Atividade Prática: Modelagem de Interação (Clínica Médica)

Monte no Draw.io o diagrama de sequência.

Acesse:

https://github.com/profmartinz/diagramasequencia

Faça o download, extraia o arquivo e abra no draw.io

Contexto:

Você está modelando um sistema para uma Clínica Médica que permite aos pacientes agendar consultas online. A funcionalidade principal é a de Agendar Consulta.

PARTE 1: Diagrama de Caso de Uso (Análise Funcional)

Abaixo está o Diagrama de Caso de Uso simplificado para a funcionalidade principal do sistema.

Caso de Uso: Agendar Consulta

Ator	Caso de Uso	Relacionamento
Paciente	Agendar Consulta	
Sistema	Gerar Confirmação	<<include>> de Agendar Consulta
Sistema	Verificar Disponibilidade	<<include>> de Agendar Consulta

PARTE 2: Diagrama de Classe (Estrutura de Dados)

O sistema de Agendamento é composto pelas seguintes classes e seus métodos essenciais.

Classe	Atributos (simplificado)	Métodos Públicos (essenciais)
Paciente	id, nome, email	+ solicitarAgendamento(data, hora, especialidade): Agendamento
Agendamento	id, dataHora, especialidade, status	+ criarNovo(paciente, dataHora, especialidade): void
ControleAgendamento	-	+ verificarDisponibilidade(data, hora, esp): boolean + salvarAgendamento(agendamento): void + enviarConfirmacao(agendamento): void

PARTE 3: O Desafio (Diagrama de Sequência)

Cenário de Sucesso: O Paciente solicita o agendamento de uma consulta em uma data e hora disponíveis.

Tarefa:

Crie um Diagrama de Sequência para o fluxo principal de sucesso do caso de uso "**Agendar Consulta**".

Passos do Fluxo (para usar no diagrama):

1. O ***Paciente*** interage com a interface (aqui representada pelo próprio ator) e chama o método *solicitarAgendamento*(data, hora, especialidade).
2. A classe ***ControleAgendamento*** é chamada para verificar se a data/hora está livre através do método *verificarDisponibilidade(data, hora, especialidade)*.
3. O ***ControleAgendamento*** recebe o retorno (positivo, pois é um cenário de sucesso).
4. O ***ControleAgendamento*** cria um novo objeto da classe ***Agendamento*** (Criação de Objeto).
5. O ***ControleAgendamento*** persiste (salva) o novo ***Agendamento*** (método *salvarAgendamento*).
6. O ***ControleAgendamento*** envia uma mensagem de confirmação ao Paciente (método *enviarConfirmacao*).
7. O Paciente recebe o resultado.

Elementos que devem ser incluídos:

- O Ator: *Paciente*
- Os objetos: *: ControleAgendamento* e *: Agendamento* (o objeto Agendamento é criado no meio do fluxo).
- As *Lifelines* e *Barras de Ativação*.
- Todas as *Mensagens* (Síncronas e de Retorno) conforme o fluxo acima.

ENTREGA: Um Diagrama de Sequência desenhado, seguindo a lógica da UML.