



UNIVERSIDADE  
FEDERAL DO CEARÁ

**DIAGRAMAS DE CLASSE**



**RESERVADO**

Responsável: Érica Miranda de Sousa	
Elaborador(es): Érica Miranda de Sousa	e-mail: ericams175@gmail.com

**HISTÓRICO**

<b>Data</b>	<b>Versão</b>	<b>Responsável</b>	<b>Alteração</b>
27/08/2020	01	Érica Miranda de Sousa	Criação dos primeiros diagramas de classe
12/09/2020	1.1	Érica Miranda de Sousa	Adaptando diagramas ao padrão de projeto <i>Strategy</i>

<b>Introdução</b>	-----	<b>5</b>
<b>Padrão de Projeto escolhido</b>	-----	<b>5</b>
<b>Diagramas de Classe</b>	-----	<b>6</b>

# ESPECIFICAÇÃO DE REQUISITOS

## 1. Introdução

### a. Objetivos

Este documento busca fornecer uma visão abstrata da estrutura e relações das classes que compõem o sistema I9life, definindo todas as classes que o sistema necessita possuir e é a base para a construção do sistema.

Este documento não está completo, portanto ainda sofrerá mudanças no decorrer do projeto.

### b. Público alvo deste documento

Time de desenvolvimento do sistema I9Life.

### c. Links

- Os diagramas foram feitos na ferramenta **Draw.io**. Link para o projeto no Draw.io: [clique aqui](#);
- Os diagramas estão armazenados no GoogleDrive. Link para a pasta com as imagens no GoogleDrive: [clique aqui](#).
- O documento de Diagramas de Classe está armazenado no GoogleDrive. Link para o documento no GoogleDrive: [clique aqui](#).

### d. Observações

Este documento não contém ainda a representação das entidades **Agenda** e **Prontuário** para os módulos *view* e *controller*, pois o time ainda não conseguiu planejar seus detalhes de implementação. Espera-se que os mesmos já possam ser adicionados na próxima versão deste documento

## 2. Padrão de Projeto escolhido

Para este sistema, foi escolhido o padrão *MVC (Model-View-Controller)*. Esta escolha se dá por o mesmo ser um padrão simples, ideal para sistemas que usam a Orientação a Objetos. Por este motivo, os seguintes diagramas estão organizados de acordo com o padrão *MVC*, seguindo suas camadas: *Model*, *view*, *controller*.

Também será utilizado o *design pattern Strategy*.

### 3. Diagramas de Classe

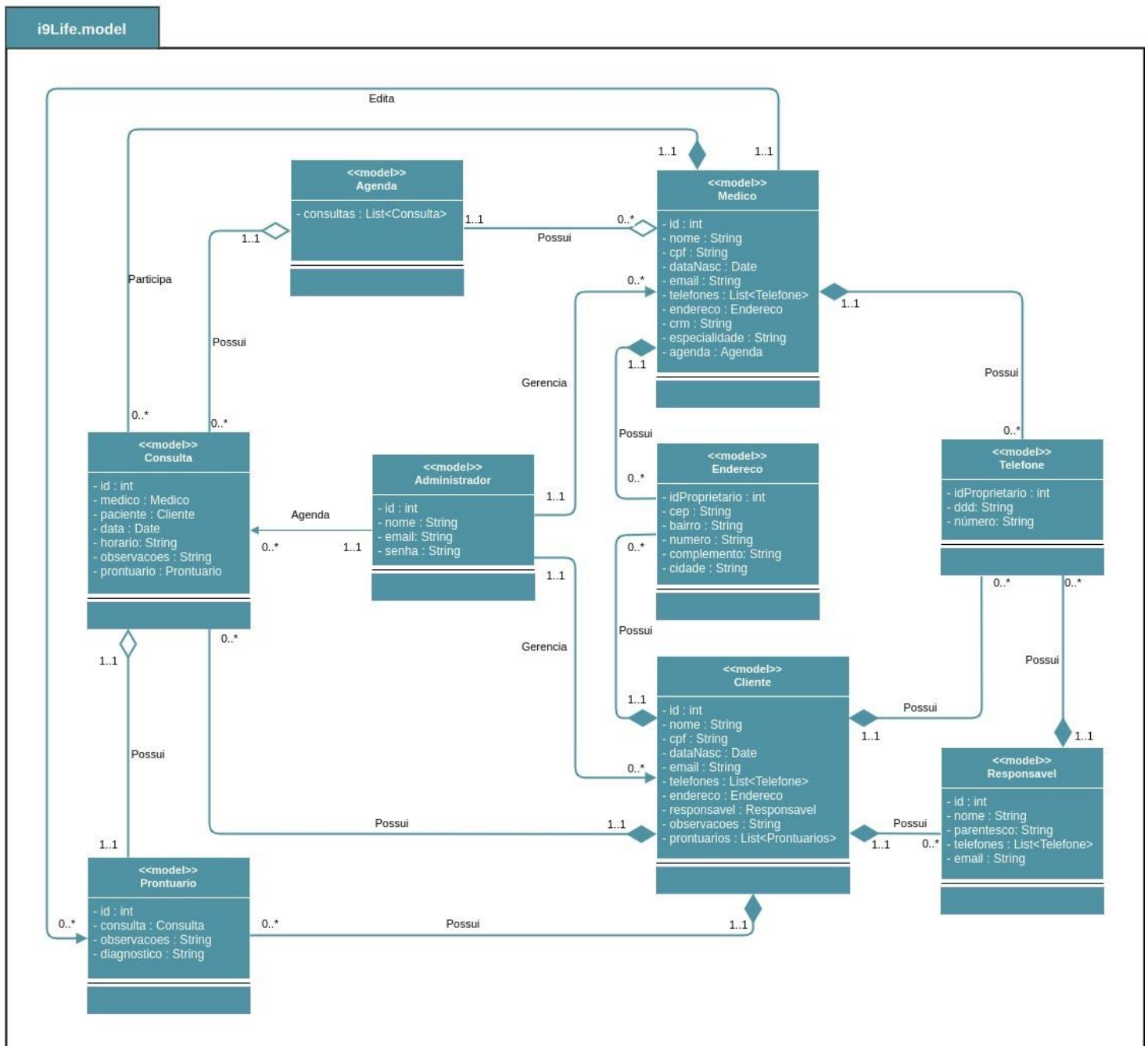
Nas seguintes seções está representado os diagramas de classe do sistema I9Life.

