Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

**Documentação de Software**

**Hackers School**

Autores:

Vitor Felizatti

Pedro Otavio

Thiago Barros Gomes

Documento de especificação de requisitos do projeto

"H.S." 

Versão: 1.3

Projeto Interdisciplinar do 4º semestre do curso de Desenvolvimento de Software Em Multiplataforma desenvolvido na Faculdade de Tecnologia de Araras (FATEC), apresentado como Trabalho de Conclusão.

**Faculdade de Tecnologia de Araras**

**DESENVOLVIMENTO DE SOFTWARE EM MULTIPLATAFORMA**

**TRABALHO DE CONCLUSÃO DO 4ºSEMESTRE.**

Orientadores:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Ricardo Akira Harada

Engenharia de Software II

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Orlando Saraiva

Desenvolvimento Web II

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Nilton Rocco

Banco de Dados Relacional

**SUMÁRIO**

**1.0 INTRODUÇÃO5**

**1.1 Objetivo5**

**1.2 Escopo6**

**1.3 Definições, acrônimos e abreviações 7**

**1.5 Visão geral8  
2.0 DESCRIÇÃO GERAL9**

**2.1 Requisitos Funcionais** **10**

**2.2 Requisitos Não Funcionais** **10**

**2.2 Requisitos de interface** **11**

**2.3 Atributos de qualidade12**

**2.4 Características dos usuários13**

**2.5 Restrições14**

**2.6 Suposições e dependências15**

**3.0 Anexo** **16**

**4.0 Ferramentas** **17**

**4.1 Linguagens** **17**

**5.0 Diagrama Caso de Uso18  
5.1 Diagrama caso de uso - instituição18**

**5.2 Diagrama caso de uso - médico19  
5.3 Diagrama caso de uso - paciente20**

**6.0 Diagrama de Sequência21**

**6.1 Diagrama de sequência instituição21  
6.2 Diagrama de sequência médico****22**

**6.3 Diagrama de sequência paciente23**

**7.0 Banco de dados24**

**7.1 Modelo Lógico24**

**8.0 Referências25**

2.3 Requisitos de interface

Detalhar os requisitos de interface do usuário, como layout e navegabilidade.

2.4 Atributos de qualidade

Definir atributos de qualidade como confiabilidade, eficiência e manutenibilidade.

2.5 Características dos usuários

Descrever os tipos de usuários e suas necessidades.

2.6 Restrições

Listar limitações técnicas, de tempo e de recursos.

2.7 Suposições e dependências

Descrever suposições feitas durante o desenvolvimento e dependências externas.

3.0 Anexo

Incluir anexos relevantes, como diagramas adicionais ou documentação técnica específica.

4.0 Ferramentas

4.1 Linguagens

Listar as linguagens de programação e ferramentas usadas, como Django, Flask, PostgreSQL, etc.

5.0 Diagrama Caso de Uso

5.1 Diagrama caso de uso - usuário

Incluir um diagrama representando os casos de uso para usuários.

5.2 Diagrama caso de uso - rank

Diagrama para o sistema de ranking.

5.3 Diagrama caso de uso - perguntas

Diagrama para a funcionalidade de perguntas e respostas.

6.0 Diagrama de Sequência

6.1 Diagrama de sequência usuário

Diagrama representando a sequência de interações dos usuários.

6.2 Diagrama de sequência rank

Diagrama de sequência para a funcionalidade de ranking.

6.3 Diagrama de sequência perguntas

Diagrama de sequência para perguntas e respostas.

7.0 Banco de dados

7.1 Modelo Lógico

Apresentar o modelo lógico do banco de dados, como tabelas e relações.

8.0 Referências

Listar todas as referências usadas na documentação.

Para começar, você pode criar um documento de texto com essas seções e ir preenchendo conforme necessário. Se precisar de ajuda com algum conteúdo específico ou com a geração de diagramas, sinta-se à vontade para perguntar!

**1.0 INTRODUÇÃO**

O projeto Hackers School visa criar uma rede social inovadora focada em hackers e entusiastas de segurança cibernética. Esta introdução fornece uma visão geral do projeto e estabelece as bases para as seções subsequentes, detalhando os objetivos, escopo, definições e estrutura geral da documentação.

