



**Projeto:** MyHealth - Management

**Aluno:** Vicente Loução Duarte

**Curso:** TGPSI23R

**Escola:** Escola Profissional de Tecnologia Digital

**Ano Letivo:** 2024/2025

# Índice

Introdução.....	2
Proposta.....	2
Requisitos do Sistema .....	3
Planeamento e Tecnologias .....	3
Desenvolvimento .....	5
Interface .....	6
Base de Dados .....	7
Funcionalidades-chave.....	8
Conclusão .....	8

## Introdução

Este relatório tem como objetivo apresentar o desenvolvimento da aplicação **"MyHealth - Management"**, desenvolvida como projeto final do segundo ano do curso de programação. Esta aplicação tem como foco a gestão de farmácias através de uma interface intuitiva criada em Windows Forms juntamente com a biblioteca MaterialSkin para um visual moderno e acessível.

## Proposta

Com o objetivo de criar soluções digitais na área da saúde, surgiu a ideia de criar uma ferramenta que permitisse às farmácias gerir eficazmente os seus recursos, como stock, reservas e funcionários. O sistema pretende uniformizar os sistemas farmacêuticos, criar visuais modernos e simples e permitir a digitalização dos processos administrativos.

## Requisitos do Sistema

A aplicação "**MyHealth - Management**" foi desenvolvida com base num conjunto de requisitos funcionais e técnicos previamente definidos. Estes requisitos garantem que a solução cumpre os objetivos propostos e oferece funcionalidades essenciais para a gestão de farmácias. Abaixo listam-se os principais requisitos identificados.

### **REQ0001 – Gestão de Utilizadores (Login e CRUD)**

O sistema deve permitir a autenticação, registo e gestão completa (CRUD) de utilizadores. Funcionalidades incluem:

- Registo de novas contas.
- Login seguro.
- Edição e eliminação de contas.
- Associação a farmácias como donos ou funcionários.
- Validação de permissões consoante o perfil.

### **REQ0002 – Gestão de Medicamentos (CRUD)**

A aplicação deve permitir:

- Registar novos medicamentos.
  - Consultar medicamentos existentes.
  - Atualizar informações (nome, quantidade, validade, etc.).
  - Eliminar medicamentos obsoletos.
- Cada medicamento deve estar associado a uma farmácia específica.

### **REQ0003 – Gestão de Encomendas / Reservas (CRUD)**

O sistema deve permitir:

- Criar novas reservas de medicamentos.
- Visualizar todas as encomendas (ativas e concluídas).
- Editar e eliminar encomendas.
- Associar reservas a utilizadores e farmácias.

### **REQ0004 – Controlo de Stock**

A aplicação deve fornecer mecanismos para:

- Visualizar o stock atual de cada medicamento.
- Atualizar quantidades (entrada/saída).
- Prevenir ruturas através de alertas visuais.
- Relacionar diretamente com vendas e reservas.

### **REQ0005 – Histórico de Encomendas**

Deve manter um registo de todas as reservas realizadas:

- Acessível por farmácia e por cliente.
- Inclui datas, medicamentos envolvidos, quantidades e estados.
- Possível exportação ou visualização filtrada.

### **REQ0006 – Gestão e Histórico de Vendas**

O sistema deve permitir:

- Registrar vendas diretamente na farmácia.
- Consultar o histórico completo de vendas por data, medicamento ou funcionário.
- Eliminar vendas incorretamente lançadas (com permissões).
- Integrar as vendas com o controlo de stock e relatórios estatísticos.

### **REQ0007 – Armazenamento de Dados em Base de Dados SQL Server**

Todos os dados da aplicação devem ser armazenados de forma estruturada e persistente numa base de dados SQL Server, garantindo:

- Segurança.
- Integridade relacional.
- Facilidade de backup e recuperação.

### **REQ0008 – Interface Gráfica (Windows Forms + MaterialSkin)**

A interface deve ser:

- Desenvolvida com Windows Forms.
- Esteticamente moderna usando a biblioteca MaterialSkin.
- Responsiva e intuitiva, com menus e separadores para cada módulo (dashboard, stock, reservas, definições, etc.).

### **REQ0009 – Perfis de Utilizador e Autenticação**

O sistema deve distinguir entre diferentes tipos de utilizadores:

- Administrador/Dono de Farmácia: acesso total à farmácia, gestão de stock, funcionários, reservas, vendas e definições.
- Funcionário: acesso limitado consoante permissões atribuídas (ex.: sem acesso à gestão de funcionários).
- O login deve validar e restringir funcionalidades com base no perfil.

