizado dos métodos ágeis. O nome foi determinado por Beck (2000) *apud* (SOMMERVILLE, 2011), visto que a abordagem foi desenvolvida para estimular práticas reconhecidas como boas a

níveis extremos. No XP os requisitos são representados como cenários, esses são chamados de estórias do usuário, tais estórias são implementadas como uma sequência de tarefas.

De acordo com Sommerville (2011), o XP compreende uma sequência de práticas que refletem o princípio dos métodos ágeis:

- a. o desenvolvimento incremental é sustentado pela entrega de pequenas e frequentes versões do sistema, os requisitos têm como base os cenários ou estórias dos clientes, usadas como suporte para decidir a funcionalidade que deve ser incluída em um incremento do sistema;
- b. o envolvimento do cliente faz-se por meio da participação ativa com a equipe de desenvolvimento, o cliente possui um representante que participa do desenvolvimento, tal representante é responsável pela definição dos testes de aceitação para o sistema;
- c. foco nas pessoas e não nos processos, são sustentados por intermédio de programação em pares, propriedade coletiva do código do sistema, sendo um processo de desenvolvimento sustentável, levando em consideração o fator humano, por isso não envolve horas de trabalho excessivas;
- d. as mudanças são aceitas através de *releases* contínuos para os clientes, do desenvolvimento *test-first* (teste antes da codificação), mediante refatoração para impedir a degeneração do código e integrar continuamente novas funcionalidades;
- e. em busca da simplicidade, o XP não antecipa desnecessariamente futuras mudanças no sistema, as necessidades imediatas são prioridades.

Para Souza (2007), XP é um conjunto estabelecido de regras, que vem obtendo um grande número de adeptos. O XP oferece condições para que os desenvolvedores reajam com eficiência às mudanças constantes no projeto.

2.8 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Atualmente existem incontáveis tecnologias e ferramentas que possibilitam o desenvolvimento de um sistema *Web*, sendo estas indispensáveis para o sucesso do desenvolvimento, alguns exemplos dessas tecnologias e ferramentas são: servidor, editor de texto, sistema gerenciador de banco de dados, linguagens de desenvolvimento e marcação. Nesse tópico será apresentado às tecnologias e ferramentas que serão utilizadas para o desenvolvimento deste trabalho.

2.8.1 HTML

A abreviação HTML vem do inglês, que significa *Hypertext Markup Language*, traduzindo para o português significa linguagem de marcação de hipertexto, o HTML é uma linguagem de marcação para a organização e formatação de páginas *Web*. Segundo W3C “Documentos HTML consistem em uma árvore de elementos e texto”, sendo estes exibidos por meio de *tags* (W3C, 2018).

O HTML aplica a estruturação da página *Web* que será exibida no navegador, porém ele não consegue desenvolver a aparência e estilização da página, por isso o HTML trabalha juntamente com *Cascading Style Sheets* (CSS), este irá definir a estilização da página *Web*.

2.8.2 XML

O XML, abreviatura de *Extensible Markup Language*, traduzida para o português, significa linguagem de marcação extensível, segundo a W3C (2016), o XML é um formato de texto simples e flexível, que tem derivação do SGML, abreviatura de *Standard Generalized Markup Language*, traduzido para o português, significa linguagem de marcação generalizada padrão, esta, especifica as normas para criação de linguagens de marcações que sejam independentes da plataforma, O XML foi originalmente planejado para enfrentar desafios da publicação eletrônica em grande escala. Para o armazenamento e dados, sendo desenvolvido para ser humanizado e legível por máquina (W3SCHOOLS, 2019).

O XML é uma linguagem de marcação de documentos, independente do *hardware* e *software* utilizados, possui padrão aberto, que fornece uma variedade de regras que descrevem o conteúdo dos documentos, tem-se como um dos fundamentais propósitos do XML, provê benefícios do SGML, que não estão permissíveis no HTML, em uma linguagem com menor tempo para aprendizagem e simples de se utilizar (SOUZA; PEREIRA; CARVALHO, 2016).

2.8.3 Javascript

O W3schools (2019) define o *Javascript* como uma linguagem de programação do HTML e da *Web*, com a curva de aprendizagem simples. O *Javascript* utilizado da maneira tradicional, como *scripts* que rodam dentro do navegador *Web*, não funciona remotamente em servidores na internet, ele tem como característica executar localmente. Desta forma, o *Javascript* possibilita às páginas *Web* serem programadas, transformadas e o seu processamento dos dados enviados e recebidos, interagindo com a marcação e exibição de conteúdo da linguagem HTML, utilizando-se do CSS para a estilização do conteúdo nas páginas *Web* (SILVA, 2015).

2.8.4 CSS

O CSS vem do inglês, que significa *Cascading Style Sheets*, traduzido para o português significa folhas de estilo em cascata, “é um mecanismo simples para adicionar estilo (por exemplo, fontes, cores, espaçamento) a documentos da *Web*” (W3C, 2019). Sendo uma linguagem descritiva do estilo de um documento HTML, o CSS descreve como será a exibição dos elementos HTML (W3SCHOOLS, 2019).

O CSS, de início, foi desenvolvido para dispor a fragmentação do conteúdo e formato de um documento de sua exibição. Essa fragmentação propiciou uma maior flexibilidade, controlando as especificações de como as características da página seriam exibidas, permitindo compartilhamento de formato e reduzindo repetições desnecessárias no conteúdo estrutural da página (TECMUNDO, 2019).

2.8.5 Bootstrap

Originalmente foi desenvolvido por um *designer* e um desenvolvedor do *twitter*, o *Bootstrap* se tornou um dos *frameworks front-end* mais conhecidos e utilizados na atualidade (BOOTSTRAP, 2019).

O *Bootstrap* foi concebido para acelerar o desenvolvimento *front-end*, Proporciona agilidade e qualidade no desenvolvimento de interfaces, este *framework* possibilita a utilização de componentes de interfaces prontos, otimizando o tempo de desenvolvimento, sendo que não é preciso criar a interface do zero (BORTOLI; RUFINO, 2016).

O *Bootstrap* possui seu código fonte aberto para alterações, possibilitando às diversas pessoas terem acesso ao código e até mesmo criar o seu próprio código baseado no *Bootstrap*. “O *Bootstrap* é a estrutura mais popular de HTML, CSS e *Javascript* para o desenvolvimento de *Websites* responsivos e móveis” (W3SCHOOLS, 2019). Propiciando assim, a agilidade no desenvolvimento, dispendo de uma ampla praticidade ao se tratar de responsividade, adequando-se a inúmeros navegadores da *Web*.

2.8.6 Php

O PHP é uma abreviatura para *Hypertext Preprocessor*, traduzido para o português significa Pré-processador de Hipertexto, é uma linguagem de programação *Web* de fonte aberta, seu código e sua documentação estão disponíveis no seu site oficial (php.net). De acordo com o índice Tiobe (2019), este, lista as linguagens de programações mais pesquisadas e expandidas no mercado, o PHP é a oitava linguagem de programação mais difundida no mês de Julho de 2019. O PHP é adequado para o desenvolvimento de sistema *Web*, é possível sua utilização em junção com o HTML.

O que diferencia o PHP das demais linguagens de desenvolvimento *Web*, que são interpretadas diretamente pelo navegador, é que o código PHP será executado no servidor, posteriormente gerando um HTML que será interpretado pelo navegador. Uma vantagem de ser utilizar o PHP é sua grande facilidade para desenvolvedores iniciantes, porém oferece inúmeros recursos avançados para um desenvolvedor já experiente (PHP, 2019).

2.8.7 Sistema de gerenciamento de banco de dados

“Um banco de dados representa algum aspecto do mundo real, às vezes chamado de minimundo ou de universo de discurso” (ELMASRI; NAVATHE, 2011). Entende-se por

dado, todas as informações que podem ser armazenadas e possuem algum significado incluso ao contexto em que ela é aplicada (MEIRE, 2016).

Um banco de dados informatizado é frequentemente mantido e acessado por meio de um *software* conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Meire (2016, p. 4), afirma que o SGBD “é uma coleção de programas que permitem ao usuário definir, construir e manipular bases de dados para as mais diversas finalidades”. O modelo de dados que é mais usual hoje em dia, para a representação e o armazenamento de dados em um SGBD, é o modelo relacional, neste modelo as estruturas têm o formato de tabelas, sendo composta por linhas e colunas (MEIRE, 2016, p. 4).

Segundo Rob e Coronel (2011, p. 7), a existência de um SGBD entre as aplicações do usuário final e o banco de dados, oferece diversas vantagens consideráveis. O SGBD possibilita que os dados sejam distribuídos por diversas aplicações e usuários, integrando visualizações distintas dos usuários com relação aos dados, sendo este em um único repositório e estando englobado.

Segundo o site oficial MySQL (2019), ele fornece um servidor de banco de dados SQL (*Structured Query Language*), possuindo seu código fonte aberto. “O *MySQL Server* destina-se a sistemas de produção de carga pesada de missão crítica, bem como para incorporação em *software*” (MYSQL, 2019).