**1.1 OBJETIVO**

O principal objetivo do projeto Hackers School é criar uma plataforma inovadora que funcione como uma rede social dedicada a hackers e entusiastas de segurança cibernética. A Hackers School visa proporcionar um ambiente colaborativo e educacional onde os usuários possam:

* Compartilhar conhecimento e experiências relacionadas à segurança cibernética.
* Participar de desafios e competições de hacking (CTFs - Capture The Flag).
* Acessar uma biblioteca de recursos educativos, incluindo tutoriais e artigos.
* Participar de fóruns de discussão e webinars.
* Conectar-se com outros profissionais e entusiastas da área para troca de ideias e networking.

A plataforma pretende promover o aprendizado contínuo e a disseminação de boas práticas em segurança cibernética, contribuindo para a formação de uma comunidade ativa e engajada.

1.2 **ESCOPO**

O projeto Hackers School inclui uma variedade de funcionalidades e serviços destinados a criar uma experiência abrangente para seus usuários. O escopo do projeto abrange:

* **Perfis de Membros**: Criação e gerenciamento de perfis individuais onde os usuários podem exibir suas habilidades, conquistas e interesses.
* **Fóruns**: Plataformas de discussão onde os membros podem trocar ideias, fazer perguntas e compartilhar conhecimento.
* **Biblioteca de Recursos**: Coleta de tutoriais, artigos e outros materiais educativos relacionados à segurança cibernética.
* **Desafios e CTFs (Capture The Flag)**: Competições que permitem aos usuários testar e aprimorar suas habilidades de hacking em um ambiente seguro e controlado.
* **Eventos e Webinars**: Organização de eventos online, como webinars e workshops, para educação contínua e networking.
* **Sistema de Rank**: Mecanismo para classificar os usuários com base em sua participação e desempenho em desafios e atividades na plataforma.
* **Perguntas e Respostas**: Seção dedicada onde os membros podem postar perguntas e obter respostas da comunidade.

Estas funcionalidades visam proporcionar um ambiente rico e colaborativo, promovendo o aprendizado contínuo e a interação entre os membros da comunidade de segurança cibernética.

**1.3 ACRÔNIMOS**

**CTF (Capture The Flag):** Competição de segurança cibernética onde os participantes tentam encontrar "bandeiras" escondidas em programas ou sistemas vulneráveis.

**UI (User Interface):** Interface do usuário, que é a parte do software com a qual o usuário interage diretamente.

**UX (User Experience):** Experiência do usuário, referindo-se à qualidade da interação do usuário com o sistema.

**API (Application Programming Interface):** Conjunto de rotinas e ferramentas para construir software e aplicações.

**HTTP (HyperText Transfer Protocol):** Protocolo utilizado para transferência de dados na web.

**HTTPS (HyperText Transfer Protocol Secure):** Versão segura do HTTP, utilizando criptografia para proteger os dados transferidos.

**SQL (Structured Query Language):** Linguagem de programação utilizada para gerenciar e manipular bancos de dados relacionais.

**REST (Representational State Transfer):** Arquitetura que utiliza HTTP para comunicação entre sistemas.

**JWT (JSON Web Token):** Padrão aberto para a criação de tokens de acesso que permitem a troca segura de informações entre partes.

**SaaS (Software as a Service):** Modelo de distribuição de software onde as aplicações são hospedadas por um provedor de serviços e disponibilizadas aos clientes pela internet.

**DDoS (Distributed Denial of Service):** Tipo de ataque cibernético onde múltiplos sistemas comprometidos são usados para sobrecarregar um alvo, tornando-o indisponível.

Esta lista será expandida conforme o desenvolvimento do projeto, garantindo que todos os termos técnicos sejam claramente definidos para facilitar a compreensão de todos os envolvidos.

**1.5 VISÃO GERAL**

Este documento de requisitos detalha as funcionalidades e características do projeto Hackers School, organizado nas seguintes seções principais:

* Introdução: Descreve os objetivos, escopo, definições e a visão geral do documento.
* Descrição Geral: Abrange requisitos funcionais e não funcionais, requisitos de interface, atributos de qualidade, características dos usuários, restrições, suposições e dependências.
* Anexos: Inclui informações adicionais relevantes ao projeto.
* Ferramentas: Lista as linguagens e ferramentas utilizadas.
* Diagramas de Caso de Uso e Sequência: Apresenta diagramas que ilustram os casos de uso e sequências de interação do sistema.
* Banco de Dados: Descreve o modelo lógico do banco de dados.
* Referências: Cita todas as fontes de informações utilizadas na documentação.

