Arquitetura de Software

> Clique Aqui para baixar o documento.

Nesta seção, apresentamos uma visão abrangente da **arquitetura de software do projeto**, detalhando os componentes essenciais, suas interações e as tecnologias aplicadas. Essa arquitetura foi projetada para garantir escalabilidade, manutenibilidade e eficiência, permitindo que todos os elementos do sistema funcionem em harmonia e cumpram as necessidades operacionais e de monitoramento.

Visão Geral da Arquitetura

A arquitetura de software adotada segue uma abordagem **modular**, na qual cada componente desempenha um papel específico e se comunica por **interfaces bem definidas**. Essa organização modular aprimora a **manutenção**, a **escalabilidade** e a **evolução contínua do sistema**, permitindo integração eficiente e expansões futuras.

Componentes

Os **componentes principais** do sistema estão organizados de forma modular e conectados por meio de interfaces bem definidas. A **Tabela 1** apresenta os componentes detalhados, destacando suas definições e papéis no funcionamento geral do sistema.

Tabela 1: *Componentes principais da arquitetura modular do sistema.*

Componente	Definição	Uso
Mission Planner	Responsável por coordenar e armazenar dados da missão, fornecendo arquivos de log para o Relatório de Voo, que realiza análises pós-voo detalhadas.	Utilizado para fornecer metadados necessários à elaboração de relatórios de voo e permitir ajustes em tempo real.
Pixhawk	Controlador de voo de alta precisão que executa as instruções do piloto e mantém a estabilidade do drone durante a operação.	Fornece dados em tempo real e é o principal responsável pelo controle físico do drone.
Raspberry	Computador de bordo que processa dados e integra sensores, atuando como hub central para a Realidade Aumentada.	Processa dados de telemetria, coordena a interação entre os sistemas e executa a Realidade Aumentada.
Óculos de FPV	Dispositivo que permite ao piloto visualizar o vídeo em primeira pessoa, exibindo a filmagem capturada pelo drone em tempo real.	Utilizado como saída visual crítica para operações de precisão e navegação assistida, ajudando o piloto a fazer ajustes durante o voo.
Realidade Aumentada	Sistema que exibe informações como a composição do vídeo, silhueta de fogo, a mira de previsão de queda e o monitoramento da bateria.	Exibe dados em tempo real nos Óculos de FPV e Notebook, suportando o piloto com feedback vital via protocolo Vtx.
Relatório de Voo	Aplicativo que recebe arquivos de log do Mission Planner e cria um dashboard detalhado com a análise dos dados coletados durante os voos.	Serve como uma ferramenta pós-voo para identificar padrões de consumo de energia, instabilidades de voo e tendências nos dados de temperatura da bateria.

Fonte: *Autoria própria. Todos os direitos reservados.*

Metas e Restrições Arquiteturais

Este tópico apresenta as **metas arquiteturais**, que orientam o design do sistema para atender aos requisitos de **desempenho, escalabilidade, confiabilidade, manutenibilidade e usabilidade**. Além disso, descreve as **restrições** que limitam as escolhas de design, como compatibilidade com hardware e software, facilidade de manutenção e integração com outros sistemas.

Metas

As metas arquiteturais definem os princípios que guiam o desenvolvimento do sistema, assegurando que ele cumpra seus objetivos operacionais e requisitos de qualidade. A **Tabela 2** apresenta essas metas de forma detalhada, destacando os principais tipos e suas descrições.

Tabela 2: *Metas arquiteturais para o design e desenvolvimento do sistema.*

Tipo	Descrição
Desempenho	Garantir que o sistema processe dados em tempo real sem atrasos perceptíveis, especialmente em operações críticas.
Escalabilidade	Projetar a arquitetura para permitir a adição de novos módulos e funcionalidades sem grandes mudanças na estrutura existente.
Confiabilidade	Assegurar que o sistema opere de forma consistente, minimizando falhas e tempo de inatividade para manter a operação contínua.
Manutenibilidade	Facilitar a atualização e a correção do sistema, permitindo que modificações e melhorias sejam feitas de forma rápida e eficiente.
Usabilidade	Assegurar que a interface de controle e os dispositivos sejam intuitivos, facilitando o uso pelos operadores.

Fonte: *Autoria própria. Todos os direitos reservados.*

Restrições

As restrições arquiteturais delimitam as opções de design e garantem que o sistema seja implementado dentro dos parâmetros técnicos e operacionais existentes. A **Tabela 3** lista essas restrições, evidenciando os desafios a serem enfrentados no desenvolvimento.

