**1. Requisitos e Funcionalidades**

Funcionalidades a serem implementadas de acordo com os requisitos estabelecidos:

1. **Criar Turmas**
2. **Adicionar Alunos a uma Turma**
3. **Utilizar OCR para Inserir Alunos**
4. **Criar Categorias de Notas (trabalhos, provas etc.)**
5. **Registrar Notas por Categoria**
6. **Registrar Presença e Ausência dos Alunos**
7. **Visualizar Histórico de Frequência por Aluno e por Turma**
8. **Editar e Excluir Dados de Alunos e Turmas**
9. **Visualizar Notas e Desempenho**

**2. Planejamento e Estrutura do Projeto**

**2.1. Design Central e Usabilidade**

Para garantir uma experiência de usuário consistente, será criada uma identidade visual única e um fluxo de usabilidade do aplicativo. A seguir, as definições:

* **Fontes:** Definição de uma tipografia padrão que será usada em todo o aplicativo.
* **Paleta de Cores:** Escolha de uma paleta de cores alinhada à proposta visual do projeto, garantindo legibilidade e usabilidade.
* **Identidade Visual Base**: Definição do estilo de botões, ícones e componentes reutilizáveis na interface.
* **Fluxo de Usabilidade:** Planejamento da navegação do usuário, assegurando que os caminhos entre telas sejam intuitivos e diretos.

**2.2. Arquitetura e Abordagem de Tecnologia**

Para garantir que o aplicativo seja escalável, modular e de fácil manutenção, será definida a arquitetura e as tecnologias base do projeto:

* **Persistência de Dados**: Uso do **ROOM** para gerenciar o armazenamento local de dados no aplicativo.
* **OCR (Optical Character Recognition)**: Integração de biblioteca OCR para inserção automática de alunos a partir de listas físicas.
* **Arquitetura de Software**: O aplicativo seguirá uma arquitetura baseada em MVVM (Model-View-ViewModel) para separar a lógica de apresentação, garantindo manutenibilidade e escalabilidade.
* **Linguagem e Frameworks**: O app será desenvolvido com **Kotlin** no Android, e a interface de usuário será criada utilizando **Jetpack Compose**.

**2. Legenda de categorias de tarefas**

|  |  |
| --- | --- |
| **Roxo** | Design, interfaces e experiência de usabilidade |
| **Azul** | Construção de layouts, responsividade e visualização dos dados (frontend) |
| **Laranja** | Persistência e lógica dos dados armazenados e sua comunicação com outros módulos (backend) |
| **Verde** | Testes unitários e de integração |
| **Cinza** | Tarefas relacionadas ao OCR (*Optical Character Recognition*) |

**3. Decomposição (breakdown) em Requisitos e Histórias de Usuário**

Transformação dessas funcionalidades em histórias de usuário e decomposição de cada uma em tarefas específicas.

**Funcionalidade 1: Criar Turmas**

* **História de Usuário: "Como um professor, quero criar novas turmas para que eu possa organizar meus alunos por classe."**
  + **Tarefas:**
    1. **Criar o design da interface de cadastro de turmas**
    2. **Criar interface de usuário para o cadastro de turmas.**
    3. **Implementar lógica de backend para criar e salvar turmas usando ROOM.**
    4. **Validar campos obrigatórios (nome da turma, ano letivo etc.).**
    5. **Testar a criação de turmas com diferentes entradas (nomes válidos, nomes duplicados etc.).**

**Funcionalidade 2: Adicionar Alunos a uma Turma**

* **História de Usuário: "Como um professor, quero adicionar alunos a uma turma para que eu possa registrar suas presenças e notas."**
  + **Tarefas:**
    1. **Criar o design da interface de adicionar alunos manualmente.**
    2. **Criar interface de usuário para adicionar alunos manualmente.**
    3. **Implementar a lógica para vincular alunos a uma turma específica no banco de dados.**
    4. **Validar campos obrigatórios (nome do aluno, número de matrícula etc.).**
    5. **Testar a adição de alunos, incluindo casos de alunos duplicados.**

