



Instituto Politécnico do Cávado e do Ave

Escola Superior Tecnologia

Engenharia de Software, Gestão de Sistemas de Informação, Integração de Sistemas Clínicos

MediForm

Docentes: Margarida Portela, Patrícia Leite, João Pedro Silva

Trabalho efetuado por:

21193 Mariana Sá

21207 Gonçalo Marques

21773 Adneiza Pontes

24211 Kiankueno Edgar Álvaro Covi

2024/2025

Licenciatura em Engenharia Informática Médica

Índice

1. Introdução.....	7
1.1 Enquadramento e Motivação	8
1.2 Estrutura do documento.....	8
2. Metodologia de trabalho	10
2.1 Ferramentas Utilizadas	10
2.2 Arquitetura funcional.....	10
2.2.1 Arquitetura API.....	11
2.3 Organização e Comunicação.....	13
2.4 Desenvolvimento e Controlo de Código	14
3. Instigadores do projeto	15
3.1 Objetivos do sistema	15
3.2 Utilizadores do sistema.....	15
3.3 Desafios a enfrentar.....	16
4. Viabilidade do projeto	17
4.1 Proposta financeira.....	17
4.1.1 Taxa de retorno e imposto	17
4.1.2 Investimento inicial	17
4.1.3 Custos e Benefícios.....	17
4.1.4 Valores Totais	18
4.1.5 Cálculo do <i>Payback</i>	18
4.1.6 Cálculo do VAL.....	19
4.2 Análise SWOT.....	20
5. Análise de Requisitos	21
3.1 Requisitos Funcionais (RF)	21
3.2 Requisitos Não Funcionais (RNF)	23
6. Diagramas	24
6.1 Diagrama de Contexto	24
6.2 Diagrama Casos de Uso.....	25
6.2.1 Gestão de Pedidos.....	26
6.3 Diagrama de Atividades	27
6.4 Diagrama ER.....	29
7. Web Services – Fornecedor.....	30
7.1 Tabela de métodos utilizados na API.....	31

8.	Testes.....	36
9.	Mock-ups.....	40
9.1	Login.....	40
9.2	Registo de uma nova conta	40
9.3	Dashboard.....	41
10.	Conclusão	42
11.	Bibliografia	43

Índice das figuras

Figura 1- Arquitetura do projeto	11
Figura 2- Componentes MVC.....	11
Figura 3- Comunicação API e Base de dados.....	12
Figura 4- Comunicação API e Base de dados.....	12
Figura 5- Análise SWOT.....	20
Figura 6- Diagrama de contexto.....	24
Figura 7- Diagrama de contexto.....	24
Figura 8- Diagrama Casos de Uso MediForm	25
Figura 9- Gestão de Pedidos.....	26
Figura 10- Diagrama de Atividades	27
Figura 11- Pedido Medicamento	28
Figura 12- Pedido Medicamento	28
Figura 13- Diagrama de entidade-relação.....	29
Figura 14- API Fornecedor (WebService)	30
Figura 15- Processar o pedido	32
Figura 16- Obter Pedido	32
Figura 17- Deletar Pedido	33
Figura 18- Resposta Pedido	33
Figura 19- Pedido negado.....	34
Figura 20- pedido aprovado.....	34
Figura 21- Erro pedido.....	35
Figura 22- Pedido efetuado MediForm	35
Figura 23- Teste pedido	36
Figura 24- Teste pedido	36
Figura 25- Resposta ao pedido	37
Figura 26- Teste resposta	38
Figura 27- Pedido com Id invalido	38
Figura 28- Teste Pedido Invalido	39

Índice das Tabelas

Tabela 1- Taxas que o projeto apresenta.....	17
Tabela 2- Investimentos iniciais	17
Tabela 3- Custos	18
Tabela 4- Benefícios	18
Tabela 5- Valores Totais	18
Tabela 6- Requisitos Funcionais	22
Tabela 7- Requisitos Não Funcionais	23
Tabela 8- Lista de métodos acessíveis na API.....	31

Lista de siglas e acrónimos

RF Requisito Funcional

RNF Requisito Não Funcional

ER Entidade-Relação

SWOT *Strength, Weaknesses, Oportunities, Threats*

INFARMED Autoridade Nacional do Medicamento e Produtos de Saúde, IP

1. Introdução

O acesso a medicamentos de boa qualidade é fundamental para garantir tratamentos adequados e bons resultados na saúde.

Nos hospitais, a gestão de medicamentos, tanto manipulados quanto gerais, deve ser bem organizada para garantir que tratamentos e que os recursos necessários estejam sempre disponíveis. Para isso, é necessário o uso de soluções tecnológicas que melhorem os processos, integrem as informações e otimizem tarefas.

Os medicamentos são ferramentas terapêuticas importantes no processo de saúde e doença, contribuindo para o aumento da expectativa e qualidade de vida da população (Costa *et al.*, 2011). Já os medicamentos manipulados são qualquer preparado oficial ou fórmula magistral preparado e dispensado sob a responsabilidade de um farmacêutico. (INFARMED, 2024).

As farmácias hospitalares recebem, guardam e distribuem medicamentos, mas o sistema atual, como mencionado no enunciado partilhado tem problemas, como a falta de integração entre os setores e a dificuldade de controlar o consumo e prever reposições. Além disso, muitos processos ainda são feitos manualmente, o que pode causar erros. MediForm foi criado para resolver essas questões, oferecendo uma solução digital que organiza e melhora a gestão do stock de medicamentos manipulados e os não manipulados.

1.1 Enquadramento e Motivação

O sistema MediForm foi desenvolvido para atender a essas necessidades, oferecendo uma solução completa que centraliza a informação do *stock*, digitaliza as requisições entre os setores e melhora os processos de reposição.

Com a MediForm, as farmácias hospitalares têm a capacidade de:

- Registrar o stock dos medicamentos
- Acompanhar os consumos de cada setor
- Automatizar o envio de requisições
- Facilitar a gestão de medicamentos, desde a requisição até à entrega do medicamento

Este projeto foi inspirado pelos desafios apresentados no enunciado partilhado, como a falta da integração das informações, a comunicação manual entre os setores associados e as farmácias, e a dificuldade de prever a quantidade de medicamentos necessária.

A MediForm é a resposta a esses problemas, oferecendo uma solução que melhora a gestão do hospital e ajuda a reduzir os riscos de falta de medicamentos.

1.2 Estrutura do documento

O presente documento está dividido em mais de 10 partes:

- A primeira parte, descreve sucintamente o que é pressuposto fazer para a gestão de stock para farmácia hospitalar dos medicamentos.
- Na segunda parte é apresentada a metodologia de trabalho, bem como ferramentas utilizadas, arquitetura funcional, organização e comunicação e desenvolvimento e controlo de código.
- Na terceira parte são os instigadores do projeto, os objetivos e a descrição das partes interessadas e os desafios a enfrentar.
- Na quarta parte, é mostrada a viabilidade do projeto.
- Na quinta parte, é apresentada a análise dos requisitos.
- Na sexta parte são apresentados os diagramas UML (*Unified Modeling Language*) desenvolvidos tendo em conta as especificações do sistema de gestão de stock para farmácia hospitalar dos medicamentos.
- Na sétima parte, Web services sobre os fornecedores.
- Na oitava parte, são os testes efetuados no Postman.
- Na nona parte, são apresentadas os *mock-ups* relativos ao que é pressuposto implementar no sistema.

- Por fim, é relatado as conclusões retiradas com o desenvolvimento deste projeto.

2. Metodologia de trabalho

2.1 Ferramentas Utilizadas

Para o desenvolvimento deste projeto, utilizamos ferramentas para que a programação seja mais organizada. Estas incluem:

Visual Studio: Utilizado para o desenvolvimento e a criação da aplicação, seguindo a arquitetura MVC(*Model-View-Controller*).

GitHub: Utilizado para controlo da versão mais recente e colaboração, permitindo guardar código de forma segura, acompanhar alterações e trabalhar em equipa de forma mais eficiente.

Postman: Esta ferramenta foi utilizada para testar os *web services*, verificando a funcionalidade e integração dos serviços.

Visual Paradigm: Utilizada para modelar e criar os diagramas.

2.2 Arquitetura funcional

Utilizamos a arquitetura MVC(*Model-View_Controller*) para uma organização eficiente código. Esta arquitetura divide o sistema em três partes principais:

- **Model:** Representa os dados e as interações com a base de dados, usando *Entity Framework* para facilitar a manipulação dos dados.
- **View:** A apresentação, é onde as páginas são geradas, com HTML dinâmico e integração de dados com o *Model*.
- **Controller:** Esta parte é intermediária, onde recebe as solicitações do utilizador, comunica com o *Model* para obter os dados necessários e retorna as *views* para o utilizador.

O diagrama mostra a interação entre estas camadas:

- A base de dados comunica com o *Model*, que é onde os dados são manipulados.
- O *Controller* recebe as ações do utilizador, consulta o *Model* e, depois, apresenta os resultados na *View*.
- O *Front End* (HTML, CSS, etc.) é o que o utilizador vê e interage, e é aqui que os dados são apresentados.

Ao usar o MVC, conseguimos organizar o código de forma modular, o que facilita tanto o teste de cada parte como a sua manutenção. Essa divisão permite que

alterações numa camada não afetem as outras camadas, tornando o sistema mais robusto e fácil de gerir.

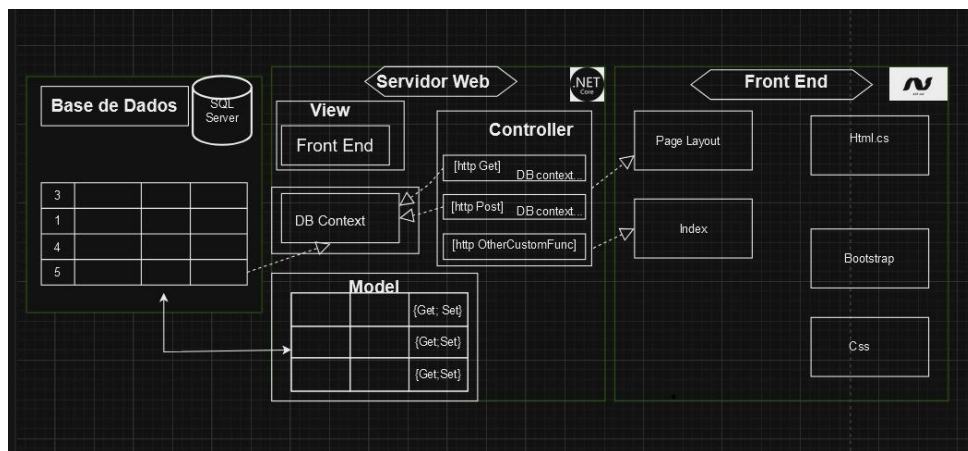


Figura 1- Arquitetura do projeto

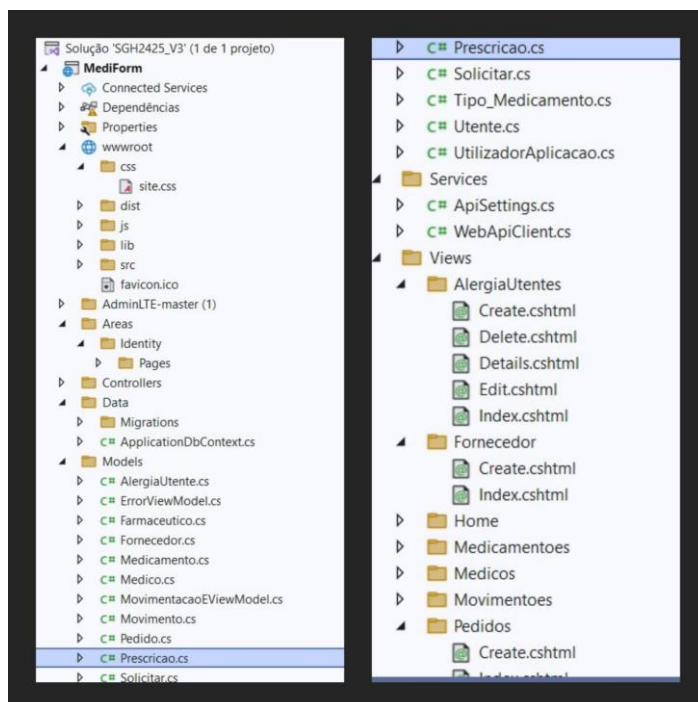


Figura 2- Componentes MVC

2.2.1 Arquitetura API

Na arquitetura da nossa web API, utilizamos o *Postman* como ferramenta principal para testar e simular as requisições HTTP feitas pelo cliente. A

comunicação entre o cliente e o servidor é feita através deste protocolo, sendo a API a responsável por fazer a intermediação entre o cliente e a base de dados.

