



MANUAL DE PROGRAMADOR

Programa: MyHealth - Management

Criador: Vicente Loução Duarte

Índice

MANUAL DE PROGRAMADOR	1
Introdução.....	2
Requisitos do Sistema	2
Base de Dados	3
Principais classes e Funções	4
Entity Framework (Acesso a Dados)	4
ConfiguracoesApp	7
ConfigManager	7
Contas	7
Theme	8
Funcionalidades Técnicas.....	8
Como Instalar?	9

Introdução

Este manual tem como objetivo fornecer toda a informação técnica necessária para compreender, instalar, modificar e dar manutenção ao sistema **"MyHealth - Management"**. O projeto foi desenvolvido em C# com Windows Forms, utilizando o MaterialSkin para o design da interface e o Entity Framework para a camada de acesso a dados.

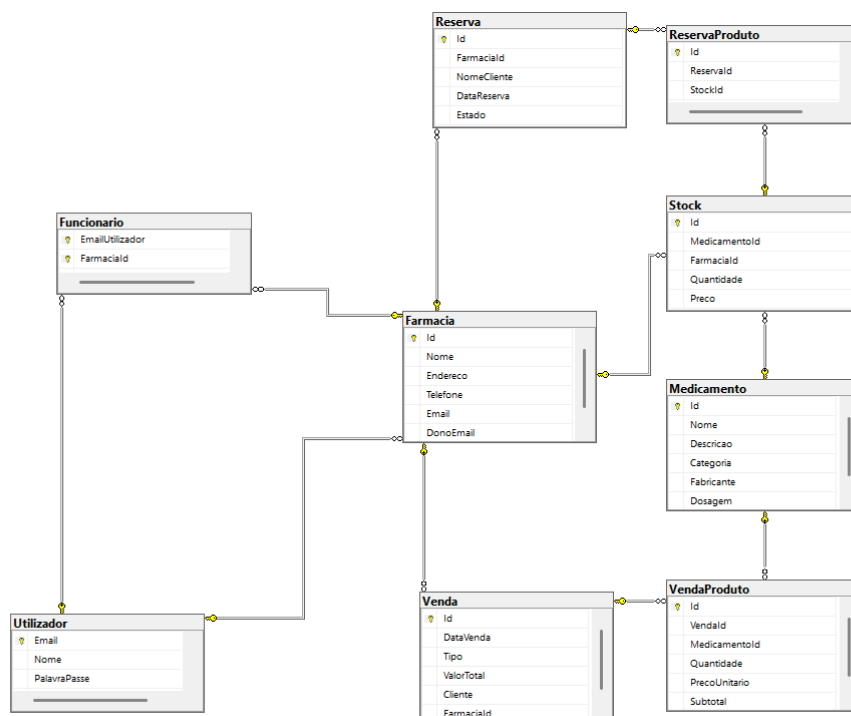
Requisitos do Sistema

Requisitos de Software:

- Microsoft Windows 10 ou superior
- .NET Framework 4.7.2 ou superior
- Visual Studio 2019 ou superior
- SQL Server (local ou remoto)
- Entity Framework 6
- Pacotes NuGet:
 - MaterialSkin.2
 - LiveCharts
 - Newtonsoft.Json

Base de Dados

- Criada com SQL Server.
- Ligação feita com Entity Framework 6.
- Tabelas principais (funcionalidade central):
 - **Utilizador** – guarda dados dos utilizadores da plataforma (ex: nome, email, palavra-passe, imagem).
 - **Funcionario** – representa os funcionários associados a uma farmácia, com a respetiva categoria.
 - **Reserva** – regista reservas de medicamentos feitas pelos clientes.
 - **Medicamento** – contém informações sobre os medicamentos disponíveis.
 - **Farmacia** – representa farmácias registadas na plataforma.
 - **Stock** – associa medicamentos às farmácias e indica a quantidade e preço de cada um.
 - **ReservaProduto** – relaciona reservas com os produtos/medicamentos reservados.
 - **Venda** – regista vendas (normais ou por encomenda) feitas pelas farmácias.
 - **VendaProduto** – detalha os medicamentos vendidos em cada venda (quantidade, preço, subtotal).



Principais classes e Funções

Entity Framework (Acesso a Dados)

Todas as classes relacionadas com dados provêm do modelo Entities, gerado automaticamente pelo Entity Framework a partir da base de dados MyHealth.

Tabela: Utilizador

Representa os utilizadores da aplicação.

- Campos:
 - - Email (chave primária)
 - - Nome
 - - PalavraPasse
 - - Imagem

Tabela: Farmacia

Representa as farmácias registadas.

- Campos:
 - - Id (chave primária)
 - - Nome
 - - Endereco
 - - Telefone
 - - Email
 - - DonoEmail (chave estrangeira para Utilizador.Email)

Tabela: Funcionario

Representa os funcionários de uma farmácia.

- Campos:
 - - Chave primária composta: EmailUtilizador, Farmaciald
 - - Categoria

Tabela: Medicamento

Contém os dados dos medicamentos disponíveis.

