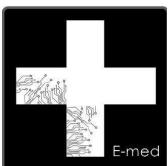
	<i>Instituto Federal de São Paulo – Campus Campinas</i> <i>Análise e Desenvolvimento de Sistemas – 4º Semestre/2021</i>		
<b>Trabalho de Arquitetura de Software</b>			
<b>Documento de Arquitetura</b>			
<b>Código/Sigla:</b>		<b>Nome do Projeto:</b>	<b>E-Med</b>

# E-Med

## Documento de Arquitetura

### 4ºSemestre/2021

**Disciplina: Arquitetura de Software**  
**Prof. Bianca Pedrosa**  
**Caio Barrio**  
**Thiago De Meneses Costa**

# Índice Analítico

1. Introdução	3
2. Metas e Restrições da Arquitetura	3
3. Modelo Arquitetural	3
4. Visão de Casos de Uso	4
5. Visão Lógica	5
5.1 Pacotes	7
6. Visão da Implementação	8
6.1 Padrões de projeto	8
7. Visão de Dados	9
8. Referências Bibliográficas	9

# Documento de Arquitetura de Software

## 1. Introdução

O intuito deste documento é apresentar como o aplicativo “E-Med” foi refatorado, usando os padrões de projeto de software, desde os padrões criacionais, estruturais e comportamentais, visando tornar o projeto mais compreensível para outros desenvolvedores e aumentar a flexibilidade e a reutilização do código existente. O documento está organizado em seções para uma melhor compreensão do leitor, começa falando das metas e restrições da arquitetura do projeto, logo abaixo é exposto um modelo da arquitetura usada no sistema que segue de seções representando as funcionalidades centrais, relevantes e significativas do sistema final.

## 2. Metas e Restrições da Arquitetura

### 2.1. Metas

Tendo em vista como principal objetivo o desenvolvimento e a refatoração do código, foram definidas metas explícitas a serem atingidas de tal modo a deixar o programa como um todo bem mais “Amigável” e receptivo a futuras alterações e possíveis novas versões, sendo possível o entendimento por quaisquer outros desenvolvedores assíduos com o mínimo de experiência e compreensão de código.

Essas metas fixadas foram:

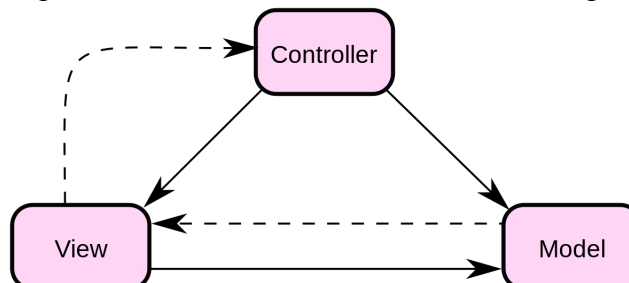
- 1- Inserção de padrões de software, durante a refatoração do aplicativo.
- 2- Trazer um código mais limpo, enxuto e coeso, seguindo os padrões SOLID e as diretrizes do Clean Code;
- 3 - Estabelecimento do padrão de arquitetura do software como MVC;

### 2.2. Restrições

O Sistema do usuário final do sistema deverá ter o JRE instalado, além de um sistema operacional Windows Vista SP2 ou superior.

