

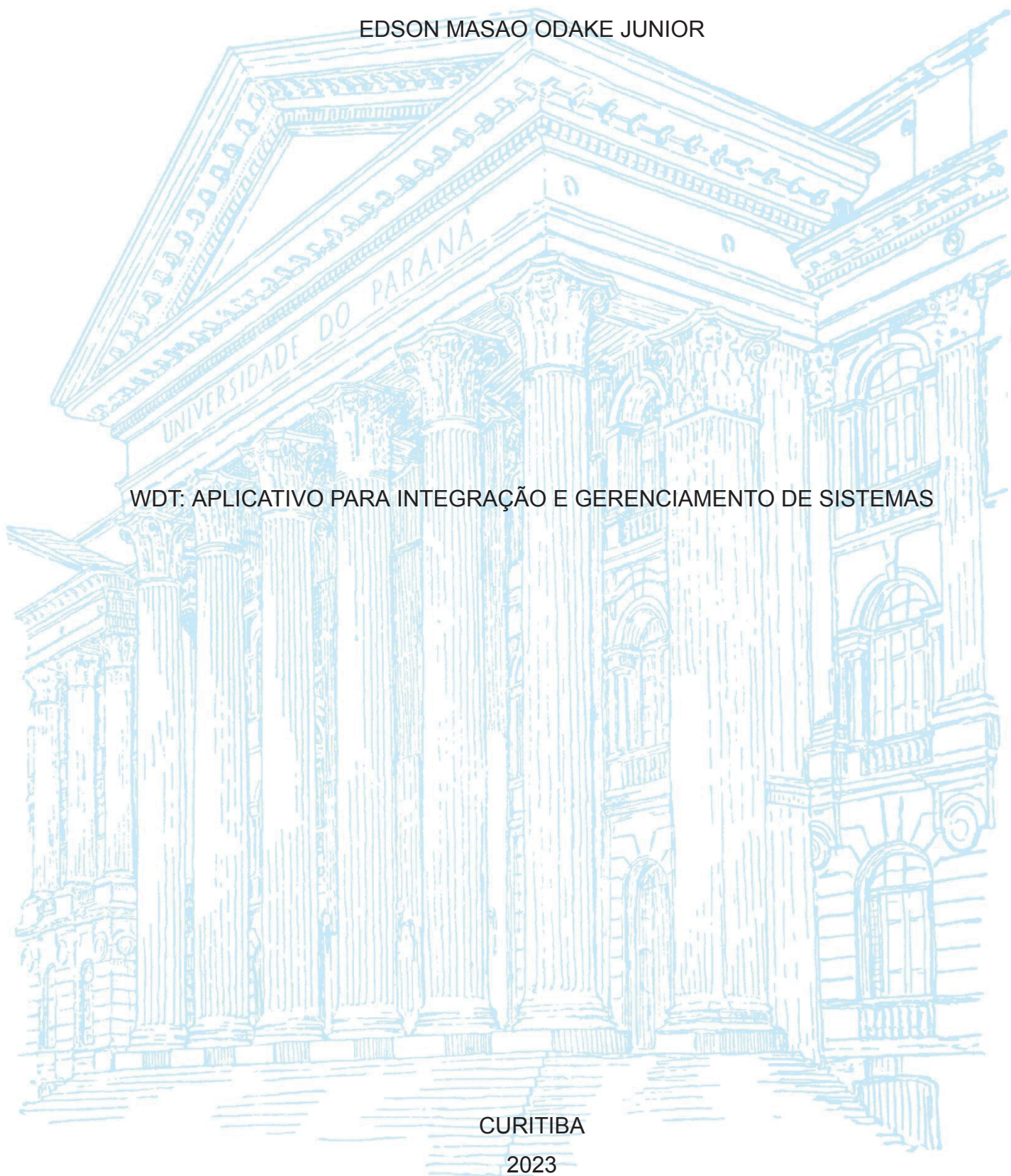
UNIVERSIDADE FEDERAL DO PARANÁ

EDSON MASAO ODAKE JUNIOR

WDT: APLICATIVO PARA INTEGRAÇÃO E GERENCIAMENTO DE SISTEMAS

CURITIBA

2023



EDSON MASAO ODAKE JUNIOR

WDT: APLICATIVO PARA INTEGRAÇÃO E GERENCIAMENTO DE SISTEMAS

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Desenvolvimento Ágil de Software, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Desenvolvimento Ágil de Software.

Orientador: Prof. Dr. Jaime Wojciechowski

CURITIBA

2023

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação DESENVOLVIMENTO ÁGIL DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **EDSON MASAO ODAKE JUNIOR** intitulada: **WDT: APLICATIVO PARA INTEGRAÇÃO E GERENCIAMENTO DE SISTEMAS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 31 de Outubro de 2023.



JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora



RAZER ANTHON NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

“Before you marry a person you should first make them use a computer with slow internet to see who they really are.” - Will Ferrell

“Technical debt’ that is not being paid down. It comes from taking shortcuts, which may make sense in the short-term. But like financial debt, the compounding interest costs grow over time. If an organization doesn’t pay down its technical debt, every calorie in the organization can be spent just paying interest, in the form of unplanned work.” — Gene Kim, *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*

RESUMO

Durante o desenvolvimento e fabricação de novos produtos, as indústrias enfrentam uma série de etapas de produção. Quanto mais complexo o produto, maior é a necessidade de etapas para garantir qualidade e cumprimento de prazos para o cliente. A automação de processos industriais desempenha um papel fundamental, com o uso de controladores e dispositivos automatizados. A automação normalmente é feita com a ajuda de controladores lógicos programáveis (CLPs), esses controladores se conectam a uma rede de dispositivos. Acima disso, existem sistemas supervisórios, que são basicamente interfaces gráficas utilizadas para o acompanhamento e envio comandos aos dispositivos da rede. No entanto, a integração desses dispositivos com um software supervisório pode ser complexa, especialmente quando envolve rastreamento de produtos e integração com bases de dados. Com o intuito de simplificar esse processo, foi desenvolvido um software especializado em automação industrial. O software oferece uma interface gráfica que permite aos usuários mapear variáveis de hardware e criar rotinas personalizadas que englobam uma série de etapas. Após o cadastro dessas rotinas, os usuários podem facilmente fazer o download do projeto, obtendo arquivos de configuração prontos para serem interpretados por aplicações externas. Essa abordagem simplifica a automação industrial, tornando-a mais eficiente e acessível, enquanto garantindo a execução das ações planejadas de forma eficaz.

Palavras-chave: Gerenciamento de projetos. Automação industrial. Integração de sistemas, Angular, Java, MongoDB

ABSTRACT

During the development and manufacturing of new products, industries face a series of production stages. The more complex the product, the greater the need for stages to ensure quality and meeting customer deadlines. Industrial process automation plays a fundamental role, using controllers and automated devices. Automation is typically done with the help of programmable logic controllers (PLCs), which connect to a network of devices. Furthermore, there are supervisory systems, which are essentially graphical interfaces used for monitoring and sending commands to the networked devices. However, integrating these devices with supervisory software can be complex, especially when it involves product tracking and database integration. In order to simplify this process, specialized industrial automation software has been developed. The software offers a graphical interface that allows users to map hardware variables and create custom routines that encompass a series of steps. After registering these routines, users can easily download the project, obtaining ready-to-use configuration files for external applications. This approach simplifies industrial automation, making it more efficient and accessible while ensuring the effective execution of planned actions.

Keywords: Project management. Industrial automation. Systems integration, Angular, Java, MongoDB

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1 - INTERAÇÃO E ARQUITETURA DO SOFTWARE | 26 |
| FIGURA 2 – DIAGRAMA DE ATIVIDADES DO DEVELOPMENT TOOL | 32 |
| FIGURA 3 – DIAGRAMA DE CASO DE USO | 33 |
| FIGURA 4 – TELA DE LOGIN | 35 |
| FIGURA 5 – TELA INICIAL | 36 |
| FIGURA 6 – TELA REDIRECIONAMENTO DECODER | 36 |
| FIGURA 7 – TELA REDIRECIONAMENTO ANALYZER | 37 |
| FIGURA 8 – TELA LISTAGEM USUÁRIO | 38 |
| FIGURA 9 – TELA CADASTRO DE USUÁRIO | 39 |
| FIGURA 10 – TELA ATUALIZAÇÃO DE USUÁRIO | 39 |
| FIGURA 11 – TELA LISTA DE PROJETOS | 41 |
| FIGURA 12 – TELA CADASTRO DE PROJETOS | 42 |
| FIGURA 13 – TELA ATUALIZAÇÃO DE PROJETOS | 43 |
| FIGURA 14 – TELA DOWNLOAD DE PROJETOS | 45 |
| FIGURA 15 – TELA LISTA DE MAPEAMENTO | 46 |
| FIGURA 16 – TELA ATUALIZAÇÃO DE MAPEAMENTO | 47 |
| FIGURA 17 – TELA CADASTRO DE MAPEAMENTO | 48 |
| FIGURA 18 – TELA GERAL DE CONFIGURAÇÃO | 50 |
| FIGURA 19 – TELA LISTA DE CONFIGURAÇÃO | 50 |
| FIGURA 20 – TELA CADASTRO DE CONFIGURAÇÃO | 51 |
| FIGURA 21 – TELA ATUALIZAÇÃO DE CONFIGURAÇÃO | 52 |
| FIGURA 22 – TELA LISTA DE ROTINAS | 54 |
| FIGURA 23 – TELA ATUALIZAÇÃO DE ROTINAS | 55 |
| FIGURA 24 – TELA CADASTRO DE ROTINAS | 56 |
| FIGURA 25 – TELA LISTA DE ETAPAS (TAGS) | 58 |
| FIGURA 26 – TELA ATUALIZAÇÃO DE EVAL TAG | 59 |
| FIGURA 27 – TELA CADASTRO DE EVAL TAG | 60 |
| FIGURA 28 – TELA ATUALIZAÇÃO DE ETHERNET TAG | 62 |
| FIGURA 29 – TELA CADASTRO DE ETHERNET TAG | 63 |
| FIGURA 30 – TELA ATUALIZAÇÃO DE TEST TAG | 65 |
| FIGURA 31 – TELA CADASTRO DE TEST TAG | 66 |
| FIGURA 32 – TELA ATUALIZAÇÃO DE WRITE TAG | 68 |

| | |
|---|----|
| FIGURA 33 – TELA CADASTRO DE WRITE TAG | 69 |
| FIGURA 34 – TELA ATUALIZAÇÃO DE COMPARE TAG | 71 |
| FIGURA 35 – TELA CADASTRO DE COMPARE TAG | 72 |
| FIGURA 36 – DIAGRAMA DE CLASSE DE IMPLEMENTAÇÃO | 75 |
| FIGURA 37 – DIAGRAMA DE CLASSE | 76 |
| FIGURA 38 – DIAGRAMA DE SEQUÊNCIA DA HU01 - FAZER LOGIN | 77 |
| FIGURA 39 – DIAGRAMA DE SEQUÊNCIA DA HU02 - CRUD USUÁRIO | 77 |
| FIGURA 40 – DIAGRAMA DE SEQUÊNCIA DA HU03 - LISTAR PROJETO | 78 |
| FIGURA 41 – DIAGRAMA DE SEQUÊNCIA DA HU04 - CADASTRAR PROJETO | 78 |
| FIGURA 42 – DIAGRAMA DE SEQUÊNCIA DA HU05 - MANTER PROJETO | 79 |
| FIGURA 43 – DIAGRAMA DE SEQUÊNCIA DA HU06 - BAIXAR PROJETO | 79 |
| FIGURA 44 – DIAGRAMA DE SEQUÊNCIA DA HU07 - LISTAR MAPEAMENTO | 80 |
| FIGURA 45 – DIAGRAMA DE SEQUÊNCIA DA HU08 - MANTER MAPEAMENTO | 80 |
| FIGURA 46 – DIAGRAMA DE SEQUÊNCIA DA HU09 - CADASTRAR MAPEAMENTO | 81 |
| FIGURA 47 – DIAGRAMA DE SEQUÊNCIA DA HU10 - LISTAR CONFIGURAÇÃO | 81 |
| FIGURA 48 – DIAGRAMA DE SEQUÊNCIA DA HU11 - CADASTRAR CONFIGURAÇÃO | 82 |
| FIGURA 49 – DIAGRAMA DE SEQUÊNCIA DA HU12 - MANTER CONFIGURAÇÃO | 82 |
| FIGURA 50 – DIAGRAMA DE SEQUÊNCIA DA HU13 - LISTAR ROTINA | 83 |
| FIGURA 51 – DIAGRAMA DE SEQUÊNCIA DA HU14 - CADASTRAR ROTINA | 83 |
| FIGURA 52 – DIAGRAMA DE SEQUÊNCIA DA HU15 - MANTER ROTINA | 84 |
| FIGURA 53 – DIAGRAMA DE SEQUÊNCIA DA HU16 - LISTAR TAG | 84 |
| FIGURA 54 – DIAGRAMA DE SEQUÊNCIA DA HU17 - MANTER EVALTAG | 85 |
| FIGURA 55 – DIAGRAMA DE SEQUÊNCIA DA HU18 - MANTER ETHERNETTAG | 85 |
| FIGURA 56 – DIAGRAMA DE SEQUÊNCIA DA HU19 - MANTER TESTTAG | 86 |
| FIGURA 57 – DIAGRAMA DE SEQUÊNCIA DA HU20 - MANTER WRITETAG | 86 |
| FIGURA 58 – DIAGRAMA DE SEQUÊNCIA DA HU21 - MANTER COMPARETAG | 87 |
| FIGURA 59 – DIAGRAMA DE BASE DE DADOS (NoSQL) | 88 |

LISTA DE TABELAS

| | |
|--|----|
| TABELA 1 – VALIDAÇÃO DE LOGIN | 38 |
| TABELA 2 – VALIDAÇÃO DE FORMULÁRIO | 43 |
| TABELA 3 – VALIDAÇÃO DE FORMULÁRIO MAPEAMENTO | 49 |
| TABELA 4 – VALIDAÇÃO DE FORMULÁRIO CONFIGURAÇÃO | 53 |
| TABELA 5 – VALIDAÇÃO DE FORMULÁRIO ROTINA | 57 |
| TABELA 6 – VALIDAÇÃO DE FORMULÁRIO EVALTAG | 62 |
| TABELA 7 – VALIDAÇÃO DE FORMULÁRIO ETHERNETTAG | 65 |
| TABELA 8 – VALIDAÇÃO DE FORMULÁRIO TESTTAG | 67 |
| TABELA 9 – VALIDAÇÃO DE FORMULÁRIO WRITETAG | 71 |
| TABELA 10 – VALIDAÇÃO DE FORMULÁRIO REGISTERCOMPARETAG | 74 |

SUMÁRIO

| | |
|--|-----------|
| 1 VISÃO GERAL DO PROJETO | 16 |
| 1.1 CONTEXTUALIZAÇÃO | 16 |
| 1.2 DESCRIÇÃO DO PRODUTO DO PROJETO | 19 |
| 1.2.1 Development Tool | 20 |
| 1.2.2 Decoder | 22 |
| 1.2.3 Analyzer | 22 |
| 1.3 FUNCIONALIDADES | 24 |
| 1.4 TECNOLOGIAS UTILIZADAS | 24 |
| 1.4.1 Java | 28 |
| 1.4.2 Angular | 29 |
| 1.4.3 MongoDB | 29 |
| 1.4.4 Spring | 30 |
| 1.4.5 Typescript | 31 |
| 2 DESENHO DO PROCESSO MACRO | 31 |
| 3 DIAGRAMA DE CASOS DE USO | 32 |
| 4 HISTÓRIAS DE USUÁRIO | 33 |
| 5 DIAGRAMA DE CLASSES | 75 |
| 6 DIAGRAMAS DE SEQUÊNCIA | 77 |
| 7 DIAGRAMA FÍSICO DO BANCO DE DADOS | 88 |
| REFERÊNCIAS | 89 |

1 VISÃO GERAL DO PROJETO

1.1 CONTEXTUALIZAÇÃO

Em toda indústria, uma série de processos ocorre sequencialmente para resultar em um produto final. Engenheiros industriais dedicam seus esforços à otimização desses processos, visando produzir o produto final com o menor custo industrial possível. Mesmo com a automação amplamente presente nos dias de hoje, é fundamental reconhecer que, em seus estágios iniciais, houve um extenso foco em estudos voltados para aprimorar a produtividade em tarefas manuais como o Fordismo (PIETRYKOWSKI, 1995; GORDON et al., 1982).

O Fordismo, caracterizado pelo trabalho padronizado e repetitivo, é um sistema de produção industrial em massa de forma eficiente e acessível, criado no início do século XX. O Fordismo foi uma estratégia implementada com o objetivo de otimizar a produtividade e elevar os padrões de qualidade na linha de produção (PIETRYKOWSKI, 1995).

À medida que a indústria amadureceu, surgiram novas ferramentas e abordagens inovadoras para aprimorar a eficiência e a qualidade da produção. Entre essas inovações, destacam-se o Six Sigma, um processo projetado para minimizar falhas nos processos de manufatura, e o desenvolvimento de softwares de simulação, além de uma variedade de outras ferramentas e técnicas. Essas novas ferramentas desempenharam um papel fundamental na otimização dos processos industriais e na redução de defeitos, contribuindo para a excelência na produção (HWAIYU GENG, 2016).

Novos processos altamente automatizados, com intervenção humana reduzida ao mínimo, surgiam. Essa abordagem resultou em melhorias notáveis na qualidade e na eficiência da produção, enquanto também reduzia a variabilidade e estabelecia tempos de produção mais precisos e previsíveis (GRAU et al., 2017).

Após a segunda guerra mundial, grandes avanços tecnológicos amadureceram de forma acelerada o desenvolvimento das indústrias (MOWERY, 2008). Com todo um leque novo de máquinas e equipamentos, o objetivo agora é tirar o trabalho da mão dos operadores e automatizar os processos com microcontroladores, máquinas de fresa ou solda elétrica.

Embora todos esses avanços tenham sido realizados, o trabalho ainda ocorre de forma sequencial, indo de etapa a etapa até termos no fim um produto completo.

Difícilmente uma empresa terá somente um produto no seu catálogo. É normal empresas multinacionais oferecerem centenas ou até milhares de produtos, como é o caso da ABB ou da Siemens (ABB, 2023; SIEMENS, 2023). E para piorar, cada produto pode ser composto por componentes que também passam pelo seu próprio processo de manufatura. Como é o caso de cartões eletrônicos com centenas de componentes.

O que se tem nas indústrias hoje não é um problema linear. São centenas de células de produção trabalhando tanto de forma sequencial quanto paralela. O objetivo que toda indústria tem, é a capacidade de monitorar e controlar o máximo possível de transações e etapas que ocorrem nos processos de manufatura.

Após a conclusão da fabricação de dispositivos eletrônicos, ocorre uma fase de testes. Durante essa etapa, os produtos são submetidos a uma série de testes funcionais para garantir que suas funcionalidades básicas estejam em perfeito estado. Posteriormente, é realizado um teste de carga, no qual os produtos são submetidos a uma operação que simula as condições de campo em que serão usados pelos clientes. Nesse processo, os produtos devem resistir a vibrações e manter sua temperatura dentro dos limites estabelecidos durante um período de tempo pré-determinado.

No caso de produções em larga escala, a maioria dos testes de produção é automatizada. Dependendo das funcionalidades dos produtos, eles são testados em hardwares significativamente diferentes. Para cada um desses hardwares, é essencial que um desenvolvedor crie um novo sistema que inclua o mapeamento de todas as entradas e saídas, protocolos de comunicação e arquitetura de rede do sistema. Além disso, é necessário desenvolver integrações com os sistemas da empresa e criar novas tabelas para armazenamento de dados. A coleta de dados é algo essencial para controle de qualidade, manutenção preventiva, rastreabilidade e outras conformidades regulatórias,

Para contextualizar melhor o leitor, tem-se a seguinte situação hipotética. A empresa BAM produz um inversor de frequência, que tem como função o controle de velocidade de motores. Esse produto foi confeccionado e se encontra no procedimento de testes funcionais. Como a empresa tem seus processos

automatizados, o operador coloca o produto em uma bandeja, faz a leitura de um QR Code para identificar o serial do produto e inicia o teste do produto em uma tela de supervisão.

O software que se encontra no supervisão faz a busca do serial daquele produto no sistema da empresa, verificando se o produto está autorizado a ser testado naquele sistema, e em sequência inicializa a rotina de teste programada para aquele determinado produto.

No início do teste, o produto se encontra em uma bandeja, exposto ao operador e ao ambiente externo. O primeiro passo que o software deve realizar é isolar esse produto, tanto para proteção do operador, para não expô-lo a altas tensões elétricas e riscos de explosões, quanto para o isolamento térmico do produto, para que seja possível controlar o aquecimento do mesmo. O software aciona um motor de passos. É importante explicar que a bandeja onde o produto se encontra está em cima de uma esteira. Portanto, a bandeja começa a caminhar pela esteira até chegar a uma cabine fechada.

Em algum ponto da cabine existe um fim de curso (dispositivo utilizado para detectar presença) que envia um sinal para o software para desacionar o motor. Os inversores têm três encaixes de cada lado, que são utilizados para colocar os cabos por onde irá passar a corrente elétrica. O software supervisão envia então um comando para acionar outros motores, providenciando o encaixe desses cabos.