De acordo com a documentação disponível no site oficial MySQL (2019), O *MySQL Workbench* é uma ferramenta gráfica para trabalhar com servidores e bancos de dados MySQL. Utilizado para o desenvolvimento SQL, pois ele permite criar e gerenciar conexões com servidores de banco de dados, executar consultas SQL nessas conexões utilizando um editor SQL integrado. O *MySQL Workbench* possui também outras funcionalidades como: a modelagem de dados, administração do servidor.

De acordo com o *Solid it* (2019), O *ranking DB-Engines* realiza a classificação dos sistemas de gerenciamento de banco de dados de acordo sua popularidade, este, lista o *MySQL* como o segundo sistema de gerenciamento de banco de dados mais popular no período de Agosto/2018 á Agosto/2019.

2.8.8 Ide's utilizadas

Para Chaves e Silva (2008), a utilização de um IDE, abreviatura de *Integrated Development Enviroment*, traduzindo para o português significa Ambiente de Desenvolvimento Integrado, que seja adequado para o desenvolvimento do *software* é de grande importância para que um desenvolvimento alcance o sucesso, essencialmente quando o desenvolvimento é aplicado para a *Web*. O *Sublime Text*, *Atom* e *Netbeans* são IDE's que serão utilizados no desenvolvimento do trabalho, todos apresentam como características serem editores de textos.

Sublime Text é um editor versátil para programadores, não necessitando de grande conhecimento sobre programação para utilizá-lo, é uma ferramenta eficiente e pronta para utilização (SUBLIME HQ, 2019). O *Sublime Text* viabiliza a edição de códigos HTML, este *software* tem interface com diferentes cores, facilitando a compreensão e construção dos códigos (TECHTUDO, 2019).

O *Atom* é um aplicativo de *desktop* que viabiliza a edição de textos. Contém código aberto, criado com integração de HTML, *JavaScript* e CSS, apresenta fácil customização. Disponibiliza a facilidade de divisão da interface em diversos painéis, para comparação dos códigos e edição (ATOM, 2019).

O *Netbeans* é uma plataforma de desenvolvimento, que compõe uma IDE bem completa, ela permite que se desenvolva, de forma rápida e apresenta uma curva de aprendizado simples para aplicativos *desktops*, móveis e *Web* em Java, possibilita também o desenvolvimento por aplicativos HTML, *JavaScript* e CSS. O *Netbeans* possui diversas ferramentas úteis e poderosas, auxiliadoras no desenvolvimento de *software*. (NETBEANS, 2019).

2.8.9 Ajax e jquery

A abreviatura AJAX vem do inglês, que significa *Asynchronous JavaScript and XML*, traduzindo para o português significa *JavaScript* assíncrono e XML, utiliza sistematicamente o *JavaScript* e o XML como alicerce, utilizando também de outras tecnologias. O AJAX faz uso dessas tecnologias para aumentar a interatividade com o usuário, as solicitações de

informação são assíncronas, ou seja, o AJAX pode ser usado para requerer solicitações para o servidor *Web* sem que seja necessário recarregar a página acessada (NIEDERAUER, 2013, p. 15).

A *jQuery* é uma biblioteca *JavaScript* pequena e que engloba vários recursos. Torna a manipulação de documentos HTML, eventos, animações e AJAX mais simples. Combina versatilidade e extensibilidade a *jQuery* simplifica a utilização de *JavaScript* pelos desenvolvedores (JQUERY, 2019).

2.8.10 Codeigniter

O *CodeIgniter* é um *framework* de desenvolvimento de aplicativos, que disponibiliza um conjunto de ferramentas para desenvolvedores que utilizam PHP, tendo como objetivo permitir o desenvolvimento de projetos com maior agilidade se comparado com a criação do código do zero. Há uma coleção de bibliotecas para atividades que são mais comuns e necessárias na maioria dos aplicativos. O *CodeIgniter* possui uma interface simples, com uma estrutura lógica para o acesso às bibliotecas (CODEIGNITER, 2019).

O *CodeIgniter* utiliza a abordagem MVC, a abreviatura vem do inglês, que significa *Model-View-Controller*, traduzindo para o português significa modelo-visão-controle, a qual permite forte separação entre a lógica e a apresentação. O padrão de desenvolvimento MVC é dividido em três camadas *Model*, *View* e *Controller*. O *Model* (Modelo): Cria comunicação da aplicação com o banco de dados fazendo operações CRUD (*Create, Read, Update and Delete*), esta camada é opcional no *CodeIgniter*. A *View* (Visão): É toda a informação apresentada ao usuário, uma *view* é uma página *Web*. O *Controller* (Controle): Serve com um intermediário entre a camada *Model* e a camada *View*, também processa requisições HTTP para gerar páginas (DEV MEDIA, 2019).

2.8.11 XAMPP

De acordo com Apache Friends (2019), “Xampp é uma compilação de *softwares* livres (comparável a uma distribuição Linux), é gratuito e é livre para ser copiado sob os termos da

GNU *General Public Licence*”. O Xampp é um pacote com os servidores fundamentais, este possui código aberto, banco de dados *MySQL*, dispõe de suporte para PHP, entre outras linguagens. Com ele, torna-se possível rodar programas/sistemas localmente, agilizando-se o desenvolvimento do *software* (TECHTUDO, 2012).

2.9 MICROSOFT WORD

O *Microsoft Word* é um eficiente processador de textos, desenvolvido pela *Microsoft*, englobando o pacote de aplicativos da *Microsoft*. Para a *Microsoft* (2019), utilizando-se o *Word* em computadores ou dispositivos móveis é possível:

- a. a criação de um documento do zero ou usar um modelo já existente;
- b. adicionar textos, imagens e vídeos;
- c. pesquisar um tópico específico e encontrar fontes verídicas;
- d. acessar documentos no computador ou dispositivos móveis com o *OneDrive*;
- e. o compartilhar documentos e o trabalhar em equipe com outras pessoas;
- f. controlar e revisar alterações.

O *Microsoft Word* possui compatibilidade com diferentes tipos de documentos de texto, “Os documentos do *Word* (.docx) são compatíveis com a maioria dos aplicativos” (MICROSOFT, 2019).

2.10 OPEN XML

Um arquivo *Open XML* é um arquivo Zip, formato de arquivo empregado para a compactação de dados armazenados no computador, tendo sua criação com base em uma determinada estrutura de pastas e arquivos, sendo esta estrutura composta por XML.

Segundo a *Microsoft* (2019) “Os formatos de arquivo *Open XML* são úteis para os desenvolvedores porque são um padrão aberto e com base em tecnologias reconhecidas: Zip e XML”. Sobre esses formatos, a compatibilidade de marcação é a capacidade de um documento expressado em linguagens de marcação para facilitar a interoperabilidade entre

aplicativos ou as diversas versões de um aplicativo, com um conjunto de recursos diversos (MICROSOFT, 2017).

2.11 OFFICE OPEN XML

O *Office Open XML* é a linguagem na qual os documentos do *Word* são gravados. Sendo um pacote compactado de arquivos que retrata o conteúdo do documento. Salvar o arquivo no formato de documento XML do *Word* fornece por completo o pacote do *Office Open XML* compactado em um arquivo XML (MICROSOFT, 2019).

A especificação de formatos de arquivos do *Office Open XML* descreve recursos possibilitando que um documento *Microsoft Word* possa ser compreendido e suportado em uma versão anterior ou posterior do *Office*, funcionando assim, conforme o esperado (MICROSOFT, 2017).

A Microsoft (2019) informa que “O *Office Open XML* também é a linguagem por trás dos documentos do *PowerPoint* e do *Excel* (e, a partir do *Office* 2013, do *Visio*)”, porém, atualmente é possível fazer a coerção de conteúdo como *Office Open XML* somente para suplementos do *Office* criados para o *Word* (MICROSOFT, 2019).

2.12 TRABALHOS CORRELATOS

Com o intuito de comprovar a relevância do tema abordado, foram pesquisados e analisados trabalhos com características semelhantes a este projeto, porém não foram encontrados trabalhos científicos semelhantes a este.

Por esse motivo, foram realizadas pesquisas relacionadas a sistemas de correções de avaliações em âmbito mais geral. Foram encontrados muitos sistemas (*softwares*) já difundidos no mercado (comercializados) que realizam a correção de avaliações de maneira automatizada, basicamente estes sistemas geram avaliações de múltiplas escolhas e um gabarito, estes são disponibilizados ao professor que realiza a impressão das avaliações e aplica aos alunos. Após a aplicação das avaliações o professor escaneia os gabaritos e o

sistema executa as correções e faz gráficos de notas e desempenho, alguns sistemas podem diferir um pouco dessa descrição, mas é relativamente assim que funcionam esses sistemas.

