

## 1. Integração com a API e Arquitetura de Backend (O Maior Avanço)

O projeto deixou de usar dados mockados e passou a se conectar com uma API real, um marco fundamental para se tornar um produto funcional.

- **Conexão com a Produção:** O commit `feat: agora se conecta com a api real em produção` é o mais significativo, indicando que o aplicativo agora opera com dados reais.
- **Criação de Serviços:** Foram criados e integrados os serviços essenciais para a comunicação com o backend:
  - `Auth Service` (serviço de autenticação).
  - `Service de Instituições`.
  - `Service de Matérias`.
- **Refatoração:** O código foi atualizado para usar esses novos serviços, e dependências de serviços de teste (como `MockedAuthService`) foram removidas, demonstrando uma maturação na arquitetura.

## 2. Novas Funcionalidades e Experiência do Usuário (Core Features)

Diversas funcionalidades que impactam diretamente o usuário final foram implementadas e aprimoradas.

- **Gerenciamento de Faltas:**
  - A adição de faltas foi modernizada, passando a usar um *dialog* (caixa de diálogo).
  - O usuário agora pode digitar o número de faltas diretamente, tornando o processo mais rápido.
  - **(Recurso Premium)** Usuários Premium ganharam a habilidade de selecionar a data específica de uma falta, e a contagem é atualizada automaticamente.
- **Tela de "Insights":**
  - Os dias da semana agora aparecem em ordem cronológica, melhorando a leitura dos dados.
  - A lógica para exibir mensagens de status (aprovado/reprovado) foi aprimorada para maior precisão, evitando duplicidades.
  - Os cards de insights receberam bordas coloridas de acordo com o status, fornecendo um feedback visual imediato.
- **Gerenciamento de Matérias e Instituições:**
  - As matérias agora são exibidas em ordem alfabética.
  - Implementada a funcionalidade para **editar uma instituição** e **limpar todas as matérias** de uma instituição.
  - O *dialog* de adicionar matéria foi redesenhado para seguir a identidade visual do restante do site.
  - Foram adicionados validadores aos formulários para garantir a integridade dos dados inseridos.

## 3. Implementação do Modelo Premium

Ficou claro o desenvolvimento de um modelo de negócio com diferenciação entre usuários gratuitos e premium.

- **Acesso Restrito:** Usuários não-premium que possuem uma instituição (possivelmente de um período premium anterior) não podem mais acessá-la.
- **Mensagens Claras:** Foi adicionada uma mensagem específica para o usuário que cancelou a assinatura premium mas ainda possui instituições cadastradas.
- **Funcionalidades Exclusivas:** A seleção de data para as faltas é um exemplo claro de um recurso exclusivo para assinantes.

#### 4. Correções de Bugs Críticos e Estabilização

Uma parte importante do trabalho foi garantir que o sistema funcione como esperado.

- **Autenticação e Conta:**
  - Corrigido um erro grave no formulário de **troca de senha**, que não estava funcionando.
  - Ajustado um bug onde o sistema acusava "e-mail em uso" incorretamente na página de configurações. A verificação agora ocorre apenas no momento do envio do formulário.
- **Funcionalidade Principal:**
  - Resolvido um problema que impedia o funcionamento correto da funcionalidade de **adicionar instituição**.
- **Atualização de Estado:** Corrigido um bug onde a interface não atualizava o status de uma matéria logo após ela ser adicionada.

#### 5. Melhorias de Interface (UI) e Responsividade

O foco na qualidade visual e na experiência em diferentes dispositivos foi evidente.

- **Responsividade:** Foi adicionada responsividade para a tela de "Insights" e para a navegação em dispositivos móveis em geral, garantindo uma boa usabilidade em telas menores.
- **Layout e Estilo:**
  - Ajustes finos no layout de grid para alinhar os cards corretamente (**removido um gap**).
  - Uso de **minmax** no CSS para melhor adaptação dos elementos na tela.
  - Alteração de cores de botões para melhorar a usabilidade.
  - Adição de *dialogs* para exibir mensagens de erro durante requisições à API, melhorando o feedback ao usuário.

#### 6. Infraestrutura e Documentação

- **Configuração de Deploy (Vercel):** Foi criado o arquivo **vercel.json** para instruir o servidor a não retornar "Not Found" quando o usuário recarrega a página. Isso é essencial para o funcionamento correto de Single-Page Applications (SPAs).
- **Documentação:** O arquivo **README.md** foi atualizado, o que é vital para a manutenção e colaboração no projeto.