Antes de aplicar uma potência elevada no produto, para garantir que não haja nenhum curto-circuito, é aplicada uma tensão elevada, e então é medida a corrente de fuga que apresenta entre as fases (cabos encaixados) e o terra. Para isso, o software envia um comando para o controlador lógico programável (CLP) solicitando o acionamento de contator para interligar o produto em componentes específicos do sistema. Contatores servem para fazer o chaveamento entre conexões, enquanto o CLP é capaz de comandar esses chaveamentos através de saídas digitais.

Com o produto conectado aos pontos devidos, o software envia um telegrama (sequência hexadecimal) para um dispositivo chamado HIPOT (CHROMA, 2023), onde é aplicada uma tensão elevada com uma potência baixa somente para garantir que o produto esteja eletricamente bem isolado em relação ao terra. O protocolo de comunicação desse produto é diferente do CLP, demandando uma diferente implementação no software.

Após a verificação da corrente de fuga, o software inicia o teste funcional do produto através dessa troca de comunicação entre o software supervisor e os demais dispositivos do sistema. No final do teste, os dados do ensaio são armazenados e os devidos apontamentos são feitos no sistema.

A BAM também tem a necessidade de testar outros produtos, como painéis utilizados em geradores eólicos, sensores inteligentes ou mesmo robos industriais. O hardware utilizado para testar o inversor provavelmente não é capaz de comportar essas novas demandas, tornando necessário o investimento em novos sistemas.

Nesse cenário, várias questões críticas emergem. Em primeiro lugar, o volume significativo de desenvolvimento de software se torna uma demanda gigante. Em segundo lugar, é essencial garantir suporte contínuo para essa crescente gama de sistemas em operação. Terceiro, com a diversificação do código-fonte, surgem desafios adicionais de gerenciamento e manutenção. Por último, a complexidade crescente torna a tarefa de documentação cada vez mais desafiadora, o que, por sua vez, implica que os engenheiros precisam alocar considerável tempo para auxiliar os técnicos responsáveis no entendimento e solução de problemas. Portanto, essas questões combinadas representam uma série de desafios para a equipe envolvida na gestão e manutenção desses vários sistemas.

Os desafios decorrentes da existência de múltiplos sistemas estão, principalmente, relacionados à necessidade de integrar diferentes tipos de hardware. Dado que a padronização de hardware se mostra economicamente inviável, fica evidente a importância de padronizar os softwares. Neste projeto, a proposta é desenvolver uma parte de um sistema que tem como finalidade a documentação de processos em uma fábrica, a automação de sistemas e a coleta de dados desses processos. Essa abordagem será implementada por meio de um conjunto de três softwares, sendo um deles o resultado deste trabalho.

1.2 DESCRIÇÃO DO PRODUTO DO PROJETO

O software desenvolvido nesse projeto faz parte de uma aplicação composta por três partes. Para facilitar o entendimento do sistema como um todo, todas as partes são explicadas nesse trabalho. Entretanto, o que será desenvolvido é um sistema independente batizado como Development Tool. Os softwares nomeados

como Analyzer e Decoder estão fora do escopo desse sistema, sendo apenas citados para uma contextualização geral.

1.2.1 Development Tool

O propósito central deste software reside na sua capacidade registrar e armazenar todos os dados necessários para que um determinado sistema seja capaz de executar tarefas de automação. Dessa forma é necessário criar um repositório que englobe todas as informações essenciais do sistema assim como dos processos realizados nesse sistema.

Para isso foram definidos algumas nomenclaturas para representar a ideia por trás do sistema. são elas:

- Projeto: O projeto contém basicamente toda e qualquer informação necessária para que uma aplicação externa (Decoder) seja capaz de interpretar e realizar a automatização de procedimentos em um determinado sistema. Portanto, para cada sistema (hardware) diferente, tem-se um projeto associado.
- Arquivos de configuração: São arquivos utilizados para a configuração do software (Decoder), nesses arquivos estão dados relacionados a base de dados e configurações específicas do sistema.
- Mapeamento: Utilizado para linkar registradores (endereços) de dispositivos a nomes específicos, facilitando o entendimento e futuras modificações.
- Rotina: Consistem em uma lista de comandos, possibilitando a execução de uma sequência de procedimentos automáticos.
- Tag: Representa um único comando, tags diferentes são capazes de exercer funções diferentes. Também é referenciado como "*step*" ou "etapa" ao longo do trabalho

O objetivo final é proporcionar uma representação detalhada de todas as etapas e procedimentos envolvidos, oferecendo uma visão simplista para facilitar a gestão e a compreensão completa do sistema em questão. Isso não apenas simplifica a manutenção e a documentação, mas também contribui para a eficiência e a transparência na gestão do projeto.

Este sistema armazena as rotinas por meio de uma lista de objetos no formato JSON. A sua funcionalidade principal é permitir que o usuário registre as etapas das rotinas usando um formulário intuitivo. Essas etapas são capturadas em um formato que pode ser facilmente interpretado pelo Decoder, uma ferramenta independente que possibilita a execução das ações necessárias pelo sistema.

A utilização de configurações em formato JSON oferece a vantagem de possibilitar a implementação de rotinas complexas, tudo isso sem exigir um conhecimento aprofundado de programação convencional. A abordagem é concebida de forma a simplificar a criação e a modificação das rotinas, facilitando a interação do usuário com o sistema tornando acessível a criação e a personalização de procedimentos envolvendo automação industrial para um público mais amplo, eliminando a necessidade de habilidades de programação específicas. Essa democratização do processo de implementação de automação não apenas economiza tempo e recursos, mas também promove a colaboração e a inovação.

As funcionalidades de algumas tags são explicadas a seguir:

- LeafTestTag: Essa tag é interpretada pelo Decoder de forma a mostrar na tela o início de um novo teste. Por exemplo, dentro de uma rotina pode-se ter vários testes, teste de isolamento, curto-circuito, carga nominal entre outros. Em cada um desses testes, tem-se vários comandos para que o teste seja efetuado. O TestTag divide esses testes para o usuário entender visualmente e intuitivamente o que está sendo feito
- LeafEthernetCommunicationTag: Essa tag recebe os dados para iniciar uma comunicação ethernet com um determinado dispositivo na rede. Dentre os dados que recebe, tem-se o IP do dispositivo, a quantidade de tentativas de comunicação que devem ser feitas, assim como uma descrição e etc.
- LeafWriteTag: Utilizada para enviar pacotes de dados para um dispositivo na rede através da conexão criada pela LeafEthernetCommunicationTag.
- LeafEvalTag: Recebe uma equação, sua função é executar essa equação e salvar o resultado em uma variável com o nome cadastrado no formulário dessa tag.

- LeafRegisterCompareTag: Utiliza a conexão criada pelo LeafEthernetCommunicationTag para requisitar dados de dispositivos na rede local. Como dado de entrada ele recebe o nome de dois dispositivos, os registradores (endereços de memória), assim como valores de tolerância e quantidade de amostras (samples). Então é feita a leitura dos dois registradores n vezes, sendo n definida pelo número de amostras. É então feito a média desses valores, e se a diferença entre os dois registradores lidos forem maior que uma tolerância, teste é dado como falho.

1.2.2 Decoder

A segunda parte deste sistema representa um software capaz de realizar a interpretação do compilado de objetos que compõem o projeto, além de executar as rotinas previamente definidas. Através desse software, o computador é habilitado a assumir, de maneira automatizada, o controle de diversos dispositivos interconectados por meio de redes Ethernet, Modbus, CAN, ou qualquer outro protocolo previamente configurado.

Nesse software, são realizados todos os apontamentos para integrações com APIs externas, bem como o armazenamento dos dados essenciais em um banco de dados centralizado. Esse processo de coleta e armazenamento de informações é fundamental para a gestão e o monitoramento contínuo do sistema.

Além disso, essa parte do sistema oferece a capacidade de interação com operadores, permitindo que eles recebam instruções e forneçam informações necessárias para a execução das rotinas.

1.2.3 Analyzer

Os dados registrados pelo software Decoder são armazenados em um banco de dados não relacional, caracterizado por sua alta variabilidade. Portanto, uma abordagem tradicional e estática para análise de dados não se adequaria eficazmente a esse cenário dinâmico e complexo.

Para viabilizar a visualização e a interpretação desses dados, é essencial criar uma estrutura flexível e adaptável, capaz de filtrar e selecionar os dados

necessários de maneira precisa. O Analyzer desempenha um papel fundamental nesse contexto, sendo uma parte integral do sistema. Sua principal função é executar consultas no banco de dados, possibilitando a construção de um pipeline personalizado para pré-processamento dos dados de acordo com as preferências do usuário.

Os dados assim processados são então armazenados em variáveis, tornando-os disponíveis para a criação de diversos dashboards personalizados. Esses dashboards oferecem uma visão ampla e detalhada dos processos em um ambiente fabril, permitindo um planejamento mais elaborado e estratégico. Essa abordagem não apenas facilita o acompanhamento em tempo real, mas também capacita a tomada de decisões informadas e a otimização contínua dos processos industriais. Portanto, o Analyzer desempenha um papel fundamental na extração de valor dos dados, transformando-os em insights acionáveis para a gestão eficiente de operações fabris.

Assim, a colaboração das três aplicações independentes se traduz em uma sinergia que permite:

- Automatizar Procedimentos: Tanto a automação de máquinas quanto a instrução de operadores tornam-se possíveis e eficazes. Isso resulta na execução eficiente de tarefas, independentemente de serem realizadas por máquinas controladas ou por intervenção humana.
- Registro de Dados Simplificado: A capacidade de salvar dados em cada etapa do sistema, de acordo com o fluxo estabelecido nas rotinas, é simplificada e altamente eficaz. Isso garante que informações cruciais sejam capturadas e armazenadas sem complicações.
- Monitoramento e Controle Integral: A interconexão das aplicações permite um monitoramento completo e um controle abrangente do sistema em operação. Essa abordagem oferece uma visão em tempo real e detalhada de todas as operações, facilitando o gerenciamento eficiente e a tomada de decisões fundamentadas.

A colaboração entre as três aplicações independentes cria um ambiente que possibilita a automação de tarefas, a captura eficiente de dados e o controle eficaz do sistema. Isso contribui significativamente para a operação eficiente e aprimoramento contínuo dos processos industriais.

1.3 FUNCIONALIDADES

As funcionalidades presentes no sistema desenvolvido neste trabalho (Development Tool) são as seguintes:

- O usuário pode fazer cadastro de outro usuário informando nome, endereço, sobrenome e senha.
- Interface para redirecionamento para o sistema externo Decoder.
- Interface para redirecionamento para o sistema externo Analyzer.
- O usuário faz cadastro de projeto informando nome, palavras chaves, nó, conexão e descrição (nós e conexões são variáveis utilizadas para entender onde o sistema fisicamente se encontra, e em quais sistemas o mesmo se encontra conectado).
- O usuário faz cadastro de mapeamento de variáveis do sistema.
- O usuário faz cadastro de rotinas do sistema.
- O usuário faz cadastro de configuração do sistema.
- O usuário faz cadastro de etapas de rotina (tags) podendo escolher entre diferentes tipos de operações.
- O usuário pode desabilitar projeto, mapeamento, configuração, rotina ou etapa (tag).
- O usuário pode fazer operações de *drag and drop* para mudar sequência de etapas da rotina (tags).
- O usuário pode fazer download das configurações de projeto para *deploy*.

1.4 TECNOLOGIAS UTILIZADAS

A arquitetura do sistema representada de forma simplificada pode ser verificada na Figura 1. O sistema como um todo pode ser fragmentado em 6 diferentes containers. Os módulos desenvolvidos neste trabalho são os containers referêntes ao bloco de Development Tool.

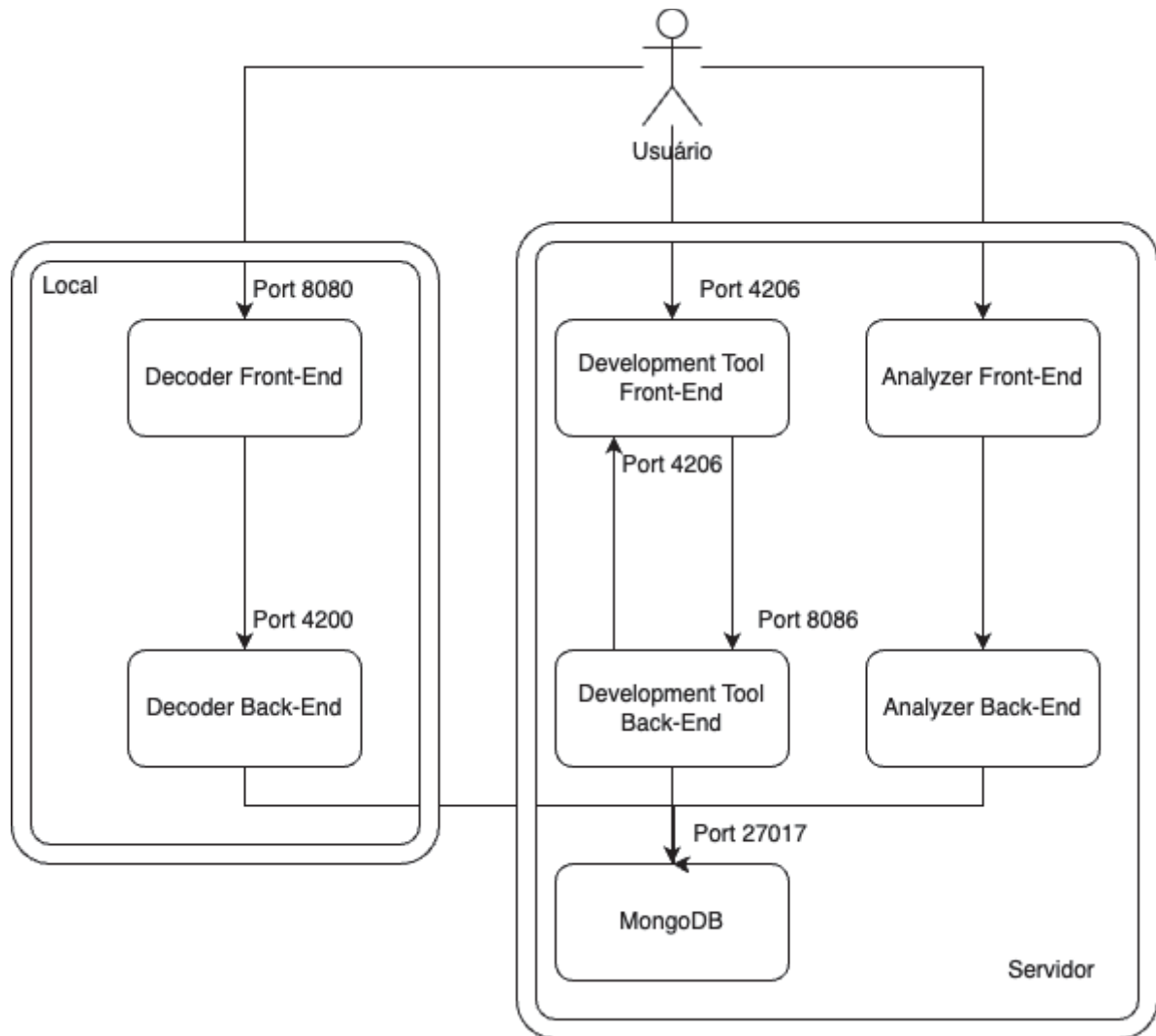
O Decoder foi concebido como uma aplicação desktop que opera em um computador local. Em sua função principal, ele acessa os dados armazenados no banco de dados por meio do aplicativo Development Tool. O Decoder desempenha o papel de interpretação dos arquivos gerados pelo Development Tool. Após executar

os procedimentos especificados em uma lista de objetos JSON, o Decoder armazena esses mesmos objetos no MongoDB, desta vez incorporando os valores resultantes do ensaio.

O Development Tool opera em um servidor e desempenha uma função de registro, que é a gestão de projetos e criação de novos projetos. Isso envolve a interação com o MongoDB, gerando arquivos de configuração necessários para garantir o correto funcionamento do Decoder. Além disso, o aplicativo tem a capacidade de armazenar os arquivos gerados em diretórios específicos no computador.

O Analyzer ajuda na simplificação da interpretação dos dados coletados, que muitas vezes são não estruturados e apresentam uma grande variabilidade. Ele é a ferramenta que possibilita a criação de queries e pipelines de dados para a geração de dashboards, proporcionando informações visuais para o processo de tomada de decisões. Essencialmente, o Analyzer atua na transformação de dados brutos em insights.

FIGURA 1. INTERAÇÃO E ARQUITETURA DO SOFTWARE.



FONTE: O Autor (2023)

O Development Tool e o Decoder compartilham da mesma base de dependência, com exceção de algumas bibliotecas específicas. Os dois têm o *front-end* desenvolvido em Angular e o *back-end* em Java.

Para o Analyzer foi decidido uma abordagem diferente, em que foi utilizado o Flutter no *front-end* e o Python no *back-end*, para aproveitar algumas facilidades do python com relação ao processamento e análise de dados.

As tecnologias empregadas são listadas a seguir. Posteriormente, as tecnologias adotadas no Development Tool serão abordadas de forma mais detalhada, destacando seus aspectos positivos e limitações.

- Development Tool e Decoder
 - Front-end
 - Angular CLI: 14.1.3.
 - Node: 16.15.1.
 - Gerenciador de Pacotes: npm 8.12.0.
 - Sistema Operacional: win32 x64.
 - Back-end
 - JDK (Java Development Kit) 11.0.10 ou superior.
 - Spring Boot 2.7.5.
- Analyzer
 - Front-end
 - Flutter: *Framework* usado para criar aplicativos móveis multiplataforma com um único código-fonte e interface de usuário altamente personalizável.
 - Dart: Desenvolvida pela Google em 2011 e frequentemente usada para desenvolver aplicativos móveis e web com o *framework* Flutter
 - Back-end
 - Python: Uma das escolhas principais para análise de dados e aprendizado de máquinas, graças à sua simplicidade, comunidade e robustas bibliotecas de já desenvolvidas.
 - Flask: Um *micro framework* web em Python, ideal para desenvolvimento de APIs de forma rápida e eficiente,

possibilitando a criação de aplicativos sem a necessidade de uma configuração exaustosa.

- Base de dados
 - MongoDB: Version v6.0.2.

1.4.1 Java

Java (ORACLE, 2023) é uma linguagem de programação orientada a objetos que foi concebida e desenvolvida em meados da década de 1990. Inicialmente, seu propósito era atender a demanda por sistemas embarcados, mas ao longo do tempo, Java expandiu para se tornar uma das linguagens mais versáteis e amplamente adotadas em diversas aplicações, incluindo softwares empresariais de larga escala.

Uma das características notáveis do Java é sua capacidade de ser uma linguagem multiplataforma. Isso se deve ao fato de que o código-fonte Java é compilado em um formato intermediário chamado *bytecode*, que é executado em uma máquina virtual Java (JVM - *Java Virtual Machine*). Essa abstração de máquina virtual permite que os programas Java sejam independentes da plataforma em que são executados. Ou seja, um software compilado em um ambiente Windows pode ser facilmente utilizado em um ambiente Linux ou qualquer outro sistema operacional que possua uma JVM compatível.

Além disso, Java se destacou no desenvolvimento de aplicações empresariais, graças à sua estabilidade, segurança e suporte para recursos avançados, como concorrência e gerenciamento de memória. Isso fez com que a linguagem fosse amplamente adotada na criação de sistemas corporativos, sistemas de gerenciamento de banco de dados, aplicativos web e uma variedade de soluções empresariais robustas.

Java também possui uma rica coleção de bibliotecas e frameworks que simplificam o desenvolvimento de software, tornando-o eficiente e produtivo. Entre essas bibliotecas, destacam-se o *Framework* Spring e o Hibernate.

Como vantagens, o Java tem uma comunidade muito ampla de desenvolvedores, é uma linguagem portátil por rodar em uma JVM e tem todas as vantagens atreladas a uma linguagem orientada ao objeto. Em contrapartida, é uma

linguagem que tende a ser muito verbosa, não é tão rápida como o C ou C++ e apresenta uma complexidade significativa pra iniciantes.

1.4.2 Angular

O Angular (ANGULAR, 2023), criado pela Google em 2016, é um framework de código aberto desenvolvido principalmente com a linguagem de programação TypeScript. Este framework utilizado principalmente no desenvolvimento front-end de aplicações web. No entanto também é possível criar aplicações desktop utilizando-o em conjunto com o Electron.

O Angular é especialmente projetado para atender às necessidades das Single Page Applications (SPAs), o que lhe confere a capacidade de proporcionar uma experiência de usuário mais fluída. Uma característica distintiva do Angular é a sua abordagem baseada em componentes, que permite a criação de interfaces de usuário reutilizáveis e modulares.

Como vantagens o Angular é uma linguagem com uma estrutura muito bem organizada, com uma estrutura bem definida e projetada já com uma mentalidade de *Test Driven Development*, além de ter um ecossistema de bibliotecas bem extenso. Entretanto, assim como o Java, o *framework* tem uma curva de aprendizado lenta, e com a constante quantidade de atualizações do *framework*, pode dificultar a manutenção de sistemas legados.

