

Relatório de Reunião de Planning e Review - Engenharia de Software

1. Informações Gerais

Data: 05/03/2025

Duração: 123 minutos

Plataforma: Discord

Facilitador: Ismael Henrique Xavier Cavalcante dos Santos

Participantes:

- Ismael Henrique Xavier Cavalcante dos Santos
 - Lucas Americo Correia de Lima
 - Maria Beatriz Martins Pontes Goncalo
 - Sara Simone Emilay de Araujo Pereira
-

2. Objetivos da Reunião

O objetivo da reunião foi revisar e discutir as principais etapas do projeto, incluindo a estrutura das telas, modelagem do banco de dados, fluxo de usuário e arquitetura de software, garantindo que todas as funcionalidades estejam validadas antes do início do desenvolvimento do backend e frontend.

3. Itens Discutidos

3.1. Fluxo de Telas e Prototipagem

- **Descrição:** Revisão dos fluxos de usuário e telas prototipadas.
- **Pontos discutidos:**
 - Limitação do plano gratuito da ferramenta, exigindo divisão dos arquivos.
 - Juntar diferentes protótipos em um único documento.
 - Definição de menus expansíveis e botões interativos.
- **Ajustes Necessários:**
 - Organizar arquivos para facilitar revisão e edição colaborativa.
- **Responsável pelos Ajustes:** Lucas Americo Correia de Lima.

3.2. Estrutura do Menu e Componentes de Navegação

- **Descrição:** Definição do layout e funcionalidade do menu lateral.
- **Pontos discutidos:**
 - Menu deve ser expansível ou fixo?
 - Possibilidade de incluir **nickname** para usuários.
 - Organização das disciplinas por curso e período.
- **Ajustes Necessários:**
 - Implementar menu lateral com opção de expansão.
 - Decidir sobre a exibição do **nickname**.
- **Responsável pelos Ajustes:** Ismael Henrique Xavier Cavalcante dos Santos.

3.3. Banco de Dados e Modelagem

- **Descrição:** Estrutura e atributos do banco de dados.
- **Pontos discutidos:**
 - Inclusão do **código da disciplina** como campo obrigatório.
 - Definição da estrutura de **períodos das disciplinas** para diferentes cursos.
 - Registro de **disciplinas equivalentes**.
- **Ajustes Necessários:**
 - Incluir campo de código da disciplina.
 - Avaliar a viabilidade da vinculação de disciplinas equivalentes.
- **Responsável pelos Ajustes:** Maria Beatriz Martins Pontes Goncalo.

3.4. Avaliação e Estatísticas das Disciplinas

- **Descrição:** Implementação de estatísticas e métricas.
- **Pontos discutidos:**
 - **Nota média** da disciplina.
 - **Dificuldade e porcentagem de desistência**.
 - Exibição de histórico de desempenho por semestre.
- **Ajustes Necessários:**
 - Implementar essas estatísticas no wireframe.
- **Responsável pelos Ajustes:** Ismael Henrique Xavier Cavalcante dos Santos.

3.5. Arquitetura de Software

- **Descrição:** Estrutura do backend e frontend.
- **Pontos discutidos:**
 - Backend baseado em Prisma e separação por camadas.
 - Frontend utilizando **React** com separação de lógica e UI.
 - Uso de **Docker** para padronização do ambiente.
- **Ajustes Necessários:**
 - Criar documentação detalhada da arquitetura.
- **Responsável pelos Ajustes:** Sara Simone Emilay de Araujo Pereira

4 Discussões Técnicas Importantes

A reunião também incluiu debates sobre aspectos técnicos essenciais.

4.1 Organização do Banco de Dados

- O grupo analisou se as disciplinas deveriam ter **códigos únicos por curso** ou **um código geral**.
- Foi discutido **como associar professores às disciplinas**.
- Debate sobre **como organizar os períodos das disciplinas**, já que algumas são ofertadas em diferentes cursos.

4.2 UI/UX - Melhorias na Interface

- Houve dúvidas sobre **como exibir a dificuldade das disciplinas** (estrelas, números, etc.).
- Surgiu a ideia de permitir que os alunos **recomendassem materiais** junto às avaliações.
- Discutiu-se **a importância da padronização** das telas para manter a consistência no design.

4.3 Arquitetura e Desenvolvimento

- Foi definida uma estrutura baseada em **camadas**, separando:
 - **Infraestrutura** (requisições e comunicação com o banco)
 - **Domínio** (regras de negócio)
 - **Serviços** (interações com APIs)
 - **Front-end (React e Hooks)**
- O grupo decidiu usar **Docker** para manter um ambiente de desenvolvimento consistente.
- Foram apresentadas as vantagens do **Prisma ORM** para manipular os dados sem escrever SQL manualmente.