



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA - CEEI
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - DEE
PERÍODO: 2025.1**

DISCIPLINA: Informática Industrial
PROFESSOR: Danilo Freire de Sousa Santos
DISCENTES: Mateus Pincho de Oliveira e Yan Abrantes Cavalcante

**1º ESTÁGIO
INFORMÁTICA INDUSTRIAL
Robôs Autônomos para Sensoriamento Industrial Inteligente - RAAI**

**CAMPINA GRANDE - PB
2025**

1 Apresentação do Projeto.....	3
1.1 Objetivo Geral.....	3
1.2 Objetivos específicos.....	4
2 Visão Geral da Solução Proposta.....	4
2.1 Funcionamento do Robô RAAI.....	4
2.2 Módulos do Sistema.....	6
2.2.1 Eletrônica.....	6
2.2.2 Planejamento e Navegação.....	7
2.2.3 Software.....	8
2.2.4 Comunicação.....	9
3 Cenários de Uso.....	11
3.1 Monitoramento Contínuo de Plantas Industriais.....	11
3.2 Monitoramento de centro de distribuição.....	12
4 Especificações e Requisitos.....	13
4.1 Requisitos Gerais.....	13
4.2 Requisitos de Eletrônica.....	14
4.3 Requisitos de Robótica e Navegação.....	17
4.4 Requisitos de Software.....	21
4.5 Requisitos de Comunicação.....	24
5 Recursos necessários.....	26
6 Plano de Projeto.....	27
6.1 Metodologia Scrum.....	27
6.2 Fases do Projeto e Definição das Sprints.....	28
7 Verificação e Validação dos Requisitos.....	31

1 Apresentação do Projeto

Conforme a necessidade de produção aumenta, a indústria busca maneiras de otimizar a produção e aumentar a sua produtividade. Para isto, tornou-se uma prática recorrente a utilização de sensores inteligentes capazes de monitorar continuamente o estado da produção, permitindo o ajuste de plantas industriais e o controle ótimo da produção.

Cada vez mais percebe-se a necessidade de tornar as indústrias inteligentes e responsivas, diminuindo o trabalho de monitoramento por operadores humanos e substituindo por sensores capazes de detectar e prever falhas que poderiam parar a produção. No entanto, para grandes fábricas, com diversas plantas industriais que operam de forma crítica, torna-se muitas vezes inviável a utilização de diversos sensores, justamente por ser necessário uma grande quantidade de sensores espalhados por toda a fábrica, além de ser necessário toda uma adaptação da estrutura física da fábrica para instalar nos locais ideais para o monitoramento eficiente dessas máquinas.

Neste sentido, a utilização de robôs autônomos para monitoramento e sensoriamento inteligente de ambientes industriais torna-se vantajosa em relação a sensores fixos, devido a sua flexibilidade em termos de movimentação, permitindo o monitoramento de diversas plantas industriais.

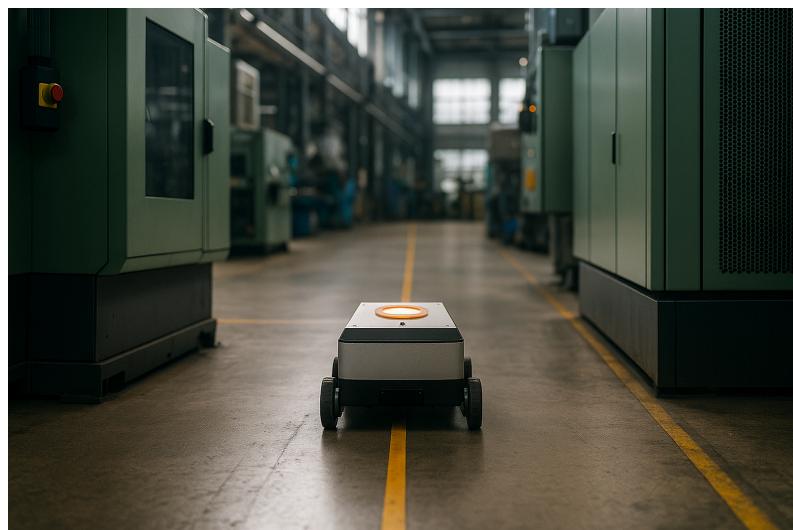


Figura 1 - Apresentação do robô RAAI

1.1 Objetivo Geral

Este projeto tem por objetivo desenvolver e validar um modelo de robô móvel autônomo para ambientes industriais capazes de realizar patrulhas periódicas com sensoriamento multimodal (visual, ambiental, térmico e sonoro), detecção de anomalias, comunicação inteligente e geração automatizada de relatórios técnicos para operadores.

1.2 Objetivos específicos

1. Desenvolver um modelo de robô móvel autônomo para ambientes industriais, equipado com sensores inteligentes capazes de monitorar variáveis de interesse em um ambiente fabril.
2. Ser capaz de detectar anomalias no ambiente, como pessoas sem Equipamento de Proteção Individual, objetos fora do lugar e possíveis causas de acidentes, como manchas de óleo e água.
3. Ser capaz de navegar de forma autônoma por um ambiente fabril, passível de mudanças, e gerar um mapa 3D do ambiente à sua volta.
4. Salvar as leituras dos sensores e, em caso de alterações e anomalias detectadas, enviar alertas aos operadores das fábricas.
5. Gerar relatórios e análises baseadas nas leituras dos sensores, sendo capaz de interagir com os operadores através de uma Interface Humano-Máquina.

2 Visão Geral da Solução Proposta

2.1 Funcionamento do Robô RAAI

O funcionamento geral do robô RAAI baseia-se em um ciclo contínuo de patrulha inteligente. Os robôs estão espalhados por todo o ambiente industrial, trabalhando em frotas para garantir cobertura integral da planta industrial. Ao receber comandos da estação-base pelo operador da fábrica, o robô, após realizar a verificação de integridade dos componentes, inicia sua missão.

Durante a patrulha, o robô navega de forma autônoma pelo ambiente, seguindo a rota pré-definida pelo operador. Simultaneamente, ele realiza a coleta contínua de dados de máquinas, como a temperatura de operação, e monitora o estado do ambiente através de sua câmera. Ao detectar anomalias, como objetos fora de lugar, pessoas em Equipamento de Proteção Individual e possíveis defeitos nas máquinas, o robô gera alertas. A cada intervalo programado, os dados e alertas são transmitidos para a estação-base.

Durante a missão, o robô pode apresentar falhas de funcionamento, ativando rotinas de diagnóstico e tratamento. Uma vez corrigido o problema ou após intervenção do operador, o robô pode retornar à execução normal. Ao final da missão, o robô retorna automaticamente à sua base de carregamento, encerrando o ciclo de funcionamento. O diagrama de estados que ilustra este funcionamento pode ser visualizado na Figura 2.

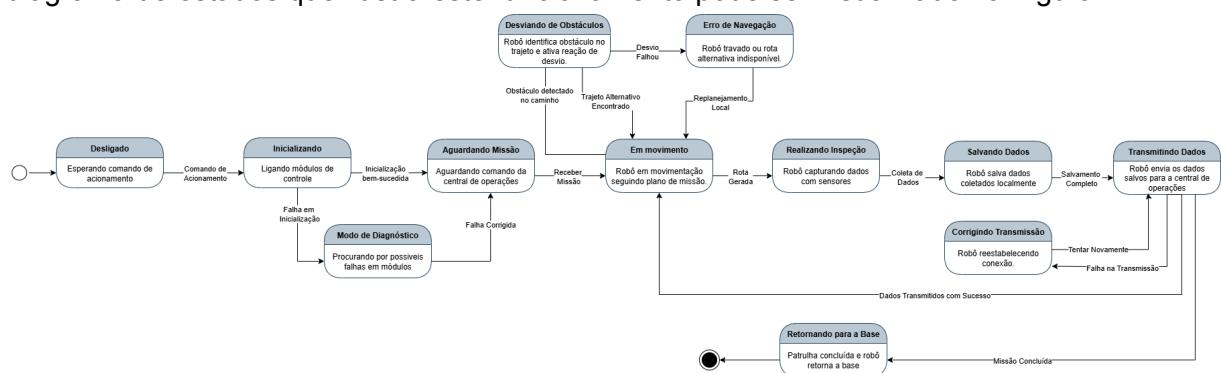


Figura 2 - Diagrama de Estados do Robô RAAI

Estado	Transição	Próximo Estado	Descrição
Desligado	Comando de Acionamento	Inicializando	Robô é ligado pela central de comando
Inicializando	Inicialização bem sucedida	Aguardando Missão	Robô não detecta falhas em componentes e está pronto para iniciar patrulha
Inicializando	Falha em Inicialização	Modo de Diagnóstico	Robô não consegue inicializar componentes e inicia modo de diagnóstico
Modo de Diagnóstico	Falha Corrigida	Aguardando Missão	Robô detecta e corrige falhas, estando pronto para iniciar patrulha
Aguardando Missão	Receber Missão	Em Movimento	Robô recebe comando para iniciar patrulha
Em Movimento	Rota Gerada	Realizando Inspeção	Robô inicia caminhos de patrulha e realiza a coleta de dados da planta industrial
Em Movimento	Obstáculo Detectado no Caminho	Desviando de Obstáculos	Robô detecta obstáculos no caminho e inicia rota alternativa
Desviando de Obstáculos	Trajeto Alternativo Encontrado	Em movimento	Robô desvia de obstáculo em rota alternativa
Desviando de Obstáculos	Desvio Falhou	Erro de navegação	Desvio de Objeto falhou e robô parou
Erro de navegação	Replanejamento Local	Em Movimento	Robô inicia replanejamento para voltar a trajeto original
Realizando Inspeção	Coleta de Dados	Salvando Dados	Dados coletados pelo robô são salvos localmente
Salvando Dados	Salvamento Completo	Transmitindo Dados	Dados salvos são enviados para a estação-base
Transmitindo Dados	Falha na Transmissão	Corrigindo Transmissão	Robô sem conexão com a estação-base
Corrigindo Transmissão	Tentar novamente	Transmitindo Dados	Robô tenta restabelecer conexão com estação-base

Transmitindo Dados	Dados transmitidos com sucesso	Em movimento	Robô envia dados a estação-base e continua a inspeção
Transmitindo Dados	Missão Concluída	Retornando a Base	Robô conclui a rota de patrulha e retorna à base de carregamento

Tabela 1 - Descrição dos estados e eventos de transição do robô RAAI

2.2 Módulos do Sistema

O sistema está dividido em 4 Módulos, sendo eles: Eletrônica, Planejamento e Navegação, Software e Comunicação. Esses módulos partilham funcionalidades entre si de modo a tornar o sistema integrado, inteligente e o mais sofisticado possível.