Tabela 3: *Restrições arquiteturais para a implementação do sistema.*

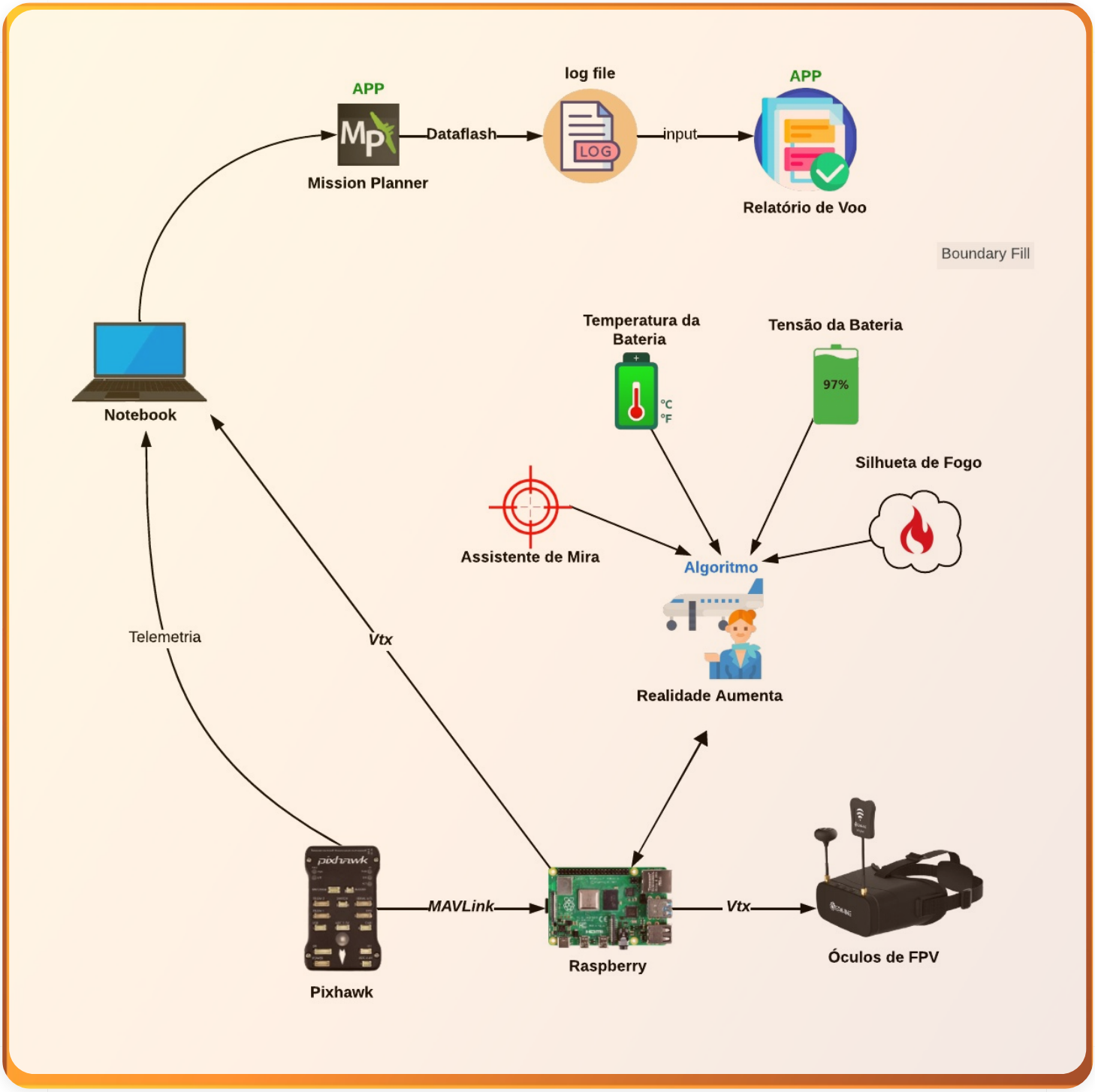
Restrição	Descrição
Limitações de Hardware	O sistema precisa ser compatível com o hardware disponível (Raspberry e Pixhawk), considerando memória e poder de processamento.
Compatibilidade de Software	O sistema deve funcionar com as versões de software que sejam compatíveis com os componentes de hardware.
Facilidade de Manutenção	O sistema deve ser fácil de manter e atualizar, com suporte para resolver problemas e implementar melhorias.
Integração Simples	O sistema precisa ser fácil de integrar com outros sistemas e tecnologias existentes ou futuras.

Fonte: *Autoria própria. Todos os direitos reservados.*

Diagrama de Arquitetura de Software

O **Diagrama de Arquitetura de Software** foi criado para representar o sistema de forma clara e organizada, destacando a interação entre seus principais componentes e as relações entre eles. Essa abordagem facilita a identificação de áreas problemáticas e promove uma colaboração eficiente entre as equipes, além de garantir uma visão comum dos objetivos do projeto. A **Figura 1** abaixo ilustra essa arquitetura.

Figura 1: *Diagrama de Arquitetura de Software.*



Fonte: *Autoria própria. Todos os direitos reservados.*

Tecnologias

O sistema foi desenvolvido com o suporte de várias tecnologias que desempenham papéis fundamentais em diferentes aspectos de sua arquitetura. Cada tecnologia foi escolhida estrategicamente para atender às necessidades específicas do projeto, garantindo eficiência, escalabilidade e facilidade de uso. A **Tabela 4** apresenta as principais tecnologias utilizadas, suas descrições e seus respectivos usos no sistema.

Tabela 4: *Tecnologias utilizadas no desenvolvimento do sistema e seus respectivos usos.*

Tecnologia	Descrição	Uso no Sistema
Linguagem de Programação	Python ou C++	Usada para o desenvolvimento do sistema, incluindo scripts de controle, algoritmos e interfaces de usuário.
Vtx	Protocolo de comunicação especializado na transmissão de vídeo	Envia o feed de vídeo em tempo real do drone para o piloto, garantindo uma visualização de alta qualidade e baixa latência

	transmissão de vídeo.	visualização de alta qualidade e baixa latência.
MAVLink	Protocolo de comunicação padronizado para veículos aéreos não tripulados.	Envia e recebe dados de telemetria entre o drone e o computador de controle, possibilitando o monitoramento em tempo real das operações de voo.
OpenCV	Framework de código aberto para aplicações de visão computacional.	Utilizado no assistente de mira para reconhecimento e processamento de imagens, ajudando no rastreamento e detecção de alvos.
Tkinter	Biblioteca nativa do Python para o desenvolvimento de interfaces gráficas.	Usada para criar interfaces interativas, visualizando dados em tempo real e gerando relatórios detalhados do voo.
Figma	Ferramenta de design de interfaces e protótipos.	Usada para criar protótipos e interfaces de usuário, facilitando a visualização e o planejamento da interação com o sistema.
Astah	Ferramenta de modelagem UML (Unified Modeling Language) para criação de diagramas de software.	Utilizada para criar os diagramas de sequência, auxiliando na documentação e no planejamento estrutural do sistema.
Lucidchart	Ferramenta online de criação de diagramas e colaboração visual.	Usada para diagramas de componentes, facilitando a compreensão e a comunicação entre equipes.