#### 3.1 Camada *Model*

A camada *Model* contém todas as entidades que compõem o sistema, assim com atributos e métodos. Nesta representação, decidiu-se omitir os métodos *get*, *set* e *toString*, pois os mesmos são componentes padrão da linguagem Java e sua ausência melhora a visualização do diagrama.

I9Life - Class Diagram

EricaSouza | August 27, 2020



[Clique aqui](#) para visualizar a imagem em maior qualidade.

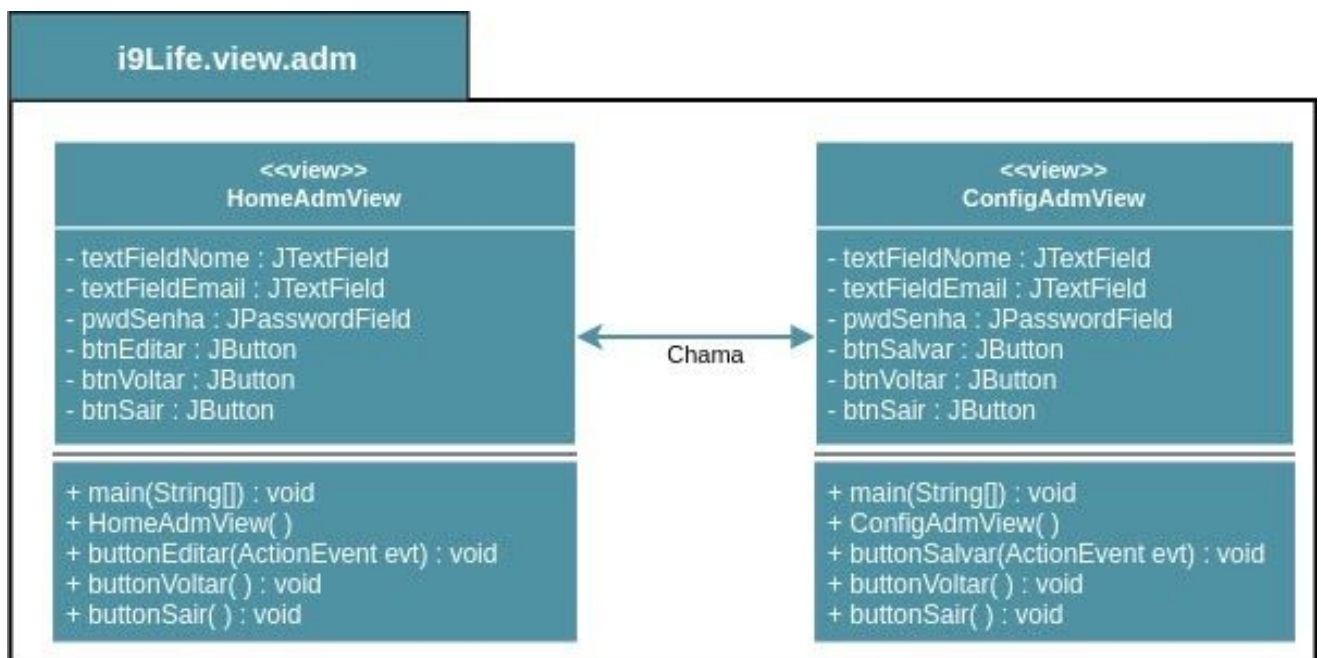
## 3.2 Camada View

A camada *View* contém todas as classes de fronteira do sistema, que fazem a comunicação com os atores externos. Neste caso específico, temos as classes que compõem as telas que compõem o sistema.

Para uma melhor visualização e entendimento, mostraremos os diagramas para cada módulo do sistema.

