



FACULDADE PROJEÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Lucas da Silva Barbosa

SISTEMA PROMETHEUS: SISTEMA DE MONTAGEM DE COMPUTADORES

Brasília, novembro de 2022



FACULDADE PROJEÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

SISTEMA PROMETHEUS: SISTEMA DE MONTAGEM DE COMPUTADORES

Projeto do Curso de Análise e Desenvolvimento de Sistemas da Faculdade Projeção, como requisito complementar na disciplina de projeto integrado e obrigatório à obtenção de aprovação na disciplina.

Brasília, 25 de novembro de 2022.

Prof. Me. Salvador Alves de Melo Junior
Orientador

AGRADECIMENTOS

A Deus por ter me dado resiliência para manter a fé e perseverança para me reinventar diante das adversidades.

A esta instituição, seu corpo docente, direção e administração que viabilizaram a minha formação enquanto Analista de Sistemas e Desenvolvedor.

Ao meu professor-orientador Salvador Alves de Melo Junior, pelo suporte durante toda a disciplina que envolveu esse projeto e pelo suporte que me deu desde o primeiro semestre na instituição.

Aos meus pais, pelo amor e apoio incondicional.

E a todos que fizeram parte da minha formação, seja com contribuições diretas ou indiretas.

SUMÁRIO

RESUMO.....	6
CAPÍTULO I	7
VISÃO INICIAL.....	7
1. INTRODUÇÃO	7
2. CONTEXTUALIZAÇÃO	7
3. PROBLEMÁTICA	8
4. SOLUÇÃO PROPOSTA	9
5. O QUE SE ESPERAR.....	11
6. OBJETIVO GERAL.....	12
6.1 OBJETIVOS ESPECÍFICOS.....	12
7. ESTRUTURA DO PROJETO	12
CAPÍTULO II	14
MODELAGEM DE DADOS	14
2.1. Diagrama de Entidade Relacional	14
2.2. Modelo de Entidade-Relacional.....	16
2.3 Dicionário de Dados.....	19
2.3.1 Tabela <i>placas_mae</i>	20
2.3.3 Tabela <i>log_acessos</i>	22
2.3.4 Tabela <i>migrations</i>	23
2.3.5 Tabela <i>armazenamentos</i>	24
2.3.6 Tabela <i>processadores</i>	25
2.3.7 Tabela <i>gabinetes</i>	27
2.3.8 Tabela <i>configuracoes</i>	29
2.3.9 Tabela <i>usuarios</i>	30
2.3.10 Tabela <i>memorias_ram</i>	32
2.3.11 Tabela <i>computadores</i>	33
2.3.12 Tabela <i>qtd_pentes_no_computador</i>	36
CAPÍTULO III	38
3.1. Conjunto de Tabelas	39
3.2. Diagrama de Caso de Uso	45
3.3. Conjunto de Descrições de Caso de Uso	46
3.3.1 Realizar Cadastro	46

3.3.2 Realizar Login.....	48
3.3.3 Manter Processador.....	50
3.3.4 Manter Computador	52
CONCLUSÃO.....	55
BIBLIOGRAFIA.....	57

RESUMO

Atualmente, é perceptível o uso de sistemas com fins de simular alguma circunstância que, se feita com recursos reais, poderia resultar em maior risco. Portanto, há uma menor tendência em se testar todos os cenários possíveis se esses mesmos testes forem feitos fora de um ambiente simulado. Isto pode limitar a extensão e a profundidade que um estudo ou um relatório podem possuir sobre a tal circunstância, o que pode resultar num menor aprendizado ou em menos inovações. No caso da montagem de computadores, além de riscos, pode estar associado um alto custo se não for utilizado um sistema para simular sua montagem. Este documento visa relatar todo o processo de prototipação de um sistema Web que oferecerá as funcionalidades de um sistema de montagem de computadores, tendo em vista suas peças e a compatibilidade entre elas. A preocupação do sistema está no aprendizado que as pessoas, principalmente os estudantes de informática, podem adquirir ao utilizar um sistema que simule uma montagem de computador ao invés de montar e desmontar computadores na prática, utilizando recursos reais.

Palavras-Chave: Arquitetura de Computadores, Montagem de Computadores, Hardware, Aprendizado de Tecnologia

CAPÍTULO I

VISÃO INICIAL

O projeto “PrometheusWeb”, a ser desenvolvido por mim, Lucas da Silva Barbosa, visa, além de exercitar os conhecimentos adquiridos ao longo da minha atividade como estudante na Faculdade Projeção como aluno de Análise e Desenvolvimento de Sistemas, prototipar funcionalidades de uma aplicação web que não se sustentará somente no conhecimento adquirido durante as disciplinas ministradas. Um dos elementos que constituem um passo além é, por exemplo, a administração de serviços web do lado do back-end por meio de requisições HTTP utilizando o estilo de arquitetura REST.

1. INTRODUÇÃO

Hoje em dia, a maioria dos profissionais, para executarem suas tarefas, devem utilizar de um computador, seja esse um minicomputador (notebooks ou smartphones), microcomputadores (computador desktop) ou outro tipo de computador. Tratando dos profissionais que utilizam um computador desktop, esses costumam utilizar aplicativos dentro de seus computadores para atingirem seus fins profissionais, como o Word para editar documentos, Excel para editar planilhas, Google Chrome para navegar em um site na Internet e diversos outros.

Uma vez expondo este contexto, percebe-se que existe uma necessidade dos profissionais que utilizam aplicações em um computador receber um bom desempenho da máquina ao se utilizar esses aplicativos. Aplicações como o Pacote Office podem consumir pouco de uma máquina, porém, profissionais que trabalham com aplicações que requerem muito da máquina como, por exemplo, aplicações de análise de dados, geoprocessamento, design 3D de jogos, entre tantos temas, devem contar com um hardware específico para poderem desempenhar bem suas tarefas.

2. CONTEXTUALIZAÇÃO

Na perspectiva de alunos de informática, a disciplina de montagem de computadores pode ser uma disciplina com abordagem prática a depender da abordagem, pois não é suficiente somente o conhecimento teórico sobre as peças individualmente e sobre o computador montado. A disciplina exige daquele que a

estuda a aquisição do conhecimento prático de montar um computador a partir de peças uma vez posto em uma situação real. Para este fim, há um jogo chamado PC Building Simulator que, como o nome sugere, simula a montagem e desmontagem de um computador a partir de suas respectivas peças. (COSTA; SILVA, 2019)

Na disciplina de Arquitetura de Computadores, os alunos devem adquirir ciência dos componentes de um sistema computacional, entendendo sua interação com outros componentes. Com esse conhecimento, também é incluso como medir o desempenho tanto dos componentes individualmente quanto do computador contendo todos os componentes necessários. Com a finalidade de complementar essa aquisição de conhecimento, é empregado de forma complementar a simulação desse mesmo sistema computacional para a execução de uma atividade para avaliar componentes do tal sistema, o que fará o aluno fixar o conteúdo e amadurecer conceitos. (DUENHA; AZEVEDO, 2016)

Ao fazer a montagem de um computador na realidade, corre-se o risco de danificar os componentes e acidentar-se, o que pode tornar o processo custoso do ponto de vista financeiro e desconfortável considerando a possibilidade de acidentes. Tendo essa situação em vista, torna-se interessante a adoção de simulações para montar um computador, pois, por meio da simulação, consegue-se atingir o fim de aprender a montar um computador e falhar em um ambiente simulado pune menos aqueles que estão montando o computador. (VIEIRA, SANTOS, LIMA, GADELHA, 2018)

3. PROBLEMÁTICA

A problemática reside no fato de que as pessoas necessitam personalizar e verificar as configurações de seus computadores desktop com a finalidade de desempenhar bem suas tarefas profissionais. Um exemplo são os profissionais que trabalham com o geoprocessamento por meio da aplicação ArcGIS. Um profissional de geoprocessamento, para entregar bem uma análise envolvendo todos os documentos necessários, o que pode incluir mapas, necessita de um hardware específico para suas finalidades.

Hoje em dia, o consumidor que está no contexto dessa problemática costuma comprar hardware em lojas virtuais como Kabum, Pichau e TerabyteShop, sejam

peças individuais ou computadores montados. Estas mesmas lojas possuem atendimento e até mesmo uma seção destinada à montagem de computadores, porém, não constituem uma plataforma de montagem de computador, simulando como aquela configuração se comportaria diante de um cenário real, levando em consideração a interação de todas as peças umas com as outras.

4. SOLUÇÃO PROPOSTA

A solução para abordar a condição atual supracitada, constitui na construção de um sistema web. Este sistema web será uma coleção de páginas web, o qual o usuário poderá cadastra-se, entrar com sua conta e interagir com o sistema para poder montar um computador. Esse sistema fará com que o usuário ganhe alguma experiência relacionada às peças de um computador e a montagem do mesmo.

Esse tipo de solução é útil e vai na mesma direção onde vão as pessoas que se preocupam com o aprendizado de montagem de computadores, o qual desempenham um alto esforço para endereçar a atual situação daqueles que precisam obter o conhecimento de montagem de computadores, principalmente de estudantes de informática e tecnologia. Evidentemente, por se tratar de um sistema web, é uma solução que converge com uma nova tendência social: a das atividades remotas. Ao invés de estar presente em um curso presencial, ou qualquer atividade similar, para obter conhecimento sobre montagem de computadores, isto é feito remotamente, via web.

As ferramentas que serão utilizadas no desenvolvimento do projeto incluem linguagens de programação, banco de dados, ambientes de desenvolvimento integrado, entre outros.

Quanto ao ambiente de desenvolvimento integrado utilizado, haverá predominância do Visual Studio Code. Caso seja necessário a mudança de IDE, a mudança será feita para o NetBeans. Tanto o Visual Studio Code quanto o NetBeans são detentores de diversos plugins e ferramentas que auxiliam o desenvolvedor a escrever melhor o código e diferenciar partes dele, como diferenciar um atributo de um método no caso de uma linguagem orientada à objetos, ou uma variável de uma função no caso de uma linguagem funcional e/ou estruturada.

