



Instituto Federal Catarinense  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**  
*Campus Blumenau*

**YURI PATRICK FURQUIM DOS SANTOS**

**SISTEMA DE GERENCIAMENTO DE EQUIPAMENTOS DE  
TECNOLOGIA DA INFORMAÇÃO: IMPLEMENTAÇÃO MONOLÍTICA E  
DISPONIBILIZAÇÃO VIA MICROSERVIÇOS**

Blumenau

2021

**YURI PATRICK FURQUIM DOS SANTOS**

**SISTEMA DE GERENCIAMENTO DE EQUIPAMENTOS DE TECNOLOGIA DA  
INFORMAÇÃO: IMPLEMENTAÇÃO MONOLÍTICA E DISPONIBILIZAÇÃO VIA  
MICROSSERVIÇOS**

Artigo apresentado como requisito parcial à conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, *campus* Blumenau, Instituto Federal Catarinense.

Orientador: Prof.<sup>a</sup>. Hylson Vescovi Netto, Dr.

Blumenau

2021

**YURI PATRICK FURQUIM DOS SANTOS**

**SISTEMA DE GERENCIAMENTO DE EQUIPAMENTOS DE TECNOLOGIA DA  
INFORMAÇÃO: IMPLEMENTAÇÃO MONOLÍTICA E DISPONIBILIZAÇÃO VIA  
MICROSSERVIÇOS**

Este artigo foi julgado adequado para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas e aprovado em sua forma final pelo curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Catarinense – *Campus Blumenau*.

Orientador: Prof.<sup>(a)</sup>. Hylson Vescovi Netto, Dr.

---

Prof.<sup>(a)</sup> Hylson Vescovi Netto, Dr.

Orientador – IFC *campus* Blumenau

**BANCA EXAMINADORA**

---

Prof.<sup>(a)</sup> Ricardo de la Rocha Ladeira, Msc.

Instituto Federal Catarinense, *campus* Blumenau

Blumenau

2021

## **SISTEMA DE GERENCIAMENTO DE EQUIPAMENTOS DE TECNOLOGIA DA INFORMAÇÃO: IMPLEMENTAÇÃO MONOLÍTICA E DISPONIBILIZAÇÃO VIA MICROSERVIÇOS**

*Yuri Patrick Furquim dos Santos*

### **RESUMO**

O presente trabalho contém a documentação de um Sistema de Gerenciamento de Equipamentos de TI (Tecnologia da Informação). O texto descreve o sistema em termos de suas funcionalidades e ferramentas utilizadas para o desenvolvimento, projeto e implementação. Os principais recursos do sistema são o cadastro e a consulta de equipamentos e um relatório dos equipamentos em estoque. O sistema de gerenciamento de equipamento de TI é uma alternativa para gerenciar os equipamentos de forma superior ao método tradicional de gerenciar em planilhas de excel. O sistema foi desenvolvido como um software monolítico e complementado com a disponibilização de informações via micros serviços.

**Palavras-chave:** java; sistema de gerenciamento; microserviços.

### **ABSTRACT**

This work contains documentation from an IT Equipment Management System (Information Technology). The text describes the system in terms of its functions and tools used for development, design, and implementation. The main features of the system are the registration and querying of equipments and a report on equipments in stock. The IT equipment management system is an alternative to managing equipment in a superior way to the traditional method of management using excel sheets. The system was developed as a monolithic software and complemented with the provision of information via micro services.

**Keywords:** java; management system; microservices.

## 1 INTRODUÇÃO

A tecnologia da informação (TI) é um recurso computacional que visa permitir a obtenção, o armazenamento, o acesso, o gerenciamento e o uso das informações. O hardware é a parte física de um computador, como monitor, teclado e etc. O software de um sistema consiste de um ou mais programas que comandam o funcionamento de um computador ou outro dispositivo eletrônico. Hardware e software são componentes da infraestrutura de TI, que:

“[...] oferece a plataforma para suporte de todos os sistemas de informação na empresa como hardware, software, tecnologia de gestão de dados, Organiza, gerencia e processa dados de negócios relacionados a estoque, clientes e fornecedores, tecnologia de rede e telecomunicações e serviços de tecnologia.” (COSTA JÚNIOR, 2014).

A tecnologia da informação está dividida em várias áreas, distribuídas em setores como hardware, software, banco de dados e infraestrutura. O setor de infraestrutura de uma organização geralmente controla o hardware e o software responsáveis pela realização das tarefas computacionais que subsidiam a tarefa principal da organização. Especificamente na área de hardware, “O técnico de infraestrutura e TI trabalha na manutenção de hardwares, softwares e redes, a fim de prevenir e consertar falhas técnicas e agilizar os processos de trabalho” (ALVES, 2015).

Os setores de TI lidam frequentemente com problemas, dentre eles a desorganização no gerenciamento dos equipamentos em estoque. Diversos setores de TI controlam os equipamentos e estoque em planilhas eletrônicas, o que pode causar desorganização. Para auxiliar no problema da organização foi desenvolvido um sistema para gerenciar todos os equipamentos de TI, o que pode aumentar a produtividade dos usuários mediante a ocorrência de menos conflitos no gerenciamento. Nesse sistema serão controlados também produtos utilizados no contexto da TI, como mouse, teclado, monitores, entre outros. A organização de recursos se torna importante, pois ajuda a estruturação da infraestrutura como um todo.

Outro ponto identificado como positivo no sentido de aprimorar o controle de equipamentos de TI em uma organização é a possibilidade de disponibilizar as informações para outros sistemas, de maneira que seja possível visualizar informações do cadastro de equipamentos em sistemas relacionados que necessitem dessas informações. Como exemplo, pode-se considerar que o planejamento do desenvolvimento de novos sistemas pode levar em

conta a disponibilidade de hardware atual na organização. Por isso, o presente trabalho estende a implementação de um simples cadastro de equipamentos, tornando as informações disponíveis por meio de microsserviços, criando assim uma fonte de informações de acesso limitado e específico, a fim de prover dados relevantes para outros sistemas.

### 1.1 TEMA/PROBLEMA

Diversos setores de TI controlam os equipamentos e estoque utilizando planilhas eletrônicas. Para facilitar esse controle, será desenvolvido um sistema que controla e gerencia os equipamentos. Além disso, informações do sistema serão disponibilizadas a outros sistemas por meio de microsserviços.

### 1.2 OBJETIVOS PROPOSTOS/SOLUÇÃO DOS PROBLEMAS

O objetivo deste trabalho é desenvolver um sistema que gerencia os equipamentos de TI, controlando os equipamentos e os produtos em estoque, por meio de suas entradas e saídas. O sistema também deve ser capaz de fornecer informações a outros sistemas por meio de microsserviços e gerar relatórios das consultas de equipamentos e produtos.

### 1.3 ESCOPO

Este trabalho abrange o controle de equipamentos de TI e pode ser aplicado a uma instituição que possua esses equipamentos. As funcionalidades do sistema são basicamente o cadastro de equipamentos e relatórios de estoque. Não são contemplados neste sistema cópia automática dos dados, nem gerenciamento de múltiplos usuários, sendo apenas um usuário principal considerado na operação do sistema.

### 1.4 VIABILIDADE DO PROJETO

O sistema pode ser executado em navegadores web, disponíveis em computadores pessoais, *tablets* e *smartphones*. Essa arquitetura não requer nenhuma configuração ou instalação, por parte do usuário, para acesso ao sistema. A implementação foi realizada com ferramentas livres, e portanto, acessíveis.

## 1.5 MÉTODO DE TRABALHO (ARQUITETURA, FERRAMENTAS, TECNOLOGIAS APLICADAS)

O desenvolvimento do projeto foi seguindo no modelo espiral, pois no processo do desenvolvimento do sistema foram revisadas todas as fases do software: definição do objetivo, avaliação, implementação e validação. Essas fases foram repetidas até a conclusão do sistema. As tabelas foram geradas a partir das classes, por meio de ferramenta de persistência de dados automatizada.

