

Saúde Integrada

Documento de Requisitos

Ellen Conceição Souza Barbosa

Rodrigo Costa Neves

Versão 1.0

Histórico de Alterações

Data	Versão	Descrição	Autor(es)
16/10/2024	1.0	Versão inicial do documento	Ellen Conceição Souza Barbosa
06/11/2024	1.1	Edição inicial do documento	Rodrigo Costa Neves
25/11/2024	1.2	Edição do Sumário e estrutura dos títulos	Ellen Conceição Souza Barbosa
25/11/2024	1.3	Adição de Diagramas de Caso de Uso	Rodrigo Costa Neves
22/01/2025	1.4	Ajuste nos requisitos não funcionais	Ellen Conceição Souza Barbosa

Conteúdo

1. Introdução

4 1.1 Visão geral do documento

4	1.2	Convenções, termos e abreviações
4	1.2.1	Identificação dos requisitos
4	1.2.2	Prioridades dos requisitos
4	2.	Descrição geral do sistema
5	2.1	Cliente
5	2.2	Usuário
5	2.3	Visão Geral do Sistema
6	3.	Requisitos funcionais (casos de uso)
6	3.1.1	[RF001] Novo Paciente
6	3.1.2	[RF 002] Visualizar Paciente
7	3.1.3	[RF 003] Novo Atendimento
8	3.1.4	[RF 004] Quantidade de Atendimento
8	3.1.5	[RF005] Histórico de Atendimento
9	3.1.6	[RF 006] Bairros Mais Atendidos
10	3.1.7	[RF 007] Pesquisar Paciente
11	4.	Requisitos não-funcionais
11	4.1	Usabilidade
11	4.1.1	[NF001] Interface Amigável
12	4.1.2	[NF002] Componentes WEB
12	4.2	Software
12	4.2.1	[NF003] Banco de Dados *adicionar o nome do banco de dados*
12	4.2.2	[NF004] Linguagem Python
13	4.2.3	[NF004] Framework Django
13	4.2.4	[NF005] Django Rest Framework
13	4.2.5	[NF006] Vue.js
13	4.3	Desempenho
13	4.3.1	[NF005] Agilidade na Execução das Operações
13	5.	Diagramas
14	5.1	Diagrama de Caso de Uso
14	5.2	Diagrama de Classe
15	6.	Referências

1. Introdução

Este documento especifica os requisitos do Sistema de Gerenciamento de atendimento

psicológicos, havendo a necessidade de interação entre UPA, SAMU, CAPS e ATS.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 – Descrição geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3 – Requisitos funcionais (casos de uso):** especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 4 – Requisitos não-funcionais:** especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho e software.
- **Seção 5 – Referências:** apresenta referências para outros documentos utilizados para a confecção deste documento.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do identificador do requisito, de acordo com a especificação a seguir:

[identificador do requisito]

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] para os requisitos funcionais e [NF001] para os não-funcionais e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

Esta seção descreve superficialmente o cliente, os futuros usuários e fornece uma visão geral do IF-SAMU.

2.1 Cliente

O Serviço de Atendimento Móvel de Urgência (SAMU) é um serviço público de saúde que opera em todo o Brasil, responsável por prestar atendimento pré-hospitalar de urgência e emergência. O SAMU atua em situações de risco de morte, como acidentes, paradas cardíacas, crises psiquiátricas e outras emergências médicas, enviando equipes especializadas até o local da ocorrência.

O principal objetivo do SAMU é garantir um atendimento rápido e eficaz, contribuindo para a redução de mortalidade e complicações graves, estabilizando o paciente até que ele possa ser encaminhado a uma unidade de saúde apropriada, como hospitais e Unidades de Pronto Atendimento (UPA).

No contexto específico dos atendimentos psiquiátricos, o SAMU desempenha um papel essencial ao responder prontamente a crises de saúde mental, como surtos psicóticos, tentativas de suicídio ou agitações severas. Para garantir um cuidado integral, o SAMU trabalha em conjunto com outros serviços de saúde, como as UPAs, Atendimento Básico de Saúde (ATS) e os Centros de Atenção Psicossocial (CAPS), formando uma rede integrada que visa proporcionar atendimento contínuo e coordenado.