Quanto às linguagens utilizadas para o desenvolvimento de sistemas, estas serão HyperText Mark-up Language, Cascading Styling Sheets, JavaScript, PHP: HyperText Preprocessor e Structured Query Language. São linguagens padrão para o desenvolvimento de aplicações web, inclusive utilizadas no mercado.

Quanto aos frameworks utilizados, estes serão Bootstrap, Angular e Laravel. Essas abstrações que possuem pacotes de código são relevantes e muito utilizadas no mercado, cada uma adicionando aos projetos de sua própria forma, quando utilizados.

Quanto às ferramentas para diagramação UML e documentação de software, será utilizado predominantemente o Astah, porém caso seja necessário a mudança de ferramenta de documentação e diagramação, será feita a mudança para o Draw.io, ou LucidChart e/ou Microsoft Office. O Astah, que é uma ferramenta respeitada no mercado, produz diagramas UML com praticidade. O Draw.io e o LucidChart também produzem diagramas, porém já não tem a mesma posição do Astah enquanto software profissional. O Microsoft Office, principalmente o Word, é também uma ferramenta bem conhecida para criação e edição de documentos e, quando um software deve ser documentado na intenção de gerar documentos que não são diagramas, um editor de texto como o Word é bem-vindo.

Quanto ao banco de dados que será utilizado, trata-se do MySQL. O banco de dados MySQL é bem popular, ultrapassando 10 milhões de instalações pelo mundo e é mantido pela Oracle Corporation.

Quanto à ferramenta para modelar o banco de dados, será utilizado o BRModeloWeb para modelagem conceitual e o MySQL Workbench para modelagem física. O BRModeloWeb é uma solução web que traz muita praticidade ao usuário, uma vez que é hospedado na web, e cumpre uma boa parte das funcionalidades do BRModelo. Já o MySQL Workbench, vindo junto da instalação do MySQL, também compartilha dos mesmos status do banco de dados MySQL, porém referentes à modelagem física de um banco de dados.

Para apoiar o desenvolvimento do sistema, serão utilizados Postman e Git. O Postman consiste em uma plataforma de API para desenvolvedores para arquitetarem, construírem e testarem suas API's, portanto, uma vez utilizando o

framework Laravel como API, faz-se útil a utilização do mesmo. O Git é um sistema gerenciador de versão para auxiliar o desenvolvedor a controlar as mudanças do software em relação tanto a requisitos funcionais quanto não funcionais.

Para edição de imagem, será utilizado o GIMP. O GIMP é um software de código aberto voltado principalmente para a edição de imagens, opção interessante e robusta para aqueles que não desejam investir em um software de edição de imagem como o Adobe Photoshop, por exemplo.

Para uso de imagens de uso livre, essas serão consumidas dos sites Unsplash e Pixabay. Nem toda a imagem pode ser utilizada em um sistema web. As imagens utilizadas devem possuir licença de uso, e é em sites como os previamente citados que podemos ter essa segurança no uso de imagens.

5. O QUE SE ESPERAR

Espera-se alcançar até o final deste trabalho, da disciplina de Projeto Integrado para Desenvolvimento Web, a demonstração dos conhecimentos obtidos durante a minha jornada enquanto estudante no UniProjeção. Essa demonstração será exibida como um protótipo de um sistema web, o qual utilizará as ferramentas expostas no tópico de Solução Proposta.

A expectativa é desenvolver e ganhar experiência no uso de tecnologias e arquiteturas que vão além do escopo abordado no período a faculdade. Como exemplo, informo que estou atuando atualmente como pesquisador no Laboratório LATITUDE no projeto NSGC (Novo Sistema de Gestão Corporativa), ao qual são utilizadas tecnologias ainda não abordadas na faculdade, porém que já são utilizadas no mercado.

Portanto, o projeto atinge sua meta a partir do momento que um protótipo funcional é apresentado contendo as tecnologias e ferramentas citadas, pois requer uma grande carga de estudo para se compreender plenamente sobre API's, requisições HTTP envolvendo Angular e Laravel e criação dinâmica de formulários no framework Angular.

6. OBJETIVO GERAL

Este sistema trata-se da apresentação de uma plataforma de montagem de computadores, no qual o usuário deve se cadastrar para acessar o sistema. Uma vez cadastrado e autenticado, pode criar, listar, alterar e deletar configurações de computadores.

6.1 OBJETIVOS ESPECÍFICOS

Tendo como perspectiva e contexto o objetivo geral, têm-se os seguintes objetivos específicos, que se traduzem como requisitos funcionais:

RF01 - Realizar Cadastro

O sistema deverá permitir o cadastro do usuário no sistema por meio de campos(inputs). Caso o e-mail já conste no sistema, não deve ser permitido um cadastro duplicado.

RF02 - Realizar Login

O sistema deve permitir a autenticação do usuário no sistema por meio de dados em campos(inputs), considerando que o usuário já possua cadastro no sistema.

RF03 – Manter Processador

O sistema deve permitir, uma vez o usuário autenticado no sistema, a manutenção de processadores no sistema (CRUD – Create, Read, Update, Destroy).

RF04 – Manter Computador

O sistema deve permitir, uma vez o usuário autenticado no sistema, a manutenção de computadores no sistema (CRUD – Create, Read, Update, Destroy).

7. ESTRUTURA DO PROJETO

Este projeto foi dividido em três capítulos, onde será apresentado, de forma objetiva, o desenvolvimento de todo o projeto integrado de desenvolvimento web.

- O Capítulo I trata de expor uma visão inicial do contexto da aplicação, englobando seu contexto, a problemática que quer abordar, como se traduz em uma solução, e como, de fato, essa solução se concretizará no sistema por meio dos objetivos que deseja atingir;

- O Capítulo II especificará a modelagem, tanto física quanto conceitual, do banco de dados MySQL, que é um banco de dados relacional. A modelagem levará em conta o MER (Modelo Entidade Relacionamento).

- O Capítulo III detalhará o Documento de Visão. O documento de visão consistirá na documentação do sistema, que será feita baseada nos conhecimentos de Engenharia de Software. Poderá incluir layout de telas, casos de uso, entre tantas coisas;

- Ao final haverá a Conclusão, Referências Bibliográficas, Glossário Anexos e Apêndices. No Capítulo 1, houve necessidade de parafrasear alguns autores de publicações científicas, portanto as referências bibliográficas já começaram a ser desenvolvidas. Caso haja necessidade de incremento e/ou alteração ao seguir os capítulos, a edição das referências será feita.

CAPÍTULO II

MODELAGEM DE DADOS

A modelagem de dados, no contexto de Banco de Dados, visa descrever formalmente a estrutura de um banco de dados, com finalidade de promover uma conexão entre aspectos altamente técnicos e computacionais com o fenômeno da realidade que visa ser modelado e abstraído. A modelagem de dados compreenderá o Diagrama de Entidade Relacional, o Modelo de Entidade Relacional e o Dicionário de Dados.

O Diagrama de Entidade Relacional é utilizado para descrever as entidades envolvidas em um domínio de negócios, com seus atributos e como elas se relacionam entre si. A forma como se dá o modelo é abstrata, sendo que esse modelo visará sua implementação em um banco de dados relacional. Uma de suas grandes vantagens é a independência de SGBD.

Já o Modelo de Entidade-Relacional possuirá as tabelas de banco de dados, seus respectivos campos a serem preenchidos, quando um registro for inserido nas mesmas, suas respectivas ligações entre si mesmas, suas chaves primárias e estrangeiras com suas respectivas “constraints”, etc. Percebe-se que não se fala mais diretamente em entidades e atributos, mas sim em tabelas, chaves primárias e estrangeiras e campos, pois já estamos nos tratando de um modelo que está mais perto da solução computacional do que da solução de negócio.

2.1. Diagrama de Entidade Relacional

Abaixo, é apresentado o Diagrama de Entidade Relacional, produzido a partir da ferramenta BR Modelo Web. Neste diagrama, percebe a exposição das entidades que definem o problema que se pretende ser resolvido com maior atenção ao domínio do que da implementação técnica, já que é um diagrama que independe de SGBD.

