

PROPOSTA DE JOGO EDUCACIONAL PARA UMA EMPRESA DE COMPONENTES ELÉTRICOS

**Bruna Pereira das Neves
Clístenes Grizafis Bento
Diego Murilo Sousa da Luz
Leonardo Pestilo dos Santos
Lucelia Mildemberger
Luiz Henrique Pereira Isbaes**

RESUMO

O presente trabalho teve como foco desenvolver protótipo de jogo digital para avaliação de conhecimentos em manutenção de motocicletas e aproveitar os benefícios da boa Governança de TI (GTI). O objetivo geral deste trabalho é desenvolver demonstração de jogo educacional por meio de protótipo envolvendo manutenção automotiva de motocicletas, tendo como objetivos específicos identificar na literatura quais os conceitos sobre governança corporativa, governança de TI, engenharia de requisitos, modelagem de software, aplicação web e prototipagem, verificar quais requisitos e ferramentas necessárias para o desenvolvimento do sistema, modelar sistema de acordo com os requisitos verificados, desenvolver protótipo de acordo com o modelo, discutir sobre os resultados obtidos. As metodologias adotadas foram pesquisa de campo, bibliográfica, documental e de internet com temas relevantes ao assunto. Para a fundamentação teórica com o objetivo de facilitar o entendimento do leitor sobre o assunto, encontram-se esclarecimentos sobre governança corporativa, governança de TI, engenharia de requisitos, UML, arquitetura de software, aplicação web e prototipagem. Foram feitos levantamento de requisitos e modelagem de software, que proporcionou a criação de um protótipo capaz de atender princípios da GTI bem como explorar o ensino de manutenção de motocicletas de maneira divertida.

Palavras-chave: Governança corporativa. Governança de TI. Desenvolvimento de Jogos. Engenharia de software. Jogo educacional.

SUMMARY

The present work focused on developing a digital game prototype to assess knowledge in motorcycle maintenance and take advantage of the benefits of good IT Governance (GTI). The general objective of this work is to develop an educational game demonstration through a prototype involving automotive maintenance of motorcycles, with the specific objectives of identifying in the literature which concepts about corporate governance, IT governance, requirements engineering, software modeling, web application and prototyping, verifying which requirements and tools are necessary for system development, modeling the system according to the verified requirements, developing a prototype according to the model, discussing the results obtained. The methodologies adopted were field, bibliographic, documentary and internet research with topics relevant to the subject. For the theoretical foundation with the aim of facilitating the reader's understanding of the subject, there are clarifications on corporate governance, IT governance, requirements engineering, UML, software architecture, web application and prototyping. Requirements gathering and software modeling were carried out, which led to the creation of a prototype capable of meeting GTI principles as well as exploring the teaching of motorcycle maintenance in a fun way.

KEYWORDS: Corporate governance. IT Governance. Game Development. Software Engineering. Educational game.

1. INTRODUÇÃO

O presente artigo é uma pesquisa de campo realizada visando apresentar um protótipo de jogo digital para avaliação de conhecimentos em manutenção de motocicletas e aproveitar os benefícios da boa Governança de Tecnologia da Informação (GTI), tendo como base conhecimentos adquiridos através de pesquisas bibliográficas, documentais e entrevista informal. Em sua importância pode contribuir positivamente para o desempenho da organização por meio de inovação em serviços, mercado e negócios, implementação e operação apropriadas de ativos de Tecnologia da Informação (TI) e continuidade do negócio e sustentabilidade.

Governança é um tema amplo e importante que engloba a forma como as organizações são dirigidas e controladas. A governança de TI, por sua vez, refere-se ao conjunto de processos, políticas e práticas que uma organização utiliza para gerenciar e monitorar o uso da tecnologia da informação (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2018).

Para entender a importância da governança de TI na indústria de jogos digitais, podemos citar os autores Jóhannesson e Söderström (2013), que afirmam que a governança de TI é essencial para garantir que os processos de negócios em uma organização sejam suportados pela tecnologia da informação. Além disso, os autores Zegreanu e Robu (2017) destacam que a governança de TI pode ajudar a garantir a qualidade e a segurança dos produtos de *software*, incluindo jogos digitais.

1.1 CONTEXTO DA SITUAÇÃO NA EMPRESA

A empresa de componentes elétricos está há mais de 60 anos no mercado de autopeças, produzindo uma ampla variedade de produtos, desde baterias e pneus até sistemas de ignição e iluminação. Ao longo dos anos, a empresa se tornou uma referência no setor, com forte presença no mercado nacional e também exportando para diversos países da América Latina e África. A empresa investe constantemente em tecnologia e inovação pra manter sua posição de liderança no mercado e oferecer produtos de alta qualidade e confiabilidade para seus clientes.

ARTIGO

Desde sua fundação, a empresa vem se destacando no mercado pela sua visão empreendedora e pela capacidade de inovar. Na década de 1990, por exemplo, lançou a primeira bateria automotiva selada produzida no Brasil, revolucionando o mercado de baterias para veículos. Além disso, a empresa tem um compromisso com a responsabilidade social e ambiental, investindo em ações de sustentabilidade e apoiando projetos sociais em diversas comunidades. Com uma trajetória de sucesso, a empresa busca estar à frente das tendências do mercado e das necessidades dos seus clientes.

Focada na produção de equipamentos de comunicação e automação, os processos da empresa envolvem desde o design e desenvolvimento de produtos até a fabricação, testes e controle de qualidade. A empresa possui um amplo portfólio de produtos, incluindo bobinas de ignição, motores, sensores, velas de ignição e outros.

A empresa possui também parcerias com empresas de manutenção elétrica de motocicletas e proporciona treinamento para futuros profissionais na área de manutenção. Com a intenção de investir em inovação e focar na continuidade e sustentabilidade do negócio, a empresa está buscando meios para a criação de um jogo digital, capaz de realizar avaliação de conhecimentos em manutenção elétrica de motocicletas, bem como proporcionar diversão em divulgações da empresa em feiras e exposições.

1.2 OBJETIVOS

Os objetivos estão separados em geral e específicos.

1.2.1 Objetivo Geral

Desenvolver demonstração de jogo educacional por meio de protótipo envolvendo manutenção automotiva de motocicletas.

1.2.2 Objetivos Específicos

ARTIGO

- a) identificar na literatura quais os conceitos sobre governança corporativa, governança de TI, engenharia de requisitos, modelagem de *software*, prototipagem e aplicação web;
- b) verificar quais requisitos e ferramentas necessárias para o desenvolvimento do sistema;
- c) modelar sistema de acordo com os requisitos verificados;
- d) desenvolver protótipo de acordo com o modelo;
- e) discutir sobre resultados obtidos.

1.3 JUSTIFICATIVA

A governança de TI é essencial para garantir que a tecnologia seja usada de maneira eficaz e alinhada aos objetivos estratégicos da organização. No contexto dos jogos digitais, a governança de TI é ainda mais importante, uma vez que as empresas deste setor precisam lidar com uma grande quantidade de dados sensíveis dos usuários e garantir a segurança e privacidade dos mesmos. Segundo Weill e Ross (2004), a governança de TI é uma maneira de garantir que a tecnologia seja usada de forma eficaz para apoiar a estratégia de negócios da empresa. Eles argumentam que a governança de TI deve ser vista como uma responsabilidade de toda a empresa, não apenas do departamento de TI, e que deve haver um equilíbrio entre a flexibilidade e a padronização na implementação de soluções tecnológicas.

De acordo com o relatório "Global Games Market Report" de 2021, o mercado de jogos digitais deve gerar uma receita de US\$ 175,8 bilhões em 2021, o que mostra a importância econômica desse setor. Nesse sentido, a governança de TI se torna fundamental para garantir a continuidade dos negócios, além de proteger a privacidade e a segurança dos usuários. Autores como Weill e Ross (2004) e Bechara e Sampaio (2014) destacam a importância da governança de TI para garantir a eficácia dos processos de gestão e a segurança dos dados das empresas.

Tendo em vista que a governança de TI é essencial para garantir a eficácia dos processos de gestão e a segurança dos dados das empresas, especialmente no

ARTIGO

contexto dos jogos digitais, onde a privacidade e a segurança dos usuários são fundamentais. destacam a importância da governança de TI para garantir a eficácia dos processos de gestão e na implementação de soluções tecnológicas buscando garantir a continuidade dos negócios da empresa.

1.4 METODOLOGIA

A pesquisa presente foi conduzida por meio da metodologia de pesquisa de campo, que envolve a observação de fatos e fenômenos, levantamento de dados e análise (MARCONI; LAKATOS, 2017). Gil (2017) destaca que essa metodologia também envolve a interação entre os membros do grupo pesquisado, utilizando técnicas de observação e questionamento para coletar informações.

Utilizou-se pesquisa bibliográfica com o objetivo de identificar os principais conceitos abordados neste trabalho, os quais são definidos por Gil (2017) e Marconi e Lakatos (2017) como aqueles desenvolvidos a partir de materiais já publicados, tais como artigos científicos, livros, ensaios críticos, entre outros.

A coleta de dados foi realizada por meio de uma entrevista informal com o responsável da empresa, que é definido por Gil (2017) como uma entrevista que tem como objetivo a coleta de dados, distinguindo-se de uma simples conversa. Marconi e Lakatos (2017) enfatizam que essa estrutura tem vantagens e limitações, sendo uma vantagem o fato de poder ser utilizada em todos os segmentos da população, mas uma desvantagem é a possibilidade de reter dados importantes.

A análise dos dados coletados foi realizada por meio da técnica 5W2H, que, conforme Lisboa e Godoy (2012), permite identificar os dados e rotinas mais importantes para uma unidade de produção ou projeto por meio de sete perguntas conforme ilustra o quadro 1.