Quando o cliente envia uma requisição, a API processa os dados e, se necessário, faz consultas ou alterações na base de dados. A API valida e manipula os dados, respondendo de volta ao cliente com as informações solicitadas. Para adicionar ou atualizar dados, a API comunica diretamente com a base de dados, registrando as alterações conforme a requisição.

Utilizamos o Postman para simular o comportamento do cliente, enviando requisições para a API e validando as respostas. Isso permitiu garantir que a API funcionasse corretamente, sem erros na comunicação ou manipulação de dados.

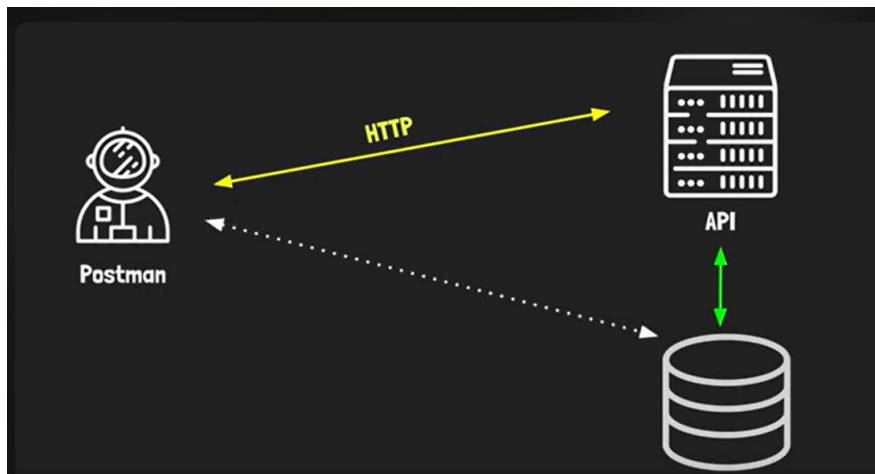


Figura 3- Comunicação API e Base de dados

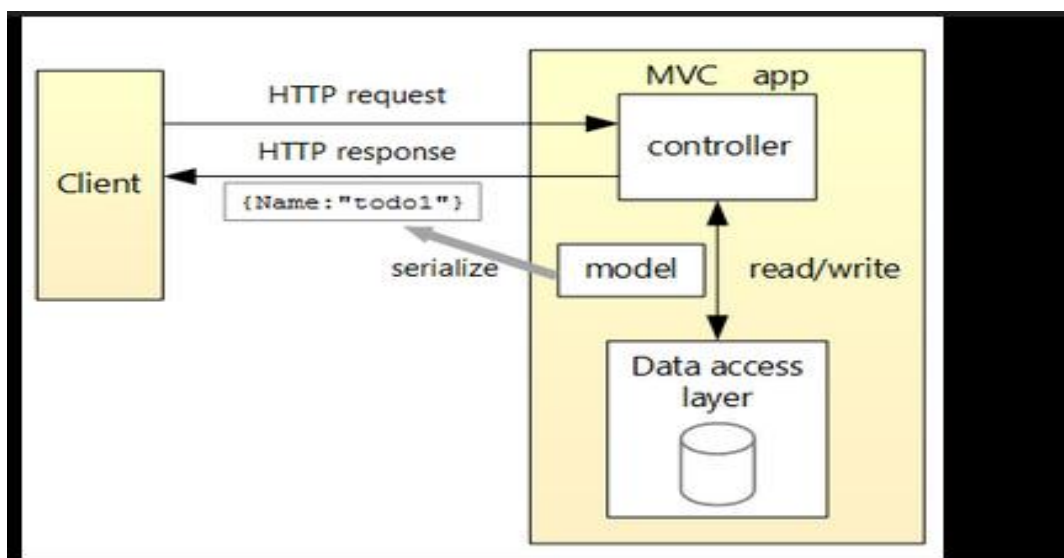


Figura 4- Comunicação API e Base de dados

2.3 Organização e Comunicação

A comunicação no projeto foi estruturada para garantir clareza, eficiência e alinhamento entre todos os membros da equipa. Além disso, utilizamos um cronograma para acompanhar as atividades e manter os prazos definidos.

2.3.1 Canais de comunicação

Discord: Foi a principal ferramenta para a comunicação, utilizado para reuniões e discussões da equipa.

WhatsApp: Utilizado também para comunicação entre a equipa.

2.3.2 Gestão de projetos

Cronograma: Criado um cronograma detalhado no Microsoft Excel, dividido em semanas e com marcos claros para cada etapa do projeto.

- O cronograma incluía prazos para entregas, revisão de código, testes e reuniões importantes.
- Atualizações no cronograma eram feitas semanalmente para refletir o progresso e ajustar possíveis desvios.

2.3.3 Estrutura das reuniões

- Reuniões assíncronas ocorreram principalmente no *Discord*, com mensagens trocadas pelo *WhatsApp* para discussão e tomadas de decisão.
- Check-ins semanais foram feitos através de mensagens estruturadas, onde cada membro atualizava o status das suas atividades e destacava desafios encontrados.
- Revisões de sprint e brainstormings também foram realizados por meio de discussões, com o apoio de documentos compartilhados quando necessário.

2.4 Desenvolvimento e Controlo de Código

O controlo do código foi essencial para assegurar a organização e a qualidade do desenvolvimento do projeto. Para tal, foi utilizada a ferramenta GitHub como a principal ferramenta de controlo de versões.

2.4.1 Estrutura de branches

O código foi mantido em branches separados, em que cada branch representava uma nova funcionalidade ou correção específica. Esta prática facilitou o trabalho paralelo dos membros da equipa e reduziu o risco de conflitos.

2.4.2 Pull requests

Todas as alterações realizadas no código eram submetidas através de pull requests, garantindo as mudanças fossem revistas por outros membros da equipa antes de serem integradas na branch principal.

Esta abordagem contribuiu para manter a qualidade do código.

2.4.3 Gestão de repositórios

Foi utilizado um único repositório, com permissões adequadas para garantir segurança e controlos de acessos.

3. Instigadores do projeto

Durante este capítulo serão especificados os instigadores do projeto, ou seja, o que motiva a realização deste projeto assim como os objetivos, a identificação dos intervenientes e dos utilizadores do sistema.

MediForm tem como finalidade a gestão de stock dos medicamentos em unidades hospitalares, com o intuito de melhorar a organização e a eficiência dos processos relacionados a estes medicamentos.

3.1 Objetivos do sistema

A criação de um sistema para gestão de *stock* dos medicamentos pressupõe melhorar a eficiência de controlo de *stock* do medicamento. Este sistema permitirá identificar as entradas e levantamento dos medicamentos, quais estão em falta e o reabastecimento do mesmo.

Com isso, será reduzido os erros, evitar desperdícios e assegurar que o medicamento esteja sempre disponível quando for necessário.

Proporcionará também uma gestão mais simples, ajudando a farmácia hospitalar a funcionar de forma mais organizada e confiável.

3.2 Utilizadores do sistema

Os utilizadores do sistema serão:

- Farmacêutico;
- Médico;
- Funcionário de logística;
- Utente.

Cada tipo de utilizador tem permissões específicas no sistema:

O **Funcionário de Logística** tem a permissão de:

- Monitorizar os relatórios sobre o estado do *stock*;
- Requisitar os medicamentos em falta;
- Registar os medicamentos;
- Listar histórico de medicamentos;
- Solicita o reabastecimento do *stock*;
- Atualizar o *stock*.

O **Farmacêutico** pode:

- Consultar medicamento manipulado prescrito;
- Atualizar o *stock*;
- Solicitar medicamentos;
- Notifica o reabastecimento;

O **médico** tem a permissão de:

- Prescrever os medicamentos.

3.3 Desafios a enfrentar

Os problemas atuais que estão a ser enfrentados são os seguintes:

- A farmácia não comunica atualmente com o *stock* nem recebe as requisições digitalmente;
- A farmácia realiza todo o processo em papel e comunicam por telefone;
- Incapacidade de se preverem *stocks* ou insumos de referência por materiais ou fármacos;
- O consumo neste setor não está inserido digitalmente no sistema.

4. Viabilidade do projeto

O estudo de viabilidade avalia a importância do produto que se visa a desenvolver e também se a relação esforço total pelo tempo sejam justificados no decorrer da criação do projeto, assim como os investimentos monetários necessários para que o projeto se desenvolva.

4.1 Proposta financeira

Neste tópico, aborda-se uma proposta financeira do projeto, isto é, a implementação da solução para o problema em questão.