Fonte: *Autoria própria. Todos os direitos reservados.*

Visão de Caso de Uso

A **Visão de Caso de Uso** descreve como o sistema atende às interações entre os atores e suas funcionalidades. Esta seção detalha os elementos essenciais, como objetivos, fluxos principais e alternativos, além de requisitos específicos. A seguir, as tabelas apresentam os principais elementos e dois casos de uso exemplares.

Legenda

A **Tabela 5** apresenta a legenda utilizada para descrever os elementos que compõem os casos de uso detalhados nesta seção.

Tabela 5: *Legenda dos elementos de casos de uso.*

Ícone	Elemento	Descrição
	Ator	O usuário ou piloto que interage com o sistema.
	Objetivo	A meta ou finalidade do caso de uso, o que se busca alcançar.
✔	**Pré-condições**	Condições que precisam ser atendidas antes da execução do caso de uso.
	Fluxo Principal	Passos sequenciais que descrevem a execução normal do caso de uso.
⚠	**Fluxo Alternativo**	Cenários alternativos ou de erro que podem ocorrer durante a execução.
	Pós-condições	O resultado esperado ou o estado final após a execução do caso de uso.
	Requisitos Especiais	Necessidades específicas para garantir que o caso de uso seja executado de maneira eficaz e satisfatória.

Fonte: *Autoria própria. Todos os direitos reservados.*

Caso de Uso 1: Visualizar Informações nos Óculos FPV

Este caso de uso descreve como o sistema permite ao piloto acessar informações críticas em tempo real por meio dos óculos FPV, otimizando o desempenho durante a operação. A **Tabela 6** apresenta os elementos detalhados deste caso de uso.

Tabela 6: *Elementos do caso de uso: Visualizar Informações nos Óculos FPV.*

Ícone	Elemento	Descrição
	Ator	Piloto

	Objetivo	Permitir que o piloto visualize, em tempo real, informações fornecidas pelos óculos FPV (ex.: dados do drone, localização, altitude).
✅	**Pré-condições**	- Óculos FPV conectados ao sistema. - Sistema operacional funcional.
	Fluxo Principal	1. O sistema transmite informações aos óculos FPV. 2. O piloto visualiza as informações em tempo real.
⚠	**Fluxo Alternativo**	**Falha na Conexão**: O piloto recebe uma mensagem de erro caso haja falha na comunicação.
	Pós-condições	Piloto visualiza as informações enquanto o sistema estiver ativo.
	Requisitos Especiais	Informações claras e legíveis, com baixa latência.

Fonte: *Autoria própria. Todos os direitos reservados.*

Caso de Uso 2: Visualizar Informações nos Notebook

Este caso de uso descreve como o sistema permite o copiloto acessar informações críticas em tempo real por meio do computador, auxiliando o desempenho durante a operação. A **Tabela 6** apresenta mais detalhes desse caso de uso.

Tabela 7: *Elementos do caso de uso: Visualizar Informações nos Óculos FPV.*

Ícone	Elemento	Descrição
	Ator	Copiloto
	Objetivo	Permitir que o piloto visualize, em tempo real, informações fornecidas pelas câmeras do drone
✅	**Pré-condições**	- Computador conectado ao sistema. - Sistema operacional funcional.
	Fluxo Principal	1. O sistema transmite informações ao computador. 2. O copiloto visualiza as informações em tempo real.
⚠	**Fluxo Alternativo**	**Falha na Conexão**: O copiloto recebe uma mensagem de erro caso haja falha na comunicação.
	Pós-condições	Copiloto visualiza as informações enquanto o sistema estiver ativo.
	Requisitos Especiais	Informações claras e legíveis, com baixa latência.

Fonte: *Autoria própria. Todos os direitos reservados.*

Caso de Uso 3: Visualizar Relatório de Voo

Este caso de uso detalha como o sistema permite ao piloto gerar e analisar relatórios de voo a partir de arquivos de log, fornecendo insights sobre o desempenho da missão. A **Tabela 7** apresenta os elementos deste caso de uso.

Tabela 8: *Elementos do caso de uso: Visualizar Relatório de Voo.*

Ícone	Elemento	Descrição
	Ator	Piloto / Usuário
	Objetivo	Permitir que o piloto visualize os relatórios de voo, apresentando gráficos alimentados por dados de um arquivo de log gerado pelo MissionPlanner.
✅	**Pré-condições**	- Aplicativo de Relatório de Voo instalado. - Arquivo de log do MissionPlanner disponível para upload.
	Fluxo Principal	1. O piloto baixa o arquivo de log após o voo. 2. O piloto faz o upload do arquivo no aplicativo de Relatório de Voo.

		2. O piloto faz o upload do arquivo no aplicativo de Relatório de Voo. 3. O aplicativo gera gráficos com as variáveis de voo.
⚠	**Fluxo Alternativo**	**Erro no Arquivo de Log**: Caso o arquivo esteja corrompido ou com dados inconsistentes, o aplicativo exibe uma mensagem de erro.
	Pós-condições	O piloto visualiza os gráficos detalhados, com as variáveis de voo ao longo do tempo, para análise do desempenho da missão.
	Requisitos Especiais	O aplicativo deve ser capaz de processar arquivos de log de diferentes durações de voo e exibir gráficos de maneira clara e interativa.