## 1.6 FERRAMENTAS

Para o desenvolvimento do sistema foram utilizadas as ferramentas Eclipse, Java, HeidiSQL, MariaDB, Apache-Tomcat, Apache POI, JDBC, JSP e Bootstrap. O Eclipse é um ambiente de desenvolvimento integrado (*Integrated Development Environment – IDE*), sendo uma ferramenta gratuita e de código aberto para os desenvolvedores de software. O Eclipse é executado em várias plataformas, como Windows, Linux e MacOS e permite criar aplicativos profissionais de desktop, empresariais, Web e móveis multiplataformas. A execução em multiplataforma é uma vantagem porque existem muitas máquinas rodando Windows e Linux. O Eclipse suporta a criação de projetos e módulos, e possui uma documentação vasta inclusive em português bem organizada. Tais recursos auxiliam o desenvolvimento rápido de aplicativos. Segundo (ARAÚJO, 2021) O Eclipse é uma IDE conhecido mais comumente para desenvolvimento em Java, no entanto, por meio de plug-ins, ele pode ser usado para desenvolver aplicações em várias linguagens, como C/C++, Python, PHP e inclusive para a plataforma Android.

No presente trabalho foi utilizada a linguagem de programação Java, que é uma linguagem portátil, pois pode ser executada em várias plataformas. O código de um programa em Java é compilado para um formato bytecode que é interpretado pela máquina virtual Java:

“Java é uma linguagem de programação orientada a objeto desenvolvida pela Sun Microsystems na década de 90, que atualmente, pertence a Oracle. Programas em Java são compilados para um bytecode, o que permite aos desenvolvedores criarem um programa uma única vez e depois executá-lo em qualquer uma das plataformas.” (BRITO, 2014).

Java ganhou espaço em diferentes dispositivos (BRITO, 2014). Existem muitas empresas de software que disponibilizam programas que são feitos nessa linguagem, tornando a instalação da máquina virtual Java uma realidade na maioria dos sistemas operacionais atuais. Como exemplo de programas que utilizam Java, podem ser citados “O acesso a serviços bancários online e os aplicativos da Receita Federal brasileira, como o programa de declaração de imposto (BRITO, 2014).

A visualização das informações armazenadas pelo sistema é realizada com uso da ferramenta HeidiSQL, que é um software livre. O objetivo do HeidiSQL é facilitar a conexão e a manipulação de dados dos sistemas de gerenciamento de banco de dados (BECKER, 2020). No HeidiSQL é possível ver e editar dados e suas estruturas, sendo suportados diversos tipos de banco de dados, como o MariaDB, MySQL, Microsoft SQL, PostgreSQL e SQLite.

No caso do presente trabalho, foi utilizado o MariaDB, um banco de dados de código aberto que foi criado pelos mesmos desenvolvedores do MySQL. As principais características do MariaDB são a sua rapidez, escalabilidade e robustez, sendo apoiado também por ferramentas, plugins e uma capacidade de armazenamento adequada para a maioria das aplicações (SOUZA, 2020). A conexão do banco de dados com o programa de gerenciamento de equipamentos é feita por meio da conexão JDBC (*Java database connectivity*), que consiste de um conjunto de interfaces e classes escritas na linguagem de programação Java (IBM, 2019).

O sistema de controle de equipamentos é executado no servidor Apache Tomcat, que é um container Web de código fonte aberto baseado em Java. O Tomcat foi criado para executar aplicações Web que utilizam tecnologias Servlets e JSPs (MEDEIROS, 2013). JSP (*Java server pages*) é uma linguagem de script com especificação aberta que tem como objetivo a geração de conteúdo dinâmico para páginas da Internet. Podemos utilizar o JSP para prover informações dinâmicas às páginas HTML (ROCHA, 2013).

O Apache POI é um framework para a plataforma Java que possibilita a leitura e a escrita de dados em um documento do Microsoft Office. Com ele, é possível ler e escrever dados em arquivos Excel, Word, PowerPoint e arquivos de e-mail do Outlook (SANTANA, 2014). O framework foi utilizado no desenvolvimento do sistema para gerar os relatórios das consultas dos equipamentos e produtos em excel.

Para aprimorar a visualização das páginas web, foi utilizada a biblioteca bootstrap, uma estrutura de front-end usada para criar sites e aplicativos da web modernos. Essa



biblioteca é de código aberto e uso gratuito, e apresenta vários modelos HTML e CSS para elementos de interface de usuário, como botões e formulários. O bootstrap também oferece suporte a extensões JavaScript (BACINGER, 2015).

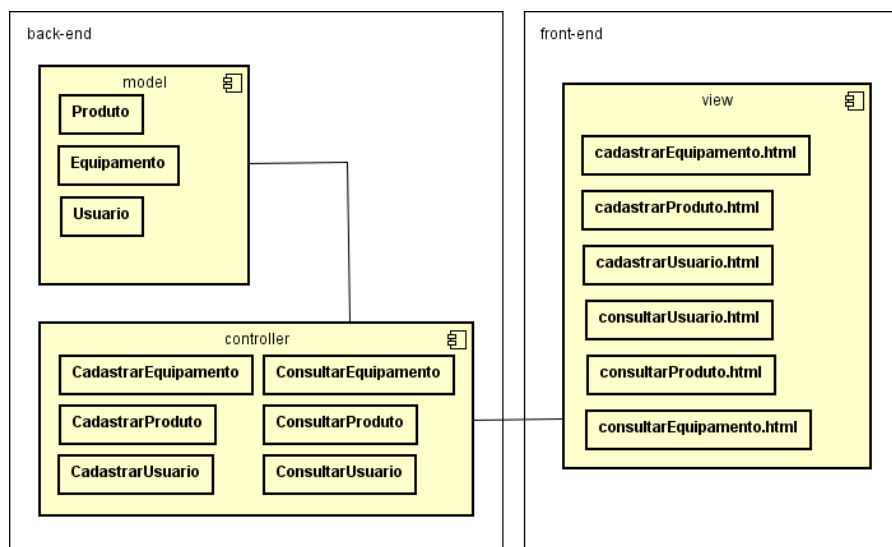
## 1.7 ARQUITETURA

A aplicação monolítica descrita neste TCC foi desenvolvida para a Web e possui a arquitetura do padrão MVC (Model View Controller):

O padrão arquitetural Model-View-Controller (MVC) é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador (MEDEIROS, 2020).

Os elementos considerados neste trabalho são o produto, o equipamento e o usuário, e se encontram definidos como modelos, integrando uma série de outros componentes (Figura 1). Para cada elemento, existem dois controladores: um responsável por efetuar os cadastros e outro por efetuar consultas. Por fim, páginas HTML exibem as informações de cada um dos itens de modelo, bem como recebem dados de novas instâncias.

Figura 1. Diagrama de componente – Arquitetura MVC



Fonte: elaborado pelo autor.

## 2 TRABALHOS CORRELATOS EXISTENTES

Existem alguns sistemas similares ao sistema apresentado neste documento. O Milvus (MILVUS, 2020) é um sistema de gerenciamento de TI inteligente, que possibilita otimizar a gestão das ações da equipe de TI, resultando em ganhos de eficiência e aumento de produtividade da equipe com a mesma estrutura. Em um único sistema funciona a gestão de ativos dos dispositivos de TI, sistema de helpdesk inteligente e de chamados, diretamente da área de trabalho ou via aplicativo mobile. A gestão de tickets é feita através de painéis personalizáveis (*dashboards*) que monitoram a situação em tempo real.

Outro sistema similar ao que está sendo descrito neste documento é o Artis, uma solução completa para rastreamento e gestão de ativos de TI (ARTIS, 2020). O Artis reduz os custos e otimiza a gestão do ambiente de *datacenter* com uma tecnologia eficiente, confiável e fácil de usar. As principais operações que o sistema disponibiliza são: rastreabilidade em tempo real, relatórios gerenciais, gestão de energia, multiplataforma, alarme, notificações e integração com sistemas legados.

## 3 REQUISITOS

Os requisitos de sistema são definidos pelo usuário, contém as funcionalidades requisitadas e suas particularidades. Os requisitos são separados em duas partes: requisitos funcionais e não funcionais.

Um requisito de sistema de software especifica uma função que o sistema ou componente deve ser capaz de realizar. Estes requisitos de software definem o comportamento do sistema, ou seja, o processo ou transformação que componentes de software ou hardware efetuam sobre as entradas para gerar as saídas (SILVA FILHO, 2008).

Requisitos não funcionais contemplam as características e aspectos internos do sistema, envolvendo especificamente a parte técnica (CELESTINO 2020). Ao contrário dos requisitos funcionais, esses requisitos geralmente não são explicitamente expostos pelo cliente, mas devem ser implicitamente compreendidos pelo desenvolvedor.

### 3.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

A seguir estão listados os requisitos funcionais (Tabela 1) e os não funcionais (Tabela 2) pertinentes ao sistema de gerenciamento de TI.