**2.0 DESCRIÇÃO GERAL**

2.1 Requisitos Funcionais

Define as funcionalidades principais que o sistema deve oferecer, como criação de perfis, fóruns de discussão, desafios de segurança (CTFs), biblioteca de recursos, eventos e webinars.

2.2 Requisitos Não Funcionais

Especifica os atributos de qualidade do sistema, como desempenho, segurança, usabilidade, confiabilidade e manutenção.

2.3 Requisitos de Interface

Detalha os requisitos de interface do usuário, incluindo layout, navegabilidade e design responsivo.

2.4 Atributos de Qualidade

Identifica os atributos de qualidade que o sistema deve atender, como escalabilidade, eficiência e compatibilidade.

2.5 Características dos Usuários

Descreve os diferentes tipos de usuários que interagirão com o sistema, suas necessidades e níveis de habilidade técnica.

2.6 Restrições

Lista as limitações técnicas, de tempo, recursos e regulatórias que impactam o desenvolvimento do projeto.

2.7 Suposições e Dependências

Identifica as suposições feitas durante o planejamento do projeto e as dependências externas que podem influenciar seu sucesso.

**2.1 REQUISITOS FUNCIONAIS**

Os requisitos funcionais definem as funcionalidades principais que o sistema Hackers School deve oferecer para proporcionar uma experiência completa e satisfatória aos usuários:

RF01: Cadastrar usuário - obrigatório

Permitir o registro de novos usuários com informações básicas e personalização de perfil.

RF02: Editar perfil - obrigatório

Permitir a edição e atualização das informações do perfil do usuário.

RF03: Criar tópico no fórum - obrigatório

Permitir que os usuários criem novos tópicos de discussão nos fóruns.

RF04: Participar em fóruns - obrigatório

Permitir que os usuários participem de discussões nos fóruns, respondendo e comentando em tópicos.

RF05: Organizar desafios CTF - obrigatório

Permitir a criação e gestão de competições de segurança do tipo Capture The Flag.

RF06: Participar em CTFs - obrigatório

Permitir que os usuários participem de competições CTF e acumulem pontos.

RF07: Acessar biblioteca de recursos - obrigatório

Disponibilizar uma coleção de tutoriais, artigos e materiais educativos.

RF08: Gerenciar biblioteca de recursos - obrigatório

Permitir a adição, edição e categorização de conteúdos na biblioteca.

RF09: Planejar eventos e webinars - obrigatório

Facilitar a organização e agendamento de eventos online, como webinars e workshops.

RF10: Participar de eventos e webinars - obrigatório

Permitir que os usuários se inscrevam e participem de eventos online.

RF11: Sistema de rank - obrigatório

Implementar um mecanismo de classificação e pontuação baseado na participação dos usuários.

RF12: Perguntas e respostas - obrigatório

Permitir que os usuários postem perguntas e respondam às dúvidas da comunidade.

RF13: Votar em respostas - desejável

Permitir que os usuários votem nas respostas para destacar as melhores contribuições.

RF14: Notificações de atividades - desejável

Enviar notificações aos usuários sobre novas respostas, mensagens ou atividades relevantes.

**2.1.1 REQUISITOS NÃO FUNCIONAIS**

**2.1.1 Desempenho:**

* Tempo de resposta: O sistema deve responder rapidamente às solicitações dos usuários, garantindo uma experiência ágil.
* Escalabilidade: Deve ser capaz de lidar com um aumento no número de usuários ou de carga de trabalho sem comprometer o desempenho.

**2.1.2 Segurança:**

* Autenticação e autorização: O sistema deve garantir que apenas usuários autorizados tenham acesso às funcionalidades e dados pertinentes.
* Proteção de dados: As informações dos usuários devem ser protegidas contra acesso não autorizado ou vazamento.

**2.1.3 Usabilidade:**

* Interface amigável: O sistema deve ser intuitivo e fácil de usar, com uma interface de usuário clara e bem projetada.
* Documentação: Deve fornecer documentação adequada para orientar os usuários sobre como utilizar o sistema de forma eficaz.

**2.1.4 Confiabilidade:**

* Disponibilidade: O sistema deve estar disponível para uso a maior parte do tempo, minimizando o tempo de inatividade não planejado.
* Tolerância a falhas: Deve ser capaz de lidar com falhas de forma robusta, minimizando o impacto nos usuários.

