

Portal do Paciente (Clínica

Saúde+) 1. Visão Geral e Objetivo

Objetivo: Este projeto é a sua principal avaliação e oportunidade de aprendizado. Você aplicará os conceitos de Análise de Requisitos, Modelagem de Negócios, Prototipação de UX e Modelagem UML (Casos de Uso, Classes, Sequência, Atividades) para desenvolver a documentação técnica completa de um sistema real, trabalhando de forma colaborativa com sua equipe.

Produto Final: Ao final dos 8 encontros, seu grupo deverá entregar um **repositório Git completo** contendo toda a documentação, modelos, diagramas e artefatos do projeto, além de uma **apresentação final (pitch)** defendendo suas soluções de análise e design.

2. O Desafio (O Caso de Negócio)

A **Clínica Saúde+** é seu novo cliente. É uma clínica de médio porte com 15 médicos de 8 especialidades, mas que enfrenta um problema crônico de eficiência.

O Sistema Atual: Todo o gerenciamento da clínica é feito por um sistema desktop antigo (chamado "ClinicSys 1.0"), instalado apenas nos computadores da recepção e sem qualquer acesso pela internet.

O Problema Central: O agendamento de consultas é o principal gargalo. 100% dos agendamentos são feitos por telefone ou presencialmente. Isso sobrecarrega as duas recepcionistas, que passam o dia ao telefone em vez de realizar tarefas administrativas importantes (como faturamento). Os pacientes reclamam que o telefone está sempre ocupado e que não têm autonomia para marcar consultas fora do horário comercial ou ver seus próprios históricos.

Sua missão é projetar a solução para este problema.

3. Personas (Os Usuários-Chave)

Para que sua solução seja eficaz, você deve focar nas necessidades dos usuários. Conheça os principais stakeholders:

- **Fábio (Paciente, 25 anos):**

- *Contexto:* Jovem, usa o celular para tudo. Trabalha em horário comercial.
- *Dor:* "Odeio ter que ligar no meu horário de almoço e ficar na espera. Quero marcar minha consulta às 22h, quando estou livre, e pelo celular."
- *Necessidade:* Autoatendimento para agendar, cancelar e consultar horários 24/7.

- **Sandra (Recepcionista, 45 anos):**

- *Contexto*: Usa o "ClinicSys 1.0" há 10 anos. É ágil no sistema antigo, mas passa 80% do dia ao telefone.
- *Dor*: "Não consigo fazer meu trabalho de faturamento porque o telefone não para. Quando o paciente liga, tenho que conferir a agenda do médico no ClinicSys e falar os horários vagos um por um. É muito lento."
- *Necessidade*: Reduzir o fluxo de ligações para focar em tarefas administrativas. Quer que o novo sistema *reserve* o horário para o paciente, mas ela ainda quer *aprovar* a consulta para evitar duplicidade e erros.
- **Dr. Marcos (Médico, 52 anos)**:
 - *Contexto*: Focado no atendimento. Sua agenda é gerenciada pela Sandra no ClinicSys.
 - *Dor*: "Às vezes a Sandra me interrompe para confirmar um 'encaixe'. Gostaria de ter minha agenda online para eu mesmo consultar do meu celular."
 - *Necessidade*: Visibilidade da sua própria agenda e menos interrupções.

5. Ferramentas e Metodologia (Git)

Para simular um ambiente de desenvolvimento profissional, todo o projeto será gerenciado com Git e GitHub (ou GitLab).

- **Repositório**: Seu grupo deverá fazer um repositório e colocar os integrantes e professor como colaboradores.
- **Branches**: Cada entrega (checkpoint) será feita em uma *feature branch* separada (ex: *feature/requisitos-bmc*).
- **Pull Requests (PRs)**: A entrega de cada checkpoint é a abertura de um **Pull Request (PR)** da sua *feature branch* para a branch *develop*.
- **Feedback (Code Review)**: O professor não dará "nota" no checkpoint. Ele fará um *code review* (revisão de código/documento) diretamente no seu PR, deixando comentários, sugestões e correções. Você deve corrigir o PR e fazer novos commits na mesma branch.

6. Entregáveis e Cronograma (Checkpoints) Seu projeto

será construído de forma incremental. Cada checkpoint é um PR no GitHub.

Checkpoint 1

Análise de Negócio e Requisitos Iniciais

- **Branch**: *feature/requisitos-bmc*
- **Pull Request 1**: Deve conter:
 1. docs/README.md: Nomes dos integrantes do grupo.
 2. docs/BMC.(md ou png): O Business Model Canvas (BMC) preenchido, detalhando como o portal agrega valor.
 3. docs/REQUISITOS.md (v1.0): A primeira versão do documento de requisitos,

contendo:

- Lista de Stakeholders (Personas).
 - No mínimo **10 Requisitos Funcionais (RF)** (O que o sistema *faz*? Ex: "Paciente deve poder...").
 - No mínimo **5 Requisitos Não Funcionais (RNF)** (Como o sistema *é*? Ex: "Sistema deve ser rápido...").
 - No mínimo **3 Regras de Negócio (RN)** (Políticas da clínica. Ex: "Paciente só pode...").

7. Critérios de Avaliação

Sua nota será composta pela **avaliação formativa** (qualidade dos PRs e resposta ao feedback) e pela **avaliação somativa** (qualidade do produto final no repositório e da apresentação).

Os principais critérios da Rubrica Final são:

1. **Clareza e Completude dos Requisitos:** Seus RFs, RNFs e RNs são claros, mensuráveis e cobrem o escopo do problema?
2. **Adequação da Modelagem UML:** Seus diagramas (DCU, DC, DS, DA) estão corretos na notação UML e são consistentes com os requisitos? A cardinalidade no DC está correta? 3. **Consideração de UX e Usabilidade:** Seus wireframes resolvem as dores das personas (Fábio, Sandra)?
4. **Coerência da Documentação e Apresentação:** A apresentação foi profissional? O repositório está organizado?
5. **Autonomia e Iniciativa (Git):** O grupo demonstrou profissionalismo no Git (commits claros, PRs bem descritos, resposta ao feedback)?