



---

## Documento de Arquitetura de Software

201900264 Ariel Marte Araújo Silva

201802774 Kaio Ribeiro Sanchez

201800227 Gustavo Henrique De Freitas Martins

201905542 Marco Feitosa Araújo

# 1. Introdução

## 1.1 Finalidade

Este documento possui como objetivo definir os aspectos de Arquitetura do Software SiTAP e é direcionado aos stakeholders do software a ser avaliado.

## 1.2 Escopo

Este documento se baseia no documento de requisitos do STAP para definir os atributos de qualidade a serem priorizados, bem como, os estilos arquiteturais que favorecem tais atributos e as representações das visões arquiteturais e seus sub-produtos.

## 1.3 – Definições, Acrônimos e Abreviações

**STAP:** Sistema de Transferência Acompanhada de Pacientes

**SOA:** Arquitetura Orientada à Serviço

**REST:** Representational State Transfer, em português Transferência Representacional de Estado.

**RESTful:** são as API's que estão em conformidade com os critérios estabelecidos pelo padrão REST.

**Stakeholders:** Indivíduo, grupo ou organização que possua interesse no Sistema.

**API:** Interface de Programação de Aplicação. São mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos.

## 1.4 – Referências

Para elaboração deste documento foi utilizado como referência os seguintes documentos:

ID	Nome do Artefato
AAS_1	Especificação do Trabalho Final da Disciplina
AAS_2	Documento de Requisitos
AAS_3	ISO-IEEE 1010
AAS_4	4+1 View

## 1.5 – Visão Geral

Neste documento se pretende descrever quais serão os requisitos e restrições utilizados para definir a arquitetura a ser implementada para a aplicação proposta, assim como os atributos de qualidade que serão priorizados conforme determinação do documento de especificação.

## 2. Contexto da Arquitetura

A seção a seguir apresenta os requisitos e restrições utilizados para definir a arquitetura de software a ser implementada, definindo quais atributos de qualidade foram priorizados e a justificativa da escolha

As definições dos padrões arquiteturais utilizados foram pautadas conforme os atributos de qualidade selecionados, as quais apresentam *trade-off* favorecendo e desfavorecendo atributos de qualidade.

### 2.1 Funcionalidades e Restrições Arquiteturais

ID	Tipo	Descrição
RAS_1	Requisitos Não-Funcionais	Usabilidade: Adequação
RAS_2	Requisitos Não-Funcionais	Usabilidade: Facilidade de uso
RAS_3	Requisitos Não-Funcionais	Segurança: Confidencialidade
RAS_4	Requisitos Não-Funcionais	Segurança: Integridade
RAS_5	Requisitos Não-Funcionais	Segurança: Autenticidade

Os RAS apresentados acima, definidos pelo documento AAS\_1 orientarão as decisões sobre estilos e padrões arquiteturais a serem utilizados na implementação do STAP, com base em sua adequação para atender os atributos de qualidade e integração uns com os outros.

Interações em Realidade Aumentada são exigidos, visto que trata-se de uma aplicação com interações imersivas avançadas para auxiliar os profissionais de saúde envolvidos no atendimento e na transferência do cuidado de pacientes entre unidades hospitalares.

Também será apresentada uma versão simplificada ao paciente para que este possa consultar suas informações e tirar dúvidas a respeito dos procedimentos médicos pelos quais ele está sendo submetido.

Para atender aos requisitos levantados é necessário implantar uma arquitetura modularizada, com responsabilidades individualizadas e independentes e um protocolo de comunicação em rede.

Assim, é clara a necessidade de um estilo arquitetural de **Componentes** como forma de viabilizar essa separação de responsabilidades. Por ser um serviço em rede teremos que observar o estilo arquitetural **Cliente-Servidor** utilizando o padrão **REST** para definir os métodos de comunicação.

Adicionalmente, temos a necessidade do sistema de se manter atualizado com os dados atuais de um paciente em transferência. Desta forma somos direcionados a implementação de um estilo arquitetural de **Publish-Subscribe** no intuito de garantir uma divulgação mais instantânea, tirando proveito da comunicação 5G e computação em borda para disponibilizar dados em tempo real para os profissionais de saúde.