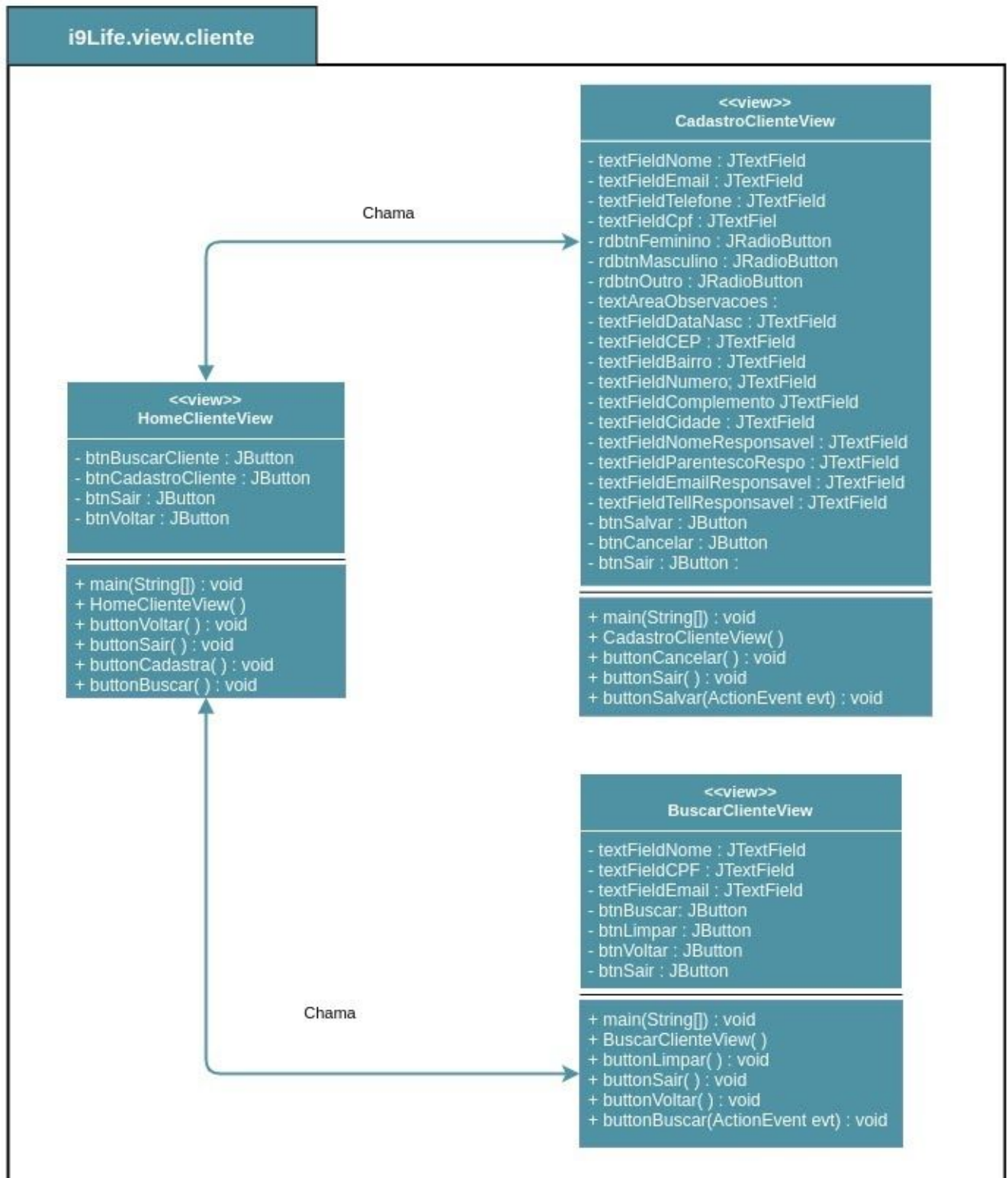
Salienta-se ainda, que as cardinalidade foram omitidas na camada *view*, já que uma tela pode *chamar* no mínimo e no máximo uma outra tela.

### - Módulo Administrador : view.adm



[Clique aqui](#) para visualizar a imagem em maior qualidade.

