

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA  
E TECNOLOGIA DO RIO GRANDE DO NORTE - CNAT**

**DIRETORIA: Diretoria Acadêmica de Gestão e Tecnologia da Informação (DIATINF)**

**DISCIPLINA: Programação Orientada a Objetos - POO - 2023.2**

**DOCENTE: Gilbert Azevedo da Silva**

**ALUNO: Felipe da Costa Ferreira**

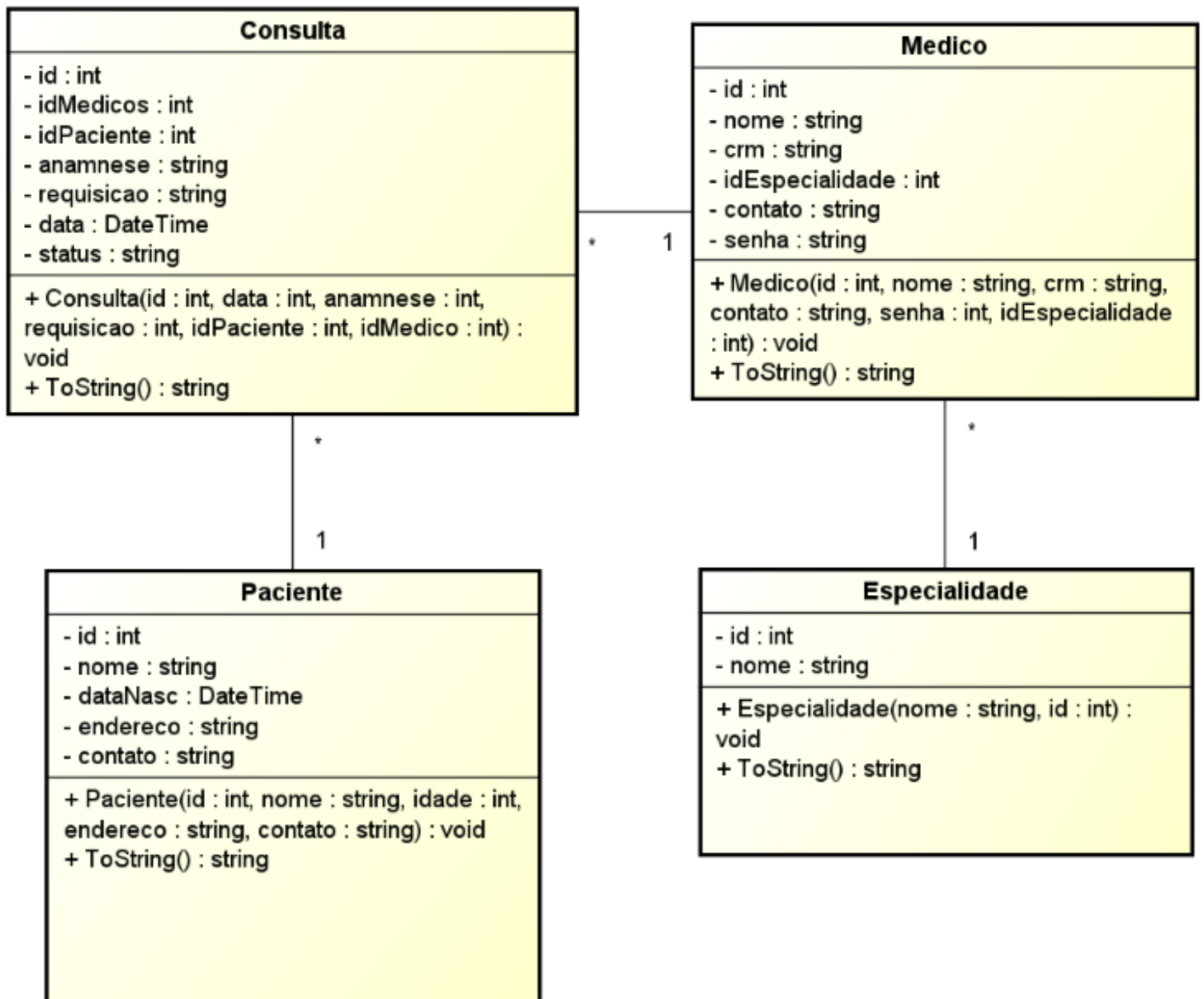
**Modelagem Final de Sistema  
Clínica Médica**