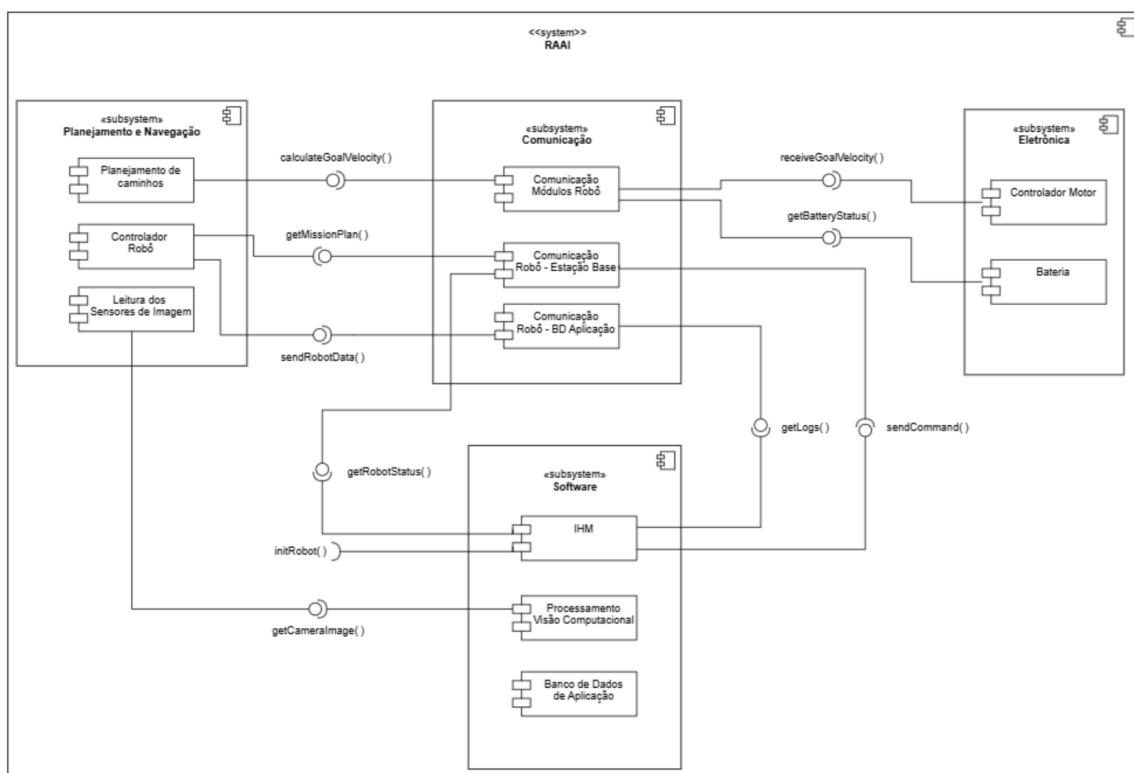


Figura 3 - Diagrama de Componentes do Sistema RAAI

2.2.1 Eletrônica

Para tornar o robô funcional, é necessário projetar o módulo de controle dos motores. Para movimentar-se, são necessários 4 motores de corrente contínua de 25 Watts de potência e 4 kgf·cm de torque. Rodas omnidirecionais são acopladas nos motores para permitir movimentos em todas as direções. Para acionar os motores, são utilizados o motor driver BTS7960. Sensores NTC são utilizados para monitoramento e gerenciamento térmico dos motores. Encoders de efeito Hall são utilizados para estimar a velocidade das rodas, essencial para os algoritmos de robótica. Para controlar todos esses módulos e garantir velocidade constante dos motores, o microcontrolador Raspberry Pi Pico RP2040 é utilizado e comunica-se com o módulo de Robótica. Um sistema de baterias será utilizado

para fornecer energia aos motores e ligar todos os módulos de controle dos motores e do robô. O robô contará ainda com um módulo de LEDs gerenciados via GPIO do RP2040 que indica o seu estado operacional (normal, erro, baixa bateria, comunicação perdida).

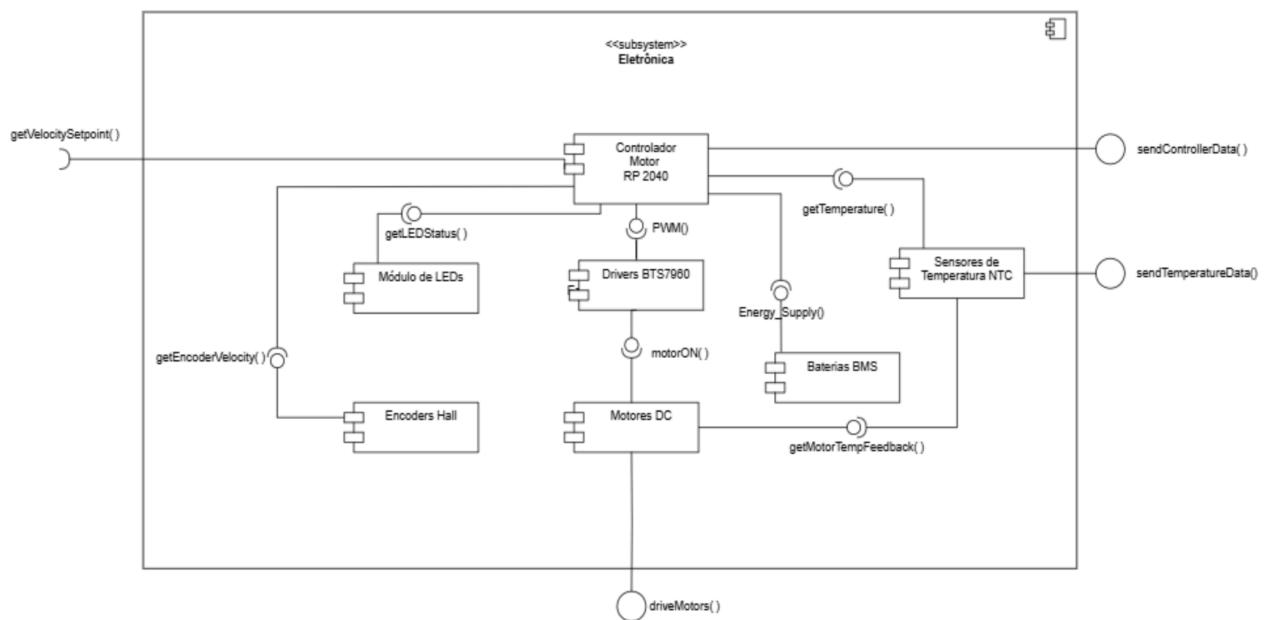


Figura 4 - Diagrama de Componentes do módulo de Eletrônica

2.2.2 Planejamento e Navegação

Além de funcional, o robô precisa ser inteligente. O módulo de controle do robô é responsável por realizar a implementação dos algoritmos de navegação autônoma, processar as leituras dos sensores de temperatura e enviá-las para a IHM na central de operações da fábrica. O módulo de robótica conta com um LIDAR 3D, câmeras e sensores de distância, equipamentos necessários para a realização de navegação autônoma segura pelo ambiente. Além disso, o robô conta com um banco de dados local e capacidade de se comunicar com a estação-base.

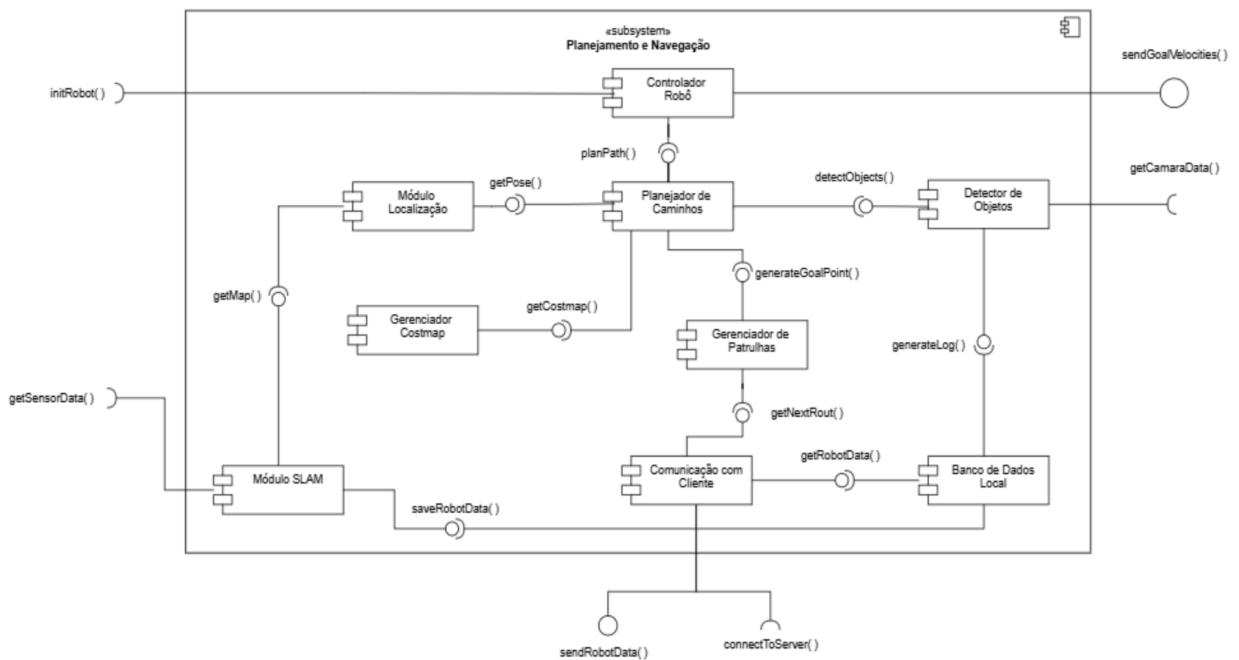


Figura 5 - Diagrama de Componentes do módulo de Planejamento e Navegação do robô

2.2.3 Software

A camada de software do robô é de extrema importância, pois ela coordena o fluxo de informações entre os módulos, integrando os sensores e a interface IHM. A Jetson irá operar com o sistema operacional Ubuntu 22.04 LTS, juntamente com o ROS 2 Humble para realizar a integração entre todos os subsistemas do robô. O robô deve processar as imagens capturadas pela câmera e as leituras dos sensores, a fim de detectar se há anomalias pré-determinadas no ambiente.