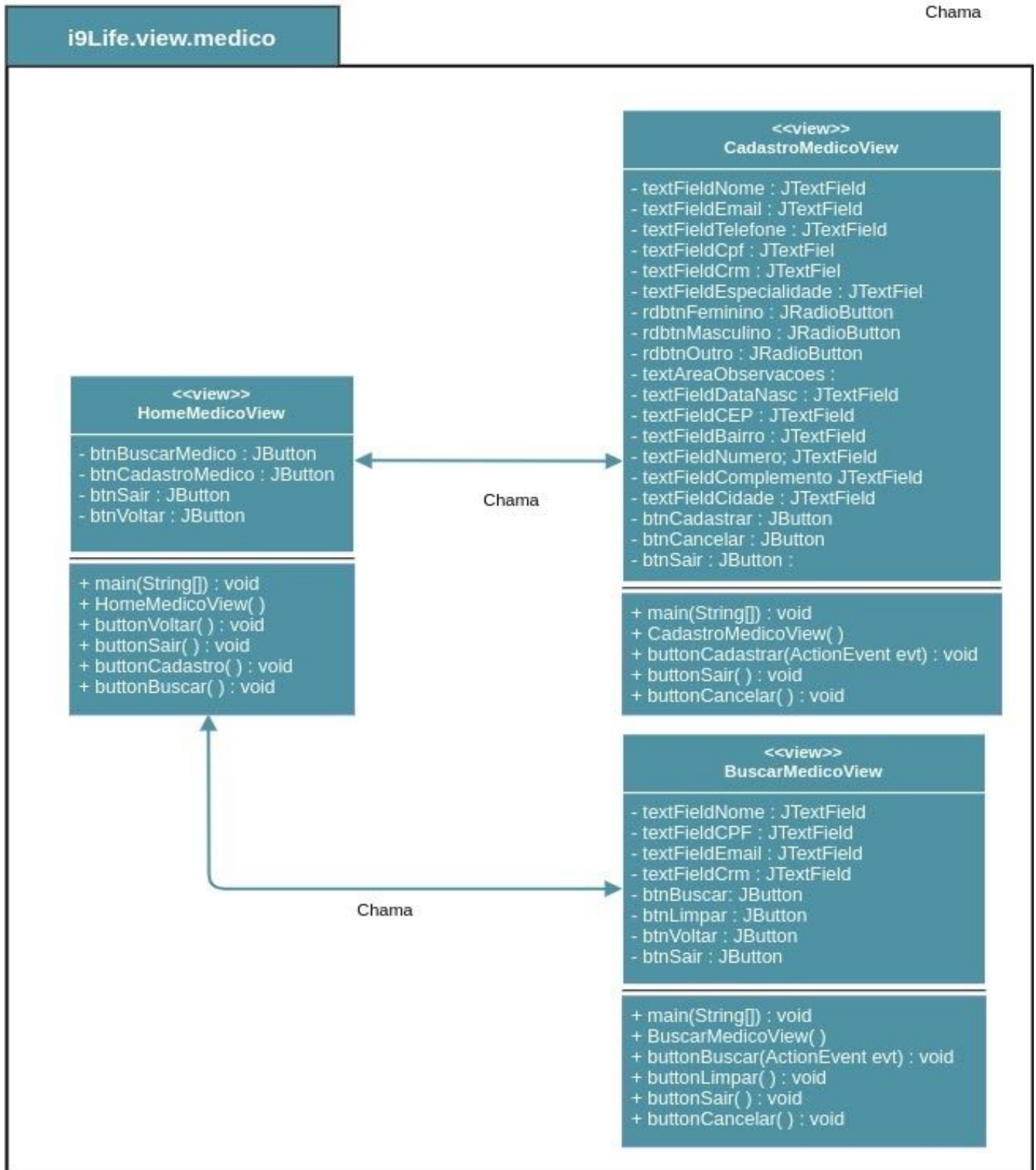
- Módulo Cliente : view.cliente



[Clique aqui](#) para visualizar a imagem em maior qualidade.