Todos esses dados e requisições serão processados por serviços individualizados e personalizados, que atenderão individualmente aos casos de uso e retornam os dados das consultas de informações e históricos médicos dos pacientes, por meio de IA utilizadas no processamento de dados, o que preconiza a utilização de um estilo arquitetural **Orientado a Serviços (SOA)**.

## **2.2 Atributos de Qualidade Prioritários**

Diante do definido no AAS\_1, o software a ser desenvolvido deve ser implementado buscando garantir os atributos da qualidade de **Usabilidade**

(adequação e facilidade de uso) e **Segurança** (confidencialidade, integridade e autenticidade).

Para atender os RAS\_3, RAS\_4 e RAS\_5 é que usaremos o estilo arquitetural de componentes, que devido a sua modularização, permite que diversos níveis de segurança sejam implantados. Também, para tornar a chamadas entre cliente e servidor será utilizado API's RESTful com autenticação OAuth e por meio de protocolo HTTPS e certificação de origem das informações que chegam ao broker.

Os RAS\_1 e RAS\_2 serão atendido pelas várias formas de interação com o sistema do lado do cliente, podendo ser utilizado juntamente com um dispositivo de Realidade Aumentada, em um programa desenvolvido utilizando a plataforma do Snapdragon Spaces ou por meio de um aplicativo mobile com IAs que ajudarão na utilização por meio de comandos por voz ou reconhecimento facial.

### 3. Representação da Arquitetura

Para representar as decisões arquiteturais definidas serão utilizadas os pontos de vista definidos no documento AAS\_4, que são:

Ponto de Vista	Visão	Diagrama(s)
Projetista	Desenvolvimento	Componentes
Desenvolvedor	Processo	Sequência
	Lógica	Classes
Implantador	Física	Implantação
-	Casos de Uso	Casos de Uso

Os tópicos a seguir detalham esses pontos de vista arquiteturais, assim como as visões e os correspondentes diagramas utilizados para sua representação.

## 4. PONTO DE VISTA DOS CASOS DE USO

O diagrama de caso de uso resume os detalhes dos usuários do seu sistema e as interações deles com o sistema. Sendo adequado para fornecer uma visão geral do relacionamento entre casos de uso, atores e sistemas.

### 4.1 Visão Geral

Os Casos de Uso para o STAP apresentam as funcionalidades do sistema de visualização de dados clínicos, conforme descrito no documento AAS\_1. Cada caso de uso é representado por uma elipse contendo seu nome do e um ator representado por um boneco palito.

Os casos de uso vinculados ao ator Profissional de Saúde são:

- Atualizar um *P.Avatar* com informações do paciente
- Identificar um paciente através do *P.Avatar*
- Atuar remotamente em um Paciente
- Simular uma intervenção em um paciente
- Identificar os pacientes pelo grau de urgência
- Receber assistência de uma IA
- Dar feedback das sugestões da IA

Os casos de uso vinculados ao ator Paciente são:

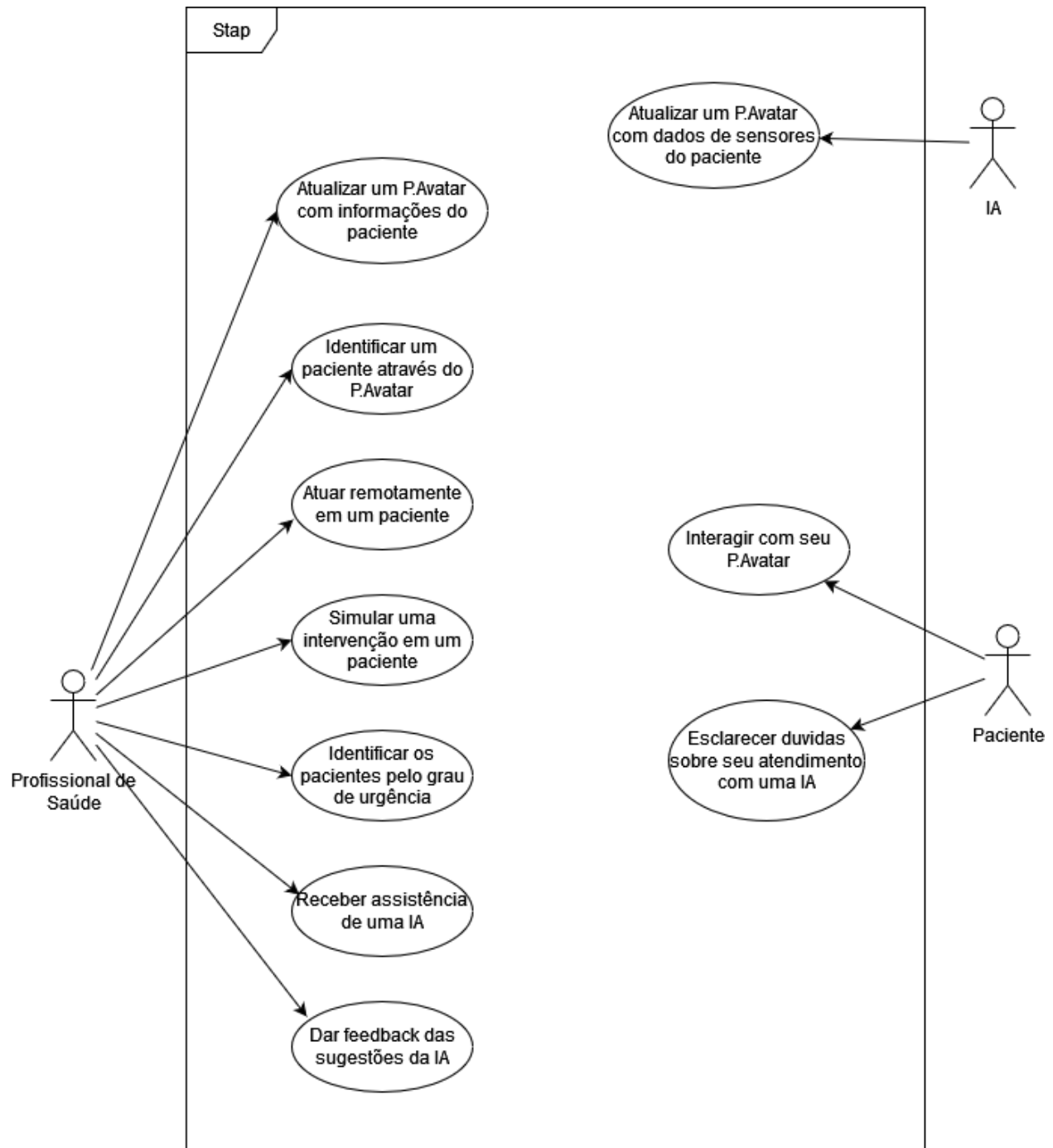
- Interagir com seu *P.Avatar*
- Esclarecer dúvidas sobre seu atendimento com uma IA

O caso de uso vinculado ao ator IA é:

- Atualizar um *P.Avatar* com dados de sensores do paciente



## 4. Visão de Casos de Uso



## **4.2.1 Requisitos Funcionais**

### **HU001 – Identificar paciente**

Como profissional da saúde

Quero identificar o paciente através de reconhecimento facial ou comando de voz através de um óculos AR/VR

Para ter acesso aos dados e prontuários do paciente de maneira rápida e segura.

### **HU002 – Atuação automática/remota sobre o paciente**

Como profissional de saúde

Quero através do P.avatar do paciente atuar de forma remota sobre o mesmo

Para administrar medicamentos ou realizar procedimentos de tratamentos via equipamentos de forma remota.

### **HU003 – Simular intervenção de tratamento no P.avatar do paciente**

Como profissional de saúde,

Quero simular uma intervenção de tratamento no P.avatar do paciente,

Para antecipar o melhor tratamento com base nas informações do *P.avatar*.

### **HU004 – Identificação do grau de urgência do atendimento**

Como profissional de saúde,

Quero identificar os pacientes por grau de urgência de atendimento seguindo o protocolo de Manchester

Para priorizar os casos que demandam maior intervenção

### **HU005 – Assistência por bot para o profissional**

Como profissional de saúde,

Quero receber o auxílio e sugestões de um bot(IA),

Para estimar o estado de saúde do paciente, receber sugestões de procedimentos de tratamento e medição com base numa base de dados de conhecimento médico confiável.