Fonte: *Autoria própria. Todos os direitos reservados.*

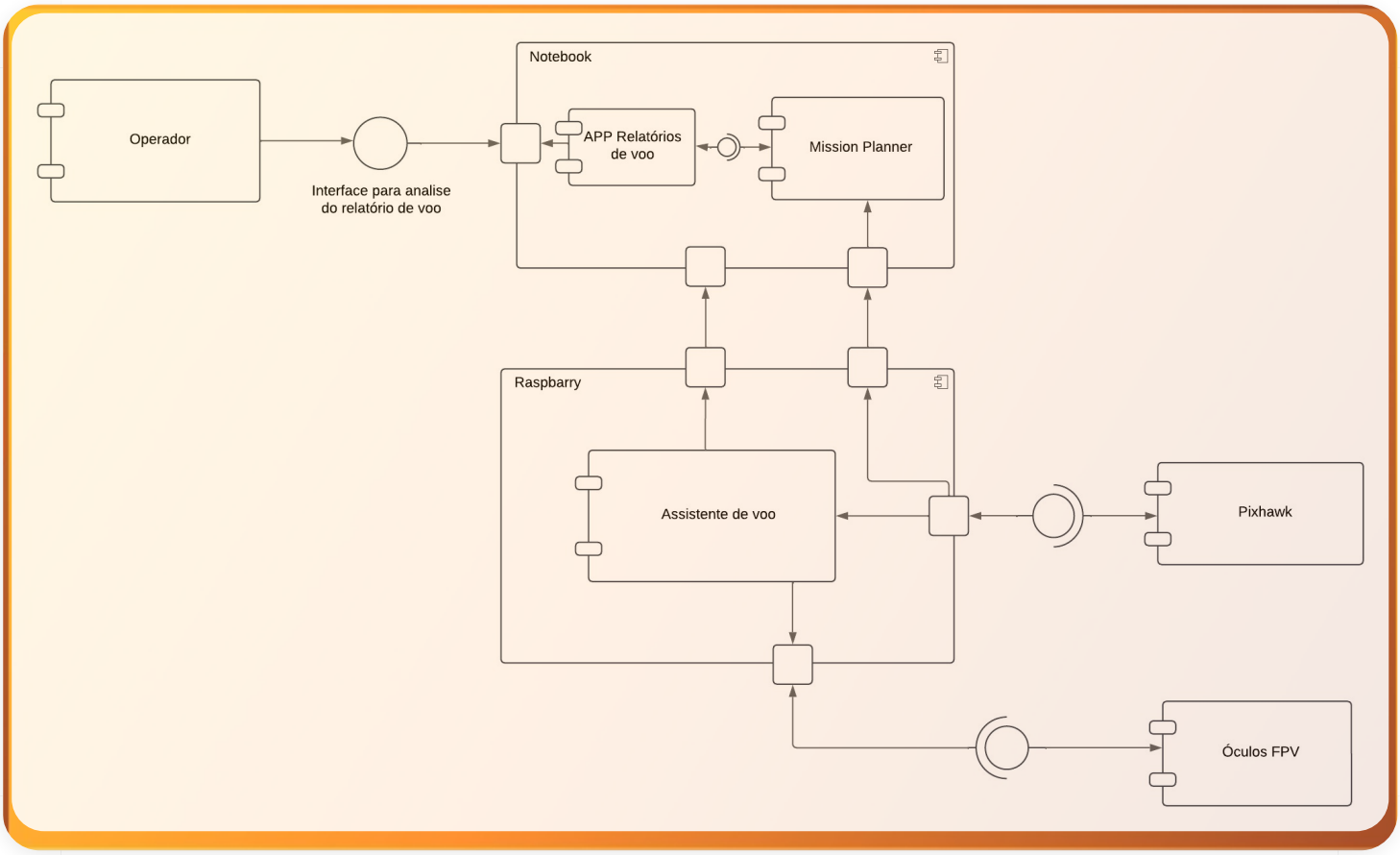
Visão Lógica

A **Visão Lógica** do sistema apresenta os componentes e as interações que compõem o fluxo de dados e as funcionalidades essenciais, abstraindo os detalhes físicos ou técnicos. Esta seção visa esclarecer como o sistema opera em nível conceitual, organizando as responsabilidades de cada módulo e as interações entre eles.

Diagrama de Componentes

O **Diagrama de Componentes** (Figura 2) ilustra os elementos principais do sistema, como o Mission Planner, Pixhawk e Raspberry Pi, e a forma como eles se conectam para entregar as funcionalidades esperadas, incluindo comunicação e monitoramento em tempo real.