Tabela 1 - Requisitos Funcionais

Número	Requisitos Funcionais	Caso de Uso
RF01	O sistema terá cadastro de usuários administrador	UC01
RF02	O sistema terá login	UC02
RF03	O sistema terá cadastro de equipamento	UC03
RF04	O sistema terá cadastro de produto	UC04
RF05	O sistema terá consulta dos equipamentos	UC05
RF06	O sistema emitirá um relatório da consulta dos equipamentos	UC05
RF07	O sistema terá consulta dos produtos em estoque	UC06
RF08	O sistema emitirá um relatório dos produtos em estoque	UC06
RF09	O sistema terá a consulta de perfil do usuário logado	UC07
RF10	O sistema disponibilizará a opção de alterar produto em estoque e remover	UC06
RF11	O sistema disponibilizará a opção de alterar equipamento em estoque e remover	UC05

Tabela 2. Requisitos Não Funcionais

Número	Requisitos Não Funcionais
RNF01	O sistema deve utilizar banco de dados MYSQL
RNF02	O sistema rodar nos navegadores Google Chrome e Mozilla Firefox
RNF03	O sistema rodará nos sistemas operacionais Windows e Linux.
RNF04	O sistema será desenvolvido na linguagem JAVA WEB
RNF05	O acesso do sistema só é disponível para os administradores

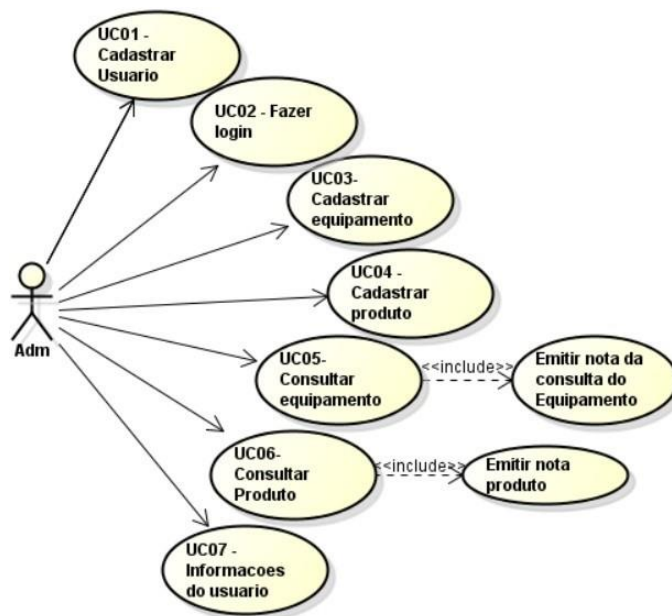
## 4 DIAGRAMAS UML

A linguagem UML possui diversos diagramas úteis para modelar os sistemas de informação sendo útil para “especificar, visualizar e documentar modelos de software orientados por objetos” (SILVA, 2017). Nesta seção serão apresentados os casos de uso, diagrama de sequência e diagrama de atividade referentes ao sistema de controle de equipamentos de TI.

### 4.1 CASOS DE USO

Os casos de uso apresentam as principais funcionalidades do sistema e a interação do usuário no sistema. No sistema de controle de equipamentos de TI, o administrador possui como principais funções cadastrar usuário, equipamento e produto (Figura 2). Existem também consultas ao equipamento e ao produto, que fazem uso de ações de emissão de notas. Por fim, o administrador pode visualizar informações sobre o usuário.

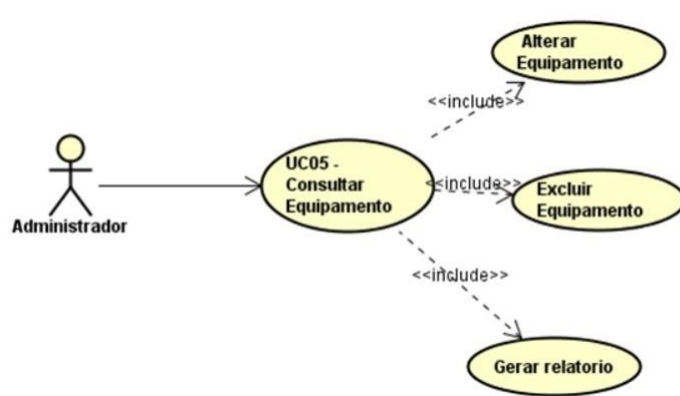
Figura 2. Caso de uso – administrador



Fonte: elaborado pelo autor.

O administrador pode realizar consultas ao cadastro de equipamentos, além de efetuar atualizações em suas informações, excluir equipamentos e gerar relatório dos equipamentos cadastrados (Figura 3).

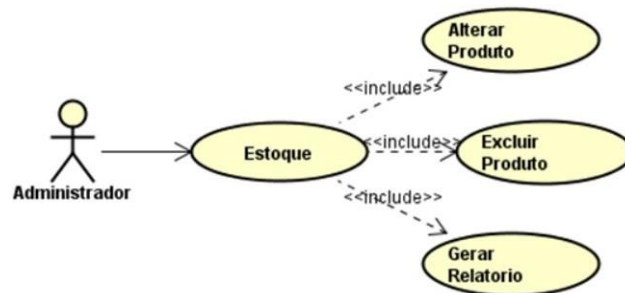
Figura 3. Caso de uso – Cadastro de equipamento



Fonte: elaborado pelo autor.

Ao consultar um produto em estoque, o administrador pode realizar as operações de alterar, excluir e gerar relatório dos produtos em estoque (Figura 4).

Figura 4. Caso de uso – Estoque



Fonte: elaborado pelo autor.

A seguir estão descritos em detalhes os casos de uso definidos para o sistema: cadastro de usuário (Tabela 3), login (Tabela 4), cadastro de equipamento (Tabela 5), cadastro de produto (Tabela 6), consulta dos equipamentos (Tabela 7), consulta dos produtos em estoque (Tabela 8) e consulta de perfil do usuário logado (Tabela 9).

Tabela 3. Cadastrar Usuário (UC01)

Objetivo:	Administrador cadastra usuário
Requisito:	RF01
Atores:	Administrador
Condição de entrada:	Cadastro Usuário
Fluxo Principal:	01 – O administrador cadastra o usuário no sistema. - Nome - Sobrenome - Telefone - E-mail - Data nascimento - Login - Senha

Fonte: elaborado pelo autor.

Tabela 4. Efetuar Login (UC02)

Objetivo:	Administrador cadastra usuário
Requisito:	RF02
Atores:	Administrador
Condição de entrada:	Cadastro Usuário
Fluxo Principal:	- O administrador tem um usuário e senha. - O administrador inicia o login com o seu usuário. - O administrador inicia o login com a sua senha
Fluxo Alternativo:	01 – Usuário não exista: - Usuário não possui conta no sistema, informar conta não existe. 02 – Usuário errado: - Se o usuário digitado errado, informar usuário incorreto. 03 – Senha errada: - Se senha digitada estiver errada, informar senha incorreta.

Fonte: elaborado pelo autor.

Tabela 5. Cadastrar Equipamento (UC03)

Objetivo:	Administrador cadastra os equipamentos
Requisito:	RF03

Atores:	Administrador
Condição de entrada:	Cadastro Equipamento
Fluxo Principal:	01 – O administrador cadastra todos os equipamentos - Tipo do equipamento (Campo obrigatório editável) - Sistema operacional (Campo obrigatório editável) - Arquitetura (Campo obrigatório editável) - Processador (Campo obrigatório editável) - Memória RAM (Campo obrigatório editável) - Filial (Editável) - Setor (Editável) - ID (Editável) - Nome do computador (Editável) - Usuário responsável (Editável)

Fonte: elaborado pelo autor.

Tabela 6. Cadastro Produto (UC04)

Objetivo:	Administrador cadastra as informações dos equipamentos
Requisito:	RF04
Atores:	Administrador
Condição de entrada:	Cadastro Produto
Fluxo Principal:	01 – O administrador cadastra todos os produtos. - Descrição do produto (Campo obrigatório editável) - Referência (Campo obrigatório editável) - Data Compra (Campo obrigatório editável) - Modelo/Tipo (Campo obrigatório editável) - Fornecedor (Campo obrigatório editável) - Quantidade (Campo obrigatório editável)

Fonte: elaborado pelo autor.