**Funcionalidade 3: Utilizar OCR para Inserir Alunos**

* **História de Usuário: "Como um professor, quero usar OCR para inserir alunos de uma lista física, para economizar tempo e evitar erros manuais."**
  + **Tarefas:**
    1. **Integrar a biblioteca de OCR ao projeto.**
    2. **Criar o design da interface para upload/captura da lista de alunos.**
    3. **Criar uma interface para upload/captura da lista de alunos.**
    4. **Processar a imagem e extrair os dados dos alunos.**
    5. **Validar e corrigir possíveis erros de reconhecimento do OCR.**
    6. **Testar o reconhecimento do OCR por meio de imagens e obter uma análise acurácia e desvio padrão de erros.**
    7. **Salvar os dados no banco de dados.**

**Funcionalidade 4: Criar Categorias de Notas**

* **História de Usuário: "Como um professor, quero criar categorias de notas (como trabalhos e provas) para poder registrar e organizar as avaliações dos alunos."**
  + **Tarefas:**
    1. **Criar design de interface para criação e edição de categorias de notas.**
    2. **Criar interface para a criação e edição de categorias de notas.**
    3. **Implementar a lógica para armazenar essas categorias no banco de dados.**
    4. **Validar a entrada de dados (nomes de categorias, pesos etc.).**
    5. **Testar a criação, edição e exclusão de categorias.**

**Funcionalidade 5: Registrar Notas por Categoria**

* **História de Usuário: "Como um professor, quero registrar notas para os alunos em cada categoria para acompanhar seu desempenho."**
  + **Tarefas:**
    1. **Criar design da interface para registrar notas de alunos em diferentes categorias.**
    2. **Criar interface para registrar notas de alunos em diferentes categorias.**
    3. **Implementar a lógica de backend para salvar essas notas no banco de dados.**
    4. **Validar as notas inseridas (intervalo válido, campos obrigatórios etc.).**
    5. **Testar a inserção e atualização de notas.**

**Funcionalidade 6: Registrar Presença e Ausência dos Alunos**

* **História de Usuário: "Como um professor, quero registrar a presença e ausência dos alunos para acompanhar sua frequência nas aulas."**
  + **Tarefas:**
    1. **Criar design de interface para marcar a presença ou ausência dos alunos.**
    2. **Criar interface para marcar a presença ou ausência dos alunos.**
    3. **Implementar a lógica de backend para registrar essas informações no banco de dados.**
    4. **Validar e garantir que os dados sejam registrados corretamente.**
    5. **Testar o registro de frequência em diferentes cenários (aulas com todos os presentes, alguns ausentes etc.).**

**Funcionalidade 7: Visualizar Histórico de Frequência**

* **História de Usuário: "Como um professor, quero visualizar o histórico de frequência de cada aluno e turma para monitorar a assiduidade."**
  + **Tarefas:**
    1. **Criar design de interface de visualização do histórico de frequência por aluno e por turma.**
    2. **Criar interface de visualização do histórico de frequência por aluno e por turma.**
    3. **Implementar a lógica para buscar e exibir essas informações no frontend.**
    4. **Testar a visualização com diferentes filtros (por data, por aluno etc.).**

**Funcionalidade 8: Editar e Excluir Dados**

* **História de Usuário: "Como um professor, quero poder editar ou excluir dados de alunos e turmas para manter as informações atualizadas."**
  + **Tarefas:**
    1. **Criar design de interface para edição de dados de alunos e turmas.**
    2. **Criar interface para edição de dados de alunos e turmas.**
    3. **Implementar a lógica de backend para suportar edição e exclusão.**
    4. **Implementar validações para evitar exclusão acidental (exemplo: confirmação antes de excluir).**
    5. **Testar as funcionalidades de edição e exclusão.**

**Funcionalidade 9: Visualizar Notas e Desempenho**

* **História de Usuário: "Como um professor, quero visualizar as notas e o desempenho dos alunos para acompanhar seu progresso ao longo do semestre."**
  + **Tarefas:**
    1. **Criar design da interface de visualização das notas por aluno e por turma.**
    2. **Criar interface de visualização das notas por aluno e por turma.**
    3. **Implementar a lógica para exibir os dados de desempenho no frontend.**
    4. **Testar a exibição das notas com diferentes parâmetros (filtro por categoria, por aluno etc.).**