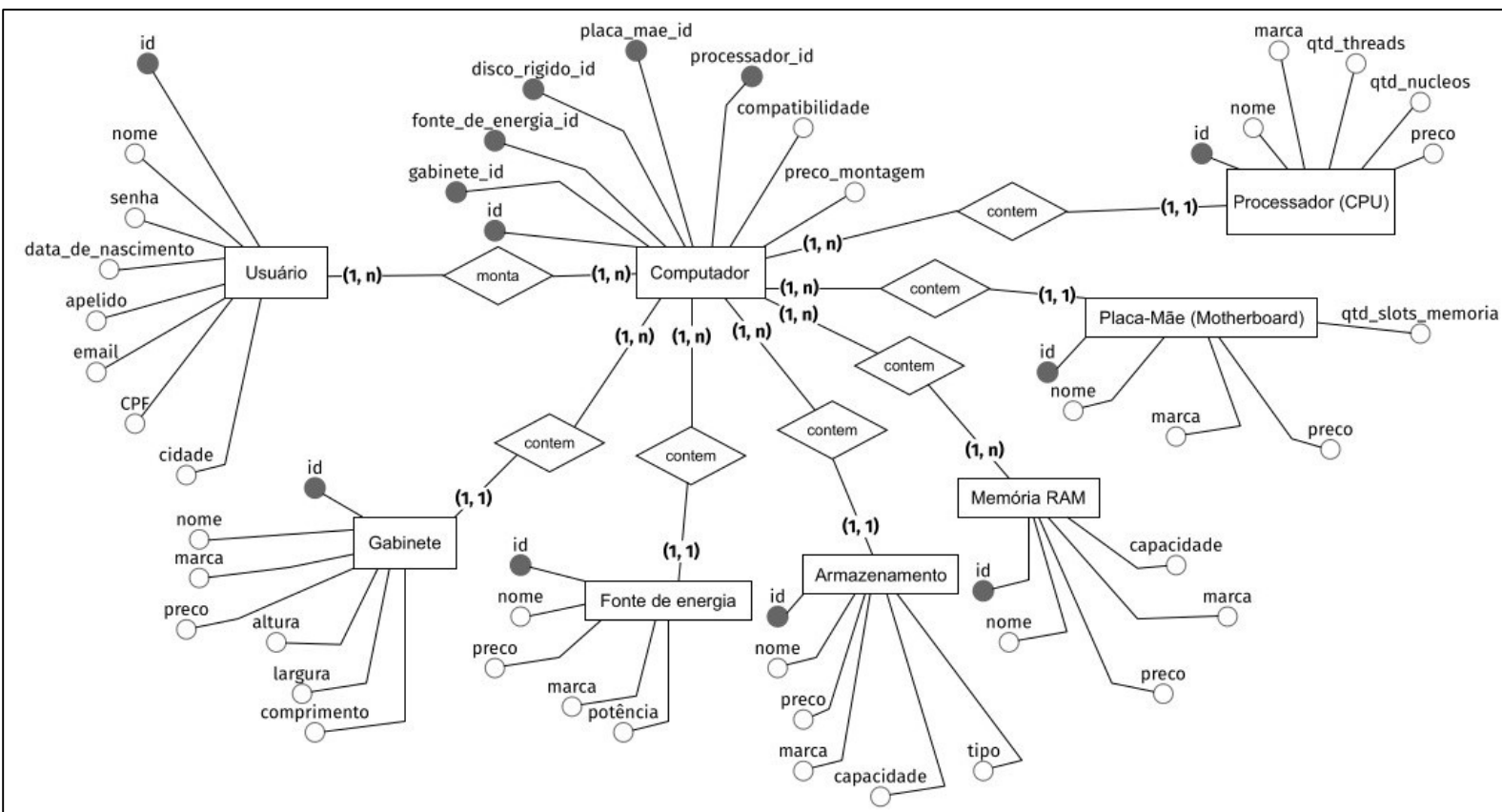


Figura 01 – Diagrama de Entidade Relacional
Fonte: BARBOSA, 2022.

2.2. Modelo de Entidade-Relacional

Na próxima página, é apresentado o Modelo de Entidade-Relacional, produzido a partir da ferramenta MySQL Workbench. Neste modelo, percebe-se a adição de quatro tabelas, uma referente aos acessos feitos às rotas do sistema (`log_acessos`), uma referente à execução do recurso de Migrations do framework Laravel (`migrations`), uma referente à implementação do relacionamento entre Usuários e Computadores (`configuracoes`) e uma referente à implementação do relacionamento entre Memórias RAM e Computadores (`qtd_pentes_no_computador`). Estas tabelas não foram declaradas como entidades no diagrama anterior, pois são pertinentes, uma vez se tratando do banco de dados com a perspectiva da solução computacional manifestada na implementação de códigos. As outras tabelas foram representadas no diagrama passado como entidades pois possuíam valor de domínio para a explicação de quais seriam as entidades, seus atributos e seus relacionamentos.

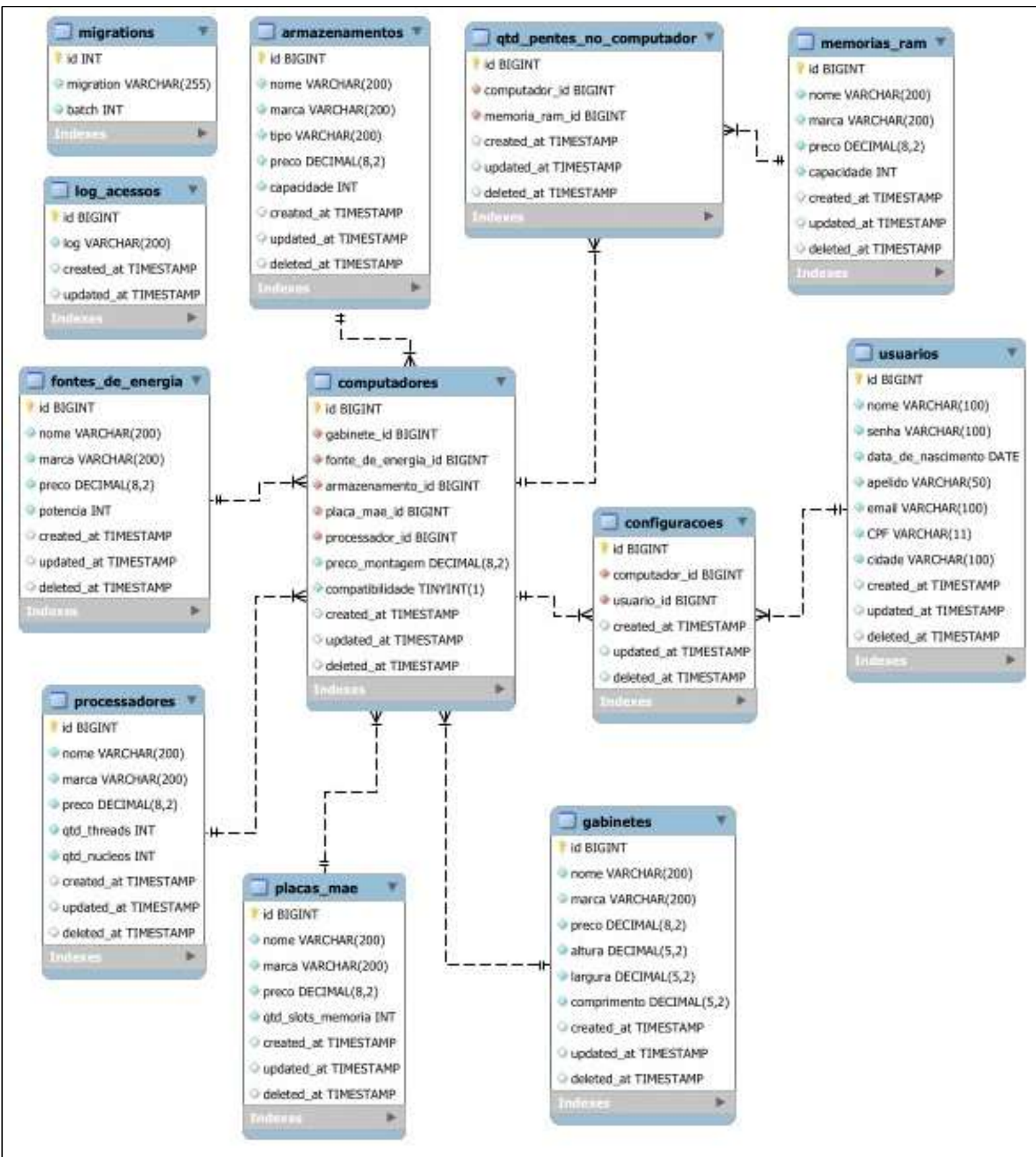


Figura 02– Diagrama de Banco de Dados: MER
Fonte: BARBOSA, 2022.

2.3 Dicionário de Dados

É importante que todo o banco de dados possua seu respectivo dicionário de dados. Este mesmo dicionário de dados consiste numa coleção de metadados, que são dados que descrevem os dados abordados em um contexto de banco de dados (dados sobre dados). Esses mesmos metadados contém definições do que são esses dados e pode conter representações sobre os elementos dos dados.

Quanto aos metadados que irão descrever os dados do Sistema Prometheus Web, é importante lembrar o uso do framework Laravel, que se utiliza da linguagem de programação PHP. As tabelas não foram criadas via script SQL, e sim via execução de código PHP (Migrations e Seeders). O propósito disto é a utilização do recurso Eloquent ORM, um mapeador relacional-objeto que, além de automatizar muitos comandos de bancos de dados (DDL, DML, entre tantos tipos), se relaciona bem com a arquitetura MVC, padrão rigorosamente seguido pelo framework.

Um dos impactos da criação das tabelas com Eloquent ORM via execução de Migrations impacta que, ao final de todas as tabelas possui a letra “s”, inferindo a ideia de plural, uma relação de registros. Outro é a inserção das colunas “created_at”, “updated_at” e “deleted_at” (traduzindo para o português: criado em, atualizado em, deletado em).

Uma grande adição com o uso do framework Laravel é a permissão de exclusão lógica ao invés de exclusão física. Ao invés de um registro ser apagado de fato do banco de dados (exclusão física), quando um registro for deletado, a coluna é preenchida com a data da exclusão, mantendo o registro no banco de dados e fazendo com que o registro se comporte como um registro não existente. Isso torna mais fácil a reversão de exclusões indesejadas.

2.3.1 Tabela *placas_mae*

Esta tabela visa armazenar dados sobre placas mãe armazenadas no sistema a serem utilizados para configuração de computadores. Estas são responsáveis pela comunicação do processador com todos os outros periféricos. Constitui uma placa de circuito impresso indispensável, tendo em vista um computador montado com todos os componentes necessários, encaixados em soquetes nessa mesma placa mãe.

Tabela 1 – Tabela “*placas_mae*”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da placa-mãe (chave primaria).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome da placa-mãe
marca	VARCHAR(200)	S	Este é o campo que guardará a marca da placa-mãe
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço da placa-mãe
qtd_slots_memoria	INT	S	Este é o campo que guardará a quantidade de slots de memória RAM da placa-mãe
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa placa-mãe. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa placa-mãe. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa placa-mãe. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
computadores	A tabela “ <i>placas_mae</i> ” possui relacionamento um para muitos com a tabela “ <i>computadores</i> ”, ou seja, uma placa-mãe pode estar inclusa em um ou múltiplos computadores, porém um computador só pode possuir uma única placa-mãe. O relacionamento é efetivado pela presença da chave estrangeira “ <i>placa_mae_id</i> ” na tabela “ <i>computadores</i> ”.		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.2 Tabela fontes_de_energia

Esta tabela visa armazenar dados sobre as fontes de energia armazenadas no sistema a serem utilizados para configuração de computadores. Essas mesmas fontes são responsáveis por converter a energia que vem da rede elétrica e distribuir, de forma adequada, a energia para os componentes.

