

# **Descrição do Projeto: Academic Connect**

## **1. Introdução e Justificativa**

A vida acadêmica em grandes instituições de ensino, como a UFSCar, é frequentemente descentralizada. Informações vitais — notas, frequência, horários, comunicados e eventos — estão espalhadas por múltiplos sistemas (Moodle, portal do aluno) e meios físicos (murais, e-mails). Essa fragmentação gera ruído na comunicação e dificulta o engajamento, exigindo que alunos e professores gastem tempo e energia navegando por diversas plataformas.

O **Academic Connect** surge como uma solução móvel integrada, desenvolvida em Flutter, com o objetivo de unificar a experiência acadêmica. O aplicativo centraliza, em uma única interface moderna e reativa, todas as ferramentas necessárias para os três pilares da comunidade acadêmica: **Alunos, Professores e o Centro Acadêmico (C.A.)**.

Utilizando uma arquitetura moderna baseada no Firebase (Auth, Firestore, Storage) e flutter\_riverpod para gerenciamento de estado, o app oferece uma experiência em tempo real, segura e acessível de qualquer lugar.

## **2. Objetivos do Projeto**

- **Centralizar** o acesso a notas, frequência, horários e materiais de aula.
- **Automatizar** processos acadêmicos manuais, como a chamada de presença, utilizando o hardware NFC do dispositivo.
- **Facilitar** a comunicação direta entre Professores, Alunos e o Centro Acadêmico.
- **Provar** a viabilidade do uso de APIs externas (arXiv.org, Google Maps) e APIs de hardware (NFC, Leitor de Arquivos) em um ecossistema Flutter.
- **Entregar** uma aplicação, com design moderno, animações e suporte completo à internacionalização (i18n).

## **3. Perfis de Usuário e Funcionalidades**

O sistema é dividido em três perfis de usuário distintos, com funcionalidades específicas:

### 3.1. Portal do Aluno

O Aluno é o consumidor principal da informação, com foco em autoatendimento e organização.

- **Dashboard (Início):** Visão geral animada de sua vida acadêmica, incluindo um card de boas-vindas, média de frequência, e uma lista das próximas provas.
- **Acesso à API (arXiv.org):** Uma aba dedicada que consome a API do arXiv.org em tempo real, exibindo os últimos artigos científicos publicados relacionados ao curso do aluno (ex: "Computer Science").
- **Disciplinas e Frequência:** Lista de todas as disciplinas matriculadas, com barras de progresso de frequência.
- **Notas e Avaliações:** Detalhamento de notas por disciplina (P1, P2, Trabalhos) e uma lista de provas futuras, com data, conteúdo e local.
- **Acesso à Matéria (API URL):** Cada disciplina possui um link direto para seu ambiente virtual (ex: Google Classroom, Moodle), utilizando o url\_launcher.
- **Localização (API Google Maps):** O local da prova (ex: "Lab 203") é clicável e abre o app Google Maps externo, focado no prédio correto do campus (UFSCar Sorocaba).
- **Cadastro de NFC:** Uma tela dedicada para o aluno escanear e salvar o UID (ID Único) do seu cartão de estudante no seu perfil, com feedback de voz (TTS) e vibração.
- **Solicitar Adaptação:** Formulário para enviar solicitações formais (ex: tempo extra de prova), permitindo o anexo de arquivos (laudos) usando o file\_picker.
- **Perfil:** Visualização e edição dos dados cadastrais (Nome, RA, Curso, Período), salvando as alterações no Firestore.

### 3.2. Portal do Professor

O Professor é o gerenciador de conteúdo acadêmico, com foco em automação e administração.

- **Dashboard (Início):** Visão geral com atalhos para as funções mais importantes (Registrar Presença, Lançar Notas) e um resumo das aulas do dia e solicitações pendentes.
- **Gerenciamento de Turmas:** Visualização de todas as suas turmas, com detalhes de alunos e horários.
- **Registro de Presença (NFC):** Funcionalidade principal. O professor inicia um "loop de leitura" contínuo. Os alunos aproximam seus cartões, e o app:
  - Valida o UID contra o banco de dados.