4.1.1 Taxa de retorno e imposto

A Tabela 1 apresenta as taxas de retorno e de imposto do projeto:

Tabela 1- Taxas que o projeto apresenta

Projeto	Taxa
<i>Taxa de retorno necessária</i>	15%
<i>Taxa de imposto</i>	23%

4.1.2 Investimento inicial

A Tabela 2 mostra valores dos investimentos iniciais para o início do projeto:

Tabela 2- Investimentos iniciais

Investimento inicial	Ano 1
<i>Desenvolvimento do sistema</i>	38 000€
<i>Taxas e Licenças</i>	25 000€
<i>Formação da equipa</i>	7 000€
<i>Inspeção técnica</i>	5 000€
<i>Total</i>	75 000€

4.1.3 Custos e Benefícios

A Tabela 3 mostra os custos ao longo dos anos para melhorar o projeto, isto para evitar falhas no sistema que possam trazer prejuízos:

Tabela 3- Custos

Custos	Ano 1	Ano 2	Ano 3	Total
<i>Manutenção</i>	500€	2 000€	5 000€	7 500€
<i>Testes de qualidade</i>	1 500 €	1 500€	1 500€	4 500€
<i>Amortização de gastos</i>	5 000€	2 000€	1 000€	8 000€
<i>Custos totais</i>	7 000€	5 500€	7 500€	20 000€

A Tabela 4 mostra os benefícios que a aplicação poderá trazer e os respetivos valores:

Tabela 4- Benefícios

Benefícios	Ano 1	Ano 2	Ano 3	Total
<i>Redução de equipas de trabalho</i>	19 200€	19 200€	19 200€	57 600€
<i>Redução de desperdícios(tempo)</i>	3 000€	3 000€	3 000€	9 000€
<i>Redução de custos operacionais(caminho/organização)</i>	920€	920€	920€	2 760€
<i>Benefícios totais</i>	23 120€	23 120€	23 120€	69 360€

4.1.4 Valores Totais

A Tabela 5 mostra os valores totais de cada ano, referentes aos benefícios líquidos, ao imposto e fluxos de caixa:

Tabela 5- Valores Totais

Totais	Ano 0	Ano 1	Ano 2	Ano 3
<i>Benefícios líquidos</i>		16 120€	17 620€	15 620€
<i>Imposto</i>		3 710€	4 055€	3 600€
<i>Valor após Imposto</i>		12 410€	13 565€	12 020€
<i>Retorno de amortização</i>		5 000€	2 000€	1 000€
<i>Fluxo de caixa</i>	-75 000€	17 410€	15 565€	13 020€
<i>Fluxo de caixa acumulado</i>	-75 000€	-57 590€	-42 025€	-29 005€

4.1.5 Cálculo do Payback

O critério do Período de Recuperação do Investimento, ou *Payback*, mede o período que o somatório dos *cash-flows* leva a igualar (recuperar) o investimento inicial. É preferido o projeto que tenha o período de recuperação mais curto.

- $PR_1 = 1 \text{ ano}$
- $PR_2 = 1 \text{ ano} + \frac{75\,000\text{€} - 17\,410\text{€}}{15\,565\text{€}} * 12 = 1 \text{ ano} + 44,4 \text{ meses (3 anos e 7 meses)}$
- $PR_3 = 1 \text{ ano} + \frac{75\,000\text{€} - 15\,565\text{€}}{13\,020\text{€}} * 12 = 1 \text{ ano} + 54,8 \text{ meses (4 anos e 7 meses)}$

4.1.6 Cálculo do VAL

$$VAL = -C_0 + \frac{C_n}{(1+i)^n}$$

- C_0 – Investimento Inicial;
- C_n – Fluxo de Caixa de cada período;
- n – Número de anos do projeto;
- i – Taxa de retorno necessária.

$$VAL = -75\,000\text{€} + \frac{17\,410\text{€}}{(1+0.15)^1} + \frac{15\,565\text{€}}{(1+0.15)^2} + \frac{13\,020\text{€}}{(1+0.15)^3} = -39\,530.6\text{€}$$

Com o VAL negativo, podemos concluir que o projeto não é viável ou não será lucrativo a longo prazo, pois o retorno descontado (considerando a taxa de 15%) não compensa o investimento inicial de -75 000€, contudo se este investimento, por exemplo, for implementado em 10 hospitais, o VAL seria positivo.

4.2 Análise SWOT

Com essa finalidade, realiza-se uma análise SWOT – *Strength, Weaknesses, Opportunities, Threats*, sendo uma técnica de planeamento estratégico que coloca o negócio em perspetiva usando as seguintes características: Forças, Fraquezas, Oportunidades e Ameaças.

A análise SWOT para este projeto encontra-se na Figura 6:



Figura 5- Análise SWOT

Relativamente às **Forças**, é inegável que dizer quanto mais inovadora e automatizada for a tecnologia implementada na gestão de stock de medicamentos, maior será a capacidade de integração de dados hospitalares sobre os medicamentos e consequentemente, os utentes. Com isso permite que os médicos e os farmacêuticos acedam às informações atualizadas sobre os medicamentos, para que se evite a duplicação das mesmas e otimização de recursos. Garante reduções de erros, logo existirá uma maior precisão na gestão de stock dos medicamentos. Assim, permitindo um acesso rápido, facilitado e organizado.

Relativamente às **Fraquezas**, ao inovar novas tecnologias reduzirá o número de mão obra, logo a equipa será mais reduzida do que pressuposto. Além disso, o custo de implementação elevada, o que acontece em todos os projetos, pode ser um entrave para a adoção imediata da nova tecnologia, dificultando a quem possui orçamentos limitados. A ausência de infraestruturas adequadas, como redes de dados estáveis, também representa um desafio, impedindo que a integração seja viável. Os custos de manutenção prolongados demandam um investimento contínuo, o que pode comprometer o orçamento a longo prazo. Por último, se existir falta de interoperabilidade entre diferentes sistemas, vai impedir a troca de informação, o que dificulta a gestão integrada dos medicamentos e também com os fornecedores.

Relativamente às **Oportunidades**, com o crescimento da integração tecnológica poderá ser possível unificar diferentes farmácias hospitalares, promovendo o acesso em tempo real na gestão de stock dos medicamentos. A interoperabilidade representa uma grande oportunidade na gestão de stock e previsão de reposição. A redução de desperdícios, auxilia e otimiza na gestão do tempo em que os medicamentos são procurados.

Relativamente às **Ameaças**, a implementação de novas tecnologias pode enfrentar resistência tanto por parte dos profissionais de saúde quanto da administração hospitalar. Essa relutância pode atrasar a adoção completa do sistema, limitando os benefícios esperados. O mercado no tema em questão é competitivo, várias empresas podem oferecer projetos similares, dificultando a venda deste novo produto. Mudanças nas regulamentações podem exigir ajustes contínuos no sistema para garantir conformidade legal, o que pode aumentar os custos operacionais e o tempo necessário para adaptação.

5. Análise de Requisitos

Os requisitos são uma parte da documentação que descreve as funcionalidades e características de um sistema. A documentação deve incluir um propósito, uma descrição geral e uma especificação de requisitos. Para contextualizar para requisitos funcionais pode ser utilizado diagrama de casos de uso. Os requisitos não funcionais podem apresentar-se com classificação pré-definida, como requisitos de qualidade. Ambos clarificam a visão do projeto, como a solução vai operar e os utilizadores vão interagir.

3.1 Requisitos Funcionais (RF)

Os requisitos funcionais – ou RF – descrevem o comportamento do sistema.

Tabela 6- Requisitos Funcionais

RF 1	Monitorizar relatórios de <i>stock</i>
RF 2	Requisitar medicamentos
RF 3	Registar medicamentos
RF 4	Listar histórico de produtos
RF 5	Gerar relatórios de <i>stock</i>
RF 6	Consultar disponibilidade de medicamentos
RF 7	Atribuir medicamentos prescritos
RF 8	Comunicar a baixa do stock
RF 9	Prescrever medicamentos
RF 10	Consultar o estado do medicamento
RF 11	Solicitar o reabastecimento dos medicamentos

De uma maneira mais aprofundada, apresentamos as seguintes descrições:

Requisitos funcionais do “Funcionário de Logística” são:

1. Monitorizar relatórios de *stock*

- 1.1 Consultar relatórios sobre o estado do *stock*;
- 1.2 Atualizar *stock*.

2. Requisitar medicamentos

- 2.1 Fazer pedidos de reabastecimento de medicamentos.

3. Registar medicamentos

- 3.1 Inserir novos medicamentos no sistema.

4. Listar histórico de produtos

- 4.1 Aceder ao histórico dos produtos consumidos.

5. Gerar relatórios de *stock*

Requisitos funcionais do “Farmacêutico” são:

6. Consultar disponibilidade dos medicamentos

- 6.1 Aceder à informação sobre os medicamentos disponíveis.

7. Atribuir medicamentos prescritos

- 7.1 Registar no sistema os medicamentos que foram prescritos ao utente.

8. Comunicar a baixa do stock

- 8.1 Informar que existe falta de medicamentos no sistema.

Requisitos funcionais do “Médico” são:

9. Prescrever os medicamentos

- 9.1 Emitir a prescrição do medicamento;
- 9.2 Alterar a prescrição consoante a condição do utente;
- 9.3 Anular o medicamento prescrito.
- 9.4 Validar Prescrição.

Requisitos funcionais do “Utente” são:

10. Consultar o estado do medicamento

- 10.1 Visualizar informações de disponibilidade.

11. Solicitar o reabastecimento dos medicamentos

- 11.1 Pedido de medicamentos em falta ou prestes a acabar.

3.2 Requisitos Não Funcionais (RNF)

Os requisitos não funcionais – também referido como RNF – irão descrever os atributos de qualidade do sistema, auxiliando na caracterização do software. A Tabela 7 representa requisitos não funcionais para a aplicação de Gestão de Stock para Farmácias Hospitalares:

Tabela 7- Requisitos Não Funcionais

RNF 1	O sistema deve ter uma interface intuitiva e responsiva
RNF 2	O estilo visual do sistema deve ser consistente em todas as páginas e elementos de interface do utilizador, mantendo a paleta de cores, fontes e ícones nas diferentes páginas
RNF 3	O sistema ser fácil de utilização e aprendizagem
RNF 4	O código da aplicação deve ser bem documentado e seguir boas práticas de desenvolvimento
RNF 5	O sistema deverá atender às normas legais aplicáveis
RNF 6	Os <i>downloads</i> dos ficheiros devem ser em formato <i>pdf</i>
RNF 7	O sistema deve manter uma estrutura básica para facilitar futuras atualizações.
RNF 8	A aplicação deve ser compatível com versões recentes dos navegadores.
RNF 9	O armazenamento dos dados pessoais deve obedecer às normas do RGPD (Regulamento Geral de Proteção de Dados)
RNF 10	O nome dos campos a inserir devem ser objetivos e de fácil interpretação

6. Diagramas

Diagrama é esquema gráfico que pretende representar, de forma simples, as relações entre os diferentes elementos de um sistema ou as variações de determinado fenómeno. (www.infopedia.pt).

A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. (lucidchart ,2024).

6.1 Diagrama de Contexto

Na figura 7, apresentamos o diagrama de contexto do sistema de gestão de stock de medicamentos. Este diagrama ilustra as interações entre os principais utilizadores, como o administrador, farmacêutico e funcionário de logística. Cada um desempenha um papel fundamental na gestão eficiente dos recursos.