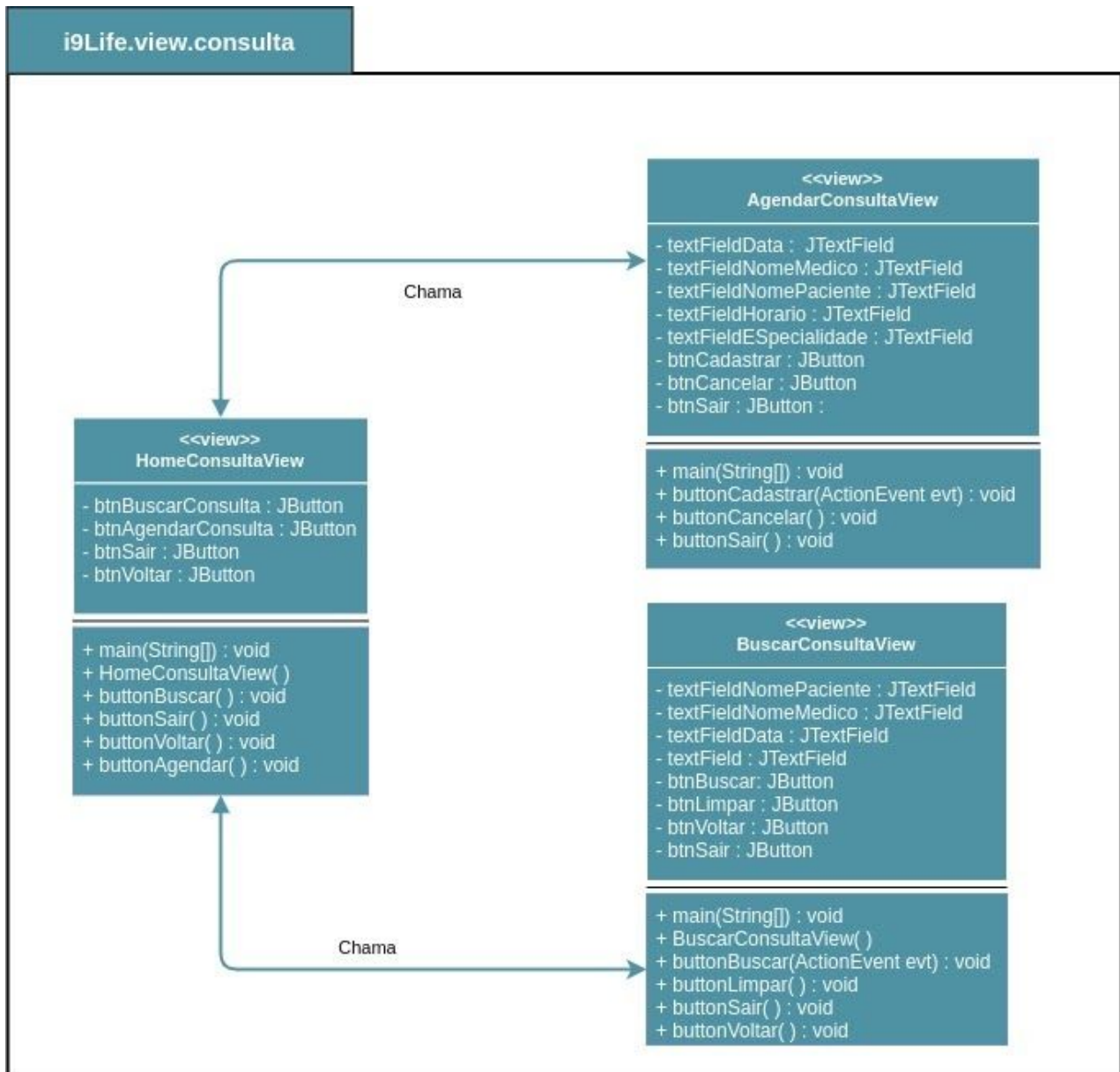


## - Módulo Médico : view.medico



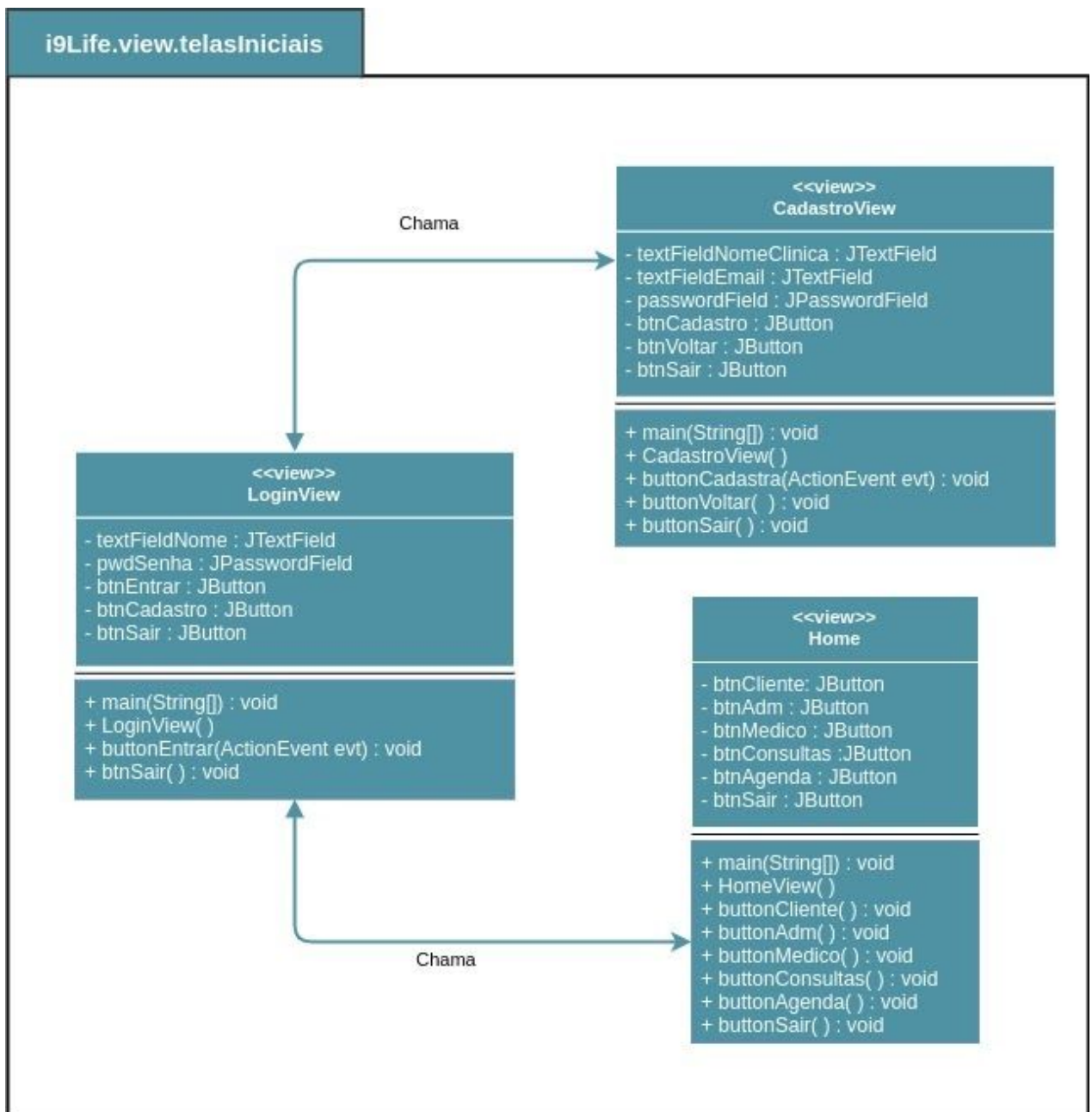
[Clique aqui](#) para visualizar a imagem em maior qualidade.