Como exemplos dos sistemas descritos acima, podem ser citados: o MestreGR, o Sisca, o *Remark Office* OMR, entre outros existentes. O MestreGR é uma ferramenta que oferece automatização do processo de correções de provas e simulados, abrangendo a correção de questões objetivas, somatórias, discursivas e redação. Também oferece a possibilidade ao professor de gerenciar o desempenho dos alunos nas avaliações, por meio de relatórios e gráficos. O MestreGR é um *software* proprietário. O Sisca (Sistema de Correção Automatizada) é uma ferramenta de correção automatizada de testes de múltipla escolha desenvolvida pela OBI, abreviatura de Olimpíada Brasileira de Informática, ela é disponibilizada gratuitamente, para escolas em geral e é utilizada na correção das provas da Modalidade Iniciação da referida competição. O *Remark Office* OMR é um sistema que faz a leitura de gabaritos e correção automática de provas, também possibilita a geração de diversos relatórios estatísticos, e atua nas áreas de: educação, corporativo, órgãos públicos, saúde e pesquisa. O *Remark Office* OMR é um *software* proprietário distribuído no Brasil por GB *Network & Print*.

Todos os sistemas (exemplos) citados acima têm em comum com este projeto, o fato de realizarem de maneira automatizada, a correção de avaliações, produzindo uma redução no tempo desse processo, possibilitando ao professor investir esse tempo na preparação de aulas e conteúdo para os alunos.

Este trabalho trata-se do desenvolvimento de uma plataforma *Web* para automatização do processo de correção de avaliações, da disciplina Informática Básica, em específico as avaliações sobre a ferramenta do pacote *Office*, o *Word*. Portanto, o processo de correção e as características deste sistema diferenciam-se dos que foram encontrados e citados anteriormente.

3. METODOLOGIA

Como fala Kauark, Manhães e Medeiros (2010, p. 53) “A metodologia é a explicação minuciosa, detalhada, rigorosa e exata de toda ação desenvolvida no método (caminho) do trabalho de pesquisa”. Nessa seção serão abordados os métodos de pesquisas empreendidos no desenvolvimento do presente trabalho, englobando a natureza de pesquisa, que irá definir a modalidade da mesma. Será apresentado também a população e amostra, bem como os instrumentos utilizados e os métodos e procedimentos.

3.1 NATUREZA DA PESQUISA

Para coletar informações sobre o processo de correção de avaliações da ferramenta *Word*, inicialmente foram realizadas pesquisas descritivas com os profissionais ministrantes da disciplina de Informática Básica. Em companhia da abordagem descritiva, foi utilizada uma abordagem qualitativa para análise dos dados.

A metodologia de pesquisa que foi utilizada para o desenvolvimento do *software* apresenta o caráter descritivo. Este tem como objetivo principal a descrição das características de determinada população ou fenômeno, podendo compreender também o estabelecimento de relações entre variáveis, entre os aspectos mais consideráveis está a utilização de técnicas padronizadas de coletas de dados (GIL, 2008).

De acordo com Marconi e Lakatos (2010, p. 170), a pesquisa qualitativa-descritiva, “Consiste em investigações de pesquisa empírica cuja principal finalidade é o delineamento ou análise de características de fatos ou fenômenos, a avaliação de programas ou isolamento de variáveis principais ou chave”. A presente pesquisa possui a natureza qualitativa, não tendo como finalidade a quantificação de dados, e sim, a qualificação dos mesmos.

3.2 POPULAÇÃO E AMOSTRA

A pesquisa realizou-se tendo como população os professores que ministram a disciplina de Informática Básica. Já a amostra corresponde aos professores de Informática Básica do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus São João Evangelista (IFMG-SJE).

3.3 INSTRUMENTOS UTILIZADOS

Com o propósito de compreender o processo de correção de provas da ferramenta do pacote *Office*, o *Word*, solicitou-se as provas da disciplina de Informática Básica, que foram aplicadas aos alunos. Depois da coleta de informações por meio da análise das referidas provas, iniciou-se a modelagem, para posteriormente iniciar a implementação do *software*.

Para o desenvolvimento do *software* utilizou-se: HTML para a estruturação do conteúdo, CSS e *Bootstrap* para a sua estilização, *JavaScript* e o PHP para a sua programação. Para a personalização de efeitos especiais nos elementos das páginas, foi utilizada a biblioteca *jQuery*, em conjunto com algumas das tecnologias citadas anteriormente, como HTML, CSS e *JavaScript*, ela possui como característica simplificação da utilização da linguagem *JavaScript*.

Para propiciar maior interatividade com o usuário, em algumas partes do sistema, que realizam requisições ao servidor *Web*, utilizou-se o AJAX, possibilitando que estas requisições sejam realizadas de maneira assíncrona.

O *framework CodeIgniter* foi empregado para o desenvolvimento em PHP, com o objetivo de aplicar o padrão de desenvolvimento MVC, aderindo assim, uma melhor organização do código, facilitando caso apareça uma manutenção futuramente.

Para a criação/preparação do ambiente de desenvolvimento, utilizou-se o Xampp. O banco de dados utilizado foi o *MySQL* em conjunto com *Workbench*, ele possui como característica ser gratuito e ser popularmente utilizado, oferece uma grande quantidade de recursos, sendo que para a manipulação do banco de dados utilizou-se o *MySQL Workbench*.

Durante o desenvolvimento do *software* utilizou-se três IDE's, assim foi possível aproveitar o melhor que cada uma delas tem a oferecer. As IDE's utilizadas são: O *Sublime*

Text, pelo fato de ser uma IDE simples e leve, o *Atom* e o *Netbeans*, por oferecerem algumas funcionalidades específicas que agilizam e auxiliam bastante o desenvolvimento.

Para o controle de versões utilizou-se o *Git*, para o gerenciamento de códigos do *Git* foi aplicado o *Bitbucket*. Para facilitar a interação com o repositório *Git*, foi utilizado o *SourceTree*, que é um *software* criado com uma interface intuitiva que facilita este processo, possibilitando obter maior produtividade.

3.4 MÉTODOS E PROCEDIMENTOS

Para o desenvolvimento do projeto foi utilizado uma abordagem híbrida, não se fixando apenas em uma metodologia específica como a tradicional ou ágil, mas selecionando partes, conceitos e abordagens das metodologias e aplicando o que foi selecionado nesse projeto. Da metodologia ágil, foi utilizada uma adaptação do *Scrum*, sendo que foram adotadas algumas práticas para gerenciar e controlar o processo de desenvolvimento, também foram utilizadas as histórias de usuários do *XP*.

As principais tarefas que compõem o trabalho realizado neste projeto são listadas e descritas abaixo:

1. coletar informações junto aos professores que ministram aulas de Informática Básica: Solicitação de avaliações aplicadas para os alunos da disciplina de Informática Básica, com os referidos professores, a fim de se obter informações relevantes sobre o processo de correção de avaliações relacionadas a ferramenta do pacote *Office*, o *Word*, possibilitando assim, compreensão deste processo que os professores realizam;
2. analisar o processo manual de correção e elaborar uma proposta de correção automática: Com a obtenção das informações sobre este processo de correção, analisar e compreender este processo, foi-se estruturado e elaborado um processo automatizado para a correção de avaliações relacionadas a ferramenta do pacote *Office*, o *Word*;
3. levantar e documentar os requisitos: Após apresentado a proposta do sistema a ser desenvolvido para a automatização do processo de correção de avaliações do *Word*, realizou-se o levantamento dos requisitos do sistema;
4. pesquisar e analisar a estruturação dos documentos do *Word*: Estudo e análise sobre como funciona a estruturação dos documentos *Word*, compreendendo os arquivos internos que compõe um documento *Word*;

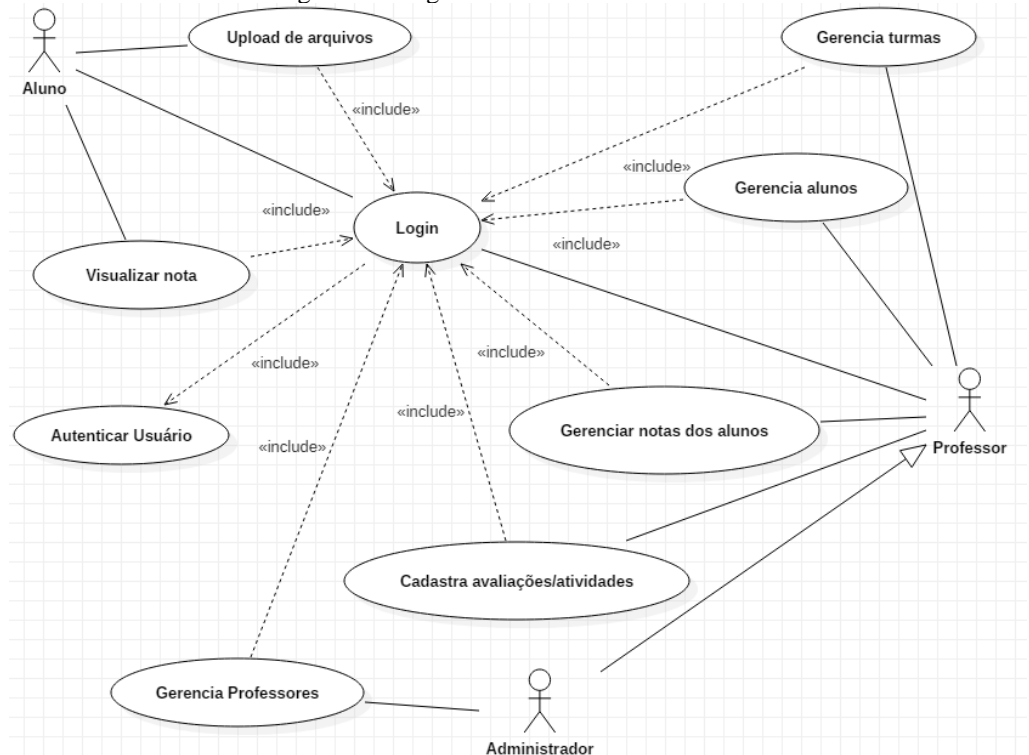
5. implementar uma versão inicial do sistema: Após a definição dos requisitos e a elaboração da proposta de automatização do processo de correção das avaliações relacionadas a ferramenta do pacote *Office*, o *Word*, desenvolver uma versão inicial da ferramenta cobrindo alguns requisitos a serem selecionados;
6. validar a versão inicial do sistema: Disponibilizar o sistema desenvolvido para que professores da disciplina possam testar sua eficiência e, por meio dos *feedbacks* destes professores, realizar a validação do sistema.