1.4.3 MongoDB

O MongoDB (MONGODB, 2023) é um sistema de gerenciamento de banco de dados NoSQL (Not Only SQL), conhecido por sua flexibilidade e capacidade de armazenar e consultar grandes volumes de dados de maneira eficiente. No MongoDB, os dados são organizados em coleções, que são análogas a tabelas em bancos de dados relacionais. Dentro de cada coleção, você armazena documentos, que são os objetos de dados individuais. Os documentos podem ter campos variados e diferentes estruturas.

Por não apresentar um esquema rígido, significa que o usuário pode adicionar ou remover campos em documentos sem afetar os outros documentos na

mesma coleção. Isso é particularmente útil quando os requisitos de dados estão evoluindo constantemente.

Bancos de dados relacionais são conhecidos por sua estrutura rígida, onde qualquer modificação no esquema requer a adição manual de novas colunas na tabela. Todos os objetos armazenados devem seguir a mesma estrutura.

Por outro lado, bancos de dados não relacionais, como o MongoDB, buscam armazenar objetos de forma mais flexível e dinâmica. Essa abordagem é particularmente útil em situações onde a estrutura dos dados pode variar consideravelmente, como em aplicações de Internet das Coisas (IoT), onde novos sensores podem ser incorporados ao sistema sem a necessidade de ajustes na estrutura de armazenamento.

A simplificação na aplicação do sistema ocorre à medida que o desenvolvedor pode focar na arquitetura do software, enquanto o esquema do banco de dados se adapta de forma correspondente.

Como vantagens, o MongoDB tem a sua flexibilidade nas estruturas de dados armazenados, além da escalabilidade e alto desempenho em aplicações bem desenvolvidas. Mas, diferente das bases de dados relacionais, o MongoDB não oferece suporte para transações ACID, existe a necessidade de criar indexações apropriadas para a aplicação e ainda a o paradigma é diferente das bases relacionais, havendo menos profissionais qualificados a utilizar a ferramenta.

1.4.4 Spring

O Spring Framework (SPRING, 2023) é uma ferramenta bem estabelecida no desenvolvimento de aplicativos Java e possui uma ampla comunidade de desenvolvedores, tornando mais fácil encontrar recursos, tutoriais e suporte online.

O Spring é modular e fornece módulos para paradigmas específicos de software. Alguns desses módulos são os seguintes:

- Spring Core: Ferramentas para injeção de dependências, gerenciamento de beans e funcionalidades básicas.
- Spring Data: Oferece serviços de integração com tecnologias como Hibernate para mapeamento objeto-relacional. Nesse trabalho é utilizado o Spring Data MongoDB, que não contém tabelas relacionais, mas sim dados no formato BSON.

- Spring Web: Disponibiliza recursos para o desenvolvimento de aplicativos web no formato MVC (Model-View-Controller). Além de Web Sockets e outras funcionalidades.
- Spring Cloud: Embora não utilizada nessa aplicação, o Spring Cloud contém funcionalidades para o desenvolvimento de sistemas baseados em microserviços.
- Spring Boot: É uma extensão que oferece facilitadores para configurar o ambiente Spring, dando flexibilidade e simplificando as configurações iniciais de uma aplicação.

Todas as ferramentas modulares do Spring fazem dele uma ferramenta extremamente ampla para qualquer tipo de projeto, entretanto, devido a sua estrutura pesada e complexa, o desempenho do Spring pode ser menor que o de tecnologias mais leves.

1.4.5 Typescript

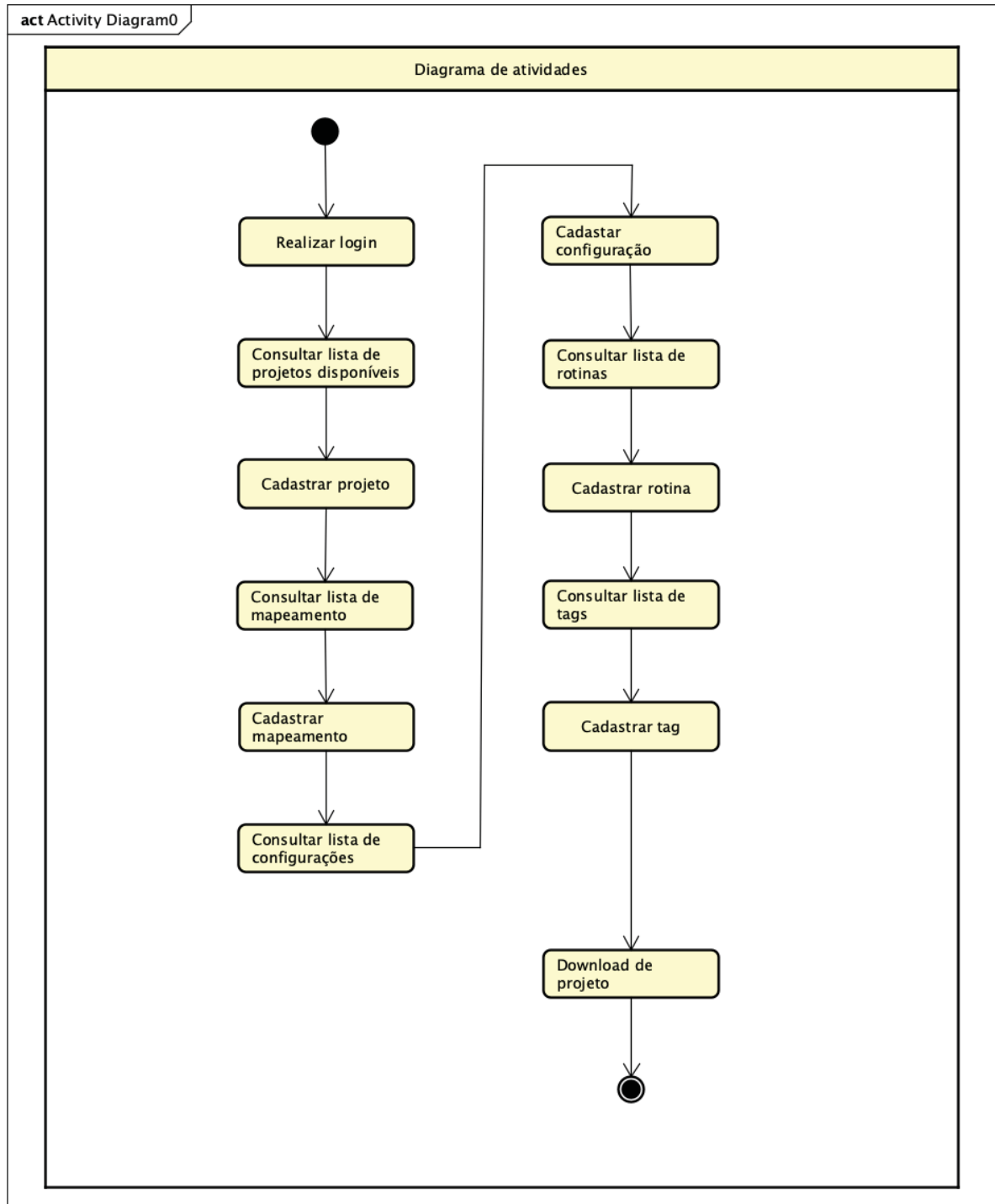
O TypeScript (TYPESCRIPT, 2023) é uma linguagem de programação desenvolvida pela Microsoft que compila para JavaScript, permitindo que os navegadores interpretem e executem o código resultante.

Seu objetivo principal é enriquecer o JavaScript com recursos adicionais, como tipagem estática, suporte a classes e interfaces, com a finalidade de torná-lo mais adequado para o desenvolvimento orientado a objetos e proporcionar maior segurança e estrutura ao código JavaScript. Dessa forma, o TypeScript visa melhorar a qualidade, a eficiência no desenvolvimento de aplicativos web e outras aplicações baseadas em JavaScript.

Embora o Typescript adicione novas funcionalidades ao Javascript, tem-se como desvantagem a necessidade de compilação do código, além de aumentar a complexidade do código.

2 DESENHO DO PROCESSO MACRO

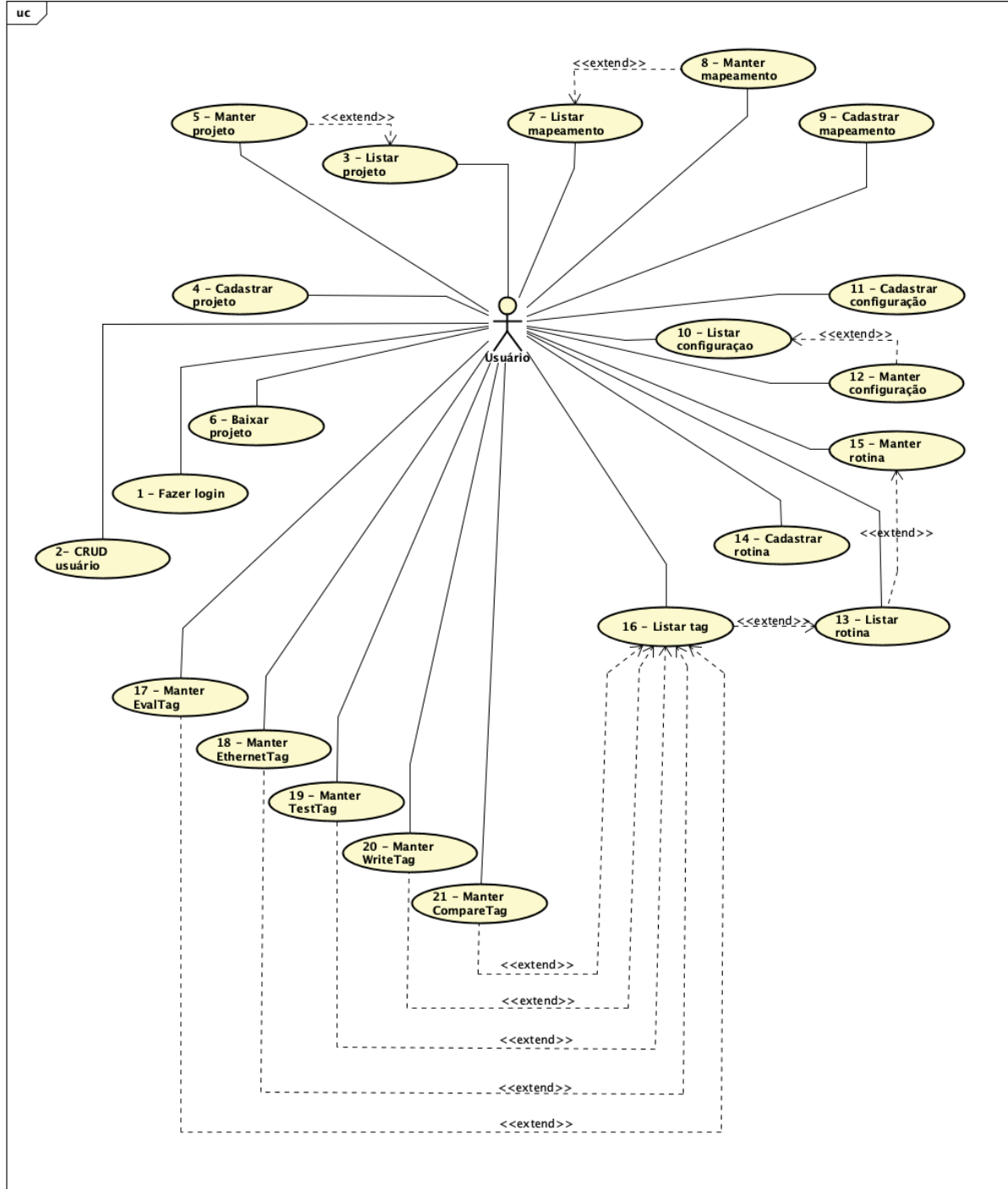
FIGURA 2 – DIAGRAMA DE ATIVIDADES DO DEVELOPMENT TOOL



FONTE: O Autor (2023)

3 DIAGRAMA DE CASOS DE USO

FIGURA 3 – DIAGRAMA DE CASO DE USO



FONTE: O Autor (2023)

4 HISTÓRIAS DE USUÁRIO

HU01 – Fazer login

HU02 – CRUD usuário

HU03 – Listar projeto

HU04 – Cadastrar projeto

HU05 – Manter projeto

HU06 – Baixar projeto

HU07 – Listar mapeamento

HU08 – Manter mapeamento

HU09 – Cadastrar mapeamento

HU10 – Listar configuração

HU11 – Cadastrar configuração

HU12 – Manter configuração

HU13 – Listar rotina

HU14 – Cadastrar rotina

HU15 – Manter rotina

HU16 – Listar tag

HU17 – Manter EvalTag

HU18 – Manter EthernetTag

HU19 – Manter TestTag

HU20 – Manter WriteTag

HU21 – Manter CompareTag

HU01 – Fazer login

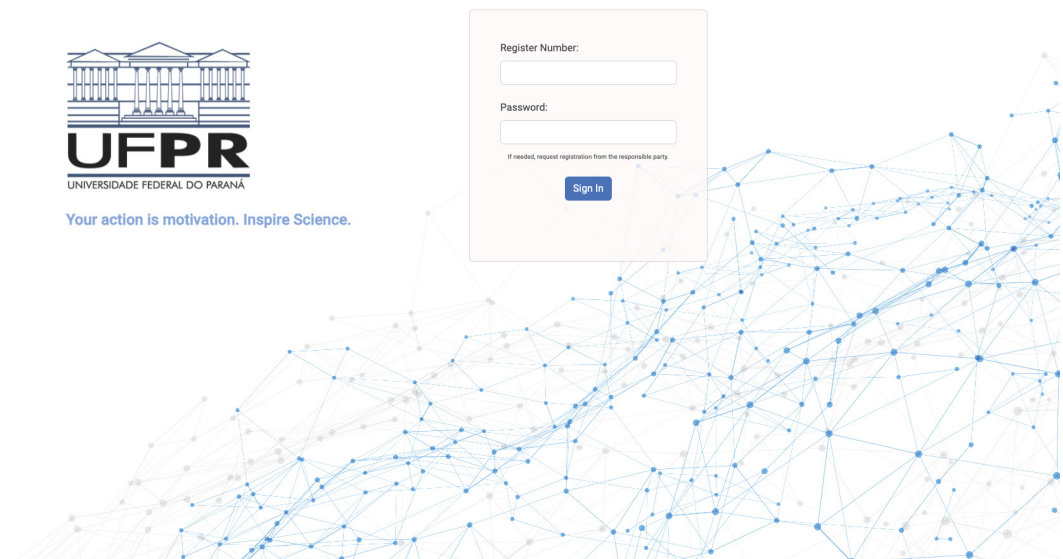
Sendo usuário cadastrado previamente

Quero fazer login

Para utilizar as funcionalidades do aplicativo

DESENHO DA(S) TELA(S)

FIGURA 4 – TELA DE LOGIN



FONTE: O Autor (2023)

FIGURA 5 – TELA INICIAL



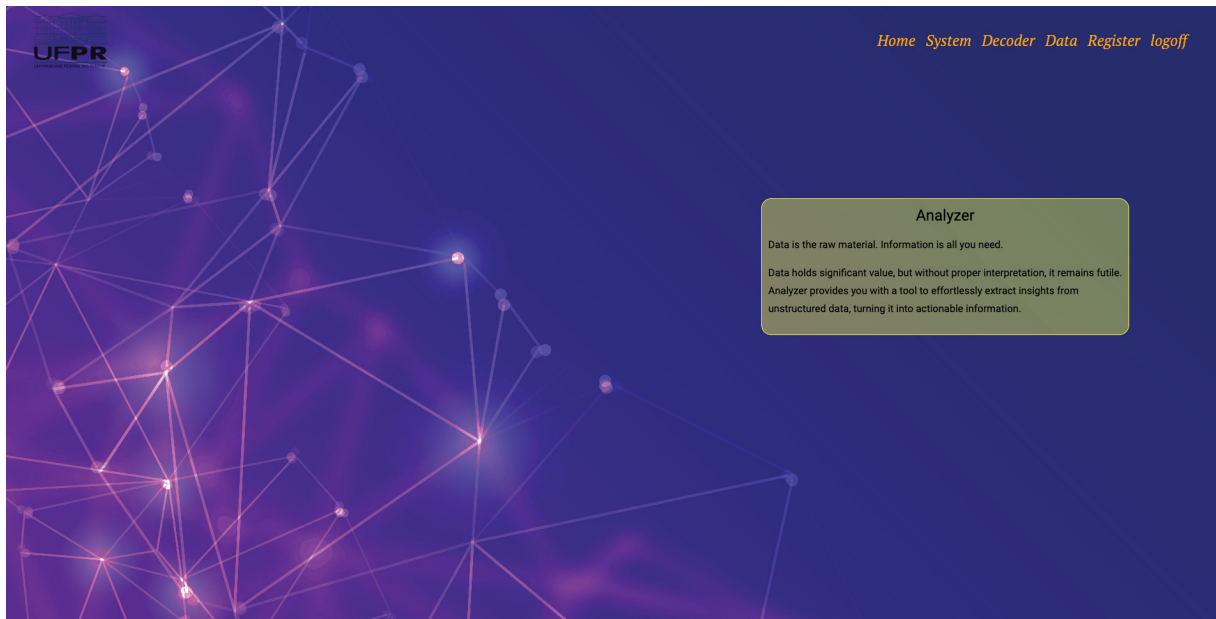
FONTE: O Autor (2023)

FIGURA 6 – TELA REDIRECIONAMENTO DECODER



FONTE: O Autor (2023)

FIGURA 7 – TELA REDIRECIONAMENTO ANALYZER



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve avisar caso login e senha não estejam cadastrados
3. Deve enviar usuário para tela principal caso os dados cadastrados forem preenchidos corretamente e o botão de "login" for apertado

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando apertado o botão de login
Então a apresentar mensagem de campo vazio (R1)
2. **Deve avisar caso login e senha não estejam cadastrados**
Dado que cadastro e senha estejam incorretos
Quando apertado o botão de login
Então apresentar mensagem de campo inválido (R1)

3. Deve enviar usuário para tela principal caso os dados cadastrados forem preenchidos corretamente e o botão de "login" for apertado

Dado que dados sejam preenchidos corretamente

Quando apertado o botão de login

Então redirecionar usuário a tela principal

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 1 – VALIDAÇÃO DE LOGIN

| Situação | Mensagem |
|----------------|-------------------------------|
| Campo vazio | "Campo não preenchido" |
| Campo inválido | "Cadastro ou senha inválido!" |

HU02 – CRUD usuário

Sendo usuário

Quero fazer cadastro e modificação de usuários

Para permitir ou negar usuários utilizarem o aplicativo

DESENHO DA(S) TELA(S)

FIGURA 8 – TELA LISTAGEM USUÁRIO

The screenshot shows a web interface for user management. At the top left is the UFPR logo. The user's current profile is displayed: Name: Edson, Register: 1590, Profile: Engineer. Navigation links include Home, System, Decoder, Data, Register, and logoff. The main section is titled 'User List' and contains a table with the following data:

| ID | Name | Action |
|-------|-------------|----------|
| Id: 2 | New User | + (Add) |
| Id: 0 | Name: Edson | ▼ (Edit) |
| Id: 1 | Name: Odake | ▼ (Edit) |

On the left side, there are filter controls: 'Number of users found: 2', 'ID min' and 'ID max' filters, 'Name' filter, 'Register Number' filter, and 'Profile' filter, all with input fields. A 'Filter' button is located at the bottom left of the filter section.

FONTE: O Autor (2023)

FIGURA 9 – TELA CADASTRO DE USUÁRIO

The screenshot shows a web interface for user management. On the left, there is a sidebar with the UFPR logo and user information: Name: Edson, Register: 1590, Profile: Engineer. Below this are filter options for ID (min/max), Name, Register Number, and Profile, with a 'Filter' button. The main content area is titled 'User List' and contains a 'New User' form and a table of existing users.

User List

Home System Decoder Data Register logoff

Name: Edson
Register: 1590
Profile: Engineer

Number of users found: 2

ID min: Filter id min | ID max: Filter id max

Nome: Filter name

Register Number: Filter cadastro

Profile: [input field]

Filter

New User [Add] [Cancel]

| ID | Name | Register Number |
|-------|-------------|-----------------|
| Id: 0 | Name: Edson | |
| Id: 1 | Name: Odake | |

FONTE: O Autor (2023)

FIGURA 10 – TELA ATUALIZAÇÃO DE USUÁRIO

The screenshot shows the same web interface as Figure 9, but with the 'Update User' form selected. The form is pre-filled with the details of the user with ID 0: Name: Edson, Register Number: 1590, Profile: Engineer. The 'Update' button is highlighted in red.