Contudo, a forma como os dados e informações são atualmente gerenciados, via planilhas compartilhadas, apresenta falhas que comprometem a segurança e integridade dos dados. Por isso, este projeto visa criar um sistema mais robusto e seguro para o controle dos atendimentos psiquiátricos e regulação dos medicamentos, garantindo maior confiabilidade e integração entre os órgãos envolvidos, otimizando o fluxo de informações e o acompanhamento dos pacientes.

2.2 Usuário

O sistema de gestão de atendimentos psiquiátricos, denominado *Saúde Integrada*, será utilizado a princípio pelos funcionários do SAMU, particularmente o médico regulador.

2.3 Visão Geral do Sistema

O sistema de gestão de atendimentos psiquiátricos para o SAMU de Guanambi, Bahia, será desenvolvido para atender às necessidades dos médicos reguladores e melhorar a organização e o controle dos atendimentos psiquiátricos. Seu objetivo principal é automatizar o processo de cadastro e gerenciamento dos pacientes e seus atendimentos, substituindo os métodos manuais atualmente utilizados, como planilhas compartilhadas, que apresentam falhas de compatibilidade e segurança.

O sistema permitirá que o médico regulador tenha acesso completo para cadastrar novos pacientes e registrar atendimentos psiquiátricos, garantindo um histórico completo e

detalhado de cada paciente. Cada atendimento poderá ser vinculado ao local onde foi realizado (UPA, SAMU, CAPS ou Atenção Básica), facilitando o controle e o monitoramento dos casos de saúde mental.

Além disso, o sistema oferecerá funcionalidades que permitirão visualizar a quantidade total de pacientes cadastrados, a quantidade de atendimentos realizados e gerar relatórios sobre os bairros com maior número de atendimentos. Essa ferramenta possibilitará uma visão clara e abrangente dos dados, ajudando a identificar padrões e direcionar recursos de forma mais eficaz.

O sistema também contará com ferramentas de busca eficiente, permitindo ao médico regulador localizar rapidamente informações sobre um paciente específico, seu histórico de atendimentos e o local onde ocorreram. O sistema será fundamental para melhorar a integração entre UPA, SAMU, ATS e CAPS, garantindo que os dados dos atendimentos psiquiátricos sejam armazenados de maneira segura e acessível, além de proporcionar insights valiosos sobre o fluxo e a distribuição dos atendimentos psiquiátricos na cidade de Guanambi.

3. Requisitos funcionais

3.1.1 [RF001] Novo Paciente

Descrição do caso de uso: O usuário deve poder registrar um novo paciente no sistema.

Ator: Médico Regulador

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário entra com os dados do paciente no formulário de cadastro. **Saídas e pós-condição:**

O sistema retorna uma mensagem informando o resultado da operação..

Fluxo de eventos principal

1. Usuário escolhe a ação de novo paciente.
2. O sistema oferece o formulário de cadastro do paciente com o código automaticamente preenchido.
3. O usuário entra com os dados do paciente e submete o formulário.
4. O sistema insere os dados submetidos no banco de dados.
5. O sistema retorna para o usuário uma mensagem informando que a operação foi realizada com sucesso.
6. O sistema pode retornar uma mensagem de paciente já cadastrado, caso o paciente já possua cadastro no banco de dados

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.2 [RF 002] Visualizar Paciente

Descrição do caso de uso: O usuário deve poder realizar a operação de visualização de pacientes cadastrados.

Ator: Secretário, Direção

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O paciente deverá estar devidamente cadastrado para poder ser visualizado. **Saídas e pós-condição:**

O sistema retorna todos os dados disponíveis referentes aos pacientes..

Fluxo de eventos principal

1. O usuário escolhe a ação de listar pacientes.
2. O sistema lista os pacientes para que o usuário selecione o que deseja visualizar.
3. O usuário seleciona o paciente desejado.
4. O usuário clica no botão "Visualizar".
5. O sistema exibe todos os dados do paciente e os atendimentos.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.3 [RF 003] Novo Atendimento

Descrição do caso de uso: O usuário deve poder realizar a operação de novo atendimento.