O levantamento de requisitos foi realizado por meio de solicitação de documentos e reuniões com professores que ministram aulas de Informática Básica, os requisitos levantados foram analisados e dentre eles escolhidos os que possuem maior relevância para o projeto, definido assim, os requisitos que devem ser implementados no sistema, sendo esses:

1. *crud* dos alunos: Criação, leitura, atualização e exclusão dos alunos;
2. *crud* dos professores: Criação, leitura, atualização e exclusão dos professores;
3. *crud* de turmas: Criação, leitura, atualização e exclusão das turmas;
4. autenticação do usuário por meio de *login*: Para a utilização do sistema é necessário estar previamente cadastrado;
5. vinculação dos alunos: Os alunos que fazem a disciplina de Informática Básica devem ser cadastrados por turmas.
6. *crud* das atividades pelos professores: Criação, leitura, atualização e exclusão das atividades pelos professores;
7. *upload* pelos alunos: Os alunos devem enviar os arquivos requeridos na atividade;
8. correção automatizada: Realizar as correções das atividades que foram solicitadas aos alunos;
9. visualização de notas: As notas obtidas pelos alunos deverão ser disponibilizadas a eles e aos professores;
10. histórico de atividades: Disponibilização de um histórico de atividades realizadas, tanto para os alunos, quanto para os professores.

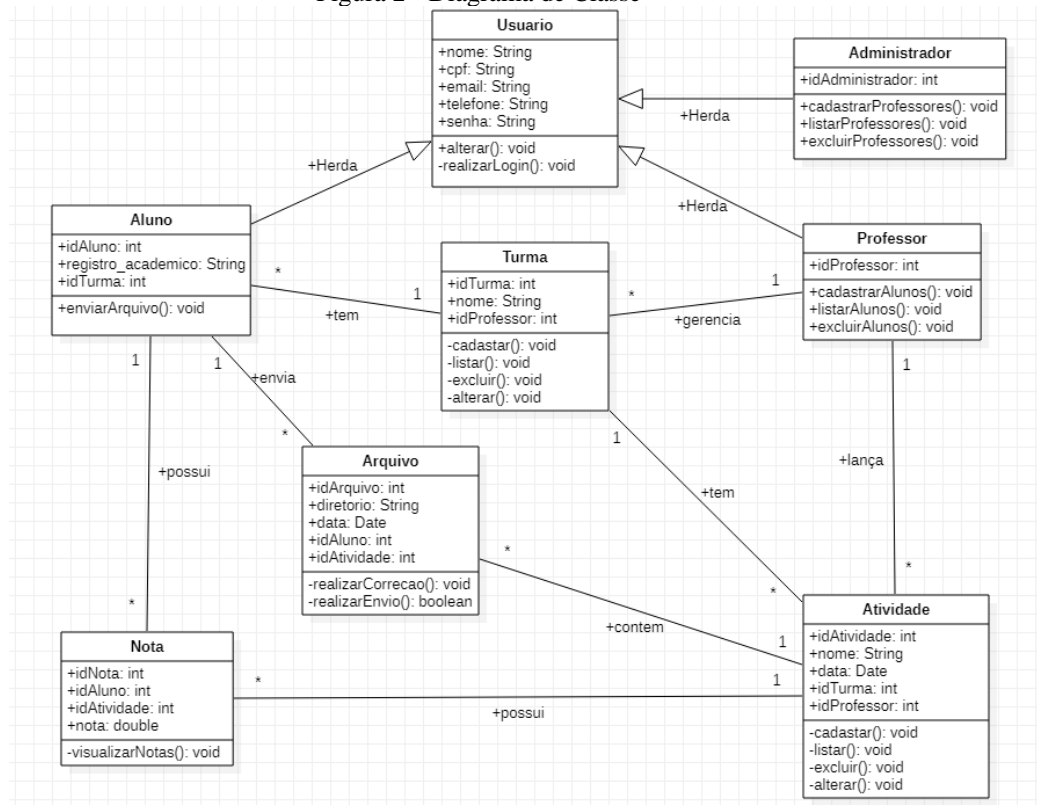
Após a definição dos requisitos, foi criada a modelagem da aplicação, sendo elaborado o diagrama de caso de uso e o de classes do sistema como um todo, para ajudar a compreender melhor o funcionamento do sistema.

Figura 1 - Diagrama de Caso de Uso



Fonte: Elaborada pelos autores.

Figura 2 - Diagrama de Classe



Fonte: Elaborada pelos autores.

Para a representação dos requisitos e funcionalidades do sistema, foi utilizada a representação em cenários, ou seja, histórias de usuário sendo estas bem descritas e detalhadas, possibilitando um melhor entendimento pelos desenvolvedores. Sendo essas:

1. como usuário do sistema preciso que este permita a autenticação do usuário por meio do preenchimento dos campos usuário e senha. Proporcionando segurança das informações presente no sistema;
2. como professor preciso que o sistema possibilite o cadastro dos alunos e turmas, alteração e exclusão dos mesmos, sendo possível identificar de que turma um aluno específico pertence;
3. como professor preciso que o sistema possibilite adicionar e excluir atividades. Caso necessário alterar as informações destas;
4. como professor quero que seja possível o aluno enviar um arquivo referente a uma atividade;
5. como professor preciso que o sistema realize as correções das atividades;
6. como professor quero que o sistema disponibilize as notas dos alunos, sendo que os mesmos teriam acesso;
7. como professor desejo poder visualizar todas as atividades já aplicadas aos alunos.

Fez-se importante a divisão de tarefas e seu detalhamento para nortear o início do desenvolvimento do sistema. Sendo essas:

1. autenticação do usuário por meio de *login*: Criar uma tabela de usuários no Banco de Dados para registrar quem terá permissão para acessar o sistema, sendo necessário esta ter um campo específico para guardar a senha do usuário. Criar uma tela para que o usuário possa realizar o *login*;
2. *crud* dos alunos: Criar uma tela para cadastrar, listar, atualizar e excluir o aluno, juntamente com a criação da tabela alunos no Banco de Dados, esta deverá conter um campo identificador e um outro para guardar a senha do aluno, entre outros geralmente utilizados neste contexto. Ao cadastrar um aluno, deverá inserir um registro na tabela de usuários, utilizando os campos semelhantes presente na tabela aluno e ao alterar ou excluir deve ser feito essas atualizações na tabela usuários;
3. *crud* dos professores: Criar uma tela para cadastrar, listar, atualizar e excluir o professor, assim como a criação da tabela professor no Banco de Dados, esta deverá conter um campo identificador e um outro para guardar a senha do professor, entre outros geralmente utilizados neste contexto. Ao cadastrar um professor, deverá inserir

- um registro na tabela de usuários utilizando os campos semelhantes presente na tabela professor e ao alterar ou excluir deve ser feito essas atualizações na tabela usuários;
4. crud de turmas: Criar uma tabela turmas no Banco de Dados e uma tela para cadastrar, listar, atualizar e excluir turmas;
 5. vinculação dos alunos: Criar uma tela para ser possível escolher os alunos e associá-los a uma turma específica já cadastrada;
 6. *crud* das atividades pelos professores: Criar uma tabela atividades no Banco de Dados, essa deverá conter um campo específico para definir a qual turma essa atividade está relacionada. Criar uma tela para cadastrar, listar, atualizar e excluir atividades;
 7. *upload* pelos alunos: Criar uma pasta no servidor, para salvar os arquivos enviados e uma tabela arquivos no Banco de Dados, essa deverá ter os campos: Identificador do aluno, da atividade e um campo para armazenar o diretório completo do arquivo. Criar funcionalidade que permita o aluno fazer *upload* de arquivo para uma atividade específica;
 8. correção automatizada: Trocar a extensão do arquivo do *Word*, docx por Zip e acessar os arquivos XML, após isso, desenvolver uma lógica para correção dos arquivos enviados pelos alunos;
 9. visualização de notas: Após as correções dos arquivos enviados pelos alunos, já terem ocorrido, é necessário que a nota respectiva do aluno, esteja disponível para que ele a visualize. Também ter que ser possível que o professor visualize a nota de todos os alunos;
 10. histórico de atividades: É necessário que seja guardado um histórico de todas as atividades que foram cadastradas no sistema, registrando as alterações realizadas na mesma, isso para proporcionar maior segurança das informações do sistema e para ser possível que o professor mantenha melhor controle sobre as atividade que ele passam aos seus alunos.