As leituras dos sensores, eventos detectados pelo robô, bem como os mapas 3D gerados da fábrica devem ser enviados a uma interface web. Um banco de dados deve armazenar as leituras dos sensores do robô, permitindo consultas futuras. Um serviço de LLM estará disponível aos operadores para consulta e insights sobre os dados capturados pelo robô, bem como para a geração de relatórios personalizados. Estas funcionalidades estarão disponíveis ao operador através de uma GUI na central de operações.

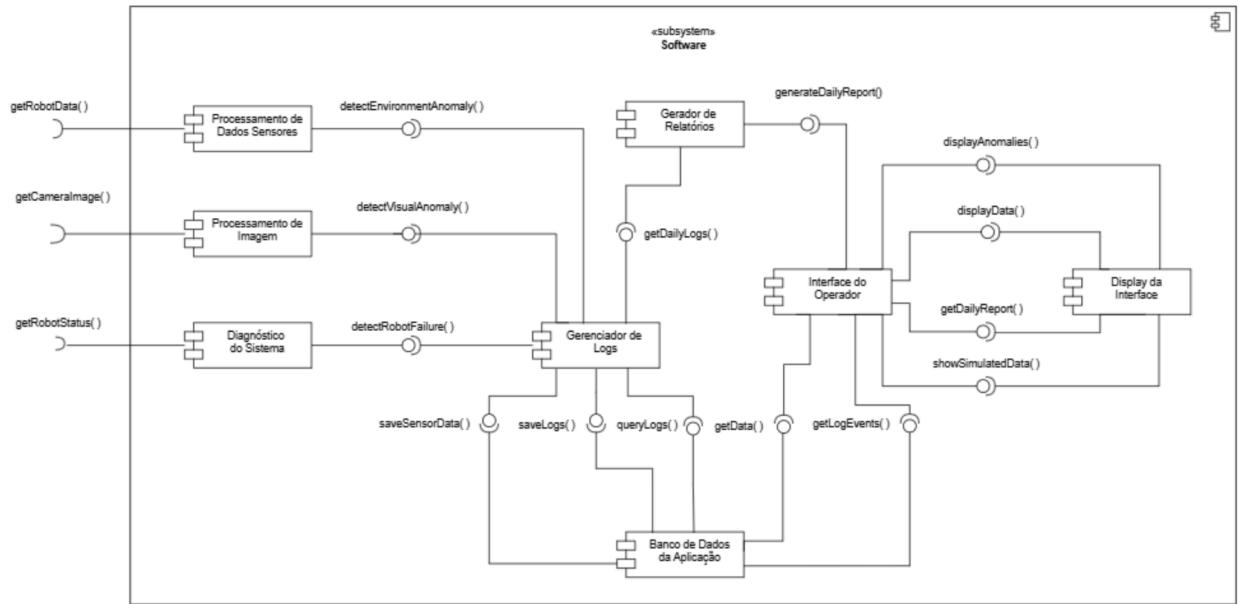


Figura 6 - Diagrama de Componentes do Software do produto

2.2.4 Comunicação

O subsistema de comunicação garante que os dados sejam transmitidos com baixa latência entre o robô, estação de comando e os demais sistemas da fábrica. No próprio robô, os subsistemas de controle se comunicam através do ROS 2.

O servidor, que armazena as informações das missões, recebe os dados de telemetria e envia comandos ao robô através de comunicação via MQTT. Perceba que esta configuração permite que haja vários robôs operando ao mesmo tempo, comunicando-se com o servidor. As informações do servidor são mostradas ao operador através de uma Interface Web, que acessa o banco de dados do servidor através de Websockets.

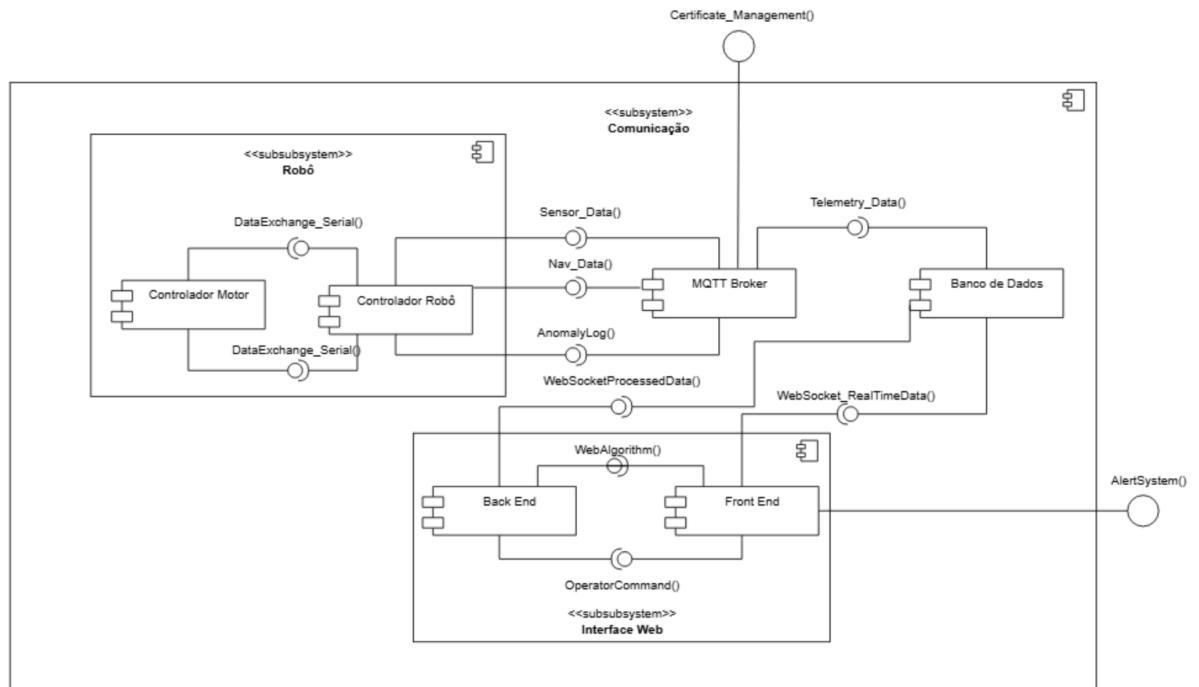


Figura 7 - Diagrama de Componentes da Comunicação do Sistema

A figura a seguir apresenta um diagrama de sequência que explicita a comunicação entre os componentes do sistema de maneira mais clara, levando em consideração o processo de solicitação, emissão e confirmação de certificados, bem como o cumprimento do requisito de geração de novos certificados a cada 90 dias.

Além disso, é importante destacar que o objeto “Autentica Certificado” faz parte do Broker MQTT, entretanto, por viés de informação e de forma a tornar o fluxo de sequência o mais exposto possível, optou-se por explicitá-lo de maneira externa ao Broker.

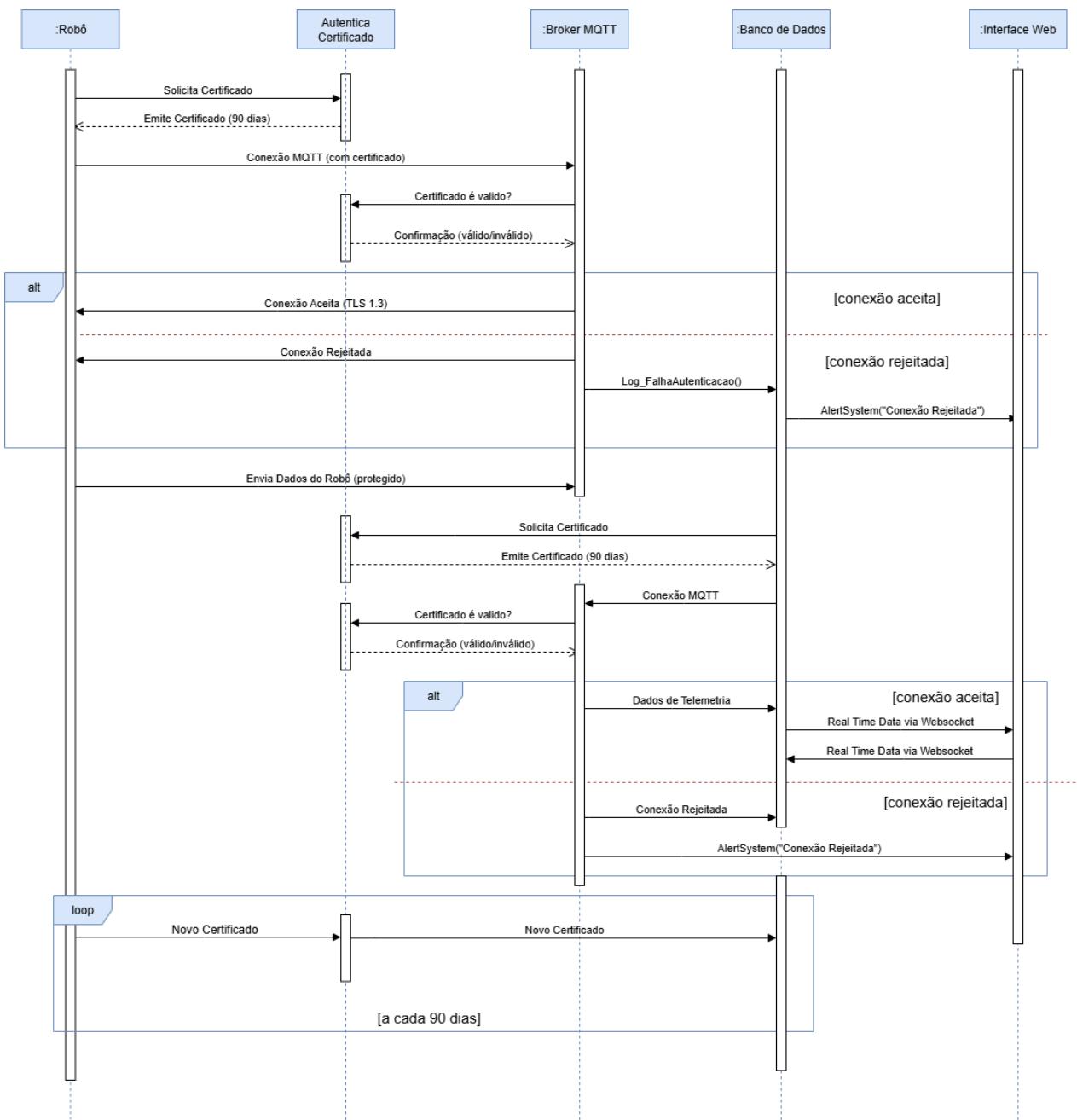


Figura 8 - Diagrama de Sequência da Comunicação entre os componentes do sistema.