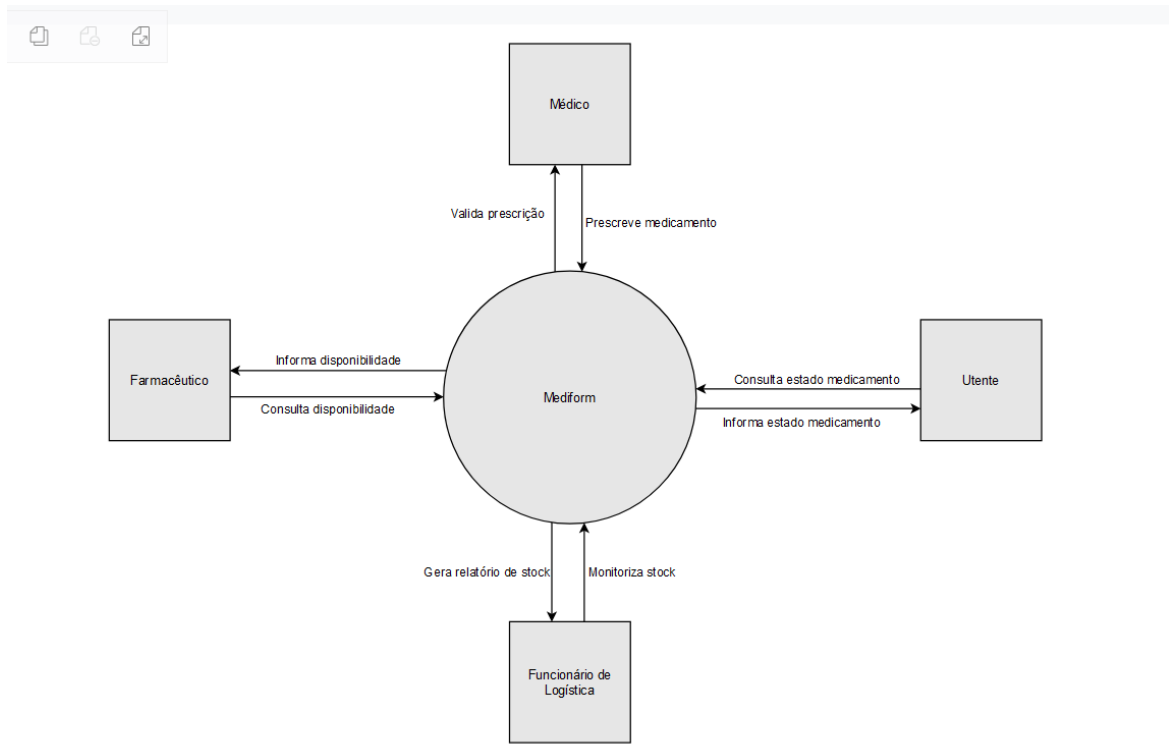


Figura 7- Diagrama de contexto

6.2 Diagrama Casos de Uso

Um diagrama de casos de uso é uma excelente ferramenta para especificar requisitos e captar estrutura global de funcionalidade.

Na Figura 8, é apresentado o diagrama de casos de uso para o sistema de Gestão de Stock para Farmácia Hospitalar de Medicamentos (MediForm). Neste diagrama, temos os principais atores e suas interações com o sistema.

O Funcionário de Logística tem a responsabilidade de gerir o stock de medicamentos. Ele pode listar o histórico, consultar o status dos medicamentos, consultar relatórios sobre o stock, registar stock e atualizar o stock conforme necessário. Além disso, ele também pode realizar processos como a reposição de medicamentos.

O Farmacêutico é responsável por verificar a disponibilidade dos medicamentos e atribuir medicamentos conforme as necessidades. Ele também tem a função de registar medicamentos no sistema.

O Médico, por sua vez, tem a função de emitir prescrições, atualizar prescrições, anular prescrições e validar prescrições, garantindo que os medicamentos prescritos sejam adequados e estejam corretamente registados.

Por fim, o Utente pode visualizar as prescrições feitas pelo médico e consultar o estado dos medicamentos, ou seja, saber se o medicamento foi entregue ou se está disponível.



Figura 8- Diagrama Casos de Uso MediForm

6.2.1 Gestão de Pedidos

Na figura 9 é mostrada como os atores interagem com o sistema na Gestão de Pedidos. O Fornecedor e a Farmácia Hospitalar (MediForm) são os principais responsáveis pelas operações.

O **Fornecedor** tem como funções principais:

- **Processar e responder a pedidos:** Ele aprova ou nega os pedidos com base na disponibilidade dos medicamentos.
- **Consultar a disponibilidade dos medicamentos:** Verifica se os medicamentos solicitados estão em *stock*.
- **Obter detalhes dos pedidos:** Pode consultar o status dos pedidos em andamento.

A **Farmácia Hospitalar (MediForm)**, por sua vez, é responsável por:

- **Emitir os pedidos:** Solicita a reposição dos medicamentos quando necessário.

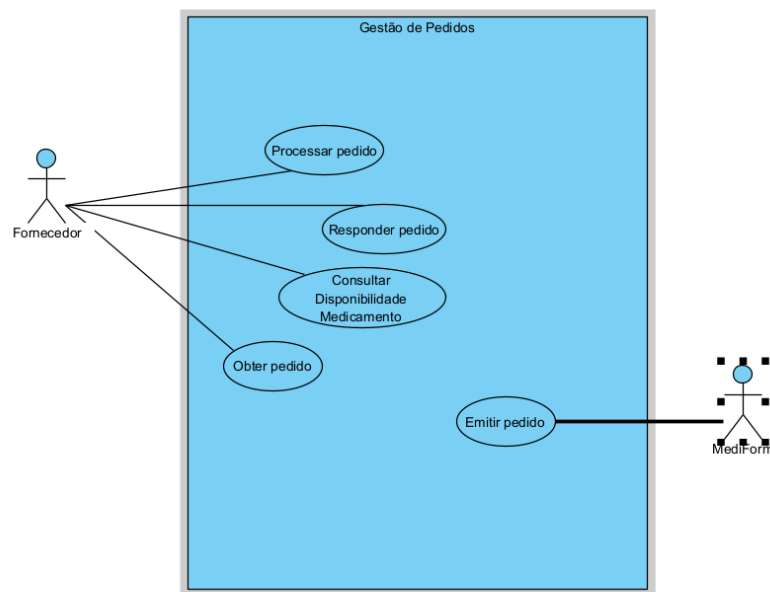


Figura 9- Gestão de Pedidos

6.3 Diagrama de Atividades

O diagrama de atividades descreve o fluxo de atividades que ocorrem no sistema. Na Figura 10 é demonstrado o diagrama de atividades relativo ao processo da requisição dos medicamentos.

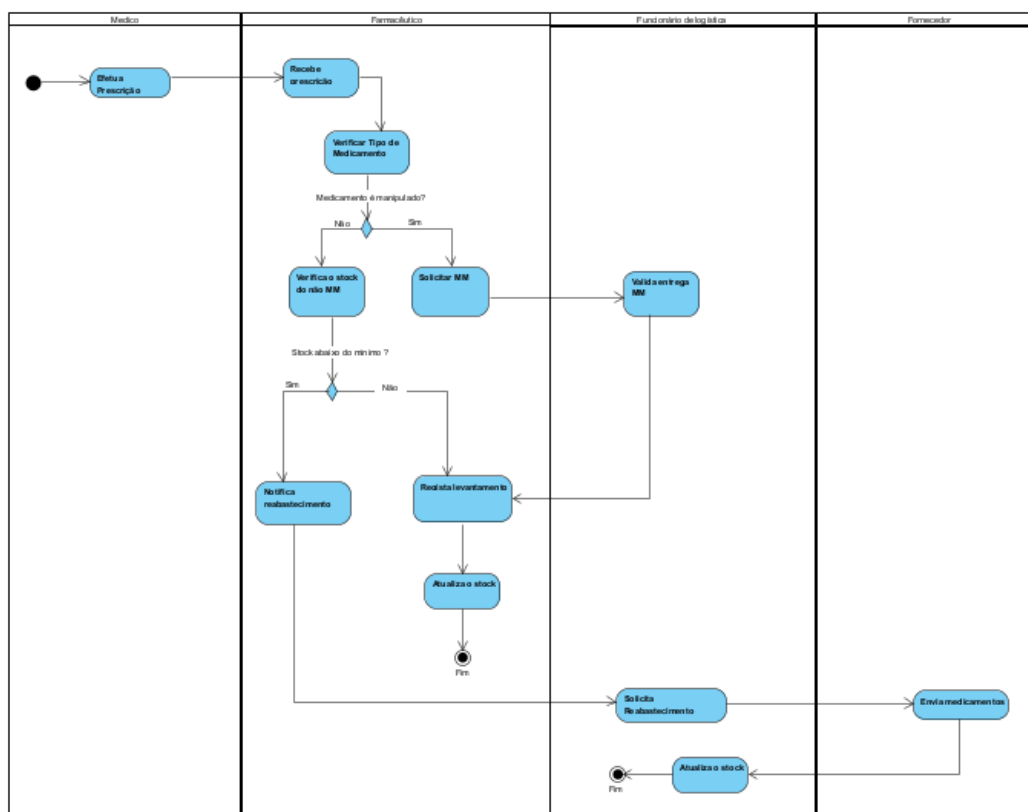


Figura 10- Diagrama de Atividades

Na figura 11, temos o diagrama de fluxo que descreve o processo de pedido de medicamentos no sistema MediForm. O Médico inicia o processo ao efetuar a prescrição de um medicamento, o que gera uma entrada no sistema para o Farmacêutico. Ao receber a prescrição, o Farmacêutico deve verificar o tipo de medicamento (se é manipulado ou não).

Se o medicamento não for manipulado, o Farmacêutico passa a verificar o stock do medicamento. Caso o stock esteja abaixo do mínimo, o Farmacêutico solicita o medicamento manipulado (caso não exista disponível no stock regular). Se o stock estiver suficiente, o processo avança com a validação da entrega do medicamento, garantindo que o item esteja disponível e pronto para ser entregue ao paciente.

Se o medicamento for manipulado, o Farmacêutico vai diretamente para o processo de solicitação de medicamentos manipulados, fazendo uma requisição

para garantir a disponibilidade do produto. Em seguida, o Funcionário de Logística realiza a validação da entrega para garantir que o processo de entrega ao paciente seja feito corretamente.

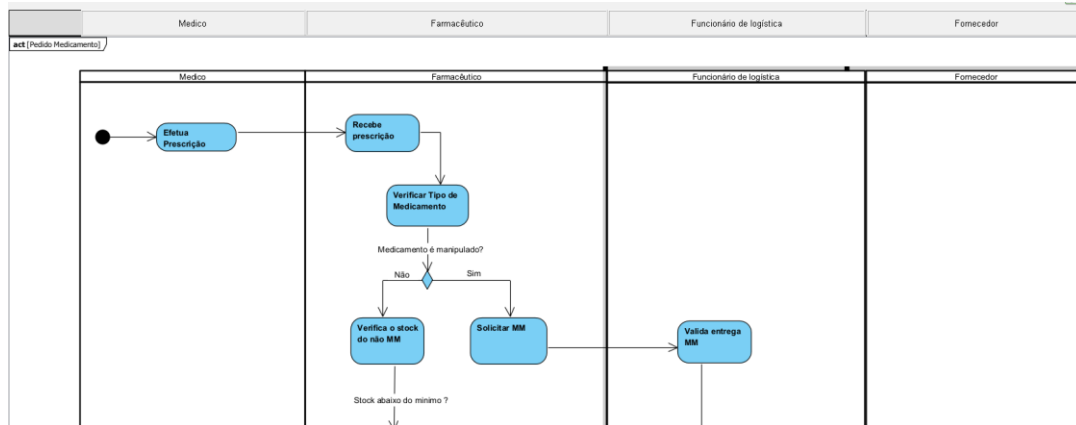


Figura 11- Pedido Medicamento

Na figura 12, o Farmacêutico realiza uma verificação de stock, e caso o stock esteja abaixo do necessário, a primeira ação será notificar o reabastecimento. Caso o stock esteja dentro dos limites, o processo avança com a atualização do stock.

Se for necessário reabastecer, o Farmacêutico registra o levantamento do medicamento e solicita o reabastecimento ao Fornecedor. O Fornecedor envia os medicamentos e, uma vez recebidos, o Farmacêutico atualiza o stock para refletir as novas quantidades disponíveis.