User List

Home System Decoder Data Register logoff

Name: Edson
Register: 1590
Profile: Engineer

Number of users found: 2

ID min: Filter id min | ID max: Filter id max

Nome: Filter name

Register Number: Filter cadastro

Profile: [input field]

Filter

Update User [Update] [Cancel] [Remove]

| ID | Name | Register Number |
|-------|-------------|-----------------|
| Id: 0 | Name: Edson | 1590 |
| Id: 1 | Name: Odake | |

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve permitir adicionar novo usuário
2. Deve permitir remover usuário
3. Deve permitir atualizar informação de usuário
4. Deve listar usuários

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve permitir adicionar novo usuário

Dado que campos estejam preenchidos

Quando pressiona botão de "Add"

Então cadastro novo usuário

2. Deve permitir remover usuário

Dado que usuário esteja cadastrado

Quando pressiona botão de "Remove"

Então usuário é removido

3. Deve permitir atualizar informação de usuário

Dado que campos de usuário sejam modificados

Quando pressiona botão de "Update"

Então o cadastro é atualizado com os novos valores

4. Deve listar usuários

Dado que existam usuários cadastrados

Quando tela de cadastro estiver acionada

Então aparece uma lista de todos os usuários

HU03 – Listar projeto

Sendo usuário

Quero ter acesso a lista de projetos

Para visualizar projetos cadastrados

DESENHO DA(S) TELA(S)

FIGURA 11 – TELA LISTA DE PROJETOS



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve listar projetos cadastrados
2. Deve abrir pop-up com informações do projeto

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve listar projetos cadastrados**
Dado que haja projeto cadastrado
Quando tela de listagem de projeto for acionada
Então lista com nome dos projetos é apresentada
2. **Deve abrir pop-up com informações do projeto**
Quando botão para abrir projeto for pressionado
Então abre tela com informações do projeto

HU04 – Cadastrar projeto

Sendo usuário

Quero cadastrar novo projeto

Para poder manter as informações

DESENHO DA(S) TELA(S)

FIGURA 12 – TELA CADASTRO DE PROJETOS

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Close"

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando botão para salvar projeto for pressionado
Então a apresentar mensagem de campo vazio (R1)
2. **Deve salvar dados ao apertar botão de "Save"**
Dado que campos estejam preenchidos
Quando botão para salvar projeto for pressionado
Então os dados são salvos
E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de projeto esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 2 – VALIDAÇÃO DE FORMULÁRIO

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU05 – Manter projeto

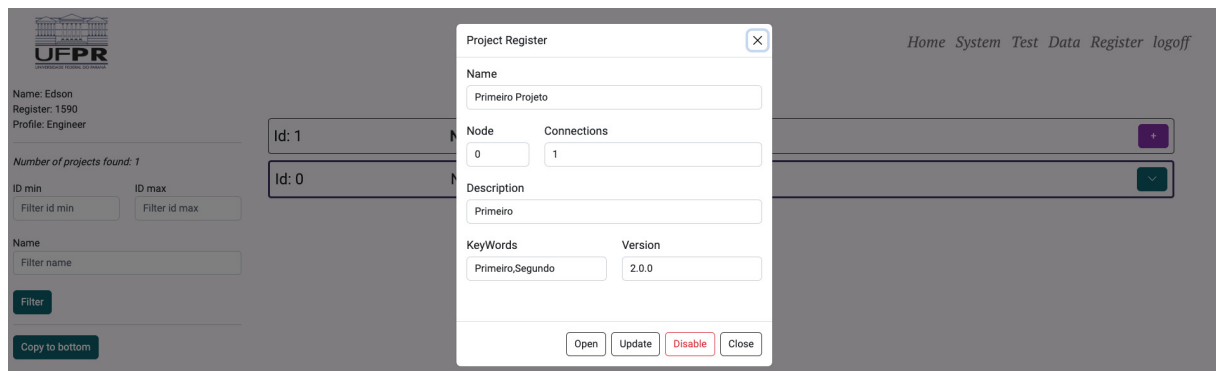
Sendo usuário

Quero atualizar ou remover projeto

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 13 – TELA ATUALIZAÇÃO DE PROJETOS



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve mostrar dados do projeto
2. Deve atualizar dados do projeto
3. Deve remover projeto

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve mostrar dados do projeto

Quando tela do projeto for acionada

Então mostrar dados do projeto

2. Deve atualizar dados do projeto

Dado que dados do projeto forem modificados

Quando botão "Update" for pressionado

Então dados do projeto são atualizados

E então usuário é redirecionado a listagem de projetos

3. Deve remover projeto

Dado que pop-up de projeto esteja aberto

Quando botão "Remove" for pressionado

Então projeto é removido

E então usuário é redirecionado a listagem de projetos

HU06 – Baixar projeto

Sendo usuário

Quero baixar arquivos de projeto

Para poder utilizar em outra aplicação

DESENHO DA(S) TELA(S)

FIGURA 14 – TELA DOWNLOAD DE PROJETOS

UFPR

Home System Test Data Register logoff

Name: Edson
Register: 1590
Profile: Engineer

Number of Mappings: 1
Number of Routines: 1
Number of Configurations: 1

Project 0

Mappings

Routines

Configurations

Download

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve apresentar botão de download

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve apresentar botão de download

Quando botão "Download" for pressionado

Então arquivos do projeto são armazenados em pasta local

HU07 – Listar mapeamento

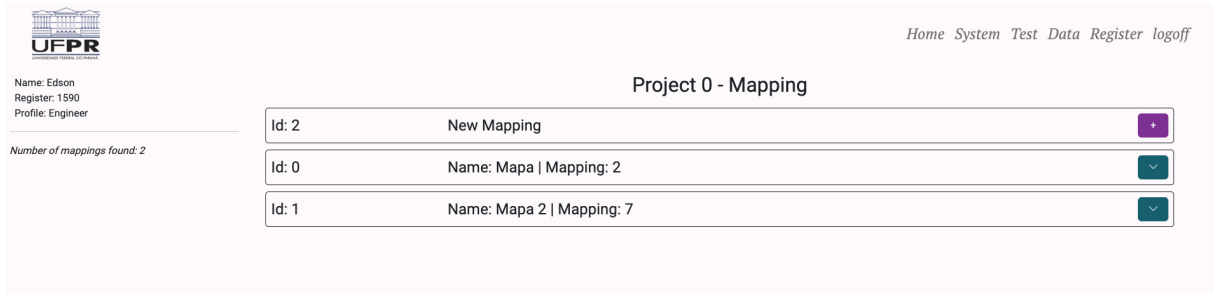
Sendo usuário

Quero ter acesso a lista de mapeamento

Para visualizar mapeamento cadastrados

DESENHO DA(S) TELA(S)

FIGURA 15 – TELA LISTA DE MAPEAMENTO



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve listar mapeamentos cadastrados
2. Deve abrir pop-up com informações do mapeamento

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve listar mapeamento cadastrados

Dado que haja mapeamento cadastrado

Quando tela de listagem de mapeamento for acionada

Então lista com nome dos mapeamentos é apresentada

2. Deve abrir pop-up com informações do mapeamento

Quando botão para abrir mapeamento for pressionado

Então abre tela com informações do mapeamento

HU08 – Manter mapeamento

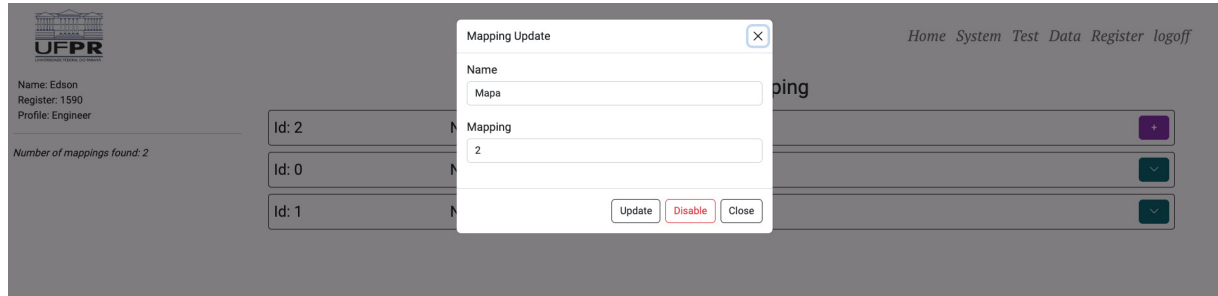
Sendo usuário

Quero atualizar ou remover mapeamento

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 16 – TELA ATUALIZAÇÃO DE MAPEAMENTO



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve mostrar dados do mapeamento
2. Deve atualizar dados do mapeamento
3. Deve remover mapeamento

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve mostrar dados do mapeamento**
Dado que hajam mapeamentos cadastrados
Quando tela do mapeamento for acionada
Então mostrar dados do mapeamento
2. **Deve atualizar dados do mapeamento**
Dado que dados do mapeamento forem modificados
Quando botão "Update" for pressionado
Então dados do mapeamento são atualizados
E então usuário é redirecionado a listagem de mapeamentos
3. **Deve remover mapeamento**
Dado que pop-up de mapeamento esteja aberto
Quando botão "Remove" for pressionado
Então mapeamento é removido

E então usuário é redirecionado a listagem de mapeamentos

HU09 – Cadastrar mapeamento

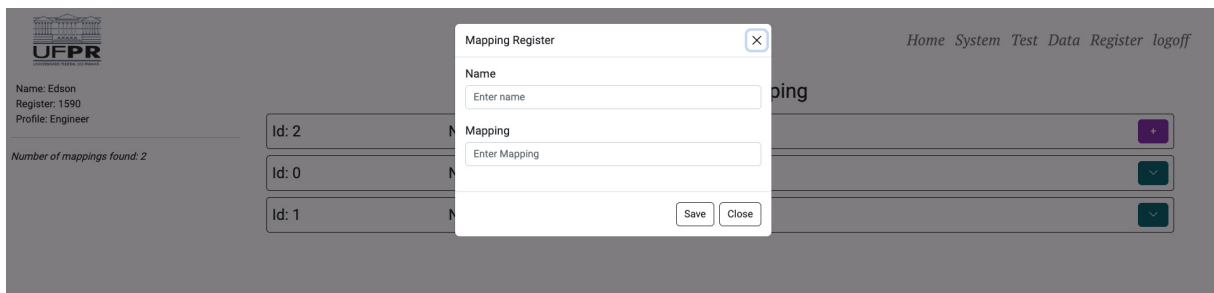
Sendo usuário

Quero cadastrar novo mapeamento

Para poder manter as informações

DESENHO DA(S) TELA(S)

FIGURA 17 – TELA CADASTRO DE MAPEAMENTO



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando botão para salvar mapeamento for pressionado
Então a apresentar mensagem de campo vazio (R1)
2. **Deve salvar dados ao apertar botão de "Save"**
Dado que campos estejam preenchidos

Quando botão para salvar mapeamento for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de cadastro de mapeamento esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 3 – VALIDAÇÃO DE FORMULÁRIO MAPEAMENTO

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU10 – Listar configuração

Sendo usuário

Quero ter acesso a lista de configurações

Para visualizar configurações cadastrados

DESENHO DA(S) TELA(S)

FIGURA 18 – TELA GERAL DE CONFIGURAÇÃO

FONTE: O Autor (2023)

FIGURA 19 – TELA LISTA DE CONFIGURAÇÃO

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve listar configuração cadastrados
2. Deve abrir pop-up com informações do configuração

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve listar configuração cadastrados

Dado que haja configuração cadastrada

Quando tela de listagem de configuração for acionada

Então lista com nome das configurações é apresentada

2. Deve abrir pop-up com informações da configuração
Quando botão para abrir configuração for pressionado
Então abre tela com informações da configuração

HU11 – Manter configuração

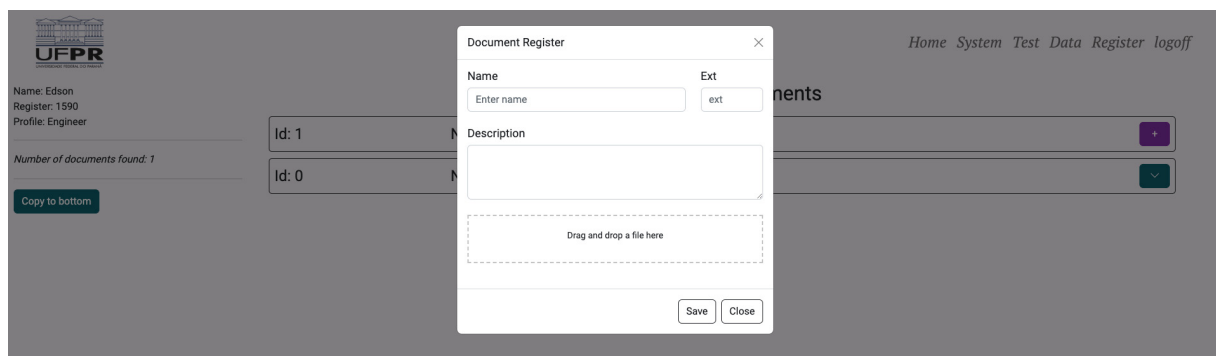
Sendo usuário

Quero atualizar ou remover configuração

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 20 – TELA CADASTRO DE CONFIGURAÇÃO



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve mostrar dados da configuração
2. Deve atualizar dados da configuração
3. Deve remover configuração

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve mostrar dados da configuração**
Quando tela da configuração for acionada
Então mostrar dados da configuração

2. Deve atualizar dados da configuração

Dado que dados de configuração forem modificados

Quando botão "Update" for pressionado

Então dados da configuração são atualizados

E então usuário é redirecionado a listagem de configurações

3. Deve remover configuração

Dado que pop-up de configuração esteja aberto

Quando botão "Remove" for pressionado

Então configuração é removida

E então usuário é redirecionado a listagem de configurações

HU12 – Cadastrar configuração

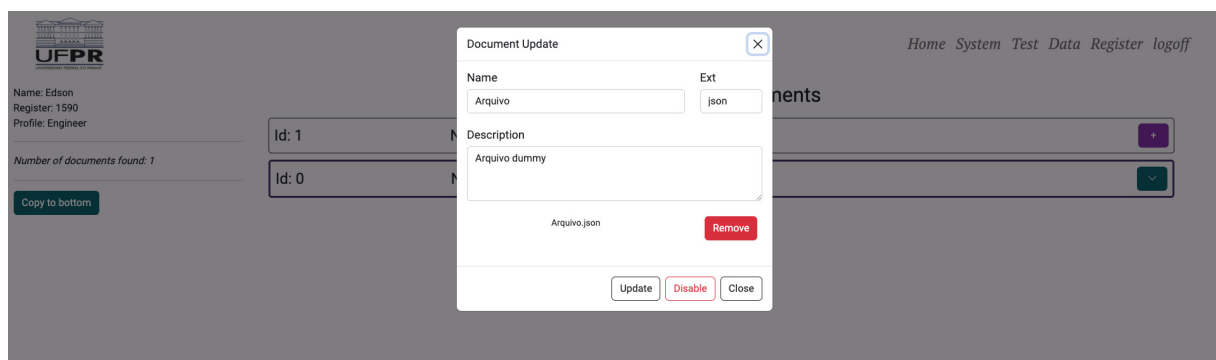
Sendo usuário

Quero cadastrar nova configuração

Para poder manter as informações

DESENHO DA(S) TELA(S)

FIGURA 21 – TELA ATUALIZAÇÃO DE CONFIGURAÇÃO



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

4. Deve avisar caso campo não esteja preenchido

Dado que campo não seja preenchido

Quando botão para salvar configuração for pressionado

Então a apresentar mensagem de campo vazio (R1)

5. Deve salvar dados ao apertar botão de "Save"

Dado que campos estejam preenchidos

Quando botão para salvar configuração for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

6. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de configuração esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 4 – VALIDAÇÃO DE FORMULÁRIO CONFIGURAÇÃO

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU13 – Listar rotina

Sendo usuário

Quero ter acesso a lista de rotina

Para visualizar rotina cadastrados

DESENHO DA(S) TELA(S)

FIGURA 22 – TELA LISTA DE ROTINAS

UFPR

Home System Test Data Register logoff

Name: Edson
Register: 1590
Profile: Engineer

Project 0 - Routines

| Id | Name |
|-------|----------------|
| Id: 2 | Name: |
| Id: 0 | Name: Rotina 1 |
| Id: 1 | Name: Rotina 2 |

Number of routines found: 2

ID min: 0 ID max: 1

Name: Filter name

Filter

Copy to bottom

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve listar rotinas cadastrados
2. Deve abrir pop-up com informações da rotina

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve listar rotinas cadastrados**
Dado que haja rotina cadastrada
Quando tela de listagem de rotina for acionada
Então lista com nome das rotinas é apresentada
2. **Deve abrir pop-up com informações do mapeamento**
Quando botão para abrir rotina for pressionado
Então abre tela com informações da rotina

HU14 – Manter rotina

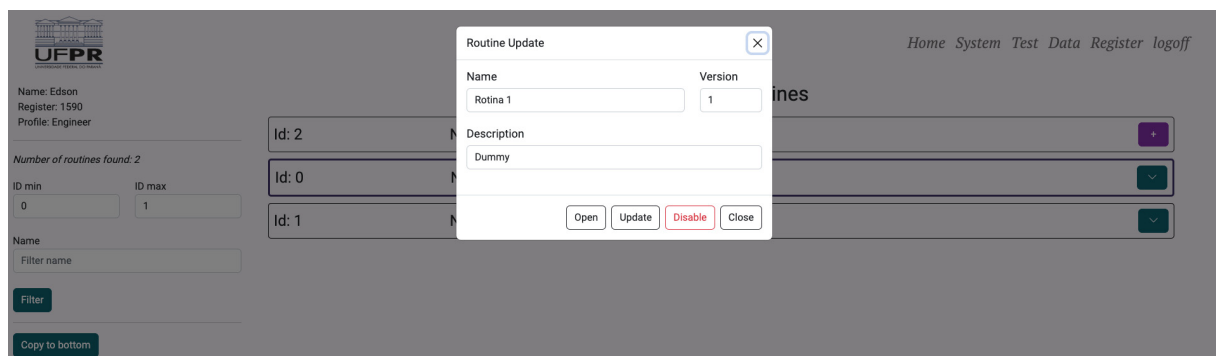
Sendo usuário

Quero atualizar ou remover rotina

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 23 – TELA ATUALIZAÇÃO DE ROTINAS



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve mostrar dados da rotina
2. Deve atualizar dados da rotina
3. Deve remover rotina

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve mostrar dados da rotina**

Quando tela da rotina for acionada

Então mostrar dados da rotina

2. **Deve atualizar dados da rotina**

Dado que dados do rotina forem modificados

Quando botão "Update" for pressionado

Então dados da rotina são atualizados

E então usuário é redirecionado a listagem de rotinas

3. Deve remover rotina

Dado que pop-up de rotina esteja aberto

Quando botão "Remove" for pressionado

Então mapeamento é removido

E então usuário é redirecionado a listagem de rotinas

HU15 – Cadastrar rotina

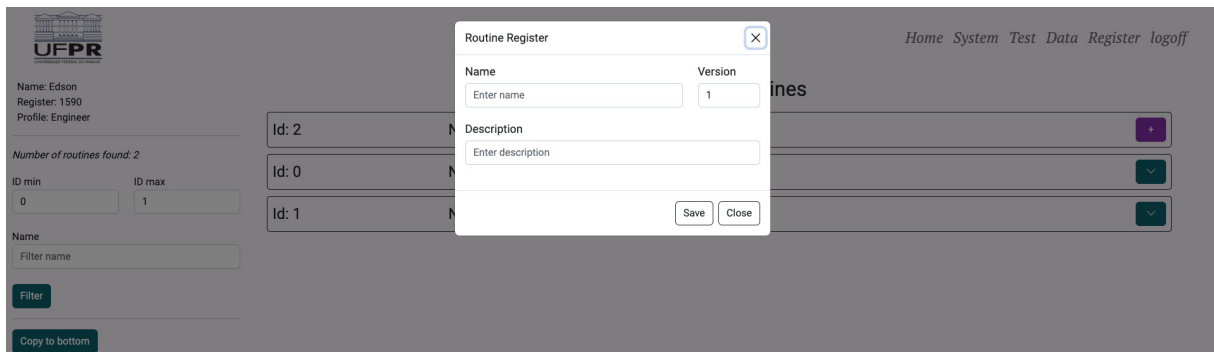
Sendo usuário

Quero cadastrar nova rotina

Para poder manter as informações

DESENHO DA(S) TELA(S)

FIGURA 24 – TELA CADASTRO DE ROTINAS



FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando botão para salvar rotina for pressionado
Então a apresentar mensagem de campo vazio (R1)

2. **Deve salvar dados ao apertar botão de "Save"**
Dado que campos estejam preenchidos
Quando botão para salvar rotina for pressionado
Então os dados são salvos
E então o usuário é redirecionado a tela de listagem

3. **Deve voltar para tela de listagem ao apertar "Cancel"**
Dado que pop-up de rotina esteja aberto
Quando botão "cancelar" for pressionado
Então o usuário é redirecionado a tela de listagem

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 5 – VALIDAÇÃO DE FORMULÁRIO ROTINA

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU16 – Listar tag

Sendo usuário

Quero ter acesso a lista de tag

Para visualizar tag cadastradas

DESENHO DA(S) TELA(S)

FIGURA 25 – TELA LISTA DE ETAPAS (TAGS)

| Id | Tag Name | Action |
|-------|--|--------|
| Id: 0 | Teste de HIPOT | ▼ |
| Id: 1 | Setup de comunicação com o PLC | ▼ |
| Id: 2 | Aciona contator KTA_CONT | ▼ |
| Id: 3 | Verificação de Curto Circuito | ▼ |
| Id: 4 | Desaciona contator KTA_CONT | ▼ |
| Id: 5 | Aciona contator KTA | ▼ |
| Id: 6 | Aplicação de alta tensão | ▼ |
| Id: 7 | Desaciona contator KTA | ▼ |
| Id: 8 | New Tag: <input type="text" value="LeafWriteTag"/> | + |

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve listar tags cadastradas
2. Deve abrir pop-up com informações do tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve listar tags cadastradas**
Dado que haja tag cadastrada
Quando tela de listagem de tag for acionada
Então lista com nome das tags é apresentada
2. **Deve abrir pop-up com informações da tag**
Quando botão para abrir tag for pressionado
Então abre tela com informações da tag

HU17 – Manter EvalTag

Sendo usuário

Quero cadastrar, atualizar ou remover EvalTag

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 26 – TELA ATUALIZAÇÃO DE EVAL TAG

The image shows a 'Tag Update' dialog box with the following fields and values:

- Tag name:** LeafEvalSharedTag
- Position:** 1
- Simple description:** teste
- Error message:** (empty)
- Tip:** (empty)
- Inline failure code:** 1
- Description:** (empty text area)
- Variavel de resultado:** fx
- Equação:** $15 * (\text{valorMedido}) + 1$

At the bottom of the dialog box, there are four buttons: 'Advanced Fields', 'Update', 'Disable', and 'Close'.