3 Cenários de Uso

3.1 Monitoramento Contínuo de Plantas Industriais

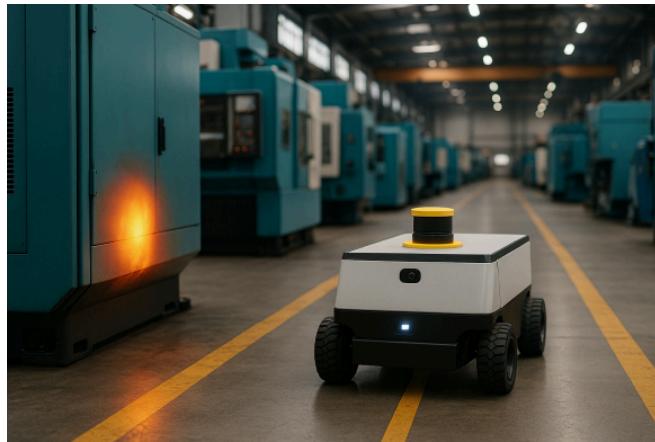


Figura 9 - Demonstração do robô operando em planta industrial

Em uma refinaria de petróleo, existem diversas plantas industriais que funcionam intermitentemente durante o dia e não podem ser paradas, pois isto significa perdas de produção. Estas plantas possuem máquinas pesadas, com possibilidade de sofrerem aquecimentos excessivos e vazamentos de fluídos ao longo de sua operação. Operadores desta planta circulam periodicamente para supervisão.

O robô é iniciado pelos operadores e percorre rotas pré-definidas durante o dia, incluindo locais que são de difícil acesso para humanos. Seu sensor térmico é capaz de identificar pontos quentes e gerar um mapa de calor da refinaria. O robô identifica manchas de óleo ou água no chão com sua câmera. Ao identificar essas anomalias, o robô envia alertas aos operadores através de sua IHM. Esses alertas podem ser alertas comuns, que significam que o robô identificou algum ponto quente ou manchas/vazamentos ou podem ser alertas de emergência, que significa que o robô não está funcionando como deveria. Ao final do dia, sua interface gera um relatório contendo todas as anomalias e pontos de falha detectados. O operador deve ter acesso ao relatório e com base nos dados e informações captadas pelo robô, identificar e traçar estratégias de comando.

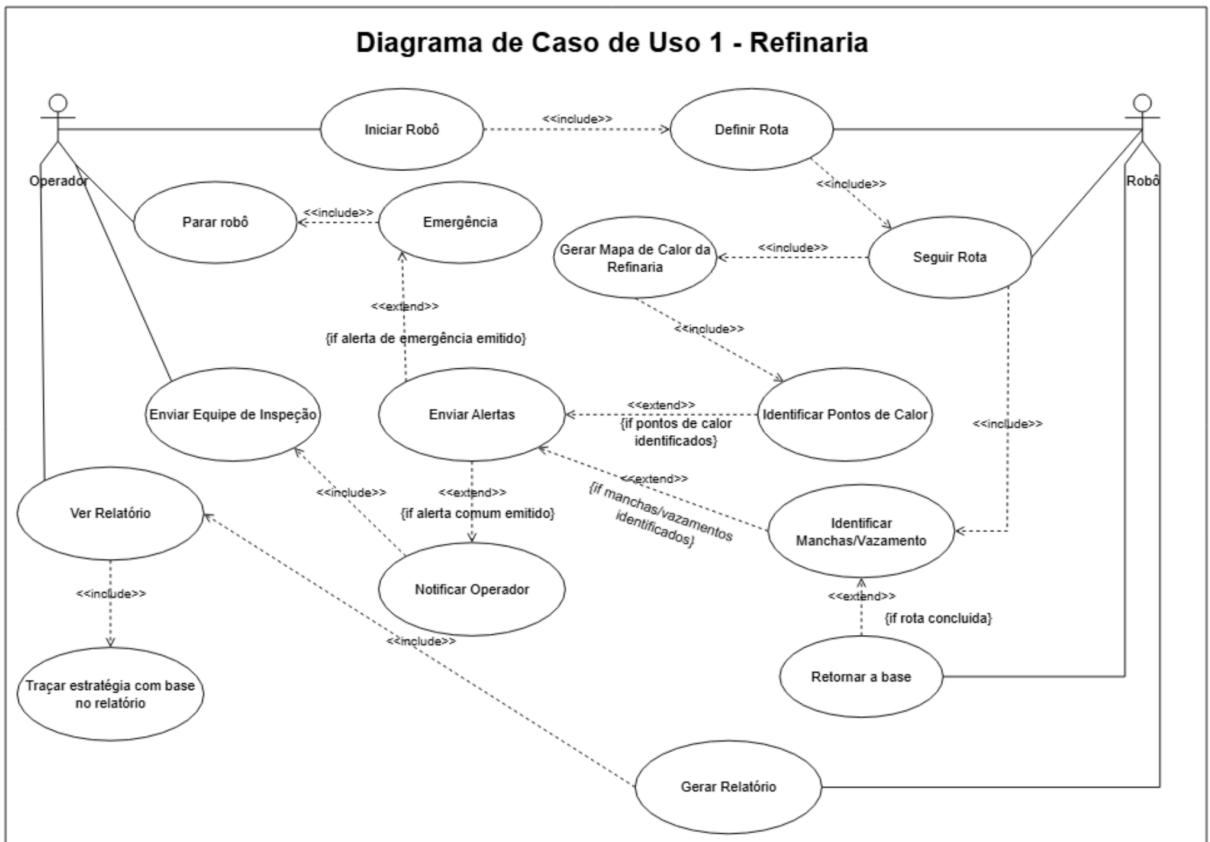


Figura 10 - Diagrama de Caso de Uso para cenário de Refinaria.

3.2 Monitoramento de centro de distribuição



Figura 11 - Demonstração do robô operando em centro de distribuição.

Em um centro de distribuição de e-commerce, há uma constante movimentação de operadores e empilhadeiras carregadas de paletes de produtos. A operação das empilhadeiras devem ser supervisionadas, impedindo o bloqueio das rotas das máquinas. Produtos críticos, como inflamáveis, devem ser monitorados constantemente, evitando acidentes.

Uma frota de robôs móveis pode ser empregada para patrulhar corredores e áreas críticas entre as operações, em horários regulares e fora do horário de funcionamento das empilhadeiras. A câmera do robô, com uma IA embarcada, é capaz de identificar pessoas sem colete refletivo, capacete ou máscara. Além disso, ela é capaz de detectar caixas fora do padrão de empilhamento ou obstáculos no caminho de empilhadeiras. O sensor de temperatura identifica zonas com possível início de foco de incêndio, e emite alertas aos operadores através de sua IHM. Semelhantemente ao caso de uso 1, os alarmes podem ser comuns, que implica na detecção de anomalias por parte do robô, ou podem ser de emergência, para o caso de falha ou problema de funcionamento no robô. Ao final do dia, sua interface gera um relatório contendo todas as anomalias detectadas. O operador deve ter acesso ao relatório gerado pelo robô a fim de traçar estratégias e comandos adaptados aos resultados obtidos.

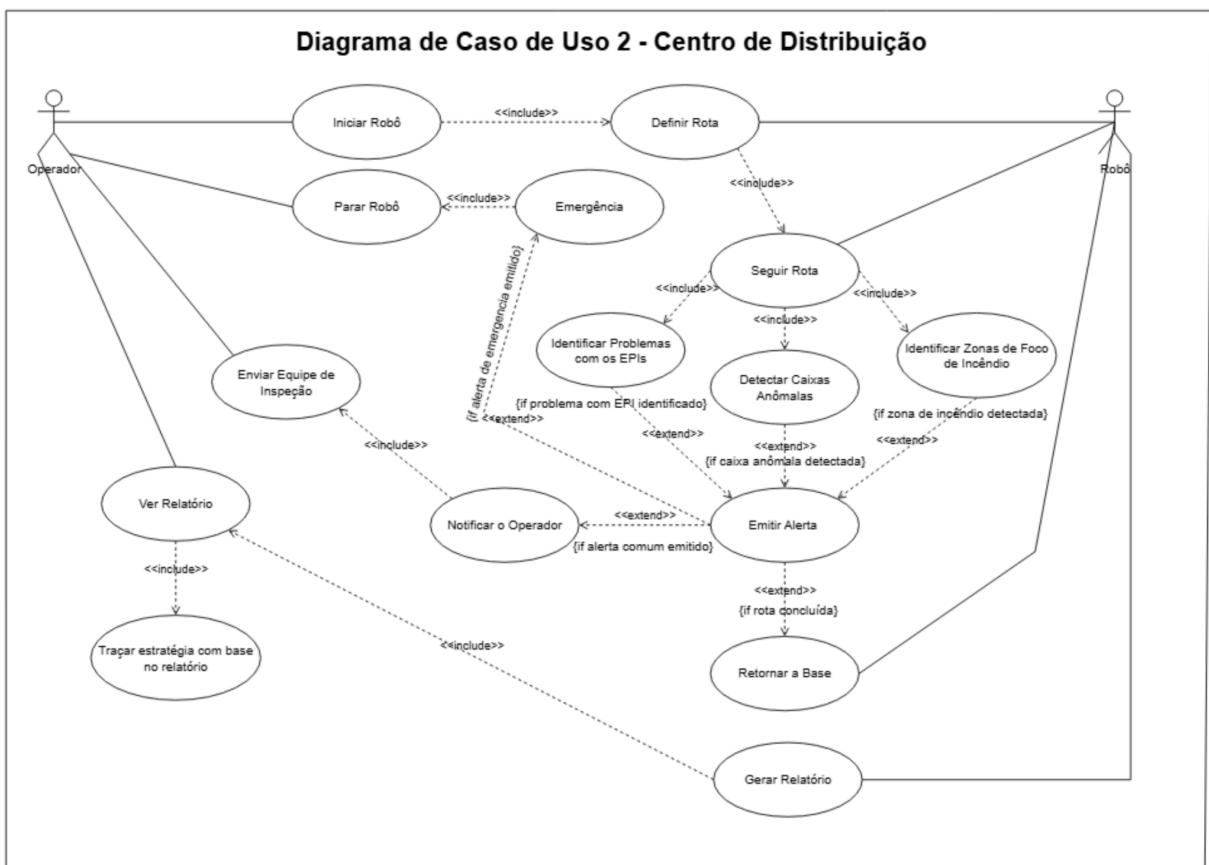


Figura 12 - Diagrama de Casos de Uso para cenário de Centro de Distribuição.