**2.1.5 Manutenção:**

* Facilidade de manutenção: O código-fonte e a arquitetura do sistema devem ser bem organizados e documentados, facilitando a manutenção e a evolução futura.
* Facilidade de atualização: Deve ser possível realizar atualizações no sistema de forma rápida e sem interrupções significativas no serviço.

**2.2 REQUISITOS DE INTERFACE**

2.2.1 RI01: Layout Responsivo

Descrição: O sistema deve ser responsivo e se adaptar a diferentes tamanhos de tela, garantindo uma experiência consistente em dispositivos desktop, tablets e smartphones.

Detalhes: O layout deve se ajustar dinamicamente para otimizar a usabilidade em diferentes dispositivos, mantendo a legibilidade e a acessibilidade.

2.2.2 RI02: Navegação Intuitiva

Descrição: A navegação dentro do sistema deve ser intuitiva e fácil de entender para os usuários.

Detalhes: Deve haver uma barra de navegação clara e consistente em todas as páginas, com categorias bem definidas e links para as seções principais do sistema, como cursos, perfis de usuário e recursos adicionais.

2.2.3 RI03: Interface Amigável para o Usuário

Descrição: A interface do usuário deve ser amigável e atrativa, promovendo uma experiência agradável para os usuários.

Detalhes: Use elementos visuais, como ícones, cores e tipografia, de forma consistente e atraente. Evite sobrecarregar as páginas com informações desnecessárias e mantenha o design limpo e organizado.

2.2.4 RI04: Feedback Visual

Descrição: O sistema deve fornecer feedback visual claro para as ações dos usuários.

Detalhes: Por exemplo, ao enviar um formulário, exibir uma mensagem de confirmação ou um indicador de carregamento para informar o progresso da ação. Isso ajuda a manter os usuários informados sobre o status de suas interações.

**2.3 ATRIBUTOS DE QUALIDADE**

AQ01: Segurança e Privacidade: A segurança dos dados de saúde é fundamental. Implementação de medidas robustas de segurança, como criptografia e autenticação, para proteger as informações confidenciais dos pacientes contra acessos não autorizados.

OBRIGATÓRIO

AQ02: Design Responsivo: O P.S.U. é acessível em diferentes dispositivos, como desktops, tablets e smartphones. O design responsivo garante que a interface se ajuste automaticamente ao tamanho da tela, proporcionando uma experiência consistente e adaptável em todos os dispositivos.

DESEJÁVEL

AQ03:Comunicação Integrada: A interface permite a comunicação entre profissionais de saúde e pacientes de forma segura e eficiente. Recursos como mensagens seguras, videochamadas e solicitação de consultas para melhorar a comunicação e o engajamento do paciente.

DESEJÁVEL

AQ04: Integração com Sistemas Externos: É importante que P.S.U. possa se integrar com outros sistemas de saúde, como sistemas de laboratório, sistemas de prescrição eletrônica e sistemas de faturamento, para facilitar o compartilhamento de informações e a colaboração entre diferentes entidades de saúde.

OBRIGATÓRIO

**2.4 CARACTERISTICAS DOS USUÁRIOS**

Este *software* destina-se a instituições hospitalares, com a finalidade de gerenciar prontuários e facilitar sua visualização por médicos, pacientes e seus familiares.

INSTITUIÇÃO: Será necessário treinamento de todos ou dos principais, para que desenvolva perfeitamente as funcionalidades e requisitos do programa dentro da instituição.

MÉDICOS: graduado em ciências médicas, será necessário o código CRM e treinamento junto a instituição.

PACIENTES: será necessário leitura, apenas, sem experiencia ou conhecimento.

**2.5 RESTRIÇÕES**

R01: Padrões de conformidade: A aplicação segue padrões de conformidade relevantes da Lei Geral de Proteção de Dados (LGPD). Esses padrões impõem regras e restrições sobre a privacidade e segurança dos dados pessoais de saúde. Somente o paciente e/ou seu responsável poderá autorizar o compartilhamento de dados.

R02: Ambientes operacionais: A aplicação de prontuário online único pode ser implantada em diferentes ambientes operacionais, como servidores locais ou em nuvem. As restrições podem variar dependendo do ambiente escolhido, incluindo requisitos de hardware, sistema operacional e configuração de rede.

R03: Segurança e privacidade: Medidas de segurança serão implementadas para proteger os dados dos pacientes contra acesso não autorizado, vazamento ou modificação indevida. Restrições adicionais incluirão criptografia de dados, autenticação de usuários, controle de acesso baseado em funções.