#### **HU006 – Oferecer feedback ao bot**

Como profissional de saúde,  
Quero oferecer feedback sobre os resultados e sugestões dados pelo bot de assistência,  
Para reforçar e aprimorar o aprendizado da máquina e melhorar sua acurácia de sugestões.

#### **HU007 – Assistência por bot para o paciente**

Como paciente,  
Quero receber assistência de um bot (IA) através de meu celular  
Para responder as minhas dúvidas quanto ao processo de atendimento, intervenções e medicamentos a qual estou sendo submetido.

#### **HU008 – Interação com o P.avatar pelo paciente**

Como paciente,  
Quero interagir por voz ou texto com o meu gêmeo digital com o auxílio de um bot  
Para aprender sobre a minha categoria de urgência de atendimento e consultar as minhas informações do prontuário.

### **4.2.2 Requisitos não-funcionais**

#### **RNF001 – Conectividade**

Descrição: Quando o profissional de saúde estiver atendendo o paciente, ele deve estar conectado a uma rede de alta velocidade e baixa latência para carregamento quase instantâneo dos dados do paciente.

#### **RNF002 – Gêmeo digital do paciente**

Descrição: O *P.avatar* do paciente deve consistir em um gêmeo digital do paciente, capaz de representar seus dados pessoais e dados do seu prontuário com as informações mais recentes de exames e sensores corporais em tempo real.

### **RNF003 – Atuação automática/remota sobre o paciente**

Descrição: O *P.avatar* de um paciente deve ser capaz de fornecer uma interface para controle automático dos atuadores ligados ao paciente para a dispensação de medicamentos e outras formas de tratamento de modo remoto.

### **RNF004 – Estimar o estado de saúde do paciente**

Descrição: O sistema poderá estimar o estado de saúde do paciente com base nos dados do *P.avatar* do mesmo correlacionando com a base de conhecimento disponível e poderá oferecer sugestões e feedbacks para aprender com erros e reforçar acertos.

## **5. PONTO DE VISTA DO PROJETISTA**

O ponto de vista do projetista é direcionado para a equipe responsável pelo projeto e desenvolvimento do software, visto que apresenta uma visão geral das partes que irão compor o sistema, suas responsabilidades, as dependências entre estes componentes, etc. Esta perspectiva foi escolhida por auxiliar na compreensão do software desde a fase de elaboração até a construção em si.

### **5.1 Visão Geral**

O modelo arquitetural proposto para o software STAP é composto por seis componentes principais: Um cliente consumidor, um cliente produtor de dados, uma *API* de Realidade Aumentada com redundância em nuvem, um Broker em nuvem responsável pela orquestração do atendimento às requisições de acesso e armazenamento de dados dos clientes, um banco de servidores em redundância provendo *web services* que atenderão a estas requisições de acordo com a disponibilidade e estado de carga do conjunto, e um componente de banco de dados também em redundância representando sistemas externos responsáveis por disponibilizar os dados necessários e vinculados ao banco de servidores.

### **5.2 Visão de Desenvolvimento**

O componente Cliente Consumidor é responsável pela interação do usuário com o sistema através de um navegador web, um dispositivo de realidade aumentada ou um *app* Android. Ele é responsável pela execução local do serviço de visualização do *P.Avatar*, por prover uma camada inicial de segurança e autenticação, e por requisitar os dados do *P.Avatar* gerado pela *API* de Realidade Aumentada com informações dos servidores.

O componente Cliente Produtor é responsável pela aquisição de dados de sensores ligados a um paciente em trânsito e pelo processamento dos mesmos, de forma certificável, para envio aos servidores através do Broker.