Tabela 7. Consultar Equipamento (UC05)

Objetivo:	Administrador realiza consulta dos equipamentos
Requisito:	RF05
Atores:	Administrador
Condição de entrada:	Consulta Equipamento
Fluxo Principal:	01 – O administrador realiza a localização dos equipamentos a partir

	do nome do computador. - Nome do computador - Filial - Setor - Tipo do equipamento - Sistema operacional - Arquitetura - Processador - RAM - Domínio - Nome do computador - Usuário responsável - Marca
Fluxo Alternativo:	01 - O administrador pode alterar os campos se algum cadastro for incoerente. - Alterar campo cadastrado 02 - O administrador pode alterar algum campo que não conferente. - Remover campo cadastrado

Fonte: elaborado pelo autor.

Tabela 8. Consultar Produto (UC06)

Objetivo:	Consultar produto em estoque
Requisito:	RF06
Atores:	Administrador
Condição de entrada:	Consultar Estoque
Fluxo Principal:	01 – O administrador consulta os produtos em estoque - Descrição do produto - Referência - Data Compra - Modelo/Tipo - Fornecedor - Quantidade
Fluxo Alternativo:	01 - O administrador pode alterar os campos - Alterar campo cadastrado 02 - O administrador pode remover algum campo que não conferente. - Remove produto cadastrado.

Fonte: elaborado pelo autor.



Tabela 9. Consulta Informação de perfil (UC07)

Objetivo:	Consultar Informações do perfil ativo no sistema.
Requisito:	RF07
Atores:	Administrador
Condição de entrada:	Informação do usuário ativo no sistema
Fluxo Principal:	01 – Administrador ativo no sistema visualizar informações do perfil: - Nome - Sobrenome - Telefone - E-mail - Data Nascimento - Usuário de login
Fluxo Alternativo:	01 - O administrador pode remover usuário cadastrado - Remover usuário cadastrado.

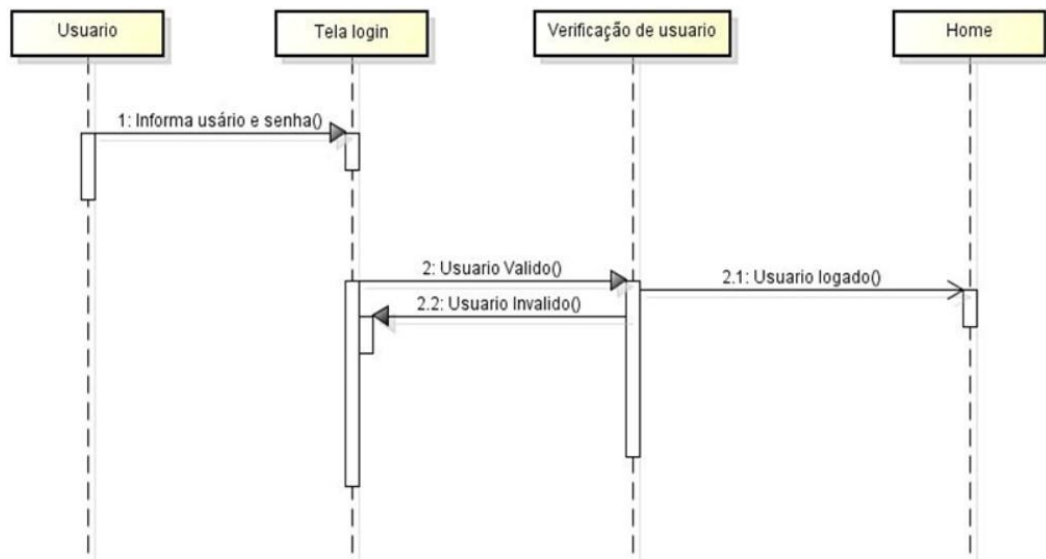
Fonte: elaborado pelo autor.

#### 4.2 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência contém fluxos de execução dos comandos executados no sistema. Primeiramente, para o usuário efetuar o login no sistema, o usuário informa login e senha (Figura 5, passo 1). Se os dados forem válidos (2), o usuário é conectado no sistema (2.1). Caso contrário (2.2), é retornada para o usuário uma mensagem de “usuário inválido”.

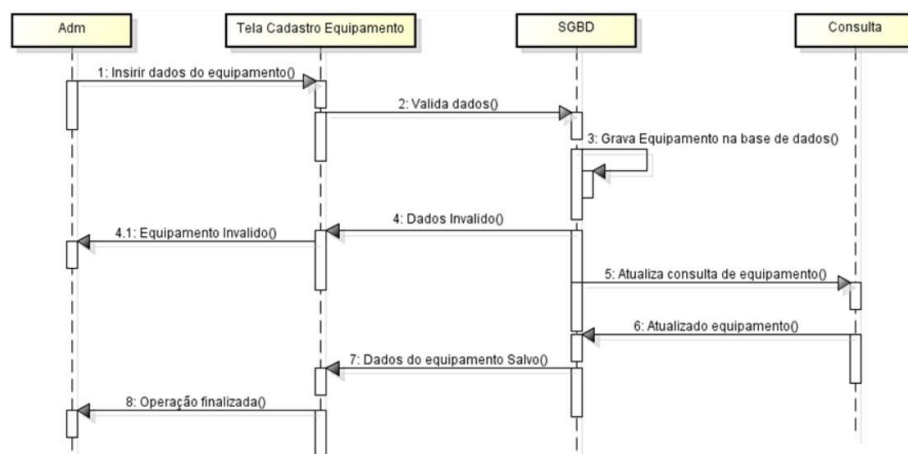
Para efetuar o cadastro de equipamento, inicialmente o usuário informa os dados do equipamento (Figura 6, passo 1). Se os dados forem válidos (2), o equipamento é gravado na base de dados (3); caso contrário (dados inválidos), retorna-se ao menu principal (4) com a mensagem de cadastro inválido (4.1). Caso o usuário precise alterar algum dado do equipamento (5), os dados atualizados são retornados os dados para a base de dados (6), e por fim os dados atualizados são enviados para a tela (7) e as operações são finalizadas (8).

Figura 5. Diagrama de Sequência – Login



Fonte: elaborado pelo autor.

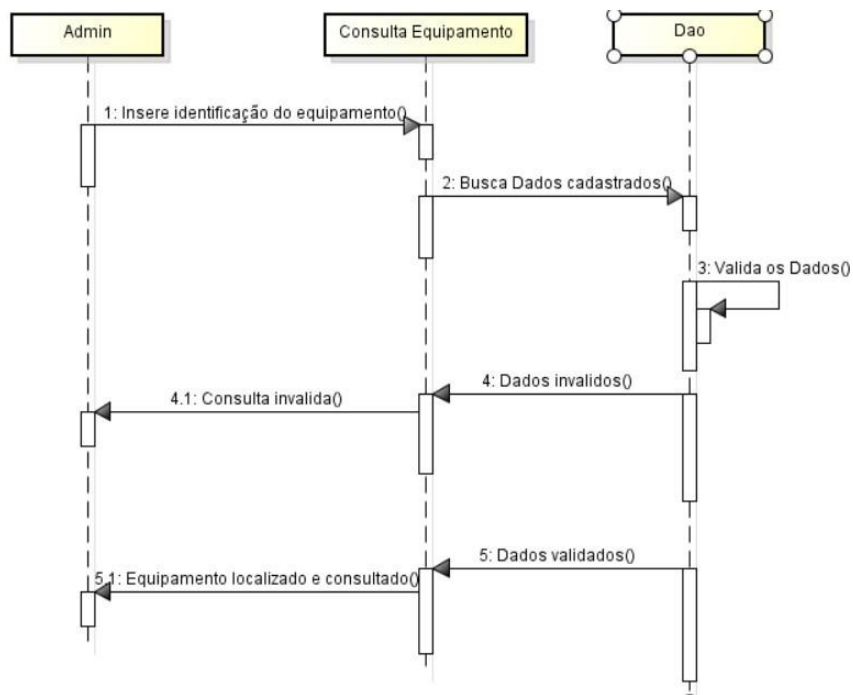
Figura 6. Diagrama de Sequência – Cadastro Equipamento



Fonte: elaborado pelo autor.

Para consultar um equipamento, o usuário informa os dados do equipamento (Figura 7, passo 1). Se os dados forem válidos (2), é retornada a consulta do equipamento na base de dados (3), caso contrário retorna-se para a tela de consulta (4) com uma mensagem de consulta inválida (4.1). Por fim, em caso de sucesso, a operação de consulta retorna os dados na tela (5), tendo sido o equipamento localizado na base de dados (5.1). As consultas de estoque e produto seguem fluxos similares a este.