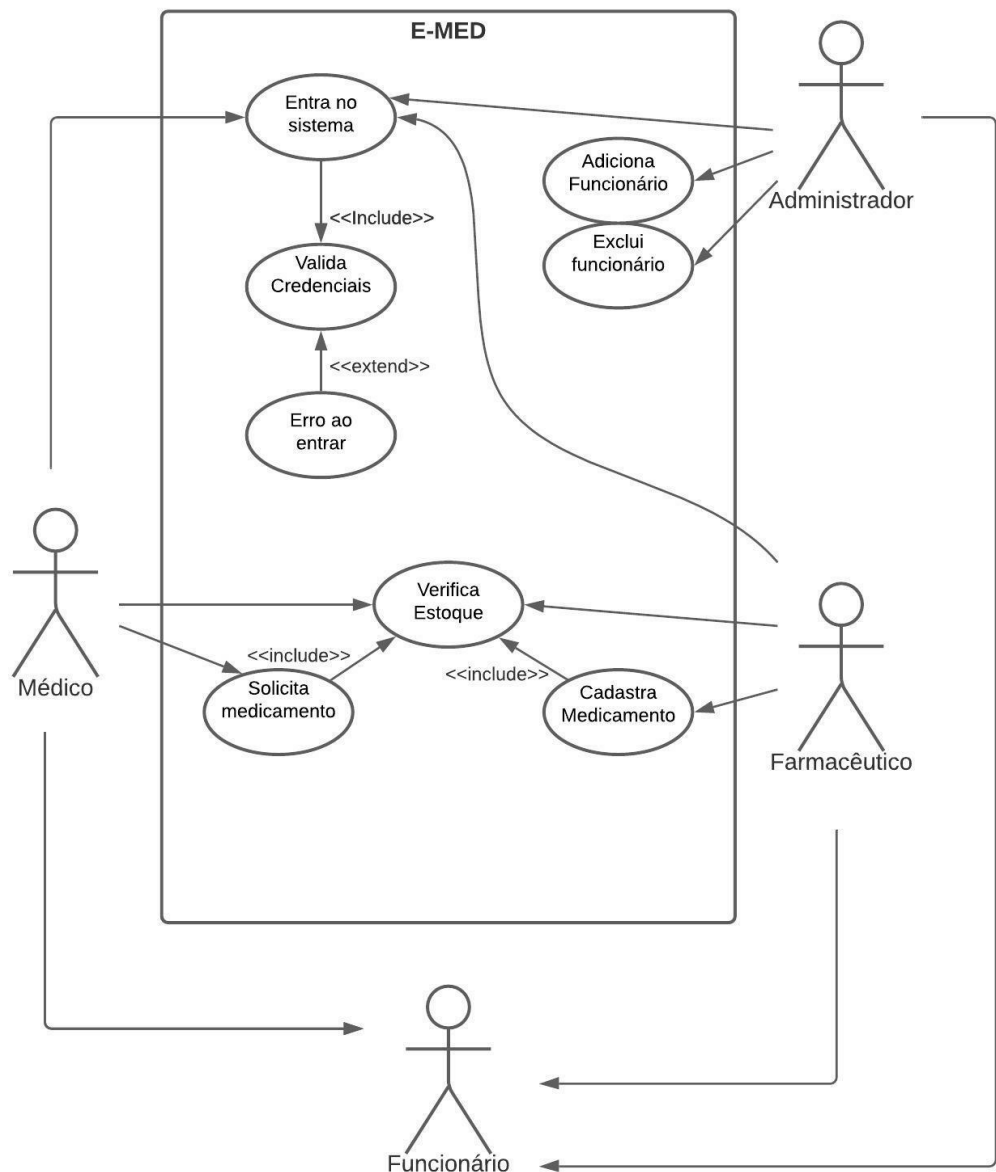
## 3. Modelo Arquitetural

A arquitetura usada no sistema é a Arquitetura MVC, onde uma de suas vantagens é a separação do código em camadas distintas de acordo com sua própria responsabilidade, essas camadas são: Visualização, Controle e Modelo de dados. Uma de suas desvantagens é a probabilidade de confundir códigos com responsabilidades diferentes da camada de negócio.