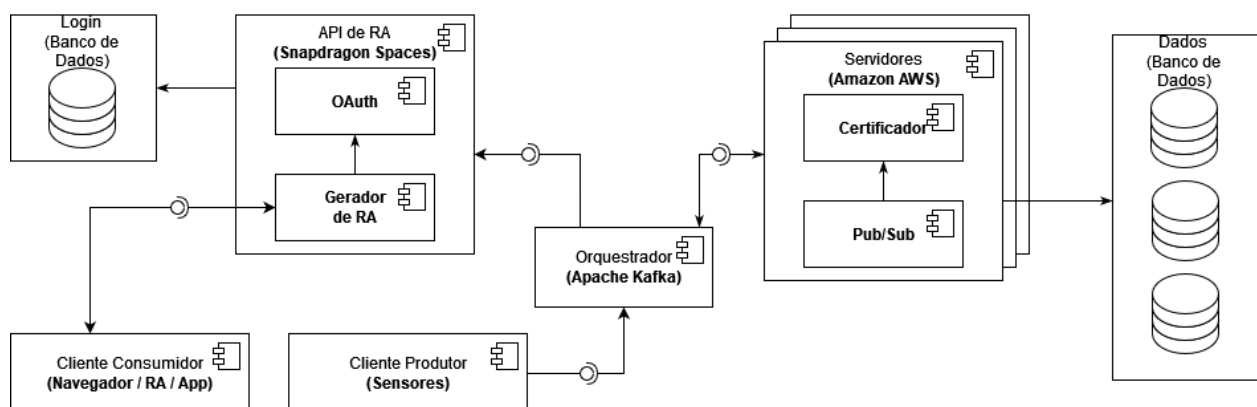
O componente API de Realidade Aumentada faz a autenticação dos usuários e requisições para o Broker de dados de um paciente, a partir dos quais gera o *P.Avatar* para interação de acordo com o tipo do usuário.

O componente Broker é responsável pela orquestração do direcionamento das requisições aos *web services* executados em servidores com maior disponibilidade operacional, de modo a permitir o acompanhamento em tempo real (ou o mais próximo possível) dos pacientes em trânsito de forma consistente, independente de sobrecarga ou falhas que porventura ocorram.

O componente Banco de Servidores é responsável por prover e executar os serviços que compõem a aplicação, tratando as requisições atribuídas pelo Broker a cada *web service*. Estes serviços fazem a certificação da origem dos dados recebidos do Cliente Produtor, a persistência destes dados no tópico correto relacionado a cada paciente, e a busca e disponibilização dos mesmo para a API de Realidade Aumentada.

O Componente Banco de Dados executa as instâncias de banco de dados que armazenam dados obtidos por sensores das unidades de saúde em tópicos específicos, que serão fornecidos à API de Realidade Aumentada para a formulação do *P.Avatar* de cada paciente no sistema.

A interação entre os componentes citados acima pode ser vista através do Diagrama UML de Componentes a seguir:



### 5.3 Detalhamento dos serviços

Como descrito no tópico anterior, a API de Realidade Aumentada faz a autenticação dos usuários e faz requisições via Broker ao banco de servidores, solicitando dados de pacientes com os quais gera o *P.Avatar*.

O serviço View provê uma interface para interação com o *P.Avatar*, a ser executada no cliente de acordo com a plataforma de acesso, através de requisições REST.

O serviço de Autenticação fornece segurança de acesso à aplicação através da validação de credenciais criptografadas e verificação de permissões de acesso, que delimitam os tipos de interação com o *P.Avatar* e a IA possíveis para o usuário.

O serviço de Geração de *P.Avatar* cria um avatar de metaverso a partir dos dados de um paciente, obtidos do banco de servidores via Broker, além das interações possíveis com ele (como simular procedimentos).

O componente Banco de Servidores fornece instâncias de web services que responderão às requisições dos clientes após orquestração via Broker.

O serviço Pub/Sub, executado em containers Docker em redundância, faz o tratamento dos dados recebidos via Broker dos Clientes Produtores, os organizando em tópicos por paciente e tipo de dado, e os persistindo em Banco de Dados. Além disso, faz a recuperação destes dados por tópico para transmissão para a API de Realidade Aumentada.

O serviço de Certificação verifica a autenticidade da origem dos dados recebidos via Broker dos Clientes Produtores, assegurando a confiabilidade da informação médica que será usada para a geração dos *P.Avatar*.

## **6. PONTO DE VISTA DO DESENVOLVEDOR**

O ponto de vista do desenvolvedor é direcionado aos projetistas e desenvolvedores do software, com o objetivo de determinar o processo operacional das funcionalidades levantadas nos requisitos e sua estrutura lógica. Nele se determina as partes principais de uma aplicação, responsáveis por definir as funcionalidades e restrições, como as classes e seus relacionamentos.