- Confirma a presença com feedback de voz (TTS): "Presença de Vinícius Souza confirmada."
  - Ignora duplicatas: "Vinícius Souza já marcou presença."
  - Alerta sobre cartões não cadastrados.
- **Chamada Manual:** Como alternativa ao NFC, o professor tem uma lista de alunos com checkboxes.
- **Bloqueio de Chamada (Lógica de Estado):** O sistema impede que um professor inicie uma *nova* chamada para uma turma no mesmo dia (mas permite *continuar* uma chamada existente).
- **Lançar Notas:** Formulário para selecionar uma turma e uma avaliação (P1, P2) e inserir as notas dos alunos, salvando no Firestore.
- **Agendar Provas:** Formulário para criar novas avaliações (título, prédio, sala, data, conteúdo), que aparecem automaticamente no painel do aluno.
- **Gestão de Solicitações:** Tela para visualizar, **Aprovar** ou **Recusar** as solicitações de adaptação enviadas pelos alunos.
- **Código de Turma (Estilo Classroom):** O professor pode gerar um código único para a turma. (A implementação futura permitiria que alunos usassem esse código para se inscrever).
- **Enviar Comunicado:** Ferramenta para o professor enviar comunicados para turmas específicas.

### **3.3. Portal do Centro Acadêmico (C.A.) / Projeto**

O C.A. é o gerenciador da comunidade, focado em eventos e comunicação em massa.

- **Dashboard (Início):** Atalhos para as funções do C.A. e uma lista de "Links Úteis" (Biblioteca, Moodle, etc.) que abre no navegador.
- **Gerenciamento de Eventos:** O C.A. pode **Criar** novos eventos (Semana da Computação, palestras) através de um formulário.
- **Presença em Eventos (NFC):** Utiliza a **mesma** lógica de loop contínuo do professor, mas para registrar a presença de participantes em um evento (validando contra o banco geral de alunos).
- **Enviar Comunicado:** Ferramenta para enviar comunicados para públicos-alvo (ex: "Todos os Alunos", "Participantes do Evento X").

## **4. Arquitetura e Tecnologias**

- **Front-End:** Flutter 3.x
- **Gerenciamento de Estado:** flutter\_riverpod (para um estado reativo e centralizado).

- **Back-End (BaaS):** Firebase
  - **Autenticação:** FirebaseAuth (Login com E-mail/Senha e Login com Google).
  - **Banco de Dados:** Cloud Firestore (Armazena perfis de usuário, turmas, notas, provas, eventos, etc.).
- **APIs de Hardware:**
  - flutter\_nfc\_kit: Para leitura do UID dos cartões.
  - flutter\_tts: Para feedback de voz na presença.
  - file\_picker: Para upload de anexos (laudos).
  - url\_launcher: Para abrir links externos (Google Maps e materiais de aula).
- **APIs Externas (Web):**
  - http e xml: Para consumir a API do **arXiv.org** (Artigos de C&T).
- **Persistência Local:**
  - shared\_preferences: Para armazenar o token de login do Firebase, o tema (claro/escuro) e a língua (i18n) preferidos pelo usuário.
- **UI/UX e Animação:**
  - **Design:** Tema customizado (AppTheme) usando as paletas Roxo/Preto e Lilás/Bege, com widgets de "vidro fosco" (CartaoVidro).
  - **Animação:** Animações de FadeIn e Slide na tela de Login, e animações de lista (FadeInListAnimation) nos dashboards para carregar o conteúdo suavemente.
- **Internacionalização (i18n):**
  - Suporte completo a Português (pt), Inglês (en) e Espanhol (es) usando AppLocalizations e o pacote intl.