QUADRO 1: DESCRIÇÃO 5W2H

5W	<i>What</i> (O que?)	Qual a atividade? Qual é o assunto? O que deve ser medido? Quais os resultados dessa atividade? Quais atividades são necessárias para o início da tarefa? Quais insumos necessários?
	<i>Who</i> (Quem?)	Quem conduz a operação? Qual a equipe responsável?

ARTIGO

		Quem executará determinada atividade? Quem depende da execução da atividade? A atividade depende de quem para ser iniciada?
	<i>Why</i> (Por quê)	Por que a operação é necessária? Ela pode ser omitida? Por que A, B e C foram escolhidos para executar a atividade?
	<i>Where</i> (Onde?)	Onde a operação será conduzida? Em que lugar? Onde a atividade será executada? Onde serão feitas as reuniões presenciais da equipe?
	<i>When</i> (Quando?)	Quando será feito? Quando será o início da atividade? Quando será o término? Quando serão as reuniões presenciais?
2H	<i>How</i> (Como?)	Como conduzir a operação? De que maneira? Como a atividade será executada? Como A, B e C vão interagir para executar esta atividade?
	<i>How Much</i> (Quanto?)	Quanto custa realizar a mudança? Quanto custa a operação atual? Qual é a relação custo/benefício? Quanto tempo está previsto?

FONTE: ADAPTADO DE LISBÔA E GODOY (2012, p. 37)

A aplicação do 5W2H pode ser muito útil para garantir que um projeto seja executado de maneira eficiente e eficaz, além de ajudar a minimizar riscos e evitar retrabalhos. Ao responder as sete perguntas, é possível identificar com clareza quais são os objetivos do projeto, quem são as pessoas responsáveis por cada etapa, quais recursos serão necessários e qual o prazo para a conclusão (GODINHO, 2010).

Com base na análise dos dados, um modelo de *software* foi desenvolvido e o protótipo do sistema foi criado na etapa seguinte.

Para Ambrose e Harris (2011) o protótipo oferece a oportunidade de testar uma ideia com o objetivo de verificar se a mesma obtém êxito na prática. Sendo que esta é parte essencial para o processo de desenvolvimento de um produto, permitindo avaliação de sua funcionalidade para que seja possível fazer a produção definitiva (MACHADO, 2013).

Finalmente, as discussões sobre os resultados obtidos nas pesquisas, levantamento de requisitos, modelagem e prototipagem do sistema foram conduzidas na última etapa do trabalho, seguido das considerações finais e referencial teórico.

2. FUNDAMENTAÇÃO TEÓRICA

Para proporcionar melhor compreensão sobre os assuntos abordados, o presente capítulo está dividido em cinco partes, sendo elas governança corporativa, governança de TI, engenharia de requisitos, UML, arquitetura de *software*, aplicação *web* e prototipagem.

2.1 GOVERNANÇA CORPORATIVA

A governança corporativa refere-se às práticas e estruturas que as empresas utilizam para gerenciar e controlar seus negócios, visando maximizar o valor para seus acionistas e outros stakeholders. Segundo a definição de Kim e Nofsinger (2016), a governança corporativa consiste em um conjunto de princípios, políticas, procedimentos e práticas que são utilizados para dirigir e controlar a empresa.

Para Monks e Minow (2011), a boa governança corporativa é essencial para o sucesso sustentável das empresas, reforçando que as empresas com governança corporativa forte superam suas concorrentes em rentabilidade, longevidade, contribuição e valor para a economia em geral. Eles também enfatizam a importância da transparência e responsabilidade dos gestores e do conselho de administração, que devem atuar no melhor interesse da empresa e de seus *stakeholders*.

2.2 GOVERNANÇA DE TI

A governança de TI é um conjunto de princípios, objetivos e estrutura que visa alinhar a tecnologia da informação com os objetivos estratégicos da organização e garantir o uso eficiente e eficaz dos recursos de TI. Segundo Weill e Ross (2004), a governança de TI deve ser orientada para o valor e fornecer uma estrutura clara de responsabilidades e processos decisórios. Já Lopes e Vasconcelos (2010) destacam a importância da governança de TI na mitigação de riscos e na garantia da conformidade com as normas regulatórias.

No que diz respeito aos objetivos da governança de TI, destacam-se a criação de valor para a organização por meio do uso estratégico da TI, o gerenciamento dos riscos relacionados à TI e o alinhamento da TI com as necessidades do negócio. Para alcançar esses objetivos, é preciso uma estrutura clara de responsabilidades e processos decisórios (WEILL; ROSS, 2004).

Uma boa governança de TI traz benefícios significativos para a organização, tais como a melhoria da eficiência e eficácia na utilização dos recursos de TI, a redução de riscos e a melhoria do alinhamento da TI com as necessidades do negócio. Além disso, segundo Lopes e Vasconcelos (2010), a governança de TI contribui para a transparência e prestação de contas, bem como para a criação de valor para a organização.

Portanto, a governança de TI é essencial para o sucesso das organizações no ambiente atual de negócios altamente competitivo e tecnológico. Ela deve ser orientada para o valor, fornece uma estrutura clara de responsabilidades e processos decisórios e contribuir para a criação de valor para a organização. Além disso, é importante destacar que a governança de TI deve ser flexível e adaptável às mudanças no ambiente de negócios (WEILL; ROSS, 2004).

2.3 ENGENHARIA DE REQUISITOS

De acordo com Pressman e Maxim (2016), a engenharia de requisitos é um amplo espectro de tarefas e técnicas que visam alcançar uma compreensão aprofundada dos requisitos de um *software*. Esse processo é considerado uma atividade crucial na engenharia de *software*, iniciando-se durante a atividade de comunicação e estendendo-se até a fase de modelagem. É importante ressaltar que a engenharia de requisitos deve ser adaptada às necessidades do processo, projeto, produto e pessoas envolvidas, a fim de maximizar sua eficácia.

Sommerviller (2011) define os requisitos da engenharia de requisitos como descrições das funcionalidades esperadas do sistema, das limitações impostas ao seu funcionamento e dos serviços que ele deve oferecer. Esses requisitos refletem

ARTIGO

as necessidades dos usuários em relação a um sistema que atende a um propósito específico, como por exemplo, a busca por informações. É importante destacar que a compreensão precisa e completa desses requisitos é fundamental para o sucesso da engenharia de *software* como um todo.

É importante destacar que os requisitos podem ser divididos em dois tipos: requisitos funcionais e requisitos não funcionais. De acordo com Sommerville (2011), os requisitos funcionais descrevem as funcionalidades que o sistema deve oferecer. Quando expressos em termos de requisitos de usuário, esses requisitos são descritos de maneira abstrata, de forma a facilitar o entendimento pelos usuários do sistema. No entanto, para a equipe de desenvolvimento, é necessário que os requisitos funcionais sejam detalhados, descrevendo minuciosamente as funções, entradas e saídas do sistema. A compreensão clara dos requisitos funcionais é crucial para garantir que o *software* desenvolvido atenda às expectativas dos usuários e seja eficiente em suas funcionalidades.

Machado (2011) os requisitos não funcionais como diferentes dos requisitos funcionais, esses requisitos não estão diretamente relacionados às funcionalidades do sistema, mas sim a padrões de qualidade, como confiabilidade, desempenho, robustez, segurança, usabilidade, portabilidade, legibilidade, qualidade e manutenibilidade, entre outros. Esses requisitos são igualmente relevantes, pois definem se o sistema será eficiente em suas tarefas e se atenderá às necessidades dos usuários. Um sistema ineficiente ou que não cumpra com os requisitos não funcionais estabelecidos não será utilizado ou não atenderá às expectativas dos usuários. Portanto, é fundamental que tanto os requisitos funcionais quanto os requisitos não funcionais sejam considerados e atendidos no processo de desenvolvimento de *software*.

A partir da realização do levantamento de requisitos é possível criar gráficos e diagramas para representar o funcionamento do *software* por meio do processo de modelagem de *software*.

2.4 UML

Segundo Fowler (2014) a UML (*Unified Modeling Language*) é um conjunto de representações gráficas que visam facilitar o entendimento e funcionamento de um *software*, sendo que a maioria desses gráficos utiliza o estilo orientado a objeto. Dentre os principais diagramas que compõem a UML, destacam-se o Diagrama de Caso de Uso, o Diagrama de Classes, o Diagrama de Objetos, o Diagrama de Colaboração, o Diagrama de Sequência, o Diagrama de Atividades, o Diagrama de Estados, o Diagrama de Componentes, o Diagrama de Implantação, o Diagrama de Pacotes, entre outros. Cada diagrama possui suas próprias funções e conceitos específicos. A UML é controlada pela OMG (Object Management Group), criada por um grupo de empresas com o objetivo de estabelecer um padrão global para o entendimento dessas representações.

De acordo com Martins (2010) a UML surgiu com a finalidade de modelar e representar *softwares*, mas atualmente sua aplicação expandiu-se para outras áreas, tais como *workflow*, estrutura e comportamento de dispositivos eletrônicos e mecânicos, design de hardware e outros processos. Além disso, a UML é amplamente adotada por empresas, sendo que a maioria delas utiliza as etapas do Rational Unified Process (RUP), que estabelecem quando e como procedimentos são executados para concluir um trabalho e alcançar um objetivo, que pode ser a criação ou atualização de um *software*.