Tabela 2 – Tabela “fontes_de_energia”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da fonte de energia (chave primaria).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome da fonte de energia
marca	VARCHAR(200)	S	Este é o campo que guardará a marca da fonte de energia
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço da fonte de energia
potencia	INT	S	Este é o campo que guardará a potência, em Watts, da fonte de energia
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa fonte de energia. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa fonte de energia. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa fonte de energia. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
computadores	A tabela “fontes_de_energia” possui relacionamento um para muitos com a tabela “computadores”, ou seja, uma fonte de energia pode estar inclusa em um ou múltiplos computadores, porém um computador só pode possuir uma única fonte de energia. O relacionamento é efetivado pela presença da chave estrangeira “fonte_de_energia_id” na tabela “computadores”.		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.3 Tabela *log_acessos*

Esta tabela visa armazenar log de acessos das rotas da API, portanto, não possuiria utilidade a presença de um recurso de exclusão lógica, já que o recurso de log tem utilidade para verificar o histórico de acesso de uma aplicação. A utilidade da coluna “updated_at” é a de sempre sua data ser igual à da coluna “created_at”. Caso haja diferença, isto poderá implicar em inconsistência nos registros da tabela.

Tabela 3 – Tabela “log_acessos”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do log de acesso (chave primaria).
log	VARCHAR(100)	S	Este é o campo que guardará o log de acesso. Conterá a rota acessada, o IP, e o método que foi acessado naquela rota.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente ao log. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente ao log. Coluna inserida pelo framework Laravel.

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.4 Tabela *migrations*

Esta tabela visa armazenar dados sobre o andamento da execução das “migrations”, que são os scripts do framework Laravel dedicados a manipular o banco de dados, seja em sua definição (DDL), quanto manipulação (DML). Esta é uma tabela puramente dedicada ao desenvolvedor de sistemas, portanto não é pertinente que ela inclua as colunas “created_at”, “updated_at” e “deleted_at”, além de ser uma tabela automaticamente criada pelo framework Laravel ao se executar a primeira migration e mantida na execução de migrations futuras.

Tabela 4 – Tabela “migrations”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da migration (chave primaria).
migration	VARCHAR/255	S	Este é o campo que guardará o nome da migration executada pelo framework.
batch	INT	S	Este é o campo que guardará o pacote(batch) em que a migration foi executada. Se a migration foi executada na primeira vez que se deu seu respectivo comando, seu valor será 1. Caso não seja a primeira vez, será um valor maior do que 1.

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.5 Tabela *armazenamentos*

Esta tabela visa armazenar dados sobre os componentes de armazenamento responsáveis por persistir dados após o computador ser desligado, como, por exemplo, HD's(Hard Disk) e SSD's (Solid State Drive).

Tabela 5 – Tabela “armazenamentos”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do armazenamento (chave primaria).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome do armazenamento
marca	VARCHAR(200)	S	Este é o campo que guardará a marca do armazenamento
tipo	VARCHAR(200)	S	Este é o campo que guardará o tipo do armazenamento (HD ou SSD)
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço do armazenamento
capacidade	INT	S	Este é o campo que guardará a capacidade, em GigaBytes, do armazenamento
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à um certo armazenamento. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à um certo armazenamento. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à um certo armazenamento. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
computadores	A tabela “armazenamentos” possui relacionamento um para muitos com a tabela “computadores”, ou seja, um armazenamento pode estar incluso em um ou múltiplos computadores, porém um computador, no contexto desse sistema, só pode possuir um único armazenamento. O relacionamento é efetivado pela presença da chave estrangeira “armazenamento_id” na tabela “computadores”.		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.6 Tabela *processadores*

Esta tabela visa armazenar dados sobre os processadores armazenados no sistema a serem utilizados para configuração de computadores. Tal como a placa-mãe, o processador é indispensável para a constituição de um computador. É responsável pelos cálculos e tomadas de decisão de um computador.

Tabela 6 – Tabela “processadores”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do processador (chave primaria).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome do processador
marca	VARCHAR(200)	S	Este é o campo que guardará a marca do processador
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço do processador
qtd_nucleos	INT	S	Este é o campo que guardará a quantidade de núcleos do processador
qtd_threads	INT	S	Este é o campo que guardará a quantidade de threads do processador
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à um certo processador. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à um certo processador. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à um certo processador. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.

RELACIONAMENTOS	
Tabela	Descrição
computadores	A tabela “processadores” possui relacionamento um para muitos com a tabela “computadores”, ou seja, um processador pode estar incluso em um ou múltiplos computadores, porém um computador só pode possuir um único processador. O relacionamento é efetivado pela presença da chave estrangeira “processador_id” na tabela “computadores”.

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.7 Tabela *gabinetes*

Esta tabela visa armazenar dados sobre os gabinetes armazenados no sistema a serem utilizados para configuração de computadores. Todos os computadores, quando utilizados em cenários reais, costumam possuir um gabinete que reúne todas suas peças devidamente montadas.

Tabela 7 – Tabela “gabinetes”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do gabinete (chave primária).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome do gabinete
marca	VARCHAR(200)	S	Este é o campo que guardará a marca do gabinete
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço do gabinete
altura	DECIMAL(5,2)	S	Este é o campo que guardará a altura do gabinete (Dimensão do gabinete)
largura	DECIMAL(5,2)	S	Este é o campo que guardará a largura do gabinete (Dimensão do gabinete)
comprimento	DECIMAL(5,2)	S	Este é o campo que guardará o comprimento do gabinete (Dimensão do gabinete)
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à um certo gabinete. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à um certo gabinete. Coluna inserida pelo framework Laravel.

deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à um certo gabinete. Coluna inserida pelo framework Laravel via biblioteca "SoftDeletes".
RELACIONAMENTOS			
Tabela	Descrição		
computadores	A tabela "gabinetes" possui relacionamento um para muitos com a tabela "computadores", ou seja, um gabinete pode estar incluso em um ou múltiplos computadores, porém um computador só pode possuir um único gabinete. O relacionamento é efetivado pela presença da chave estrangeira "gabinete_id" na tabela "computadores".		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.8 Tabela *configuracoes*

Esta tabela visa armazenar dados sobre as configurações de computador que um usuário em específico possui relacionado a ele. É importante ressaltar que é uma tabela que implementa o relacionamento muitos para muitos de usuários para computadores. Uma vez que um usuário associe um computador à si mesmo na aplicação, deve ser registrada, na tabela “*configuracoes*”, as chaves estrangeiras necessárias (*computador_id* e *usuario_id*).

Tabela 8 – Tabela “*configuracoes*”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da configuração (chave primária).
usuario_id	BIGINT	S	Este é o campo que guardará o identificador único do usuário (chave estrangeira).
computador_id	BIGINT	S	Este é o campo que guardará o identificador único do computador (chave estrangeira).
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa configuração. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa configuração. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa configuração. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
usuarios	A tabela “ <i>configuracoes</i> ” possui relacionamento um para muitos com a tabela “ <i>usuarios</i> ”, implementando o relacionamento muitos para muitos de “ <i>usuarios</i> ” para “ <i>computadores</i> ”. Conclui-se que, uma configuração está associada somente a um usuário, porém um usuário pode possuir várias configurações associadas à certos computadores, ou seja, possuir mais de uma configuração. Quanto à esse relacionamento, verifica-se a presença da chave estrangeira “ <i>usuario_id</i> ” na tabela “ <i>configuracoes</i> ”.		
computadores	A tabela “ <i>configuracoes</i> ” possui relacionamento um para muitos com a tabela “ <i>computadores</i> ”, implementando o relacionamento muitos para muitos de “ <i>computadores</i> ” para “ <i>usuarios</i> ”. Conclui-se que, uma configuração está associada somente a um computador, porém um computador pode possuir várias configurações associadas à certos usuários, ou seja, possuir mais de uma configuração. Quanto à esse relacionamento, verifica-se a presença da chave estrangeira “ <i>computador_id</i> ” na tabela “ <i>configuracoes</i> ”.		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.9 Tabela *usuarios*

Esta tabela visa armazenar dados sobre os usuários do sistema. Seu CPF deve passar pela validação dos dois últimos dígitos verificadores e seu e-mail deve possuir a configuração “nome@domínio.com”.

Tabela 9 – Tabela “usuarios”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do usuário
nome	VARCHAR(100)	S	Este é o campo que guardará o nome do usuário
senha	VARCHAR(100)	S	Este é o campo que guardará a senha criptografada do usuário.
data_de_nascimento	DATE	S	Este é o campo que guardará a data de nascimento do usuário.
apelido	VARCHAR(50)	S	Este é o campo que guardará o apelido do usuário.
email	VARCHAR(100)	S	Este é o campo que guardará o endereço de e-mail do usuário.
CPF	VARCHAR(11)	S	Este é o campo que guardará o CPF do usuário sem pontos e traço.
cidade	VARCHAR(100)	S	Este é o campo que guardará a cidade do usuário
created_at	TIMESTAMP	N	Este é o campo que guardará a data de criação do cadastro do usuário ou a data da inserção do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data de alteração do cadastro do usuário ou a data da edição do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data de exclusão do cadastro do usuário ou a data da exclusão do registro correspondente à um certo usuário. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
tentativas	A tabela “usuarios” possui relacionamento muitos para muitos com a tabela “perguntas”, que é implementada via tabela “tentativas” com um relacionamento		

	<p>um para muitos. Ou seja, um usuário pode possuir várias tentativas, porém, uma tentativa está associada, por vez, a um único usuário.</p> <p>O relacionamento é efetivado pela presença da chave estrangeira “usuario_id” na tabela “tentativas”.</p>
--	--

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.10 Tabela *memorias_ram*

Esta tabela visa armazenar dados sobre as memórias RAM armazenadas no sistema a serem utilizados para configuração de computadores. É responsável pelo armazenamento temporário de informações digitais utilizadas pelo processador.

Tabela 10 – Tabela “memorias_ram”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da memória RAM(chave primaria).
nome	VARCHAR(200)	S	Este é o campo que guardará o nome da memória RAM
marca	VARCHAR(200)	S	Este é o campo que guardará a marca da memória RAM
preco	DECIMAL(8,2)	S	Este é o campo que guardará o preço da memória RAM
capacidade	INT	S	Este é o campo que guardará a capacidade, em GigaBytes, da memória RAM
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa memória RAM. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa memória RAM. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa memória RAM. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
qtd_pentes_no_computador	A tabela “memorias_ram” possui relacionamento muitos para muitos com a tabela “computadores”, que é implementada via tabela “qtd_pentes_no_computador” com um relacionamento um para muitos. Ou seja, uma memória RAM pode estar em uma ou mais quantidades em um computador, porém, um slot de memória RAM (pente) somente comportará uma memória RAM por vez. O relacionamento é efetivado pela presença da chave estrangeira “memoria_ram_id” na tabela “qtd_pentes_no_computador”.		