### **6.1 Visão Geral**

A modelagem do processo de aquisição e visualização dos dados de um paciente, o fluxo destes dados a cada passo desde sua obtenção em um sensor até as interações do usuário com o *P.Avatar*, permitem uma compreensão melhor de quais as entidades e ações deverão fazer parte do software sendo desenvolvido.

A partir desta percepção, é modelada a lógica do sistema, suas classes e métodos e como estes se relacionam um com o outro. Com isso, os desenvolvedores de software têm uma estrutura clara sobre a qual implementar a aplicação.

### **6.2 Visão de Processo**

Como dito acima, a percepção do processo pelo qual os dados são adquiridos pelos sensores na unidade de saúde, processados e persistidos, para posteriormente formar uma base para a criação de um *P.Avatar* com o qual os usuários poderão interagir é um passo importante na estruturação da aplicação sendo desenvolvida. Devido à relativa complexidade da natureza do fluxo destes dados, é imprescindível que haja clareza nesta compreensão por parte de todos os envolvidos no desenvolvimento do software.

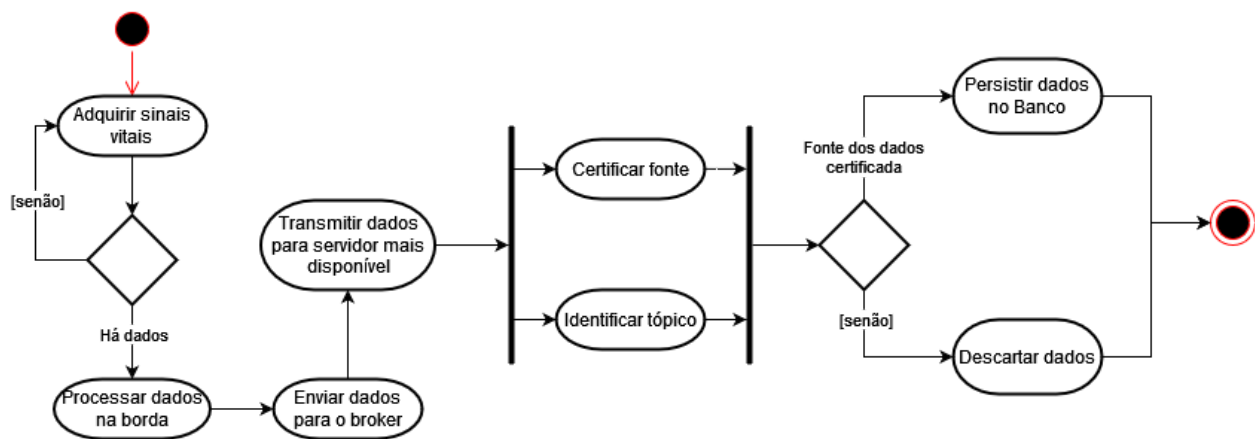


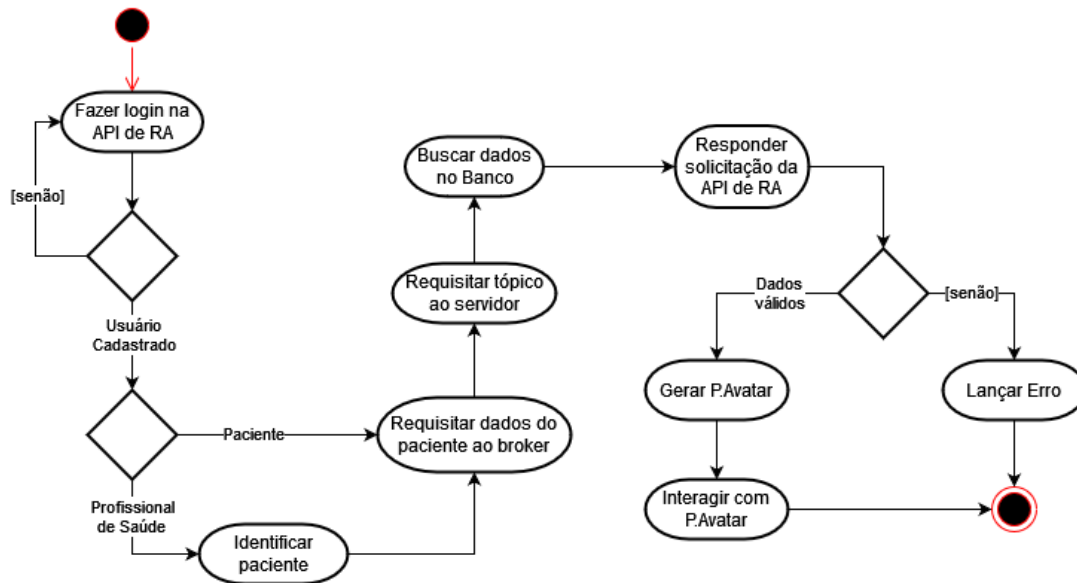
Os dados são obtidos através de sensores bioelétricos e de localização, e computados na borda da rede em que se encontram, para então serem enviados ao broker que os repassará a um servidor mais disponível no banco para persistência em banco de dados.