4 Especificações e Requisitos

4.1 Requisitos Gerais

G - 01

O desenvolvimento do projeto RAAI deve ser organizado em sprints de curta duração, com entregas incrementais e reuniões de acompanhamento periódicas, seguindo a metodologia Scrum.

Descrição

Aconteceram sprints de 2 semanas com planejamento, revisão e retrospectiva documentadas. Cada sprint deve possuir como *definition of done* uma funcionalidade tangível (ex: navegação básica, leitura de sensores, IHM inicial).

Possíveis Erros

1. Entregas incompletas no final da sprint.
2. Acúmulo de tarefas.

Abordagem para Tratamento dos Erros

O scrum master deve, em conjunto com o product owner, repriorizar o backlog e dividir tarefas longas em subtarefas executáveis. Os atrasos devem ser registrados e o tempo de entrega da equipe reestimado.

G - 02

Todos os módulos do projeto devem ser desenvolvidos de forma modular e integrados continuamente para evitar incompatibilidades tardias.

Descrição

Cada subsistema possui softwares específicos que interagem com o código-fonte do robô. O sistema deve estar atualizado e mantido em um repositório privado no Github, com controle de versão e branchs. Toda a comunicação entre subsistemas deve ser feita através de interface ROS 2.

Possíveis Erros

1. Diferentes versões do código-fonte em subsistemas distintos
2. Não realização de testes unitários

Abordagem para Tratamento dos Erros

Revisões de subsistemas devem ser realizadas a cada 2 sprints, sendo avaliada a integração dos resultados atuais com as entregas passadas. A documentação tem que ser alimentada a cada entrega e diagramas atualizados. Deve-se buscar automatizar a rotina de testes de integração.

4.2 Requisitos de Eletrônica

ELEC - 01 Acionamento dos Motores - Requisito Funcional

O sistema eletrônico deve acionar quatro motores DC de 25 W/4 kgf·cm por meio de dois drivers BTS7960, garantindo controle individual de velocidade e sentido de rotação.

Descrição

O acionamento dos motores utilizará PWM a 20 kHz gerado pelo RP2040 em quatro canais independentes. A corrente de armadura é monitorada via shunts de $0,01\ \Omega$ e ADC interno (12 bits).

Possíveis Erros

1. Sobre-corrente ($> 30\ A$) ou curto nos terminais.
2. Falha no sinal PWM (ruído ou perda).
3. Driver BTS7960 aquecendo além de $100\ ^\circ C$.

Abordagem para Tratamento dos Erros

Em caso de sobre-corrente, o controlador deve interromper o PWM e desabilitar o pino de enable do canal afetado. A falha deve ser registrada e enviada ao módulo de robótica. Após 5 segundos, tente religar os motores. Em caso de falhas sucessivas, interromper a operação até intervenção do operador.

ELEC - 02 Feedback de Velocidade - Requisito Funcional

O sistema deve estimar a velocidade de cada roda usando encoders Hall de dois canais.

Descrição

O sistema utilizará os sensores encoders para determinar a velocidade e o sentido de rotação dos motores a partir dos pulsos em quadratura emitidos pelo sensor.

Possíveis Erros

1. Perda de pulsos por ruído eletromagnético.
2. Conectores de encoder soltos.
3. Overflow de contadores.

Abordagem para Tratamento dos Erros

Em caso de leituras muito ruidosas, deve ser implementado um filtro digital para tratamento das leituras. Se a falha se repetir, rejeitar a leitura e estimar a velocidade utilizando a próxima leitura. Um log de manutenção deve ser gerado, indicando o canal e a hora do ocorrido.

ELEC - 03 Controle de Velocidade e Corrente - Requisito Funcional

O módulo controlador dos motores deve manter a velocidade e a corrente de cada motor dentro de uma margem de erro de 2% do set-point fornecido pelo módulo de robótica.

Descrição

O controle deve ser implementado no microcontrolador RP2040 e ser executado a 500Hz. A arquitetura de controle escolhida é um PI em cascata de corrente e velocidade para manter estas variáveis dentro do limite de operação da máquina.

Possíveis Erros

1. Referência ausente (Falha no envio no módulo de robótica).
2. Saturação prolongada (> 3 s) indicando carga excessiva.

Abordagem para Tratamento dos Erros

Em casos de não recebimento do setpoint pelo módulo de robótica a mais de 5 segundos, o controlador dos motores deve interromper a operação e gerar um log de manutenção para o operador.

ELEC - 04 Armazenamento de Energia - Requisito Funcional

O sistema deve alimentar motores e eletrônica a partir de um pack Li-ion 24 V / 10 Ah, com proteção contra subtensão e sobrecorrente.

Descrição
O sistema irá possuir um Sistema de Gerenciamento de Baterias (<i>Battery Management System - BMS</i>) com capacidade de fornecer os 12V e 8A necessários para a operação nominal dos motores. O sistema deve possuir um limite de 20V e 30A, para encerrar a operação no caso de atingir esses limites.
Possíveis Erros
<ol style="list-style-type: none"> 1. Queda de tensão abaixo de 21 V sob carga. 2. Temperatura de pack > 60 °C.
Abordagem para Tratamento dos Erros
Em caso de subtensão, o BMS deve emitir um alerta sonoro para que o operador interrompa a missão do robô e o leve para a estação de carregamento.

ELEC - 05 Interface de comunicação com módulo de robótica - Requisito Funcional
O RP2040 deve trocar comandos e telemetria com a Jetson Orin em tempo real.
Descrição
O controlador dos motores enviará os dados de telemetria, como corrente dos motores, tensão da bateria e velocidade das rodas. A Jetson deve enviar os novos setpoints de velocidade, calculados pelo módulo de robótica embarcado.
Possíveis Erros
<ol style="list-style-type: none"> 1. Perda de comunicação entre módulos
Abordagem para Tratamento dos Erros
O sistema deve entrar em modo de IDLE e sinalizar com um LED vermelho. O sistema deve realizar 3 tentativas de realizar a conexão entre os módulos. Após isto, em caso negativo, o sistema deve gerar um Log de manutenção para os operadores.

ELEC - 06 Proteção contra ruído eletromagnético - Requisito Funcional
O robô deve possuir imunidade a interferências eletromagnéticas em sinais de sensores e comunicações
Descrição
O robô deve apresentar proteção contra ruídos a sinais de sensores, como os de efeito Hall ou shunts de corrente, utilizando filtros digitais e analógicos, bem como utilizar o protocolo RS-485 com isolamento galvânico para a troca de dados entre a Jetson Orin e o RP2040.
Possíveis Erros
<ol style="list-style-type: none"> 1. Leitura inconsistente de encoders e shunts. 2. Corrupção de pacotes na comunicação serial.
Abordagem para Tratamento dos Erros
Utilizar o filtro de Kalman para estimar valores válidos quando ruído for detectado nos sensores e

gerar log de manutenção visando diagnóstico.

ELEC - 07 Gerenciamento térmico - Requisito Não Funcional

O robô deve monitorar a temperatura de componentes críticos para evitar danos por superaquecimento.

Descrição

O sistema deve realizar monitoramento contínuo via sensores NTC dos drivers, motores e das baterias.

Possíveis Erros

1. Superaquecimento de drivers ($>100^{\circ}\text{C}$), motores ($>80^{\circ}\text{C}$) ou bateria ($>60^{\circ}\text{C}$).
2. Falha na leitura dos sensores NTC.

Abordagem para Tratamento dos Erros

Caso a temperatura atinja 90% da temperatura limite, logs de alerta devem ser enviados e a velocidade do robô deve ser reduzida. Por outro lado, caso a temperatura atinja o limite máximo para quaisquer um dos três componentes, o robô deve desligar imediatamente e emitir um sinal sonoro.

ELEC - 08 Sistema de LEDs - Requisito Não Funcional

O sistema deve possuir LEDs indicadores de estado operacional (normal, erro, baixa bateria, comunicação perdida).

Descrição

Módulo de LEDs RGB gerenciado via GPIOs do RP2040, com padrões definidos para os diferentes estados.

Possíveis Erros

1. Falha na transição entre os estados.
2. LEDs queimados ou com defeito.

Abordagem para Tratamento dos Erros

Verificação de funcionamento dos LEDs no boot. Em caso de erro, sinalizar via alerta sonoro.

4.3 Requisitos de Robótica e Navegação

NAV - 01 Navegação autônoma no ambiente fabril - Requisito Funcional

O robô deve ser capaz de se locomover de forma autônoma entre pontos pré definidos da planta industrial.

Descrição

O robô utiliza rotas predefinidas pelos operadores para navegar por pontos de interesse no mapa global da fábrica, através de implementação utilizando o Nav2 Stack do ROS 2.

Possíveis Erros

1. Falha na geração de rotas.
2. Falha em parada ao chegar no destino.

Abordagem para Tratamento dos Erros

O robô realizará novas tentativas de planejamento caso não seja encontrado nenhum caminho viável. Após 3 falhas, o robô deve desligar os motores e enviar um log de falha para a central de operações e solicitar intervenção remota.

NAV - 02 Evitar áreas marcadas - Requisito Funcional

O robô deve evitar navegar por áreas marcadas como inseguras.

Descrição

O robô receberá no seu stack de navegação uma área delimitada no mapa global que ele deve evitar navegar, colocado-a no seu *costmap*.

Possíveis Erros

1. O robô navega por uma área proibida

Abordagem para Tratamento dos Erros

Ao detectar que está sobre uma área proibida, o robô deve marcar aquele ponto em seu mapa local e retornar ao último ponto conhecido fora da área delimitada.

NAV - 03 Evitar obstáculos - Requisito Funcional

O robô deve ser capaz de detectar e desviar de obstáculos inesperados em tempo real, garantindo segurança durante a navegação.

Descrição

O robô utiliza o sensor LiDAR para atualizar o seu mapa local continuamente. O robô deve ser capaz de detectar objetos que estejam sobre a sua rota a menos de 20 cm de distância e possuir um tempo máximo de reação de 300 ms após detecção para desviar do obstáculo.

Possíveis Erros

1. Objeto não detectado a tempo.
2. Desvio forçado gera saída do trajeto ou colisão lateral.

Abordagem para Tratamento dos Erros

Se houver falha na detecção, o robô deve reduzir a velocidade. Caso o desvio gere travamento, o robô deve parar, voltar e replanejar localmente uma nova rota.

NAV - 04 Geração e atualização de mapa local - Requisito Funcional

O sistema deve ser capaz de construir e atualizar um mapa local do ambiente a partir de sensores a bordo, permitindo operações em ambientes parcialmente desconhecidos.

Descrição

O robô utiliza a toolbox de SLAM com atualização do mapa produzido via leituras do LiDAR. Os mapas locais gerados serão enviados a cada 30s para a estação-base.