Ator: Secretário

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O paciente selecionado deve estar cadastrado. Caso não esteja, é necessário cadastrar um novo paciente.

Saídas e pós-condição:

O sistema retorna uma mensagem confirmando o novo atendimento realizado ao paciente já cadastrado.

Fluxo de eventos principal

1. Usuário escolhe a ação de novo atendimento.
2. O sistema lista os pacientes cadastrados para que o usuário selecione o paciente que recebeu a nova consulta.
3. O sistema altera as informações contidas no banco de dados.
7. O sistema exibe os atendimentos realizados prestados aquele paciente.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.4 [RF 004] Quantidade de Atendimento

Descrição do caso de uso: O sistema deve mostrar a quantidade de atendimento realizados

Ator: Secretário

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

Sempre que o usuário realizar um novo atendimento, a quantidade deve mudar.

Saídas e pós-condição:

O sistema retorna o número de atendimentos realizados.

Fluxo de eventos principal

1. Usuário escolhe a ação visualizar atendimento.

Fluxos secundários

2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.5 [RF 005] Histórico de Atendimento

Descrição do caso de uso: O sistema deve mostrar a quantidade de atendimento prestado para aquele paciente.

Ator: Secretário

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

Sempre que o usuário realizar um novo atendimento, o sistema armazena no cadastro do paciente.

Saídas e pós-condição:

O sistema retorna o histórico de atendimento.

Fluxo de eventos principal

1. Usuário escolhe a ação visualizar paciente.

Fluxos secundários

2. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.6 [RF005] Bairros mais Atendidos

Descrição do caso de uso: O sistema deve mostrar os bairros que possuem mais atendimentos.

Ator: Secretário

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário entra com os dados do paciente a cada novo atendimento no formulário de atendimento.

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação.

Fluxo de eventos principal .

2. O sistema oferece a aba que mostra os bairros mais atendidos.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

3.1.7 [RF005] Pesquisar Paciente

Descrição do caso de uso: O sistema deve possuir a barra de pesquisa pelo nome do paciente.

Ator: Secretário

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições:

O usuário entra com o nome do paciente antes de registrar um como novo paciente

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação.

Fluxo de eventos principal .

2. O sistema oferece a aba que mostra a aba de pesquisa.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4. Requisitos não-funcionais

4.1 Usabilidade

Esta seção descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário e *help on-line*.

4.1.1 [NF001] Interface Amigável

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.1.2 [NF002] Componentes WEB

A interface deverá utilizar elementos comuns a usuários de sistemas web, como campos de texto, *combo-boxes*, *links* e botões, sem muito rebuscamento. A idéia é

focar nos aspectos operacionais sem se preocupar tanto com a beleza da tela, de modo a facilitar o uso por usuários iniciantes.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.2 Software

Esta seção descreve os requisitos não-funcionais associados aos softwares que devem ser utilizados para o desenvolvimento do sistema.

4.2.1 [NF003] Banco de Dados *adicionar o nome do banco de dados*

O sistema deve utilizar um banco de dados *adicionar banco de dados* para fazer o armazenamento de dados.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.2.2 [NF004] Linguagem Python

Visando criar um produto com maior extensibilidade, reusabilidade e flexibilidade, deve-se adotar Python como linguagem principal de desenvolvimento, seguindo cuidadosamente as técnicas de orientação a objetos.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.2.3 [NF004] Framework Django

Com o objetivo de se ter uma interface web de fácil desenvolvimento, o framework Django foi adotado como o principal framework do projeto.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.2.4 [NF005] Django Rest Framework

Visando criar um produto com maior extensibilidade, reusabilidade e flexibilidade, deve-se adotar Python como linguagem principal de desenvolvimento, seguindo cuidadosamente as técnicas de orientação a objetos.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.2.5 [NF006] Vue.js

Vue se destaca por sua simplicidade e flexibilidade, oferecendo ferramentas modernas como data binding reativo e DOM virtual, enquanto mantém uma curva de aprendizado acessível. Ele adota uma abordagem declarativa para renderizar dados no DOM, permitindo que desenvolvedores foquem na lógica da aplicação, sem a necessidade de manipular diretamente o DOM.

