

Consulta Interativa

Documento de Arquitetura de Software

1. Introdução

1.1. Finalidade

A arquitetura de software proposta tem como principal objetivo mitigar os problemas de comunicação e registro de documentos clínicos entre diferentes unidades hospitalares de forma segura e consistente, mas também com interfaces intuitivas e eficientes. Para tanto, o projeto visa construir uma solução baseada em Metaverso, Inteligência Artificial (IA) e monitoramento de sinais vitais a distância para agilizar e qualificar o atendimento e a transferência do cuidado.

1.2. Escopo

A arquitetura inclui um banco de dados centralizado para armazenar os prontuários eletrônicos dos pacientes e os avatares P-Avatar associados, bem como um servidor que permite aos profissionais de saúde acessar o sistema através de óculos de realidade aumentada (AR) e virtual (VR). O sistema também inclui dois tipos de bots: um para os profissionais de saúde (assistente conselheiro) e outro para os pacientes (atendimento personalizado).

O escopo da arquitetura inclui a identificação ou cadastro de pacientes e a atualização ou criação de prontuários eletrônicos e P-Avatar, bem como a consulta e modificação de informações pelos profissionais de saúde e acesso a uma versão simplificada do P-Avatar pelos pacientes através de smartphones convencionais. A arquitetura também inclui a implementação de uma arquitetura de camadas e o padrão de projeto de software de mapeamento de objetos-relacionais (ORM) para facilitar o acesso e a modificação dos dados pelos profissionais de saúde.

2. Contexto da Arquitetura

2.1. Funcionalidades arquiteturais

Como dito anteriormente, o sistema inclui um **banco de dados**, um **servidor** para acesso ao **sistema** com um óculos de realidade aumentada e virtual e dois **bots**.

Para manter a funcionalidade do banco de dados, utilizaremos uma base armazenada em um ambiente em nuvem próprio, para que seja acessado de qualquer lugar, que deve ser mantido, e será um dos componentes da aplicação.

O servidor, que será acessado através de um **gateway**, também deve ser armazenado em um ou mais ambientes, que será acessado pelo profissional de saúde. Esse servidor terá acesso direto à base local e deve possuir equipamentos necessários para a interação com a realidade virtual/aumentada.

Os bots precisarão ser mantidos em um servidor em nuvem, mas que também possua conexão com a base local, para que ele possa ser um assistente confiável, além do atendimento personalizado ao paciente, que depende totalmente dos dados do P-avatar.

2.2. Atributos de qualidade prioritários

O software terá uma arquitetura modularizada, composta por camadas de microsserviços, para que as responsabilidades da aplicação sejam bem definidas. Com esse tipo de separação, fica mais fácil a interação entre o software e o ecossistema abordado.

Pelo fato do software interagir com uma base de dados que possui informações sensíveis sobre pacientes e funcionários, um dos pilares da aplicação deve ser a **Segurança**. Esse atributo pode ser favorecido pela arquitetura modularizada supracitada.

Além disso, por lidar com questões médicas e emergenciais, uma falha no sistema que não seja rapidamente consertada, pode prejudicar a vida de um paciente, colocando sua vida em risco ou até levando o mesmo a óbito. Dessa forma, a **Manutenibilidade** do código é, sem dúvida, outro atributo indispensável, para que os erros sejam corrigidos de forma rápida e eficiente.

Outra questão é que, por conta de ser um sistema que deve ser utilizado em vários ambientes, sistemas operacionais diferentes e equipamentos de realidade virtual/aumentada diferentes, outro atributo de qualidade é a **Portabilidade**, que deve ser facilitada pela utilização de microsserviços separados.

Por último, o sistema deve ter alta **Confiabilidade**. Os dados armazenados devem ser persistidos, deve haver tratamento de exceções e deve aguentar um grande fluxo de usuários.

3. Representação da Arquitetura

Conforme definido nos tópicos anteriores, a arquitetura do software será no padrão **4+1** visões, com as visões **lógica, de processo, de desenvolvimento, física** e de **casos de uso**.

O padrão 4+1 é um modelo de visualização usado para descrever a arquitetura de sistemas intensivos em software, com base no uso de múltiplas visualizações simultâneas. Visualizações são usadas para descrever o sistema do ponto de vista de diferentes partes interessadas, como usuários finais, desenvolvedores, engenheiros de sistema e gerentes de projeto. É um modelo de **alto nível**, que busca atender aos requisitos funcionais e não funcionais.