Figura 7. Diagrama de Sequência – Consulta de Equipamento

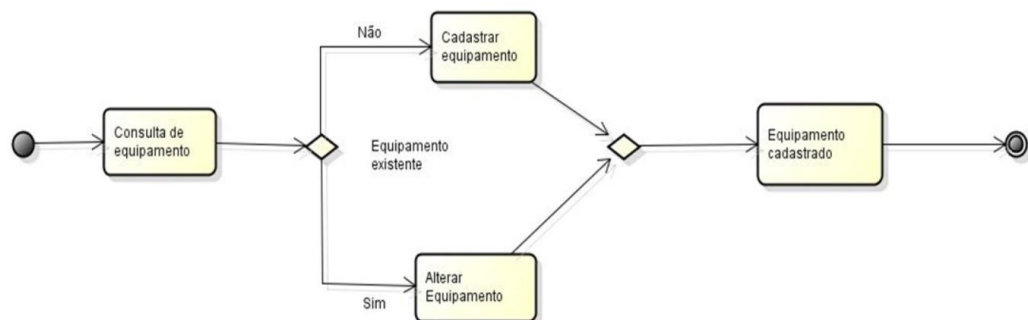


Fonte: elaborado pelo autor.

#### 4.3 DIAGRAMAS DE ATIVIDADE

Os diagramas de atividade demonstram o fluxo de atividade em um único processo. Na atividade no cadastro de equipamento (Figura 8), caso um equipamento esteja cadastrado no sistema e haja alguma informação a ser modificada, é acionada a atividade de alteração do equipamento. Caso o equipamento não esteja cadastrado, é realizado o cadastro do equipamento.

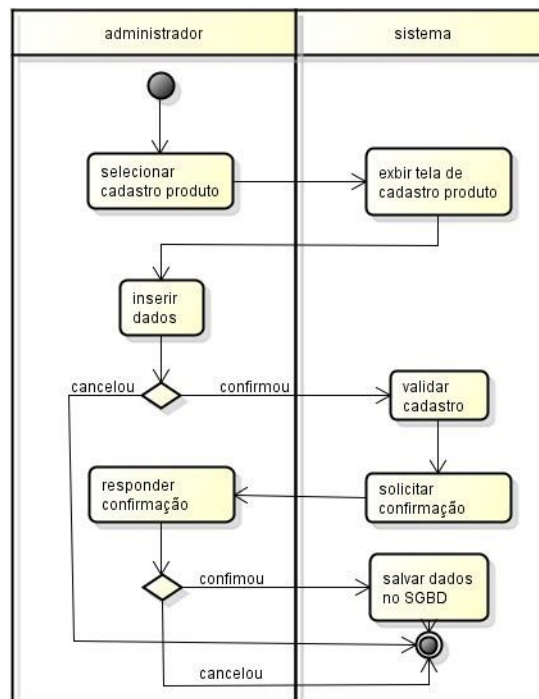
Figura 8. Diagrama de Atividade – Cadastro Equipamento



Fonte: elaborado pelo autor.

Um novo produto é cadastrado da seguinte forma (Figura 9): o administrador inicia o cadastro selecionando essa opção no menu, e a partir da tela de dados insere as informações. Ao salvar os dados, validações são feitas nas informações do cadastro, e após a confirmação do usuário os dados são armazenados no SGBD. Outra forma de finalizar essa atividade é quando o usuário cancela a operação de inserção de dados.

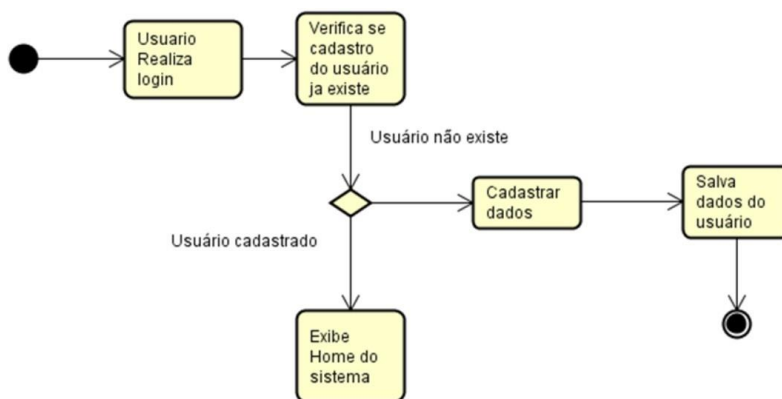
Figura 9. Diagrama de Atividade – Cadastro Produto



Fonte: elaborado pelo autor.

Na atividade de cadastro de um usuário, inicialmente pode-se realizar o login, quando então o sistema realiza a validação de login (Figura 10). Se o usuário já possuir cadastro no sistema a tela inicial é exibida; caso contrário, o acesso é negado e solicita-se que o cadastro do usuário seja iniciado. Após o cadastro, os dados são salvos, de maneira que no próximo login o fluxo realizado pelo usuário seja o fluxo no qual o usuário existe.

Figura 10. Diagrama de Atividade – Cadastro Usuário



Fonte: elaborado pelo autor.

#### 4.4 DIAGRAMA DE CLASSES

O diagrama de classes representa a estrutura para a manipulação dos objetos no desenvolvimento do sistema:

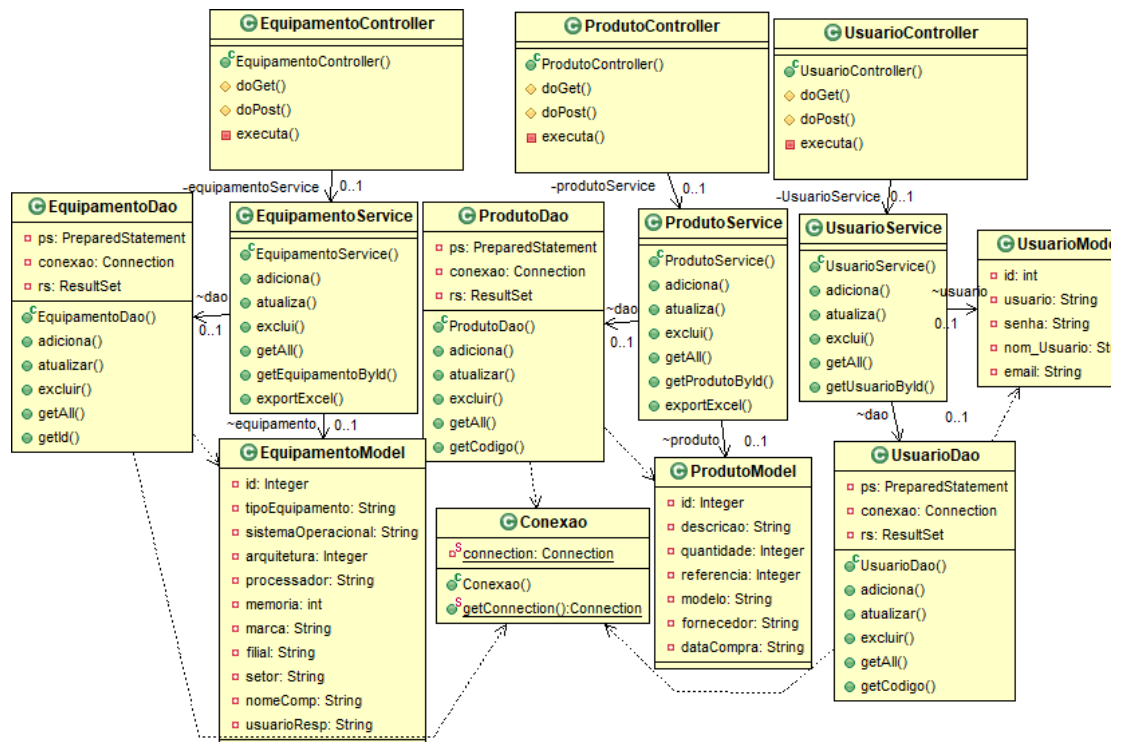
Diagramas de classes estão entre os tipos mais úteis de diagramas UML pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos. (LUCIDCHART, 2021).

A figura 11 representa as classes para o planejamento e o desenvolvimento do sistema, com os seus respectivos atributos, operações e métodos. As classes de modelo são Equipamento, Usuário e Produto, enquanto as demais classes implementam ações de um sistema CRUD que são: cadastrar, consultar, atualizar e deletar. As classes de serviço de equipamento, produto e usuário são as que implementam as operações do CRUD.