Após a modelagem iniciou-se a etapa de implementação, ou seja, de desenvolvimento do sistema, nesta etapa analisou-se o diagrama de caso de uso e classes do sistema e implementou-se às histórias de usuário, com o objetivo de se obter uma versão inicial do sistema.

4. RESULTADOS E DISCUSSÕES

Para dar início ao desenvolvimento, foi necessária a identificação de quais arquivos XML da estruturação interna do *Word* possuía as informações que seriam úteis para o objetivo do presente trabalho. As informações mais relevantes para a realização da correção relacionada à formatação do texto de um arquivo *Word* se encontram em um arquivo chamado *document.xml* localizado dentro da pasta *word*, sendo possível acessar esse arquivo ao se trocar a extensão de um documento *Word*, ou seja, *.docx* para *.zip*, o que o torna manipulável como arquivo compactado comum, no qual é possível acessar e visualizar o arquivo XML. Um exemplo de um arquivo *document.xml* completo está disponibilizado no Apêndice A.

No arquivo *document.xml*, se encontram muitas informações importantes, desde o próprio texto como todas as propriedades do mesmo. Existem algumas configurações padrão que são aquelas que ao se criar um documento *Word* elas já estão aplicadas, como exemplo a fonte é Calibri, o tamanho de fonte é 11 e o alinhamento está à esquerda. Por serem configurações padrão, elas não são armazenadas inicialmente no arquivo *document.xml*, ou seja, caso não exista nesse arquivo a *tag* referente à fonte utilizada, sabe-se então que a utilizada foi a padrão e esta lógica se aplica a todas as configurações padrão.

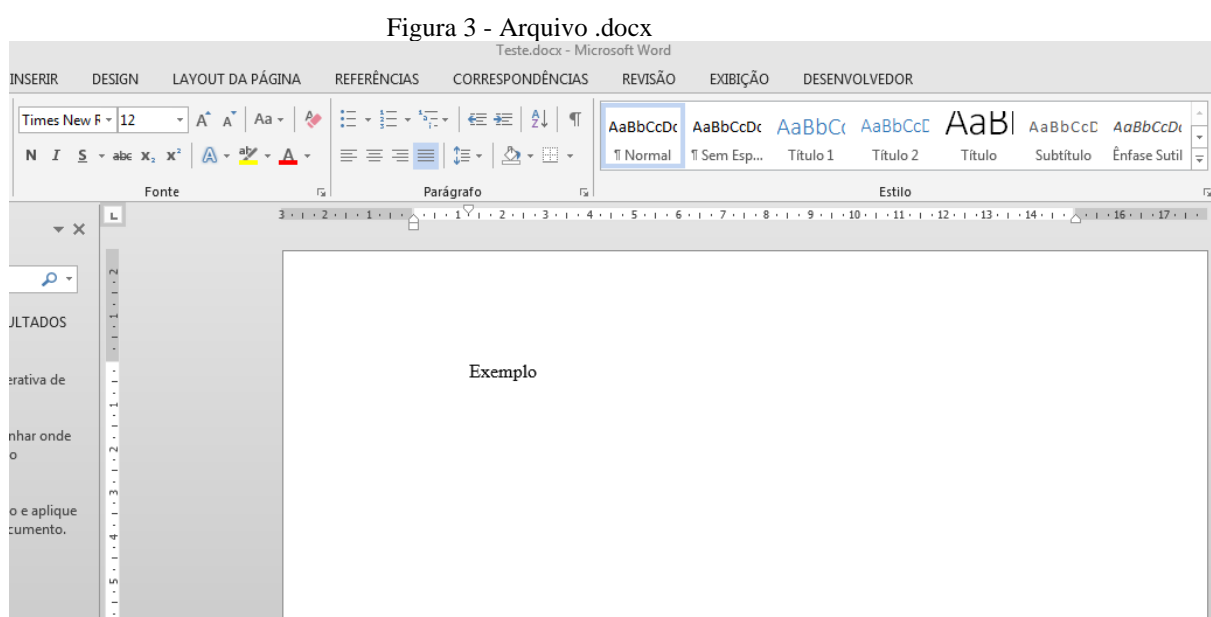
Algumas das informações de um documento *docx* são registradas no arquivo *document.xml*, de forma distinta, como por exemplo: O tamanho da fonte que é registrado com o valor ao dobro, ou seja, se no documento *Word* o texto possui o tamanho 12, no arquivo XML estará registrado como 24, em se tratando do alinhamento do texto é feito o registro em inglês: *both* refere-se ao alinhamento justificado, *right* a direita, *center* centralizado e, como já citado anteriormente, o alinhamento padrão é a esquerda, por isso não é armazenado.

A unidade de medida utilizada no arquivo XML do *document.xml* é *twip*, ou seja, o valor que for definido no documento *.docx* é registrado no *document.xml* em *twips*, 567 *twips* equivale a 1 cm. Por exemplo, ao selecionar o recuo inicial do parágrafo como direita 1 cm e esquerda 1,5 cm, no arquivo XML estará registrado nas *tags* *w:right=851* e *w:left=567*, essas *tags* são respectivamente o recuo a direita e esquerda do parágrafo inicial, sendo que 1cm convertido em *twips* equivale a 567(1x567) e 1,5 à aproximadamente 851(1,5x567). Foi observado a mesma situação no recuo especial do parágrafo, ao definir no *docx* o recuo especial do parágrafo como primeira linha 1,25 cm, no arquivo XML estará registrado na *tag* *w:firstLine=709*, visto que 1,25 cm convertido em *twips* equivale à aproximadamente 709

(1,25x567), caso seja selecionado deslocamento no recuo especial do parágrafo, está contida no *document.xml* a tag *w:hanging*, a opção nenhum é uma configuração padrão, não sendo armazenada no arquivo XML.

O espaçamento antes e depois do parágrafo é definido em ponto (pt), 1 ponto no docx equivale a 20 *twips* no *document.xml*. Ao selecionar 6 pt antes e automático no campo depois no documento docx, o *document.xml* irá conter a tag *w:before=120 w:after=100*, dado que 6 pt convertido em *twips* é 120 (6x20) e 100 é o valor automático em *twips*. No documento docx no campo espaçamento entre linhas, selecionando as opções simples, 1,5 ou duplo, terá sua representação no *document.xml* como respectivamente *w:line=240*, *w:line=360*, *w:line=480*.

Em busca de maior compreensão, será representado um exemplo de um arquivo em .docx e sua representação em XML. A Figura 3, representa a informações salva em um arquivo .docx.



Fonte: Elaborada pelos autores.

Como observado no documento .docx, a palavra “Exemplo”, apresenta a fonte *Times New Roman*, tamanho 12. Na Figura 4, é apresentada a configuração de parágrafo aplicada.

Figura 4 - Configuração de parágrafo aplicada

Parágrafo

Recuos e espaçamento **Quebras de linha e de página**

Geral

Alinhamento: Justificada

Nível do tópico: Corpo de Texto Recolhidos por padrão

Recuo

Esquerda: 0 cm Especial: Primeira linha Pos: 1,25 cm

Direita: 0 cm

Espelhar recuos

Espaçamento

Antes: 0 pt Espaçamento entre linhas: 1,5 linhas Em:

Depois: 8 pt

Não adicionar espaço entre parágrafos do mesmo estilo

Visualização

Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior Parágrafo anterior

Exemplo de como é representado um arquivo .docx no document.xml

Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte Parágrafo seguinte

Tabulação... Definir como Padrão OK Cancelar

Fonte: Elaborada pelos autores.

Após salvar o documento .docx, com as configurações apresentadas. Foi convertido o arquivo .docx para Zip. Ao acessar o arquivo Zip foi encontrada algumas pastas, dentre elas está a pasta *Word*, nesta pasta encontra-se o arquivo *document.xml*. Ao acessar este arquivo, é apresentado o XML referente ao documento .docx. A Figura 5, irá representar um exemplo.

Figura 5 - Representação XML do arquivo Word

```

- <w:body>
- <w:p w:rsidP="00254FE5" w:rsidRDefault="00FA2B98" w:rsidRPr="002360CE" w:rsidR="005A436D">
- <w:pPr>
  <w:spacing w:lineRule="auto" w:line="360"/>
  <w:ind w:firstLine="709"/>
  <w:jc w:val="both"/>
- <w:rPr>
  <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman" w:ascii="Times New Roman"/>
  <w:sz w:val="24"/>
  <w:szCs w:val="24"/>
</w:rPr>
</w:pPr>
- <w:r>
- <w:rPr>
  <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman" w:ascii="Times New Roman"/>
  <w:sz w:val="24"/>
  <w:szCs w:val="24"/>
</w:rPr>
  <w:t>Exemplo</w:t>
</w:r>
  <w:bookmarkStart w:name="_GoBack" w:id="0"/>
  <w:bookmarkEnd w:id="0"/>
</w:p>
- <w:sectPr w:rsidRPr="002360CE" w:rsidR="005A436D">
  <w:pgSz w:w="11906" w:h="16838"/>
  <w:pgMar w:gutter="0" w:footer="708" w:header="708" w:left="1701" w:bottom="1417" w:right="1701" w:top="1417"/>
  <w:cols w:space="708"/>
  <w:docGrid w:linePitch="360"/>
</w:sectPr>
</w:body>
</w:document>

```

Fonte: Elaborada pelos autores.

