

med.AI

SPRINTS	INÍCIO E FIM	ÉPICO	DEFINIÇÃO DE PRONTO
1	04/09 - 10/09	Preparação inicial	Repositório criado no GitHub, wireframes principais definidos no Figma, ambiente de desenvolvimento configurado no VSCode, esqueleto da aplicação desenhado no Figma.
2	11/09 - 17/09	Modelo Entidade-Relacionamento (MER)	Diagrama ER criado e revisado, modelo documentado e aprovado pela equipe de desenvolvimento.
3	18/09 - 24/09	Listagem de Médicos	Página de listagem de médicos implementada, dados de médicos e especialidades exibidos corretamente no frontend, funcionalidade testada.
4	25/09 - 01/10	Cadastro e Login	Funcionalidades de cadastro e login implementadas e testadas, usuários podem se cadastrar, fazer login e acessar sua conta.
5	02/10 - 08/10	Agendamento de Consultas	Funcionalidade de agendamento implementada, pacientes conseguem agendar consultas com médico.

Sprint 1: Preparação Inicial

Objetivo: Organizar o ambiente de desenvolvimento e criar a estrutura inicial do projeto.

História do Usuário:

Como **paciente e usuária**, quero ter uma **visão clara do projeto** para facilitar meu entendimento do aplicativo.

Etapas:

1. Criar repositório no GitHub para centralizar o código.

2. Definir o layout básico com wireframes no Figma para as páginas principais.
3. Configurar o ambiente de desenvolvimento no VSCode com as bibliotecas necessárias (HTML, CSS, JavaScript e Python).
4. Estruturar o esqueleto inicial da aplicação no Figma, incluindo home, login e listagem de médicos.

Definição de Pronto:

- Repositório criado no GitHub com código básico.
- Wireframes principais definidos no Figma.
- Ambiente de desenvolvimento configurado no VSCode.
- Esqueleto da aplicação desenhado e revisado no Figma.

Sprint 2: MER (Modelo Entidade Relacionamento) da Aplicação

Objetivo: Definir e estruturar o modelo entidade-relacionamento (ER) do MED.Ai, que descreve a relação entre pacientes, médicos, consultas e outros elementos-chave da aplicação.

História do Usuário:

- Eu, como **administradora da aplicação**, quero um **modelo entidade-relacionamento bem estruturado** para que seja possível organizar e armazenar os dados de usuários, médicos, consultas e avaliações de forma clara e eficiente.

Etapas:

1. **Identificar as entidades principais** da aplicação
2. **Definir os atributos** de cada entidade
3. **Estabelecer os relacionamentos** entre as entidades
4. **Criar o diagrama entidade-relacionamento (ER)**, visualizando como os dados se conectam e fluem pela aplicação
5. **Revisar e ajustar o modelo** conforme necessário

Definição de Pronto: O diagrama ER foi criado, revisado e aprovado.

Sprint 3: Listagem de Médicos

Objetivo: Exibir a lista de especialidades disponíveis e seus respectivos médicos.

História do Usuário:

Como **paciente**, quero **visualizar a lista de médicos e especialidades** para escolher o profissional mais adequado às minhas necessidades.

Etapas:

1. Criar página de listagem de médicos no Figma, com informações com nome, CRM e especialidade.
2. Implementar a funcionalidade de listagem de médicos no frontend, integrando com o backend para buscar os dados.
3. Testar a funcionalidade de exibição de médicos no frontend para verificar se todas as informações estão corretas.
4. Revisar e realizar ajustes de design e funcionalidade conforme necessário.

Definição de Pronto:

- Página de listagem de médicos implementada e funcionando corretamente.
- Dados de médicos e especialidades exibidos com precisão no frontend.

Sprint 4: Cadastro e Login

Objetivo: Implementar a funcionalidade de cadastro e login de usuários.

História do Usuário:

Eu, como **usuário**, quero **poder me cadastrar e fazer login** para acessar minha conta e serviços personalizados no sistema.

Etapas:

1. Criar tela de cadastro no frontend com validação de campos.
2. Implementar a lógica de cadastro no backend, incluindo o armazenamento de dados dos usuários.
3. Criar tela de login no frontend.

4. Implementar a funcionalidade de login e gerenciamento de sessões no backend.
5. Testar o processo completo de cadastro e login, garantindo a segurança dos dados.

Definição de Pronto:

- Funcionalidades de cadastro e login implementadas e testadas.
- Usuários podem se cadastrar, fazer login e acessar sua conta.

Sprint 5: Agendamento de Consultas

Objetivo: Implementar a funcionalidade de agendamento de consultas com os médicos.

História do Usuário:

Como **paciente**, quero **agendar uma consulta com um médico** para garantir meu atendimento na data desejada.

Etapas:

1. Criar interface no frontend para seleção de data e hora para o agendamento.
2. Implementar lógica no backend para salvar e gerenciar os agendamentos.
3. Validar se a data e hora escolhidas estão disponíveis no sistema.
4. Testar o processo completo de agendamento para garantir que as datas sejam armazenadas corretamente.
5. Implementar notificações para o usuário confirmar a consulta agendada.

Definição de Pronto:

- Funcionalidade de agendamento implementada e funcionando.
- Pacientes conseguem agendar consultas com médicos com datas e horários disponíveis.
- Notificações de confirmação funcionando corretamente.