Possíveis Erros

1. Falhas de geração dos mapas

Abordagem para Tratamento dos Erros

Se o mapa gerado tiver um deslocamento na posição real do robô, ele deve ser descartado.

NAV - 05 Localização do robô - Requisito Funcional

O robô deve estimar continuamente sua localização, com erro máximo de 10 cm da sua posição real no mapa global.

Descrição

O robô usará os dados de odometria e LiDAR para estimar sua localização no mapa global do ambiente. A posição do robô no [tf2](#) deve ser atualizado com atraso máximo de 5 ms.

Possíveis Erros

1. Erro acumulativo por falha de calibração de sensores .

Abordagem para Tratamento dos Erros

Se o erro estimado de posição ultrapassar 50 cm, o robô deve retornar à última posição segura. Em caso de detecção de falha nos sensores, realizar a sua calibração.

NAV - 06 Agendamento de Tarefas- Requisito Funcional

O robô deve ser capaz de realizar patrulhas periódicas, de modo a seguir um cronograma de atividades definido remotamente pela estação base.

Descrição

O módulo de robótica deve verificar a agenda de patrulhas e iniciar missões automaticamente nos horários programados. O tempo do sistema deve ser sincronizado via NTP com o servidor a cada 6 horas para garantir precisão.

Possíveis Erros

1. Perda da hora do sistema após reinício.
2. Missão iniciada com atraso superior a 30 s.
3. Falha na sincronização com servidor NTP.

Abordagem para Tratamento dos Erros

Em caso de ausência de horário válido, o robô deve solicitar sincronização manual via IHM. Caso não seja possível sincronizar com NTP, o sistema deve operar com hora local e sinalizar imprecisão ao operador.

NAV - 07 Log de anomalias visuais detectadas - Requisito Funcional

O módulo de robótica deve enviar um log estruturado de cada anomalia visual detectada durante a patrulha.

Descrição

Para cada evento visual detectado como anomalia (ex: mancha de óleo, pessoa sem EPI, obstáculo fora do padrão), o sistema deve transmitir: tipo de anomalia, posição estimada da anomalia, *timestamp* e imagem da anomalia.

Possíveis Erros

1. Falha na compressão da imagem.
2. Duplicação de logs para o mesmo evento.

Abordagem para Tratamento dos Erros

Agrupar eventos com mesma posição (± 10 cm) e tipo em um intervalo de 10 s. Logs devem ser salvos em CSV local se o envio falhar.

NAV - 08 Recalibração automática de sensores - Requisito Não Funcional

O robô deve realizar uma calibração periódica dos sensores de navegação após as operações.

Descrição

O robô deve realizar uma rotina de calibração do LIDAR e da odometria a cada 1 semana, bem como ajustar os parâmetros do SLAM com base nos dados coletados.

Possíveis Erros

1. Falha na execução da rotina de calibração.
2. Falha no ajuste ou coleta de dados.

Abordagem para Tratamento dos Erros

Em caso de erro o robô deve enviar os logs e retornar a base, solicitando calibração manual por parte dos operadores/engenheiros.

NAV - 09 Conservação de Energia - Requisito Não Funcional

O robô deve otimizar o consumo energético durante patrulhas longas, levando em consideração o nível da bateria e a distância percorrida.

Descrição
O robô deve ajustar dinamicamente sua velocidade máxima e a frequência de varredura do LIDAR, de modo a economizar o máximo possível o desgaste nas baterias, bem como no aumento de sua temperatura.
Possíveis Erros
<ol style="list-style-type: none"> 1. Velocidade máxima para o nível de bateria atual ultrapassada 2. Tempo de descarga menor que o tempo de distância até a estação de carregamento mais próxima.
Abordagem para Tratamento dos Erros
O robô deve retornar a base pelo menor caminho e ser reconfigurado quanto a otimização de bateria.

4.4 Requisitos de Software

SOF - 01 Detecção visual de anomalias - Requisito Funcional
O sistema de software embarcado deve identificar anomalias visuais, como manchas de óleo no chão, pessoas sem EPI ou obstáculos fora do padrão.
Descrição
A detecção de anomalias será feita com modelo YOLOv8, rodando no Jetson Orin. As imagens são capturadas pela câmera RGB a 30 fps. As anomalias são mapeadas com rótulos semânticos.
Possíveis Erros
<ol style="list-style-type: none"> 1. Baixa luminosidade ou reflexos comprometendo a detecção. 2. Alto uso de CPU/GPU bloqueando processamento em tempo real.
Abordagem para Tratamento dos Erros
A câmera deve ser configurada de forma a se adaptar a luminosidade e reflexos na lente. Em caso de alto uso de hardware, a câmera deve diminuir a taxa de frames por segundo.

SOF - 02 Armazenamento e recuperação de dados de sensores - Requisito Funcional
O sistema deve registrar os dados coletados de sensores em um banco de dados local.
Descrição
O sistema deve gravar os dados de leitura a cada 1 s no banco de dados local e sincronizar com a estação-base a cada 60 s. Cada entrada no banco deve conter timestamp, ID do robô, tipo de dado e valor.
Possíveis Erros

- | |
|------------------------------|
| 1. Falta de espaço em disco. |
|------------------------------|

Abordagem para Tratamento dos Erros

O sistema deve realizar uma limpeza de dados com data de criação há 30 dias.

SOF - 03 Interface para operadores - Requisito Funcional

O sistema deve disponibilizar uma interface interativa na estação-base.

Descrição

A interface de interação com os operadores deve ser acessível por desktop, para visualização de dados, mapas e alertas. A interface recebe atualizações em tempo real via WebSocket.

Possíveis Erros

1. Falha no carregamento de dados.
2. Perda de conexão com o WebSocket.

Abordagem para Tratamento dos Erros

A interface deve permitir recarregamento manual pelo usuário para reconexão do WebSockets e com o banco de dados da aplicação.

SOF - 04 Geração automática de relatórios - Requisito Funcional

O sistema da estação-base deve gerar relatórios diários sobre as anomalias detectadas, com dados e recomendações baseadas em problemas pré-definidos.

Descrição

O sistema utiliza os dados coletados durante o dia e gera um relatório com visualizações das variáveis monitoradas na fábrica e logs de anomalias detectadas.

Possíveis Erros

1. Geração de resposta inconsistente ou vazia.

Abordagem para Tratamento dos Erros

Deve-se garantir que houve a captura de dados no dia solicitado para a geração do relatório. A estrutura do relatório deve ser validada antes da implementação da funcionalidade.

SOF - 05 Diagnóstico remoto do sistema - Requisito Funcional

O sistema deve permitir o acesso remoto em tempo real ao status operacional de todos os subsistemas do robô.

Descrição

O robô deve possuir um Dashboard web com visualização hierárquica dos subsistemas (Eletrônica, Robótica, Sensores e Comunicação), onde deve exibir métricas críticas atualizadas a cada 2 segundos (temperatura de componentes, status de conexão, log de erros).

Possíveis Erros
<ol style="list-style-type: none"> 1. Inconsistência nos dados exibidos (ex: valores congelados). 2. Latência na atualização dos dados.
Abordagem para Tratamento dos Erros
Realizar testes de tratamento de dados a fim de encontrar o motivo das inconsistências e realizar stress e práticas de desempenho a fim de monitorar a situação da rede.

SOF - 06 Simulador do robô - Requisito Não Funcional
O sistema da estação base deve possuir um modelo do robô, capaz de simular as dimensões físicas, comportamento, operação, controle e comunicação do robô.
Descrição
O sistema deve conter um simulador do robô a fim de observar e entender as dinâmicas que regem seu comportamento, visando a implementação de práticas de controle ou experimentos, funcionando como gêmeo digital.
Possíveis Erros
<ol style="list-style-type: none"> 1. Para o desenvolvimento de um simulador, parte-se de um modelo matemático-físico que nunca consegue ser exatamente igual ao elemento real, perdendo-se exatidão e precisão.
Abordagem para Tratamento dos Erros
Pode-se estimar o modelo do robô com base nos parâmetros de maior interesse, de modo a essa diferença ser pequena em relação ao mundo real.

SOF - 07 Simulação de falhas - Requisito Não Funcional
O sistema deve incorporar um módulo de injeção controlada de falhas para validação de robustez em ambientes de desenvolvimento e homologação.
Descrição
Permitir a avaliação do comportamento do robô em condições adversas sem riscos operacionais, garantindo estabilidade (testando os limites dos algoritmos de controle) e robustez.
Possíveis Erros
<ol style="list-style-type: none"> 1. Configuração incorreta do ambiente. 2. Corrupção de dados. 3. Ativação acidental.
Abordagem para Tratamento dos Erros
Emprego de senha para a configuração dessa aba de simulação e aplicação de algoritmos de rotina para análise dos dados.

SOF - 08 Estruturação de logs - Requisito Não Funcional

O sistema deve implementar um padrão unificado para geração, armazenamento e recuperação de logs em toda a infraestrutura de software da central.

Descrição

Utilizar de formato JSON para estruturação dos logs, implementando os níveis como da forma a seguir:

Níveis: DEBUG (diagnóstico), INFO (operações), WARNING (anomalias), ERROR (falhas), CRITICAL (emergências)

Possíveis Erros

1. Saturação de armazenamento.
2. Perda de log durante falhas.

Abordagem para Tratamento dos Erros

Limpeza de logs com base em sistema de prioridade e tempo e padronização entre os logs para cada subsistema.

4.5 Requisitos de Comunicação

COM - 01 Comunicação sem fio entre robô e estação-base - Requisito Não Funcional

O sistema de comunicação deve permitir a troca contínua de dados entre o robô e a estação-base via rede sem fio, com baixa latência e cobertura no ambiente industrial.

Descrição

O sistema estará conectado via Wi-Fi na banda de 5 GHz e cobertura por toda planta industrial. Os pacotes de telemetria são enviados a cada 1 s via ROS2. O tempo de resposta da estação-base a comandos críticos é menor que 200 ms.

Possíveis Erros

1. Perda de sinal Wi-Fi.
2. Perda de pacotes
3. Atraso na entrega.

Abordagem para Tratamento dos Erros

O robô deve manter um buffer dos dados de sensores para reenvio após reconexão. Em caso de persistir a falha na conexão por 30 segundos, o robô deve desligar os motores e parar.

COM - 02 Comunicação interna entre módulos do robô - Requisito Funcional

Os módulos eletrônicos embarcados (RP2040, Jetson Orin) devem comunicar-se entre si utilizando comunicação serial, permitindo troca de comandos de controle e dados de sensores.

