



Projeto: MyHealth - Management

Aluno: Vicente Loução Duarte

Curso: TGPSI23R

Escola: Escola Profissional de Tecnologia Digital

Ano Letivo: 2024/2025

Índice

Introdução.....	2
Proposta.....	2
Público alvo	3
Requisitos Funcionais (RF).....	3
Requisitos Técnicos (RT)	4
Requisitos Não Funcionais (RNF)	5
Planeamento e Tecnologias	6
Desenvolvimento	6
Interface	7
Base de Dados	8
Funcionalidades-chave.....	8
Conclusão	9
Referências Bibliográficas.....	9

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.

Público alvo

A **MyHealth – Management** tem como público alvo farmácias que desejam modernizar e ou facilitar a sua gestão. Atende os requisitos tanto de donos como de funcionários.

Requisitos Funcionais (RF)

REQ001 – Gestão de Contas de Utilizador (Autenticação e CRUD)

- O sistema deve permitir o registo, autenticação e gestão de utilizadores com validação de permissões.
- Funcionalidades incluídas:
 - Registo de novas contas de utilizador.
 - Associação de contas a farmácias (como Dono ou Funcionário).
 - Acesso controlado com base no perfil.
 - Logout e gestão de sessões.
 - Interface para alterar dados da conta.

REQ002 – Gestão de Farmácias

- Possibilidade de adicionar, editar ou remover farmácias.
- Cada farmácia é associada a um dono.
- Visualização de detalhes da farmácia, incluindo nome, localização e contacto.
- Interface para associar funcionários a uma farmácia.

REQ003 – Gestão de Stock de Medicamentos

- Adição de medicamentos ao stock.
- Atualização da quantidade em stock.
- Associação de cada stock a uma farmácia e a um medicamento base.
- Prevenção de ruturas com alertas visuais para stock baixo.
- Visualização detalhada do stock atual por farmácia.
- Integração com vendas e reservas para atualização automática.

REQ004 – Gestão de Encomendas/Reservas

- Criação de novas reservas por parte dos funcionários.
- Registo de data, estado, lista de medicamentos e quantidades.

- Consulta de reservas ativas, pendentes e concluídas.
- Edição e cancelamento de reservas.
- Integração com o stock e relatórios.

REQ005 – Gestão e Histórico de Vendas

- Registo manual de vendas na farmácia.
- Associação a data, farmácia e medicamentos vendidos.
- Eliminação de vendas incorretamente registadas.
- Atualização automática do stock ao realizar uma venda.
- Geração de estatísticas para dashboard.

REQ006 – Sistema de Definições Personalizável

- Tema escuro / claro com persistência das preferências (ficheiro JSON).
- Opção de manter sessão iniciada.

REQ007 – Pesquisa e Filtros Avançados

- Barra de pesquisa por palavras-chave (nome, email, medicamento, etc.).

REQ008 – Dashboard com Estatísticas

- Total de vendas.
- Vendas por medicamento.
- Percentagem de vendas.
- Vendas normais vs Vendas por Reserva.
- Stock baixo.
- Reservas pendentes.
- Gráficos criados com LiveCharts 2 (LiveChartsCore.SkiaSharpView.WinForms).
- Atualização automática com base nos dados da base de dados.

Requisitos Técnicos (RT)

RT001 – Armazenamento com Base de Dados SQL Server

- Toda a informação da aplicação é persistida numa base de dados relacional.
- Integridade referencial garantida com chaves estrangeiras e constraints.

RT002 – Ligação com Entity Framework

- Utilização do ORM Entity Framework para acesso aos dados.
- Classes modelo bem definidas com Data Annotations.
- Utilização de LINQ para consultas e operações CRUD.

RT003 – Interface Gráfica com Windows Forms + MaterialSkin

- UI moderna baseada em Material Design com MaterialSkin.
- Design visual padronizado em todas as páginas.
- Navegação com separadores (TabControl) para: Dashboard, Reservas, Stock, Funcionários, Vendas, Definições.
- Layout responsivo com espaçamento e alinhamento consistentes.
- Utilização de ícones, cores e feedback visual para boa usabilidade.

RT004 – Segurança e Controlo de Acesso

- Validação de login e senha no arranque da aplicação.
- Validação de inputs e mensagens de erro personalizadas.

RT005 – Manutenção e Extensibilidade

- Estrutura preparada para integração futura de novas funcionalidades.

Requisitos Não Funcionais (RNF)

RNF001 – Usabilidade

- Interface simples e intuitiva para utilizadores não técnicos.
- Ícones e cores com contraste adequado.
- Feedback visual (mensagens de sucesso, erro, carregamento).

RNF002 – Desempenho

- Abertura da aplicação e carregamento de dados em tempo aceitável (<2s em listas).
- Otimização de consultas com uso de indexes onde necessário.

RNF003 – Fiabilidade

- Tolerância a falhas: tratamento de exceções em todas as operações críticas.
- Mensagens de erro claras para problemas de rede ou base de dados.

RNF004 – Persistência

- Todas as alterações e preferências do utilizador guardadas localmente (ficheiro JSON).
- Dados na base de dados são persistentes mesmo após falha de energia.