## Planeamento e Tecnologias

- **Linguagem de Programação:** C# (.NET Framework)
- **Interface:** Windows Forms com MaterialSkin
- **Gráficos:** LiveCharts 2 (SkiaSharpView)
- **Base de Dados:** SQL Server + Entity Framework
- **Outras Tecnologias:** Ficheiro JSON para guardar definições

## Desenvolvimento

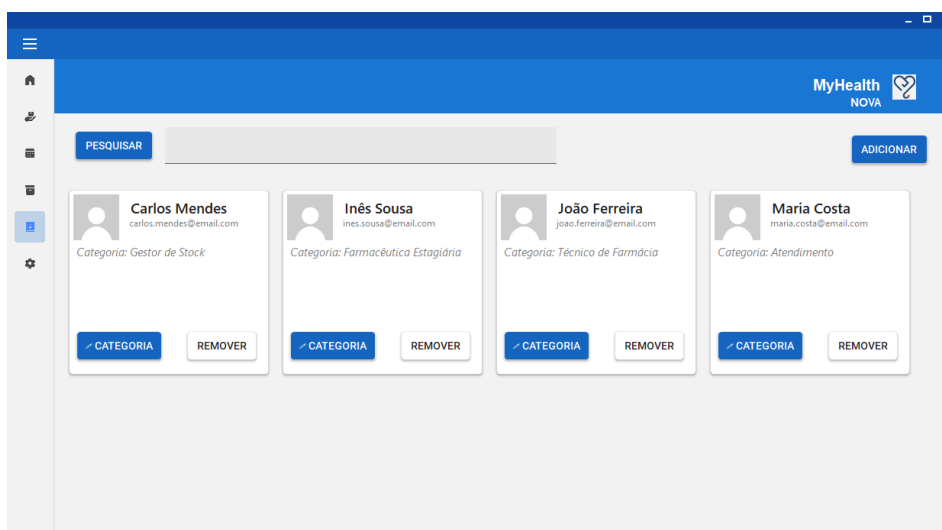
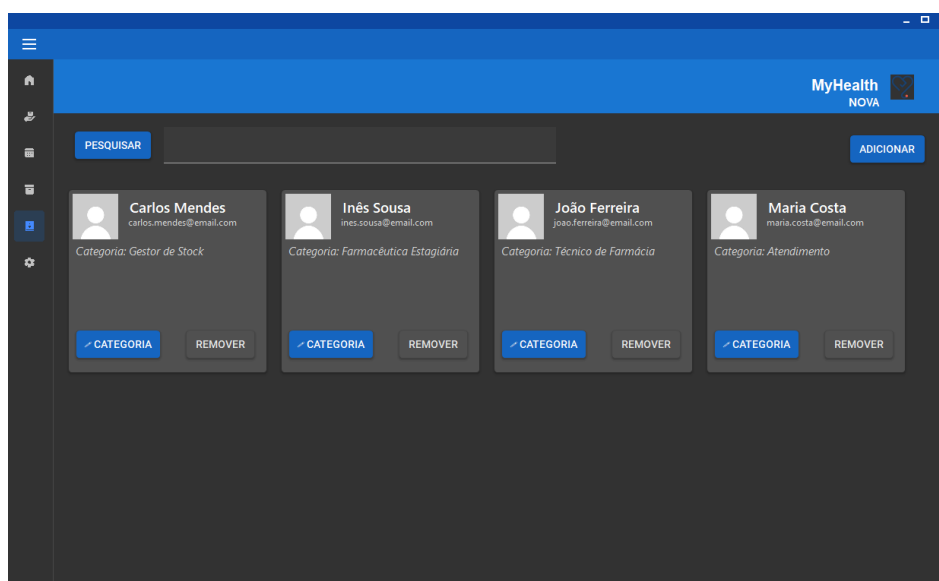
Comecei por trabalhar na parte do backend, ou seja, na base de dados e nas classes responsáveis pela lógica da aplicação. Criei primeiro as tabelas principais como Utilizador, Funcionário, Reserva e Medicamento, e usei o Entity Framework para ligar estas tabelas ao C#. Depois de garantir que o acesso aos dados estava a funcionar corretamente, passei a desenvolver a interface, página por página.

De seguida passei para a página inicial (Dashboard), depois avancei para as secções de vendas, reservas, stock e funcionários. Em cada uma, implementei primeiro a interface gráfica e no fim liguei os dados e as funcionalidades. No final, corriji alguns erros e fiz alguns ajustes.

Esta forma de trabalhar ajudou-me a manter o projeto organizado e a testar cada página individualmente.