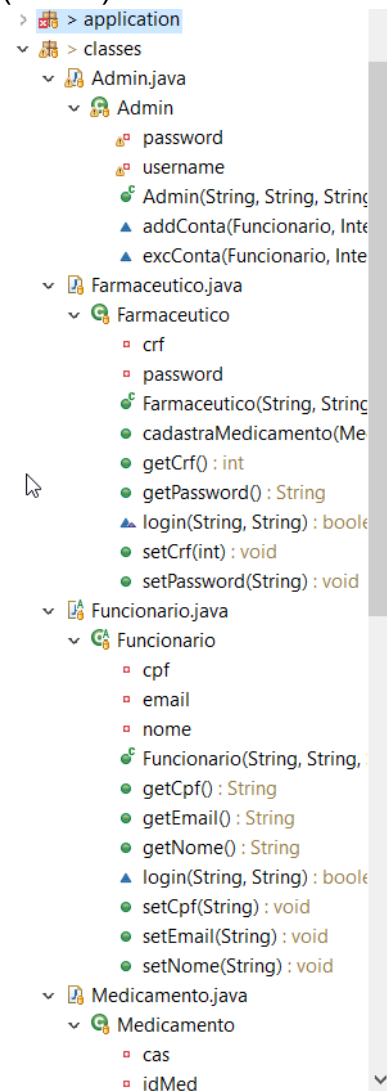
## 4. Visão de Casos de Uso

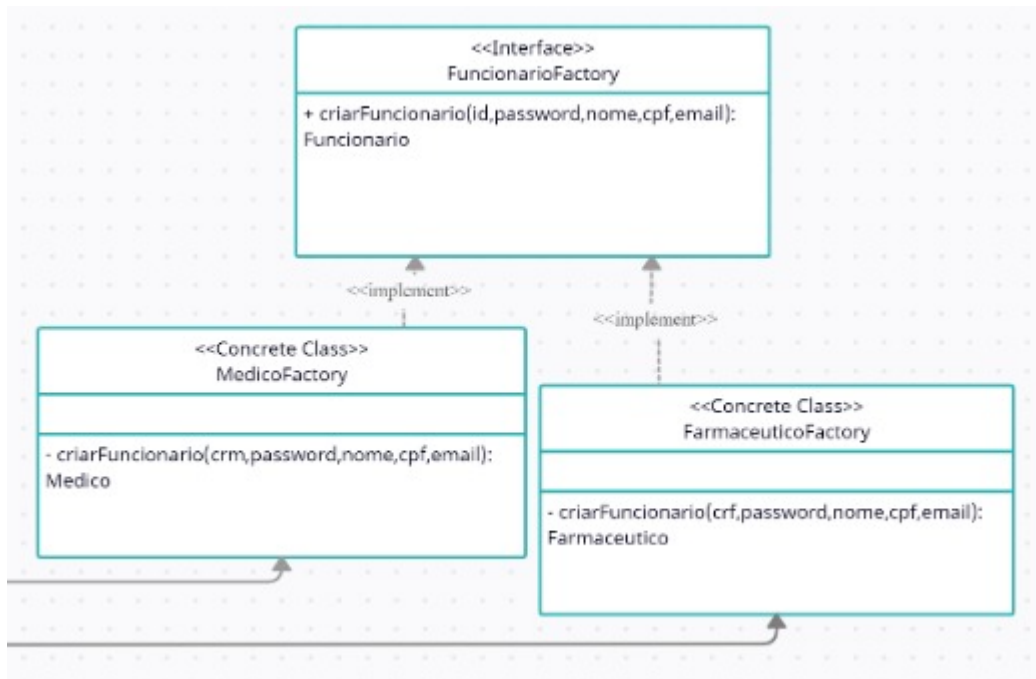
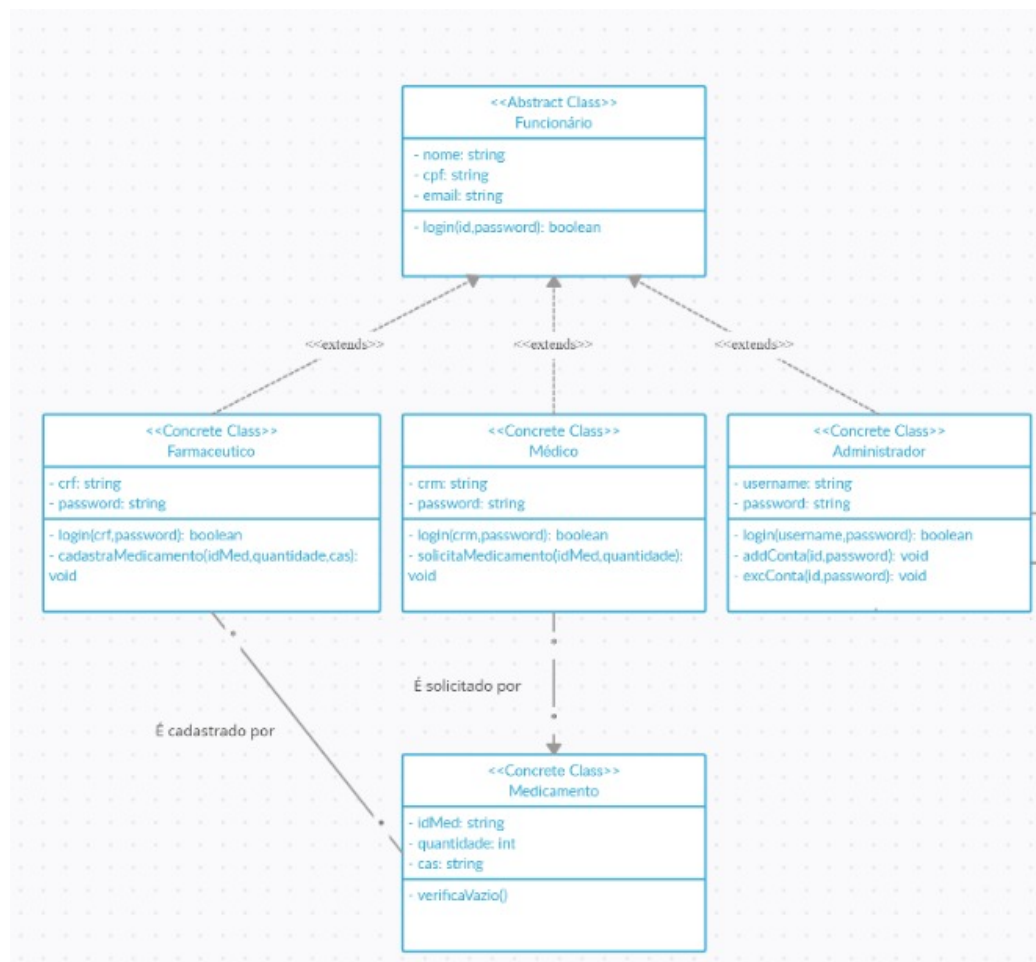
No diagrama de casos de uso, os atores são funcionários, estando divididos em: médico, farmacêutico e administrador, todos eles têm em comum pertencerem a classe funcionário e entrarem no sistema utilizando suas credenciais, as quais são validadas e autenticadas, em caso de erro o sistema retorna um aviso, para inserir corretamente as informações correspondentes, o farmacêutico tem a capacidade de cadastrar medicamentos no sistema, desde seu nome, e o número CAS, tanto quanto sua quantidade, o médico pode solicitar o pedido de algum medicamento em caso de necessidade e ambos os profissionais podem verificar o estoque de medicamentos disponíveis a todo momento no sistema, já o administrador, pode adicionar novos funcionários tanto quanto excluir os mesmos, sejam eles farmacêuticos ou médicos.