R04: Integração de sistemas: O P.S.U. poderá integrar-se a outros sistemas de saúde. Restrições podem surgir em relação aos padrões de integração, como a utilização de interfaces padronizadas e a compatibilidade com os formatos de dados utilizados pelos sistemas existentes.

R05: Desempenho: A aplicação é projetada levando em consideração o desempenho para lidar com um grande número de pacientes, registros e acessos simultâneos. Restrições podem ser impostas em relação ao tempo de resposta, capacidade de processamento, armazenamento de dados e largura de banda.

R06: Requisitos de acesso: Dependendo das regulamentações e políticas internas, podem existir restrições em relação ao acesso aos registros médicos eletrônicos. Isso pode incluir requisitos de autenticação como restrições de acesso baseadas em funções.

**2.6 SUPOSIÇÕES E PENDÊNCIAS**

Existem suposições e dependências que podem afetar a especificação. Esses fatores podem incluir:

SP01:Disponibilidade de acesso à Internet: Será necessário acesso à Internet.

SP02: Capacidade de *hardware* dos dispositivos dos usuários: Os usuários poderão acessar dos seguintes dispositivos (computadores, *smartphones*, tablets) com capacidade suficiente para executar a aplicação.

SP03: Mudanças nas regulamentações de saúde: As regulamentações relacionadas aos prontuários eletrônicos podem ser atualizadas ao longo do tempo. Mudanças nas regulamentações locais, regionais ou nacionais podem afetar os requisitos da aplicação, exigindo ajustes para cumprir as novas normas.

SP04: Mudanças nas práticas médicas e protocolos de tratamento: As práticas médicas e os protocolos de tratamento podem evoluir com base em novas descobertas científicas e avanços médicos. Isso pode influenciar os requisitos da aplicação, como a inclusão de campos adicionais para registrar informações específicas ou a necessidade de acompanhar novos parâmetros de saúde.

SP05: Evolução da tecnologia de segurança e criptografia: À medida que as tecnologias de segurança e criptografia avançam, os requisitos de segurança da aplicação também podem ser afetados. A necessidade de adotar novos padrões de segurança ou criptografia pode surgir para garantir a proteção adequada dos dados.

SP06: Mudanças na infraestrutura de TI: Mudanças na infraestrutura de TI da organização ou no ambiente operacional podem impactar os requisitos do prontuário online único. Por exemplo, uma migração para uma nova plataforma de hospedagem em nuvem ou a atualização de sistemas operacionais.

SP07: Mudanças nas expectativas dos usuários: À medida que os usuários se familiarizam com a aplicação e suas funcionalidades, suas expectativas e necessidades podem evoluir. O feedback dos usuários, pesquisas de satisfação e análise de uso podem levar a mudanças nos requisitos para melhor atender às expectativas dos usuários.

**3 ANEXO**

**Métodos utilizados:**

*Brainstorming*: Utilizamos para levantar ideias e dar início a uma discussão que ainda não está formada. Assim rapidamente se levanta uma grande quantidade de ideias.

Para sabermos a necessidade de quem vai utilizar o produto ou saber detalhes sobre o funcionamento de um processo qualquer. Coletando assim as informações como são, permitindo aferir a usabilidade de um processo.

Pensando em grupo, com participantes que podem ser internos ou externos ao projeto. No *brainstorming*, os participantes falam livremente, sem coibição.

Protótipos: Utilizamos para coletar um *feedback* mais concreto das partes interessadas. Sendo atualizado diversas vezes, até solidificar os requisitos.

Um subproduto da solução principal que deve mostrar as funcionalidades que se desejem avaliar durante o levantamento de requisitos de um projeto.