### 5 MODELAGEM DE DADOS

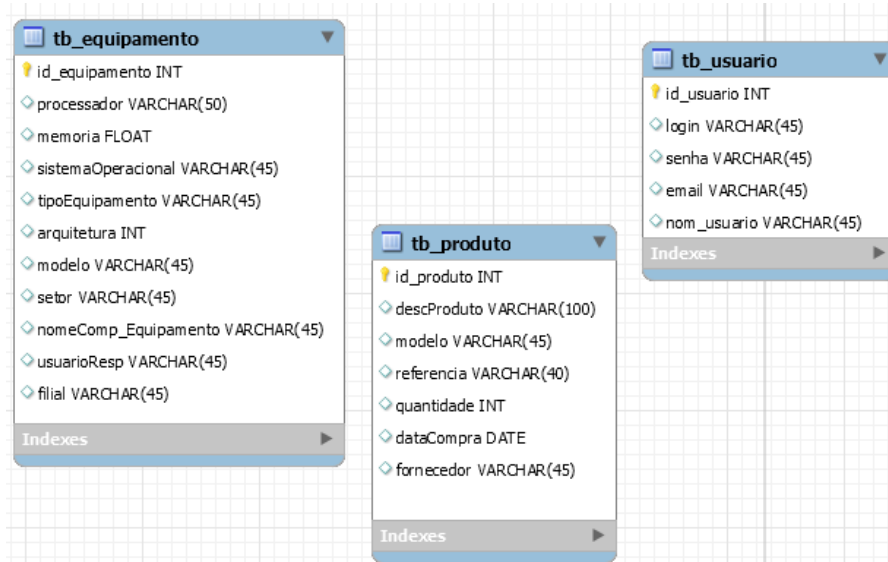
Nesta seção serão apresentados o modelo de entidade e relacionamento (MER) e o dicionário de dados. O MER representa uma forma abstrata da estrutura que o banco de dados da aplicação possui, e o dicionário de dados apresenta todos os elementos do banco de dados, tais como as entidades e seus atributos. A estrutura lógica do banco de dados abriga as tabelas de cadastro do equipamento, usuário e do produto (Figura 12).

Figura 11. Diagrama de classes Sistema Monolítico



Fonte: elaborado pelo autor.

Figura 12. Diagrama de entidade e relacionamento



Fonte: elaborado pelo autor.

O dicionário de dados consiste em uma lista organizada com todos os elementos do banco de dados, tais como as entidades e seus atributos. As entidades são conjuntos de

informações que integram as tabelas no banco de dados. Os atributos são os campos da entidade, ou seja, um atributo é um campo da tabela. As entidades login, consulta de equipamento, estoque, cadastro equipamento, cadastro produto e cadastro informação estão dispostas a seguir (Tabelas 10 a 15) os atributos e suas propriedades.

Tabela 10. Entidade tb\_usuario

Atributo	Classe	Domínio	Tamanho	Descrição
id_usuario	Determinante	Numérico		Atribuir entidade única no banco
login	Simples	Texto	45	Login do usuário
senha	Simples	Texto	45	Senha do usuário
nom_usuario	Simples	Texto	45	Nome do usuário
email	Simples	Texto	45	E-mail do usuário

Tabela 11: Entidade tb\_equipamento

Atributo	Classe	Domínio	Tamanho	Descrição
id_equipamento	Determinante	Numérico		Atribuir entidade única para cada equipamento cadastrado
tipoEquipamento	Simples	Texto	45	Desktop, notebook, servidor
sistemaOperacional	Simples	Texto	45	
arquitetura	Simples	Numérico		
Processador	Simples	Texto	45	
Memoria	Simples	Texto	45	Memória RAM do equipamento
Modelo	Simples	Texto	45	
Filial	Simples	Texto	45	Filial na qual o equipamento se localiza

Setor	Simples	Texto	45	
NomeComp	Simples	Texto	45	Hostname do equipamento
usuarioResp	Simples	Texto	45	Usuário responsável pelo equipamento

Tabela 12: Entidade tb\_produto

Atributo	Classe	Domínio	Tamanho	Descrição
id_produto	Determinante	Numérico		Atribuir entidade única no banco
descricao	Simples	Texto	100	
quantidade	Simples	Numérico		
referencia	Simples	Texto	45	
modelo	Simples	Texto	45	
fornecedor	Simples	Texto	45	
dataCompra	Simples	Data	45	

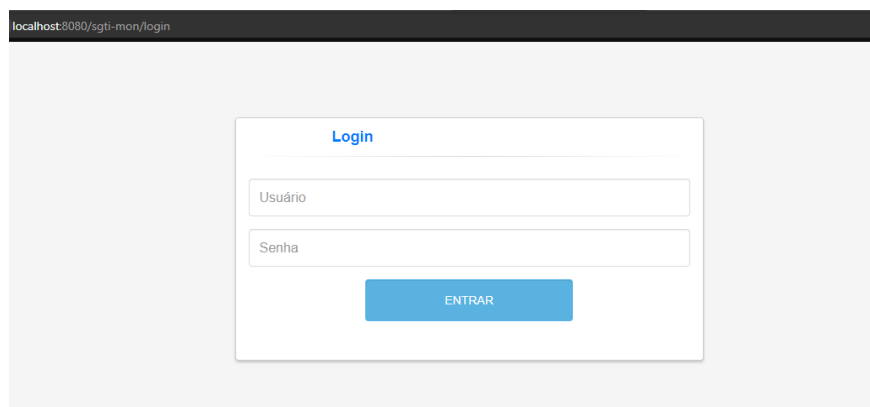
## 6 PROJETO DE INTERFACE

O projeto de interface tem o intuito de demonstrar as interfaces do sistema. As seguintes páginas estão disponíveis para navegação: login, cadastro usuário, cadastro equipamento, cadastro produto, consulta de equipamento e consulta de produto em estoque. As funções das páginas informadas são de criação, alteração, busca e listagem das informações. O sistema inicia por meio de um acesso que requer o fornecimento de um login e uma senha (Figura 13).

A Figura 14 ilustra a tela de cadastro do usuário. Para realizar as operações no sistema é necessário realizar o cadastro do usuário no sistema. Os campos nome, e-mail e login e senha são obrigatórios para o cadastro.

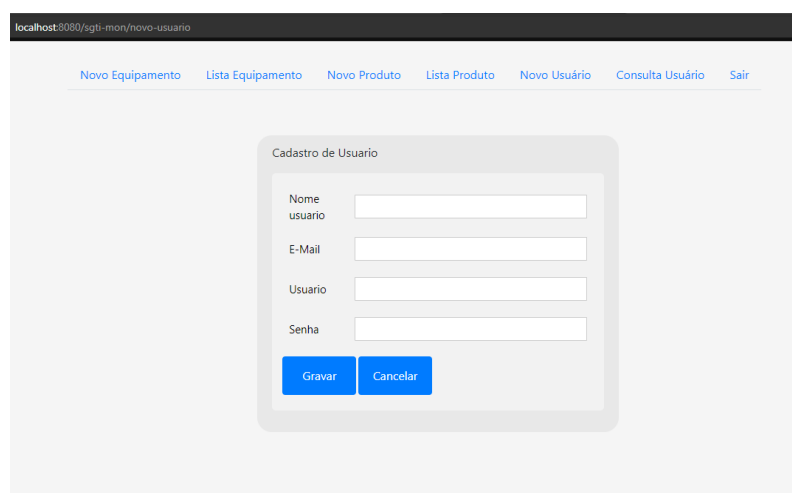


Figura 13. Login



Fonte: elaborado pelo autor

Figura 14. Cadastro de Usuário



Fonte: elaborado pelo autor

A Figura 15 ilustra a tela de cadastro das informações do equipamento, setor no qual o equipamento está localizado e o usuário que será o responsável do equipamento. As informações do equipamento são inseridas nessa página do sistema.

A Figura 16 está ilustrando a tela de cadastro do produto. Nesse cadastro é realizada a entrada dos novos produtos em estoque, sendo informados a descrição do produto, quantidade, referência, modelo, fornecedor e a data da compra do produto.