## 5. Visão Lógica

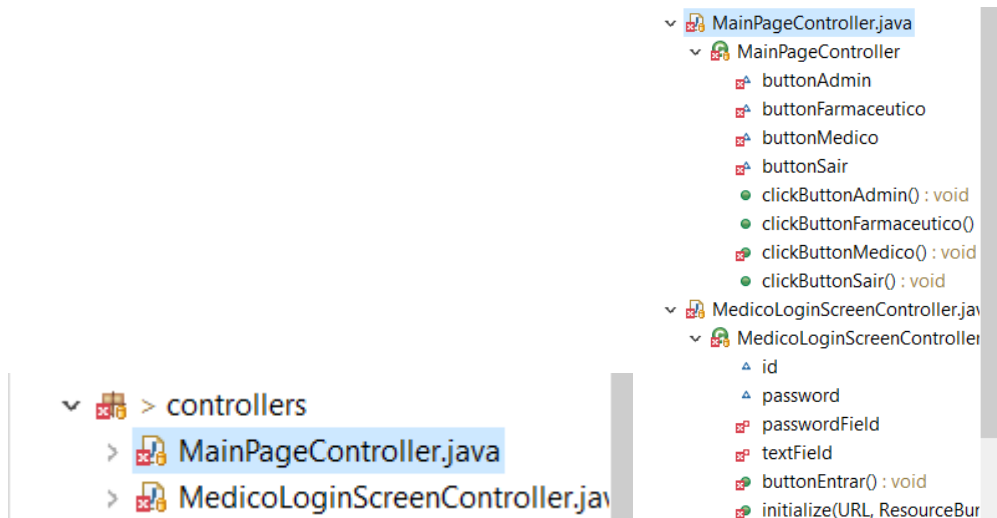
Basicamente o projeto está dividido em 3 pacotes principais que contemplam o padrão MVC - Model, View e Controller onde cada pacote principal representa uma camada desse padrão. O controller é responsável pelo envio de comandos a camada modelo e está inteiramente ligado à camada da visualização(view), visto que fornece todos comandos necessários para a troca entre a camada de visualização e interações com os componentes na tela, a camada Model toma a responsabilidade pela leitura e escrita dos dados, supre todos os recursos necessários para efetuar a comunicação com o banco de dados, seja eles para efetuar a inserção quanto para a atualização dos dados. E a View fica responsável por mostrar os dados na tela através da linguagem de marcação de interface (FXML).