FONTE: O Autor (2023)

FIGURA 27 – TELA CADASTRO DE EVAL TAG

The image shows a 'Tag Register' dialog box with the following fields and controls:

- Tag name:** Input field containing 'LeafEvalSharedTag'.
- Position:** Input field containing '0'.
- Simple description:** Text area.
- Error message:** Input field.
- Tip:** Input field.
- Inline failure code:** Text area.
- Description:** Text area.
- Variavel de resultado:** Input field.
- Equação:** Input field.
- Buttons:** 'Advanced Fields', 'Save', and 'Close'.

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"
4. Deve mostrar dados da tag
5. Deve atualizar dados da tag
6. Deve remover tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

- 1. Deve avisar caso campo não esteja preenchido**
 - Dado que** campo não seja preenchido
 - Quando** botão para salvar mapeamento for pressionado
 - Então** a apresentar mensagem de campo vazio (R1)

2. Deve salvar dados ao apertar botão de "Save"

Dado que campos estejam preenchidos

Quando botão para salvar tag for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de tag esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

4. Deve mostrar dados da tag

Quando tela da tag for acionada

Então mostrar dados da tag

5. Deve atualizar dados da tag

Dado que dados de tag forem modificados

Quando botão "Update" for pressionado

Então dados da tag são atualizados

E então usuário é redirecionado a listagem de tags

6. Deve remover tag

Dado que pop-up de tag esteja aberto

Quando botão "Remove" for pressionado

Então tag é removido

E então usuário é redirecionado a listagem de tags

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 6 – VALIDAÇÃO DE FORMULÁRIO EVALTAG

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU18 – Manter EthernetTag

Sendo usuário

Quero cadastrar, atualizar ou remover EthernetTag

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 28 – TELA ATUALIZAÇÃO DE ETHERNET TAG

The screenshot shows a 'Tag Update' dialog box with the following fields and values:

- Tag name: LeafEthernetCommunicationTag
- Position: 0
- Simple description: comunica com algo
- Error message: (empty)
- Tip: (empty)
- Inline failure code: 1
- Description: (empty text area)
- Communication name: ADS
- Ip: 123.123.123.123
- Trials: 3
- Address: 123
- Time between: 100
- Port: 123

Buttons at the bottom: Advanced Fields, Update, Disable, Close.

FONTE: O Autor (2023)

FIGURA 29 – TELA CADASTRO DE ETHERNET TAG

The image shows a 'Tag Register' window with the following fields and values:

- Tag name: LeafEthernetCommunicationTag
- Position: 0
- Simple description: (empty)
- Error message: (empty)
- Tip: (empty)
- Inline failure code: (empty)
- Description: (empty)
- Communication name: (empty)
- Ip: (empty)
- Trials: 3
- Address: (empty)
- Time between: 100
- Port: (empty)

Buttons at the bottom: Advanced Fields, Save, Close.

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"
4. Deve mostrar dados da tag
5. Deve atualizar dados da tag
6. Deve remover tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando botão para salvar mapeamento for pressionado
Então a apresentar mensagem de campo vazio (R1)

2. Deve salvar dados ao apertar botão de "Save"

Dado que campos estejam preenchidos

Quando botão para salvar tag for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de tag esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

4. Deve mostrar dados da tag

Quando tela da tag for acionada

Então mostrar dados da tag

5. Deve atualizar dados da tag

Dado que dados de tag forem modificados

Quando botão "Update" for pressionado

Então dados da tag são atualizados

E então usuário é redirecionado a listagem de tags

6. Deve remover tag

Dado que pop-up de tag esteja aberto

Quando botão "Remove" for pressionado

Então tag é removido

E então usuário é redirecionado a listagem de tags

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 7 – VALIDAÇÃO DE FORMULÁRIO ETHERNETTAG

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU19 – Manter TestTag

Sendo usuário

Quero cadastrar, atualizar ou remover TestTag

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 30 – TELA ATUALIZAÇÃO DE TEST TAG

The image shows a 'Tag Update' dialog box with the following fields and values:

- Tag name:** LeafTestTag
- Position:** 1
- Simple description:** Teste
- Error message:** (empty)
- Tip:** (empty)
- Inline failure code:** 1
- Description:** (empty text area)
- Test name:** Testando

At the bottom of the dialog, there are four buttons: 'Advanced Fields', 'Update', 'Disable', and 'Close'.

FONTE: O Autor (2023)

FIGURA 31 – TELA CADASTRO DE TEST TAG

The image shows a 'Tag Register' dialog box with the following fields and controls:

- Tag name:** Input field containing 'LeafTestTag'.
- Position:** Input field containing '0'.
- Simple description:** Input field.
- Error message:** Input field.
- Tip:** Input field.
- Inline failure code:** Input field.
- Description:** Large text area.
- Test name:** Input field.
- Buttons:** 'Advanced Fields', 'Save', and 'Close'.

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"
4. Deve mostrar dados da tag
5. Deve atualizar dados da tag
6. Deve remover tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
 - Dado que** campo não seja preenchido
 - Quando** botão para salvar mapeamento for pressionado
 - Então** a apresentar mensagem de campo vazio (R1)
2. **Deve salvar dados ao apertar botão de "Save"**
 - Dado que** campos estejam preenchidos
 - Quando** botão para salvar tag for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de tag esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

4. Deve mostrar dados da tag

Quando tela da tag for acionada

Então mostrar dados da tag

5. Deve atualizar dados da tag

Dado que dados de tag forem modificados

Quando botão "Update" for pressionado

Então dados da tag são atualizados

E então usuário é redirecionado a listagem de tags

6. Deve remover tag

Dado que pop-up de tag esteja aberto

Quando botão "Remove" for pressionado

Então tag é removido

E então usuário é redirecionado a listagem de tags

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 8 – VALIDAÇÃO DE FORMULÁRIO TESTTAG

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU20 – Manter WriteTag

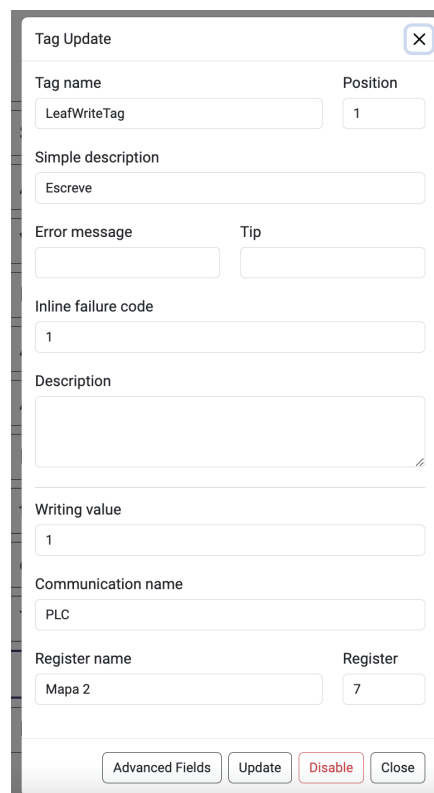
Sendo usuário

Quero cadastrar, atualizar ou remover WriteTag

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 32 – TELA ATUALIZAÇÃO DE WRITE TAG



The image shows a 'Tag Update' dialog box with the following fields and values:

- Tag name:** LeafWriteTag
- Position:** 1
- Simple description:** Escreve
- Error message:** (empty)
- Tip:** (empty)
- Inline failure code:** 1
- Description:** (empty text area)
- Writing value:** 1
- Communication name:** PLC
- Register name:** Mapa 2
- Register:** 7

At the bottom of the dialog, there are four buttons: 'Advanced Fields', 'Update', 'Disable', and 'Close'.

FONTE: O Autor (2023)

FIGURA 33 – TELA CADASTRO DE WRITE TAG

The image shows a 'Tag Register' form with the following fields and values:

- Tag name: LeafWriteTag
- Position: 0
- Simple description: (empty)
- Error message: (empty)
- Tip: (empty)
- Inline failure code: (empty)
- Description: (empty)
- Writing value: (empty)
- Communication name: (empty)
- Register name: (empty)
- Register: (empty)

Buttons at the bottom: Advanced Fields, Save, Close.

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"
4. Deve mostrar dados da tag
5. Deve atualizar dados da tag
6. Deve remover tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. **Deve avisar caso campo não esteja preenchido**
Dado que campo não seja preenchido
Quando botão para salvar mapeamento for pressionado
Então a apresentar mensagem de campo vazio (R1)

2. Deve salvar dados ao apertar botão de "Save"

Dado que campos estejam preenchidos

Quando botão para salvar tag for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de tag esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

4. Deve mostrar dados da tag

Quando tela da tag for acionada

Então mostrar dados da tag

5. Deve atualizar dados da tag

Dado que dados de tag forem modificados

Quando botão "Update" for pressionado

Então dados da tag são atualizados

E então usuário é redirecionado a listagem de tags

6. Deve remover tag

Dado que pop-up de tag esteja aberto

Quando botão "Remove" for pressionado

Então tag é removido

E então usuário é redirecionado a listagem de tags

REGRAS DE NEGÓCIO DA HISTÓRIA

R1 – Dados incorretos

TABELA 9 – VALIDAÇÃO DE FORMULÁRIO WRITETAG

| Situação | Mensagem |
|-------------|------------------------|
| Campo vazio | "Campo não preenchido" |

HU21 – Manter CompareTag

Sendo usuário

Quero cadastrar, atualizar ou remover CompareTag

Para poder atualizar as informações

DESENHO DA(S) TELA(S)

FIGURA 34 – TELA ATUALIZAÇÃO DE COMPARE TAG

The screenshot shows a 'Tag Update' dialog box with the following fields and values:

- tagName:** LeafRegisterCompareTag
- Position:** 1
- Simple description:** Leitura
- Error message:** (empty)
- Tip:** (empty)
- Inline failure code:** 1
- Description:** (empty text area)
- Reference communication name:** ADS
- Reference register name:** Mapa 3
- Ref. register:** 34
- OnTest communication name:** PLC
- OnTest register name:** Mapa4
- OnTest reg.:** 1232
- Tolerancy:** 10
- Calculate by:** Percentage

At the bottom of the form, there are four buttons: 'Advanced Fields', 'Update', 'Disable', and 'Close'.

FONTE: O Autor (2023)

FIGURA 35 – TELA CADASTRO DE COMPARE TAG

Tag Register

Tag name: LeafRegisterCompareTag Position: 0

Simple description

Error message Tip

Inline failure code

Description

Reference communication name

Reference register name Ref. register

OnTest communication name

OnTest register name OnTest reg.

Tolerancy: 10 Calculate by: Percentage

Advanced Fields Save Close

FONTE: O Autor (2023)

CRITÉRIOS DE ACEITAÇÃO

1. Deve avisar caso campo não esteja preenchido
2. Deve salvar dados ao apertar botão de "Save"
3. Deve voltar para tela de listagem ao apertar "Cancel"
4. Deve mostrar dados da tag
5. Deve atualizar dados da tag
6. Deve remover tag

CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

1. Deve avisar caso campo não esteja preenchido

Dado que campo não seja preenchido

Quando botão para salvar mapeamento for pressionado

Então a apresentar mensagem de campo vazio (R1)

2. Deve salvar dados ao apertar botão de "Save"

Dado que campos estejam preenchidos

Quando botão para salvar tag for pressionado

Então os dados são salvos

E então o usuário é redirecionado a tela de listagem

3. Deve voltar para tela de listagem ao apertar "Cancel"

Dado que pop-up de tag esteja aberto

Quando botão "Cancel" for pressionado

Então o usuário é redirecionado a tela de listagem

4. Deve mostrar dados da tag

Quando tela da tag for acionada

Então mostrar dados da tag

5. Deve atualizar dados da tag

Dado que dados de tag forem modificados

Quando botão "Update" for pressionado

Então dados da tag são atualizados

E então usuário é redirecionado a listagem de tags

6. Deve remover tag

Dado que pop-up de tag esteja aberto

Quando botão "Remove" for pressionado

Então tag é removido

E então usuário é redirecionado a listagem de tags

REGRAS DE NEGÓCIO DA HISTÓRIA

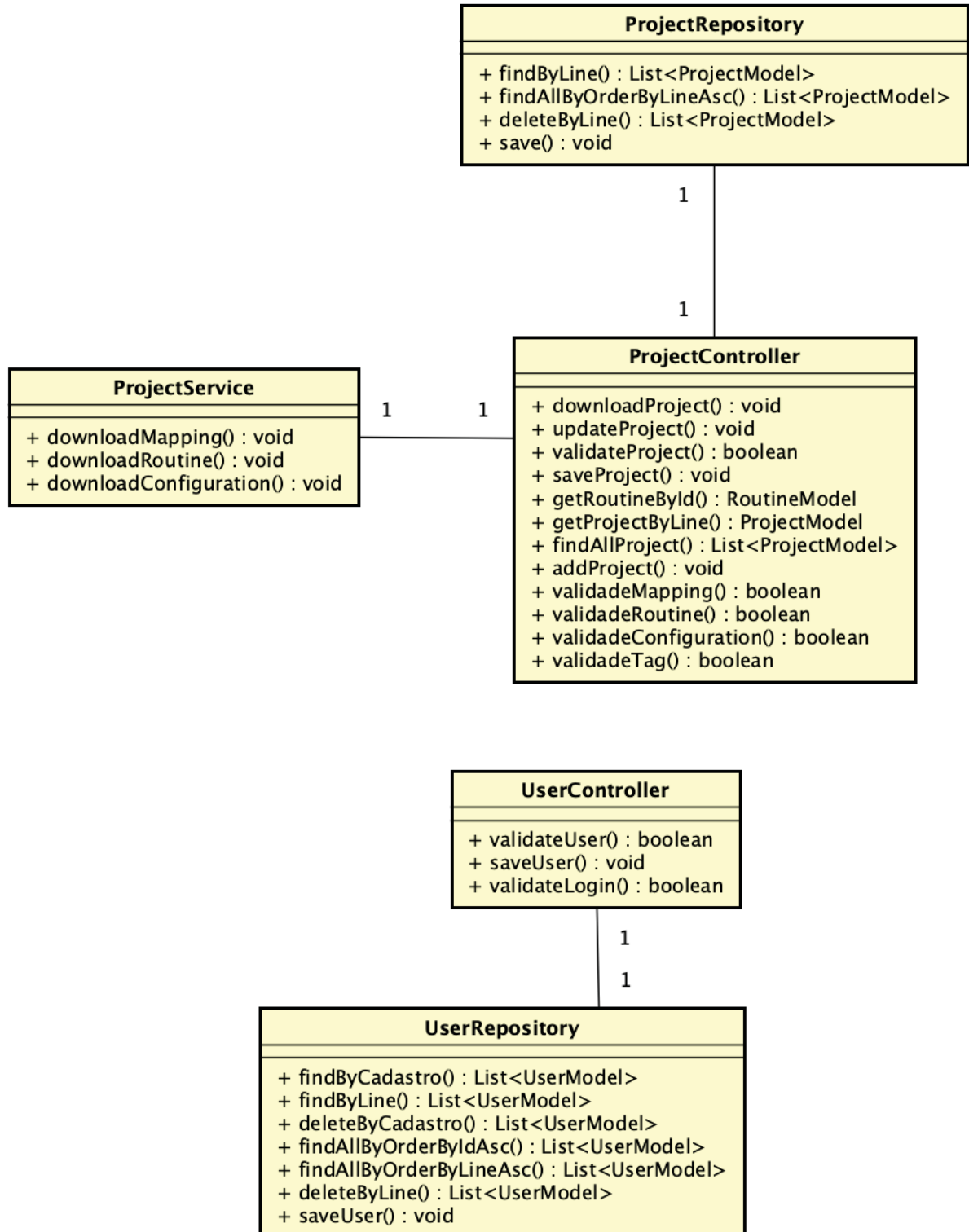
R1 – Dados incorretos

TABELA 10 – VALIDAÇÃO DE FORMULÁRIO REGISTERCOMPARETAG

| Situação | Mensagem |
|-----------------|------------------------|
| Campo vazio | "Campo não preenchido" |

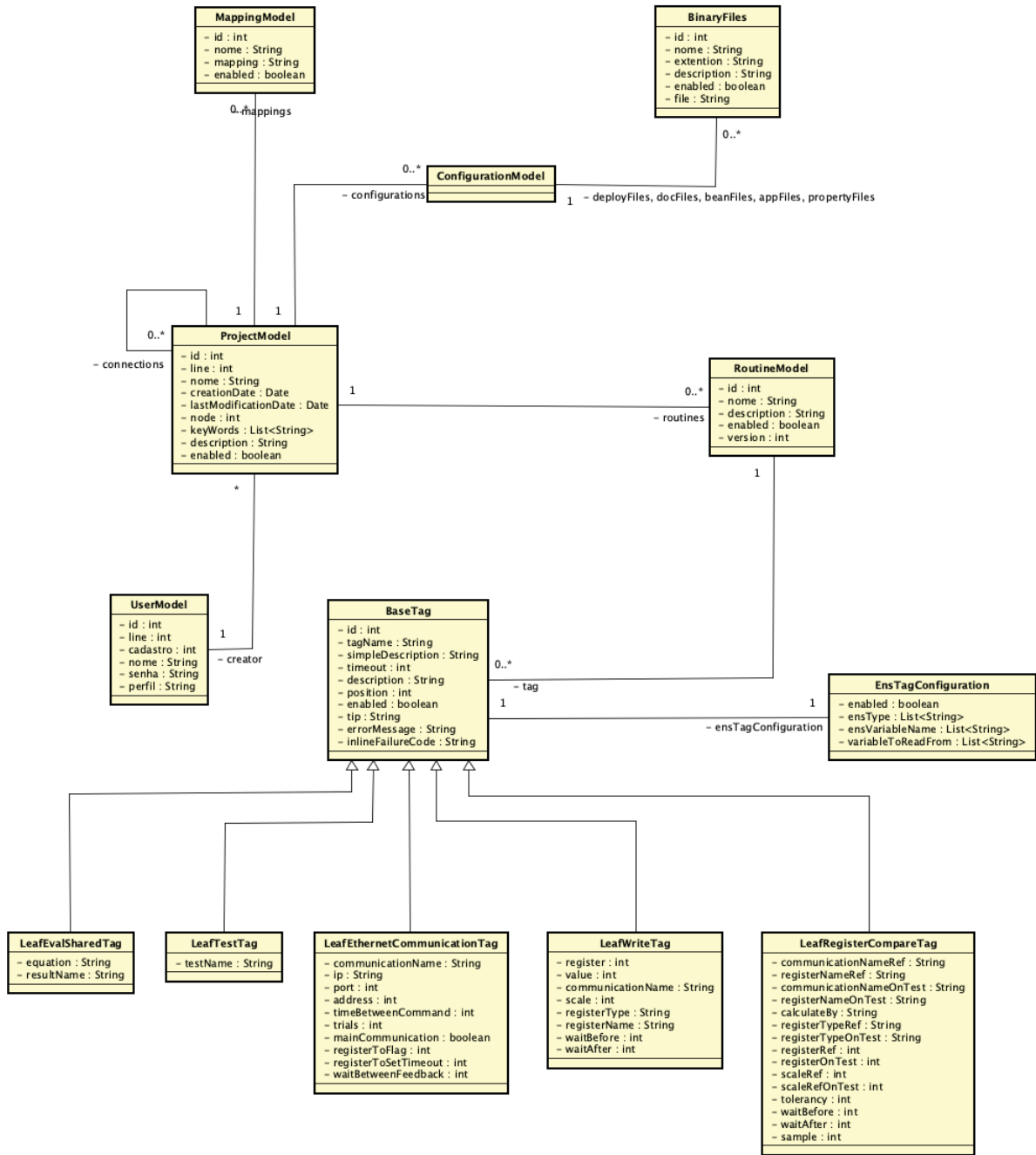
5 DIAGRAMA DE CLASSES

FIGURA 36 – DIAGRAMA DE CLASSE DE IMPLEMENTAÇÃO



FONTE: O Autor (2023)