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.11 Tabela *computadores*

Esta tabela visa armazenar dados sobre os computadores perguntas do sistema de montagem de computadores.

Tabela 11 – Tabela “computadores”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único do computador (chave primaria).
gabinete_id	BIGINT	S	Este é o campo que guardará o identificador único do gabinete (chave estrangeira).
fonte_de_energia_id	BIGINT	S	Este é o campo que guardará o identificador único da fonte de energia (chave estrangeira).
armazenamento_id	BIGINT	S	Este é o campo que guardará o identificador único do armazenamento (chave estrangeira).
placa_mae_id	BIGINT	S	Este é o campo que guardará o identificador único da placa mãe (chave estrangeira).
processador_id	BIGINT	S	Este é o campo que guardará o identificador único do processador(chave estrangeira).
preco_montagem	DECIMAL(8,2)	S	Este é o campo que guardará o preço de montagem desse computador.
compatibilidade	TINYINT(1)	S	Este é o campo que guardará se as peças desse computador são compatíveis entre si ou não.
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel.

updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa pergunta. Coluna inserida pelo framework Laravel via biblioteca "SoftDeletes".
RELACIONAMENTOS			
Tabela	Descrição		
armazenamentos	A tabela "computadores" possui relacionamento um para muitos com a tabela "armazenamentos", ou seja, um computador pode possuir, no contexto do sistema, somente um armazenamento, porém um armazenamento pode estar associado a vários computadores. O relacionamento é efetivado pela presença da chave estrangeira "armazenamento_id" na tabela "computadores".		
fontes_de_energia	A tabela "computadores" possui relacionamento um para muitos com a tabela "fontes_de_energia", ou seja, um computador pode possuir, no contexto do sistema, somente uma fonte de energia, porém uma fonte de energia pode estar associada a vários computadores. O relacionamento é efetivado pela presença da chave estrangeira "fonte_de_energia_id" na tabela "computadores".		
processadores	A tabela "computadores" possui relacionamento um para muitos com a tabela "processadores", ou seja, um computador pode possuir somente um processador, porém um processador pode estar associado a vários computadores. O relacionamento é efetivado pela presença da chave estrangeira "armazenamento_id" na tabela "computadores".		
placas_mae	A tabela "computadores" possui relacionamento um para muitos com a tabela "placas_mae", ou seja, um computador pode possuir somente uma placa-mãe, porém uma placa-mãe pode estar associada a vários computadores. O relacionamento é efetivado pela presença da chave estrangeira "placa_mae_id" na tabela "computadores".		
gabinetes	A tabela "computadores" possui relacionamento um para muitos com a tabela "gabinetes", ou seja, um computador pode possuir somente um gabinete, porém um gabinete pode estar associado a vários computadores. O relacionamento é efetivado pela presença da chave estrangeira "gabinete_id" na tabela "computadores".		
configuracoes	A tabela "computadores" possui relacionamento muitos para muitos com a tabela "usuarios", que é implementada via tabela "configuracoes" com um relacionamento um para muitos. Ou seja, um computador pode possuir várias configurações, porém, uma configuração possui, por vez, um único computador. O relacionamento é efetivado pela presença da chave estrangeira "computador_id" na tabela "configuracoes".		
qtd_pentes_no_computador	A tabela "computadores" possui relacionamento muitos para muitos com a tabela "memorias_ram", que é implementada via tabela "qtd_pentes_no_computador" com um relacionamento um para muitos. Ou seja, um computador pode possuir vários slots para inserção de memórias RAM, porém, um slot está associado à um computador por vez.		

O relacionamento é efetivado pela presença da chave estrangeira "computador_id" na tabela "qtd_pentes_no_computador".

Fonte: BARBOSA, Lucas da Silva, 2022.

2.3.12 Tabela *qtd_pentes_no_computador*

Esta tabela visa armazenar dados sobre a quantidade de slots possíveis para pentes de memória RAM em um. É importante ressaltar que é uma tabela que implementa o relacionamento muitos para muitos de memórias RAM para computadores. Uma vez que uma memória RAM esteja associada à um computador, deve ser registrada, na tabela “*qtd_pentes_no_computador*”, as chaves estrangeiras necessárias (*computador_id* e *memoria_ram_id*).

Tabela 12 – Tabela “*qtd_pentes_no_computador*”

Campo	Tipo/Tamanho	Obrigatório	Comentário
id	BIGINT	S	Este é o campo que guardará o identificador único da quantidade de pentes disponíveis por computador (chave primária).
computador_id	BIGINT	S	Este é o campo que guardará o identificador único do computador (chave estrangeira).
memoria_ram_id	BIGINT	S	Este é o campo que guardará o identificador único da memória RAM(chave estrangeira).
created_at	TIMESTAMP	N	Este é o campo que guardará a data da inserção do registro correspondente à uma certa quantidade de pentes disponíveis por computador. Coluna inserida pelo framework Laravel.
updated_at	TIMESTAMP	N	Este é o campo que guardará a data da edição do registro correspondente à uma certa quantidade de pentes disponíveis por computador. Coluna inserida pelo framework Laravel.
deleted_at	TIMESTAMP	N	Este é o campo que guardará a data da exclusão do registro correspondente à uma certa quantidade de pentes disponíveis por computador. Coluna inserida pelo framework Laravel via biblioteca “SoftDeletes”.
RELACIONAMENTOS			
Tabela	Descrição		
memorias_ram	A tabela “ <i>qtd_pentes_no_computador</i> ” possui relacionamento um para muitos com a tabela “ <i>memorias_ram</i> ”, implementando o relacionamento muitos para muitos de “ <i>memorias_ram</i> ” para “ <i>computadores</i> ”. Conclui-se que, um slot de memória RAM disponível está associado somente a uma memória RAM, porém uma memória RAM pode possuir vários slots de memória RAM disponíveis, estes por sua vez associados à certos computadores. Quanto a esse relacionamento, verifica-se a presença da chave estrangeira “ <i>memoria_ram_id</i> ” na tabela “ <i>qtd_pentes_no_computador</i> ”.		

computadores	<p>A tabela “qtd_pentes_no_computador” possui relacionamento um para muitos com a tabela “computadores”, implementando o relacionamento muitos para muitos de “computadores” para “usuarios”. Conclui-se que, um slot de memória RAM está associado somente a um computador, porém um computador pode possuir vários slots de memória RAM disponíveis, estes por sua vez associados à certas memórias RAMs.</p> <p>Quanto a esse relacionamento, verifica-se a presença da chave estrangeira “computador_id” na tabela “qtd_pentes_no_computador”.</p>
--------------	--

Fonte: BARBOSA, Lucas da Silva, 2022.

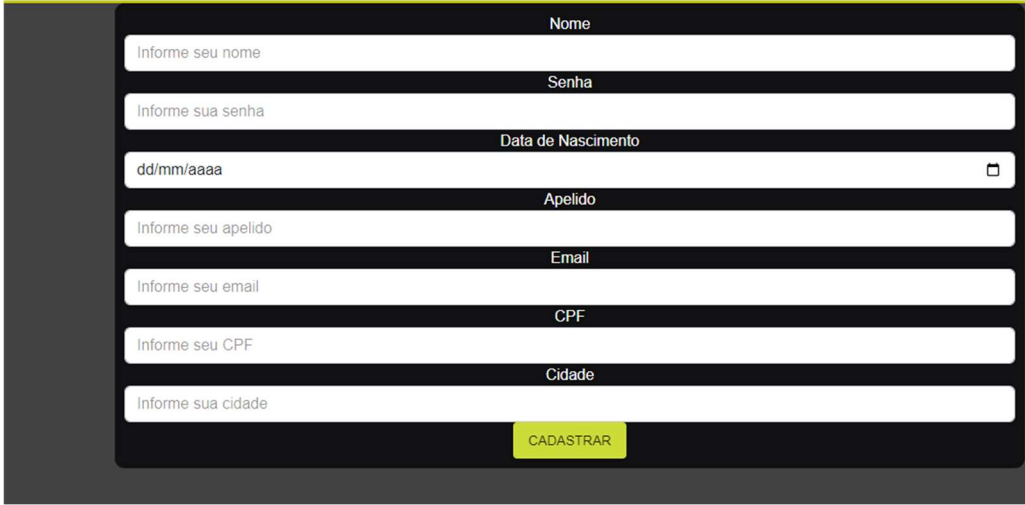
CAPÍTULO III

O Sistema Prometheus Web consiste num sistema que viabiliza a montagem de computadores a partir de hardware e pretende possibilitar que o usuário, uma vez cadastrado e autenticado, tenha acesso à computadores e processadores para que possa controlá-los. O sistema deve ser visualizado por meio de um navegador, como o Google Chrome, já que usa tecnologias como o HTML, que é uma linguagem de marcação que é processada em navegadores.

A criação do sistema se inspira no fato de que há pessoas que demandam um melhor entendimento de montagem e arquitetura de computadores devido à presente tecnologia na sociedade em forma de diversos tipos de computadores que vão até mesmo além dos desktops. A demanda por profissionais de Tecnologia da Informação só tem aumentado, o que certamente é visível nas mudanças da sociedade e de seus processos. Futuramente, será de extrema utilidade que haja um maior entendimento sobre hardware por conta da crescente presença de artefatos inteligentes, como geladeiras inteligentes, televisões *smart*, etc. Esse entendimento pode começar com o entendimento de um computador desktop e evoluir a partir desse ponto.