2.4.1 Diagrama de Classes

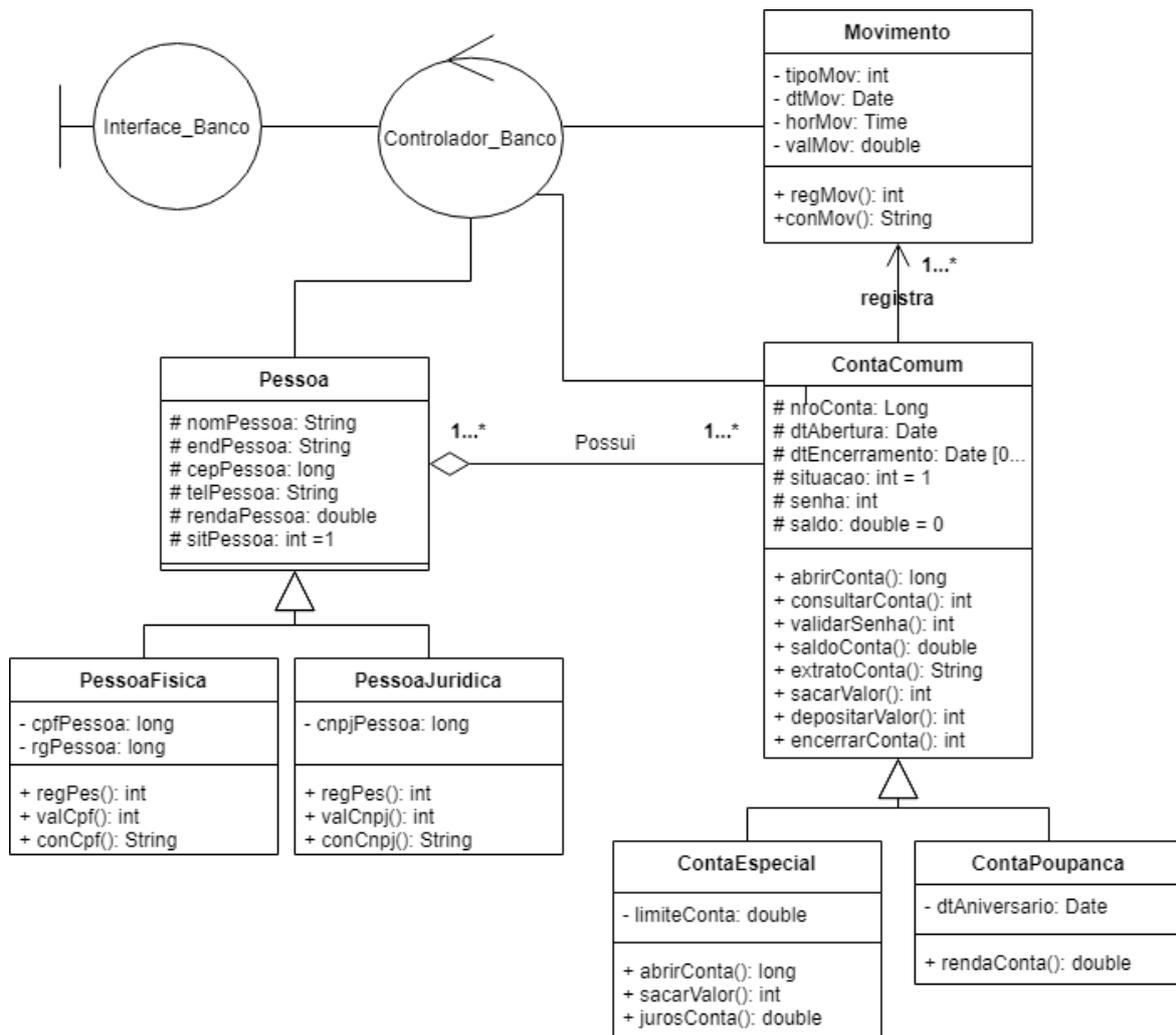
Segundo Franco (2012), ao analisar um sistema é preciso que o analista de sistemas crie abstrações, regras e conceitos que descrevam as funcionalidades do programa. O diagrama de classes possibilita essa descrição, permitindo que, por meio de abstrações, sejam definidos recursos, interações e informações necessárias para a execução do sistema.

O diagrama de classes ilustra as classes que integram o sistema e suas interações. Esse diagrama é composto, principalmente, por classes, interfaces e

ARTIGO

relacionamentos. Cada classe possui nome, atributos e métodos, e se refere a objetos do mesmo modelo, sendo que cada definição de classe tem um nome único. É importante que o nome da classe seja um substantivo para indicar o tipo de objeto representado (FRANCO, 2012).

FIGURA 1: EXEMPLO DIAGRAMA DE CLASSES



FONTE: ADAPTADO DE GUEDES (2011, p. 32)

A figura 1 ilustra o exemplo de um diagrama de classes aplicado em um sistema bancário.

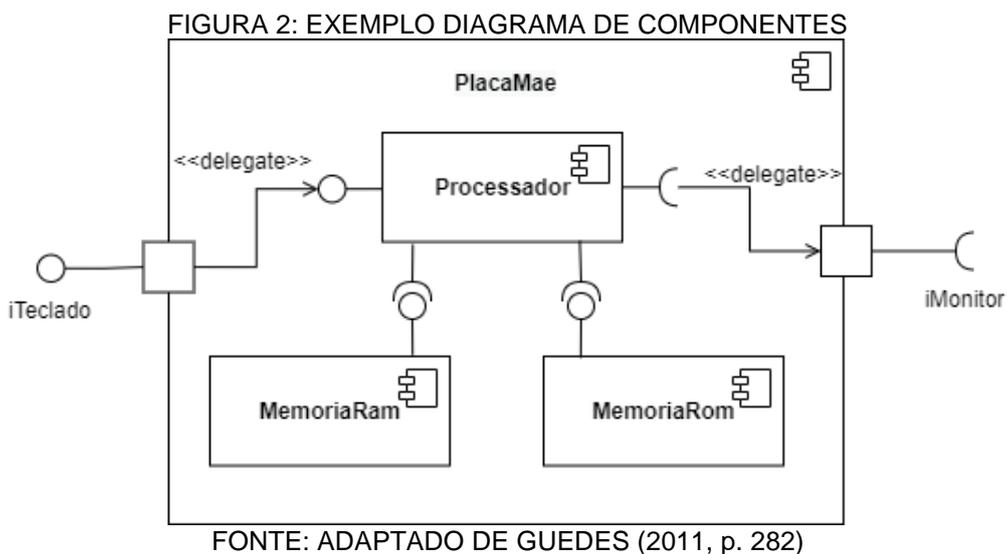
De acordo com Guedes (2011), o diagrama de classes é considerado um dos mais relevantes da UML e é utilizado como base para a maioria dos outros diagramas.

ARTIGO

Esse diagrama tem como finalidade definir a estrutura das classes em um sistema, especificando seus atributos e métodos, bem como suas associações e interações para troca de informações. Portanto, é um elemento crucial para a modelagem de sistemas orientados a objetos.

2.4.2 Diagrama de Componentes

Conforme afirmado por Medeiros (2010), os componentes são elementos fundamentais de um sistema e são representados em um diagrama de componentes. No UML 2, esses componentes são representados por um retângulo com um símbolo composto por três retângulos em seu interior, conforme demonstrado na figura 2. Para modelar a comunicação entre componentes, é utilizado o conceito de interface, que consiste em uma linha que conecta um componente a outro, interrompida por um círculo e uma linha côncava. O círculo representa o componente que envia informações, enquanto a linha côncava representa o componente que recebe as informações. Um exemplo ilustrativo pode ser observado na figura 2, que apresenta o diagrama de componentes de uma placa mãe de computador.



De acordo com Guedes (2011), o diagrama de componentes é empregado como uma forma de documentar a estrutura dos elementos físicos de um sistema,

possibilitando uma melhor compreensão e facilitando a reutilização de código. A modelagem irá auxiliar na criação de um protótipo do projeto.

2.5 ARQUITETURA E SOFTWARE

Conforme Pressman e Maxim (2016) a arquitetura de um *software* é uma representação que permite a análise da efetividade do projeto em atender aos requisitos, possibilitando mudanças em um estágio em que o projeto é relativamente fácil e reduzindo os riscos associados à construção do *software*.

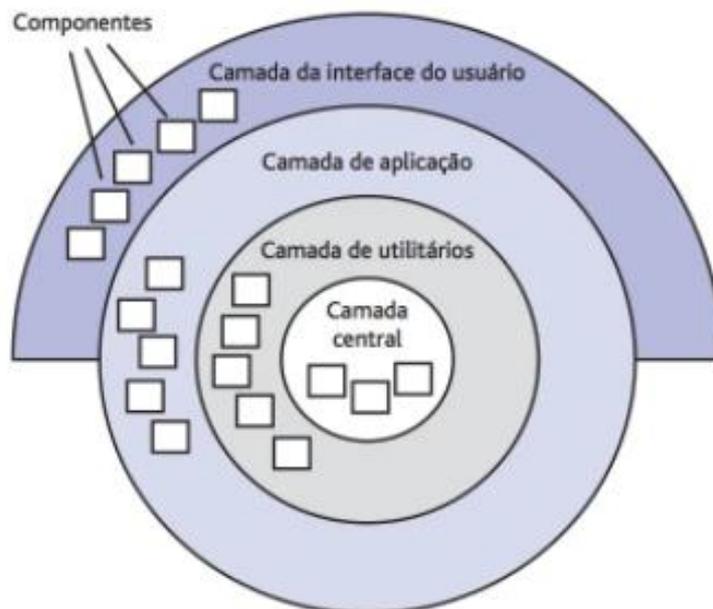
Para Bass, Clements e Kazman (2012), a arquitetura de software pode ser definida como a estrutura ou estruturas do sistema que englobam os componentes de *software*, suas propriedades altamente visíveis e as relações entre esses componentes.

Dentre os tipos de arquiteturas de *software* existentes serão apresentadas a arquitetura de fluxo de dados, em camadas e orientada a objetos.

2.5.1 Arquitetura em Camadas

Segundo Pressman e Maxim (2016), a arquitetura em camadas é estruturada em várias camadas diferentes, começando por uma camada que utiliza uma linguagem próxima à linguagem humana e progredindo gradualmente em direção à linguagem de máquina. A figura 4 ilustra a estrutura da arquitetura em camadas.

