Relatório Final

Projeto Final Tecnologias Internet III

Bruno Silva 2022110 2LEI

Gestão Medicação



Prof João Rebelo

Ano Letivo 2023/2024

ISTEC – Instituto Tecnologias Avançadas Porto

Índice

[Introdução 3](#_Toc170425634)

[Metodologia 4](#_Toc170425635)

[Diagrama de Classes 5](#_Toc170425636)

[Diagrama de Use-Cases 6](#_Toc170425637)

[Tabela Base Dados 7](#_Toc170425638)

[Models 8](#_Toc170425639)

[Views 10](#_Toc170425640)

[Controllers 12](#_Toc170425641)

[Visão FrontEnd do projeto 14](#_Toc170425642)

[Conclusão 15](#_Toc170425643)

# Introdução

O projeto apresentado tenta solucionar e ajudar na organização do consumo de medicação para uma pessoa.

Em Portugal, e com uma população cada vez mais envelhecida, é normal os utentes esquecerem-se da toma de certos medicamentos, ou até nem terem referência sobre os mesmos para consulta futura, quer por perda da documentação, quer por esquecimento.

Principais objetivos da solução:

* Fazer a gestão de medicação de uma pessoa
* Fazer a gestão do calendário de consumo
* Criação, Leitura, Atualização e Eliminação de medicamentos na plataforma

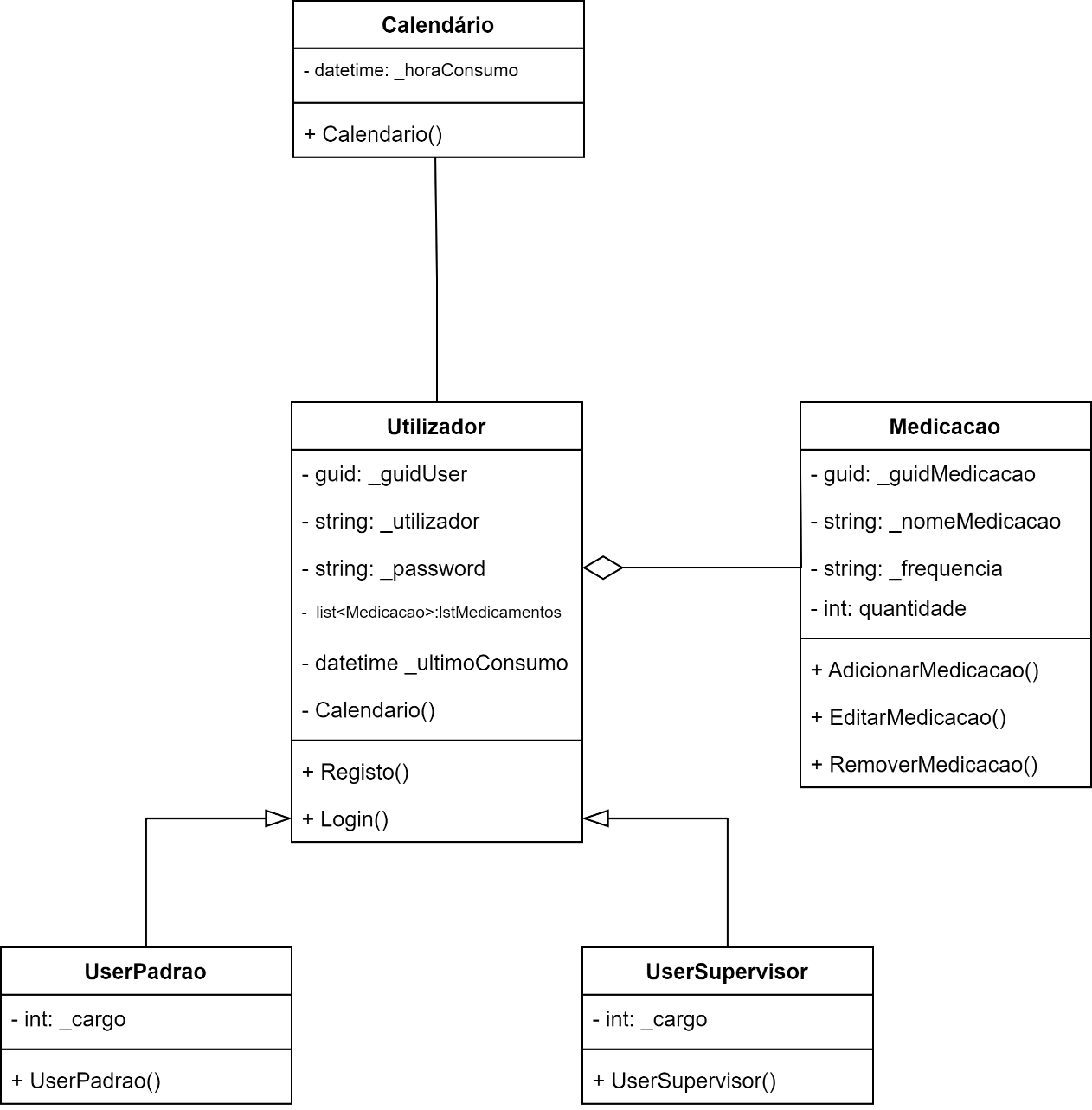
Deixar a nota que a solução, apesar de ser um CRUD, tem apenas a parte de Criação e Leitura funcionais.