- **Módulo Consulta : view.consulta**



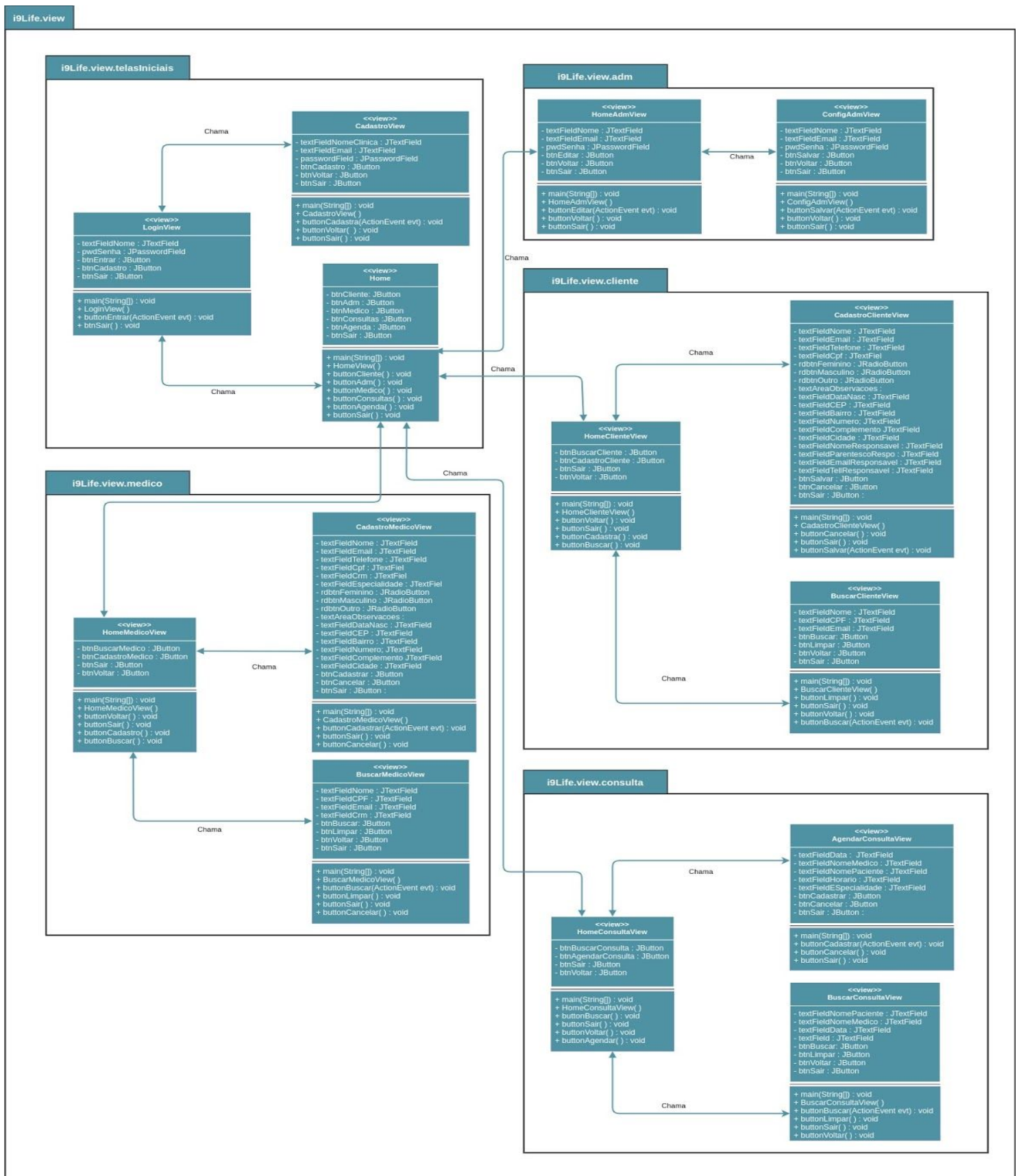
[Clique aqui](#) para visualizar a imagem em maior qualidade.

- **Módulo Telas Iniciais (login e cadastro de Administrador) :**  
**view.telasIniciais**



[Clique aqui](#) para visualizar a imagem em maior qualidade.