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

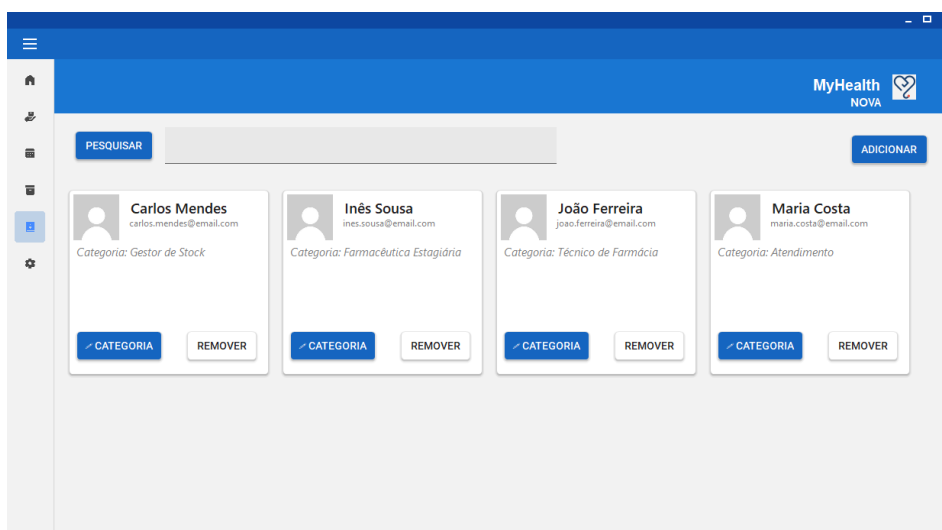
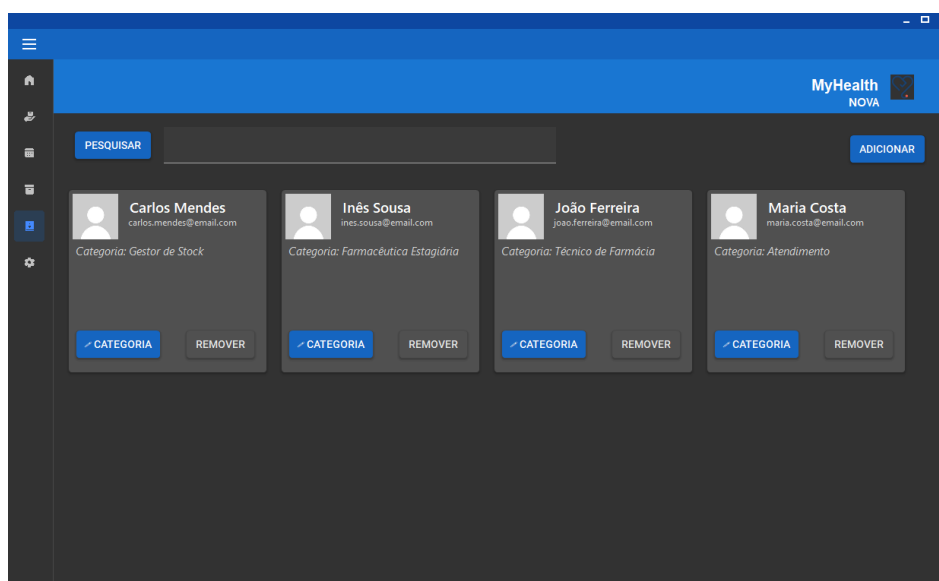
Comecei por trabalhar na parte do backend, ou seja, na base de dados e nas classes responsáveis pela lógica da aplicação. Criei primeiro as tabelas principais como Utilizador, Funcionário, Reserva e Medicamento, e usei o Entity Framework para ligar estas tabelas ao C#. Depois de garantir que o acesso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

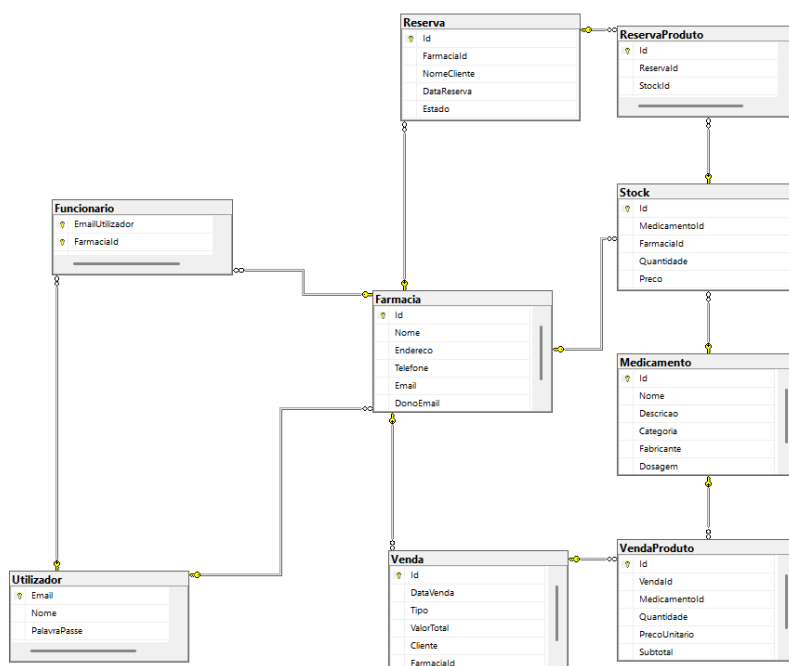
A interface foi feita com MaterialSkin para oferecer um design limpo, moderno e consistente com as tendências atuais. Foi utilizado um layout baseado em abas (TabControl), o que permitiu separar de forma clara cada área funcional da aplicação, como o dashboard, gestão de stock, reservas, funcionários 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/>