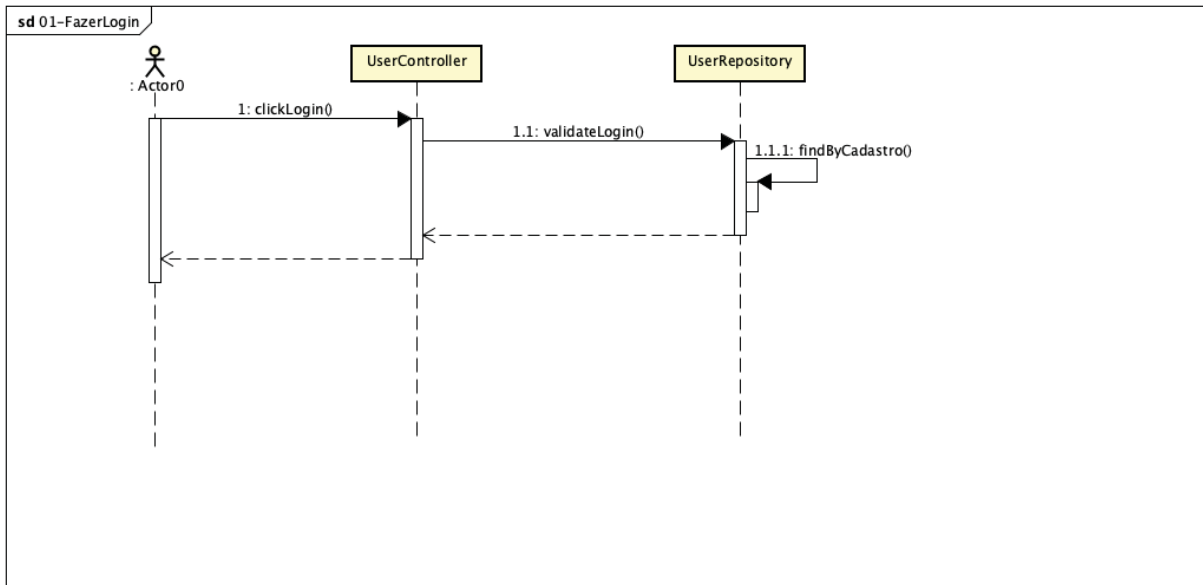
FIGURA 37 – DIAGRAMA DE CLASSE



FONTE: O Autor (2023)

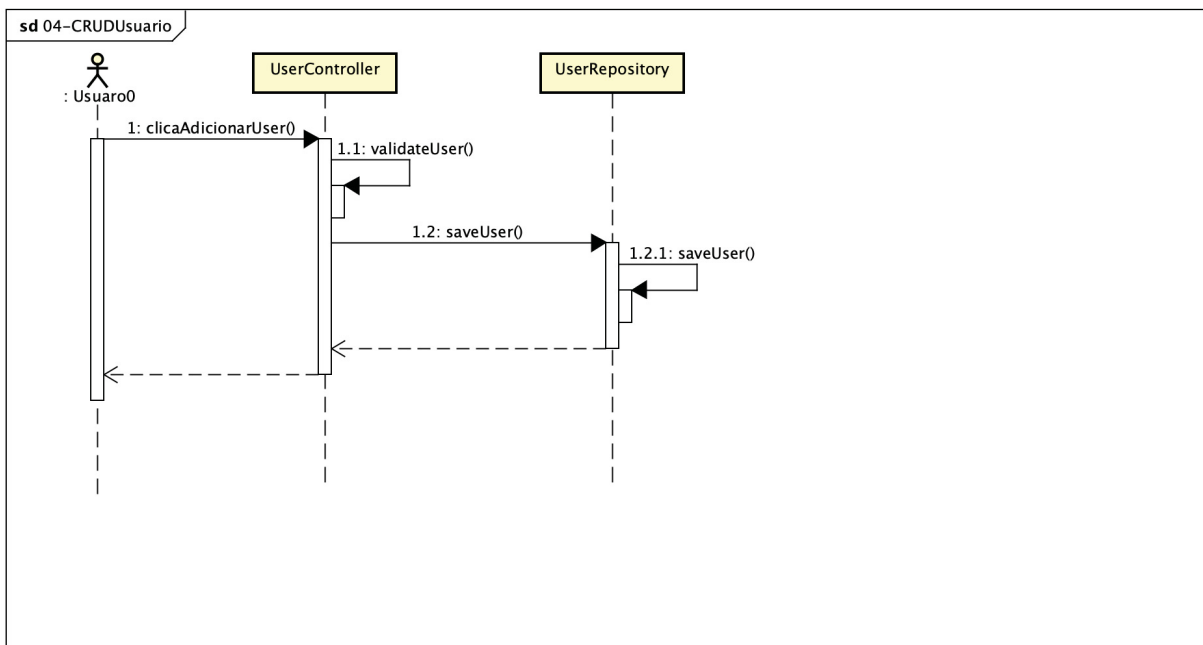
6 DIAGRAMAS DE SEQUÊNCIA

FIGURA 38 – DIAGRAMA DE SEQUÊNCIA DA HU01 - FAZER LOGIN



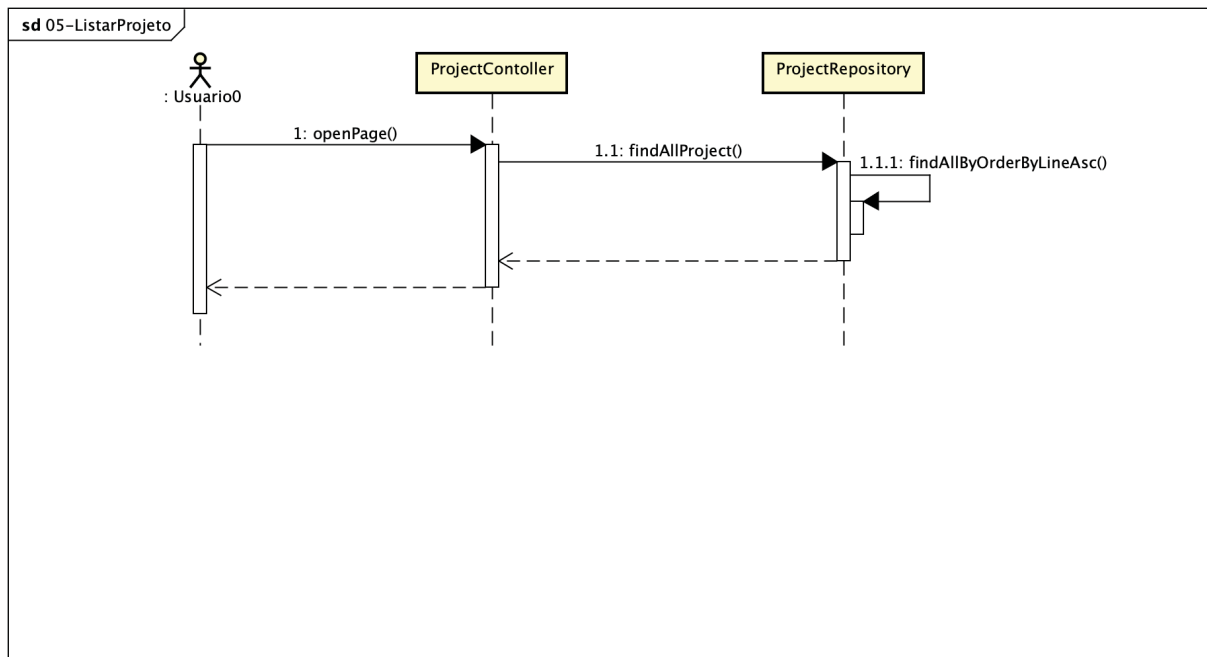
FONTE: O Autor (2023)

FIGURA 39 – DIAGRAMA DE SEQUÊNCIA DA HU02 - CRUD USUÁRIO



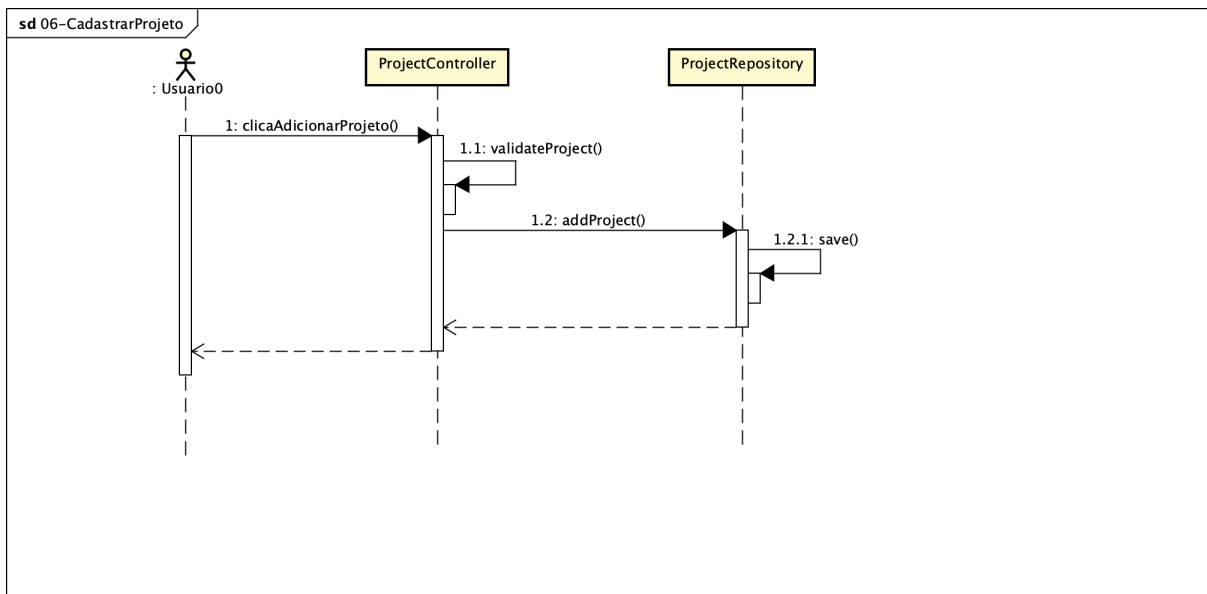
FONTE: O Autor (2023)

FIGURA 40 – DIAGRAMA DE SEQUÊNCIA DA HU03 - LISTAR PROJETO



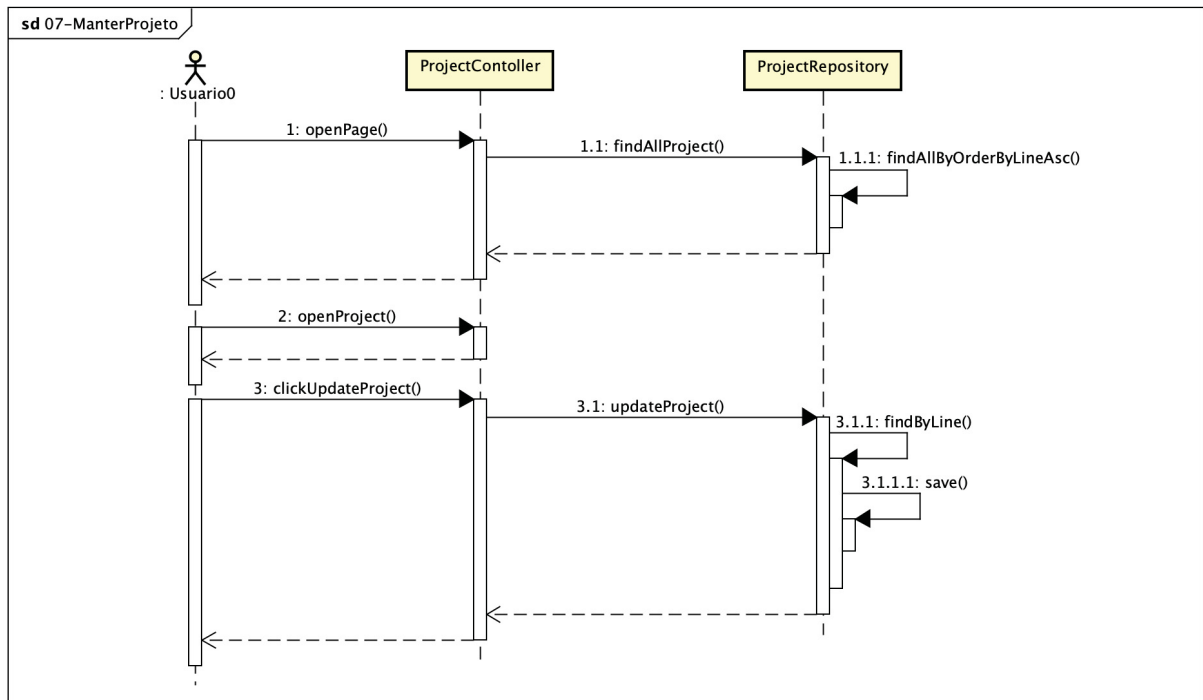
FONTE: O Autor (2023)

FIGURA 41 – DIAGRAMA DE SEQUÊNCIA DA HU04 - CADASTRAR PROJETO



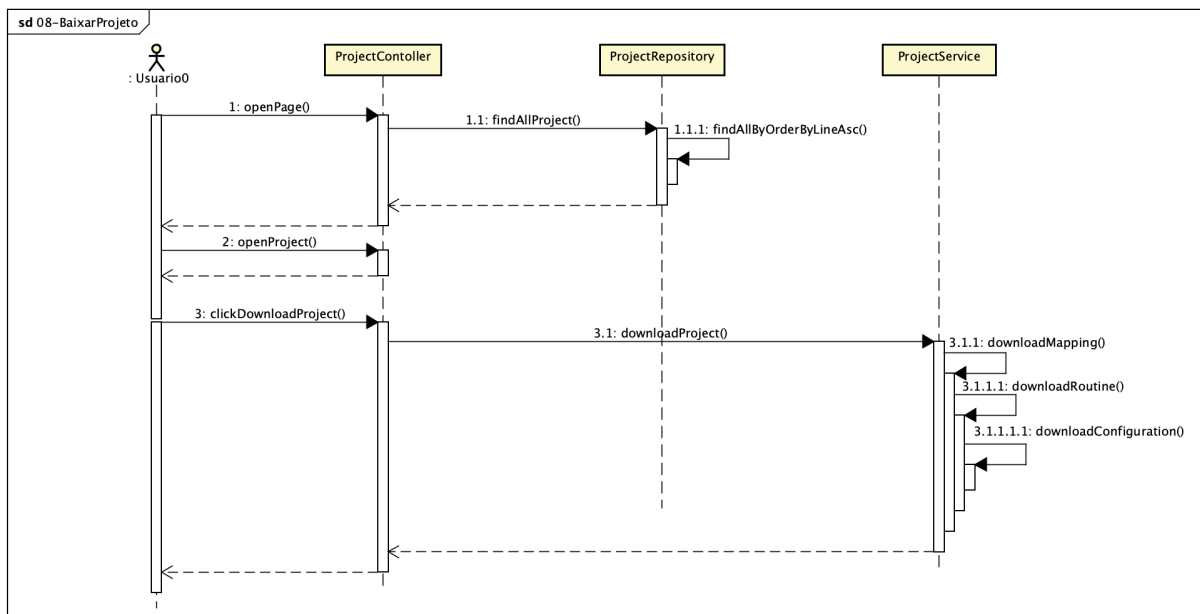
FONTE: O Autor (2023)

FIGURA 42 – DIAGRAMA DE SEQUÊNCIA DA HU05 - MANTER PROJETO



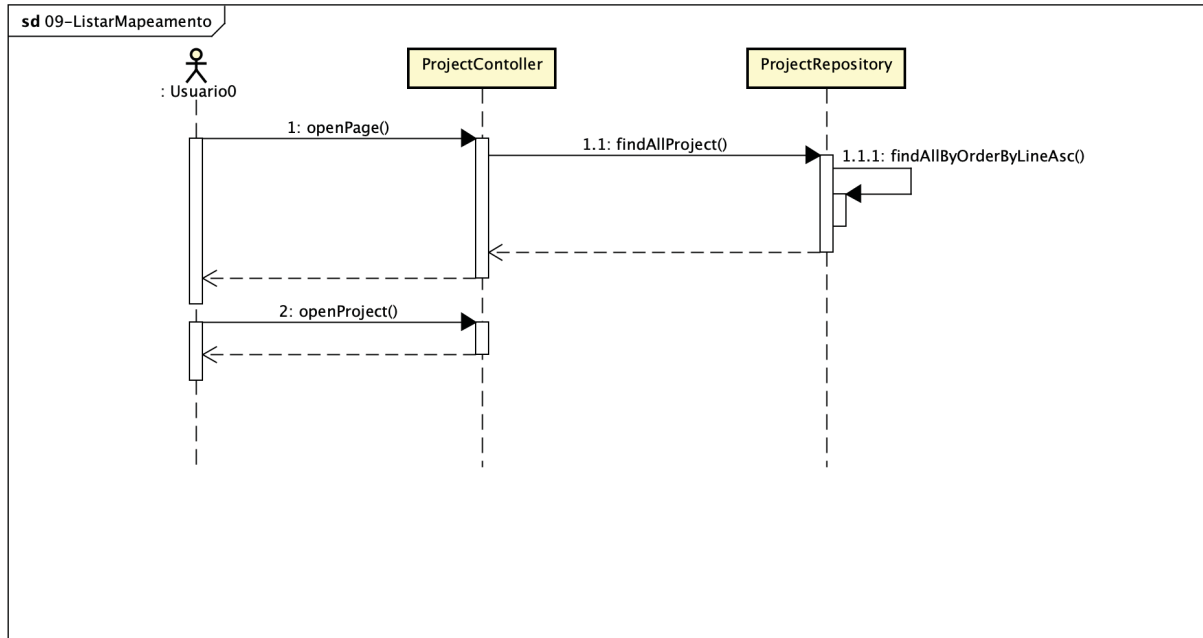
FONTE: O Autor (2023)

FIGURA 43 – DIAGRAMA DE SEQUÊNCIA DA HU06 - BAIXAR PROJETO



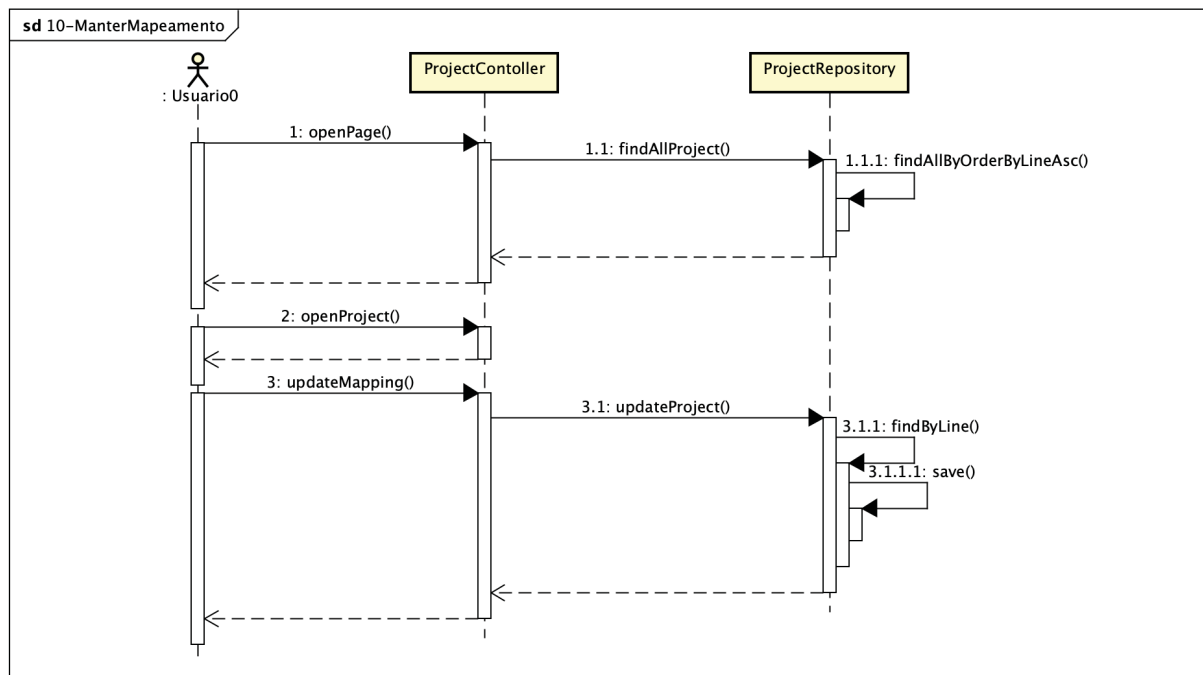
FONTE: O Autor (2023)