Nesta figura é apresentada informações referentes a fonte, tamanho, alinhamento, espaçamento em linhas, entre outras informações do texto.

As tabelas de configurações de título e texto utilizadas neste projeto, foram criadas utilizando o modulo *migrations*. Este não utiliza o SQL para atualização do banco de dados, todas as inserções/alterações das informações são feitas pela aplicação, utilizando código PHP, foi implementada classes com instruções para inserir/atualizar as informações das tabelas presente no banco de dados. É utilizado arquivos de *migrations* para cada tabela utilizando o comando cmd: “php index.php *migrate generate* nome_da_tabela”, por meio deste comando é criado a tabela nome_da_tabela, depois de todas as tabelas criadas utiliza-se o comando: “php index.php *migrate*”, este irá executar as tabelas no banco de dados.

4.1 VERSÃO INICIAL

Para a versão inicial foram selecionados dois requisitos, correção automatizada e *upload* de arquivos. Pelo fato do período estipulado para o desenvolvimento do projeto ser curto, concentrou-se os estudos e a implementação na parte mais complexa do projeto, a correção automatizada, para oferecer suporte e testar a validade da correção automatizada,

implementou-se o *upload* de arquivos. Estes requisitos validados nortearão possíveis implementações futuras.

Para a confecção da logomarca do sistema, Figura 6, foi considerado a utilidade do sistema. SYS é uma abreviação de *systems*, traduzida para o português significa sistema e *Correction*, traduzida para o português significa correção. A união destas palavras originou “*SysCorrection*”, nome este que faz menção a um sistema de correção. As cores azul e branca foram escolhidas por serem padrão do *Word*.

Figura 6 - Logomarca



Fonte: Elaborada pelos autores.

A Figura 7 apresenta a tela inicial de configurações, nela está contido dois botões, “configurações do título” e “configurações do texto”.

Figura 7 - Tela inicial de configurações



Fonte: Elaborada pelos autores.

Ao clicar no botão “configurações do título”, é apresentada uma tela, solicitando o preenchimento dos dados referente a configurações do título conforme a Figura 8.

Figura 8 - Tela de configurações do título

Configurações do Título ×

Fonte

Arial ▾

Tamanho

8 ▾

Negrito?

Sim
 Não

Itálico?

Sim
 Não

Sublinhado?

Sim
 Não

Alinhamento

Esquerda ▾

Fonte: Elaborada pelos autores.

Esta tela é composta pelos seguintes campos: “Fonte”, “Tamanho”, “Negrito?”, “itálico?”, “Sublinhado?” e “Alinhamento”, e os demais botões “Cancelar” e “Salvar configurações”. Após o preenchimento de todos os campos de forma correta, há duas opções para finalizar a tela, o usuário poderá clicar em “Cancelar” e ser redirecionado para a tela inicial de configurações ou “Salvar configurações” e ser redirecionado para a tela inicial de configurações com as respectivas configurações salvas. O ícone *check* ficará cinza caso não exista configuração no banco, ou seja, não tenha salvo nenhuma configuração de título, se existir configuração de título cadastrada no banco, o ícone *check* ficará verde, a Figura 9 representa estas situações.

Figura 9 - Configurações do título registradas com sucesso



Fonte: Elaborada pelos autores.

Ao clicar no botão “Configurações do texto”, é apresentada uma tela solicitando o preenchimento dos dados para configurações do texto conforme a Figura 10.

Figura 10 - Tela de configurações do texto

Configurações do texto ✕

Fonte
Opções... ▾

Tamanho
Escolher... ▾

Negrito?
 Sim
 Não

Itálico?
 Sim
 Não

Sublinhado?
 Sim
 Não

Alinhamento
Opções... ▾

Recuo(início do parágrafo)
Esquerda Direita

Espaçamento entre parágrafos
Antes Depois

Espaçamento entre linhas
Opções... ▾

Especial
Opções... ▾

Fonte: Elaborada pelos autores.

Esta tela possui os seguintes campos: “Fonte”, “Tamanho”, “Negrito?”, “Itálico?”, “Sublinhado?”, “Alinhamento”, “Recuo”, “Espaçamento entre parágrafos”, “Espaçamento entre linhas” e “Especial”, além dos demais botões “Cancelar” e “Salvar configurações”. Após o preenchimento de todos os campos de forma correta, segue-se os mesmos passos da tela de configurações de título, há duas opções para finalizar a tela, o usuário poderá clicar em “Cancelar” e ser redirecionado para a tela inicial de configurações ou “Salvar configurações” e ser redirecionado para a tela inicial de configurações com as respectivas configurações salvas. Após isso o *check* ficará verde, mostrando ao usuário que há configuração salva no banco, Figura 11.

Figura 11 - Configurações do texto registradas com sucesso



Fonte: Elaborada pelos autores.

Na tela inicial há uma seta para avançar para a próxima página, ao usuário clicar irá direcioná-lo para a tela de *upload*, Figura 12.

Figura 12 - Avançar para próxima página



Fonte: elaborada pelos autores.

Realizada as configurações de título e texto, o usuário terá as configurações desejadas registradas no banco de dados, na quais serão utilizadas para realização das correções dos arquivos que serão enviados. Na tela de *upload* há um campo que utiliza a biblioteca *dropzone*, está possibilita fazer *upload* de arquivos com visualização de imagens, de forma que ao clicar e arrastar um arquivo no quadro “Clique ou arraste seus arquivos para cá!”, ele é enviado, Figura 13.

Figura 13 - Tela de Upload de arquivos



Fonte: elaborada pelos autores.

Ao selecionar e arrastar um arquivo para dentro do quadro “Clique ou arraste seus arquivos para cá!” ou clicar nele e selecionar os arquivos e mandar abrir, é exibido o ícone do *Word* para os arquivos que foram enviados, conforme a Figura 14.

Figura 14 - Enviar arquivos para correção



Fonte: Elaborada pelos autores.

Ao passar o mouse sobre o ícone do *Word* dos arquivos que foram enviados, é mostrado as informações relativas daquele arquivo, sendo: o tamanho do arquivo e seu nome. Ao tirar o mouse de cima do ícone essas informações desaparecem novamente, como mostra a Figura 15.

Figura 15 - Efeito ao passar o mouse em cima



Fonte: Elaborada pelos autores.

Neste momento o documento .docx é enviado para uma pasta no servidor nomeada como “arquivos”. Nesta pasta ficarão salvos todos os documentos .docx enviados para correção. Pode-se ser enviar mais de um arquivo para correção por vez, pois a correção é executada de maneira independente sobre cada um deles.

Após a correção é exibido uma tela com o relatório da correção, neste é detalhado os acertos e erros encontrados, tanto do título quanto texto, como mostra a Figura 16. Essa correção é realizada de acordo com as configurações que estão no banco de dados.

Figura 16 - Relatório



Arquivo	Mensagem
teste27.docx	<p>Erros do Título: Fonte do título: Times New Roman. Tamanho da fonte: 12. Título negrito: Não. Título itálico: Não. Título sublinhado: Não.</p> <p>Está tudo correto com o texto!</p>
teste28.docx	<p>Erros do Título: Fonte do título: Times New Roman. Tamanho da fonte: 12. Título negrito: Não. Título itálico: Não. Título sublinhado: Não.</p> <p>Está tudo correto com o texto!</p>

Fonte: Elaborada pelos autores.

A tabela relatório possui duas colunas, uma para registrar o nome do arquivo que foi corrigido e a outra para mostrar a mensagem da correção referente ao respectivo arquivo.

5. CONSIDERAÇÕES FINAIS

No presente trabalho foi analisado e estudado o processo de correção de avaliações referentes a informática básica da ferramenta *Word*. Teve-se como objeto de estudo os professores de informática básica do IFMG-SJE. Por meio do estudo e observação foi averiguada a presente necessidade de otimização e automatização dos processos de correção de avaliações.

Este trabalho foi desenvolvido com objetivo de elaborar e validar uma proposta de uma plataforma *Web (software)* para realizar a correção automática de avaliações realizadas no *Microsoft Word*.

Os objetivos e etapas propostos neste trabalho foram alcançados, visto que foi compreendido o processo de correção manual realizado pelos professores, documentado, levantado os requisitos do sistema, compreendido a estruturação dos documentos *Word* e implementado uma versão inicial para validação. O sistema em sua versão inicial realiza a validação de correções de texto e título em um documento *Word*, a partir da entrada das informações requeridas pelo professor e retorna ao usuário se a resposta está correta ou não. Para a verificação se o texto e título estão de acordo com o requerido pelo professor foi feita algumas verificações como nome e tamanho da fonte, alinhamento, recuo do parágrafo à esquerda, à direita e especial, espaçamento antes e depois do parágrafo e também espaçamento entre linhas.