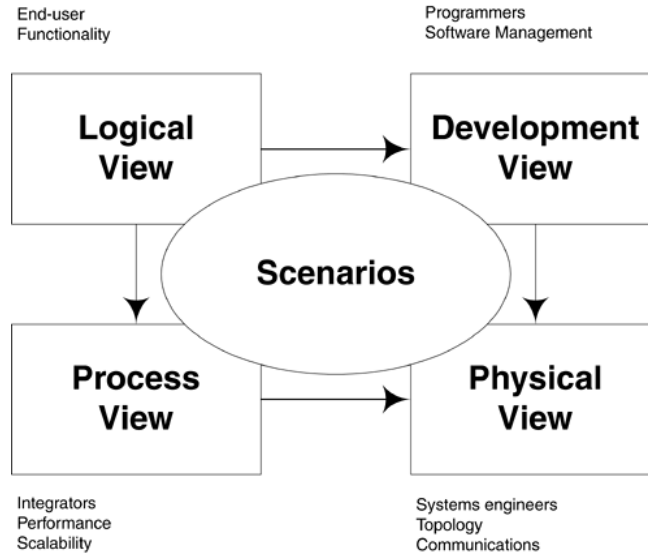


Figura 1. Imagem representando gráfica do modelo 4+1.

4. Visão Lógica

4.1. Descrição

A visão lógica é responsável por definir como a estrutura dos componentes do software será realizada e foi escolhida para auxiliar na construção da arquitetura proposta.

4.2. Detalhamento

O sistema será composto por um banco de dados centralizado **PostgreSQL**, que armazena os prontuários eletrônicos dos pacientes, bem como avatares P-Avatar associados. Os profissionais da saúde usam óculos de realidade aumentada e virtual para acessar o banco de dados e interagir com os P-Avatar dos pacientes.

Na parte do **servidor**, foi optado pelo uso da tecnologia **Node.js**, que é um ambiente de execução Javascript *server-side* projetado para criação de aplicativos de rede altamente escaláveis. Além disso, o framework de Node.js **Express.js** será utilizado para criação de uma **API REST**, que será responsável por realizar a comunicação entre o **frontend** e o banco de dados. Por essa API serão transmitidos dados de pacientes para eventuais cadastros no banco de dados e consultas de informações realizadas por médicos. Tanto a ferramenta Node.js como Express.js serão utilizadas para criação de um **gateway**, de onde serão chamados os outros serviços, como as interfaces e interações AR/VR

Na parte do **Client**, ou **frontend**, foi optado pela utilização da tecnologia **React**, que é uma biblioteca Javascript utilizada para criação de interfaces. A biblioteca React foi escolhida por sua alta compatibilidade com as demais ferramentas já selecionadas (Node.js e Express.js) e também por sua capacidade de componentização e encapsulamento, recursos muito desejados no projeto como esse.

Alguns dos componentes presente no frontend serão: *Router*, que será responsável por gerenciar todas as rotas da aplicação; *userAuth*, que será responsável por autenticar o usuário por meio de requisições ao servidor e *routerAuth*, que será responsável por autenticar as permissões de acesso às rotas pelos usuários.

Para as visualizações e interface **AR/VR**, as tecnologias selecionadas foram:

- **Babylon.js**, que é um mecanismo 3D em tempo real que usa uma biblioteca Javascript para exibir gráficos 3D em um navegador da Web via HTML5;
- **WebXR Device API**, que é uma interface de programação de aplicativos da Web que descreve o suporte para acessar dispositivos de realidade aumentada e realidade virtual;
- **Ajax**, que é um conjunto de técnicas de desenvolvimento da Web que usa várias tecnologias da Web no lado do cliente para criar aplicativos da Web assíncronos.

Além disso, também serão utilizadas tecnologias como HTML e CSS.

O sistema inclui também dois tipos de bots: um para os profissionais de saúde (assistente conselheiro) e outro para os pacientes (atendimento personalizado). Foi escolhido implementar dois tipos de bots: um para os profissionais de saúde (assistente conselheiro) e outro para os pacientes (atendimento personalizado). Isso permite que os profissionais de saúde tenham acesso a sugestões e informações adicionais enquanto atendem aos pacientes, e que os pacientes possam ter um atendimento mais personalizado e menos estressante.