Descrição

A comunicação entre o módulo controlador dos motores e o controlador do robô devem se comunicar via serial, através do micro_ros. As mensagens trocadas a cada 20 ms contém set-points de velocidade, feedback dos encoders, e alertas de falha.

Possíveis Erros

1. Comando inválido ou fora do intervalo esperado.

Abordagem para Tratamento dos Erros

Os comandos recebidos devem ser validados, eliminando aqueles que estão fora do intervalo permitido.

COM - 03 Comunicação via MQTT - Requisito Funcional

O servidor que armazena a informação das missões deve ser capaz de receber os dados de telemetria do robô e enviar comandos ao mesmo via protocolo MQTT.

Descrição

Garantir transmissão e leitura confiável com baixa latência das leituras de sensores para armazenamento no banco de dados via protocolo MQTT. A arquitetura seguirá um modelo publisher-subscriber, onde cada robô atua como publisher, enviando dados para um broker MQTT configurado com criptografia TLS 1.3. O servidor central, por sua vez, atuará como subscriber, recebendo essas mensagens e armazenando-as no banco de dados operacional.

Possíveis Erros

1. Broker indisponível por >30 segundos.
2. Saturação do banco de dados

Abordagem para Tratamento dos Erros

O robô deve possuir um buffer local para retransmissão pós falha e o armazenamento do banco de dados deve ser observado constantemente pelo operador.

COM - 04 Comunicação segura - Requisito Funcional

Garantir que toda comunicação entre o robô e o servidor utilize autenticação por certificados digitais.

Descrição

Implementar autenticação mútua por meio da criptografia TLS 1.3 com rotação automática de certificados a cada 90 dias.

Possíveis Erros

1. Certificado expirado ou inválido.
2. Rotação não realizada.

Abordagem para Tratamento dos Erros

Bloquear comunicações sensíveis a ataques e alterações (módulos de controle) e alertar o

operador via IHM.

COM - 05 Logs de falha de comunicação - Requisito Não Funcional

Registrar sistematicamente todas as falhas na comunicação entre subsistemas.

Descrição

Registrar sistematicamente todas as falhas na comunicação entre subsistemas, registrando o momento em que ocorreram, os módulos envolvidos (origem/receptor), o tipo de falha e o último pacote enviado. O log gerado deve ser disposto no banco de dados.

Possíveis Erros

1. Falhas não capturadas.
 2. Sobrecarga de logs na base de dados.

Abordagem para Tratamento dos Erros

Algoritmo de limpeza de logs de acordo com níveis de severidade e geração de relatório, bem como análise de padrões de erro.

Os requisitos expostos podem ser visualizados no diagrama de requisitos na Figura 13

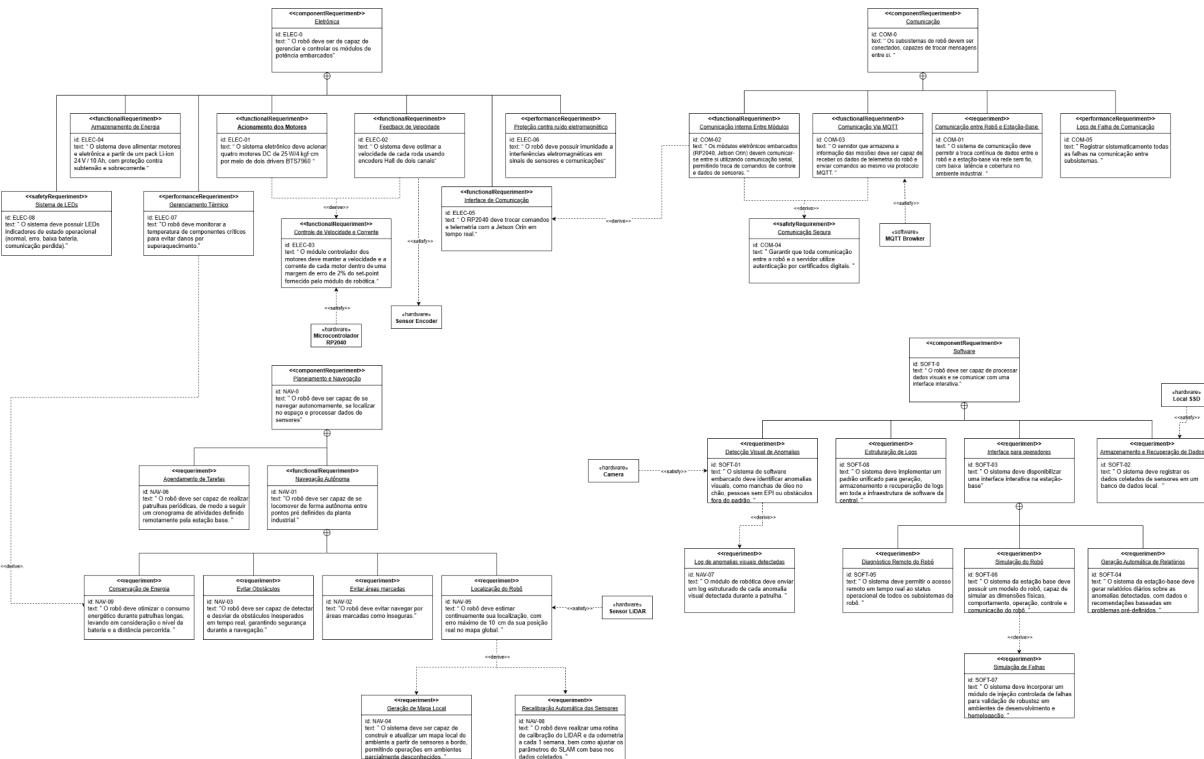


Figura 13 - Diagrama de Requisitos do Robô RAAI

5 Recursos necessários

Descrição	Módulo	Custo
Jetson Orin	Controlador Robô	R\$ 2100,00
LiDAR 3D	Controlador Robô	R\$ 1600,00
Câmera RGB	Controlador Robô	R\$ 230,00
Sensor Temperatura	Controlador Robô	R\$ 500,00
Sensor Distância IR	Controlador Robô	R\$ 80,00
Chassi de Alumínio	Controlador Motor	R\$ 350,00
RP 2040	Controlador Motor	R\$ 50,00
4 x Motores	Controlador Motor	R\$ 380,00
2x CI Motor Driver BTS7960	Controlador Motor	R\$ 90,00
4 x Sensor Encoder	Controlador Motor	R\$ 200,00
Roteador Longo Alcance	Comunicação	R\$ 300,00

Total estimado por robô: R\$ 5.870

6 Plano de Projeto

6.1 Metodologia Scrum

Como definido no Requisito Geral 1 (**G - 01**) na seção de Especificações e Requisitos do projeto, o desenvolvimento do sistema RAAI deve ser organizado seguindo a metodologia Scrum, com entregas incrementais desenvolvidas em sprints de curta duração e reuniões de acompanhamento periódicas entre os times envolvidos.

A metodologia Scrum é composta por três fases majoritárias:

1. Fase Inicial: Define o planejamento macro, onde é definido os objetivos a serem alcançados e a arquitetura de alto nível.
2. Desenvolvimento: Ciclos de desenvolvimento (*sprints*) que entregam incrementos do sistema definidos a cada sprint.
3. Fechamento: Completa o projeto, recebe aceite do cliente e avalia as lições aprendidas.

Na figura abaixo, um diagrama explicita o princípio fundamental da metodologia Scrum, dividindo o ciclo de sprint em quatro fatores: **avaliação, seleção, desenvolvimento e revisão**.

Partindo de um planejamento inicial, onde será definido as especificações e requisitos, objetivos, ideia de mercado e arquitetura macro bem definida, é realizado o processo de **avaliação** e **seleção** para definição de cada uma das *sprints*. Isto é, será definido (seleção) qual o tema a ser abordado durante o ciclo com base na necessidade e no estado atual do projeto (avaliação), sua duração e os atores que irão agir naquele *sprint*.

Após definição da *sprint*, é dado início ao processo de **Desenvolvimento**, caracterizado pela implementação da funcionalidade definida no ciclo por parte dos atores envolvidos. Dado o fim desta fase, é feito uma **Revisão** do que foi desenvolvido, à fim de confirmar que o objetivo da *sprint* foi cumprido dentro do prazo estimado.

Ao fim de cada *sprint*, o ciclo se repete, ou seja, é feito uma avaliação do que foi desenvolvido do sistema e o que falta ser desenvolvido para nova seleção de implementação para definição de uma nova *sprint*.

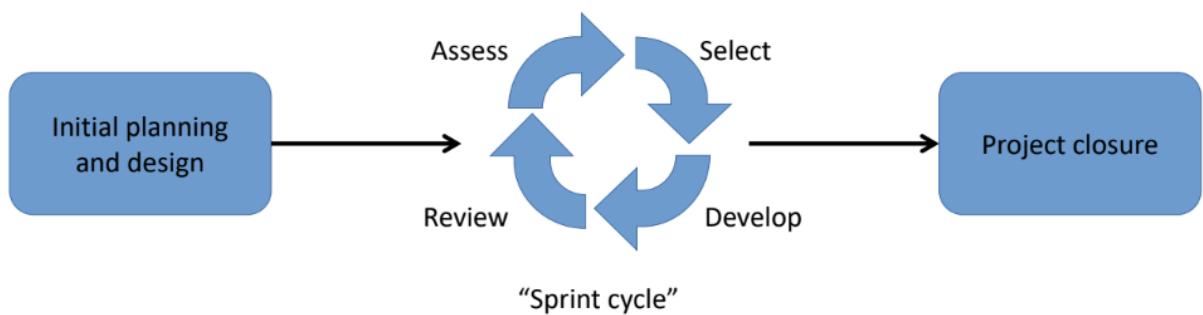


Figura 14 - Metodologia Scrum

As *sprints* possuem duração de 2 a 4 semanas, com reuniões *daily* entre as equipes de desenvolvimento para discutir a implementação e adaptações a serem realizadas. O plano de projeto define previamente qual funcionalidade será implementada, os atores e a duração de cada *sprint*, funcionando como um mapa a ser seguido pelos desenvolvedores do projeto.

6.2 Fases do Projeto e Definição das Sprints

Após a fase de concepção do RAAI, composta por análise de mercado, identificação dos objetivos a serem alcançados com o produto, casos de uso do produto e uma visão geral da arquitetura utilizada, a fase seguinte é norteada pelo levantamento de requisitos e de recursos necessários. Essas fases correspondem à fase inicial prevista na metodologia Scrum. Os atores envolvidos nessa fase são os *Product Owners*: Mateus Pincho de Oliveira e Yan Abrantes Cavalcante.