Como este estudo foi um modelo inicial para correção de avaliações de *Word*, irá auxiliar como suporte para implementações futuras. O diagrama de caso de uso e o diagrama de classes elaborados neste trabalho, foram para nortear essas implementações futuras. Espera-se nas próximas versões, a realização do teste de aceitação da versão inicial do trabalho, liberação das funcionalidades para vinculação do banco de dados com cadastro para os alunos e professores, sessão para *login* do usuário (professor e aluno), telas específicas para o professor e aluno, relatório de desempenho e atribuição de notas as avaliações, além da correção de documentos com estruturação mais complexa do que aqueles usados até então.

REFERÊNCIAS

ALBINO, Raphael Donaire; SOUZA, Cesar Alexandre de; PRADO, Edmir Parada Vasques. **Benefícios alcançados através de um modelo de Gestão Ágil de Projeto em uma empresa de jogos eletrônicos**. 2013. 15 f. TCC (Graduação) - Curso de Administração, Usp Universidade de São Paulo, São Paulo, 2013.

APACHE FRIENDS. **Sobre**. Disponível em: <https://www.apachefriends.org/pt_br/about.html>. Acesso em: 11 jun. 2019.

ATOM. Disponível em: <<https://atom.io/>>. Acesso em: 17 maio 2019.

BITBUCKET. Disponível em: <<https://bitbucket.org/product>>. Acesso em: 18 maio 2019.

BOOTSTRAP. Disponível em: <<https://getbootstrap.com/docs/4.3/about/overview/>>. Acesso em: 11 abr. 2019.

BORTOLI, N. d. S. de; RUFINO, R. R. **Conceito para o desenvolvimento web utilizando spring boot, bootstrap e angular js**. 2016.

CHAVES, A. M.; SILVA, G. D. **Proposta de uma arquitetura de software e funcionalidades para implementação de um ambiente integrado de desenvolvimento para a Linguagem PHP**. In: I JORNADA CIENTÍFICA E VI FIPA DO CEFET, 2008, Bambuí. Disponível em: <http://www.bambui.ifmg.edu.br/jornada_cientifica/str/artigos_aprovados/informatica/68-CO-5.pdf>. Acesso em: 17 maio 2019.

CODEIGNITER. **Guia do Usuário do Code Igniter Versão 2.2.6**. Disponível em: <<https://codeigniter.com/userguide2/>>. Acesso em: 19 maio 2019.

DEVMEDIA. **Agile development: XP e Scrum em uma abordagem comparativa**. Disponível em: <<https://www.devmedia.com.br/agile-development-xp-e-scrum-em-uma-abordagem-comparativa/30808>>. Acesso em: 04 maio 2019.

DEVMEDIA. **Introdução ao framework PHP CodeIgniter**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-framework-php-codeigniter/27346>>. Acesso em: 11 jun. 2019.

ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados**. 6a ed. São Paulo: Person, 837 p. 2011.

GUEDES, Gilleanes TA. **UML 2: Guia Prático**. 2. ed. São Paulo: Novatec, 2014.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

GIORDANE, Rubie José. **Word 2010**. Clube de Autores. p. 111. 2014.

GIT. Disponível em: <<https://git-scm.com/>>. Acesso em: 18 maio 2019.

GUDWIN, Ricardo R. **Introdução à Linguagem UML**. 2010. Disponível em:<<http://www.dca.fee.unicamp.br/~gudwin/ftp/ea976/Estruturais2010.pdf>>. Acesso em: 20 abr. 2019.

JQUERY. **O que é JQUERY?**. Disponível em: <<http://jquery.com/>>. Acesso em: 20 maio 2019.

KAUARK, F. S.; MANHÃES, F. C.; MEDEIROS, C. H. **Metodologia da pesquisa: um guia prático**. Bahia: Via Litterarum, 2010.

LIMA, Greick Roger de Carvalho. Benefícios das metodologias ágeis no gerenciamento de projetos de Tecnologia da Informação (TI). **Revista Especialize on-line**, Goiás, dez. 2015. Disponível em: <<https://www.ipog.edu.br/revista-especialize-online/edicao-n10-2015/beneficios-das-metodologias-ageis-no-gerenciamento-de-projetos-de-tecnologia-da-informacao-ti/>>. Acesso em: 05 ago. 2019.

LIMA, Igor Ribeiro; FREIRE, T. C.; COSTA, Heitor Augustus Xavier. **Implantação e Adaptação do Scrum em um Laboratório de Pesquisa e Desenvolvimento de Projetos de Software**. Revista de Sistema de Informação, Macaé, n. 9, p. 16-23, 2012. Disponível em: <http://www.fsma.edu.br/si/edicao9/FSMA_SI_2012_1_Principal_2.pdf>. Acesso em: 05 maio 2019.

MAINART, D. A.; SANTOS, C. M. **Desenvolvimento de Software: Processos Ágeis ou Tradicionais? Uma visão crítica**. In: VII ENCONTRO ANUAL DE COMPUTAÇÃO, 2010.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas, 2010.

MEIRE, Msc Regilan. **Banco de Dados**, 2016. Disponível em: <<http://www.regilan.com.br/wp-content/uploads/2013/10/Apostila-Banco-de-Dados.pdf>>. Acesso em: 18 abr. 2019.

MICROSOFT. **O que é o Word?** 2019. Disponível em: <<https://support.office.com/pt-br/article/o-que-%C3%A9-o-word-ae9c7ff-f9c5-415f-80dc-103ad5e344d7?ui=pt-BR&rs=pt-BR&ad=BR>>. Acesso em: 22 abr. 2019.

MICROSOFT. **Diferenças entre o formato de Texto OpenDocument (.odt) e o formato do Word (.docx)**, 2019. Disponível em: <<https://support.office.com/pt-br/article/diferen%C3%A7as-entre-o-formato-de-texto-opendocument-odt-e-o-formato-do-word-docx-d9d51a92-56d1-4794-8b68-5efb57aebfdc>>. Acesso em: 22 abr. 2019.

MICROSOFT. **Introdução à compatibilidade de marcação (Open XML SDK)**, 2017. Disponível em: <<https://docs.microsoft.com/pt-br/office/open-xml/introduction-to-markup-compatibility>>. Acesso em: 22 abr. 2019.

MICROSOFT. **Bem-vindo ao Open XML SDK 2.5 para Office**, 2017. Disponível em: <<https://docs.microsoft.com/pt-br/office/open-xml/open-xml-sdk>>. Acesso em: 22 abr. 2019.

MICROSOFT. **Criar suplementos melhores para o Word com o Office Open XML Create better add-ins for Word with Office Open XML**, 2019. Disponível em: <<https://docs.microsoft.com/pt-br/office/dev/add-ins/word/create-better-add-ins-for-word-with-office-open-xml>>. Acesso em: 22 abr. 2019.

MYSQL. **Capítulo 1 Informações Gerais**, 2019. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>>. Acesso em: 19 abr. 2019.

NETBEANS. **Recursos do NetBeans IDE**, 2019. Disponível em: <<https://netbeans.org/features/index.html>>. Acesso em: 17 maio 2019.

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. 2. ed. São Paulo: Novatec, 2013. 299 p.

OLIVEIRA, Aristóteles da Silva. Perspectivas para formação de professores. In: MERCADO, Luís Paulo Leopoldo (Org.). **Percursos na Formação de Professores com Tecnologias da Informação e Comunicação na Educação**. 1. ed. – EDUFAL. 2007.
PHP. **O que é PHP ?**, 2019. Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 18 abr. 2019.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011. 780 p.

PRESSMAN, R. S.; MAXIN, B. R. **Engenharia de Software: uma abordagem profissional**. 8. ed. São Paulo: McGraw Hill Brasil, 2016.

ROB, Peter; CORONEL, Carlos. **Sistemas de banco de dados: projeto, implementação e gerenciamento**. Tradução: All Tasks. Revisão Técnica: Ana Paula Appel. 8. ed. Norte americana. São Paulo: Cengage Learning, p. 7, 2011. Disponível em:<https://issuu.com/cengagebrasil/docs/9788522107865_sistemas-de-banco-de-dados> Acesso em: 18 abr. 2019.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.

SCHWABER, Ken; SUTHERLAND, Jeff. Um guia definitivo para o Scrum: **As regras do jogo**, 2017. Disponível em:<<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 23 abr. 2019.

SILVA, Carlos Eduardo Azevedo Costinhas da. **Um estudo de caso sobre a adoção de práticas ágeis em um ambiente tradicional**. 2013. 103 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, 2013.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?**, 2015. Disponível em:<<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. Acesso em: 18 abr. 2019.

SOLID IT. **Classificação DB-Engines**, 2019. Disponível em: < <https://db-engines.com/en/ranking> >. Acesso em: 02 ago. 2019.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Ed., São Paulo: Addison Wesley Brasil, 2007.