5. Visão de Processos

5.1. Descrição

A visão de processos lida com os aspectos dinâmicos do sistema, explica os processos do sistema e como eles se comunicam, e se concentra no comportamento do sistema em tempo de execução.

5.2. Detalhamento

De forma detalhada, pode-ser dividir os processos e o comportamento do sistema nos seguintes passos:

- Quando um paciente chega à unidade de saúde, o profissional de saúde usa os óculos de AR/VR visualizando o P-Avatar do paciente, para identificá-lo e acessar seu prontuário eletrônico.
- O profissional de saúde pode então atualizar o prontuário eletrônico com novas informações e gerar ou atualizar o P-Avatar do paciente com auxílio da IA.
- Os profissionais de saúde podem consultar o P-Avatar do paciente dentro do Metaverso para obter informações e realizar modificações conforme necessário.

- Os pacientes podem acessar uma versão simplificada do seu P-Avatar através de um smartphone convencional e interagir com o bot de atendimento personalizado.

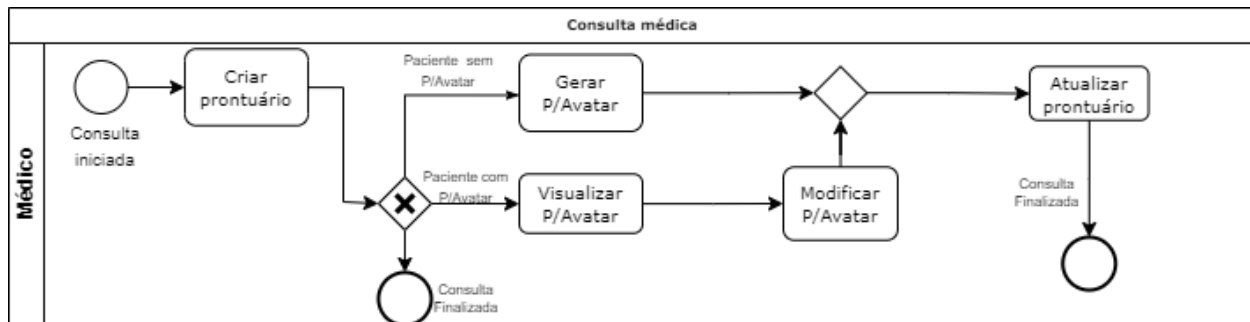


Figura 2. Diagrama de processos da consulta médica do sistema.

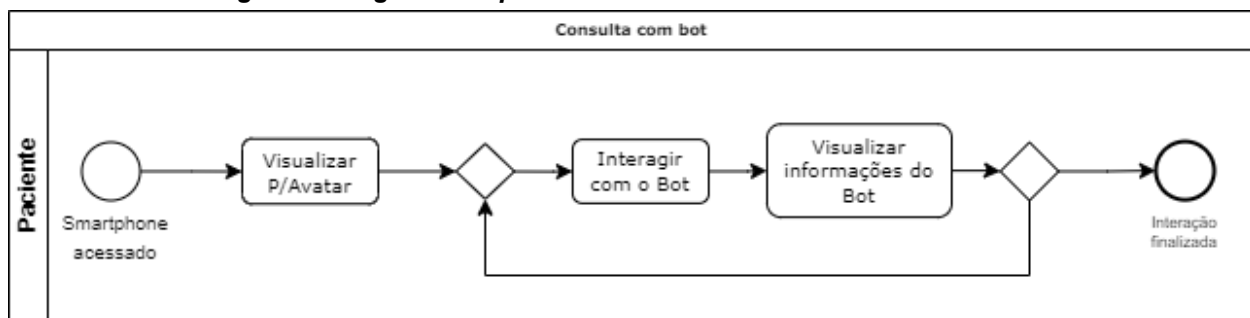


Figura 3. Diagrama de processos da consulta com o bot do sistema.

6. Visão de Desenvolvimento

6.1. Descrição

A visão de desenvolvimento ilustra o sistema da perspectiva de um programador e se preocupa com o gerenciamento de software. Essa visão também é conhecida como visão de implementação.