## Interface

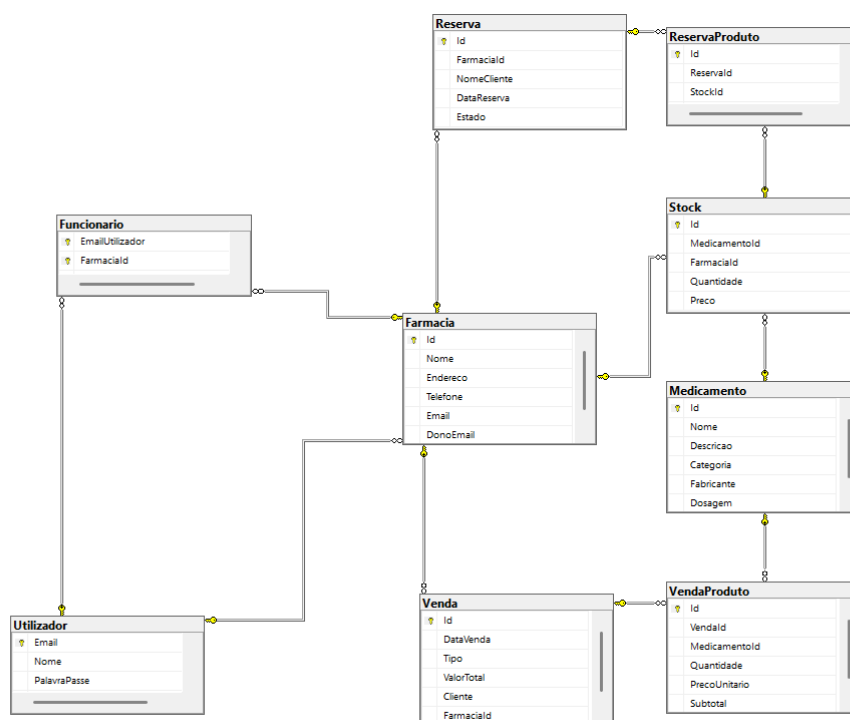
A interface foi feita com MaterialSkin para oferecer um design limpo, moderno e consistente com as tendências atuais. Foi utilizado um layout baseado em abas (TabControl), o que permitiu separar de forma clara cada área funcional da aplicação, como o dashboard, gestão de stock, reservas, funcionários e definições. Cada aba contém componentes adaptados à sua função específica, como tabelas de dados, botões, caixas de texto, combo boxes e switches. O objetivo foi tornar a navegação fácil, intuitiva e acessível. As cores e temas também podem ser ajustados pelo utilizador, melhorando a experiência visual.



## Base de Dados

A base de dados foi criada em SQL Server, com um conjunto de tabelas ligadas entre si por chaves primárias e estrangeiras. Entre as tabelas principais estão Utilizador, Funcionario, Reserva, Medicamento, Stock e Farmacia. Cada tabela foi pensada para representar uma parte importante da gestão da farmácia.

Utilizei o Entity Framework como ORM (Object Relational Mapper), o que facilitou muito a ligação entre a base de dados e o código em C#. Com esta abordagem, foi possível aceder, adicionar, editar e remover dados através de objetos no código, sem precisar escrever comandos SQL diretamente. Isto tornou o desenvolvimento mais rápido e menos propenso a erros.



## Funcionalidades-chave

- **Gestão de Vendas e Stock:** Foi desenvolvida uma interface dedicada à consulta e edição do stock disponível, bem como ao acompanhamento de vendas realizadas através das reservas efetuadas. O utilizador pode visualizar os medicamentos existentes, suas quantidades, e atualizar facilmente as informações. Esta gestão permite à farmácia manter o controlo de inventário e garantir que não faltam produtos essenciais.
- **Gráficos com LiveCharts:** A página inicial apresenta gráficos de barras e de linhas usando a biblioteca LiveCharts 2, que mostram dados como número de reservas e estado do stock. Estes gráficos são atualizados dinamicamente com base na base de dados.
- **Definições Personalizadas:** O utilizador pode escolher entre tema claro e escuro, mudar o tamanho da fonte e ativar ou desativar notificações. Estas preferências são guardadas num ficheiro JSON, que é lido automaticamente ao iniciar a aplicação para manter as definições escolhidas anteriormente. Foram criadas várias tabelas, como Utilizador, Funcionario, Reserva, Medicamento, entre outras. Utilizou-se o Entity Framework para facilitar o acesso aos dados.

## Conclusão

A aplicação cumpre os objetivos definidos, sendo uma ferramenta útil para gestão de farmácias. Foram adquiridos conhecimentos sólidos em desenvolvimento desktop com C#, bem como no uso de bibliotecas modernas e acesso a bases de dados. Como melhorias futuras, poderia ser feita uma integração com uma API para validação de receitas médicas ou sincronização com a aplicação móvel do cliente.

## Referências Bibliográficas

- <https://learn.microsoft.com/pt-pt/dotnet/desktop/winforms/>
- <https://learn.microsoft.com/pt-pt/sql/sql-server/>
- <https://chat.openai.com/>
- <https://www.youtube.com/>