Este processo garante que o stock de medicamentos seja sempre mantido atualizado e disponível para a realização de tratamentos hospitalares, prevenindo faltas de medicamentos e mantendo a continuidade dos cuidados.

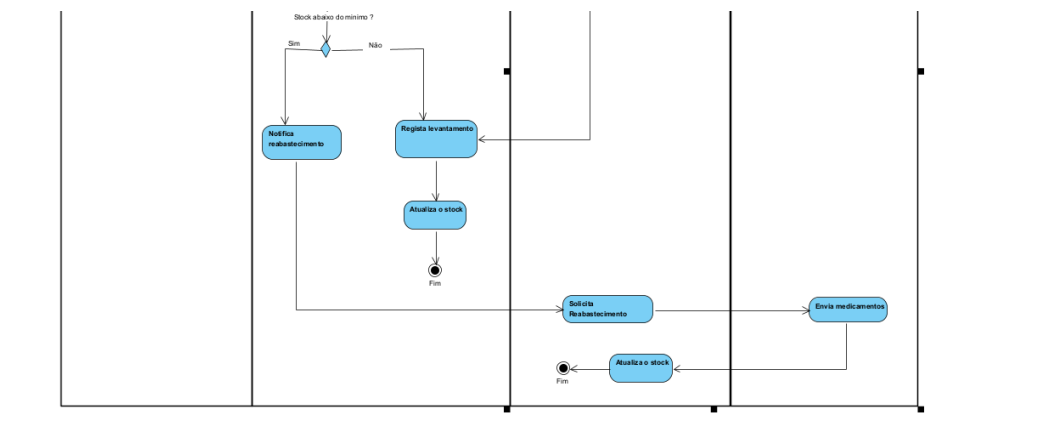


Figura 12- Pedido Medicamento

6.4 Diagrama ER

O diagrama de Entidade-Relação (ER) é uma representação visual que mostra as entidades de um sistema e as suas interações. Cada entidade é caracterizada por atributos, e as relações entre elas são representadas por linhas. Na figura 13, está representado o diagrama ER do sistema de gestão de stock farmacêutico hospitalar, ilustrando as principais entidades, como Utente, Médico, Farmacêutico e Medicamento, e como estas se relacionam.

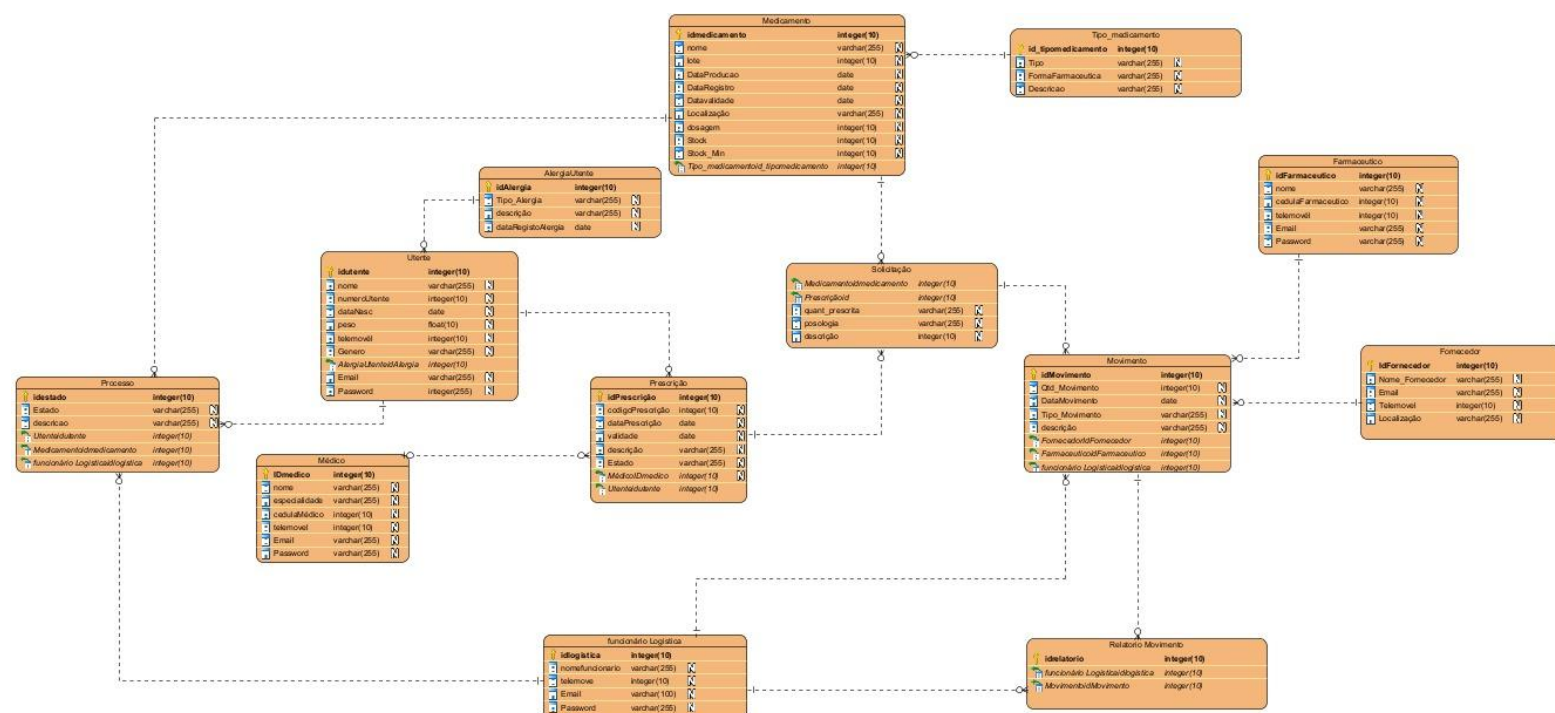


Figura 13- Diagrama de entidade-relação

7. Web Services – Fornecedor

Neste projeto o *Web Service* foi implementado para gerir a comunicação entre a aplicação de gestão de stock de medicamentos e os sistemas externos, como o fornecedor e a farmácia. O objetivo principal deste *Web Service* é permitir a realização de pedidos de medicamentos de forma simples e eficiente.

Este *Web Service* segue o modelo REST (*Representational State Transfer*), utilizando os métodos HTTP POST, GET, PUT e DELETE para realizar as operações necessárias:

- **POST**: Utilizado para criar pedidos de medicamentos, permitindo à farmácia enviar as informações necessárias ao sistema de gestão de stock.
- **GET**: Permite consultar o estado de um pedido ou verificar informações sobre o stock de medicamentos disponíveis.
- **PUT**: Usado para atualizar os detalhes de um pedido existente, como a quantidade de medicamentos solicitados ou o estado do pedido.
- **DELETE**: Utilizado para cancelar ou remover pedidos de medicamentos que não são mais necessários ou que foram erroneamente registados.

Esses métodos garantem uma comunicação eficiente e segura com a aplicação, permitindo que os dados sejam manipulados de forma rápida e com baixo risco de erros. Assim, o *Web Service* foi projetado para ser simples, mas ao mesmo tempo robusto, oferecendo todos os métodos necessários para gerir os pedidos de medicamentos no sistema de forma eficiente.

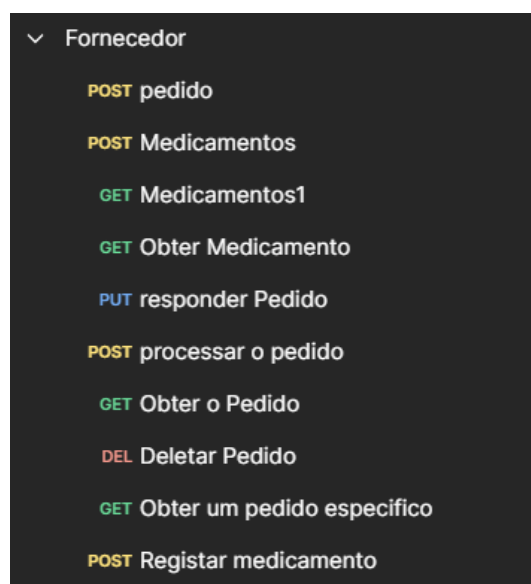


Figura 14- API Fornecedor (WebService)

7.1 Tabela de métodos utilizados na API

Tabela 8- Lista de métodos acessíveis na API

Método	Rota	Operação
POST Pedido	https://localhost:44395/api/Pedido	Fazer Pedido
POST Medicamentos	https://localhost:44395/api/Stock	Registrar Medicamento
GET Medicamento1	https://localhost:44395/api/Stock/1	Obter Medicamento
GET Obter Medicamento	https://localhost:44395/api/Stock	Obter um Medicamento
PUT Responder Pedido	https://localhost:44395/api/pedido/44/responder	Responder ao pedido
POST Processar Pedido	https://localhost:44395/api/pedido/processar	Processar o pedido
GET Obter Pedido	https://localhost:44395/api/Pedido	Obter Pedido
DELETE Deletar Pedido	https://localhost:44395/api/pedido/60	Deletar Pedido
GET Obter um pedido	https://localhost:44395/api/pedido/25	Obter Pedido

As rotas da API foram desenvolvidas para assegurar que todas as operações essenciais do sistema possam ser realizadas de forma simples e eficiente. Através do padrão RESTful, utilizando mensagens em formato JSON, é possível gerir o registo de medicamentos, processar pedidos e consultar informações de forma direta.

Uma rota permite que novos pedidos sejam criados, onde o utilizador informa o fornecedor, o medicamento e a quantidade desejada. Outra rota é responsável por verificar se há stock suficiente para atender ao pedido, aprovando ou recusando automaticamente, com base na disponibilidade. Caso necessário, um administrador pode ajustar a quantidade aprovada ou negar o pedido utilizando uma rota específica para resposta.

Para a gestão de medicamentos, existe uma funcionalidade que permite adicionar novos itens ao stock, bem como consultar todos os medicamentos registados ou apenas os

detalhes de um específico. Também é possível excluir um pedido caso ele não seja mais necessário, garantindo que apenas informações relevantes permaneçam nos sistemas.

Na figura 15 demonstra a rota de processar o pedido em que é utilizado o método POST e é enviada a mensagem em formato JSON.