FIGURA 4: ARQUITETURA EM CAMADAS



FONTE: PRESSMAN E MAXIM (2016)

Conforme Zenker *et al.* (2019), a arquitetura em camadas consiste em níveis que vão desde a interface, que é a camada mais externa, até a camada central, que é a mais próxima à linguagem de máquina.

2.5.2 Arquitetura Orientada a Objetos

Segundo Zenker *et al.* (2019), a arquitetura orientada a objetos é baseada em classes e objetos com certo grau de encapsulamento. Nessa arquitetura, os componentes do sistema manipulam dados encapsulados por meio de operações, enquanto a coordenação e comunicação entre os componentes ocorrem por meio de mensagens (PRESSMAN; MAXIM, 2016).

Depois de projetada a arquitetura de *software* já é possível definir se o sistema será desenvolvido no formato de aplicativo ou aplicação web.

2.5.3 Arquitetura Entidade e Relacionamento

A arquitetura de banco de dados entidade e relacionamento é uma técnica de modelagem de dados que permite aos desenvolvedores representar as informações de um sistema de maneira organizada e estruturada. Segundo os autores Pressman e Maxim (2016) o modelo entidade e relacionamento é um dos mais populares e amplamente utilizados para representar a estrutura de um banco de dados. Ele é composto por entidades, que são objetos do mundo real, e relacionamentos, que representam as conexões entre essas entidades.

Já os autores Barbosa e Freitas (2018), destacam que o modelo entidade e relacionamento permite aos desenvolvedores descrever com precisão as relações entre as entidades de um sistema, como por exemplo, a relação entre um cliente e um pedido de compra. Essas relações podem ser classificadas como um-para-um, um-para-muitos ou muitos-para-muitos.

O diagrama utilizado para representar entidade e relacionamento é muito semelhante ao diagrama de classes da UML.

Em resumo, a arquitetura de banco de dados entidade e relacionamento é uma técnica para os desenvolvedores, pois permite representar de forma clara e estruturada a estrutura de um banco de dados. O objetivo é facilitar o processo de desenvolvimento e manutenção de sistemas, e garantir eficiência e qualidade nos resultados.

2.6 APLICAÇÃO WEB

De acordo com Mariano e Melo-Minardi (2017), a aplicação *web* é dividida em dois polos de ação: o *back-end*, que é responsável pelos códigos que fazem o sistema funcionar corretamente com todas as suas funções, executado na máquina, e o *front-end*, que é responsável pelo desenvolvimento da parte visual da aplicação, que funcionará diretamente na máquina do usuário. O desenvolvimento *web* é utilizado por todas as organizações que possuem um site ou um sistema na internet. Para a criação de uma aplicação *web*, é possível utilizar diversas linguagens, sendo as mais utilizadas o HTML, CSS, PHP, JavaScript e SQL.

Segundo Purewal (2010), criar uma aplicação web é uma tarefa desafiadora e que requer organização. Para isso, é comum utilizar pastas padrão que visam estruturar e tornar o projeto funcional. Na aplicação *web*, a parte visual é desenvolvida com linguagem de marcação, enquanto a parte lógica é realizada com linguagem de programação, e as informações são armazenadas em um banco de dados.

2.6.1 Linguagem de Marcação

Eis e Ferreira (2012) afirmam que a linguagem de marcação é frequentemente utilizada para publicações de conteúdos visuais, tais como textos, imagens, vídeos e áudios, especialmente em aplicações *web*.

A linguagem de marcação é constituída de *tags* (etiquetas em inglês) que descrevem conteúdos dentro de uma aplicação *web* e podem estar presentes em diferentes documentos. Essas *tags* são as palavras-chave que descrevem a formatação e exibição do conteúdo na página *web*. O HTML é a linguagem de marcação padrão utilizada para criar páginas *web* e estruturar todo o seu conteúdo (W3SCHOOLS.COM, 2023).

2.6.2 Linguagem de Programação

De acordo com Ascenio e Campos (2012), a programação refere-se ao processo de criação de programas para o processamento de dados. Para que o computador entenda o que precisa ser executado, é necessário escrever a programação em uma linguagem própria, denominada linguagem de programação, e o texto escrito nessa linguagem é chamado de código-fonte.

Para Manzano (2014), a escrita de programas exige a utilização de pelo menos duas ferramentas: um editor de textos para o código-fonte e outro para convertê-lo em código de máquina, que o computador possa compreender. Essas ferramentas, assim como outras, podem estar disponíveis em um único pacote, conhecido como Ambiente Integrado de Desenvolvimento (IDE, do inglês *Integrated*

ARTIGO

Development Environment). Existem diversas opções de IDEs disponíveis, adaptadas para diferentes linguagens de programação.

2.6.3 Banco de Dados

Conforme Korth, Silberchatz e Sudarshan (2012), o banco de dados é uma coleção de dados que representam informações específicas de um determinado domínio. De acordo com Barboza e Freitas (2018), um banco de dados eficiente é composto por uma estrutura de informações que, em conjunto, dão significado ao contexto em que estão inseridas. Além disso, um sistema de gerenciamento de banco de dados (SGBD) é utilizado para definir, manipular e construir bases de dados para diferentes finalidades.

Machado (2020) destaca que, para realizar consulta e manipulação de dados em um banco de dados, é utilizada a linguagem de banco de dados, sendo que a linguagem mais adotada atualmente como padrão é a *Structured Query Language* (SQL, linguagem estruturada de pesquisa), devido à sua capacidade de manipular e consultar dados de forma estruturada, baseada em um modelo relacional.

A linguagem SQL age como uma ponte intermediária entre as atividades que precisam ser realizadas e o banco de dados, permitindo a realização de inserções, remoções, edições ou qualquer outra atividade com os dados que já estão no banco de dados ou irão para ele, facilitando assim a manipulação dos dados armazenados (BARBOZA; FREITAS, 2018).

2.7 PROTOTIPAGEM

Morais *et al.* (2018) explicam que protótipos são versões preliminares de um produto, desenvolvidas de forma rápida e econômica, que podem ser testadas e compartilhadas pelos envolvidos no projeto. Essa fase é classificada como experimental e seu objetivo é encontrar a melhor solução para um produto.

As tecnologias de prototipagem rápida, também conhecidas como prototipagem de baixo custo, têm se popularizado por serem cada vez mais

ARTIGO

acessíveis, apresentarem menor tempo de produção e serem confiáveis. De acordo com Machado (2013), esse tipo de prototipagem está relacionado a um conjunto de tecnologias para a criação de simulações de objetos virtuais baseados em modelos físicos, permitindo a avaliação do produto antes da criação de um protótipo físico, possibilitando a realização de alterações que não foram planejadas inicialmente. Dessa forma, os protótipos são versões de baixo custo de um produto, utilizadas na fase experimental do projeto com o objetivo de encontrar a melhor solução para um produto.

Santos (2006) destaca que o processo de prototipagem é eficiente em termos de custo-benefício, graças à economia obtida com retrabalhos. Os protótipos podem ser classificados em três níveis de fidelidade: baixa, média e alta, sendo que cada nível se refere à proximidade do protótipo em relação ao projeto final.

QUADRO 2: NÍVEIS DE PROTÓTIPOS

Protótipos de baixa fidelidade	Tem o propósito de testar funcionalidades mais simples e gerais do projeto, em outras palavras serve para testar de forma simplificada a interação do usuário com o sistema, sem a necessidade de se preocupar com interface. Os protótipos de baixa fidelidade, também chamados de rascunhos ou sketches.
Protótipos de média fidelidade	Envolvem mais as características de interface juntamente das funcionalidades básicas do sistema, podem não ter funcionalidades tão abrangentes, porém a interação do usuário com o sistema é melhor desenvolvida através de funções básicas de design, botões clicáveis e entre outras funcionalidades mais próximas do modelo final.
Protótipos de alta fidelidade	Possuem tanto características de interface quanto funcionalidades de interação ao usuário bem próximas ao desenvolvimento final, muitas vezes podem até serem utilizados como testes funcionais do sistema, o objetivo dos protótipos de alta fidelidade é verificar a usabilidade do projeto para que possa ser validado aos usuários.

FONTE: ADAPTADO DE SANTOS (2006, p. 261)

Depois pronto o protótipo é apresentado para a parte interessada avaliar o resultado, sugerir possíveis correções.

3. LEVANTAMENTO DE REQUISITOS

ARTIGO

No presente capítulo será abordado o processo de levantamento de requisitos, no qual foram coletados e analisados os requisitos de negócio, funcionais e não funcionais para o sistema a ser desenvolvido.

Inicialmente os requisitos foram coletados por meio de entrevista informal realizada em uma reunião com as partes interessadas, com o objetivo de compreender as necessidades do cliente/usuário do sistema. Em seguida foi realizada uma análise de cenário para simular o funcionamento do sistema.

Após realizada a coleta dos dados foi utilizado a técnica 5W2H, que apresentou os seguintes resultados:

- a) **What:** desenvolver um sistema de simulação para aprendizagem de manutenção elétrica de motocicletas;
- b) **Who:** sistema desenvolvido para estudantes de manutenção elétrica de motocicletas, destinado para que os professores possam avaliar aprendizado;
- c) **Why:** para auxiliar no aprendizado de manutenção elétrica de motocicletas, investir em inovação e focar na continuidade e sustentabilidade do negócio;
- d) **Where:** será usado em sala de aula e cursos *onlines* (eventualmente em feiras);
- e) **When:** será usado no horário mais conveniente ao usuário e/ou de acordo com solicitação de professor;
- f) **How:** Por meio de aplicação *web* disponível na internet ou intranet;
- g) **How Much:** Não foi definido valor, podendo ser um produto vendido ou alugado.

Com o auxílio da ferramenta 5W2H foram desenvolvidos os requisitos de negócio, funcionais e não funcionais:

Requisito de negócio:

- a) utilizar jogo de simulação de manutenção elétrica de motos para reforçar aprendizado de estudantes e usar em eventos diversos, estando de

acordo com os princípios de GTI de investir em inovação e focar na continuidade e sustentabilidade do negócio.