Quando um usuário tentar acessar estes dados, ele fará login em uma API de Realidade Aumentada, que fará a autenticação de suas credenciais e requisitará do broker os dados do paciente específico, a partir dos quais criará um P.Avatar com o qual este usuário poderá interagir, de acordo com as permissões de seu papel.

Um paciente poderá visualizar os diagnósticos registrados e tirar dúvidas sobre eles com uma Inteligência Artificial vinculada à aplicação. Um profissional de saúde poderá visualizar os dados médicos do paciente, consultar e definir seu grau de urgência, diagnosticar sua condição com o auxílio da I.A., simular intervenções através do gêmeo virtual e atuar remotamente no paciente, se ou quando necessário.

Os processos que organizam estas ações podem ser vistos nos Diagrama UML de Atividades a seguir:





## 6.3 Visão Lógica

A partir do diagrama de atividades, os projetistas do software visualizam mais claramente as entidades que compõem o sistema, suas ações permitidas e necessárias, e a partir disso definem as classes que formarão a aplicação, seus relacionamentos umas com as outras e métodos públicos e privados necessários para que a aplicação cumpra a função requisitada.

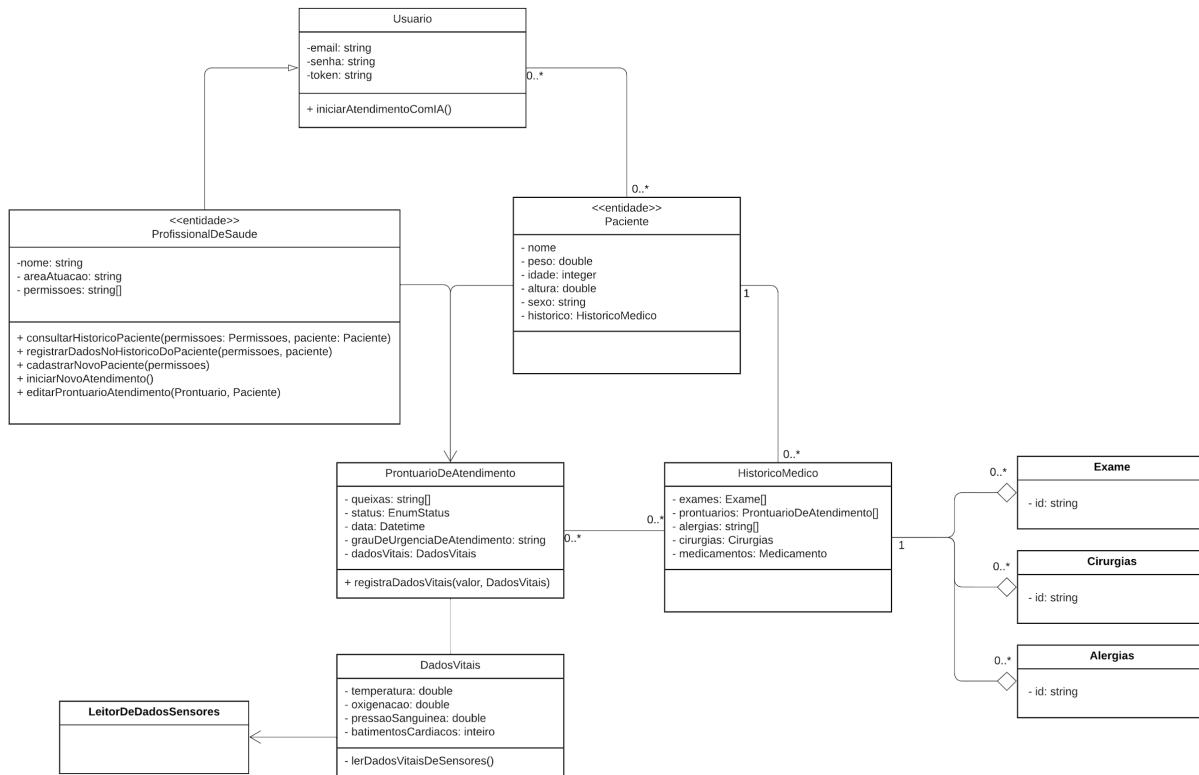
### 6.3.1 Detalhamento das classes

Há dois tipos de Usuario, Paciente e ProfissionalDeSaude. Ambos têm dados de autenticação (email, senha e token), e um Paciente tem também dados pessoais (nome, altura, peso, idade, sexo, e um históricoMedico) enquanto um ProfissionalDeSaude tem dados relevantes à sua atuação (nome, permissões e areaAtuacao).

Eles consultam um ProntuarioDeAtendimento, composto por queixas, data, grauDeUrgenciaDeAtendimento, status e dadosVitalis (batimentosCardiacos, pressaoSanguinea, temperatura e oxigenacao, obtidos através de um ou mais LeitorDeDadosSensores).

Estes prontuários formam o HistoricoMedico (vinculado a um único Paciente), junto com dados de exames, cirurgias, alergias e medicamentos.

Estas entidades e seus relacionamentos podem ser vistos no Diagrama UML de Classes a seguir:



## **7. PONTO DE VISTA DO IMPLANTADOR**

O ponto de vista do implantador tem como alvo a equipe de implantação, sendo responsável por definir as ferramentas e ambiente computacional pretendidos para que o software funcione de forma otimizada. Este ponto de vista foi escolhido devido à complexidade da aplicação, com diversos componentes independentes e que demandam interoperabilidade, sendo executados em containers virtualizados em nuvem, processados na borda da rede, e/ou em redundância.