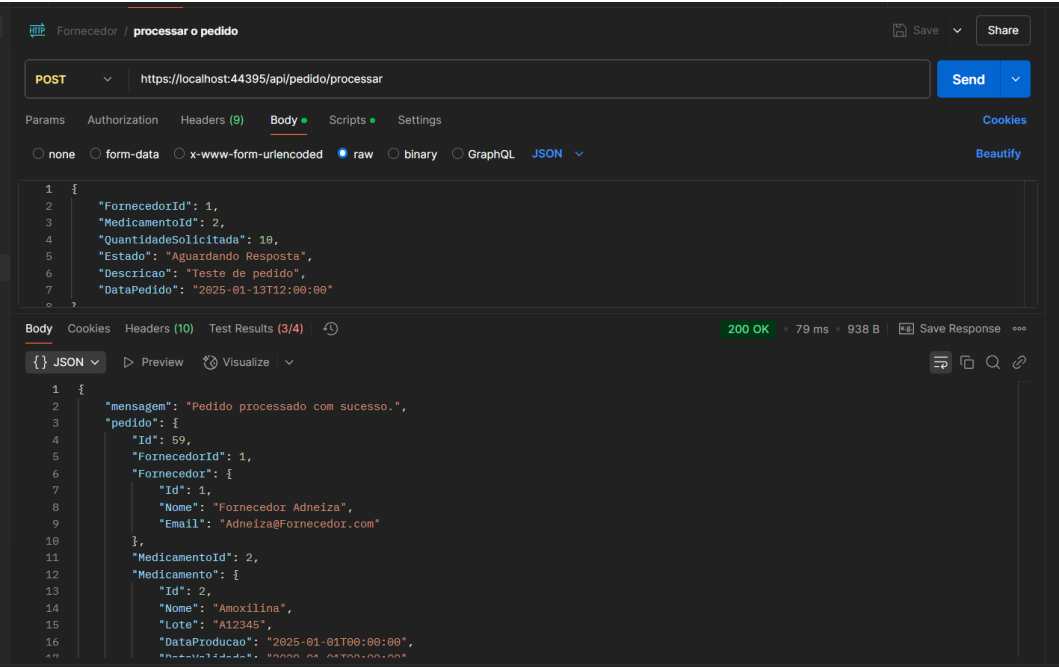


Figura 15- Processar o pedido

Foi realizada uma operação *GET* para obter a resposta aos pedidos efetuados.

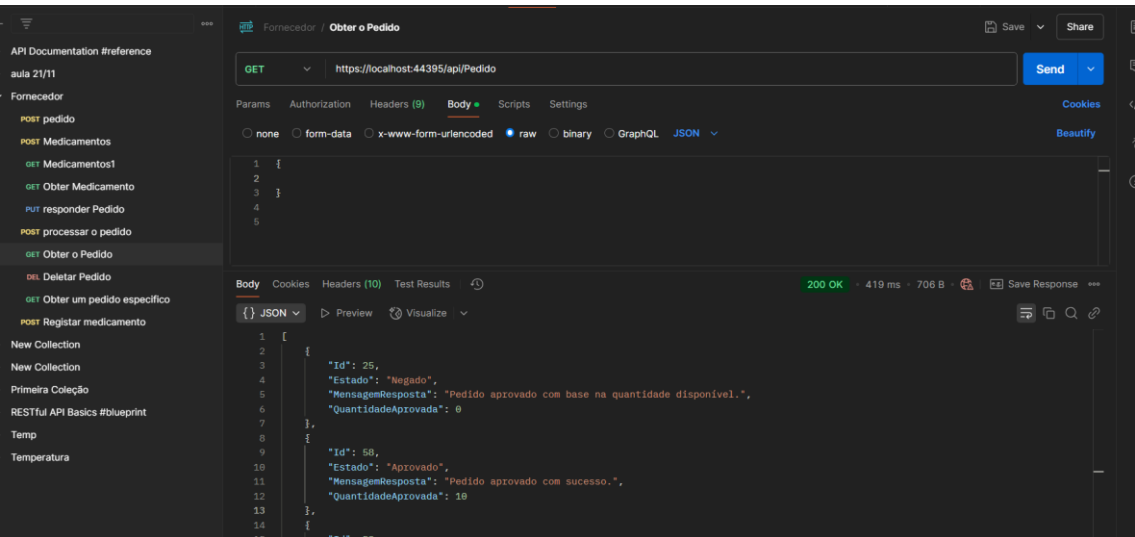


Figura 16- Obter Pedido

A seguir, na figura 17 mostra a rota associada a eliminação do pedido.

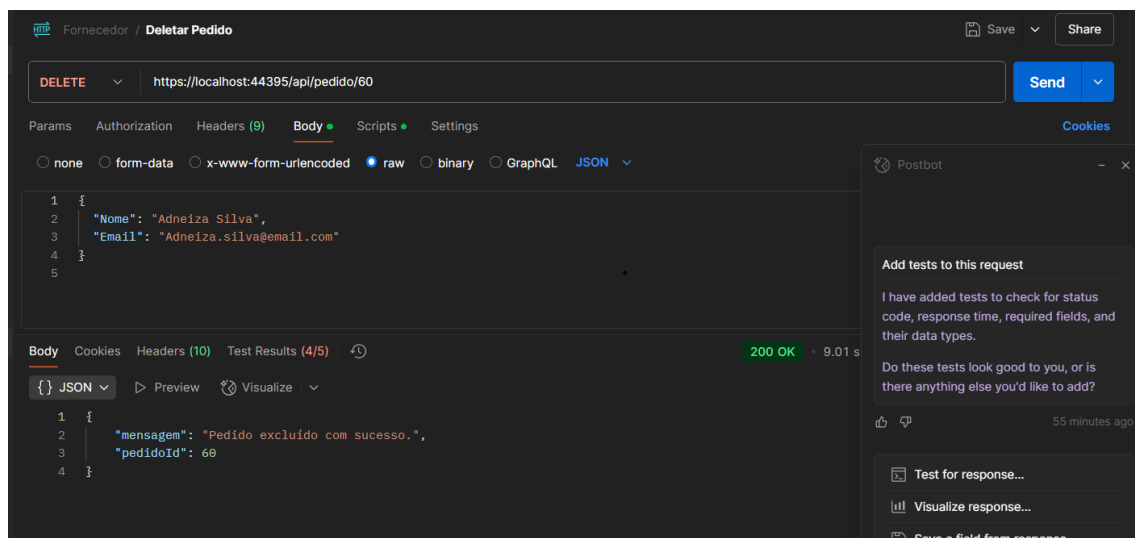


Figura 17- Deletar Pedido

Na figura 18, é apresentada a rota que da resposta ao pedido.

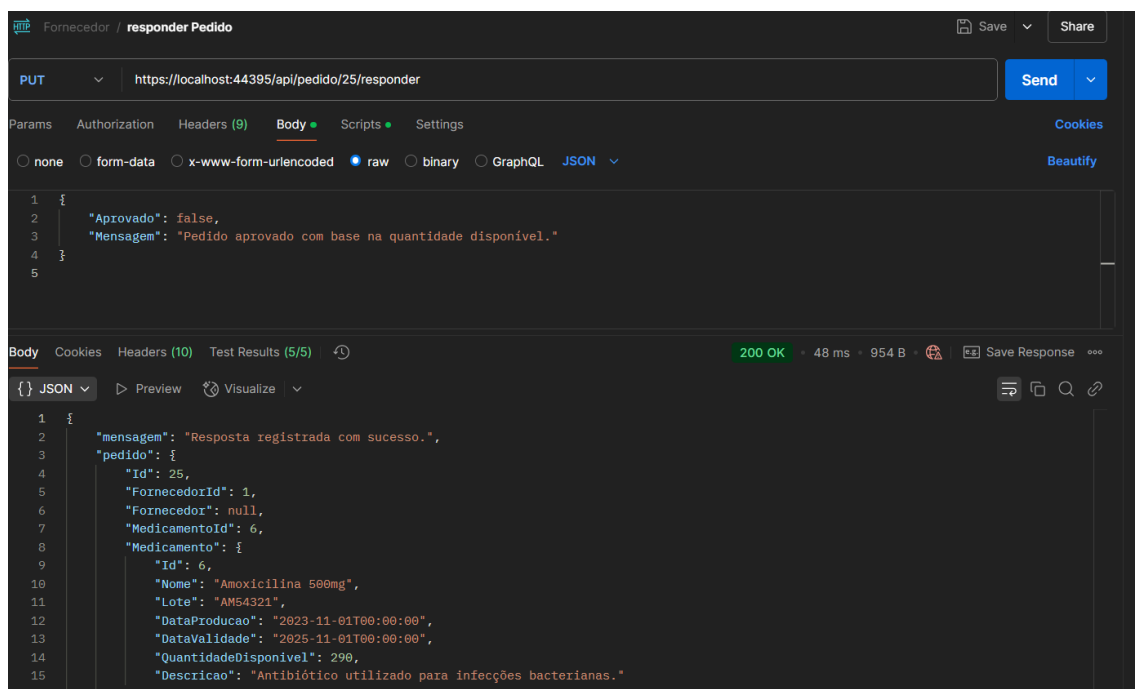


Figura 18- Resposta Pedido

Quando a MediForm realiza um pedido de medicamentos através da API, o sistema responde automaticamente indicando se o pedido foi aprovado ou negado. Se a base de dados da Web API não tiver stock suficiente disponível, o pedido é negado. Caso o stock seja suficiente, o pedido é aprovado e registado automaticamente na base de dados.

Na Figura 19, é ilustrada a lista de pedidos, incluindo casos em que um pedido foi negado pelo fornecedor. A lista apresentada começa com ID a partir do 29, pois, anteriormente, existiam outros pedidos registrados. Contudo, devido a inconsistências de dados, os pedidos anteriores foram removidos para garantir a integridade da informação no sistema.

HomeContact

Lista de Pedidos

Pedido enviado para a API com sucesso.

Resposta da API: {"mensagem":"Pedido processado com sucesso.,"pedido":{"Id":62,"FornecedorId":3,"Fornecedor":{"Id":3,"Nome":"Fornecedor Adneiza","Email":"Adneiza@Fornecedor.com"},"MedicamentId":1,"Medicamento":{"Id":1,"Nome":"Paracetamol","Lote":"A12345","DataProducao":"2025-01-01T00:00:00","DataValidade":"2026-01-01T00:00:00","QuantidadeDisponivel":900,"Descricao":"Medicamento para dores e febres"},"QuantidadeSolicitada":10000,"QuantidadeAprovada":0,"MensagemResposta":"Estoque insuficiente. Pedido negado.,"Estado":"Negado","Descricao":"...","DataPedido":"2023-07-06T07:07:00"}}}

ID	Fornecedor	Medicamento	Quantidade Solicitada	Quantidade Aprovada	Estado	Data do Pedido	Descrição	A
29	Fornecedor Adneiza	Paracetamol	100		Aguardando Resposta	07/07/2025	Taster	
30	Fornecedor Covi	Amoxilina	100		Aguardando Resposta	02/02/2025	...	
31	Fornecedor Covi	Paracetamol	10000		Negado	06/07/2023	...	

Criar Novo Pedido

Figura 19- Pedido negado

Na figura 20, apresenta o pedido aprovado pelo fornecedor (Web Service).

HomeContact

Lista de Pedidos

Pedido enviado para a API com sucesso.

Resposta da API: {"mensagem":"Pedido processado com sucesso.,"pedido":{"Id":63,"FornecedorId":3,"Fornecedor":{"Id":3,"Nome":"Fornecedor Adneiza","Email":"Adneiza@Fornecedor.com"},"MedicamentId":1,"Medicamento":{"Id":1,"Nome":"Paracetamol","Lote":"A12345","DataProducao":"2025-01-01T00:00:00","DataValidade":"2026-01-01T00:00:00","QuantidadeDisponivel":700,"Descricao":"Medicamento para dores e febres"},"QuantidadeSolicitada":200,"QuantidadeAprovada":200,"MensagemResposta":"Pedido aprovado com sucesso.,"Estado":"Aprovado","Descricao":"REPOSITORY","DataPedido":"2025-04-01T21:00:00"}}}

ID	Fornecedor	Medicamento	Quantidade Solicitada	Quantidade Aprovada	Estado	Data do Pedido	Descrição	
29	Fornecedor Adneiza	Paracetamol	100		Aguardando Resposta	07/07/2025	Taster	
30	Fornecedor Covi	Amoxilina	100		Aguardando Resposta	02/02/2025	...	
31	Fornecedor Covi	Paracetamol	10000		Negado	06/07/2023	...	
32	Fornecedor Covi	Paracetamol	200		Aprovado	01/04/2025	REPOSITORY	

Criar Novo Pedido

Copyright © 2024-2025 SGH-System.