Requisitos funcionais:

- a) **o sistema deve ter menu de jogo:** o sistema deve possuir um menu inicial onde o usuário possa configurar e colocar seu nome antes de iniciar o jogo;
- b) **o sistema de ter Ordem de Serviço (OS):** as tarefas a serem realizadas pelo usuário devem vir por meio de OS, onde é relatado a falha constatada na motocicleta, lista de peças trocadas e valor do orçamento;
- c) **o jogo deve ter sistema de pontuação:** deve ser possível avaliar o nível de conhecimento do usuário por meio de pontuação;
- d) **o sistema deve permitir troca de peças:** deve ser possível que o usuário remova peça danificada da motocicleta e adicione peça nova;
- e) **o sistema deve ter instrução:** deve ter em algum lugar do jogo uma opção para o usuário receber ajuda por meio de instruções;
- f) **o sistema deve ter login:** o usuário só poderá acessar o sistema de simulação mediante login e senha previamente cadastrados pelo gerenciador (professor) do sistema;
- g) **o sistema deve ter ranking de usuários:** o sistema deve exibir em algum lugar os usuários mais bem ranqueados, permitindo a instituição bonificar essas pessoas;
- h) **o sistema deve ter ajuste de dificuldade:** o sistema deve permitir que o usuário ajuste a dificuldade de acordo com suas necessidades;
- i) **o sistema deve ter estoque de peças:** o sistema deve permitir que o usuário colete peças novas no estoque para substituir a peça danificada.

Requisitos não funcionais:

- a) **o sistema deve ser em aplicação web:** deve ser possível acessar (jogar) o sistema nos navegadores *web* para *desktops* populares no mercado;
- b) **o sistema deve ter banco de dados:** deve haver banco de dados relacional para o sistema.

Com os requisitos levantados foi possível realizar modelagem do sistema para apresentar às partes interessadas como o sistema funcionará e aos desenvolvedores como deve ser desenvolvido.

4. MODELAGEM

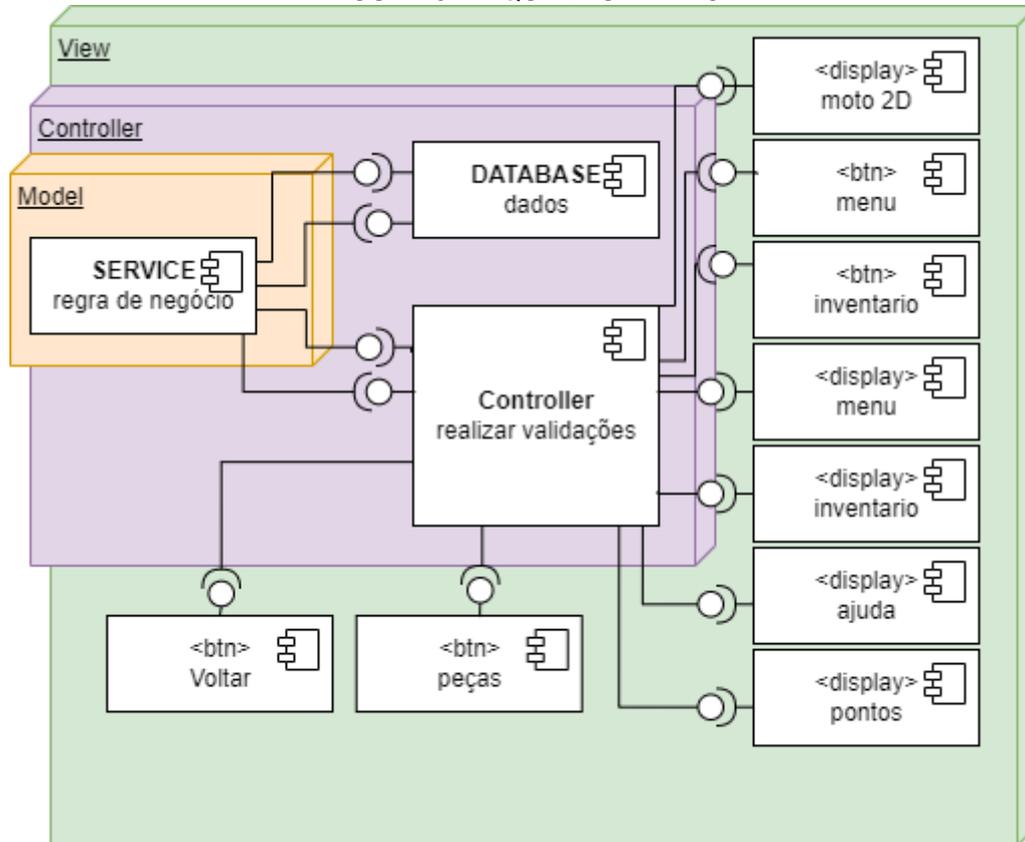
A modelagem dos requisitos levantados foi realizada por meio dos diagramas da UML, sendo dividido em arquitetura principal do sistema, arquitetura *backend* e arquitetura de banco de dados.

4.1 ARQUITETURA PRINCIPAL DO SISTEMA

Para modelagem da arquitetura principal do sistema, foi escolhido uma arquitetura em camadas conhecida como *Model*, *View* e *Controller* (MVC), cujo padrão é amplamente utilizado na engenharia de *software* para projetar *software* de qualidade. Segundo Pressman e Maxim (2016) e Bass, Clements e Kazman (2012), o MVC é recomendado por sua capacidade de melhorar a modularidade, a escalabilidade e a manutenibilidade do *software*. Neste padrão, o modelo é responsável pela gestão dos dados e lógica de negócios, enquanto a visão representa a interface de usuário e o controlador lidam com a interação do usuário e coordena as ações do modelo e da visão.

A figura 5 mostra como está estruturado a arquitetura em camadas junto com o MVC.

FIGURA 5: ARQUITETURA MVC



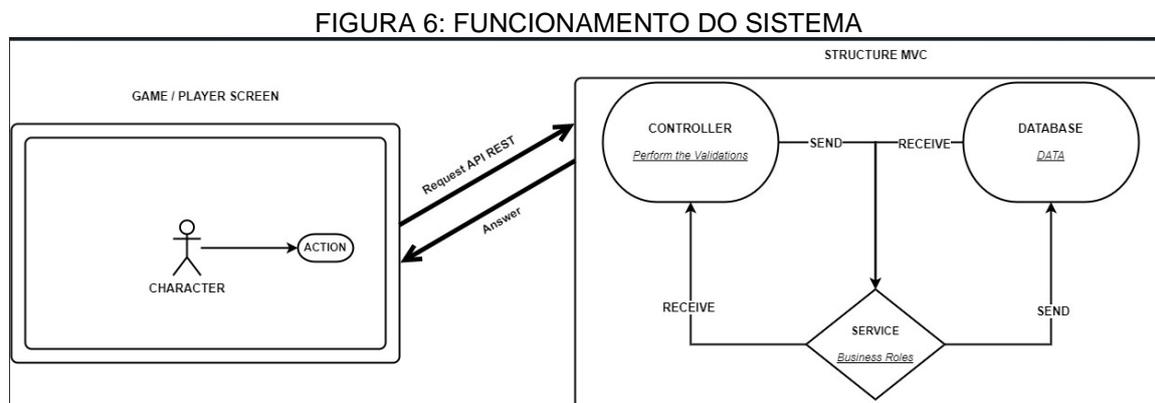
FONTE: OS AUTORES (2023)

As camadas da arquitetura possuem componentes de acordo com seus níveis:

- a) **View:** Descreve os componentes que devem aparecer na interface gráfica do sistema, o qual o usuário irá interagir, a legenda <btn> representa que é um botão e <display> informação exibida na tela;
- b) **Controller:** Descreve os componentes que fazem a comunicação entre o modelo e a interface. O componente *controller* realiza chamada de funções, envia e recebe mensagens. O componente *DATABASE* é responsável pelos processos que envolvem armazenamento ao banco de dados;

c) **Model:** Descreve os componentes responsáveis pelas regras de negócio pré-estipuladas.

Além da arquitetura em camadas, para melhor compreensão do funcionamento do sistema, foi utilizado outro diagrama para representar o funcionamento do sistema, conforme ilustra a figura 6.



FONTE: OS AUTORES (2023)

Conforme mostra a figura 6, o usuário irá interagir com o sistema por meio de uma interface de web. Cada ação que o usuário realizar será enviada ao sistema por meio de requisição, onde o *controller* irá receber, tratar a informação e enviar ao *service*. O *service* por sua vez vai verificar se o *controller* precisa de alguma informação do banco de dados, após a verificação fará ou não a consulta ao banco de dados e retornará uma resposta ao *controller* que responderá ao usuário por meio da interface *web*.

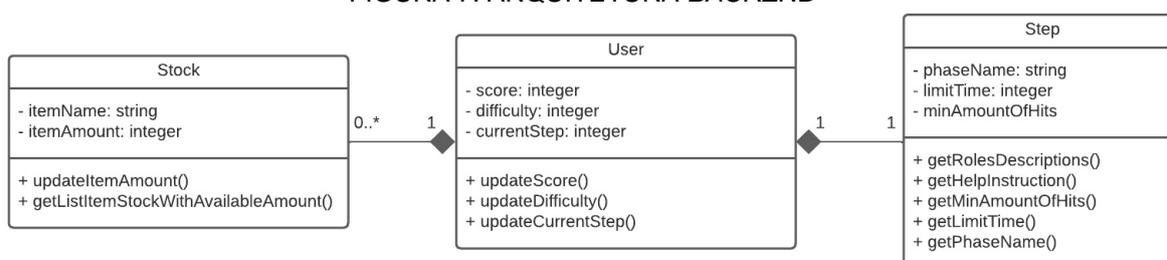
A arquitetura principal já possibilita criação de protótipo de telas e o desenvolvimento do sistema responsável pela interação entre o usuário e o jogo.