## 1. Descrição do sistema

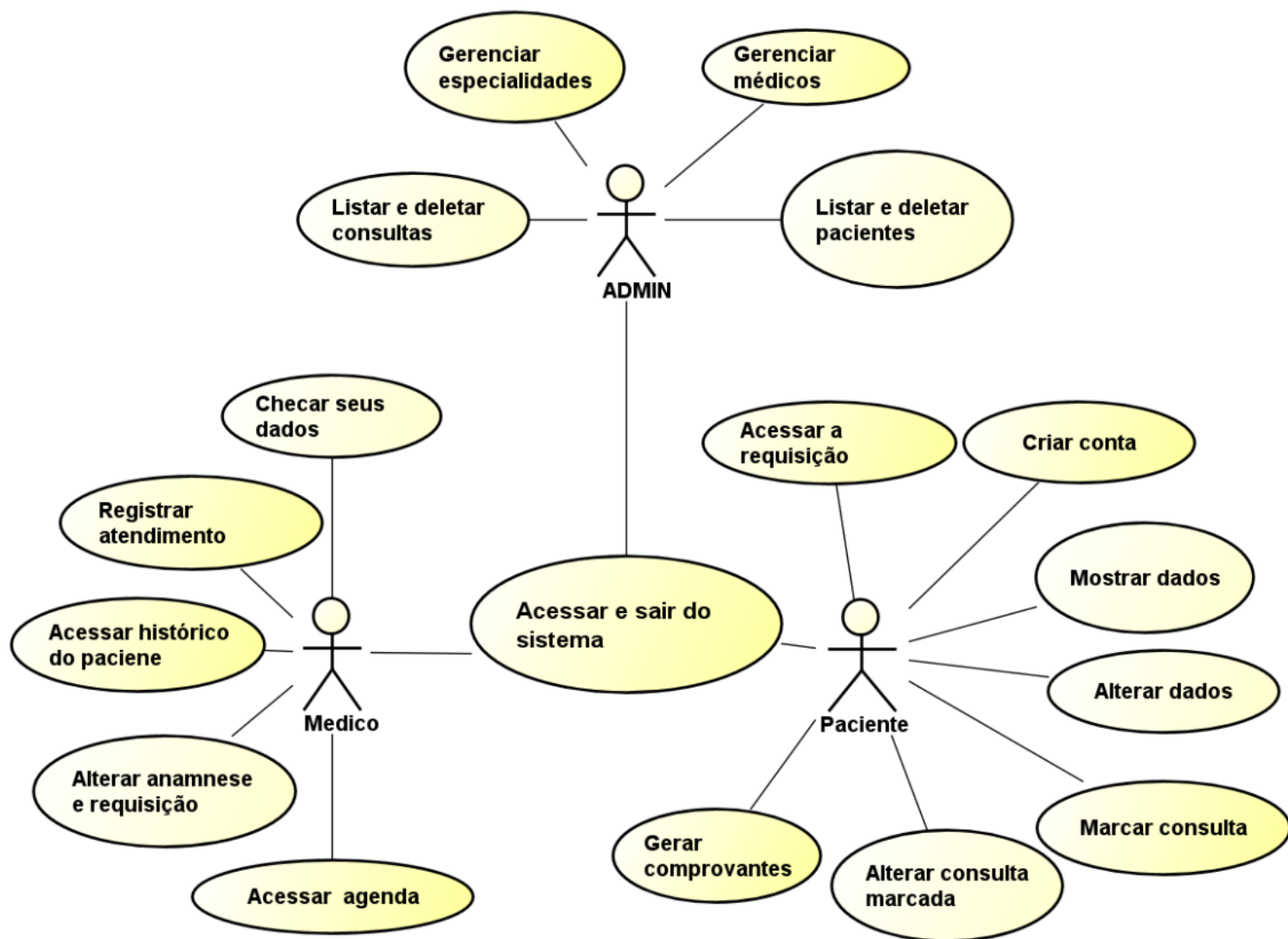
Uma clínica médica solicitou o desenvolvimento de um protótipo de sistema para melhorar o atendimento fornecido. A proposta inicial é que o software permita que um administrador possa gerenciar especialidades, médicos, pacientes e consultas. Os profissionais poderão checar sua agenda do dia escolhido, iniciar a consulta, gerenciar o histórico do paciente e checar suas informações. Já os pacientes podem se cadastrar, acessar o sistema, marcar uma consulta (emitindo o comprovante) e ter acesso a requisição passada pelo médico.