FIGURA 44 – DIAGRAMA DE SEQUÊNCIA DA HU07 - LISTAR MAPEAMENTO



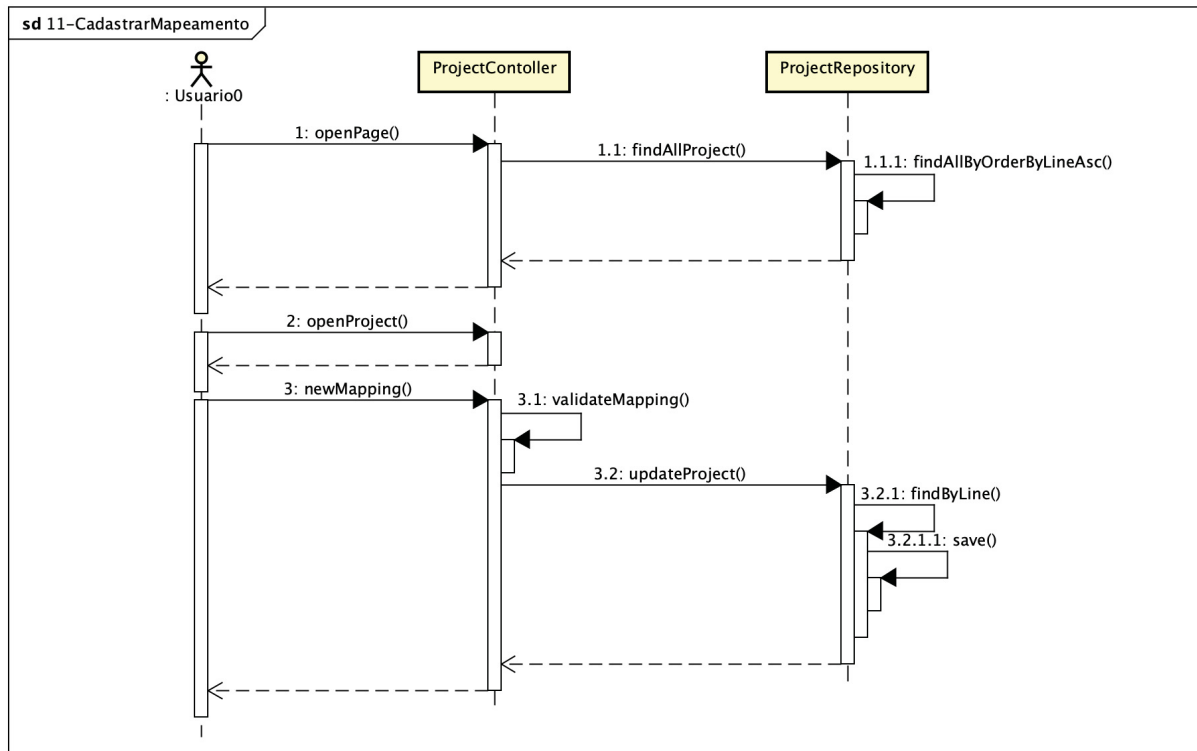
FONTE: O Autor (2023)

FIGURA 45 – DIAGRAMA DE SEQUÊNCIA DA HU08 - MANTER MAPEAMENTO



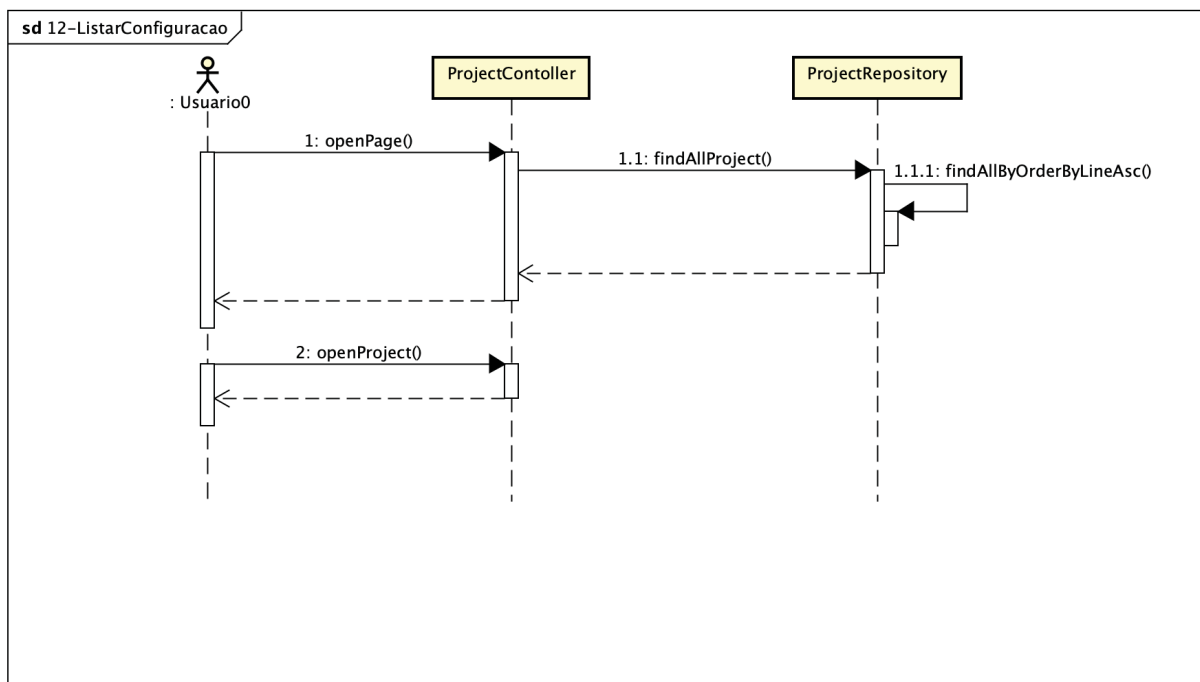
FONTE: O Autor (2023)

FIGURA 46 – DIAGRAMA DE SEQUÊNCIA DA HU09 - CADASTRAR MAPEAMENTO



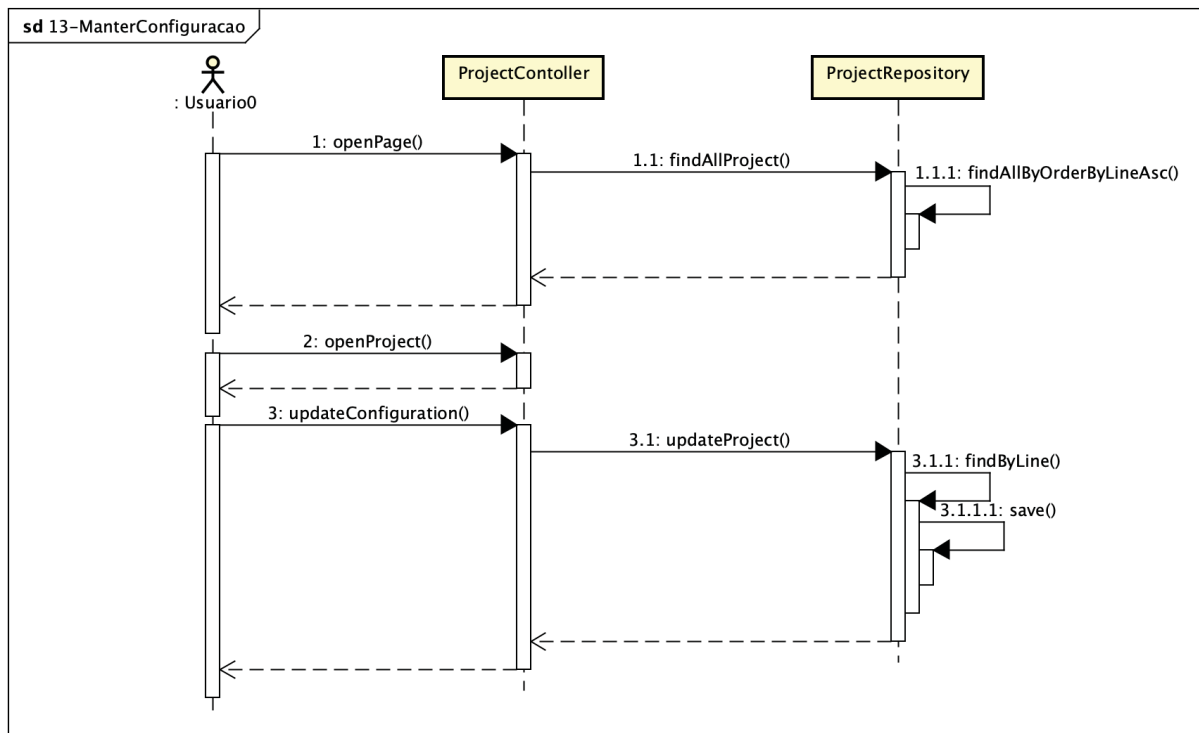
FONTE: O Autor (2023)

FIGURA 47 – DIAGRAMA DE SEQUÊNCIA DA HU10 - LISTAR CONFIGURAÇÃO



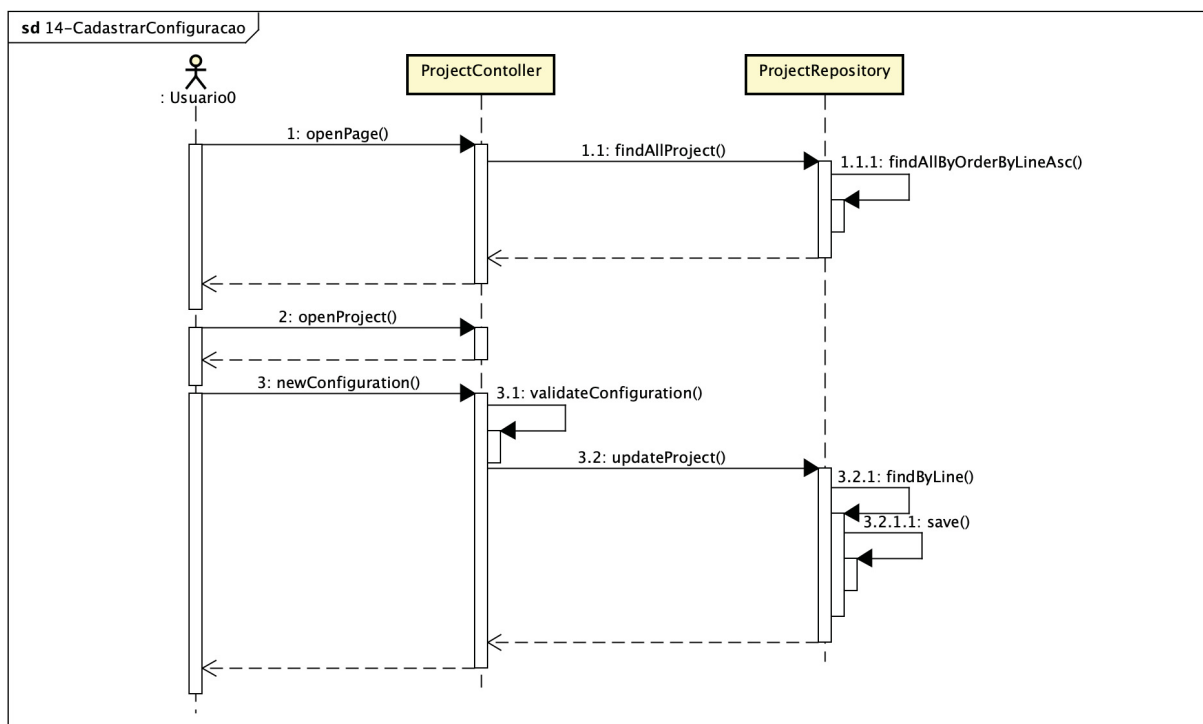
FONTE: O Autor (2023)

FIGURA 48 – DIAGRAMA DE SEQUÊNCIA DA HU11 - CADASTRAR CONFIGURAÇÃO



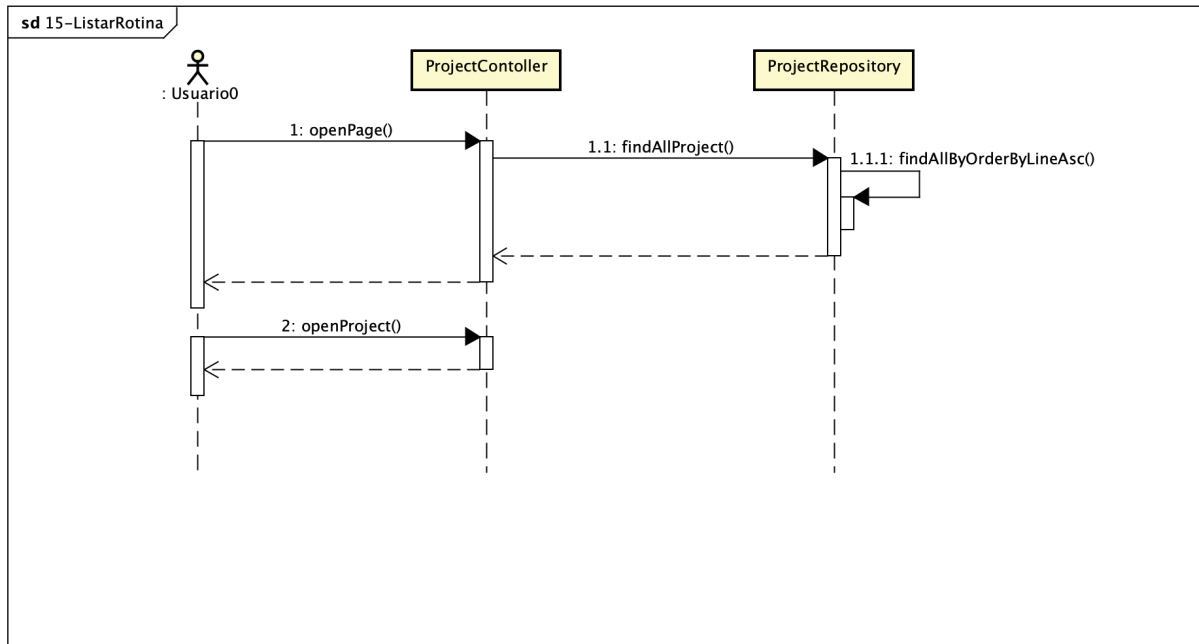
FONTE: O Autor (2023)

FIGURA 49 – DIAGRAMA DE SEQUÊNCIA DA HU12 - MANTER CONFIGURAÇÃO



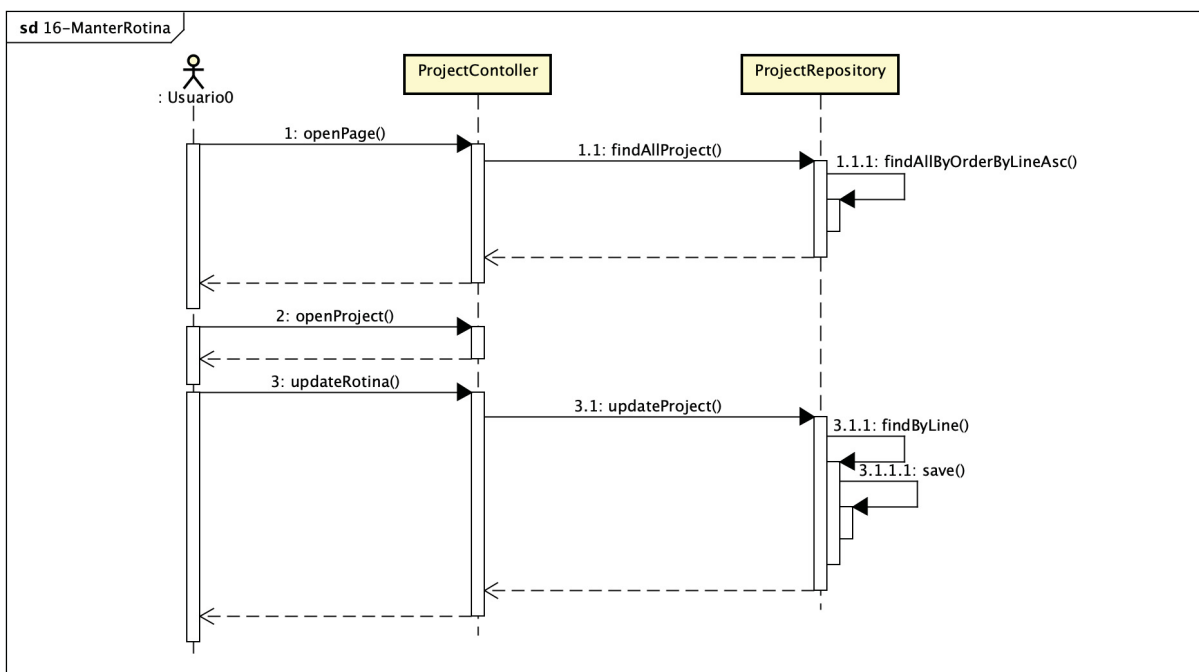
FONTE: O Autor (2023)

FIGURA 50 – DIAGRAMA DE SEQUÊNCIA DA HU13 - LISTAR ROTINA



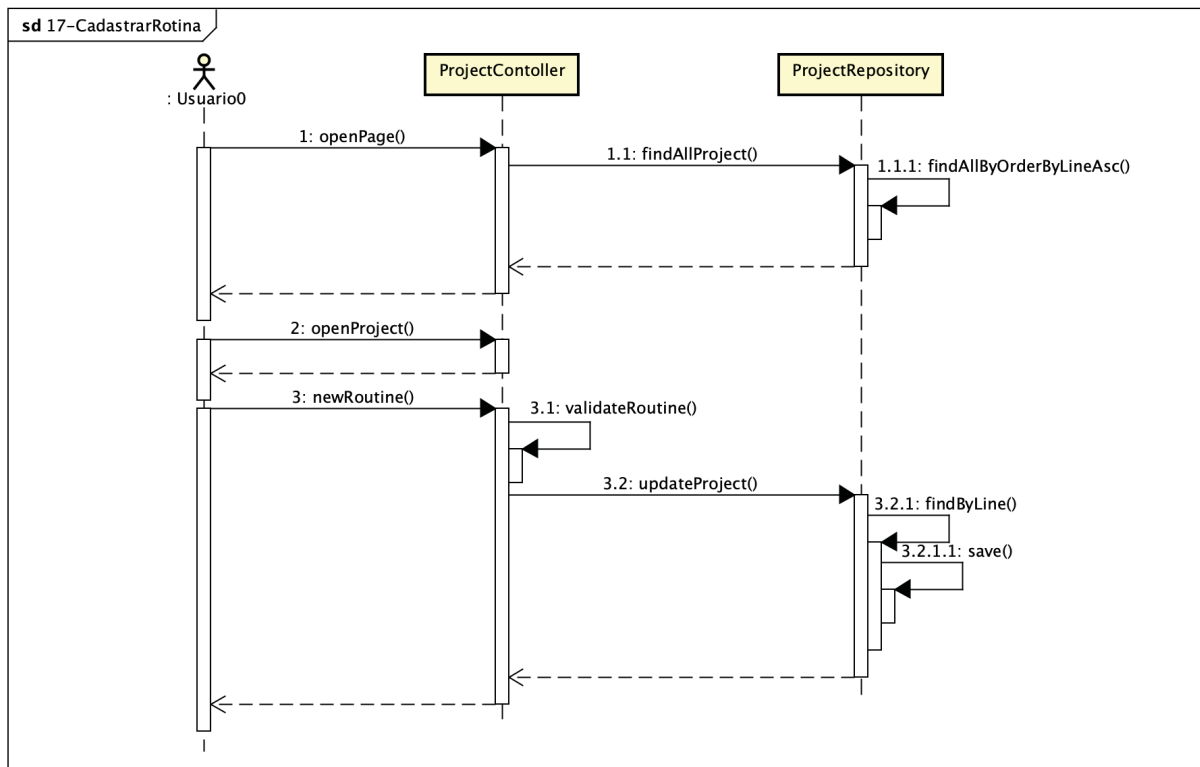
FONTE: O Autor (2023)

FIGURA 51 – DIAGRAMA DE SEQUÊNCIA DA HU14 - CADASTRAR ROTINA



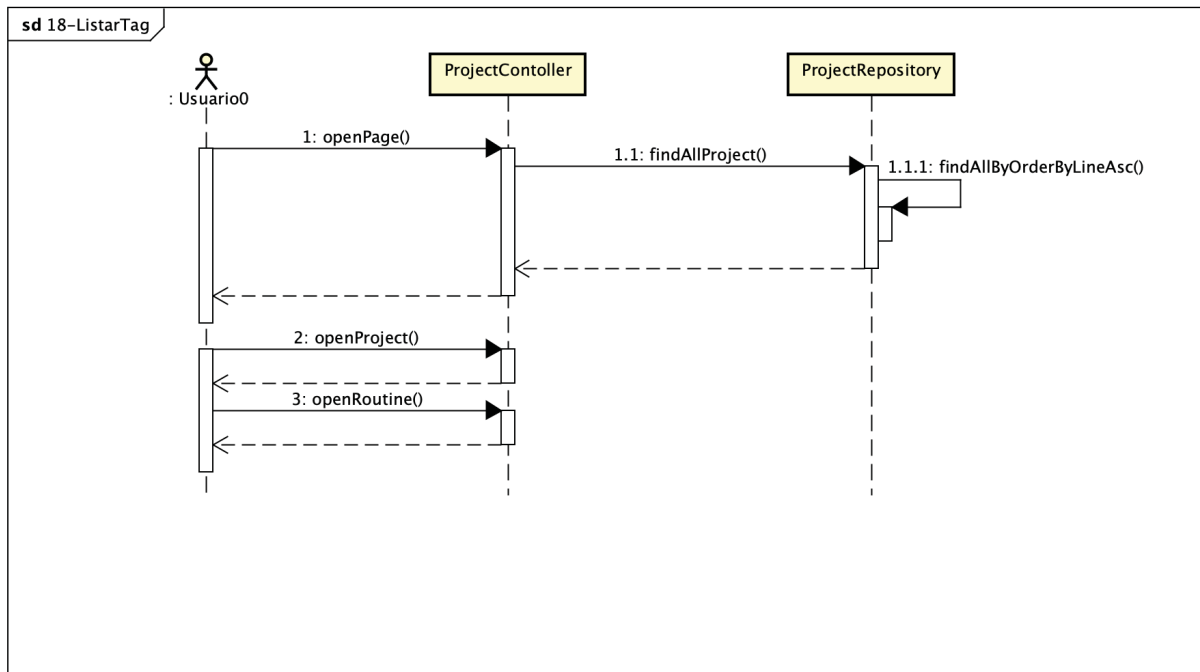
FONTE: O Autor (2023)