4.2 ARQUITETURA *BACKEND*

Foi escolhida a arquitetura orientada a objetos com base no diagrama de classe da UML para modelar o funcionamento do sistema no *backend*. O objetivo é

facilitar a compreensão dos desenvolvedores sobre essa parte do sistema. O modelo construído para o *backend* é ilustrado na figura 7.

FIGURA 7: ARQUITETURA *BACKEND*



FONTE: OS AUTORES (2023)

Conforme mostra o modelo o desenvolvimento do *backend* envolve três classes, tendo como classe principal a de usuário (*user*) com os atributos de pontuação, dificuldade e passo atual, contando com os métodos de atualização desses atributos. Além desses atributos poderão ser adicionados nome, unidade de ensino e outras informações caracterizadas como relevantes.

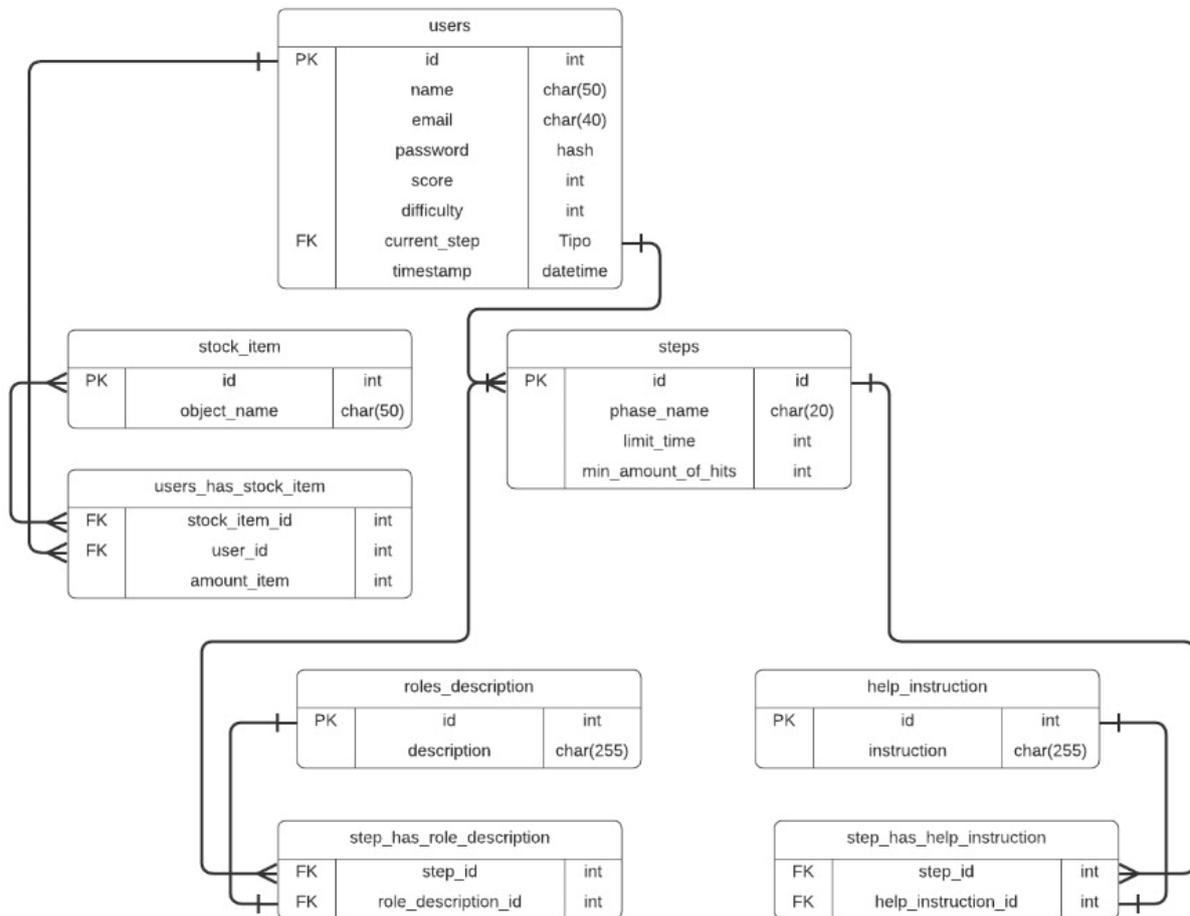
A segunda classe descrita no modelo é a classe de estoque (*stock*) com atributos e métodos relacionados aos itens e a terceira classe está relacionada com as missões e é chamada de *step* onde tem o nome da fase, tempo estimado e valores de pontuação, junto com seus respectivos métodos.

A arquitetura do *backend* permitirá que a equipe de desenvolvimento crie essa parte do sistema de maneira modular, podendo ser adicionado outras classes e ser utilizado em outros sistemas.

3.3 ARQUITETURA BANCO DE DADOS

Para a modelagem de banco de dados, foi escolhido o diagrama de entidade e relacionamento, cujo objetivo é facilitar os desenvolvedores sobre quais dados e informações devem ser armazenadas e como armazená-las, a figura 8 mostra o diagrama criado.

FIGURA 8: ARQUITETURA BANCO DE DADOS



FONTE: OS AUTORES (2023)

De acordo com o diagrama cada retângulo é uma entidade ou objeto e as linhas são as relações entre eles, então o usuário (*user*) pode ter várias peças de estoque relacionadas a ele, assim como várias missões (*steps*). Cada missão pode ter um único usuário relacionado a ela e várias descrições e instruções de ajuda.

A arquitetura do banco de dados permitirá que a equipe de desenvolvimento crie essa a estrutura dos dados que serão armazenados, podendo acrescentar, consultar, modificar e remover itens sem prejudicar o sistema.

5. PROTOTIPAGEM

Para realização da prototipagem utilizou-se uma ferramenta gratuita de chamada Figma¹, que permitiu o trabalho de maneira colaborativa.

No início do processo de prototipagem buscou-se informações referentes ao estilo que seria utilizado no sistema, quais cores e tipografia seriam o padrão. Após discussão e contato com a empresa foram definidas as cores conforme ilustra a figura 9.

FIGURA 9: CORES DO PROTÓTIPO
PALETA DE CORES



FONTE: OS AUTORES (2023)

As cores foram escolhidas de acordo com o guia de estilo da empresa, que possui tonalidades de laranja e preto. Após escolhidas as cores foram definidas quais seriam as fontes e seus tamanhos. A figura 10 mostra a tipografia escolhida para o protótipo².

¹ <https://www.figma.com/>

² Para visualização do protótipo completo acesse:
<https://www.figma.com/proto/7oK4culEAdGAKqtc4QkhQK/Simulador?type=design&node-id=103-9&scaling=contain&page-id=0%3A1&starting-point-node-id=103%3A9>

FIGURA 10: TIPOGRAFIA DO PROTÓTIPO

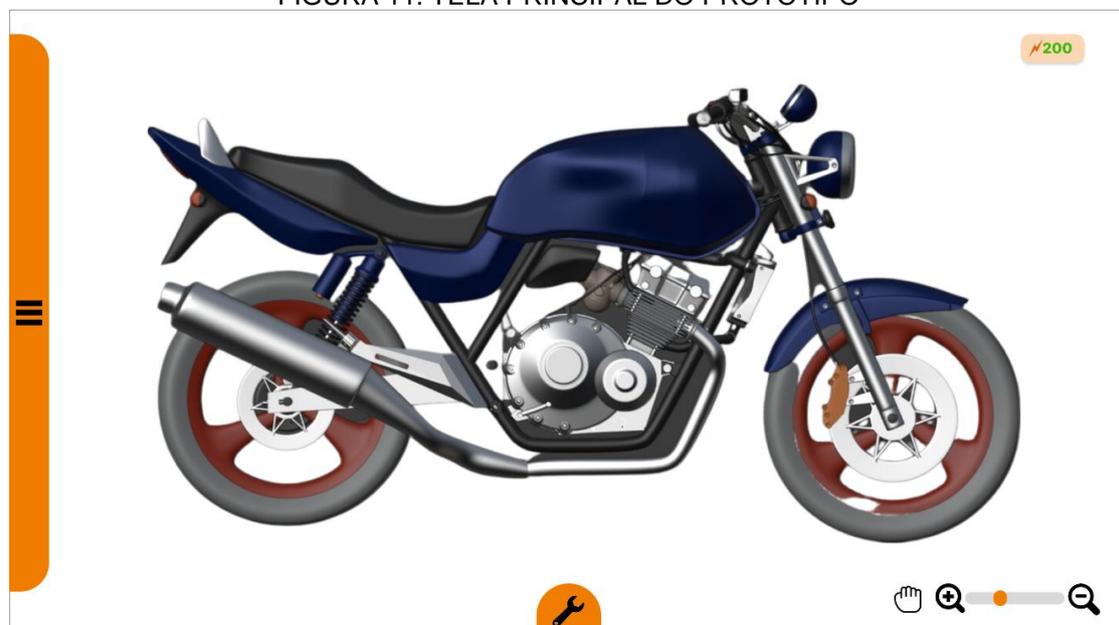
39 Thin Ultra Condensed	85 Extra Black
47 Light Condensed	67 Bold Condensed
45 Light	59 Ultra Condensed
49 Light Ultra Condensed	53 Extended
57 Condensed	63 Bold Extended
55 Roman	73 Black Extended
65 Bold	93 Extra Black Extended
75 Black	

FONTE: OS AUTORES (2023)

Assim como nas cores a tipografia segue o guia de estilo da empresa combinando tipo da fonte com o seu tamanho.