Figura 2: *Diagrama de Componentes.*



Fonte: *Autoria própria. Todos os direitos reservados.*

Diagrama de Sequência

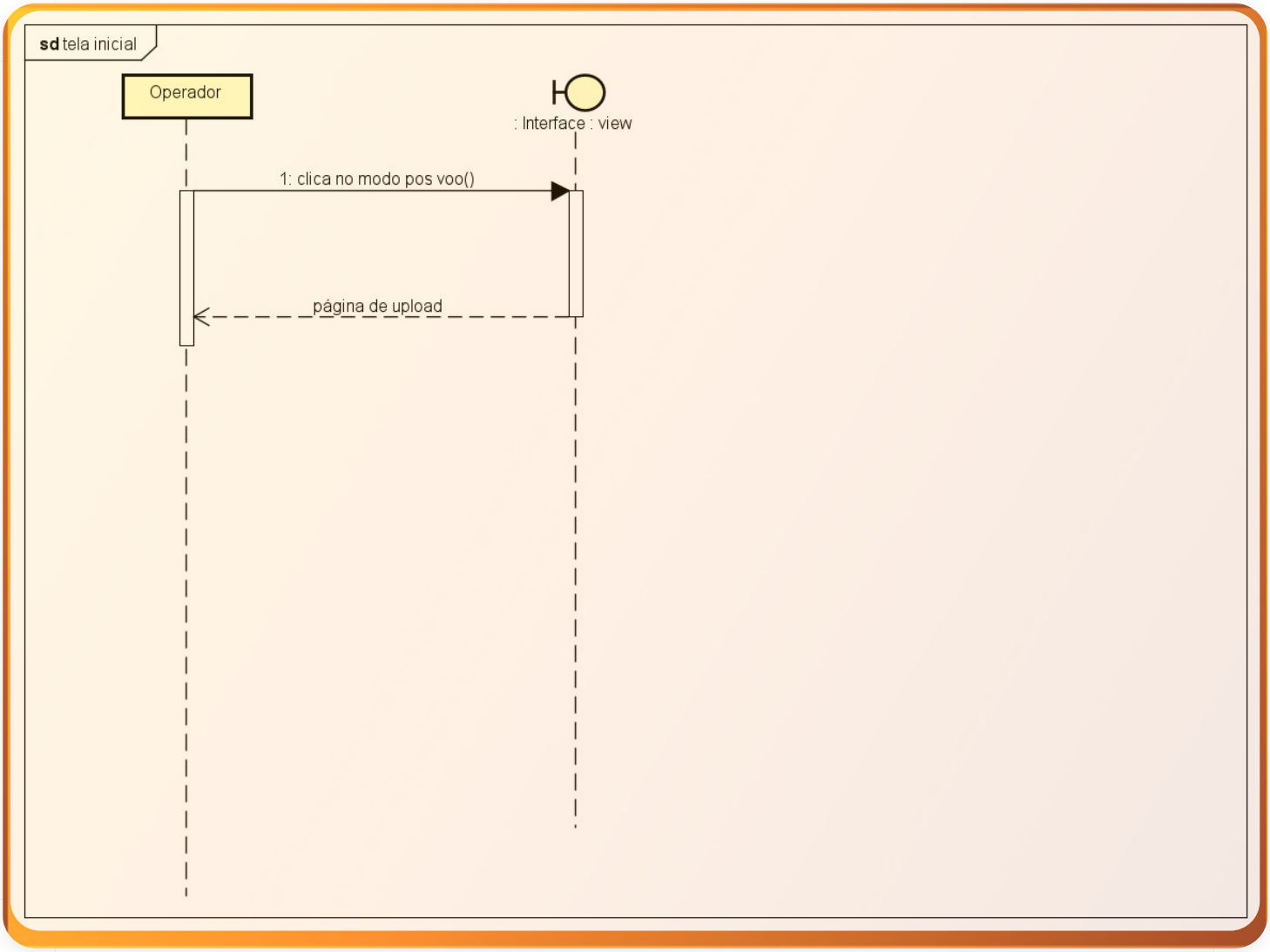
Os diagramas de sequência mostram como os componentes do sistema interagem em fluxos específicos, detalhando os passos e mensagens trocadas. No projeto, composto pelos sistemas **Relatório de Voo** e **Assistência de Voo**, os tópicos **Fluxos: Relatório de Voo** e **Assistência de Mira** apresentam os diagramas de sequência de cada sistema.

Fluxos de Relatório de Voo

FLUXO 1: PÁGINA INICIAL

O **Diagrama de Sequência da Página Inicial** (Figura 3) mostra o fluxo de informações desde o acesso inicial do piloto até a exibição da página inicial do sistema.