## - Camada View completa

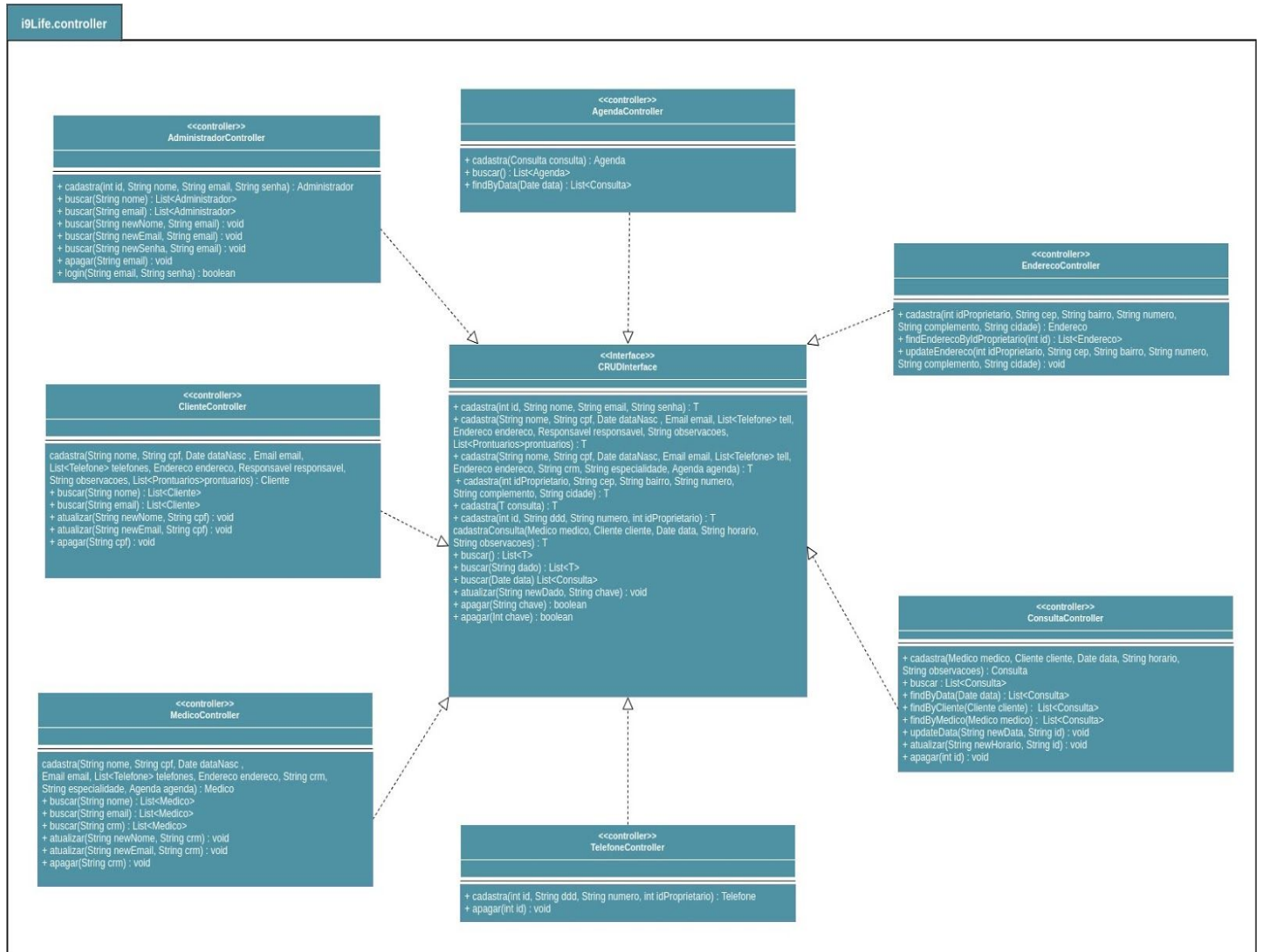


[Clique aqui](#) para visualizar a imagem em maior qualidade.



### 3.3 Camada Controller

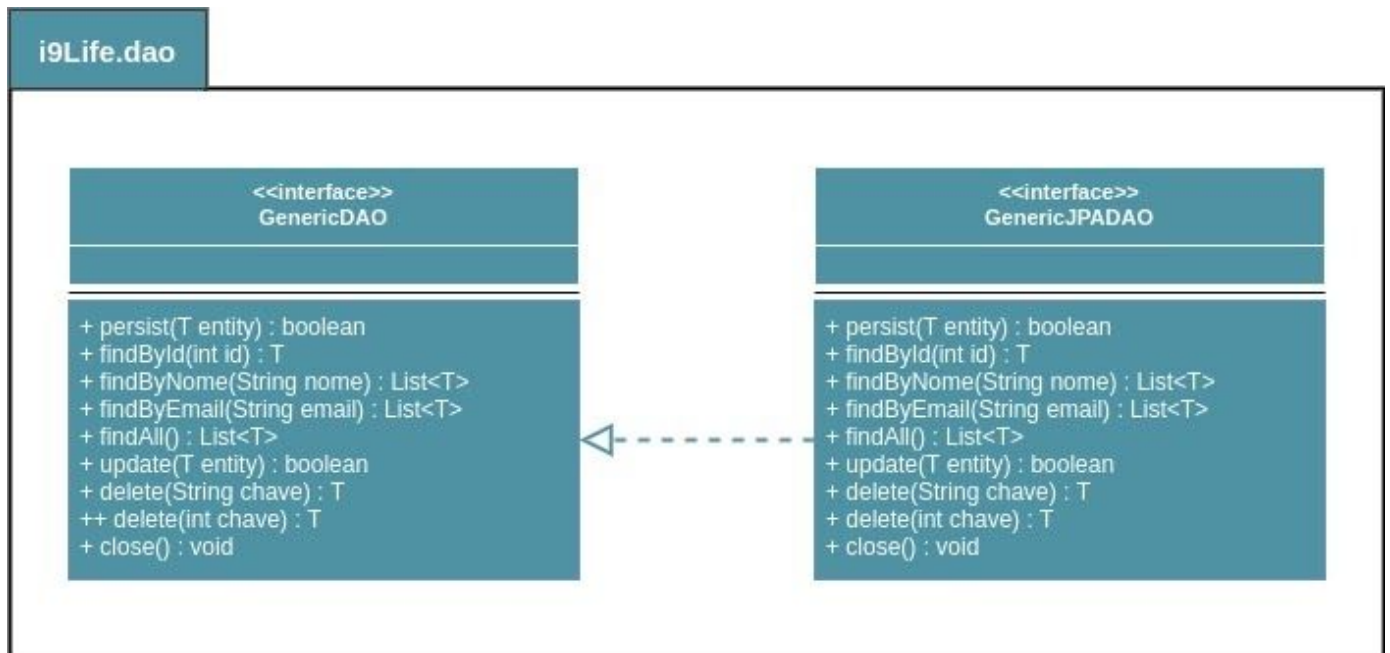
O responsável por receber todas as requisições do usuário. Seus métodos chamados actions são responsáveis por uma página, controlando qual *model* usar e qual view será mostrado ao usuário.