Abaixo está a modelagem desse sistema proposto pelo médico.

## 2. Diagrama de classes - Entidades



### 3. Diagrama de classes - Casos de Uso



#### 4. Diagrama de classes - Operações

<<static>>

View

```
+ PacienteInserir(nomeP : string, nascP : DateTime, enderecoP : string,
contatoP : string, senhaP : int) : void
+ ListarPacientes() : List<Paciente>
+ PacienteExcluir(id : int) : void
+ PacienteAtualizar(id : int, nome : string, dataNasc : DateTime, endereco :
string, contato : string) : void
+ MedicoInserir(nome : string, crm : string, especialidade : string, convenio :
string) : void
+ MedicosListar() : List<Medico>
+ MedicoExcluir() : void
+ MedicoAtualizar(nome : string, crm : string, especialidade : string, convenio :
string, id : int) : void
+ EspecialidadeInserir(nome : string) : void
+ EspecialidadeListar() : List<Especialidade>
+ EspecialidadeAtualizar(nome : string, listaMed : List<Medico>, id : int) : void
+ EspecialidadeExcluir(id : int) : void
+ ConsultaInserir(data : DateTime, anamnese : string, requisicao : int) : void
+ ListarConsultas() : List<Consulta>
+ AtualizandoConsulta(idC : int) : void
+ ConsultaExcluir(idC : int) : void
+ LoginPaciente() : Paciente
+ MarcarConsulta(idPac : int, idMed : int, data : DateTime) : void
+ MedicosPorEspecialidade(int idE : int) : void
+ GerarComprovante(UserId : int, data : DateTime) : void
+ PegaUltimaRequisicao(idU : int) : void
+ PacienteAtualizar(id : int, nNome : string, data : DateTime, nEndereco :
string, nContato : int) : void
+ AlterandoConsultaMarcada(idC : int) : void
+ ListarMinhasConsultasMarcadas(idP : int) : void
+ ListarTodasAsMinhasConsultas(idP : int) : void
+ LoginMedico() : Medico
+ ConsultaIniciar(idMed : int, idP : int, dataHoje : DateTime) : void
+ Consulta(consulta : Consulta) : Consulta
+ HistoricoDoPaciente(idM : int) : void
+ AlterarConsulta(consulta : Consulta) : void
+ BuscarAgenda(idUser : int, dataHoje : DateTime) : List<Consulta>
+ TodosOsHistoricos() : List<Consulta>
+ ConsultaAtualizar(idC : int) : List<Consulta>
```