FIGURA 52 – DIAGRAMA DE SEQUÊNCIA DA HU15 - MANTER ROTINA



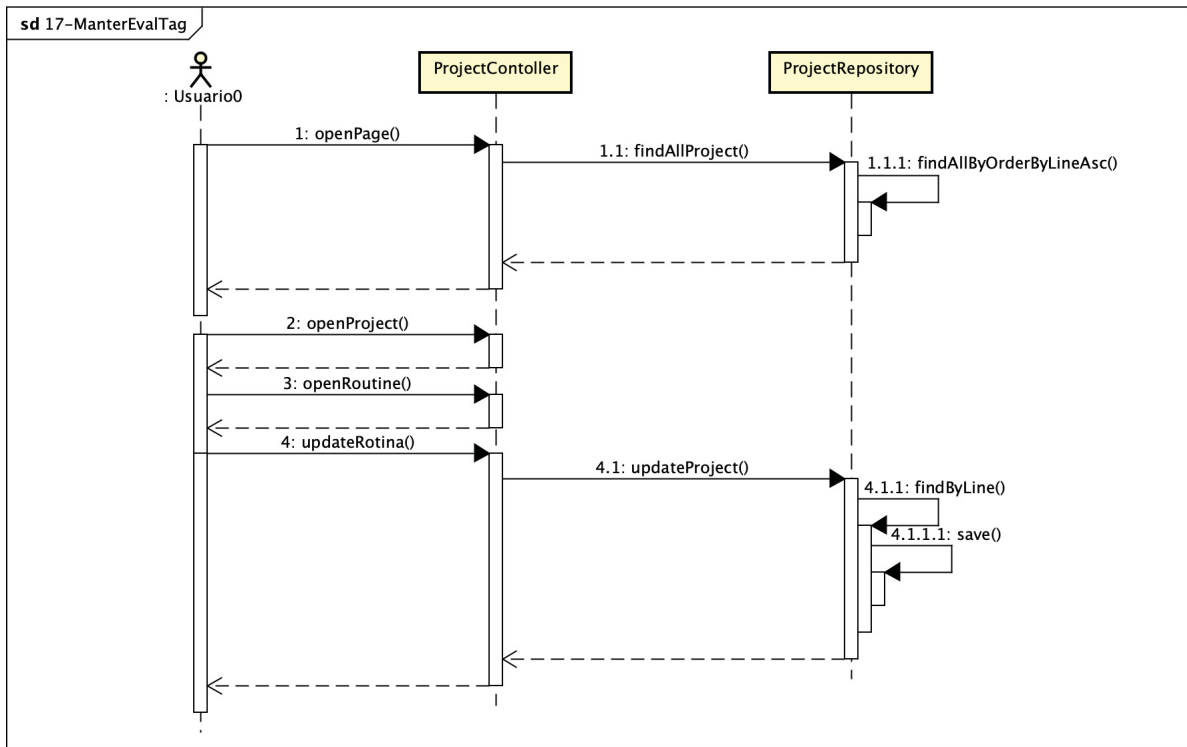
FONTE: O Autor (2023)

FIGURA 53 – DIAGRAMA DE SEQUÊNCIA DA HU16 - LISTAR TAG



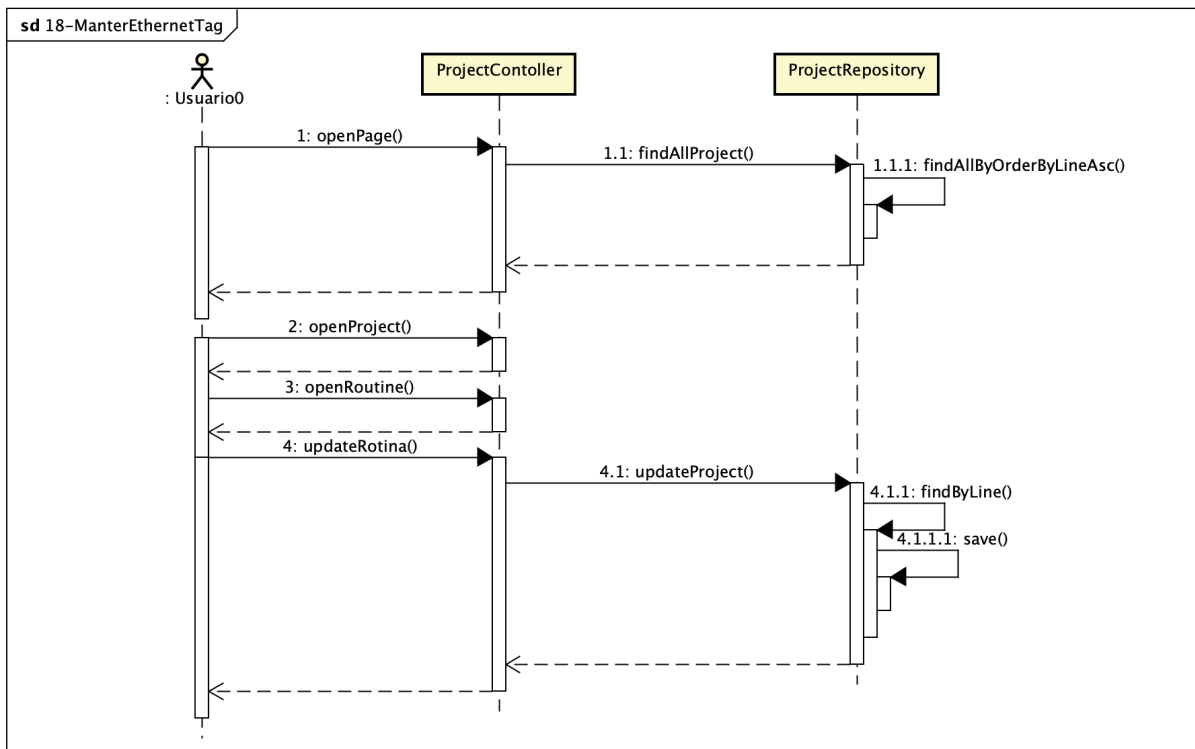
FONTE: O Autor (2023)

FIGURA 54 – DIAGRAMA DE SEQUÊNCIA DA HU17 - MANTER EVALTAG



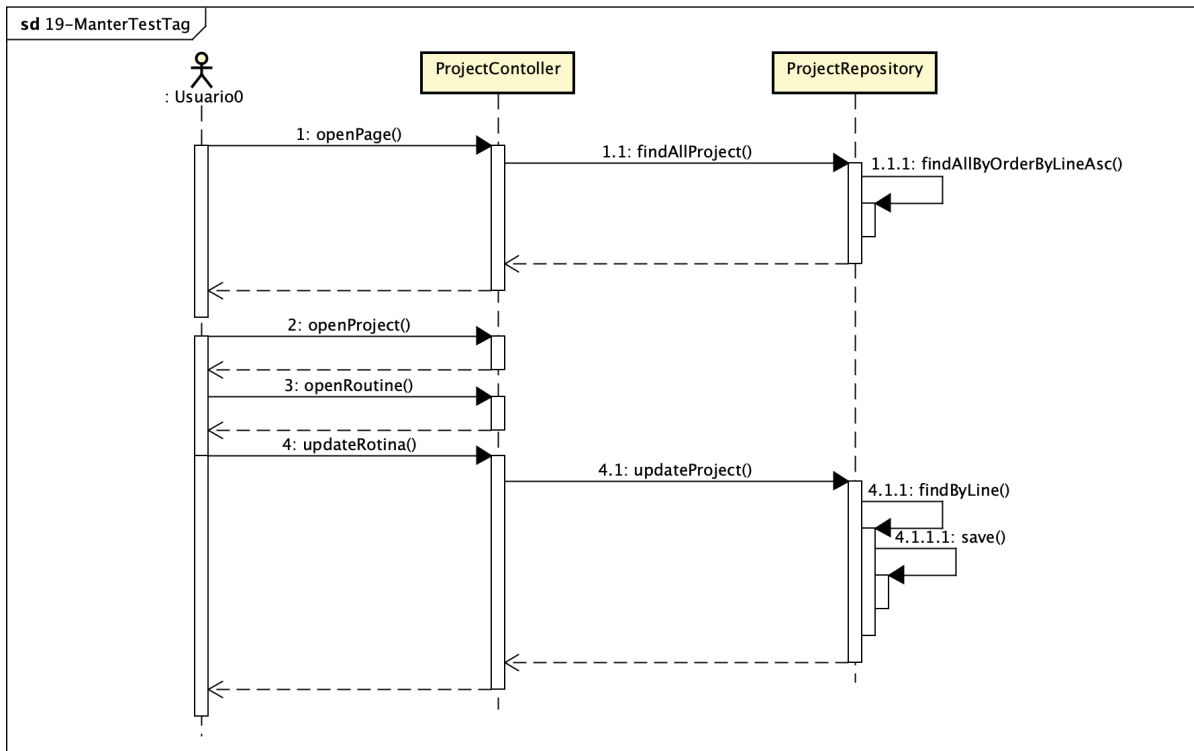
FONTE: O Autor (2023)

FIGURA 55 – DIAGRAMA DE SEQUÊNCIA DA HU18 - MANTER ETHERNET TAG



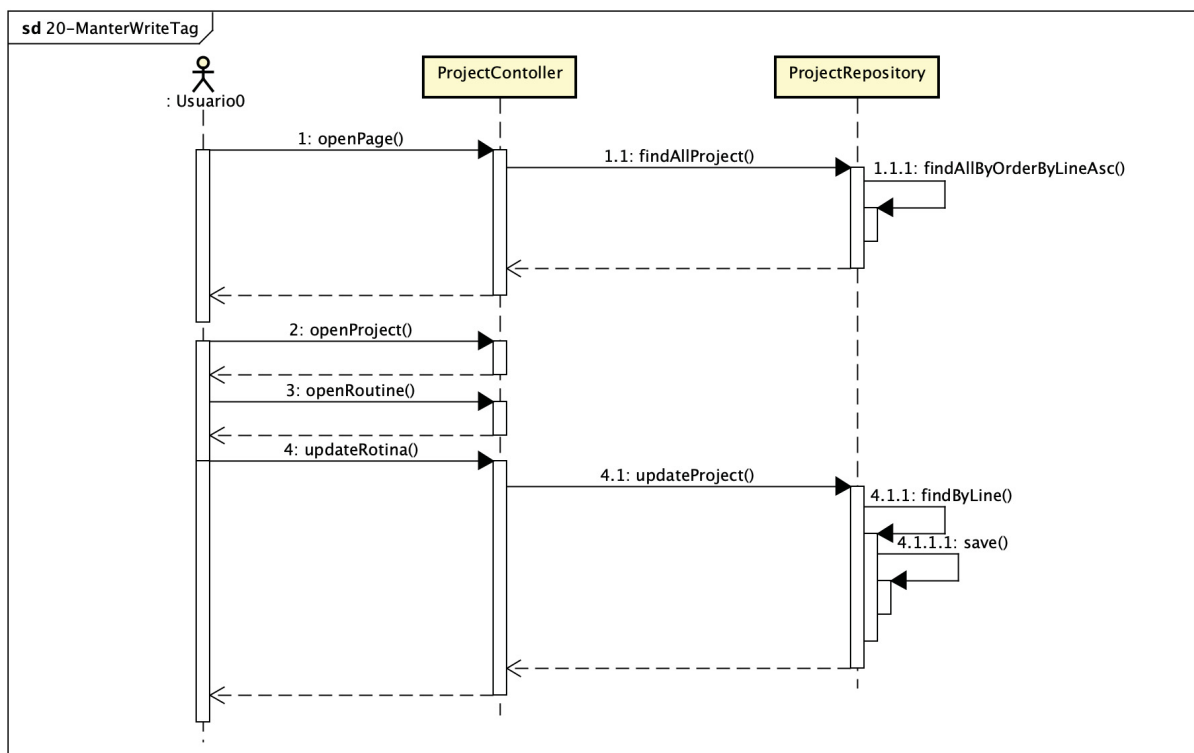
FONTE: O Autor (2023)

FIGURA 56 – DIAGRAMA DE SEQUÊNCIA DA HU19 - MANTER TESTTAG



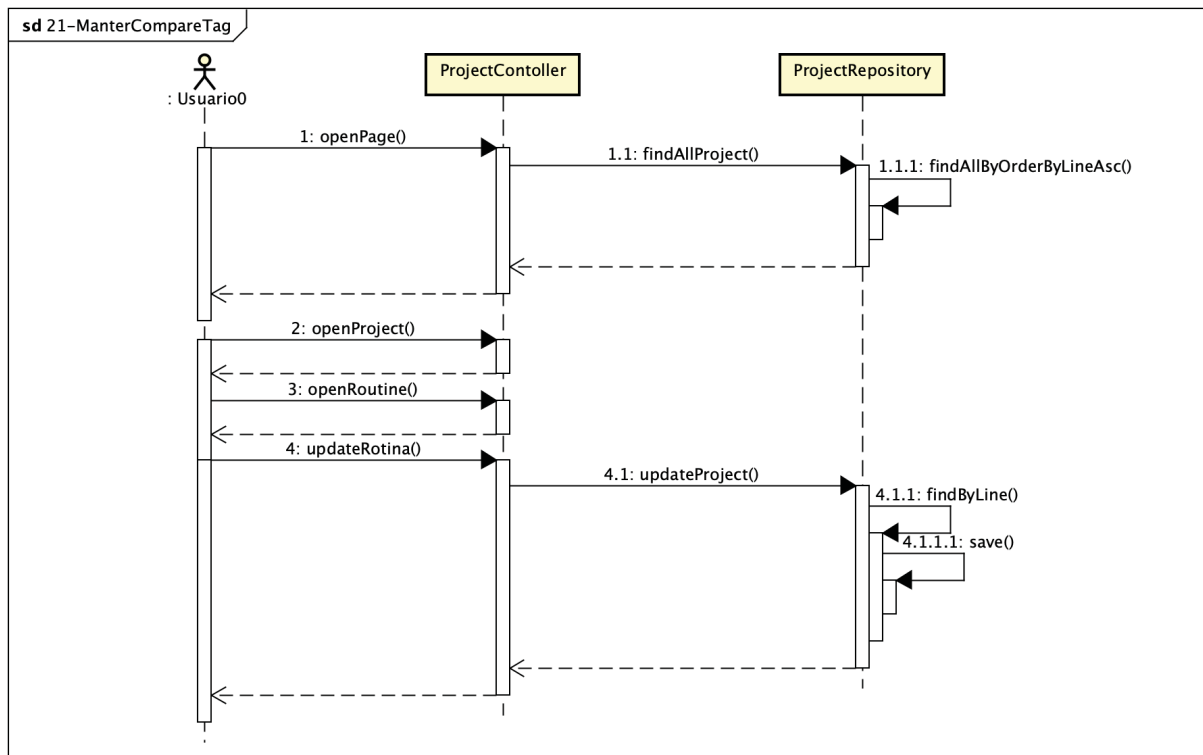
FONTE: O Autor (2023)

FIGURA 57 – DIAGRAMA DE SEQUÊNCIA DA HU20 - MANTER WRITETAG



FONTE: O Autor (2023)

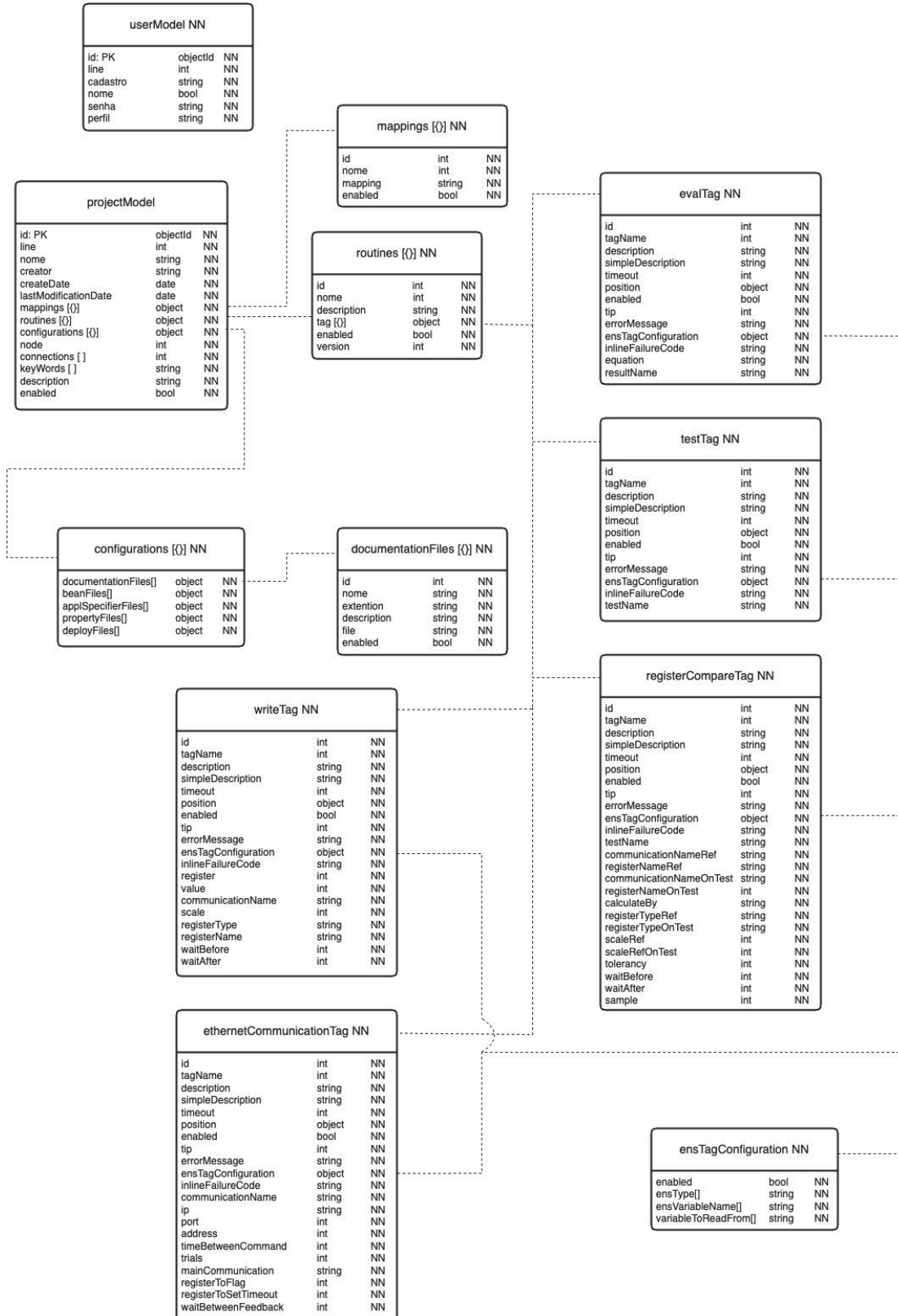
FIGURA 58 – DIAGRAMA DE SEQUÊNCIA DA HU21 - MANTER COMPARETAG



FONTE: O Autor (2023)

7 DIAGRAMA FÍSICO DO BANCO DE DADOS

FIGURA 59 – DIAGRAMA DE BASE DE DADOS (NoSQL)



FONTE: O Autor (2023)

REFERÊNCIAS

ABB. ABB Brasil. Disponível em: <<https://new.abb.com/br>>. Acesso em: 15 de outubro de 2023.

CHROMA. Automated Test Equipment em: <<https://www.chromausa.com/>>. Acesso em: 01 de novembro de 2023.

GENG, H. Manufacturing Engineering Handbook, Second Edition. 2nd Revised ed. edição ed. New York Chicago, Colo. San Francisco Athen London Madrid Mexico City Milan New Delhi Singapore Sydney Toronto: McGraw-Hill Companies, 2015.

GORDON, D. M.; EDWARDS, R.; REICH, M. Segmented Work, Divided Workers: The historical transformation of labor in the United States. Cambridge: Cambridge University Press, 1982.

GRAU, A. et al. Industrial robotics in factory automation: From the early stage to the Internet of Things. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, out. 2017.

MONGODB. MongoDB. Disponível em: <<https://www.mongodb.com/>>. Acesso em: 15 de outubro de 2023.

MOWERY, D. C. Plus ca change: Industrial R&D in the “third industrial revolution”. Industrial and Corporate Change, v. 18, n. 1, p. 1–50, 17 out. 2008.

ORACLE. Java. Disponível em: <<https://www.oracle.com/br/java/>>. Acesso em: 15 de outubro de 2023.

PIETRYKOWSKI, B. Fordism at Ford: Spatial Decentralization and Labor Segmentation at the Ford Motor Company, 1920-1950. Economic Geography, v. 71, n. 4, p. 383, out. 1995.

SIEMENS. Siemens Brasil. Disponível em: <<https://www.siemens.com/br>>. Acesso em: 15 de outubro de 2023.

SPRING. Spring Framework. Disponível em:

<<https://spring.io/projects/spring-framework>>. Acesso em: 15 de outubro de 2023.

TYPESCRIPT. TypeScript. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 15 de outubro de 2023.