6.2. Detalhamento

O sistema proposto será desenvolvido com tecnologias de realidade aumentada (AR) e virtual (VR), bem como inteligência artificial (IA). O banco de dados será implementado em um servidor descentralizado e será acessado pelos profissionais de saúde através dos óculos de AR/VR.

Os bots serão desenvolvidos usando linguagem de programação compatível com o sistema.

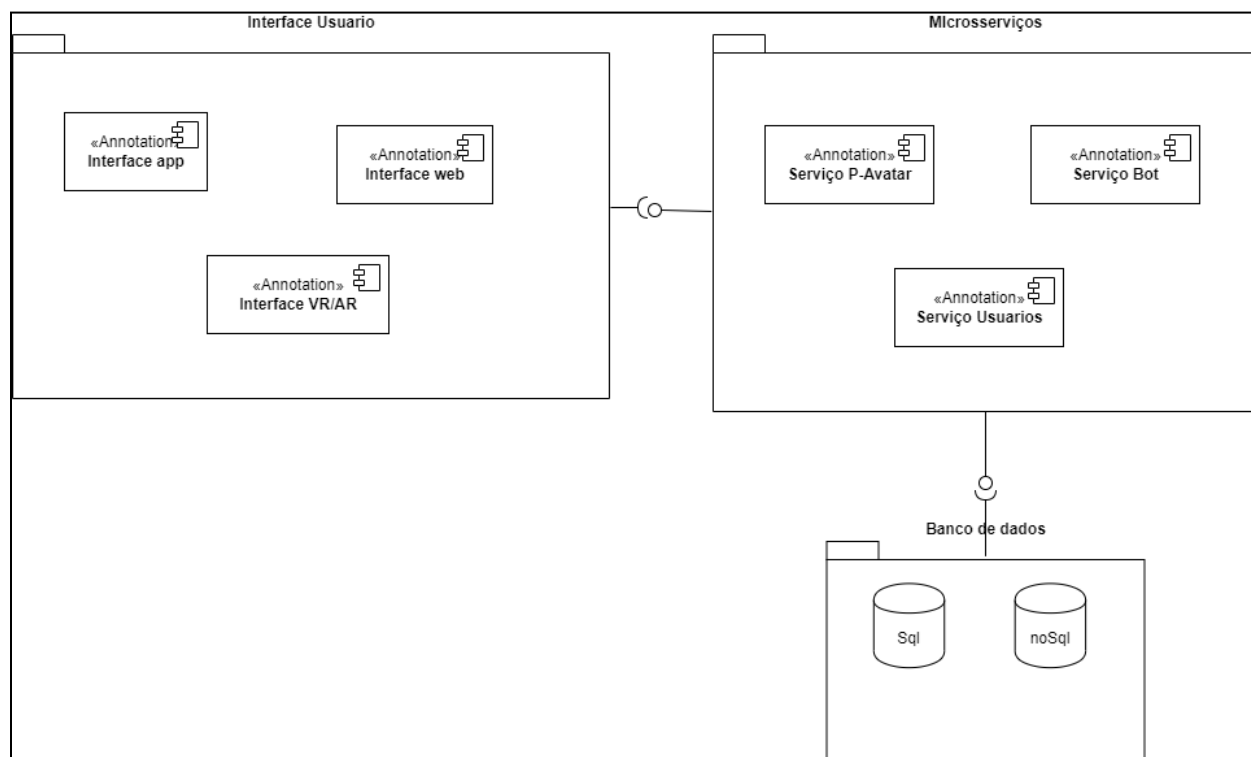


Figura 4. Imagem representando componentes do sistema.

7. Visão Física

7.1. Descrição

A visão física define se os componentes essenciais do sistema precisam de um ambiente físico próprio ou não para a interação/implantação, junto com os equipamentos que permitem a interação com tais componentes.

7.2. Detalhamento

O sistema será implementado em microserviços e será acessado por profissionais de saúde em diferentes unidades hospitalares através de óculos de AR/VR, então esses equipamentos junto com um servidor de acesso são necessários para a interação com o sistema. Além disso, os pacientes poderão acessar uma versão simplificada do sistema através de smartphones convencionais, para que possam visualizar seus dados de prontuário, consultas, fichas, etc.