## 5. Diagrama de classes - Classes de Persistência

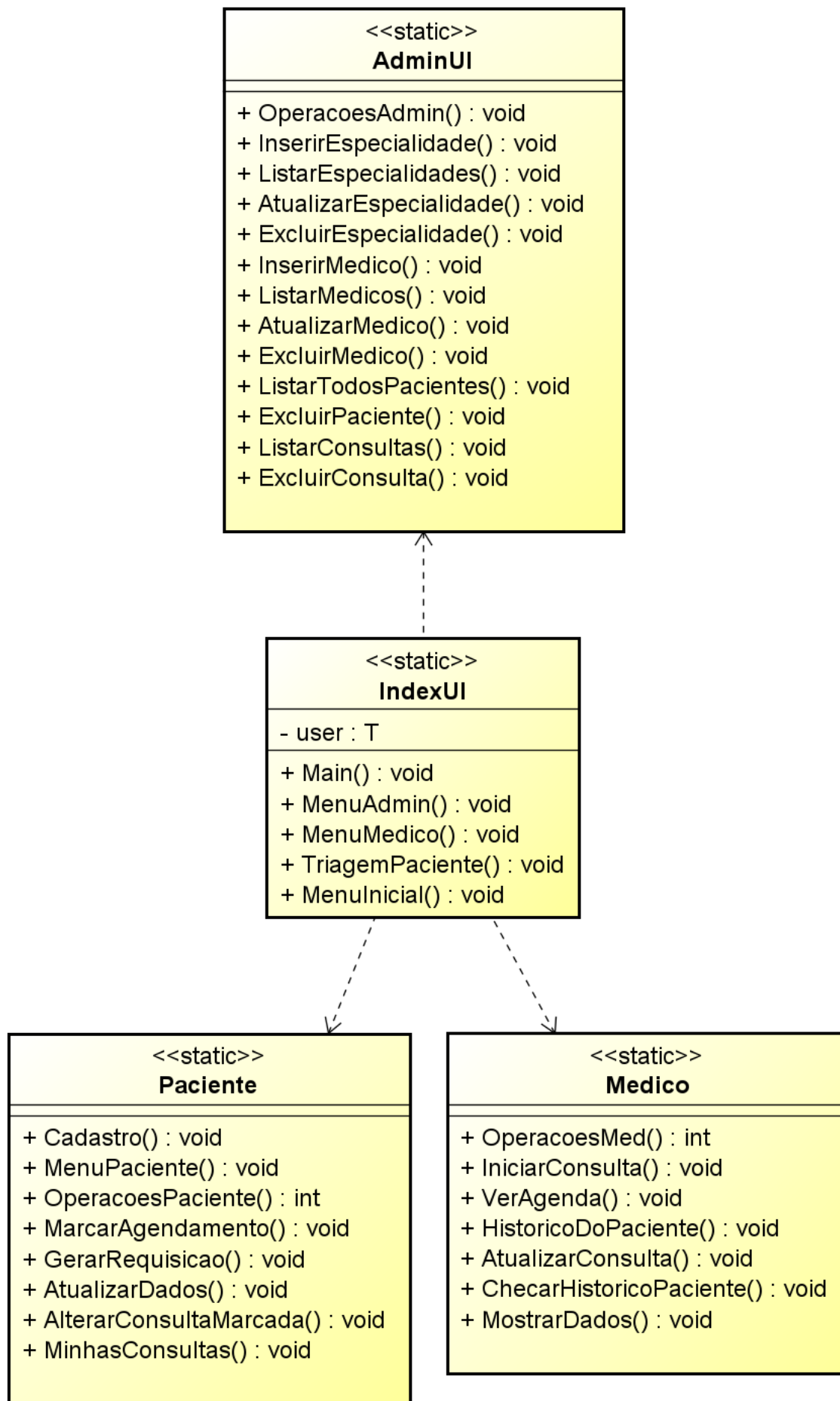
| <b>&lt;&lt; NPaciente &gt;&gt;</b>  |
|---|
| - pacientes : List<Paciente>  |
| + InserirPaciente() : void<br>+ ListarPaciente() : void<br>+ AtualizarPaciente() : void<br>+ ExcluirPaciente() : void |

| <b>&lt;&lt; NConsulta &gt;&gt;</b>  |
|---|
| - consultas : int   |
| + InserirConsulta() : void<br>+ ListarConsulta() : void<br>+ ExcluirConsulta() : void<br>+ AtualizarConsulta() : void |

| <b>&lt;&lt; NMedico &gt;&gt;</b>  |
|---|
| - medicos : List<Medico>  |
| + InserirMedico() : void<br>+ ListarMedico() : void<br>+ ExcluirMedico() : void<br>+ AtualizarMedico() : void |

| <b>&lt;&lt; NEspecialidade &gt;&gt;</b>   |
|---|
| - especialidades : List<Especialidade>  |
| + InserirEspecialidade() : void<br>+ ListarEspecialidade() : void<br>+ ExcluirEspecialidade() : void<br>+ AtualizarEspecialidade() : void |

## 6. Diagrama de classes - Interface com o Usuário



## 7. Link do repositório do GitHub

<https://github.com/Felipe-CF/Projeto-POO.git>