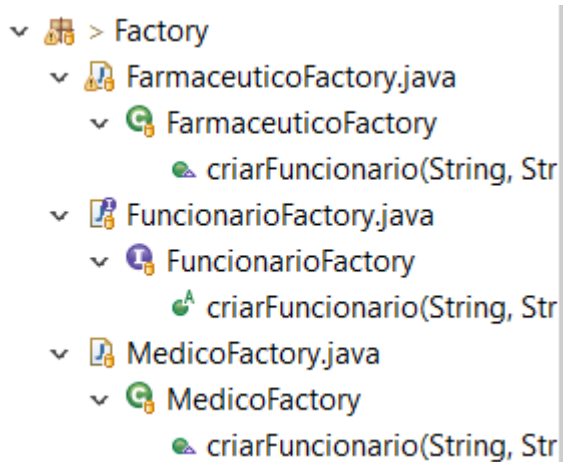


## 5.1 Pacotes

### 5.2 controllers

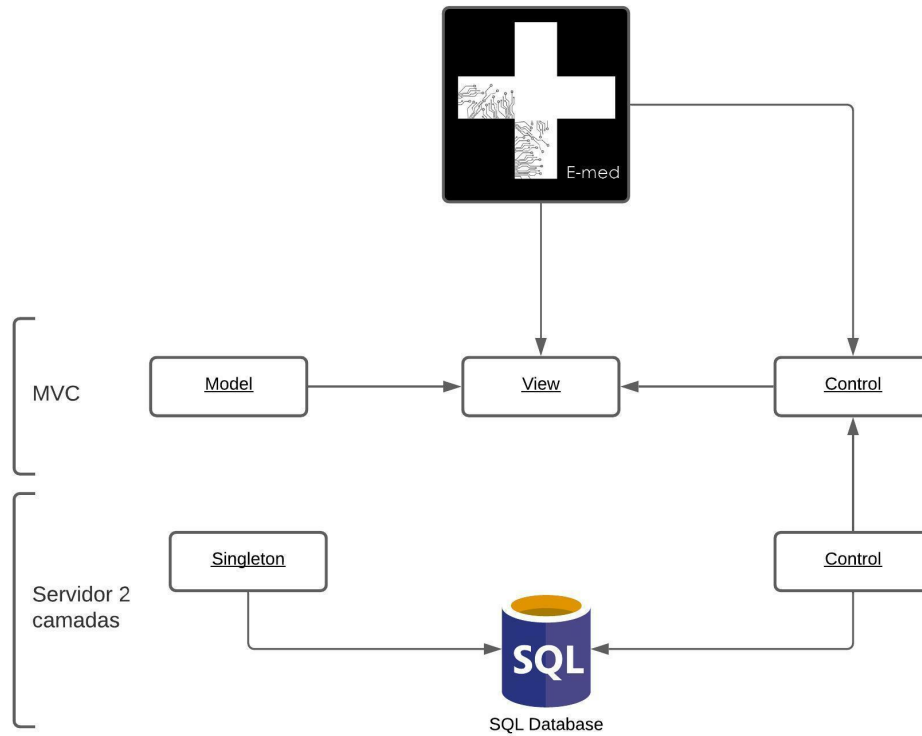


### 5.3 Factory



## 6. Visão da Implementação

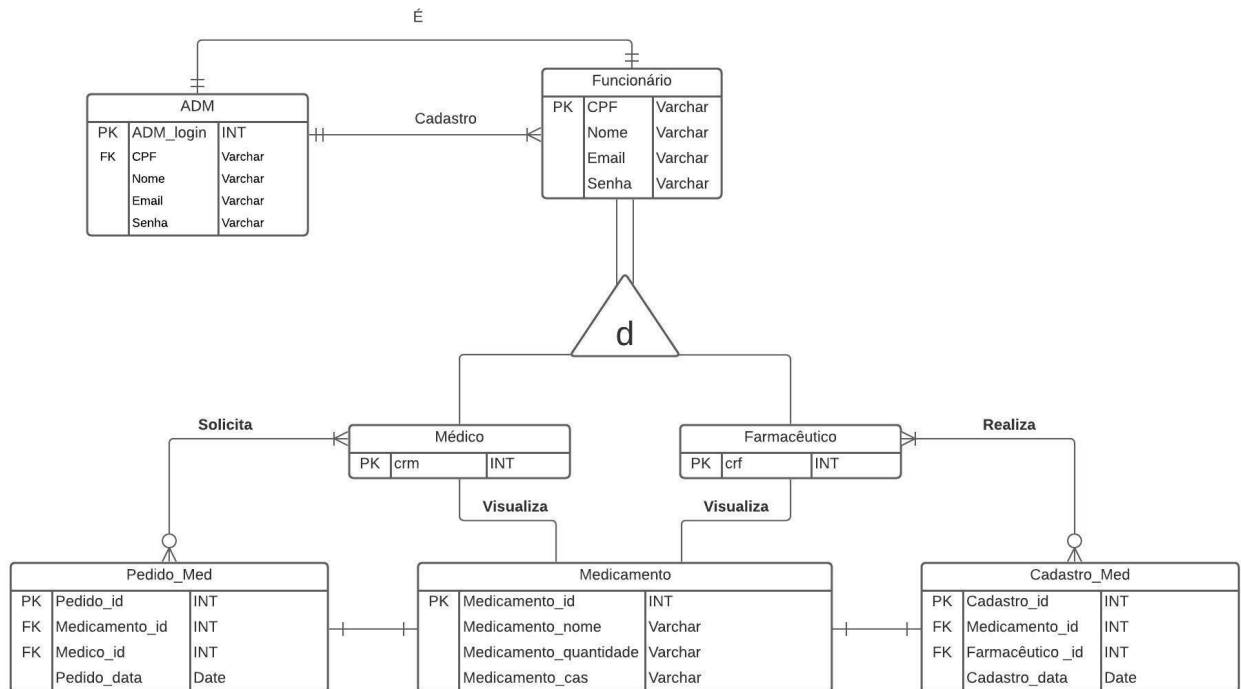
O programa se baseia principalmente no modelo de arquitetura de software: MVC (Model View Control) onde são implementados os comportamentos do sistema, além também, do servidor que conta com o Singleton para a persistência dos dados e o Controller.





## 7. Visão de Dados

Com base no diagrama entidade-relacionamento abaixo, o sistema foi desenvolvido.



## 8. Referências Bibliográficas

Padrão de Projeto Singleton em Java. Por Higor Em 2012. Disponível em :  
<<https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392>>. Acesso em  
02 de Julho de 2021.

Arquitetura de software. Organizador Giocondo Marino Gallotti. Bibliografia  
universitária Pearson, 1ª Edição, 2017.

Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. Head First Design Patterns.  
O'Reilly Media, 2004.