# Metodologia

O projeto constitui a seguinte metodologia:

* Um utilizador
* Para cada utilizador, vários medicamentos (lista)
* Para cada utilizador, apenas um calendário
* Um utilizador pode ser tanto um utilizador padrão como um utilizador supervisor (para gestão da plataforma)

# Diagrama de Classes



# Diagrama de Use-Cases

Uma imagem com texto, captura de ecrã, círculo, design

Descrição gerada automaticamente

# Tabela Base Dados

A base de dados é composta por tabelas tUtilizadores e tMedicação, ambas responsáveis por armazenar tanto os respetivos utilizadores e a medicação de cada um.

Tabela tUtilizadores:

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Tabela tMedicação:

Uma imagem com texto, captura de ecrã, número, ecrã

Descrição gerada automaticamente

# Models

Nas classes Conta e Medicamento, defini as regras de negócio para cada classe.

A classe Conta define as regras de negócio para cada um dos seus atributos e métodos. No caso, todos os atributos são tanto de leitura como de escrita.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

A classe Medicamento também segue a mesma ideia:

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

A classe MedicamentoHelper contém as funções essenciais do código, como a de Listar a medicação para um determinado utilizador, por exemplo.

A mesma herda a partir da classe HelperBase.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

A classe ContaHelper encarrega-se de todas as operações para as contas da solução, como a respetiva autenticação, registo e serialização para a sessão.

Uma imagem com texto, captura de ecrã, ecrã, software

Descrição gerada automaticamente

# Views

As views possuem todas as interfaces gráficas que o utilizador irá diretamente interagir na solução.

Como base para a View, foi usado no Layout a navbar para toda a plataforma, e a respetiva função entre o Login/Logout.

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

Nas views de Login e Registar, foram usados inputs, para fazerem o request POST para o backend, para efetuar o Login ou o Registo, respetivamente.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

Uma imagem com captura de ecrã, texto, software, Software de multimédia

Descrição gerada automaticamente

Para as views de Medicamento, em Editar e Listar, o acesso às mesmas é limitado na View, onde verifica se o utilizador tem uma sessão válida ativa; caso contrário, redireciona para a raíz do servidor.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

# Controllers

Os controladores na solução estão a efetuar as funções principalmente do backend, para cada tipo de request presente vindo dos outros componentes da solução.

A classe ContaController faz interface com os requests e tipos de requests recebidos do frontend, neste caso para o Login, Logout, Sessão e Registo.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

A mesma ideologia acontece para a classe MedicamentoController, com o Listar e Editar:

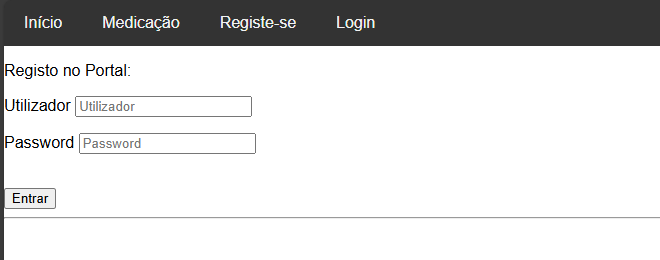
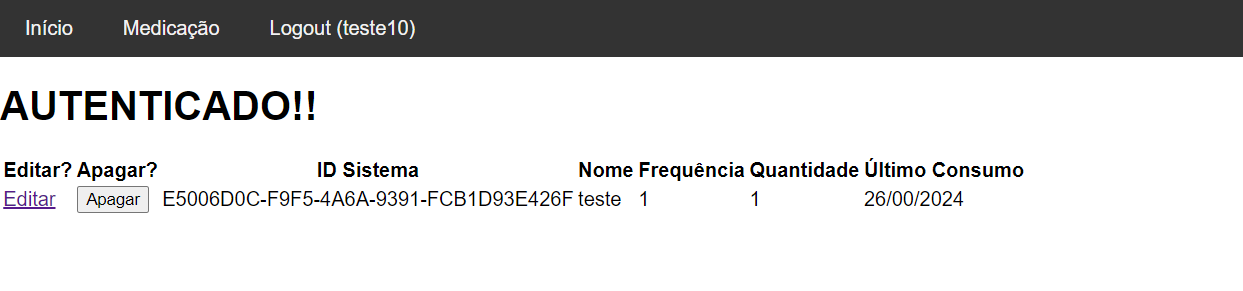
Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

# Visão FrontEnd do projeto

Em continuação com o desenvolvimento das Views, o FrontEnd dispõe-se do seguinte:

* Página de Login de Utilizadores:  
  Uma imagem com texto, captura de ecrã, software, Tipo de letra

  Descrição gerada automaticamente
* Página de Registo de Utilizadores:  
  
* Página de Listagem de Medicamentos no Utilizador:  
  
* Página para edição de um medicamento:  
  Uma imagem com texto, captura de ecrã, Tipo de letra

  Descrição gerada automaticamente

# Conclusão

O trabalho, mesmo não ficando com as funções do CRUD completas, foi desafiante, ainda mais em ASP .NET Core, tecnologia que foi iniciante para mim neste ano. Igualmente o modelo MVC mostrou ser um bocado desafiante para mim.

Porém, foi completo para entender melhor o modelo MVC e as suas propriedades, aprofundar algumas habilidades com o ASP .NET Core, e também no próprio Visual Studio.

De igual forma, também consegui explorar as ferramentas do Git/GitHub, e do GitHub Desktop, todas elas usadas em força neste projeto para logging de alterações e análise de commits.