7.3. Vantagens do Microserviço

Escalabilidade flexível: Os microserviços permitem que cada serviço seja escalado de forma independente para atender à demanda do recurso de aplicativo oferecido por esse serviço.

Resiliência: A independência do serviço aumenta a resistência a falhas do aplicativo.

Liberdade tecnológica: As arquiteturas de microserviços não seguem uma abordagem generalista..

Código reutilizável: A divisão do software em módulos pequenos e bem definidos permite que as equipes usem funções para várias finalidades.

W

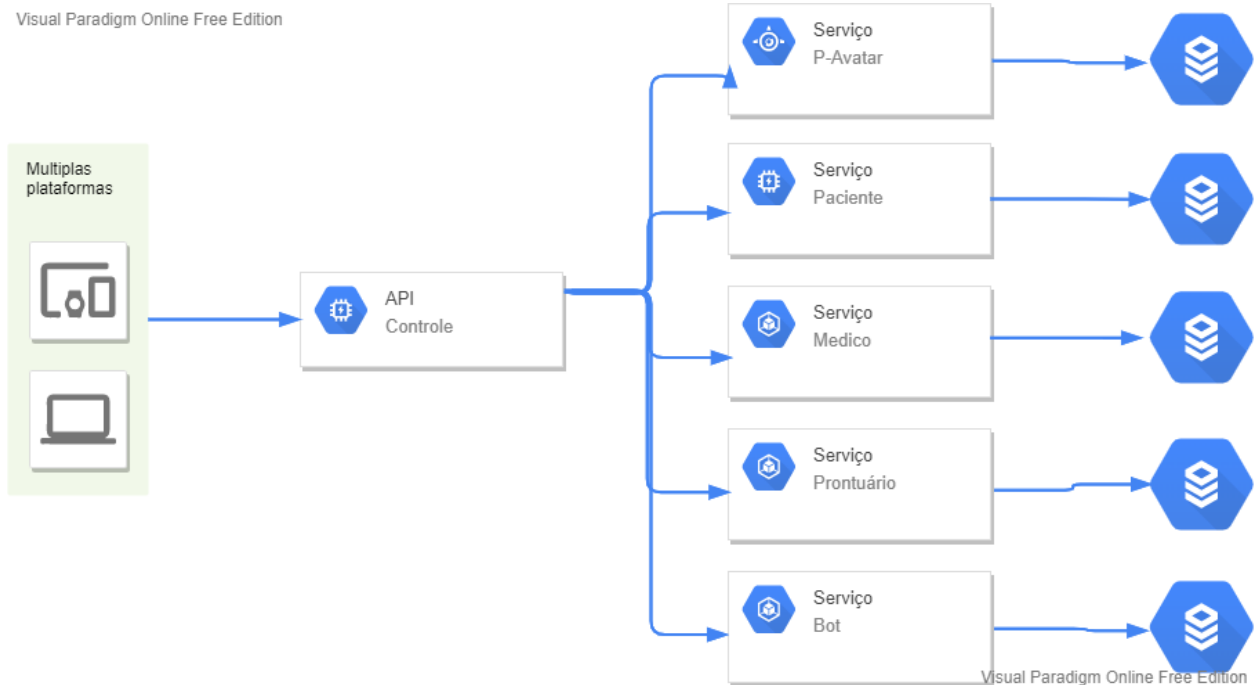


Figura 5. Imagem representando a visão física do sistema, com todos seus componentes.

Dispositivos do sistema:

- **Mobile:** Paciente
- **AR/VR:** Médicos
- **Desktop:** Médicos

Serviços identificados:

- **P-Avatar**
- **Prontuário**

- Bot
- Paciente
- Médico

8. Visão de Casos de Uso

8.1. Descrição

Para fornecer uma base para o planejamento da arquitetura e de todos os outros artefatos que serão gerados durante o ciclo de vida do software, é gerada, na análise de requisitos, uma visão chamada **visão de casos de uso**. Só existe uma visão de casos de uso para cada sistema. Ela ilustra os casos de uso e cenários que englobam o comportamento, as classes e riscos técnicos significativos do ponto de vista da arquitetura. A visão de casos de uso é refinada e considerada inicialmente em cada iteração do ciclo de vida do software.

8.2. Detalhamento

Os requisitos do sistema dão origem aos seguintes casos de uso, aqui descritos como pequenos cenários que serão representados em um diagrama de classe de uso.