Figura 15. Cadastro de equipamento

Fonte: elaborado pelo autor

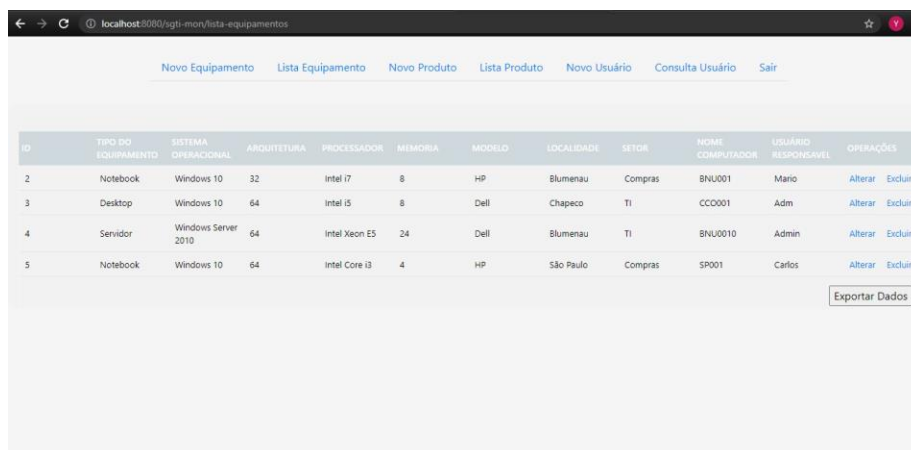
Figura 16. Cadastro de produto

Fonte: elaborado pelo autor

A Figura 17 está ilustrando a tela de consulta de informações do equipamento, além de exibir o departamento que se encontra o equipamento e o usuário responsável. É possível nessa tela realizar as ações de alteração e remoção dos equipamentos cadastrados.

Na Figura 18 encontra-se a tela de estoque com os produtos de TI cadastrados no sistema. Nessa tela é possível executar as ações de alteração do produto e também a remoção do produto do estoque.

Figura 17: Consulta de equipamentos

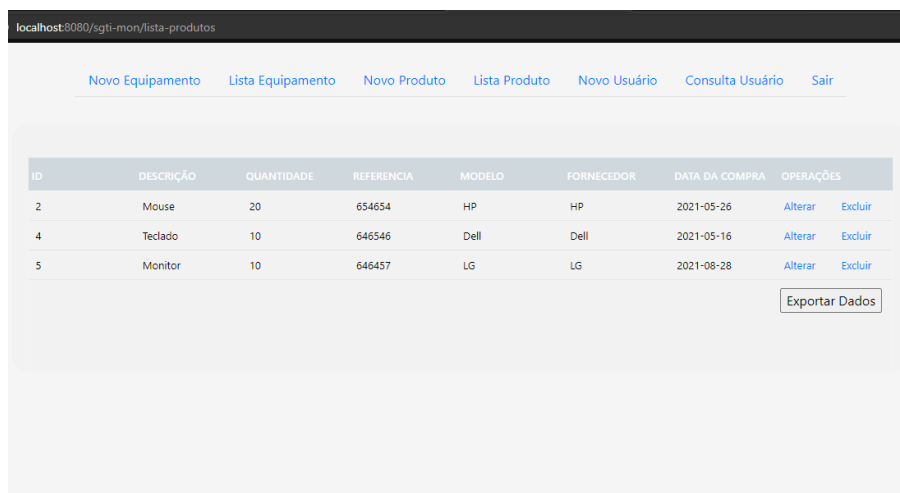


ID	TIPO DO EQUIPAMENTO	SISTEMA OPERACIONAL	ARQUITETURA	PROCESSADOR	MEMORIA	MODELO	LOCALIDADE	SETOR	NOME COMPUTADOR	USUÁRIO RESPONSÁVEL	OPERAÇÕES
2	Notebook	Windows 10	32	Intel i7	8	HP	Blumenau	Compras	BNU001	Mario	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Desktop	Windows 10	64	Intel i5	8	Dell	Chapeco	TI	CCC001	Adm	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Servidor	Windows Server 2010	64	Intel Xeon E5	24	Dell	Blumenau	TI	BNU0010	Admin	<a href="#">Alterar</a> <a href="#">Excluir</a>
5	Notebook	Windows 10	64	Intel Core i3	4	HP	São Paulo	Compras	SP001	Carlos	<a href="#">Alterar</a> <a href="#">Excluir</a>

Exportar Dados

Fonte: elaborado pelo autor

Figura 18. Consulta ao estoque



ID	DESCRIÇÃO	QUANTIDADE	REFERENCIA	MODELO	FORNECEDOR	DATA DA COMPRA	OPERAÇÕES
2	Mouse	20	654654	HP	HP	2021-05-26	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Teclado	10	646546	Dell	Dell	2021-05-16	<a href="#">Alterar</a> <a href="#">Excluir</a>
5	Monitor	10	646457	LG	LG	2021-08-28	<a href="#">Alterar</a> <a href="#">Excluir</a>

Exportar Dados

Fonte: elaborado pelo autor

## 7 IMPLEMENTAÇÃO

A linguagem de programação utilizada para desenvolver o presente sistema foi Java, utilizando páginas JSP para exibir as informações nas páginas HTML. O sistema foi desenvolvido de forma monolítica, isto é, em um único projeto. Uma outra implementação foi criada para permitir vislumbrar como seria possível disponibilizar as informações do sistema monolítico para um outro sistema que pudesse obter os dados e exibi-los em outra interface (outro front-end). Os códigos implementados estão disponíveis nestes repositórios públicos:

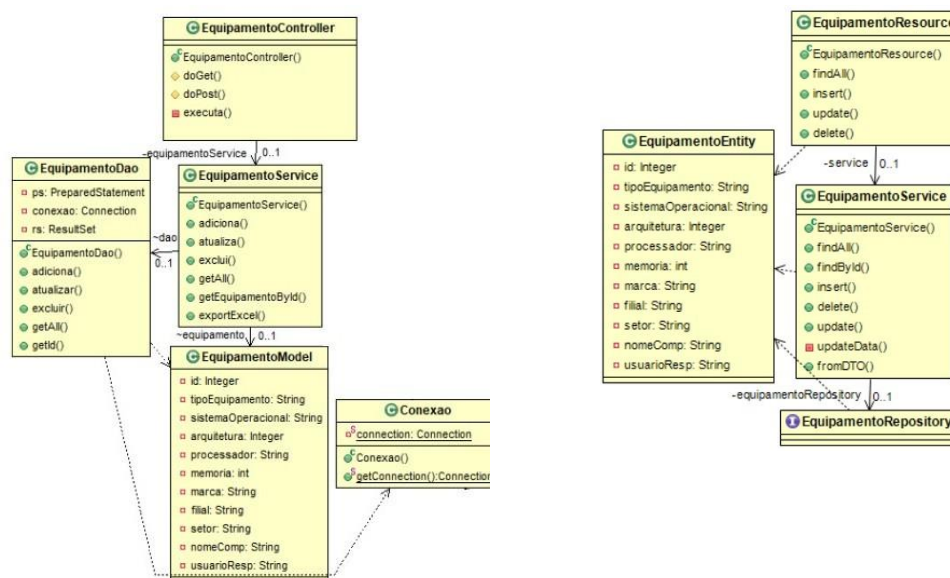
- Sistema monolítico: [https://gitlab.com/yuri\\_patrick347/sgti-mono](https://gitlab.com/yuri_patrick347/sgti-mono)
- Microserviço: [https://gitlab.com/yuri\\_patrick347/sgti-service](https://gitlab.com/yuri_patrick347/sgti-service)

## 7.1 ARQUITETURA MONOLÍTICA E MICROSERVIÇOS

O sistema de controle de equipamentos de TI desenvolvido neste trabalho possui arquitetura monolítica: uma única aplicação contém as camadas de visão, modelo e controle. Para demonstrar como seria a separação dessas camadas, as informações de modelo e controle foram replicadas em uma outra aplicação, sob forma de microserviços. Dessa maneira, os dados do sistema podem ser consultados por outras interfaces ou por outros sistemas, abrindo dessa forma o acesso ao sistema por outra via: os microserviços.

A modelagem de classes nesses dois tipos de arquitetura possui diferenças que serão descritas a seguir. Na arquitetura monolítica, a classe que contém os dados de um equipamento, por exemplo, é denominada EquipamentoModel, enquanto no sistema orientado a microserviços foi utilizado o nome EquipamentoEntity (Figura 19).