Tendo definido os requisitos e recursos a serem utilizados, a fase seguinte é caracterizada pela definição das sprints (realizada pelo Scrum Master) com base na necessidade e importância de cada funcionalidade dentro do escopo do projeto e no cumprimento dos requisitos levantados. Essa fase é seguida pelo desenvolvimento das sprints (realizada pela equipe de desenvolvimento) com base no ciclo definido na metodologia Scrum, em que é posto em prática pela equipe de desenvolvimento de cada módulo do RAAI.

Em seguida, têm-se a fase de testes do sistema (realizada pela equipe de *Quality Assurance*), caracterizada por avaliar se o sistema RAAI cumpre com os seus requisitos e com os seus objetivos. Além disso, essa fase é essencial para encontrar defeitos de implementação, seja na parte de *software*, *hardware* ou comunicação. Essa fase dá início ao fechamento do projeto, pois é nela em que o sistema começa a se tornar um produto para o mercado. Por fim, temos a fase de conclusão, visando o aceite do cliente e a disponibilização do produto no mercado. Todos esses processos da metodologia Scrum são avaliados e coordenados pelo *Scrum Master*.

- Equipe de Desenvolvimento:

Equipe	
Product Owner	Mateus Pincho de Oliveira e Yan Abrantes Cavalcante
Scrum Master	Yan Abrantes Cavalcante
Desenvolvedores Eletrônica	Engenheiros Eletricistas/Técnicos
Desenvolvedores Navegação	Engenheiros de Controle e Automação/Técnicos
Desenvolvedores de Software	Engenheiros de Software
Integração da Comunicação	Engenheiros de Computação/Engenheiros Eletricistas

- Etapas do Projeto:

Etapas do Projeto	
Concepção do Produto	Análise de mercado, identificação dos objetivos a serem alcançados com o produto, casos de uso do produto e visão geral da arquitetura utilizada.
Levantamento de Requisitos e Recursos	Definição dos requisitos do sistema e dos recursos necessários para sua implementação.
Definição das Sprints	Criação das sprints com base nos requisitos e na prioridade do projeto.
Desenvolvimento das Sprints	Desenvolvimento associado a sprint selecionada pela equipe. Inclusão de reuniões diárias, revisão do que foi desenvolvido durante a sprint e revisão por parte do Product Owner ao fim de cada sprint.
Testes do Sistema	Avaliar se o sistema RAAI cumpre com os seus requisitos e objetivos, além de identificar e corrigir erros.

Fechamento	Finalização do projeto e dos trâmites visando o aceite do cliente e avaliação das lições aprendidas.
-------------------	--

Tabela 2 - Definição das etapas do projeto

- Cronograma Macro:

Cronograma Macro		
Especificação dos Requisitos	1 semana	Proposição/Avaliação de requisitos.
Configuração do Hardware e Eletrônica	2 meses	Aquisição, instalação e configuração dos dispositivos de hardware e eletrônica.
Desenvolvimento do Módulo de Robótica	1 mês	Configuração das plataformas para o módulo de navegação.
Desenvolvimento dos algoritmos de Visão Computacional e Software	3 meses	Desenvolvimento dos algoritmos de visão computacional e demais funcionalidades de software.
Configuração de Comunicação e Segurança entre os módulos e dispositivos	2 meses	Integração entre os módulos e dispositivos do sistema.
Integração do Sistema e Testes de Validação e Verificação	1 mês	Verificar se o produto atinge os requisitos via testes de validação e verificação. Eliminar erros.
Feedback do Cliente e Tratamento de Erros	2 semanas	Avaliação de feedbacks do usuário final. Melhorias no sistema.
Documentação	2 semanas	Documentação da versão final do projeto.
Encerramento	1 semana	Revisão na documentação e conclusão do projeto.

Tabela 3 - Definição do cronograma macro do projeto

- Ferramentas utilizadas:

FERRAMENTAS	
IDE	Visual Studio Code
Linguagem	C++ e Python
SGBD	MySQL
Controle de Versão	Git
Gerenciamento e Repositório	GitHub
Testes e Integração	Google Test (gtest)
Documentação do Código	Doxygen

Tabela 4 - Definição das ferramentas a serem utilizadas no projeto

7 Verificação e Validação dos Requisitos

Título	Teste de acionamento dos Motores	Código	T01
Pré-Condição	Robô energizado e motores conectados.		
Execução	Enviar comandos PWM via Jetson e verificar giro e sentido de cada motor.		
Critério de Aceitação	Todos os motores respondem corretamente a comandos de direção e velocidade.		
Rastreabilidade	ELEC-01, ELEC-03		

Título	Verificação de Feedback dos Encoders	Código	T02
Pré-Condição	Robô com motores acionados e rodas girando livremente.		
Execução	Comparar valores estimados dos encoders com leitura real por tacômetro digital.		
Critério de Aceitação	Velocidade e direção detectadas condizem com movimento real.		
Rastreabilidade	ELEC-02		

Título	Validação do Controle dos Motores	Código	T03
Pré-Condição	Definir setpoint fixo no microcontrolador.		
Execução	Executar programa de controle e registrar os logs do controlador.		
Critério de Aceitação	Corrente e velocidade dentro da margem de 2% do setpoint.		
Rastreabilidade	ELEC-03		

Título	Teste de Navegação Autônoma	Código	T04
Pré-Condição	Mapa carregado e rota definida.		
Execução	Criar uma rota simulada no ambiente fabril e monitorar execução do Nav2.		

Critério de Aceitação	Robô completa missão sem colisão e para nos pontos corretos.
Rastreabilidade	NAV-01, NAV-03, NAV-05

Título	Detecção de Objeto em Tempo Real	Código	T05
Pré-Condição	Objeto colocado a 20 cm do robô.		
Execução	Colocar objeto inesperado na rota e medir tempo de reação.		
Critério de Aceitação	Robô detecta e desvia do obstáculo em até 1 segundo.		
Rastreabilidade	NAV-03		

Título	Geração de Mapa Local	Código	T06
Pré-Condição	Ambiente com obstáculos.		
Execução	Comparar mapa local gerado com planta conhecida.		
Critério de Aceitação	Mapa reflete corretamente o ambiente e atualiza com movimentos.		
Rastreabilidade	NAV-04		

Título	Estimativa de Posição	Código	T07
---------------	-----------------------	---------------	-----

Pré-Condição	Trajetória conhecida do robô.
Execução	Validar posição do robô no mapa com referência física de marcação no solo.
Critério de Aceitação	Erro de posição abaixo de 10 cm.
Rastreabilidade	NAV-05

Título	Missão Agendada por Horário	Código	T08
Pré-Condição	Agenda definida remotamente.		
Execução	Agendar missão e verificar tempo de início da navegação.		
Critério de Aceitação	Robô inicia missão no tempo correto (± 30 s).		
Rastreabilidade	NAV-06		

Título	Detecção de Anomalias Visuais	Código	T09
Pré-Condição	Presença de manchas/pessoas sem EPI.		
Execução	Anomalias detectadas corretamente com timestamp e tipo.		
Critério de Aceitação	Simular eventos com objetos de teste e verificar registros gerados.		

Rastreabilidade	SOF-01, NAV-07
------------------------	----------------

Título	Armazenamento Local de Dados	Código	T10
Pré-Condição	Robô em operação contínua.		
Execução	Verificar integridade dos dados no banco local após operação do robô.		
Critério de Aceitação	Dados gravados a cada segundo no banco local.		
Rastreabilidade	SOF-02		

Título	Sincronização com Estação Base	Código	T11
Pré-Condição	Robô conectado ao Wi-Fi.		
Execução	Monitorar tráfego de rede com log MQTT.		
Critério de Aceitação	Dados transmitidos a cada 60 s com sucesso.		
Rastreabilidade	SOF-02, COM-01		

Título	Interface com Operadores	Código	T12
Pré-Condição	Robô em missão com estação ativa.		

Execução	Acessar dashboard e verificar atualização de status.
Critério de Aceitação	Dados exibidos em tempo real na interface Web.
Rastreabilidade	SOF-03, SOF-05

Título	Geração de Relatório Diário	Código	T13
Pré-Condição	Dados disponíveis de uma patrulha.		
Execução	Validar relatório PDF exportado da aplicação.		
Critério de Aceitação	Relatório gerado com gráficos, tabelas e imagens.		
Rastreabilidade	SOF-04, NAV-07		

Título	Diagnóstico Remoto	Código	T14
Pré-Condição	Robô e estação-base conectados.		
Execução	Simular erro e verificar tempo de exibição no dashboard.		
Critério de Aceitação	Interface mostra dados críticos atualizados a cada 5 segundos.		
Rastreabilidade	SOF-05		

Título	Simulação do Robô	Código	T15
Pré-Condição	Simulador instalado e configurado.		
Execução	Executar cenário no Gazebo e comparar logs com testes reais.		
Critério de Aceitação	Resposta do modelo simulado dentro de tolerância aceitável.		
Rastreabilidade	SOF-06		

Título	Injeção de Falhas	Código	T16
Pré-Condição	Ambiente de desenvolvimento ativo.		
Execução	Forçar perda de sensor ou comunicação e observar comportamento.		
Critério de Aceitação	Robô responde corretamente à falhas forçadas.		
Rastreabilidade	SOF-07		

Título	Estrutura de Logs	Código	T17
Pré-Condição	Robô em operação com eventos ocorrendo.		
Execução	Validar arquivos gerados no sistema de arquivos local e remoto.		

Critério de Aceitação	Logs gerados e com níveis corretos.
Rastreabilidade	SOF-08, COM-05

Título	Comunicação Interna entre Módulos	Código	T18
Pré-Condição	RP2040 e Jetson Orin conectados via USB.		
Execução	Logar os pacotes trocados e verificar sincronismo.		
Critério de Aceitação	Dados trocados via serial com integridade a cada 20 ms.		
Rastreabilidade	ELEC-05, COM-02		

Título	Comunicação MQTT com Servidor	Código	T19
Pré-Condição	Broker MQTT em operação.		
Execução	Monitorar publicação e subscrição com ferramenta MQTT .		
Critério de Aceitação	Dados publicados e recebidos com latência menor que 500 ms.		
Rastreabilidade	COM-03, COM-04		

Título	Conservação de Energia	Código	T20
---------------	------------------------	---------------	-----

Pré-Condição	Bateria abaixo de 30% durante patrulha.
Execução	Simular patrulha longa e monitorar ajuste automático de velocidade.
Critério de Aceitação	Robô reduz consumo e retorna à base com segurança.
Rastreabilidade	NAV-09, ELEC-04