Figura 3: *Fluxo da página inicial.*

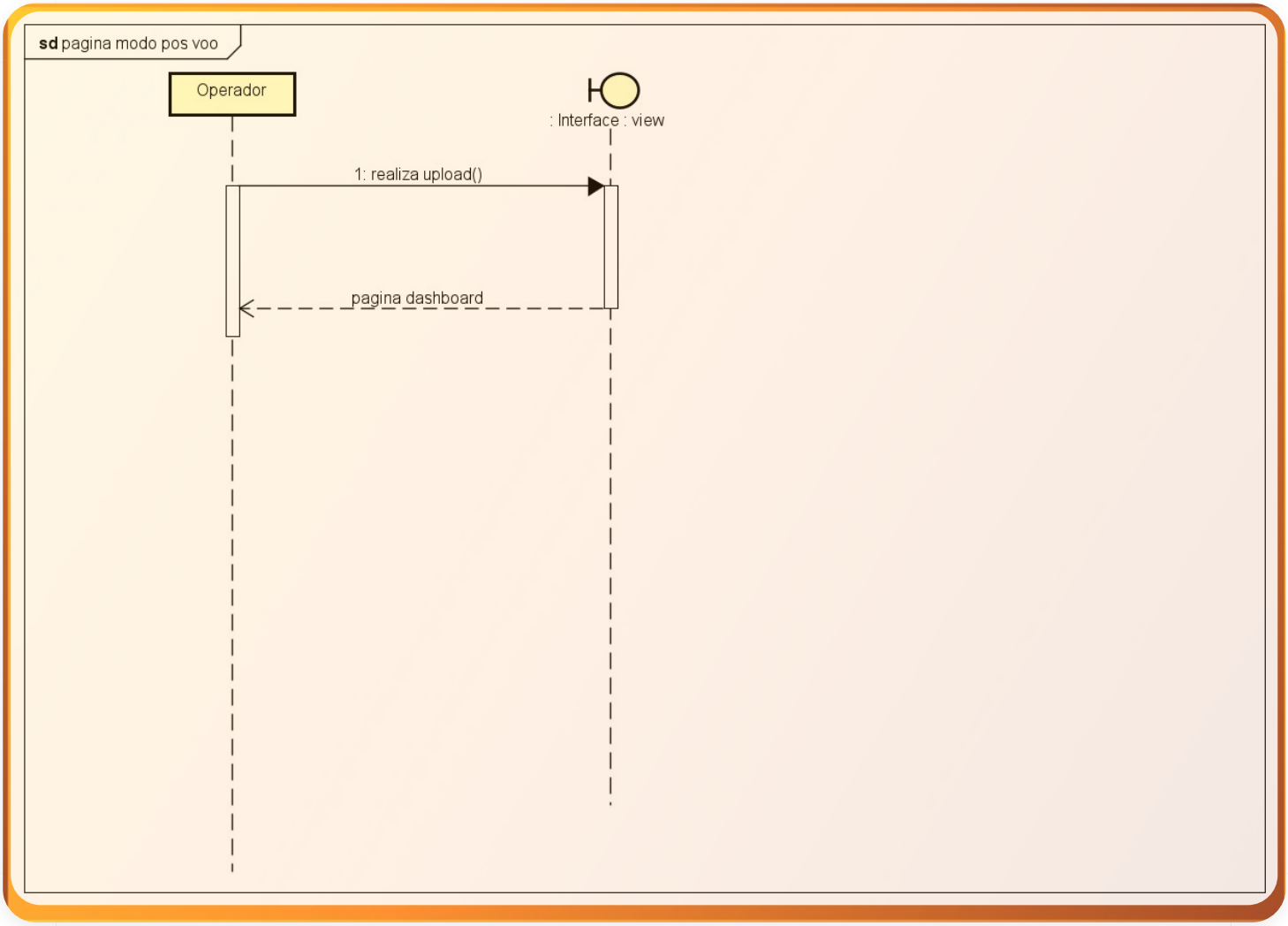


Fonte: *Autoria própria. Todos os direitos reservados.*

FLUXO 2: PÁGINA DE PÓS-VOO

A **Página de Pós-Voo** (Figura 4) descreve a interação necessária para exibir relatórios e análises baseados nos dados do voo.

Figura 4: *Fluxo da página de pós-voo.*

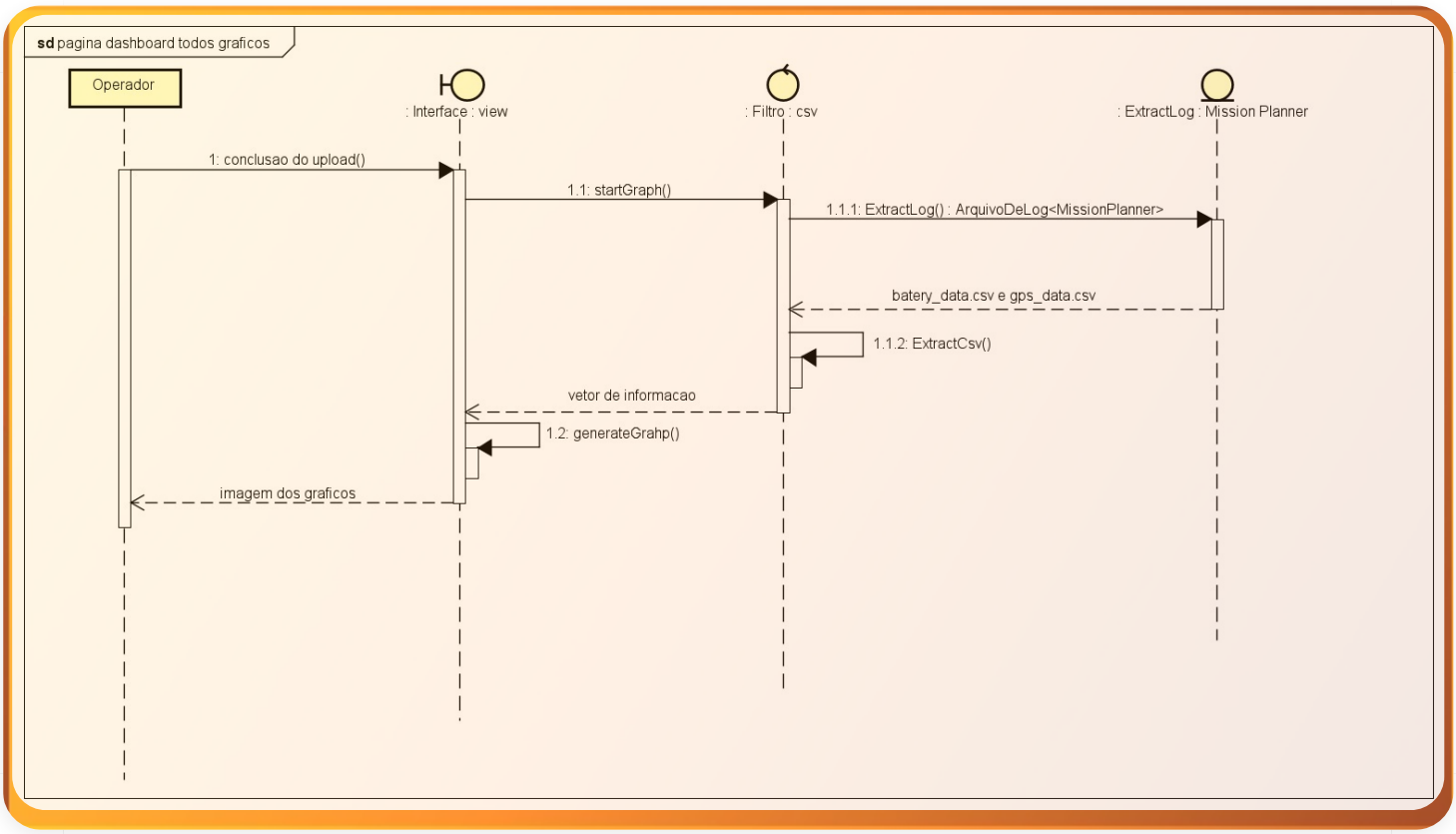


Fonte: *Autoria própria. Todos os direitos reservados.*

FLUXO 3: DASHBOARD DE TODOS OS GRÁFICOS

O **Dashboard de Todos os Gráficos** (Figura 5) apresenta como os gráficos de desempenho são carregados e exibidos para o piloto e/ou copiloto.