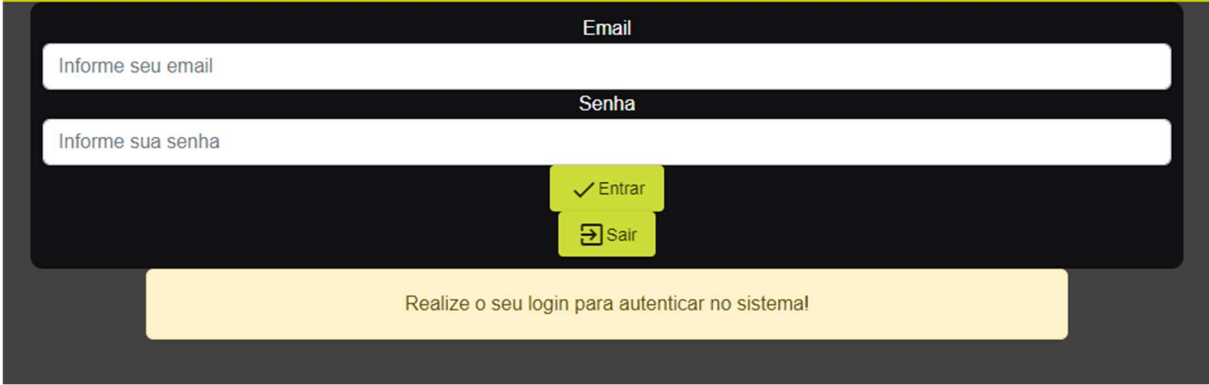
3.1. Conjunto de Tabelas

Tabela 1 – Realizar Cadastro

Realizar Cadastro
Caso de Uso: UC001 – Realizar Cadastro

Funcionalidades:
RF001 – Realizar Cadastro
<p>Esta tela tem como funcionalidade servir um formulário destinado à entrada de dados relacionados ao cadastro de usuário</p> <ol style="list-style-type: none"> 1. No campo “Nome” deve ser informado um nome com no mínimo 3 caracteres e no máximo 40 caracteres. Além desses requisitos, o nome de usuário deve ser único no banco de dados e é um campo obrigatório. 2. No campo “Senha” deve ser informado uma senha com no mínimo 3 dígitos e no máximo 40 dígitos, além de ser um campo obrigatório. 3. No campo “Data de Nascimento” deve ser informado uma data e é um campo obrigatório. 4. No campo “Apelido” deve ser informado um apelido com no mínimo 3 caracteres e no máximo 40 caracteres, além de ser um campo obrigatório. 5. No campo “Email” deve ser informado um e-mail possuindo os caracteres “@” e “.” para delimitar o nome e o domínio. Além desses requisitos, o e-mail do usuário deve ser único no banco de dados e é um campo obrigatório. 6. No campo “CPF” deve ser informado um número de CPF que seja válido (dois últimos dígitos verificadores). Além desses requisitos, o CPF do usuário deve ser um CPF único no banco de dados e é um campo obrigatório. 7. No campo “Cidade” deve ser informada uma cidade com no mínimo 3 caracteres e no máximo 40 caracteres, além de ser um campo obrigatório.

Fonte: BARBOSA, 2022.

Tabela 2 – Realizar Login

Realizar Login
Caso de Uso: UC002 – Realizar Login
 <p>The image shows a login interface on a dark background. It features two white input fields: the top one is labeled 'Email' with the placeholder 'Informe seu email', and the bottom one is labeled 'Senha' with the placeholder 'Informe sua senha'. Below these fields are two green buttons: 'Entrar' with a checkmark icon and 'Sair' with a door icon. At the bottom of the interface is a yellow rectangular box containing the text 'Realize o seu login para autenticar no sistema!'.</p>
Funcionalidades:
RF002 – Realizar Login
<p>Esta tela tem como funcionalidade servir um formulário destinado à entrada de dados relacionados a autenticação de um certo usuário para que possa utilizar o sistema.</p> <ol style="list-style-type: none"> 1. No campo “Email” deve ser informado um e-mail possuindo os caracteres “@” e “.” para delimitar o nome e o domínio. Além desses requisitos, o e-mail do usuário deve ser único no banco de dados e é um campo obrigatório. 2. No campo “Senha” deve ser informado uma senha com no mínimo 3 dígitos e no máximo 40 dígitos, além de ser um campo obrigatório.

Fonte: BARBOSA, 2022.







Mentor Process

Caso de Uso: UC003 – Manter Processador

Listar Processador

Criar Processador

Listar Processador ?

Identificador	Nome	Marca	Quantidade de núcleos	Quantidade de threads	Preço	Editar	Excluir
1	Processador Intel Core i3-10105F, 3.7Ghz (4.4Ghz Turbo) Cache 6MB, LGA1200	Intel	4	8	589.90	 Atualizar Processador 1	 Deletar Processador 1
2	Processador Intel Core i7-11700	Intel	8	16	499.90	 Atualizar Processador 2	 Deletar Processador 2
3	Processador Intel Core i5-10400F,	Intel	6	12	699.90	 Atualizar Processador	 Deletar Processador

RF003 – Manter Processador

Esta tela tem como funcionalidade servir uma tabela destinada ao controle de dados relacionados a todos os processadores existentes no sistema.

1. Caso o usuário possua dúvida sobre o que representa cada coluna, com o Identificador, Nome, e as demais colunas, há um botão com um sinal de interrogação ao lado do título da tabela. Ao clicar no botão, é aberta uma janela com explicação sobre as operações e as colunas.
2. Ao final de um registro, há dois botões, um responsável pela edição no processador e outro pela exclusão.
3. Caso o usuário opte por editar o processador, ele pode clicar no botão “Atualizar Processador” e será aberta uma janela disponibilizando um formulário para a entrada de dados relacionadas à atualização, ou completa ou parcial, daquele processador em específico.
4. Já caso o usuário deseje excluir um processador, ele pode clicar no botão “Deletar Processador” e será aberta uma janela relacionada à exclusão daquele processador em específico.
5. Caso o usuário queira criar um processador novo, ele deve clicar no botão “Criar Processador”, que se encontra ao lado de “Listar Processador”. Nesse botão estará disposto um formulário destinado à entrada de dados relacionados ao cadastro de um processador. Os campos que devem ser preenchidos são também explicados em um botão com sinal de interrogação caso o usuário necessite das explicações no momento de execução do aplicativo.
6. Quanto ao formulário de criação de processador:

- a No campo “Nome do Processador” deve ser informado um nome que tenha no mínimo 3 caracteres e ser único no banco de dados, além de ser um campo obrigatório.
- b No campo “Marca do Processador” deve ser informado uma marca que tenha no mínimo 3 caracteres, além de ser um campo obrigatório.
- c No campo “Preço do Processador” deve ser informado um preço, ou seja, deve ser numérico, e deve ter no mínimo 3 dígitos. Também constitui um campo obrigatório.
- d No campo “Quantidade de Núcleos do Processador” deve ser informado um número e é um campo obrigatório.
- e No campo “Quantidade de Threads do Processador” deve ser informado um número e é um campo obrigatório.

Fonte: BARBOSA, 2022.

Tabela 4 – Manter Computador

Manter Computador

Caso de Uso: UC004 – Manter Computador

Listar Computador

Criar Computador

Associar Computador

Listar Computador

ID do Computador	ID do Gabinete	Nome do Gabinete	ID da Fonte de Energia	Nome da Fonte de Energia	ID do Armazenamento	Nome do Armazenamento	ID da Placa Mãe	Nome da Placa Mãe	ID do Processador	Nome do Processador	Preço de Montagem	Compatibilidade entre Peças	ID da Memória RAM	Nome da Memória RAM	Quantidade de Memória(s) RAM	Editar	Excluir
1	1	Gabinete Gamer TGT Carbon, Rainbow, Mid Tower, Lateral de Vidro, Preto	1	Fonte Corsair VS600, 500W, 80 Plus White	1	HD WD Blue 1TB 3.5" SATA III 6GB/s	1	Placa Mãe Biostar H510MHP, DDR4, Socket LGA1200, M-ATX, Chipset Intel H510	1	Processador Intel Core i3-10105F, 3.7Ghz (4.4Ghz Turbo) Cache 6MB, LGA1200	450.00	Compatível	1	Memoria Team Group T-Force Vulcan Pichau, DDR4, 3000Mhz, Vermelha	1 pente	<div>Atualizar Computador 1</div>	<div>Deletar Computador 1</div>
2	2	Gabinete Acer Predator Orion 5000, Preto	2	750 W MWE Bronze series (PFC ativo), 80 PLUS Bronze, ATX 12V V2.5 BIVOLT	2	SSD GIGABYTE 1TB M.2 2280 PCIe 3.0 X4 NVME	2	Placa Mãe Intel B560 Predator com RGB	2	Processador Intel Core i7-11700	500.00	Compatível	2	Memoria Gamer Predator Orion 5000, DDR4, 3200 MHz	2 pentes	<div>Atualizar Computador 2</div>	<div>Deletar Computador 2</div>

Funcionalidades:

RF004 – Manter Computador

Esta tela tem como funcionalidade servir uma tabela destinada ao controle de dados relacionados a todos os computadores existentes no sistema que estão relacionados usuário.

1. Caso o usuário possua dúvida sobre o que representa cada coluna, como ID Computador, ID do Gabinete, e as demais colunas, há um botão com um sinal de interrogação ao lado do título da tabela. Ao clicar no botão, é aberto uma janela com explicações sobre as operações e as colunas.
2. Ao final de um registro, há dois botões, um responsável pela edição do computador e outro pela exclusão.
3. Caso o usuário opte por editar o computador, ele pode clicar no botão “Atualizar Computador” e será aberto uma janela disponibilizando um formulário para entrada de dados relacionadas à atualização, ou completa ou parcial, daquele computador específico.
4. Já caso o usuário deseje excluir um computador, ele pode clicar no botão “Deletar Computador” e será aberto uma janela relacionada à exclusão daquele computador em específico.
5. Caso o usuário queira criar um processador novo, ele deve clicar no botão “Criar Computador”, que se encontra entre os botões “Listar Computador” e “Associar Computador”. Nesse botão, estará disposto um formulário destinado à entrada de dados relacionados ao cadastro de um computador. Os campos que devem ser preenchidos são também explicados em um botão com sinal de interrogação caso o usuário necessite das explicações no momento de execução do aplicativo.
6. Quanto ao botão “Associar Computador”, é nele que haverá a tabela contendo todos os computadores do sistema independente de associação com usuários. O usuário pode ir nessa tabela e associar computadores para si. Esse comportamento existe visando emular a modelagem do banco de dados que dispõe uma relação M:N entre usuários e computadores.
7. Quanto ao formulário de criação de computador:

- a No campo “Identificador do Gabinete” deve ser informado um número de identificador único para gabinete que exista no banco de dados. Constitui um campo obrigatório.
- b No campo “Identificador da Fonte de Energia” deve ser informado um número de identificador único para fonte de energia que exista no banco de dados. Constitui um campo obrigatório.
- c No campo “Identificador do Armazenamento” deve ser informado um número de identificador único para armazenamento que exista no banco de dados. Constitui um campo obrigatório.
- d No campo “Identificador da Placa Mãe” deve ser informado um número de identificador único para placa mãe que exista no banco de dados. Constitui um campo obrigatório.
- e No campo “Identificador da Memória RAM” deve ser informado um número de identificador único para memória RAM que exista no banco de dados. Constitui um campo obrigatório.
- f No campo “Quantidade da Memória RAM” deve ser informado um número para a quantidade de pentes de memória, não podendo ser menor que 1 e maior do que a capacidade da placa mãe. Constitui um campo obrigatório.
- g No campo “Preço para Montagem das peças” deve ser informado preço, ou seja, um dado numérico. Constitui um campo obrigatório.
- h No campo “Compatibilidade” deve ser marcado ou não, visando que é um campo que trata se as peças informadas são compatíveis ou não entre si, constituindo um computador real. Constitui um campo booleano e obrigatório.

