



**Universidade Estadual de Maringá**

**Departamento de Informática**

**Curso:** Informática

**Disciplina:** Programação de Sistemas Web – 5188

**Professora:** Juliana Keiko Yamaguchi

## **Programação de Sistemas WEB**

### **Trabalho Prático: Parte I**

Integrante	RA
Douglas Mezuraro	95676
Tatiane Lie Takeshima	87971
Victor Glauber L. Silva	68474

## **Sumário**

[Domínio do sistema](#)

[Entidades do sistema](#)

[Diagrama de classes](#)

[Arquitetura](#)

[Diagrama de pacotes](#)

## **Domínio do sistema**

Este documento especifica os requisitos de um projeto de software para uma clínica médica, fornecendo as informações necessárias para o desenvolvimento do mesmo.

A necessidade de criação do sistema surgiu pois os médicos da clínica não terem controle do fluxo dos pacientes que atendem e consequentemente quando precisam recuperar informações sobre os últimos atendimentos despendem muito tempo procurando nos prontuários físicos de atendimento. Nem sempre a secretária consegue se lembrar de confirmar as consultas com os pacientes com um dia de antecedência, sendo que muitos deles esquecem e os médicos ficam com tempo não preenchido em seus consultório.

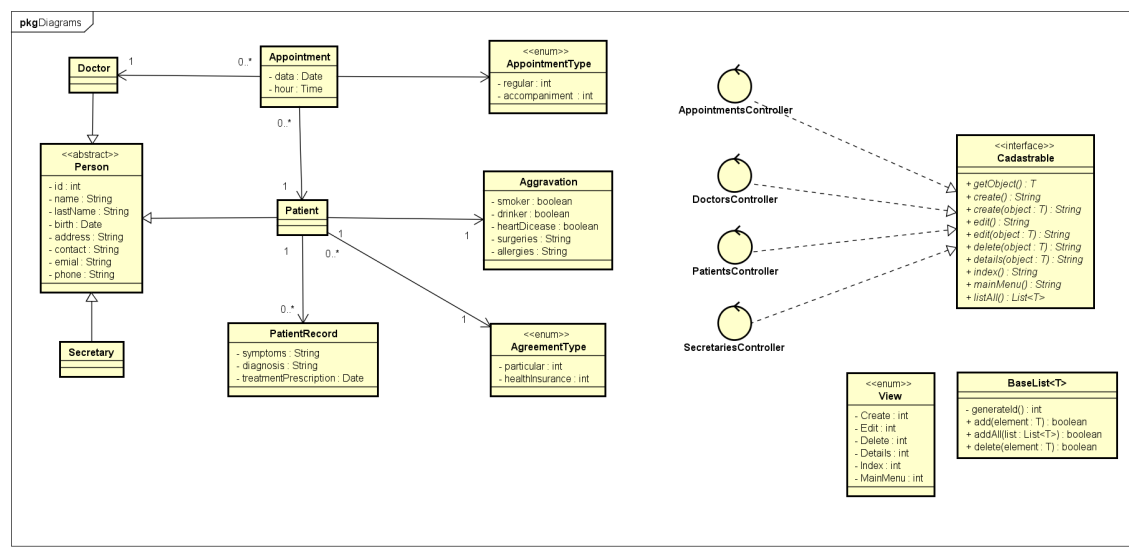
A clínica em questão, por ser de pequeno porte, não possui grande quantidade de equipamentos tecnológicos, apenas dois computadores pessoais, uma utilizado por um usuário que trabalha na recepção e um computador pessoal para cada médico da clínica, todos com acesso a internet.

## Entidades do sistema

Tipo	Nome	Objetivo
Pacote: br.uem.din.medicalclinic.model		
Entidade	Aggravation	Indicar os agravantes de riscos que um paciente possui, como, se o paciente fuma, bebe, possui doenças cardíacas, alergias que possui e cirurgias já realizadas
Enumerado	AgreementType	Indicar o tipos de convênio que um paciente pode ter, sendo eles, particular e plano de saúde
Entidade	Appointment	Indicar as informações sobre a consulta, como, o médico que irá consultar, o paciente a ser consultado, a data da consulta e o tipo da consulta
Enumerado	AppointmentType	Indicar o tipo que uma consulta pode ser, sendo elas, normal e de retorno
Classe	Base	Classe de qual toda entidade cadastrável do sistema herda, possui o atributo id que é o indicador único de cada registro pelo qual é usado para recuperar o mesmo
Entidade	Doctor	Indicar as informações pessoais de um médico (para mais detalhes ver classe <a href="#">Person</a> )
Entidade	Patient	Indicar as informações pessoais de um paciente (para mais detalhes ver classe <a href="#">Person</a> )
Entidade	PatientRecord	Indicar o prontuário de uma consulta de um paciente, como os sintomas apresentados, o diagnóstico feito pelo médico e a prescrição do tratamento
Entidade	Person	Indicar os dados pessoais em que todas pessoas do sistema tem em comum, como, nome, último nome, aniversário, endereço, nome da pessoa para contato, endereço de e-mail e número de telefone
Entidade	Secretary	Indicar as informações pessoais de um secretário (para mais detalhes ver classe <a href="#">Person</a> )