Figura 5: *Fluxo do dashboard com todos os gráficos.*

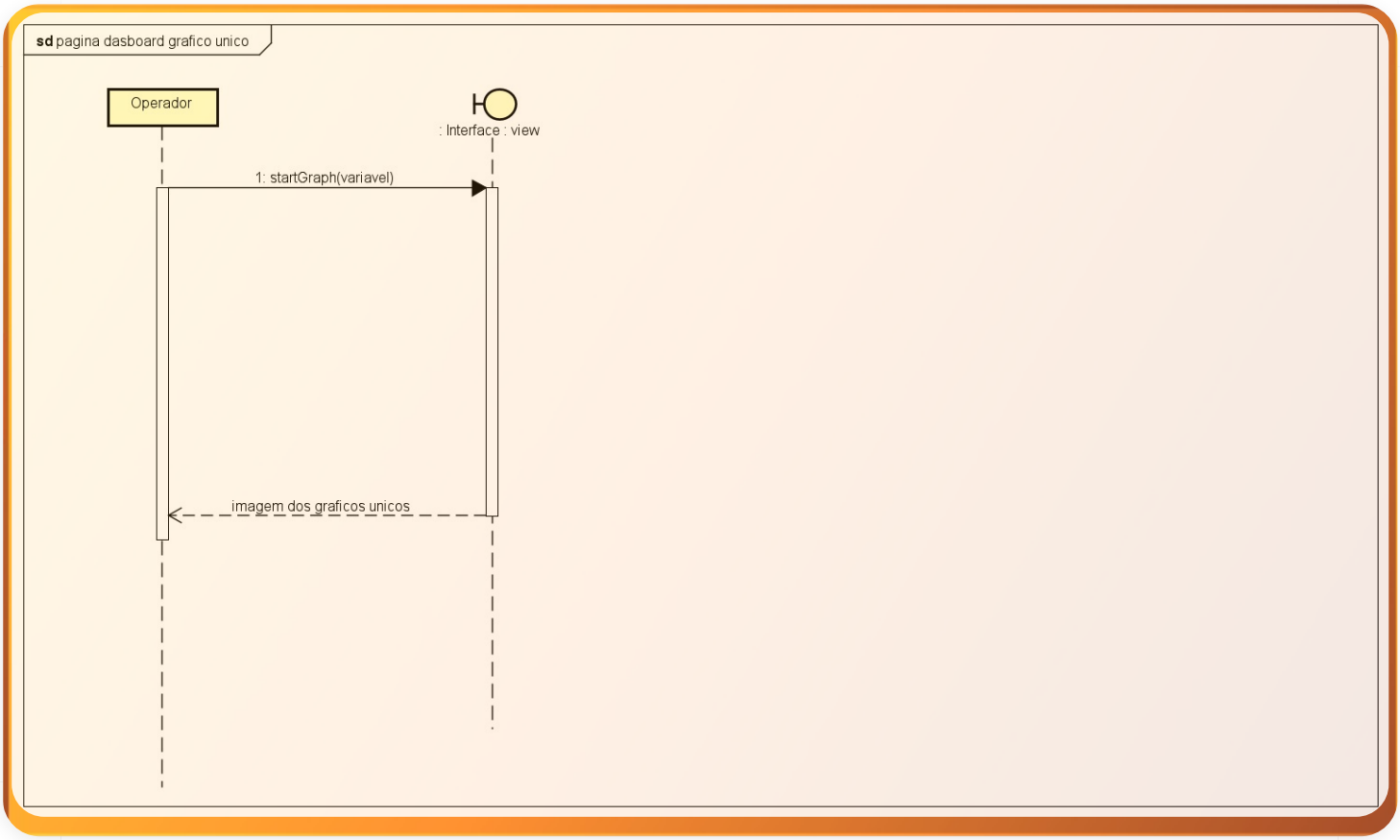


Fonte: Autoria própria. Todos os direitos reservados.

FLUXO 4: PÁGINA DE GRÁFICOS ÚNICOS

A **Página de Gráficos Únicos** (Figura 6) detalha o fluxo necessário para exibir gráficos individuais baseados em métricas específicas.