Figura 19. Comparativo de classes dos sistemas monolítico e com microserviços



Fonte: elaborado pelo autor

Ambas as classes (EquipamentoModel e EquipamentoEntity) possuem os dados da entidade, e portanto são similares nas duas arquiteturas; elas possuem métodos set e get para cada atributo, que não foram incluídos para manter os diagramas simples. Na arquitetura monolítica, foi realizada uma implementação de persistência com uso de um objeto de acesso a dados (Data Access Object - DAO), sendo a classe EquipamentoDao referente à essa implementação. Com microserviços, foi utilizado o framework Hibernate para acesso aos dados, dispensando portanto a necessidade de classe específica de acesso a dados. Esse fato ocorre porque a implementação da classe de modelo com uso do hibernate agrega nos dados informações de persistência, por meio de anotações. Outra característica decorrente do uso do hibernate é que as tabelas são geradas pelo framework, enquanto na versão monolítica (com DAO) as tabelas são criadas de forma independente (manualmente pelo desenvolvedor). Essa diferença da forma de persistência se refere à implementação, e não está ligada diretamente à arquitetura utilizada (ou seja, a arquitetura monolítica poderia ter sido implementada com hibernate e a arquitetura de microserviços poderia ter feito uso de um componente DAO).

As classes DAO do sistema monolítico se conectam ao banco de dados por meio de uma classe criada com o nome de Conexao. Na versão de microserviços, as informações de conexão com o banco de dados são inseridas em um arquivo de configuração chamado “application.properties” (do framework Spring boot). Informações de conexão, na versão monolítica, são utilizadas pela classe EquipamentoDAO, enquanto na versão de microserviços as informações de conexão são disponibilizadas à classe de modelo (EquipamentoEntity) por meio da classe EquipamentoRepository.

Uma parte do sistema consiste nas classes responsáveis por realizar a interação entre as camadas de visão e modelo. Na arquitetura monolítica, essa camada é implementada pela classe EquipamentoController, e corresponde à camada de controle na arquitetura MVC. Essa classe trata as requisições GET e POST que realizam as funções de incluir, alterar, excluir e listar equipamentos. Essas requisições são solicitadas pelas páginas JSP incluídas no sistema monolítico. As páginas JSP contém códigos HTML e Java para serem capazes de tornar o conteúdo dinâmico. Na arquitetura de microserviços, a classe que contém o tratamento de requisições HTTP se chama EquipamentoResource, e contém as APIs (Access Programming Interface - API) que podem atender outras aplicações e também o front-end. As respostas dessas API são fornecidas no formato JSON, quando necessário (operações de exclusão e atualização de dados não possuem conteúdo em suas respostas).

As classes `EquipamentoService` possuem a mesma funcionalidade em ambas as arquiteturas: são trechos de códigos (métodos utilitários) que estão disponíveis para o controlador, na versão monolítica, ou para o resource, na versão de microsserviços.

Um sistema monolítico funciona em um único projeto, acoplando o front-end e o back-end. O sistema não pode ser desmembrado para ser executado em partes separadas. As tecnologias de front-end e back-end geralmente são a mesma neste tipo de sistema. Já na implementação com microsserviços, é possível compartilhar o back-end com outros sistemas, ou tornar o sistema acessível a partir de outros front-ends. As tecnologias de back-end e front-end podem ser desenvolvidas em tecnologias diferentes, e esses componentes podem até mesmo ser executados em servidores diferentes. A arquitetura de sistemas tem evoluído da construção de sistemas monolíticos para sistemas separados em front-end e back-end.

## **8 CONSIDERAÇÕES FINAIS E TRABALHO FUTUROS**

O presente trabalho apresentou um sistema para gerenciar os equipamentos e os produtos em estoque, em contraposição ao método antigo de organizar dados em planilha eletrônica, ou em bloco de notas, utilizando então uma maneira mais organizada. Dessa forma, o processo de gerenciamento e controle se torna mais eficiente, gerando rapidez e segurança. Além disso, foi realizada uma implementação das funcionalidades básicas do sistema usando uma arquitetura de microsserviços, considerando apenas a parte de manipulação de dados do sistema por meio de APIs (back-end).

No sistema completo, a utilização do sistema mostrou-se eficiente e de fácil uso, tornando as informações mais organizadas e evitando a perda de dados. O sistema mostra as informações com clareza e de forma coerente ao usuário do sistema. São executados diversos procedimentos, como a movimentação de equipamentos e produtos, a troca de equipamento para outro usuário, a listagem dos produtos em estoque e o controle de entrada e saída dos produtos em estoque. Esses fluxos de processos são controlados de uma melhor maneira em um sistema de gerenciamento de equipamento de TI do que em planilhas ou arquivos de texto.

No desenvolvimento dos micro serviços, o conjunto de funcionalidades desenvolvidas fica acessível a outros sistemas. É possível também desenvolver mais de um front-end em diferentes tecnologias (por exemplo: desktop, web e mobile) de forma que todas essas interfaces acessem o mesmo conjunto de informações, disponibilizadas por APIs em um back-end. O sistema torna-se, portanto, mais flexível.

## REFERÊNCIAS

ARAÚJO, Carlos. **O IDE Eclipse - Revista easy Java Magazine 19 - Parte1**, 2012. Disponível em: <<https://www.devmedia.com.br/o-ide-eclipse-revista-easy-java-magazine-19-parte1/24796>>. Acesso em 28 agosto de 2021.

ARTIS. **Solução completa para rastreamento e gestão de ativos de TI**, 2020. Disponível em: <<https://nexxto.com/solucoes/artis/>>. Acesso em 20 de outubro de 2021.

BACINGER, Tomislav. **What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How**, 2015. Disponível em: <<https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>>. Acesso em 28 de agosto de 2021.

BECKER, Ansgar. **Heidisql**, 2020. Disponível em <<http://www.heidisql.com/>>. Acesso em 20 de outubro de 2021.

BRITO, Edivaldo. **Java: Entenda para que serve o software e os problemas da sua ausência**, 2014. Disponível em <<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2014/11/java-entenda-para-que-serve-o-software-e-os-problemas-da-sua-ausencia.htm>>. Acesso em 01 de novembro de 2021.

CELESTINO, André Luis. **A importância dos requisitos não-funcionais**, 2013. Disponível em: <<http://www.profissionaisti.com.br/2013/02/a-importancia-dos-requisitos-nao-funcionais/>>. Acesso em 20 de outubro de 2021.

COSTA JÚNIOR, Llaim. **Sistemas de Informação Gerenciais**, 2014. Disponível em: <<http://www2.ic.uff.br/~ilaim/Sistemas%206.pdf>>. Acesso em 01 de novembro de 2021.

IBM. **What is JDBC**, 2019. Disponível em: <<https://www.ibm.com/docs/en/informix-servers/12.10?topic=started-what-is-jdbc>>. Acesso em 28 agosto de 2021.

LUCIDCHART. **O que é diagrama de classe.** Disponível em:<<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>> Acesso em 20 de outubro de 2021.

MEDEIROS, Higor. **Introduzindo o servidor de aplicação Apache Tomcat**, 2013. Disponível em: <<https://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939>>. Acesso em 17 de novembro de 2021.

MILVUS. **Gestão de ativos de TI**, 2020. Disponível em: <<https://milvus.com.br/gestao-de-ativos-e-inventario/>>. Acesso em 01 de novembro de 2021.

MOREIRA, Pedro Felipe Marques; BEDER, Delano Medeiros. **Desenvolvimento de Aplicações e Microsserviços: Um estudo de caso.** Revista TIS, v. 4, n. 3, 2016.

SANTANA, Eduardo. **Apache POI: Manipulando Documentos em Java**, 2014. Disponível em:<<https://www.devmedia.com.br/apache-poi-manipulando-documentos-em-java/31778>>. Acesso em 28 agosto de 2021.

SILVA FILHO, Antonio Mendes da. **Engenharia de Software 3 - Requisitos Não Funcionais**, 2008. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>>. Acesso em 20 de outubro de 2021.

SILVA, Ivanila; ALTURAS, Bráulio; CARNEIRO, Alberto. **UML modeling tools: Assessment in perspective of users.** In: 2017 12th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2017. p. 1-6.

SOUZA, Ivan. **MariaDB ou MySQL: qual tecnologia de banco de dados escolher**, 2020. Disponível em:<<https://rockcontent.com/br/blog/mariadb/>>. Acesso em 28 agosto de 2021.

ROCHA, Fabio. **Introdução ao Java Server Pages - JSP**, 2012. Disponível em:<<https://www.devmedia.com.br/introducao-ao-java-server-pages-jsp/25602>>. Acesso em 28 agosto de 2021.