SOMMERVILLE, I. **Engenharia de software**. Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves. 9. ed. São Paulo: Pearson Prentice Hall, 2011. p. 529.

SOURCETREE. Simplicidade e poder em uma linda GUI do Git, 2019. Disponível em: <<https://www.sourcetreeapp.com/>>. Acesso em: 18 maio 2019.

SOUZA, Luciano Malaquias de. **Academos: Método Ágil XP (Extreme Programming)**. v. 3, n. 3, p. 3, jul. /dez. 2007. Disponível em: <http://intranet.fainam.edu.br/aceso_site/fia/academos/revista3/6.pdf>. Acesso em: 23 abr. 2019.

SOUSA, Artur Afonso de; PEREIRA, José Luís; CARVALHO, João Álvaro. A Linguagem XML numa Perspectiva de Bases de Dados. In: ATAS DA CONFERÊNCIA DA ASSOCIAÇÃO PORTUGUESA DE SISTEMAS DE INFORMAÇÃO. 2016.

SUBLIME HQ. **Conceitos básicos**, 2019. Disponível em: <https://sublime-text-unofficial-documentation.readthedocs.io/en/latest/basic_concepts.html#overview>. Acesso em: 17 maio 2019.

TECMUNDO. **O que é css ?**, 2009. Disponível em: <<https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm> >. Acesso em: 12 abr. 2019.

TECHTUDO. **Editor de código HTML com interface personalizada**, 2012. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/sublime-text.html>>. Acesso em: 17 maio 2019.

TECHTUDO. **O que é Xampp e para que serve**, 2012. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/02/o-que-e-xampp-e-para-que-serve.html> >. Acesso em: 11 jun. 2019.

TIOBE. **TIOBE Index for August 2019**, 2019. Disponível em: <<https://www.tiobe.com/tiobe-index/> >. Acesso em: 02 ago. 2019.

VIDOR, Joe Jackson Sangalli *et al.* A Gestão dos Processos de Produção e as Parcerias Globais para o Desenvolvimento Sustentável dos Sistemas Produtivos. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 33 ., 2013, Salvador. **Análise das metodologias ágeis XP e Scrum no desenvolvimento de um software de gerenciamento de abastecimento**. 9 p. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2013_TN_STP_184_049_22827.pdf>. Acesso em: 04 maio 2019.

W3C. **HTML 5,3**, 2018. Disponível em: <<https://www.w3.org/TR/html53/introduction.html#html-vs-xhtml> >. Acesso em: 12 abr. 2019.

W3C. **Extensible Markup Language (XML)**, 2016. Disponível em: <<https://www.w3.org/XML/>>. Acesso em: 12 abr. 2019.

W3C. Página inicial das folhas de estilo em cascata, 2019. Disponível em:
<<https://www.w3.org/Style/CSS/Overview.en.html#translations>>. Acesso em 22 maio 2019.

W3SCHOOLS. Tutorial CSS, 2019. Disponível em:
<<https://www.w3schools.com/css/default.asp>>. Acesso em: 12 abr. 2019.

W3SCHOOLS. Tutorial Bootstrap 3, 2019.
Disponível em: <<https://www.w3schools.com/bootstrap/default.asp>>. Acesso em: 12 abr. 2019.

W3SCHOOLS. Tutorial JavaScript, 2019.
Disponível em: <<https://www.w3schools.com/js/default.asp>>. Acesso em: 18 abr. 2019.

APÊNDICES

APÊNDICE A – Exemplo de um arquivo document.xml.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<w:document mc:Ignorable="w14 w15 w16se w16cid wp14"
xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingShape"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk"
xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wordprocessingGroup"
xmlns:w16se="http://schemas.microsoft.com/office/word/2015/wordml/synex"
xmlns:w16cid="http://schemas.microsoft.com/office/word/2016/wordml/cid"
xmlns:w15="http://schemas.microsoft.com/office/word/2012/wordml"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:am3d="http://schemas.microsoft.com/office/drawing/2017/model3d"
xmlns:aink="http://schemas.microsoft.com/office/drawing/2016/ink"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:cx8="http://schemas.microsoft.com/office/drawing/2016/5/14/chartex"
xmlns:cx7="http://schemas.microsoft.com/office/drawing/2016/5/13/chartex"
xmlns:cx6="http://schemas.microsoft.com/office/drawing/2016/5/12/chartex"
xmlns:cx5="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex"
xmlns:cx4="http://schemas.microsoft.com/office/drawing/2016/5/10/chartex"
xmlns:cx3="http://schemas.microsoft.com/office/drawing/2016/5/9/chartex"
xmlns:cx2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex"
xmlns:cx1="http://schemas.microsoft.com/office/drawing/2015/9/8/chartex"
xmlns:cx="http://schemas.microsoft.com/office/drawing/2014/chartex"
xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas">
  - <w:body>
    - <w:p w:rsidP="007519CA" w:rsidRDefault="007519CA" w:rsidR="003F56DC"
      w14:textId="77777777" w14:paraId="3F011B7D">
      - <w:pPr>
        <w:jc w:val="center"/>
        - <w:rPr>
          <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
            w:ascii="Times New Roman"/>
          <w:b/>
          <w:bCs/>
          <w:sz w:val="24"/>
          <w:szCs w:val="24"/>
        </w:rPr>
      </w:pPr>
    - <w:r w:rsidRPr="007519CA">
      - <w:rPr>
        <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
          w:ascii="Times New Roman"/>
        <w:b/>
        <w:bCs/>
        <w:sz w:val="24"/>
        <w:szCs w:val="24"/>
      </w:rPr>
    </w:r>
  </w:p>
  </w:body>
</w:document>

```

```

        </w:rPr>
        <w:t>Isso é um exemplo de um título</w:t>
    </w:r>
</w:p>
- <w:p w:rsidP="007519CA" w:rsidRDefault="007519CA" w:rsidR="007519CA"
w14:textId="77777777" w14:paraId="28886261">
  - <w:pPr>
    - <w:rPr>
      <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
w:ascii="Times New Roman"/>
      <w:b/>
      <w:bCs/>
      <w:sz w:val="24"/>
      <w:szCs w:val="24"/>
    </w:rPr>
  </w:pPr>
</w:p>
- <w:p w:rsidP="0087138E" w:rsidRDefault="007519CA" w:rsidR="007519CA"
w14:textId="79A1A53B" w14:paraId="7D755BC7" w:rsidRPr="0087138E">
  - <w:pPr>
    <w:spacing w:lineRule="auto" w:line="360" w:after="240" w:before="120"/>
    <w:ind w:firstLine="709"/>
    <w:jc w:val="both"/>
  - <w:rPr>
    <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
  </w:rPr>
  </w:pPr>
- <w:r w:rsidRPr="007519CA">
  - <w:rPr>
    <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
  </w:rPr>
  <w:t>Isto é um exemplo de um pequeno trecho de texto</w:t>
</w:r>
- <w:r w:rsidR="003D24B4">
  - <w:rPr>
    <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
  </w:rPr>
  <w:t xml:space="preserve"> aleatório</w:t>
</w:r>
- <w:r w:rsidRPr="007519CA">
  - <w:rPr>
    <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>

```

```

        <w:szCs w:val="24"/>
    </w:Pr>
    <w:t xml:space="preserve">.</w:t>
</w:Pr>
- <w:r w:rsidR="003D24B4">
  - <w:Pr>
    <w:Fonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
      w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
    </w:Pr>
    <w:t>Este</w:t>
  </w:Pr>
- <w:r w:rsidRPr="007519CA">
  - <w:Pr>
    <w:Fonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
      w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
    </w:Pr>
    <w:t xml:space="preserve"> é um exemplo de um pequeno trecho de texto. Isto
      é um exemplo de um pequeno trecho de texto.</w:t>
  </w:Pr>
- <w:r w:rsidR="003D24B4">
  - <w:Pr>
    <w:Fonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
      w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
    </w:Pr>
    <w:t xml:space="preserve"> Sim, um pequeno trecho de texto </w:t>
  </w:Pr>
- <w:r w:rsidR="003D24B4">
  - <w:Pr>
    <w:Fonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
      w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
    </w:Pr>
    <w:t>aleatório</w:t>
  </w:Pr>
- <w:r w:rsidR="003D24B4">
  - <w:Pr>
    <w:Fonts w:cs="Times New Roman" w:hAnsi="Times New Roman"
      w:ascii="Times New Roman"/>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
    </w:Pr>
    <w:t>.</w:t>
  </w:Pr>
  <w:bookmarkStart w:id="0" w:name="_GoBack"/>
  <w:bookmarkEnd w:id="0"/>
</w:p>

```

```
- <w:sectPr w:rsidR="007519CA" w:rsidRPr="0087138E">  
  <w:pgSz w:w="11906" w:h="16838"/>  
  <w:pgMar w:gutter="0" w:footer="708" w:header="708" w:left="1701"  
    w:bottom="1417" w:right="1701" w:top="1417"/>  
  <w:cols w:space="708"/>  
  <w:docGrid w:linePitch="360"/>  
</w:sectPr>  
</w:body>  
</w:document>
```