

Estruturas do Programa:

O software é destinado às necessidades de uma clínica, concentrando nos pacientes, nos atendimentos e nos procedimentos. A estrutura do programa é modular, cada funcionalidade foi organizada em classes distintas, separando claramente suas responsabilidades.

- **Paciente:** Representa um paciente e suas informações pessoais, como nome, data de nascimento, sexo e CPF, possui métodos GET/SET para obtenção e inserção de informações.
- **Procedimento:** Apresenta detalhes de um procedimento como nome, código e duração. Possui métodos GET/SET para obtenção e inserção de informações.
- **Atendimento:** Representa um atendimento médico associado a um paciente e a um procedimento específico, com sua data de realização. Possui métodos GET/SET para obtenção e inserção de informações.

Classes Gerenciadoras:

Cada funcionalidade possui sua classe gerenciadora dedicada, onde são implementadas funcionalidades específicas e regras de negócio. As classes gerenciadoras são estáticas, dispensando a necessidade de instânciação: “GerenciarPaciente”, “GerenciarProcedimento” e “GerenciarAtendimento”.

- **GerenciarPaciente:**

Responsável pelo gerenciamento de pacientes, mantendo uma lista para armazenamento das instâncias de Paciente. Contém métodos para cadastrar novos pacientes, listar pacientes e buscar pacientes com base em critérios específicos.

O método para Cadastrar Paciente, solicita as informações do Paciente, realiza a validação dos dados fornecidos por meio de estruturas condicionais, e faz a chamada do método Buscar, que verifica pelo CPF a Lista de Paciente para verificar se já houve o cadastro.

O método “Buscar” é um método de sobrecarga, que na primeira implementação lista todos os pacientes cadastrados. A segunda implementação é fornecido como parâmetro um dado de busca que

realiza a procura do paciente por aquele dado, que pode ser pelo nome, CPF e data de nascimento.

- **GerenciarProcedimento:**

Para gerenciar os procedimentos foi criado uma Lista para armazenar as instâncias de Procedimento. Contém métodos para cadastrar novos procedimentos, listar procedimentos, buscar procedimentos, contar quantos procedimentos foram realizados em um período, contar a duração dos procedimentos realizados por período e validar se um procedimento pode ser realizado.

O método para cadastrar Procedimento, solicita informações do Procedimento, realiza validações na entrada da duração do procedimento e realiza o cadastro do procedimento. Embora a criação dinâmica não tenha sido solicitada, esses procedimentos são essenciais para simular possíveis cenários.

O método “Buscar” é um método de sobrecarga que na primeira implementação lista todos os procedimentos cadastrados. Na segunda implementação recebe como parâmetro o código do procedimento e realiza a busca.

O método “QuantProcedimentoPeriodo” solicita duas datas e realiza a contagem da quantidade de procedimentos realizados durante o período, podendo ser escolhido um específico ou todos. A checagem dos procedimentos ocorre por estruturas condicionais que fazem a checagem através do código do procedimento.

O método “DuracaoProcedimentoPeriodo” solicita duas datas e realiza a contagem da duração em horas dos procedimentos realizados durante o período, podendo ser escolhido um específico ou todos. O método utiliza estruturas condicionais para chegar à preferência do usuário e dependendo utiliza o foreach para realizar a contagem de apenas um procedimento ou realiza verificações dos procedimentos em estruturas condicionais utilizando o foreach.

O método “validaProcedimento” verifica se o paciente pode ou não realizar o procedimento por meio de estruturas condicionais com base nas exigências necessárias de cada procedimento.

- **GerenciaAtendimento:**

Para gerenciar os atendimentos foi criada uma lista para armazenar as instâncias de atendimento. Contém métodos para cadastrar novos atendimentos, listar atendimentos e buscar atendimentos por data.

O método de cadastrar um novo atendimento, solicita um dado do paciente, podendo ser seu nome, CPF ou sua data de nascimento, realiza a chamada do método `Buscar` presente no `GerenciarPaciente` e realiza a busca do paciente cadastrado na `Lista de Paciente`. Caso o paciente não seja encontrado, o usuário poderá fazer seu cadastro. Após, é solicitada a data do procedimento, sendo validada por uma estrutura condicional e solicita o procedimento desejado, o método realiza a chamada do método `"Buscar"` presente no `GerenciarProcedimento` e faz a busca do procedimento pelo seu código. O método valida se o paciente pode ou não realizar o procedimento com base nas exigências dos procedimentos, faz a chamada do método `"ValidaProcedimento"` presente no `GerenciarProcedimento`, se retornado `true` realiza o cadastramento do atendimento, retornando `false` realiza um impedimento no atendimento.

O método `"Buscar"` solicita uma data e lista os atendimentos realizados na data fornecida.

O método `"VerificarAtendimento"` é utilizado pelo método `"ValidaProcedimento"` presente no `GerenciarProcedimento`, como um método de auxílio que realiza a busca se o paciente realizou o procedimento de Ultrassonografia Obstétrica ou Ultrassonografia de Próstata nos últimos 3 meses antecedentes da data da consulta.

Menu Interativo:

O arquivo principal, `"Program.cs"`, apresenta um menu interativo que elenca as funcionalidades principais (`Paciente`, `Atendimento` e `Procedimento`). Cada opção do menu redireciona para a classe responsável por aquela funcionalidade. Todas as classes gerenciadoras possuem métodos de menu que representam as opções específicas daquela classe.

Decisões e Justificativas:

Separação de Responsabilidades: Cada classe tem uma responsabilidade, seguindo o princípio de responsabilidade única. A separação permite uma fácil manutenção.

Métodos Específicos: Cada classe possui métodos específicos para realizar suas tarefas.

Validação de Dados: Implementação de validações de dados para garantir a integridade das informações.

Utilização de Listas: Utilização de listas para armazenar instâncias de pacientes, procedimentos e atendimentos. Facilita a manipulação dinâmica dos dados.

Menus Interativos: Implementação de menus interativos para facilitar a interação do usuário.

Tratamento de Exceções: Inclusão de tratamento de exceções para lidar com possíveis erros durante a entrada do usuário.