Fonte: BARBOSA, 2022.

3.2. Diagrama de Caso de Uso

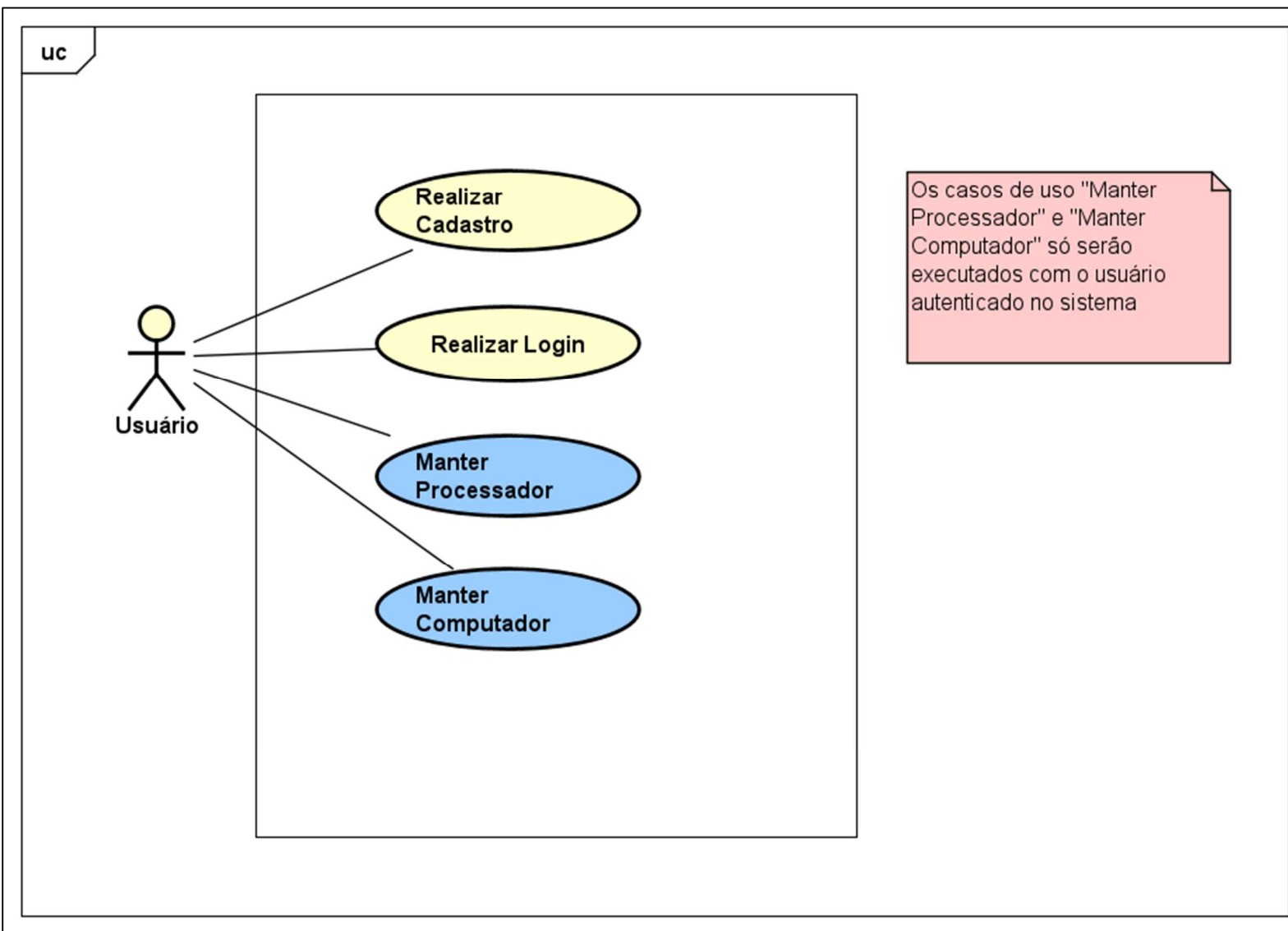


Figura 03– Diagrama de Caso de Uso
Fonte: BARBOSA, 2022.

3.3. Conjunto de Descrições de Caso de Uso

3.3.1 Realizar Cadastro

Nome UC	<i>Realizar Cadastro</i>
Atores	<i>Usuário</i>
Participação do ator	Permite o registro em banco de dados das informações de um certo usuário
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Prometheus com a finalidade de fazer um cadastro no sistema para acessar mais funcionalidades. Uma vez na página “/home” do sistema, o usuário deve clicar em “Cadastro” para ser redirecionado à página “/realizar-cadastro”. É nesta página que o cadastro pode ser realizado.	
Pós-condições	
O <i>Usuário</i> está cadastrado no banco de dados do Sistema Prometheus.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Cadastro”
2.	O sistema carrega uma tela contendo o formulário de cadastro
3.	O <i>Usuário</i> entra com seu nome, CPF, senha, e-mail, apelido, data de nascimento e cidade.
4.	O <i>Usuário</i> submete o cadastro das informações pessoais
5.	O sistema valida a entrada de dados (CPF, senha, e-mail, ...).
6.	O sistema emite a mensagem de indicação de sucesso
7.	O caso de uso é finalizado
Fluxos Alternativos	
<u>A01 - O usuário cancela o cadastro</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém o formulário de cadastro do usuário.
2.	O sistema cancela a operação
3.	O caso de uso é finalizado
Fluxos de Exceção	
<u>E01 – O usuário já está cadastrado</u>	
1.	O sistema verifica que o CPF e/ou o e-mail inserido já foi utilizado alguma vez.
2.	O sistema envia a mensagem referente ao campo já utilizado.”

3.	O sistema retorna ao fluxo principal (P3) para reentrada do(s) dado(s) do <i>Usuário</i>
E02 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido
2.	Após o processo de validação, o sistema impede a requisição referente ao cadastro do <i>Usuário</i> .
3.	O sistema retorna ao fluxo principal (P3) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando
E03 – <u>O CPF do usuário é inválido</u>	
1.	O <i>Usuário</i> entra com número de CPF inválido;
2.	Após o processo de validação, o sistema impede a requisição referente ao cadastro de <i>usuário</i> .”
3.	O sistema retorna ao fluxo principal (P3) para entrada do CPF do <i>Usuário</i> que está inválido

3.3.2 Realizar Login

Nome UC	<i>Realizar Login</i>
Atores	<i>Usuário</i>
Participação do ator	Permite a autenticação de um certo usuário no sistema de acordo com as informações persistidas no banco de dados
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Prometheus com a finalidade de fazer login no sistema para acessar mais funcionalidades. Uma vez na página “/home” do sistema, o usuário deve clicar em “Login” para ser redirecionado à página “/realizar-login”. É nesta página que a autenticação pode ser realizada.	
Pós-condições	
O <i>Usuário</i> está autenticado no Sistema Prometheus, o que lhe permitirá acessar mais funcionalidade.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Login”
2.	O sistema carrega uma tela contendo o formulário de login
3.	O <i>Usuário</i> entra com seu e-mail e senha.
4.	O <i>Usuário</i> submete o formulário de login
5.	O sistema valida a entrada de dados.
6.	O sistema emite a mensagem de indicação de sucesso
7.	O caso de uso é finalizado
Fluxos Alternativos	
A01 - <u>O usuário cancela o login</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém o formulário de login
2.	O sistema cancela a operação
3.	O caso de uso é finalizado
Fluxos de Exceção	
E01 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido
2.	Após o processo de validação, o sistema impede a requisição referente ao login
3.	O sistema retorna ao fluxo principal (P3) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando
E02 – <u>As credenciais são inválidas</u>	

1.	O <i>Usuário</i> entra com uma combinação de e-mail e senha que não há no sistema
2.	Após o processo de validação, o sistema responde a requisição apontando que a combinação não há no sistema
3.	O sistema retorna ao fluxo principal (P3) para uma nova entrada de credenciais que sejam válidas.

3.3.3 Manter Processador

Nome UC	<i>Manter Processador</i>
Atores	<i>Usuário</i>
Participação do ator	Permite que um certo usuário controle as perguntas no sistema, podendo criá-las, alterá-las, deletá-las e fazer uma leitura de todos os processadores.
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Prometheus com a finalidade de controlar todos os processadores do sistema. Uma vez na página “/manter-processador” do sistema e devidamente autenticado, o <i>Usuário</i> pode prosseguir com suas operações.	
Pós-condições	
O <i>Usuário</i> modificou o(s) processador(es) do Sistema Prometheus como desejou.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Controlar Processador”.
2.	O sistema carrega uma tela contendo uma tabela com todas as perguntas e dois botões, um para listar todas as perguntas e outro para criar uma nova pergunta.
3.	O <i>Usuário</i> clica no botão “Criar Processador”.
4.	O sistema apresenta um formulário destinado à criação de um novo processador.
5.	O <i>Usuário</i> preenche todos os campos necessários.
6.	O <i>Usuário</i> submete o formulário.
7.	O sistema exibe uma mensagem referente à criação da pergunta.
8.	O caso de uso é finalizado.
Fluxos Alternativos	
A01 - <u>O usuário cancela o cadastro do processador</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém ou a tabela listando todas as perguntas ou o formulário de criação de processador.
2.	O sistema cancela a operação.
3.	O caso de uso é finalizado.
A02 - <u>O usuário deseja ver a listagem dos processadores</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Processador”, mantendo-se na tela responsável pela listagem de processadores.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o processador de seu interesse e lê seus dados.
3.	O caso de uso é finalizado.