- Campos:
 - - Id (chave primária)
 - - Nome
 - - Descricao
 - - Categoria
 - - Fabricante
 - - Dosagem

Tabela: Stock

Representa o stock de medicamentos em cada farmácia.

- Campos:
 - - Id (chave primária)
 - - Medicamentoid (chave estrangeira para Medicamento.Id)
 - - Farmaciald (chave estrangeira para Farmacia.Id)
 - - Quantidade
 - - Preco

Tabela: Reserva

Representa as reservas feitas por clientes em farmácias.

- Campos:
 - - Id (chave primária)
 - - Farmaciald (chave estrangeira para Farmacia.Id)
 - - NomeCliente
 - - DataReserva
 - - Estado

Tabela: ReservaProduto

Lista os medicamentos incluídos em cada reserva.

- Campos:
 - - Id (chave primária)
 - - Reservald (chave estrangeira para Reserva.Id)
 - - StockId (chave estrangeira para Stock.Id)
 - - Quantidade

Tabela: Venda

Regista as vendas efetuadas, com ou sem reserva.

- Campos:
 - - Id (chave primária)
 - - DataVenda
 - - Tipo ('encomenda' ou 'normal')
 - - ValorTotal
 - - Cliente
 - - FarmaciaId (chave estrangeira para Farmacia.Id)

Tabela: VendaProduto

Lista os medicamentos vendidos em cada venda.

- Campos:
 - - Id (chave primária)
 - - VendaId (chave estrangeira para Venda.Id)
 - - MedicamentoId (chave estrangeira para Medicamento.Id)
 - - Quantidade
 - - PrecoUnitario
 - - Subtotal

Classe: Entities

- Representa o contexto da base de dados gerado pelo Entity Framework.
- Permite aceder às tabelas da base de dados através de propriedades do tipo `DbSet<T>`, como:

- - `DbSet<Utilizador>` Utilizador
- - `DbSet<Farmacia>` Farmacia
- - `DbSet<Funcionario>` Funcionario
- - `DbSet<Medicamento>` Medicamento
- - `DbSet<Stock>` Stock
- - `DbSet<Reserva>` Reserva
- - `DbSet<ReservaProduto>` ReservaProduto
- - `DbSet<Venda>` Venda
- - `DbSet<VendaProduto>` VendaProduto

ConfiguracoesApp

Representa as preferências/configurações do utilizador (tema, idioma, tamanho da fonte, notificações, etc.).

Propriedades:

- ModoEscuro (*bool*)
- Idioma (*string*)
- TamanhoFonte (*string*)
- NotificacoesAtivas (*bool*)
- AtualizacoesAutomaticas (*bool*)
- ManterSessaoIniciada (*bool*)
- UtilizadorAtual (*string*)

ConfigManager

Classe estática responsável por carregar e guardar as configurações da aplicação no ficheiro config.json.

Funções principais:

- Carregar()
Carrega as configurações do ficheiro JSON ou cria um ficheiro com valores por defeito.
- Guardar()
Guarda as configurações atuais no ficheiro config.json.
- Load()
Lê diretamente as configurações, se existirem.

Contas

Classe responsável pela gestão de conta e autenticação dos utilizadores.

Campos Estáticos:

- Email (*string*) – Email do utilizador autenticado.
- Farmacia (*int*) – ID da farmácia associada ao funcionário (se aplicável).

Funções principais:

- Login(string email, string password)
Autentica um utilizador ao verificar as credenciais na base de dados.
- CriarConta(string email, string password)
Cria um novo utilizador, após validar que o email ainda não existe.
- ValidarEmail(string email)
Valida o formato do email (trim, lowercase e expressão regular).

Theme

Classe responsável pela aplicação dinâmica do tema claro ou escuro através da biblioteca MaterialSkin.

Funções principais:

- TemaAtual
Retorna o tema atual definido nas configurações.
- AplicarTema(MaterialForm form, bool modoEscuro)
Aplica dinamicamente o tema selecionado ao formulário.

Funcionalidades Técnicas

a) Gestão de Funcionários

- Listagem, adição e pesquisa.
- Integração com a tabela de Utilizador.

b) Gráficos com LiveCharts

- Dashboard com gráfico de barras e linha.
- Ligado a dados reais da base de dados.

c) Sistema de Preferências

- Guardar tema, idioma e fonte em AppSettings.json.
- Carregado automaticamente no arranque.

Como Instalar?

1. Criar a Base de Dados

- Abra o SQL Server Management Studio.
- Crie uma base de dados chamada MyHealth.
- Corra o script SQL fornecido para criar as tabelas.

2. Definir a Connection String

- No ficheiro App.config, dentro de <connectionStrings>, coloque:

```
<add name="Entities" connectionString="Data  
source=(localdb)\MSSQLLocalDB;Initial Catalog=MyHealth;Integrated  
Security=True;" providerName="System.Data.SqlClient" />
```

- Alternativamente, use os dados do seu servidor SQL.

3. Atualizar o Modelo

- Botão direito no ficheiro .edmx → **"Update Model from Database"**.

4. Executar a Aplicação

- Carregue em **F5** ou no botão **Start** no Visual Studio.