Após a definição do guia de estilo do protótipo, foi desenvolvido a tela do sistema tendo como base a arquitetura construída, conforme mostra a figura 11.

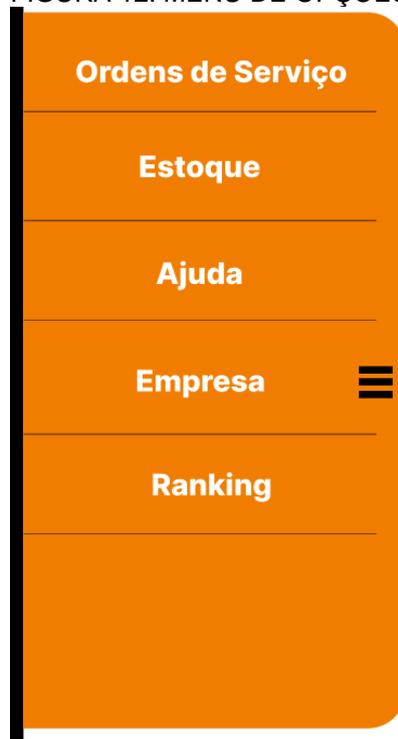
FIGURA 11: TELA PRINCIPAL DO PROTÓTIPO



FONTE: OS AUTORES (2023)

A tela principal foi construída de maneira que o jogador tenha acesso as principais funcionalidades do jogo. Ao centro está localizado a moto que o objeto principal do sistema, nela o usuário pode fazer interações de remover e adicionar peças. No lado esquerdo em laranja encontra-se o menu com opções para o jogo, conforme mostra a figura 12, em baixo com símbolo de uma chave de boca encontra-se inventario onde são exibidas de peças retiradas da moto, no canto inferior direito tem a opção de ampliar e reduzir a imagem da moto e no canto superior direito é exibido a pontuação do jogador, que pode ser utilizada como critério de avaliação.

FIGURA 12: MENU DE OPÇÕES



FONTE: OS AUTORES (2023)

Por meio do menu de opções, ilustrado na figura 12, o usuário é capaz de acessar as opções:

- a) **ordem de serviço:** mostra a descrição do problema a ser revolido, caixa de texto para o usuário escrever diagnóstico de falha, lista de verificação e botão para enviar o diagnóstico e assim receber a pontuação;

ARTIGO

- b) **estoque:** onde são exibidos peças disponíveis no estoque para colocar na moto;
- c) **ajuda:** executa tutorial de ajuda ensinando como o jogo funciona;
- d) **Empresa:** redireciona para página da empresa com catálogos e instruções de uso das peças;
- e) **ranking:** mostra classificação de jogadores de acordo com a pontuação.

Cada missão do jogo é um problema que precisa ser resolvido na moto, o problema é relatado por meio da ordem de serviço, onde por meio da descrição do problema o jogador deve identificar o defeito da moto, substituir a peça defeituosa e escrever diagnóstico e enviar, a figura 13 mostra exemplo de ordem de serviço.

FIGURA 13: ORDEM DE SERVIÇO

Ordem de Serviço

Descrição do problema

Bem Vindo ao Pit Stop Magnetron. Aqui você pode praticar e obter conhecimento sobre manutenção de motocicletas. Quer saber mais? Volte ao Menu Principal e clique em "Ajuda"

Essa moto não está ligando, verifique a situação.

Diagnóstico do Mecânico

[Empty text input field]

Lista de Verificação

- Vela de Ignição
- Bobina
- Botão Partida Elétrica

Enviar

FONTE: OS AUTORES (2023)

ARTIGO

No exemplo da figura 13 o problema apresentado é que a moto não liga, esse problema pode ser ocasionado por diversas peças defeituosas, então o jogador deve identificar, qual ou quais dessas peças estão causando esse problema. Caso tenha dificuldades o jogador pode clicar em ajuda, e o jogo irá mostrar os passos que deve realizar, a opção de ajuda afeta na pontuação do jogador. A figura 14 mostra exemplo de tutorial de ajuda após o jogador clicar, sugerindo que ele remova uma peça específica.

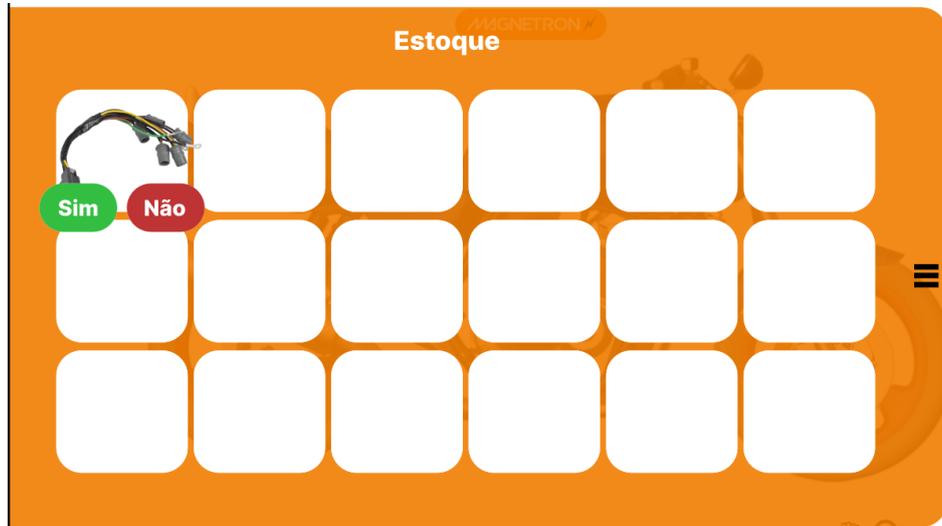
FIGURA 14: TUTORIAL DE AJUDA PARTE 1



FONTE: OS AUTORES (2023)

Após clicar na peça o jogador é orientado a ir ao estoque para adquirir uma peça nova para reposição, conforme mostra a figura 15.

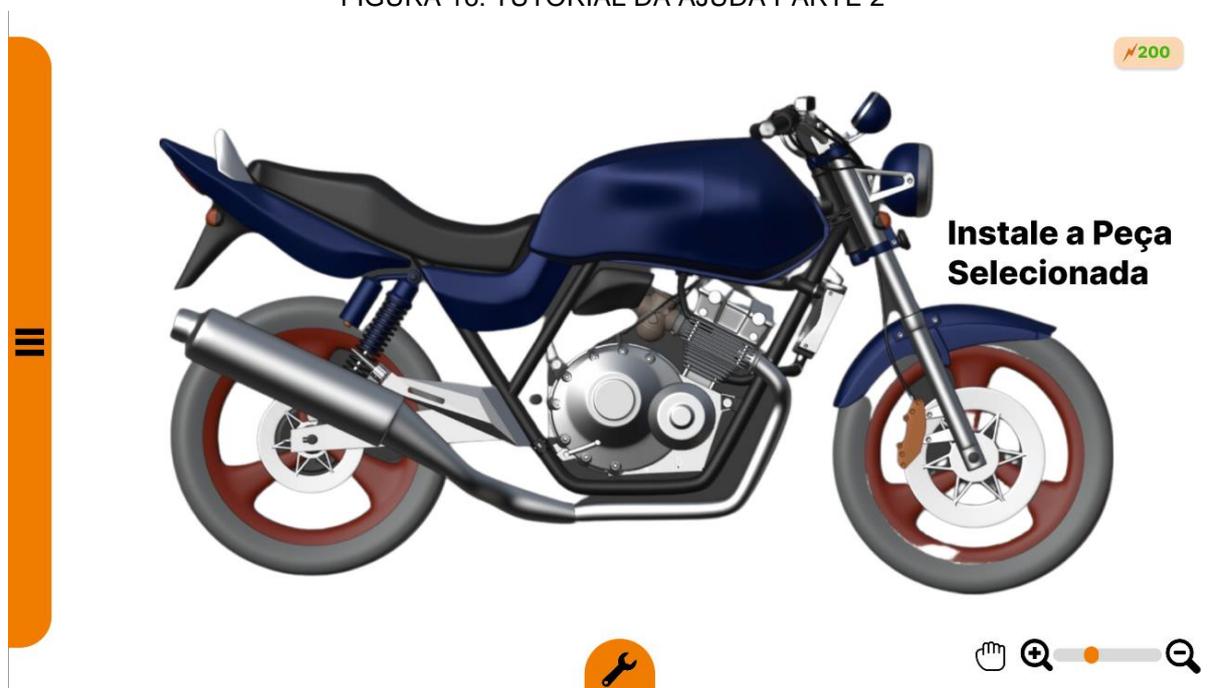
FIGURA 15: TUTORIAL DA AJUDA PARTE 2



FONTE: OS AUTORES (2023)

Após selecionar a peça o jogador deve confirmar se realmente quer adquirir aquela peça, ao clicar em sim é redirecionado tela principal onde deve colocar a peça nova na moto, conforme mostra a figura 16.

FIGURA 16: TUTORIAL DA AJUDA PARTE 2





FONTE: OS AUTORES (2023)

Após a instalação da peça o jogador é orientado a ir para a opção de ordem de serviço, escrever o diagnóstico, preencher a lista de verificação e clicar em enviar. Após clicar em enviar é atualizado a pontuação e inicia-se uma nova missão. Caso o usuário esse o diagnóstico é recebido uma notificação informando que a moto continua com o problema e o jogador pode continuar tentando.

O processo de pontuação ocorre por meio do tempo que foi necessário até identificar o diagnóstico correto e a relevância das peças retiradas, gerando pontuação dinâmica e permitindo identificar dificuldades de aprendizado de acordo com as missões.