Figura 20- pedido aprovado

Na figura 21, é apresentada um erro ao criar o pedido.

Criar Pedido

Erro(s):

- Erro ao criar o pedido. Tente novamente.
- The Fornecedor field is required.
- The Medicamento field is required.
- The MensagemResposta field is required.

FornecedorId

FornecedorSantos

MedicamentoId

Amoxicilina

QuantidadeSolicitada

8

Descricao

..

Estado

Aguardando Resposta

DataPedido

07 / 07 / 7777 , 07:07 AM

Figura 21- Erro pedido

Na figura 22, é apresentada um excerto da base de dados em que apresenta os pedidos efetuados pela MediForm.

100 %									
Results Messages									
	Id	QuantidadeSolicitada	Estado	QuantidadeAprovada	Descricao	MedicamentoId	FornecedorId	DataPedido	MensagemResposta
1	25	10	Negado	0	Pedido de reposição	6	1	2025-01-10 00:00:00.000	Pedido aprovado com base na quantidade disponível.
2	58	10	Aprovado	10	Teste de pedido	2	1	2025-01-13 12:00:00.000	Pedido aprovado com sucesso.
3	59	10	Aprovado	10	Teste de pedido	2	1	2025-01-13 12:00:00.000	Pedido aprovado com sucesso.
4	61	100	Aprovado	100	...	2	3	2025-02-02 08:08:00.000	Pedido aprovado com sucesso.
5	62	10000	Negado	0	...	1	3	2023-07-06 07:07:00.000	Estoque insuficiente. Pedido negado.
6	63	200	Aprovado	200	REPOSITORY	1	3	2025-04-01 21:00:00.000	Pedido aprovado com sucesso.

Figura 22- Pedido efetuado MediForm

8. Testes

Apesar de ser possível efetuar diversos testes ao software desenvolvido, devido à falta de tempo para aprender novas ferramentas, foi utilizado o software *Postman* para testar o processo do pedido de medicamento.

```
1  pm.test("Response status code is 200", function () {
2    pm.response.to.have.status(200);
3  });
4
5
6  pm.test("Response has the required fields", function () {
7    const responseData = pm.response.json();
8
9    pm.expect(responseData).to.have.property("mensagem");
10   pm.expect(responseData).to.have.property("pedido");
11   pm.expect(responseData.pedido).to.have.property("Id");
12   pm.expect(responseData.pedido).to.have.property("FornecedorId");
13   pm.expect(responseData.pedido).to.have.property("MedicamentoId");
14   pm.expect(responseData.pedido).to.have.property("Medicamento");
15   pm.expect(responseData.pedido).to.have.property("QuantidadeSolicitada");
16   pm.expect(responseData.pedido).to.have.property("QuantidadeAprovada");
17   pm.expect(responseData.pedido).to.have.property("Estado");
18   pm.expect(responseData.pedido).to.have.property("Descricao");
19   pm.expect(responseData.pedido).to.have.property("DataPedido");
20 });
21
22
23 pm.test("DateProducao and DataValidade are in a valid date format", function () {
24   const responseData = pm.response.json();
25 }
```

Figura 23- Teste pedido

Foram efetuados os seguintes testes:

Linha 1 – Devolve o código 200(pedido efetuado)

Linha 6 – Resposta com “dados obrigatórios preenchidos”

Linha 23– Código de resposta com “Erro no formato da data”

Linha 32 – Tempo de resposta inferior a 200ms.

```
PASSED Response status code is 200
PASSED Response has the required fields
FAILED DateProducao and DataValidade are in a valid date format | AssertionError: expected '2025-01-01T00:00:00' to match /\d{4}-\d{2}-\d{2}$/
FAILED Response time is less than 200ms | AssertionError: expected 613 to be below 200
```

Figura 24- Teste pedido

```
https://localhost:44395/api/pedido/25/responder

ation Headers (9) Body ● Scripts ● Settings

1 pm.test("Response status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5
6 pm.test("Response time is less than 200ms", function () {
7   pm.expect(pm.response.responseTime).to.be.below(200);
8 });
9
10
11 pm.test("Validate the structure of the pedido object", function () {
12   const responseData = pm.response.json();
13
14   pm.expect(responseData).to.be.an('object');
15   pm.expect(responseData.pedido).to.exist.and.to.be.an('object');
16   pm.expect(responseData.pedido.Id).to.exist.and.to.be.a('number');
17   pm.expect(responseData.pedido.FornecedorId).to.exist.and.to.be.a('number');
18   pm.expect(responseData.pedido.Fornecedor).to.not.exist;
19   pm.expect(responseData.pedido.MedicamentoId).to.exist.and.to.be.a('number');
20   pm.expect(responseData.pedido.Medicamento).to.exist.and.to.be.an('object');
21   pm.expect(responseData.pedido.Medicamento.Id).to.exist.and.to.be.a('number');
22   pm.expect(responseData.pedido.Medicamento.Nome).to.exist.and.to.be.a('string');
23   pm.expect(responseData.pedido.Medicamento.Lote).to.exist.and.to.be.a('string');
24   pm.expect(responseData.pedido.Medicamento.DataProducao).to.exist.and.to.be.a('string');
25   pm.expect(responseData.pedido.Medicamento.DataValidade).to.exist.and.to.be.a('string');
26   pm.expect(responseData.pedido.Medicamento.QuantidadeDisponivel).to.exist.and.to.be.a
```

Figura 25- Resposta ao pedido

Foram efetuados os seguintes testes:

Linha 1– Devolve o código 200(estado do pedido)

Linha 6 – Tempo de resposta inferior a 200ms.

Linha 11– Pedido validado

Linha 32 – Medicamento obtido

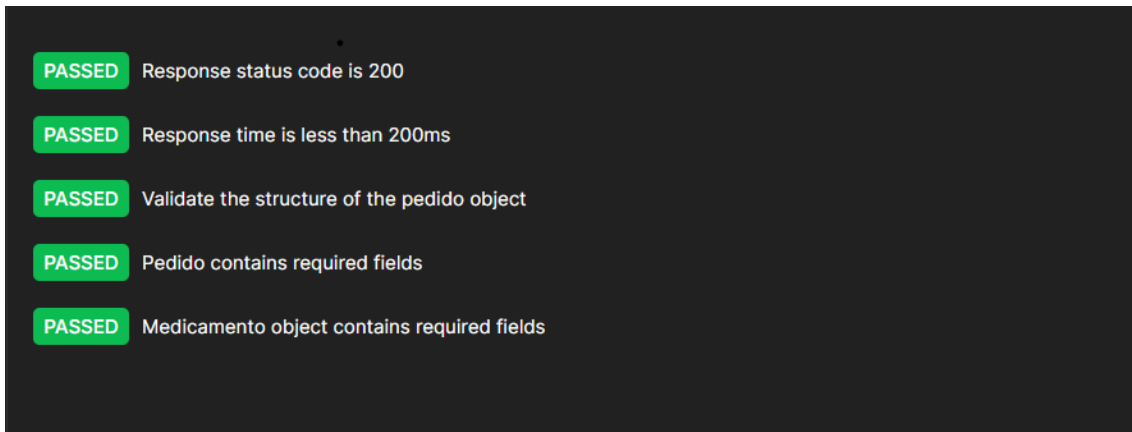


Figura 26- Teste resposta

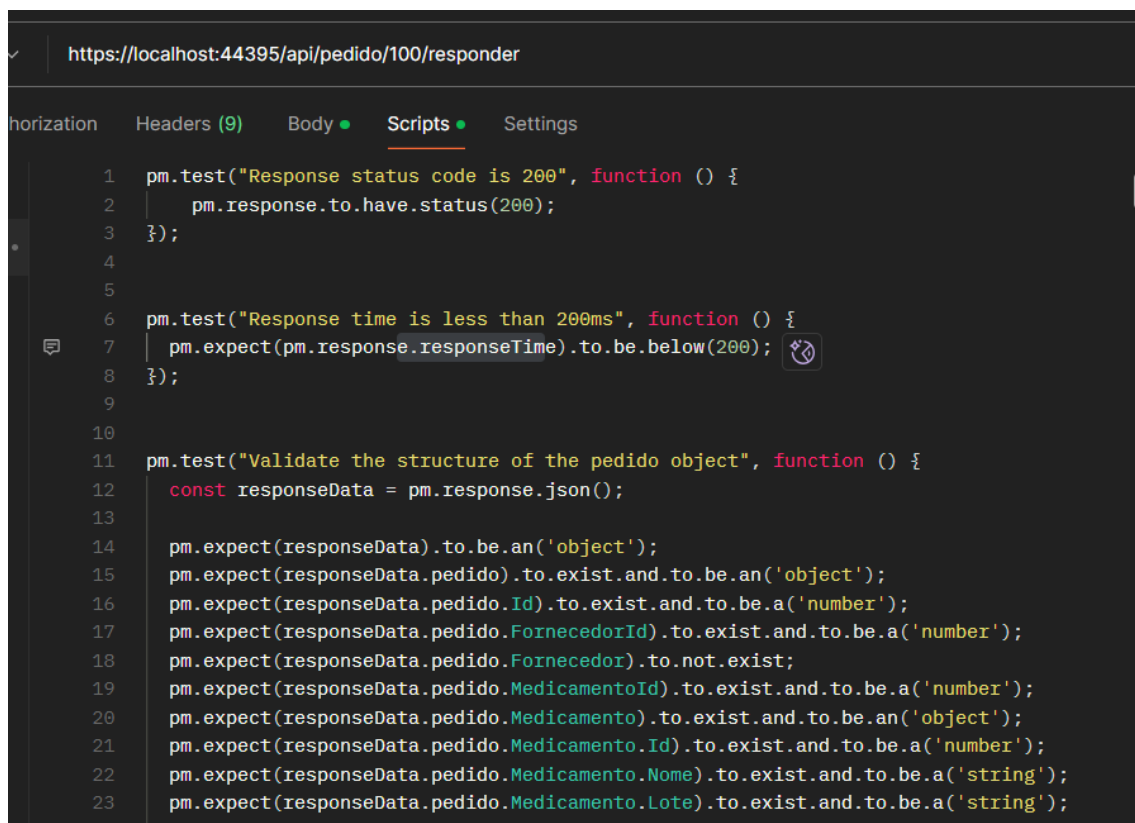


Figura 27- Pedido com Id invalido

Foram efetuados os seguintes testes:

Linha 1– Devolve o código 200(estado do pedido)

Linha 6 – Tempo de resposta inferior a 200ms.