Prioridade: ■ Essencial □ Importante □ Desejável

Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

4.3 Desempenho

Esta seção descreve os requisitos não-funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

4.3.1 [NF005] Agilidade na Execução das Operações

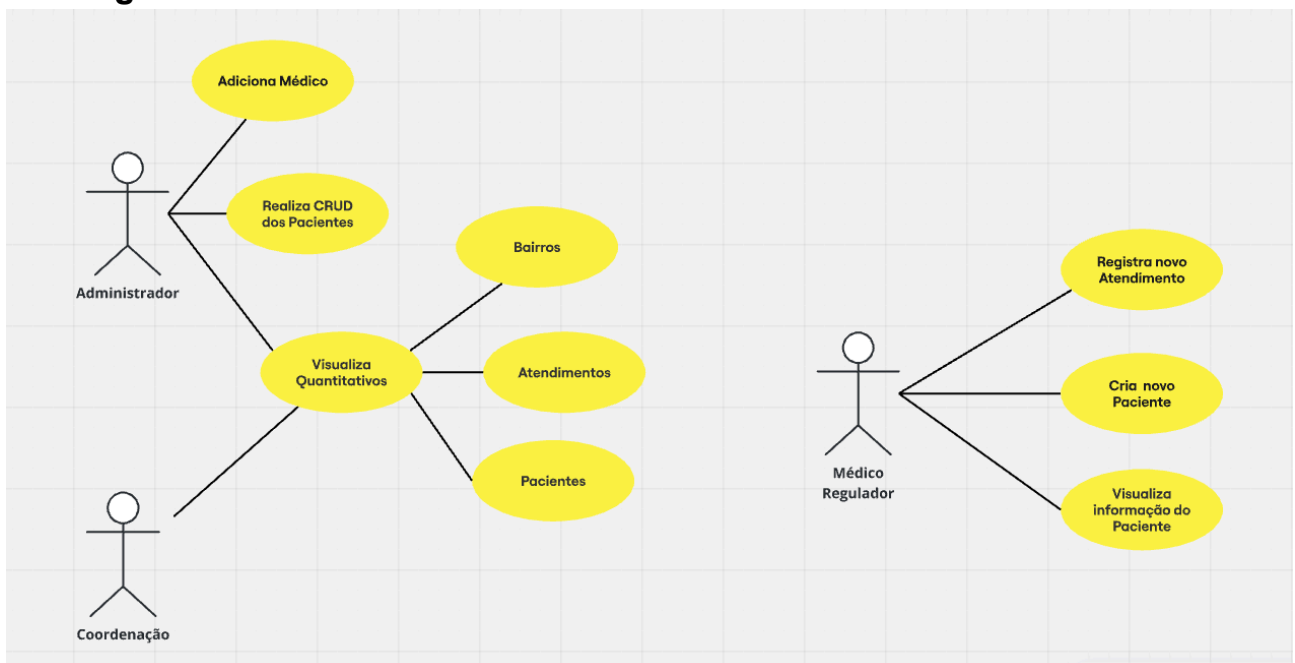
O sistema deve executar as operações no menor tempo possível, visando dar uma maior agilidade ao processo.