Figura 6: *Fluxo do dashboard com um gráfico.*



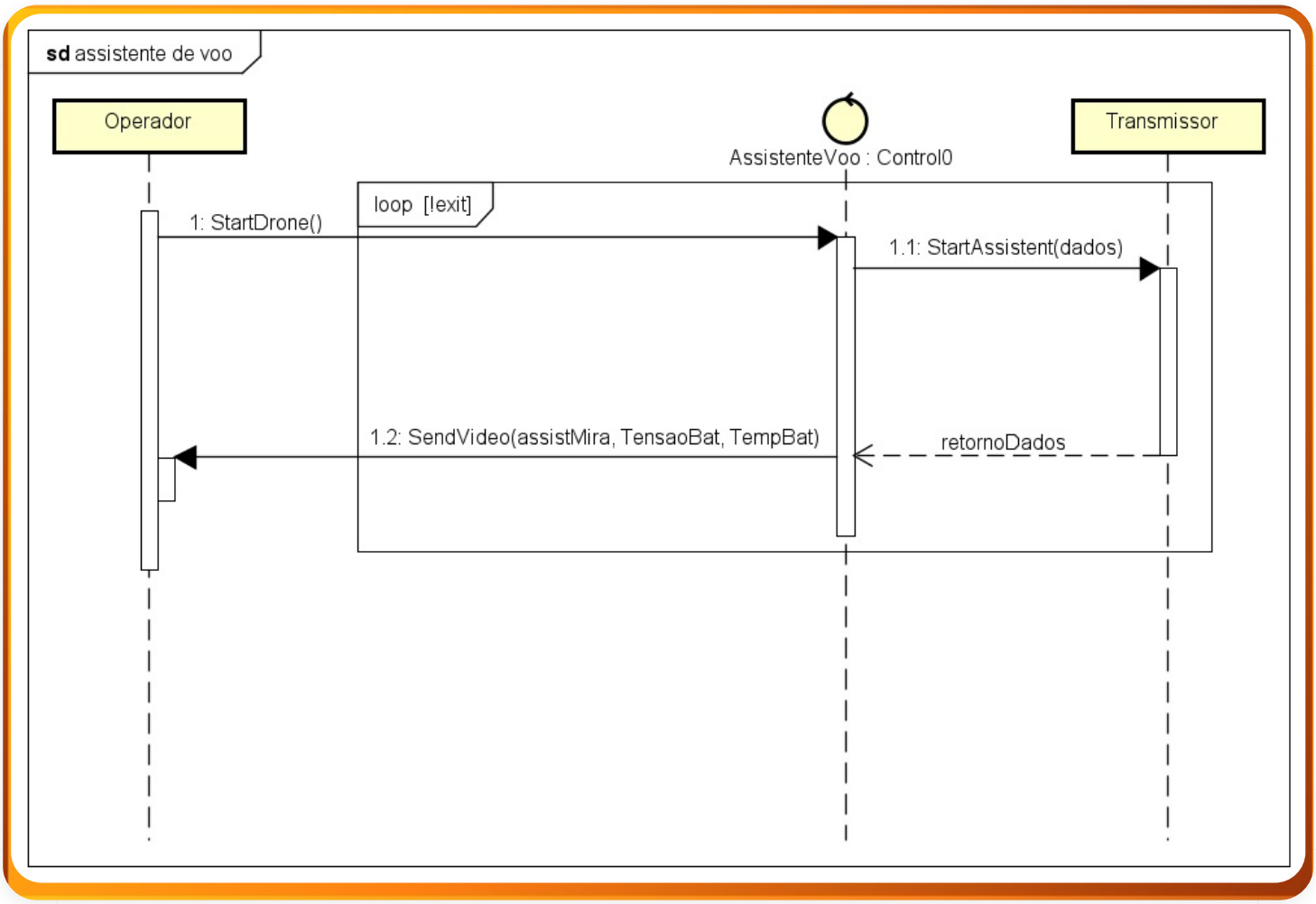
Fonte: *Autoria própria. Todos os direitos reservados.*

Fluxos de Assistência de Mira

FLUXO 1: START ASSITENTE DE VOO

O **Diagrama de Sequência da Realidade Aumentada** (Figura 7) mostra o fluxo de informações desde o acesso inicial do piloto até a exibição do vídeo no **FPV** ou **notebook**.

Figura 7: *Fluxo Assistente de Voo.*



Fonte: *Autoria própria. Todos os direitos reservados.*

Referências

1. **ELECTRON.** *What is Electron?*
   Acesso em: 03 nov. 2024.
2. **ARDUINO.** *What is Arduino?*
   Acesso em: 03 nov. 2024.
3. **EDRAWSOFT.** *Diagrama de Arquitetura de Sistema: Um Tutorial Completo*
   Acesso em: 03 nov. 2024.
4. **CIN.** *Conceito: Visão Lógica*
   Acesso em: 13 nov. 2024.
5. **LUCIDCHART.** *Diagrama de Componentes UML: o que é, como fazer e exemplos*
   Acesso em: 13 nov. 2024.
6. **LUCIDCHART.** *O que é um Diagrama de Sequência UML?*
   Acesso em: 13 nov. 2024.
7. **VISUAL PARADIGM.** *Tudo o que você precisa saber sobre diagramas de sequência.*
   Acesso em: 22 nov. 2024.

Tabela de Versionamento

Versão	Data	Descrição	Autor(es)

1.0	03/11/2024	Criação inicial e estrutura do artefato	Gustavo Martins e Gabriel Ferreira
1.1	13/11/2024	Desenvolvimento de uma nova versão do diagrama de arquitetura de software e atualização da imagem	Gustavo Martins, Felipe Freire, e Gabriel Ferreira
1.2	14/11/2024	Desenvolvimento dos tópicos **Componentes**, **Tecnologias**, **Caso de Uso 1**	Deivid Carvalho
1.3	14/11/2024	Correção de erros e aprimoramento na organização e visualização do artefato, com inclusão de legendas, desenvolvimento dos tópicos **Metas e Restrições Arquiteturais**, **Caso de Uso 2** e **Visão Lógica**, e uso de tabelas e ícones para maior clareza.	Gustavo Martins
1.4	14/11/2024	Desenvolvimento dos **Diagramas de Componentes e de Sequência**	Felipe Freire e Gustavo Martins
1.5	21/11/2024	Revisão e correção geral e evolução do artefato	Gustavo Martins Ribeiro
1.6	21/11/2024	Inclusão de Ferramentas no Tópico **Tecnologias**	Felipe Freire
1.7	22/11/2024	Inclusão do Diagrama de Sequência da Realidade Aumentada	Felipe Freire, Gabriel Ferreira, Gustavo Martins