- Quando um paciente chega à unidade de saúde, o profissional de saúde usa os óculos de AR/VR para identificar o paciente e acessar o prontuário eletrônico.
- O profissional de saúde pode atualizar o prontuário eletrônico e criar ou atualizar o P-Avatar do paciente com auxílio da IA.
- Os profissionais de saúde podem consultar o P-Avatar do paciente dentro do Metaverso para obter informações e realizar modificações conforme necessário.
- Os pacientes podem acessar uma versão simplificada do seu P-Avatar através de um smartphone convencional e interagir com o bot de atendimento personalizado.

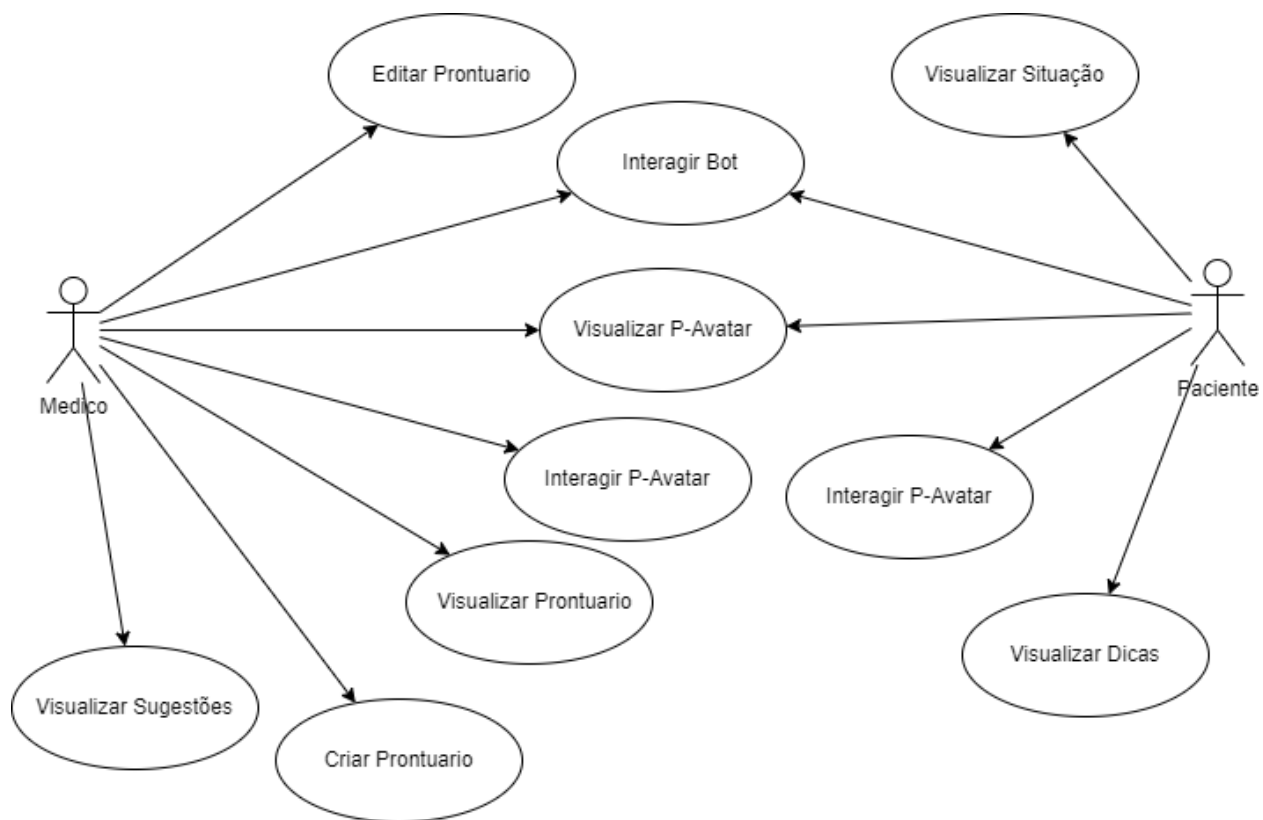


Figura 6. Diagrama de casos de uso UML representando os cenários de casos de uso vindos da análise de requisitos do sistema.