Prioridade: □ Essencial ■ Importante □ Desejável

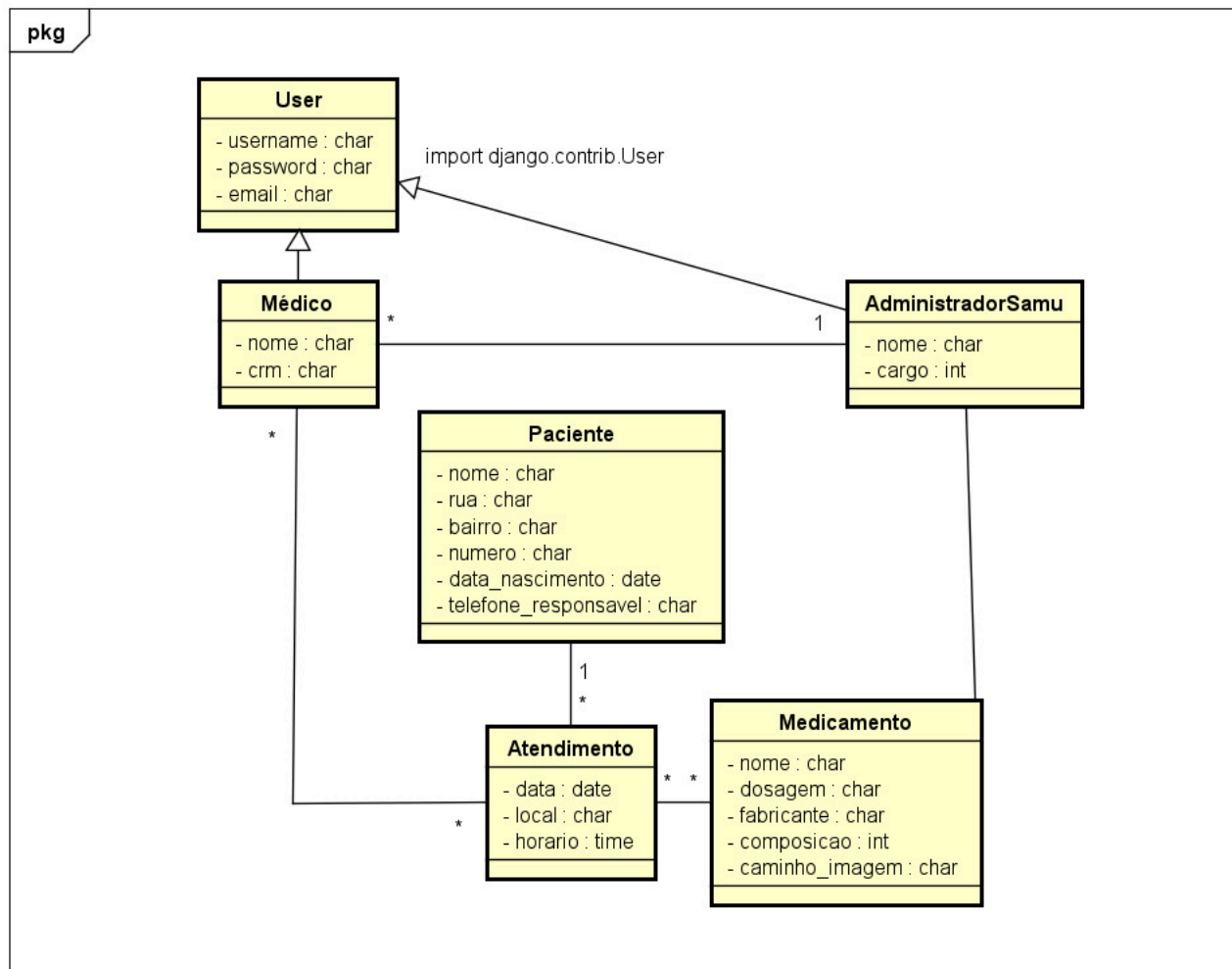
Casos de uso associados: [RF001], [RF002], [RF003], [RF004], [RF005], [RF006], [RF007], [RF008], [RF009], [RF010], [RF011], [RF012], [RF013], [RF014], [RF015], [RF016], [RF017], [RF018].

5. Diagramas

5.1 Diagrama de Caso de Uso



5.2 Diagrama de Classe



6. Referências

Rumbaugh, J., “UML – Guia do Usuário”. Editora Campus.

Proposta Técnica - *Saúde Integrada*) – Versão 2.0

Django: DJANGO Software Foundation. *Django documentation*. Disponível em: <https://www.djangoproject.com/>. Acesso em: 6 nov. 2024.

Django REST Framework: VAGUE, Tom Christie. *Django REST Framework documentation*. Disponível em: <https://www.django-rest-framework.org/>. Acesso em: 6 nov. 2024.

KYRIAKIDIS, A.; MANIATIS, K.; YOU, E. **The Majesty of Vue.js** 2. Leanpub, 2017.