**4.0 FERRAMENTAS**

FER01: *Visual Studio Code* – codificação

FER02: *Heidi* – banco de dados

FER03: *Zeal* – documentação

FER04: *Draw.io*

**4.1 LINGUAGENS**

LING01: *JavaScript*

LING02: HTML

LING03: CSS

LING04: PHP

LING5: SQL

5.0 DIAGRAMA DE CASO DE USO

**5.1 INSTITUIÇÃO**

**Diagrama

Descrição gerada automaticamente**

Figura 01 – Diagrama de Caso de Uso

**5.2 MÉDICO**

**Diagrama

Descrição gerada automaticamente**

Figura 2 – Diagrama de Caso de Uso

**5.3 PACIENTE**

**Diagrama

Descrição gerada automaticamente**

Figura 3 – Diagrama de Caso de Uso

6.0 DIAGRAMA DE SEQUÊNCIA

**6.1 INSTITUIÇÃO**

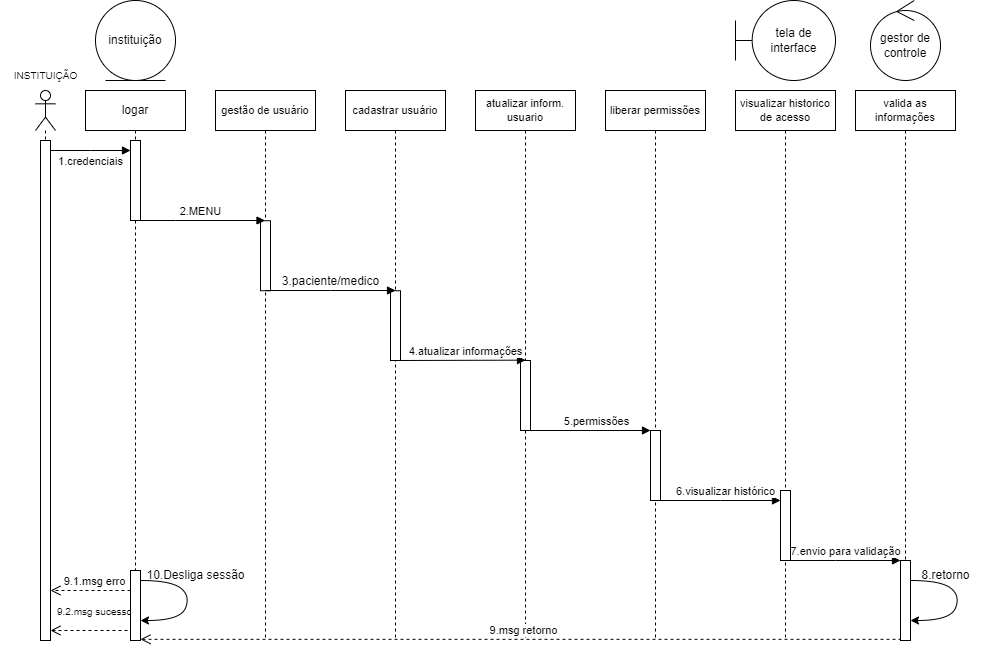


Figura 04 representa DS da instituição para o cadastramento de novos usuários sendo ele médico ou paciente, até a interface, onde após entra o gestor de controle ou o programador, alimentando o banco de dados e averiguando erros, administrando o P.S.U. remotamente.

**6.2 MÉDICO**

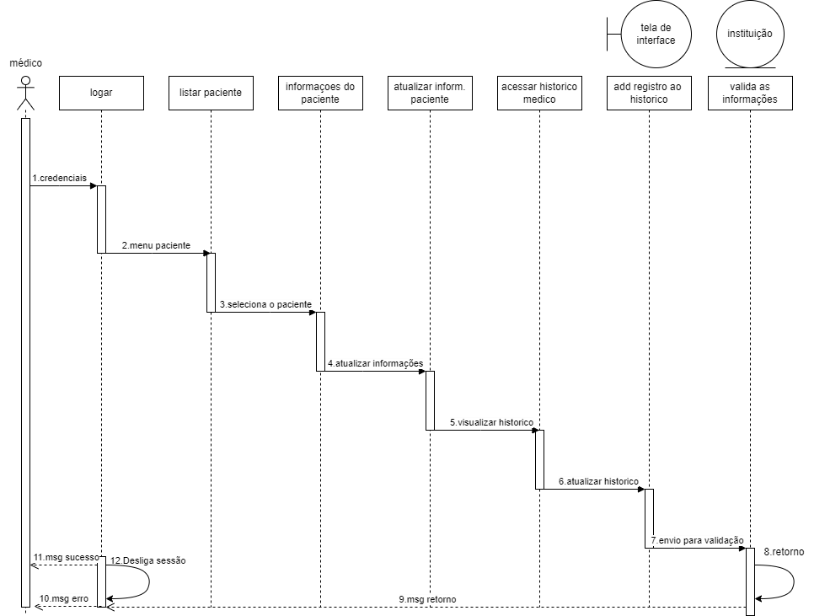


Figura 05 do DS representando o percurso sequencial médico ao sistema para o atendimento ao paciente, até a interface, onde após a instituição médica ou gestor, quem valida as informações. e retorna a informação ao sistema, o mesmo(P.S.U.) devolve uma mensagem de erro ou sucesso.