Linha 11– Pedido invalidado

Linha 32 – Medicamento não encontrado

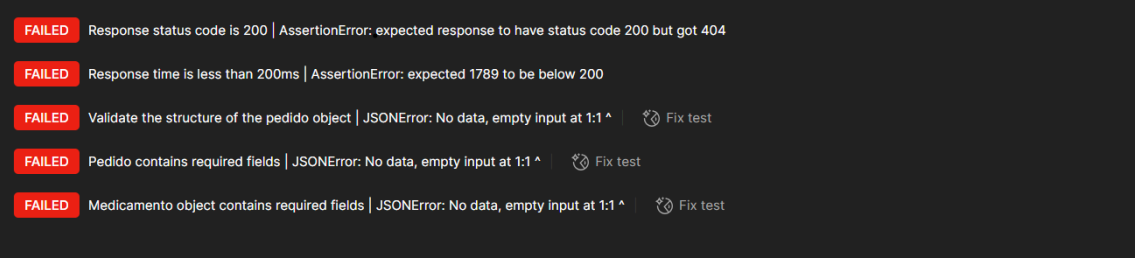
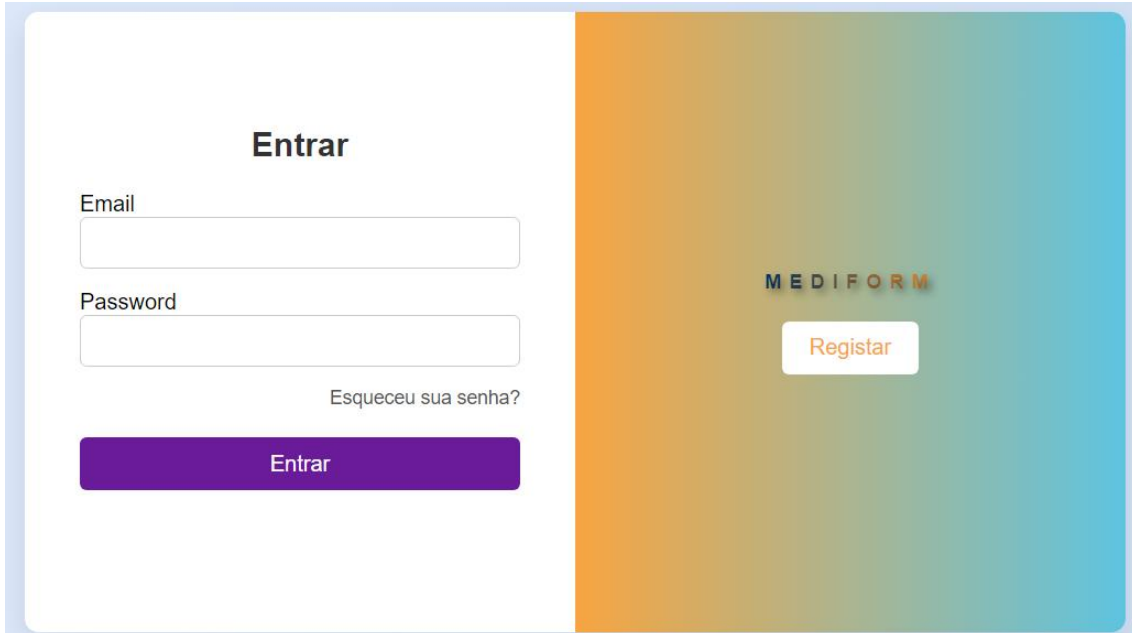


Figura 28- Teste Pedido Invalido

9. Mock-ups

9.1 Login



The login form is titled "Entrar" and is located on the left side of a split-screen layout. The right side features a gradient background with the "MEDIFORM" logo and a "Registrar" button. The login form includes fields for "Email" and "Password", a link for "Esqueceu sua senha?", and a purple "Entrar" button.

Entrar

Email

Password

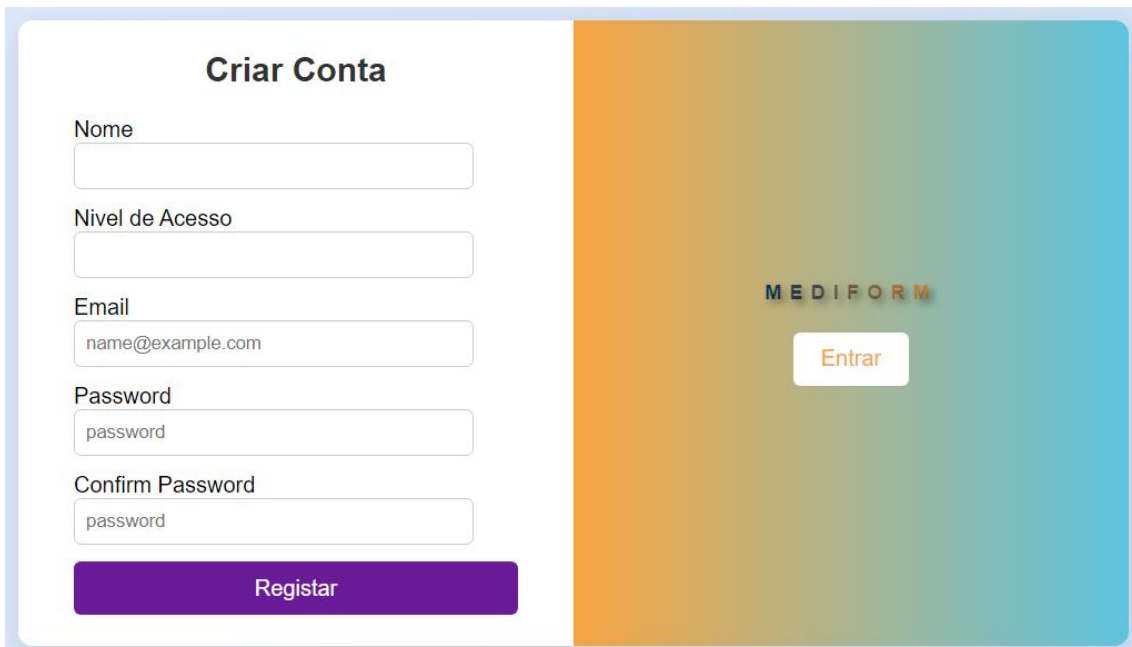
[Esqueceu sua senha?](#)

Entrar

MEDIFORM

Registrar

9.2 Registo de uma nova conta



The registration form is titled "Criar Conta" and is located on the left side of a split-screen layout. The right side features a gradient background with the "MEDIFORM" logo and an "Entrar" button. The registration form includes fields for "Nome", "Nivel de Acesso", "Email", "Password", and "Confirm Password", and a purple "Registrar" button.

Criar Conta

Nome

Nivel de Acesso

Email

Password

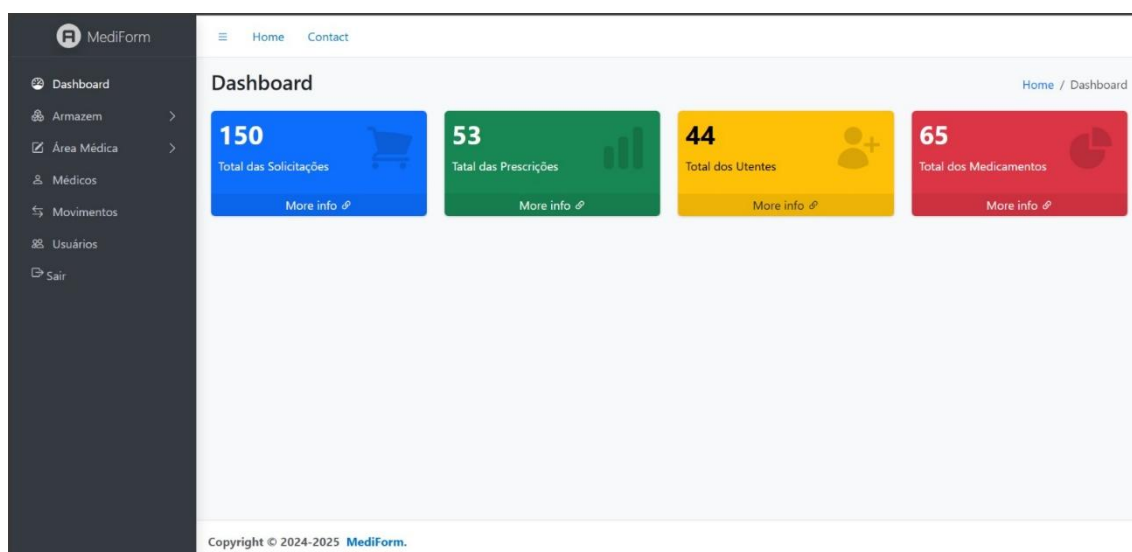
Confirm Password

Registrar

MEDIFORM

Entrar

9.3 Dashboard



10. Conclusão

Apesar de não correr como planeado, o desenvolvimento do sistema de MediForm foi uma experiência desafiadora e trouxe resultados significativos para a automação e melhoria do controlo de medicamentos no ambiente hospitalar non futuro. Embora não tenhamos conseguido concluir todos os processos que inicialmente idealizámos, a base criada para o sistema é sólida.

Ao longo da execução, conseguimos implementar um conjunto de funcionalidades que permitem uma gestão mais ágil e segura dos medicamentos, o que, sem dúvida, contribuirá para um ambiente hospitalar mais organizado e eficiente.

No entanto, a falta de tempo para concluir todas as funcionalidades planeadas deixou algumas lacunas. A principal delas foi a integração automática com o fornecedor de medicamentos, o que dificultou a reposição de stock de forma completamente automatizada. Além disso, não foi possível finalizar a parte de notificações para avisar sobre o baixo stock ou a necessidade de reabastecimento.

Embora as lacunas e as frustrações encontradas o desenvolvimento correu como esperado.

Para melhorias futuras, pretendemos implementar as funcionalidades em falta e melhorar o sistema para que seja mais automatizado.

Algumas melhorias a implementar são a automação de reabastecimento, a criação de relatórios mais detalhados sobre o uso de medicamentos e o aumento de segurança, diminuindo assim em perdas futuras.

11. Bibliografia

- <https://www.infarmed.pt/web/infarmed/entidades/medicamentos-uso-humano/inspecao-medicamentos/medicamentos-manipulados>
- <https://learn.microsoft.com/pt-br/aspnet/core/mvc/overview?view=aspnetcore-9.0>
- <https://www.canva.com/>
- <https://www.ordemfarmaceuticos.pt/pt/noticias/detalhes.php?id=3323>
- <https://pt.stackoverflow.com/>
- Consultamos o Youtube para ver uma maneira simplificada de explicação Angular, ASP.NET e SQL Server.
- Stack Overflow. (2018). **Usando o Identity com Entity Framework para fazer relacionamento one-to-many com foreign key**. Disponível em: <https://pt.stackoverflow.com/questions/332197/usando-o-identity-com-entity-framework-para-fazer-relacionameto-one-to-many-co>. Acesso em 2024.
- Cristo, D. S. R., Cristo, S. B. M. N., Cristo, G. A. M., Cristo, J. S. A. M., Cristo, G. A. L. O., & Neto, A. M. C. (2018). **Impacts caused by polypharmacy on the elderly: an integrative review**. Revista de Saúde Digital, 10(2). <https://doi.org/10.33448/rsd-v10i2.12263>

Repositório Utilizado:

GitHub: <https://github.com/MarianaIPCA/MEDIFORM>