Como plano de ação para transformar o protótipo em sistema sugere-se conversa com professor responsável em ministrar aulas de manutenção elétrica de motocicletas para identificar e criar missões pertinentes ao aprendizado, bem como as peças relevantes para serem removidas e colocadas na motocicleta. Após isso solicitar para a empresa imagens de cada peça da moto para que possa ser adicionada ao jogo. É necessário também contratar a equipe de desenvolvimento de software para criar o jogo de acordo com a arquitetura. Por fim providenciar um

servidor *web* para disponibilizar o acesso ao jogo. Os prazos para o plano de ação podem variar de acordo com os recursos disponíveis na empresa.

Após a etapa de prototipagem, iniciou-se a discussão sobre os resultados obtidos.

6. DISCUSSÃO DE RESULTADOS

Após o desenvolvimento do protótipo foi possível validar se o resultado obtido estava de acordo com os requisitos e com as arquiteturas.

A primeira avaliação foi sobre o requisito de negócio, que é utilizar jogo de simulação de manutenção elétrica de motos para reforçar aprendizado de estudantes e usar em eventos diversos, estando de acordo com os princípios de GTI de investir em inovação e focar na continuidade e sustentabilidade do negócio, cujo resultado foi positivo, pois o protótipo gera valor para a empresa por ter sido projetado para promover o produto e a marca, demonstrando perícia em componentes elétricos e criando percepção positiva junto aos usuários.

Além dos princípios de GTI do requisito de negócio, percebeu-se que o princípio de estratégia também foi atendido pois o jogo está alinhado aos objetivos estratégicos da empresa, buscando agregar valor ao oferecer experiência educativa e aprimorar o conhecimento sobre os componentes elétricos.

Após avaliar o requisito de negócio iniciou-se os requisitos funcionais, os requisitos funcionais tiveram dois que não foram contemplados no protótipo, sendo o primeiro deles o sistema deve ter login e o segundo o sistema deve ter ajuste de dificuldade, o primeiro não foi realizado no protótipo pois é trivial ao sistema a tela de login e está relacionado ao acesso e não jogo, já o outro não foi contemplado pois foi escolhido que o ajuste de nível de dificuldade fosse estipulado por parte de quem cria as missões, que será feito por outro sistema acessando o mesmo banco de dados. Os requisitos não funcionais foram atendidos.

O protótipo foi construído usando a arquitetura principal do sistema, atendendo todo o modelo, já as outras arquiteturas não foram contempladas pois são arquiteturas para o desenvolvimento do jogo e não carregam aspectos visuais.

Após a criação da primeira versão do protótipo, foi realizada reunião com a empresa para validar se o resultado estava de acordo com a expectativa, foi sugerido algumas mudanças que foram realizadas resultando no protótipo final apresentado no trabalho.

7. CONSIDERAÇÕES FINAIS

Foi possível atingir todos os objetivos da pesquisa elaborada, onde percebeu-se a importância de Governança de TI e jogos digitais no ambiente industrial, além da necessidade de conhecer os processos de engenharia de software para a criação de um jogo, onde foi indispensável identificar inicialmente quais seriam os tópicos a serem levantados, o que permitiu a compreensão da demanda de aplicação e decidir quais seriam os requisitos de negócio, funcionais, não funcionais e possibilitando o início da modelagem do jogo.

Depois de ter concluído a modelagem do jogo iniciou-se a prototipagem, concebendo maior união e conhecimento entre os participantes da equipe. Depois do protótipo desenvolvido foi verificado e validado se estava de acordo com os requisitos e modelo.

Encontrou-se dificuldade em realizar encontros entre os membros da equipe e a empresa devido a conflitos de agenda, todos os participantes da equipe interagiram em participar da pesquisa de maneira em que nenhum ficou de fora ou responsável por um assunto ou capítulo inteiro.

O desenvolvimento do jogo possibilitou a associação entre teoria e prática, promovendo interação entre os membros da equipe, assim como a ciência de que um plano idealizado mentalmente pode resultar em algo distinto na prática para pessoas diferentes.

ARTIGO

Como foi visto que o jogo é comercialmente viável, para trabalhos futuros aconselha a realização de interação do menu de jogador, ajuste de volume por meio do jogo, adição de mais peças, motocicletas, objetos para atualizações e missões, além do desenvolvimento do sistema de criação de missões e gestão de aprendizagem.

REFERÊNCIAS

ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. **Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java**. 3 ed. São Paulo: Pearson, 2012.

AMBROSE, G; HARRIS, P. **Design thinking**. Tradução de Mariana Belloi. rev. Porto Alegre: Bookman, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 38500 – Tecnologia da informação – Governança da TI para organização**. Rio de Janeiro, 2018.

BASS, L; CLEMENTS, P; KAZMAN, R. **Software architecture in Practice**. 3 ed. massachusetts:Addison-Wesley Professional, 2012.

BARBOSA, F. F. M.; FREITAS, P. H. C. **Modelagem e desenvolvimento de banco de dados**. 1 ed. Porto Alegre: Sagah, 2018.

BECHARA, F; SAMPAIO, M. **Governança de TI: a nova revolução empresarial**. Brasport, 2014.

EIS, D.; FERREIRA, E. **HTML5 e CSS3 com farinha e pimenta**. 1 ed. São Paulo: Tabless, 2012.

FOWLER, M. **UML Essencial: Um breve guia para linguagem-padrão de modelagem de objetos**. 3 ed. Tradução de João Tortello. São Paulo: Bookman, 2014.

FRANCO, C. A. G. S. **Ferramenta Computacional para o Apoio ao Processo de Acompanhamento de Aquisição e Desenvolvimento das Competências do Aluno do Curso de Medicina**. 142 f. Dissertação (Mestrado em Tecnologia em Saúde) - Pontifícia Universidade Católica do Paraná, Curitiba, 2012.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 6 ed. São Paulo: Atlas, 2017.

ARTIGO

GODINHO, L. A. de C. **Gerenciamento de projetos**. Rio de Janeiro: Elsevier, 2010.

GUEDES, G. T. A. **UML 2: uma abordagem prática**. 3 ed. São Paulo: Novatec Editora, 2011.

JÓHANNESSON, P.; SODERSTROM, E. *IT governance: the key to strategic alignment and value delivery*. **Journal of Information Technology Management**, v. 4, n. 3, p. 1-11, 2013.

KIM, K. A.; NOFSINGER, J. R. **Corporate governance**. Pearson Education, 2016.

KORTH, H. F.; SILBERSCHATZ, A; SUDARSHAN, S. **Sistema de banco de dados**. 6 ed. Rio de Janeiro: campus, 2012.

LISBÔA, M. G. P.; GODOY, L. P. Aplicação do método 5W2H do processo produtivo do produto: a joia. **Iberoamerican Journal of Industrial Engineering**. Florianópolis – SC, V. 4, n. 7, p. 32-47, 2012.

LOPES, S. C. S.; VASCONCELOS, A. C. Governança de TI: uma revisão sistemática de literatura. **Revista de Administração e Inovação**, São Paulo, v. 7, n. 1, p. 5-25, 2010. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1808-23862010000100002&lng=pt&nrm=iso>. Acesso em: 09 mar. 2023.

MACHADO, F. N. R. **Análise e gestão de requisitos de software: onde nascem os sistemas**. 1 ed. São Paulo: Érica, 2011.

MACHADO, F. N. R. **Banco de dados: projeto e implementação**. 4 ed. São Paulo: Érica, 2020.

MACHADO, L. C. **Modelo conceitual integrando prototipagem rápida e delineamento de experimentos na concepção de novos produtos**. 95 f. Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2013.

MANZANO, J. A. N. G. **Programação de computadores com C++**. 1 ed. São Paulo: Érica, 2014.

MARIANO, Diego; MELO-MINARDI, Raquel. *Introdução à Programa Web para Bioinformática: HTML, CSS, PHP e JavaScript*. 1 ed. Belo Horizonte: KDP Amazon, 2017.

MARCONI, M. A.; LAKATOS, Eva M. **Fundamentos de Metodologia Científica**. 8 ed. São Paulo: Atlas, 2017.

MARTINS, J. C. C. **Gerenciando Projetos de desenvolvimento de Software com PMI, RUP e UML**. 5 ed. Rio de Janeiro: Brasport, 2010.

ARTIGO

MEDEIROS, E. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo: Pearson Makron Books, 2010.

MONKS, R. A. G.; MINOW, N. **Corporate governance**. John Wiley & Sons, 2011.

MORAIS, I. S. *et al.* **Introdução a big data e internet das coisas (IoT)**. Porto Alegre: SAGAH, 2018.

NEWZOO. **Global games market report**. 2021. Disponível em: <<https://newzoo.com/insights/articles/global-games-market-revenues-to-reach-175-8-billion-in-2021/>>. Acesso em: 21 mar. 2023.

PRESSMAN, R; MAXIM, B. **Engenharia de Software: uma abordagem profissional**. 8.ed. Porto Alegre: AMGH, 2016.

PUREWAL, S. **Learning web app development: build quickly with proven javascript techniques**. 1 ed. United States: O'Reilly, 2014.

SANTOS, Robson L. G. **Usabilidade de interfaces para sistemas de recuperação de informação na web: estudo de caso de bibliotecas on-line de universidades federais brasileiras**. 347 f. Tese (Doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

SOMMERVILLER, I. **Engenharia de Software**; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirama. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

W3SCHOOLS.COM. **HTML Introduction**. Disponível em: <https://www.w3schools.com/html/html_intro.asp>. Acesso em: 27 mar. 2023.

WEILL, P.; ROSS, J. W. **IT Governance: How Top Performers Manage IT Decision Rights for Superior Results**. Boston: Harvard Business School Press, 2004.

ZEGREANU, B; ROBU, A. *Software Quality Assurance through IT Governance*. **Procedia Economics and Finance**, v. 39, p. 79-86, 2017.

ZENKER, Aline M et al. **Aquitetura de sistemas**. 1 ed. Porto Alegre: SAGAH, 2019.



Esta obra está licenciada com Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.
[Recebido/Received: Abril 30, 2023; Aceito/Accepted: Agosto 29, 2023]