**6.3 PACIENTE**

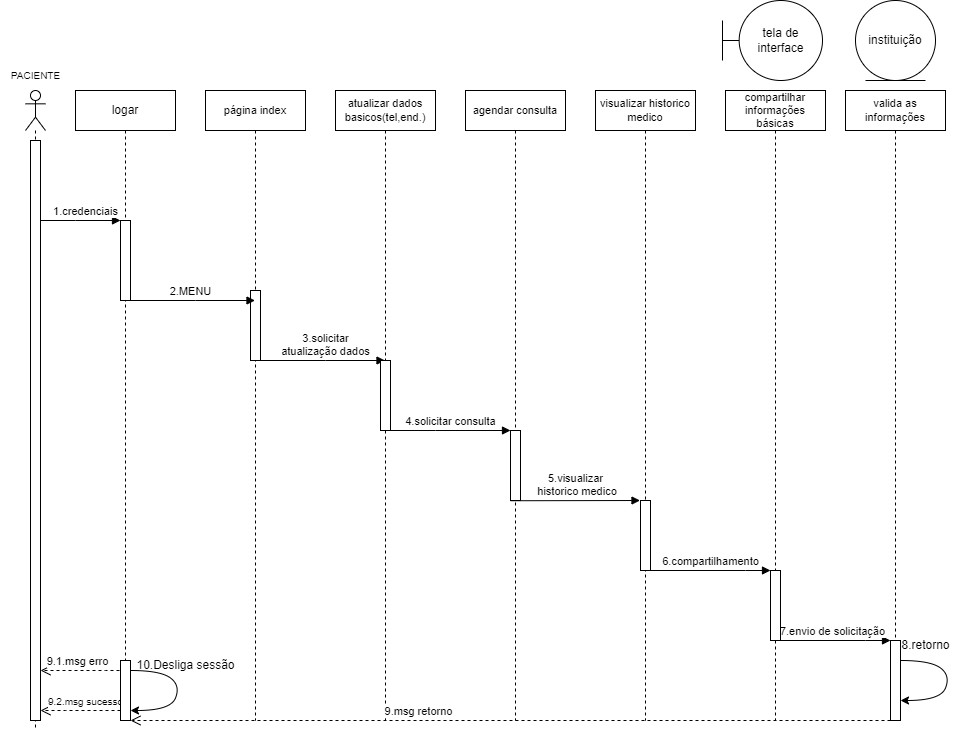


Figura 06 do DS representando o percurso sequencial paciente ao sistema para atualização de dados básicos, agendamento, visualização de seu histórico médico e compartilhamento, até a interface, onde após a instituição médica ou gestor, quem valida as informações solicitadas. e retorna a informação ao paciente pelo P.S.U, o mesmo devolve uma mensagem de erro ou sucesso.

**7.0 BANCO DE DADOS**

**7.1 Modelo lógico**

**Interface gráfica do usuário

Descrição gerada automaticamente com confiança média**

Figura 07 Modelo lógico do banco de dados representando as relações dos dados armazenados e manipulados pelo P.S.U.

**8. Referências**

Prontuários pesquisados e usados como referência:

https://www.gov.br/pt-br/noticias/saude-e-vigilancia-sanitaria/2021/04/voce-conhece-o-conecte-sus

[Você conhece o Conecte SUS? (www.gov.br)](https://www.gov.br/pt-br/noticias/saude-e-vigilancia-sanitaria/2021/04/voce-conhece-o-conecte-sus)

O prontuário eletrônico do paciente no sistema de saúde brasileiro: uma realidade para os médicos?

<https://www.saudedireta.com.br/docsupload/1370271458PEP.pdf>

[1370271458PEP.pdf (saudedireta.com.br)](https://www.saudedireta.com.br/docsupload/1370271458PEP.pdf)

LGPD: Leis regulam acesso ao prontuário e defendem privacidade do paciente — Empresa Brasileira de Serviços Hospitalares (www.gov.br)

<https://www.gov.br/ebserh/pt-br/hospitais-universitarios/regiao-sul/hu-ufsc/comunicacao/noticias/leis-regulam-acesso-ao-prontuario-e-defendem-privacidade-do-paciente>