A03 - <u>O usuário deseja atualizar um processador</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Processador”, mantendo-se na tela responsável pela listagem de processadores.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o processador de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente ao processador desejado, clica no botão responsável pela atualização.
4.	O sistema disponibiliza um formulário para a atualização de processadores.
5.	O <i>Usuário</i> preenche os campos conforme é solicitado e submete as alterações.
6.	O caso de uso é finalizado.
A04 - <u>O usuário deseja excluir um processador</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Processador”, mantendo-se na tela responsável pela listagem de processadores.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o processador de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente ao processador desejado, clica no botão responsável pela exclusão.
4.	O sistema sinaliza qual é o processador a ser deletado em questão.
5.	O <i>Usuário</i> pressiona o botão referente à exclusão.
6.	O caso de uso é finalizado.
Fluxos de Exceção	
E01 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido.
2.	Após o processo de validação, o sistema impede a requisição referente à criação de processador.
3.	O sistema retorna ao fluxo principal (P5) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando.
E02 – <u>Os dados inseridos são inválidos</u>	
1.	O <i>Usuário</i> entra com uma entrada de dados que não passa pelas verificações que o sistema faz.
2.	Após o processo de validação, o sistema responde a requisição apontando os erros relativos à entrada de dados.
3.	O sistema retorna ao fluxo principal (P5) para uma nova entrada de dados que sejam válidos.

3.3.4 Manter Computador

Nome UC	<i>Manter Computador</i>
Atores	<i>Usuário</i>
Participação do ator	Permite que um certo usuário controle os computadores no sistema associados a ele, podendo criá-las, alterá-las, deletá-las e fazer uma leitura de todos os computadores.
Pré-condições	
O <i>Usuário</i> deve ter conexão com a Internet para acessar a página Web do Sistema Prometheus com a finalidade de controlar todos os computadores do sistema. Uma vez na página “/manter-computador” do sistema e devidamente autenticado, o <i>Usuário</i> pode prosseguir com suas operações.	
Pós-condições	
O <i>Usuário</i> modificou o(s) computador(es) associado(s) a ele no Sistema Prometheus como desejou.	
Fluxo Principal	
1.	O <i>Usuário</i> seleciona no cabeçalho a opção “Controlar Computador”.
2.	O sistema carrega uma tela contendo uma tabela com todos os computadores e três botões, um para listar todos os computadores, um para criar um novo computador e, por último, um para associar algum dos computadores do sistema para si mesmo.
3.	O <i>Usuário</i> clica no botão “Criar Computador”.
4.	O sistema apresenta um formulário destinado à criação de um novo computador.
5.	O <i>Usuário</i> preenche todos os campos necessários.
6.	O <i>Usuário</i> submete o formulário.
7.	O sistema exibe uma mensagem referente à criação do computador.
8.	O caso de uso é finalizado.
Fluxos Alternativos	
<u>A01 - O usuário cancela o cadastro de pergunta</u>	
1.	O <i>Usuário</i> fecha o navegador ou sai da página que contém ou a tabela listando todos os computadores associados à um usuário, ou o formulário de criação de computadores ou a tabela listando todos os computadores do sistema.
2.	O sistema cancela a operação.
3.	O caso de uso é finalizado.
<u>A02 - O usuário deseja ver a listagem dos computadores associados à si</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Computador” ou “Associar Computador”, mantendo-se na tela responsável pela listagem de computadores associados à si mesmo.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o computador de seu interesse e lê seus dados.

3.	O caso de uso é finalizado.
A03 - <u>O usuário deseja atualizar um computador</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Computador” ou “Associar Computador”, mantendo-se na tela responsável pela listagem de computadores.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o computador de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente ao computador desejado, clica no botão responsável pela atualização.
4.	O sistema disponibiliza um formulário para a atualização de computadores.
5.	O <i>Usuário</i> preenche os campos conforme é solicitado e submete as alterações.
6.	O caso de uso é finalizado.
A04 - <u>O usuário deseja excluir um computador</u>	
1.	O <i>Usuário</i> não clica no botão “Criar Computador”, mantendo-se na tela responsável pela listagem de computadores.
2.	O <i>Usuário</i> , ao navegar pela listagem, acha o computador de seu interesse e lê seus dados.
3.	O <i>Usuário</i> , no registro referente ao computador desejado, clica no botão responsável pela exclusão.
4.	O sistema sinaliza qual é o computador a ser deletado em questão.
5.	O <i>Usuário</i> pressiona o botão referente à exclusão.
6.	O caso de uso é finalizado.
A05 - <u>O usuário deseja associar um computador à si mesmo</u>	
1.	O <i>Usuário</i> clica no botão “Associar Computador”, mantendo-se na tela responsável pela listagem de computadores.
2.	O sistema apresenta uma tabela contendo todos os computadores do sistema, independente de associação com usuário.
3.	O <i>Usuário</i> , após ler os registros, escolhe um computador de sua preferência e clica no botão responsável por associar um certo computador à si mesmo.
4.	O sistema exibe uma mensagem referente à associação do computador.
5.	O caso de uso é finalizado.
Fluxos de Exceção	
E01 – <u>Campo requerido não fornecido</u>	
1.	O <i>Usuário</i> não entra com algum campo requerido.

2.	Após o processo de validação, o sistema impede a requisição referente à criação de computador.
3.	O sistema retorna ao fluxo principal (P5) para entrada do(s) dado(s) do <i>Usuário</i> que está faltando.
E02 – <u>Os dados inseridos são inválidos</u>	
1.	O <i>Usuário</i> entra com uma entrada de dados que não passa pelas verificações que o sistema faz.
2.	Após o processo de validação, o sistema responde a requisição apontando os erros relativos à entrada de dados.
3.	O sistema retorna ao fluxo principal (P5) para uma nova entrada de dados que sejam válidos.

CONCLUSÃO

O projeto foi desenvolvido, tanto na parte documental quanto na parte computacional, no contexto da arquitetura e montagem de computadores e a experiência que as pessoas tem ao tentar adquirir conhecimento sobre essa área da tecnologia, principalmente estudantes de informática. É mencionado em mais de um artigo científico o alto custo e risco de se montar um computador, o que incentiva que essa montagem seja feita em um ambiente simulado e digital, e o projeto desenvolvido foi na direção desse posicionamento.

É perceptível que os objetivos específicos foram alcançados com sucesso. O usuário, ao entrar no sistema, pode efetuar seu cadastrado, e, após isso, pode efetuar seu login. Estando autenticado no sistema, pode controlar processadores e computadores. Ao controlar tanto processadores quanto computadores, o sistema possibilita que o usuário faça operações de banco de dados já bem conhecidas, como inserir um registro, alterar esse mesmo registro, seja de forma completa ou incompleta, deletar esse registro e ler todos os registros presentes no banco de dados.

A relevância do projeto reside no fato dele existir enquanto aplicação web, ou seja, um sistema que roda em um navegador e oferece funcionalidades ao usuário por meio de um ambiente digital, ao invés de analógico. O mercado e a sociedade estão passando por mudanças e soluções que não consideram o ambiente virtual estão possuindo menos relevância do que aquelas que consideram.

A construção do projeto não apresentou dificuldades significativas, mas sim, oportunidades significativas. Por meio do projeto, pode-se exercitar o uso do framework Laravel, que utiliza a linguagem PHP e possui muitos recursos para lidar com API's RESTful, por meio das "Controllers", e com tabelas em bancos de dados, por meio do recurso de "ORM". Exercita-se também o uso básico de criptografia de senha e a aplicação de uma arquitetura MVC.

O sistema, apesar de já apresentar um protótipo funcional de um sistema para montagem de computadores, está aberto para melhorias, o que se traduziriam em implementação de mais funcionalidades por meio de inserção de códigos ou refatoração dos códigos que já estão no sistema visando uma melhor manutenibilidade do código que já está escrito ou aprimoramento e expansão das funcionalidades já existentes. Tanto o Angular quanto o Laravel suportam

autenticação via JWT, o que não é utilizado no presente sistema. A melhoria que poderia ser implementada poderia ser uma autenticação mais rigorosa para evitar acessos indesejados via JWT.

BIBLIOGRAFIA

COSTA, R.; GRASEL, P. Ensino de Montagem de Computadores utilizando o PC Building Simulator: um Relato de Experiência. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, Rio Grande do Sul, v. 8, p. 544-552, 11 de Novembro de 2019. Disponível em: < <http://ojs.sector3.com.br/index.php/wcbie/article/view/8996>>. Acesso em: 25 out de 2022.

DUENHA, L.; AZEVEDO R. Utilização dos simuladores do MPSoCBench para o ensino e aprendizagem de Arquitetura de Computadores **International Journal of Computer Architecture Education (IJCAE)**. Porto Alegre, v.5, n.1, p. 26-31. dez de 2016. Disponível em: < http://www2.sbc.org.br/ceacpad/ijcae/v5_n1_dec_2016/IJCAE_v5_n1_dez_2016_paper_5_vf.pdf>. Acesso em: 25 out de 2022.

VIEIRA, V.; SANTOS, A. R.; LIMA, D.; GADELHA. B. Experiências na adoção de Laboratórios Virtuais para o ensino de Montagem e Manutenção de Computadores. **SBC Open Library**. Porto Alegre, v. 15, n. 1, p. 11-20, out. 2018. Disponível em: < <http://ojs.sector3.com.br/index.php/wie/article/view/7867>>. Acesso em: 27 out de 2022.