### **7.1 Visão Geral**

Como exposto anteriormente, a arquitetura apresenta seis componentes principais: um cliente consumidor, um cliente produtor, uma API de realidade aumentada, um broker orquestrador, um banco de servidores em nuvem executando web services independentes em containers virtualizados, e bancos de dados NoSQL contendo os dados dos pacientes, organizados em tópicos.

### **7.2 Visão Física**

Os componentes citados são nós-físicos do software, com configurações específicas e utilizando ferramentas escolhidas para seu funcionamento ideal, como hardware, sistema operacional e software externos.

Os dados dos sensores de monitoramento do paciente são processados na borda da rede e enviados via requisições REST para o Broker.

A API de Realidade Aumentada é executada em servidor na nuvem, assim como seu banco de dados associado contendo os dados de autenticação dos usuários cadastrados

Os web services do Pub/Sub e de certificação de origem são executados em máquinas virtuais rodando paralelamente em contêineres nos servidores em



processamento na borda da rede e posterior transmissão via requisições REST para o broker.

O nó-físico da API de Realidade Aumentada é composto por containers Docker virtualizados executados em redundância e na infraestrutura de nuvem AWS, com o software desenvolvido com Snapdragon Spaces rodando em uma plataforma Ubuntu..

O broker funciona em um servidor específico na nuvem, com sistema operacional Ubuntu, e para sua execução é necessária a instalação do software Apache Kafka.

O nó-físico do servidor é, assim como a API de Realidade aumentada, um conjunto de containers Docker virtualizados, executados em nuvem de forma redundante. A aplicação funcionará sobre uma plataforma Ubuntu.

Os nós-físicos de Login e Dados são abstrações das aplicações externas ao software, responsáveis pela disponibilização de bancos de dados NoSQL MongoDB.

A comunicação entre os nós-físicos cliente produtor e consumidor, o nó-físico API de Realidade Aumentada e o nó-físico orquestrador se dá por requisições REST via HTTP. O orquestrador se comunica com o nó-físico do servidor através do framework Kafka Connect. Os nós-físicos de API RA e servidor se comunicam com os nós-físicos de bancos de dados através do serviço em nuvem Atlas.