[Clique aqui](#) para visualizar a imagem em maior qualidade.

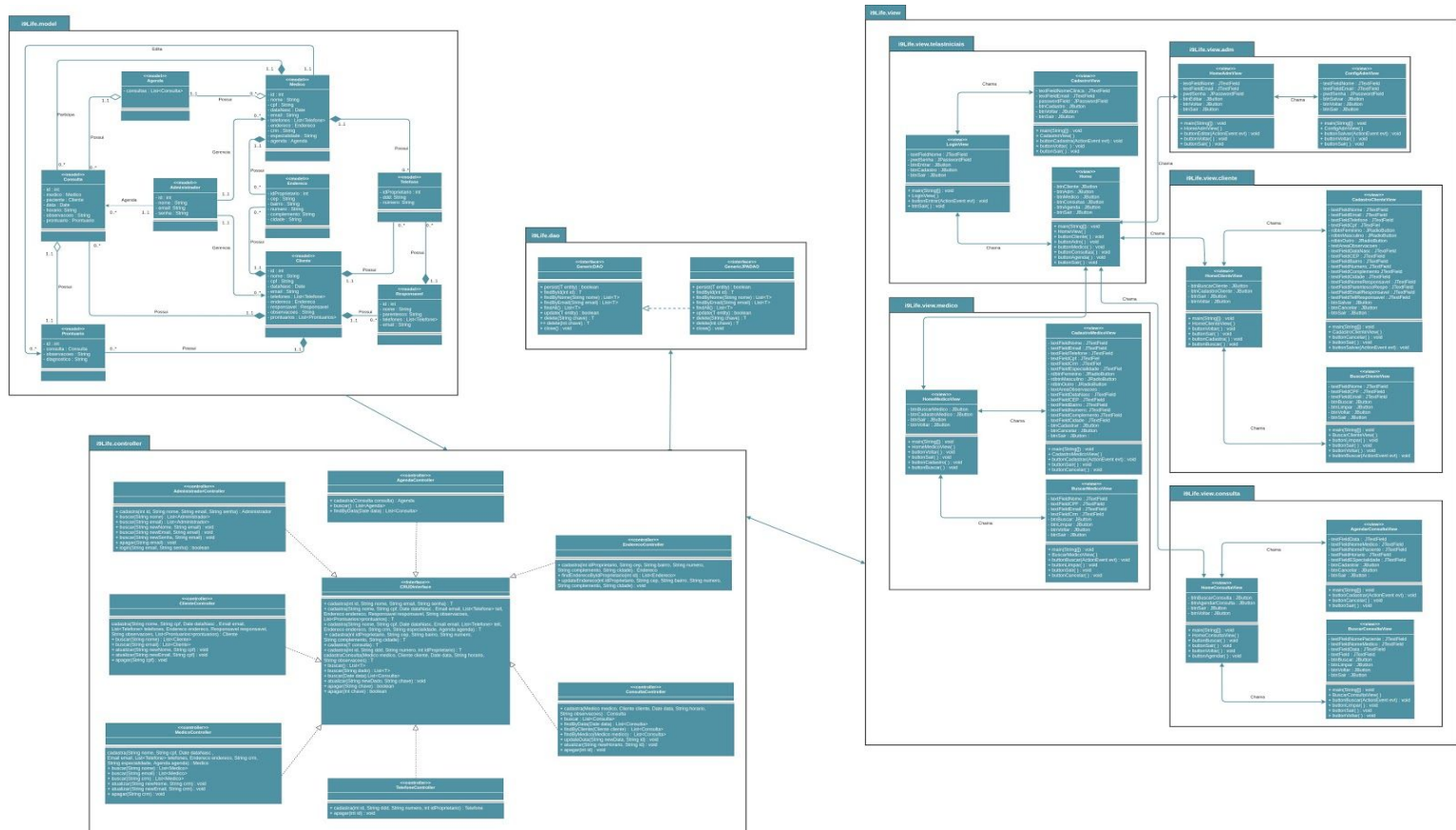
### 3.4 Camada *DAO*

A camada *DAO* é um padrão de projetos onde um objeto provê uma interface que abstrai o acesso a dados, lê e grava a partir da origem de dados (banco de dados, arquivo, memória, etc.); e encapsula o acesso aos dados, de forma que as demais classes não precisam saber sobre isso.



[Clique aqui](#) para visualizar a imagem em maior qualidade.

### 3.5 Visão geral do sistema



[Clique aqui](#) para visualizar a imagem em maior qualidade.