Pacote: br.uem.din.medicalclinic.controller		
Controlador	AppointmentsController	Controlador da entidade <a href="#">Appointment</a>
Controlador	DoctorsController	Controlador da entidade <a href="#">Doctor</a>
Controlador	PatientsController	Controlador da entidade <a href="#">Patient</a>
Controlador	SecretariesController	Controlador da entidade <a href="#">Secretary</a>
Pacote: br.uem.din.medicalclinic.utils		
Classe	BaseList	Classe que implementa representa uma lista de objetos que herdam da classe <a href="#">Base</a> onde é possível fazer as operações de CRUD
Enumerado	View	Enumerado que contém os nomes das páginas web

Diagrama de classes



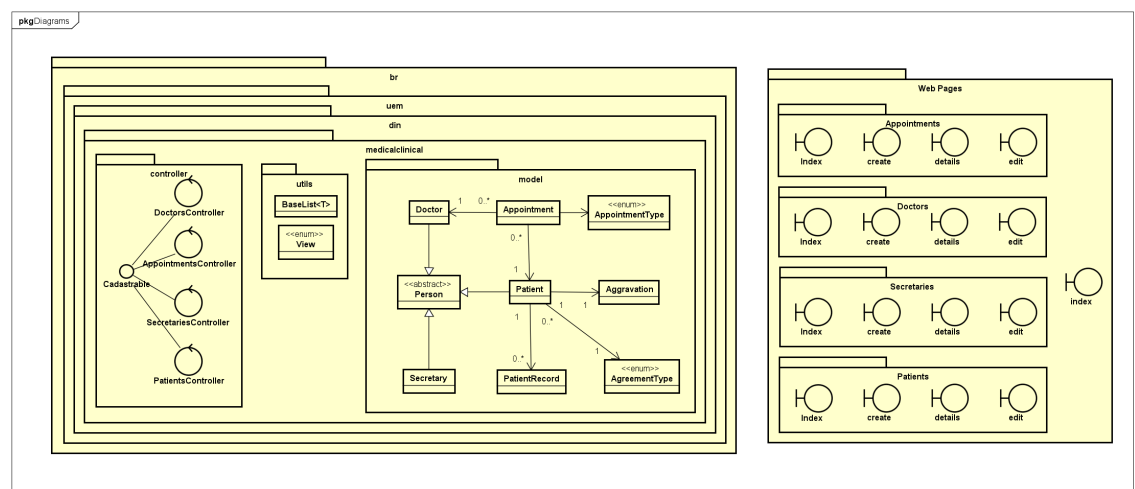
Dentre as classes descritas acima temos a tabela que indica quais entidades foram implementadas o CRUD:

Entidade	Cadastrar	Editar	Excluir	Detalhar
Aggravation	Não	Não	Não	Não
Appointment	Sim	Sim	Sim	Sim
Doctor	Sim	Sim	Sim	Sim
Patient	Sim	Sim	Sim	Sim
PatientRecord	Não	Não	Não	Não
Secretary	Sim	Sim	Sim	Sim

Arquitetura

Utilizado o padrão MVC (*model-view-controller*) baseado em cliente-servidor.

Diagrama de pacotes



O sistema foi organizado em três pacotes principais seguindo o MVC:

Model	Pacote onde estão todas as classes entidades levantadas com a análise de requisitos. As entidades são classes são abstrações de algo real na clínica médica. É também a classe que possuem as informações que irão ser persistidas
View	Pacote onde estão todas as visões (páginas HTMLs), as views estão organizadas em pastas com o nome da entidade que a representa. As visões são responsáveis pela exibição de dados para o usuário. Atualmente no sistema existe uma view para cada operação de CRUD de cada entidade
Controller	Pacote onde estão os controladores das entidades, o controlador na aplicação tem o papel de receber a informação e ação da view e a partir disto controlar e atualizar o modelo. O controlador também devolve para o browser qual visão irá ser exibida de acordo com a ação executada pelo usuário.