

Documentação do Aplicativo SkillUpPlus 2030+

1. Introdução e Contexto do Projeto

O aplicativo **SkillUpPlus 2030+** foi desenvolvido em **React Native** utilizando o framework **Expo Router** para atender aos requisitos da Global Solution (GS) do 1º Semestre de 2025. O projeto visa apoiar a requalificação profissional e a inclusão digital, alinhando-se aos **Objetivos de Desenvolvimento Sustentável (ODS)** da ONU, especificamente o ODS 4 (Educação de Qualidade) e o ODS 8 (Trabalho Decente e Crescimento Econômico).

O foco principal do desenvolvimento foi a implementação de uma **estrutura de navegação híbrida** robusta e a utilização de componentes nativos e Hooks do React Native para criar uma experiência de usuário intuitiva e funcional.

2. Estrutura de Diretórios e Organização do Código

O projeto segue a convenção de roteamento baseada em arquivos do Expo Router, que organiza as telas e os layouts de navegação dentro da pasta `app/`.

Estrutura Principal

`SkillUpPlus2030Plus/`

```
|   └── app/
|       |   └── (drawer)/
|       |       |   └── (tabs)/
|       |       |       |   └── _layout.tsx      # Layout da Tab Navigation
|       |       |       |   └── home.tsx        # Tela Home
|       |       |       |   └── assessment.tsx    # Tela de Autoavaliação
|       |       |       |   └── trilhas.tsx      # Tela de Trilhas
|       |       |       |   └── progress.tsx     # Tela de Progresso
|       |       |   └── _layout.tsx      # Layout da Drawer Navigation
```

```
| |   └── profile.tsx      # Tela de Perfil  
| |   └── settings.tsx    # Tela de Configurações  
| └── _layout.tsx        # Layout da Stack Navigation (Root)  
└── index.tsx            # Tela de Login/Inicial  
  
└── assets/  
    └── babel.config.js  
└── package.json  
└── README.md
```

3. Implementação da Navegação Híbrida

A navegação do aplicativo combina os três principais tipos de navegação do React Navigation, aninhados de forma lógica:

3.1. Stack Navigation (Fluxo Sequencial)

- **Arquivo:** `app/_layout.tsx`
- **Função:** Controla o fluxo inicial do aplicativo. A tela de Login (`index.tsx`) é a primeira a ser carregada. Após o login (ou ao continuar como convidado), o usuário é direcionado para o grupo (`drawer`), que representa a área principal do aplicativo.

3.2. Drawer Navigation (Menu Lateral)

- **Arquivo:** `app/(drawer)/_layout.tsx`
- **Função:** Define o menu lateral que permite acesso rápido às seções principais.
 - O grupo (`tabs`) é o item principal do Drawer, representando o conteúdo central.
 - As telas `profile.tsx` e `settings.tsx` são rotas de nível superior acessíveis diretamente pelo Drawer.

3.3. Tab Navigation (Abas Inferiores)

- **Arquivo:** `app/(drawer)/(tabs)/_layout.tsx`
- **Função:** Define as abas inferiores que organizam as funcionalidades centrais do aplicativo (Home, Autoavaliação, Trilhas e Progresso). Este grupo está aninhado dentro do Drawer, garantindo que o menu lateral esteja sempre acessível nas telas principais.

4. Documentação das Telas e Funcionalidades

4.1. Tela de Login/Inicial (app/index.tsx)



Componentes/Funcionalidades	Aplicação
TextInput	Campos de Usuário e Senha .
TouchableOpacity / Button	Botão Entrar e Continuar como Convidado .
Hooks (<code>useState</code>)	Gerenciamento do estado local dos campos de login.

Componentes/Funcionalidades	Aplicação
Persistência de Dados	Utiliza <code>AsyncStorage</code> para simular o armazenamento de um <code>userToken</code> após o login bem-sucedido (Usuário: <code>aluno</code> , Senha: <code>123</code>). Este recurso atende ao requisito opcional de bonificação.
<code>Alert</code>	Exibição de mensagens de sucesso ou erro de login.
Navegação	Redireciona para <code>/(drawer)/(tabs)/home</code> após o login.

4.2. Tela Home (app/(drawer)/(tabs)/home.tsx)

≡ SkillUpPlus 2030+

Bem-vindo ao SkillUpPlus
2030+

Seu caminho para a requalificação digital.

Trilhas em Destaque

Inteligência Artificial
4 módulos | Iniciante

Sustentabilidade
4 módulos | Iniciante

Ações Rápidas

Autoavaliação Ver Trilhas

Componentes/Funcionalidades	Aplicação
ScrollView	Permite a rolagem vertical da tela, ideal para exibir conteúdo dinâmico e listas de cursos.
Image	Exibição do logotipo e banners de cursos (simulados com placeholders).
TouchableOpacity	Utilizado para criar cards de cursos clicáveis e botões de Ações Rápidas .
StyleSheet	Organização modular dos estilos para melhor manutenção.

4.3. Tela de Autoavaliação (app/(drawer)/(tabs)/assessment.tsx)

≡ SkillUpPlus 2030+

Autoavaliação de Competências

Descubra seu nível e receba recomendações personalizadas.

Área de Interesse

Inteligência Artificial (IA)

Competência Específica (Ex: Python, Liderança)

Digite a competência

Nível de Experiência (0 a 10)

Ex: 7

AVALIAR E GERAR RECOMENDAÇÕES

Home Autoavaliação Trilhas Progresso

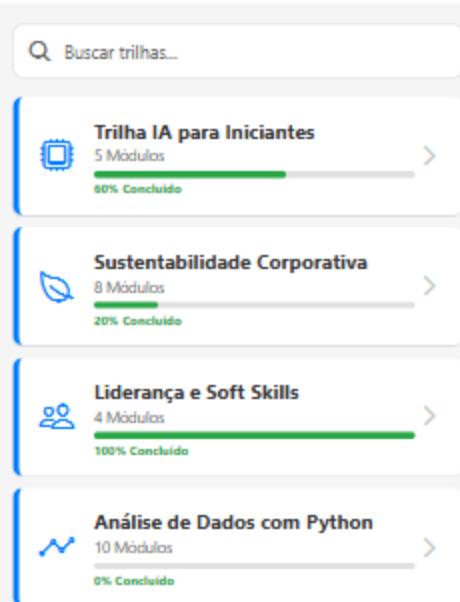
Componentes/Funcionalidades	Aplicação
<code>TextInput</code>	Campos para preenchimento da Competência Específica e Nível de Experiência.
<code>Picker</code>	Seleção da Área de Interesse (IA, Sustentabilidade, etc.), conforme requisito.
Formulário Validado	A função <code>handleAssessment</code> verifica se os campos obrigatórios foram preenchidos antes de exibir o alerta de envio.
Hooks (<code>useState</code>)	Gerenciamento do estado dos campos do formulário.

4.4. Tela de Trilhas (app/(drawer)/(tabs)/trilhas.tsx)

≡ SkillUpPlus 2030+

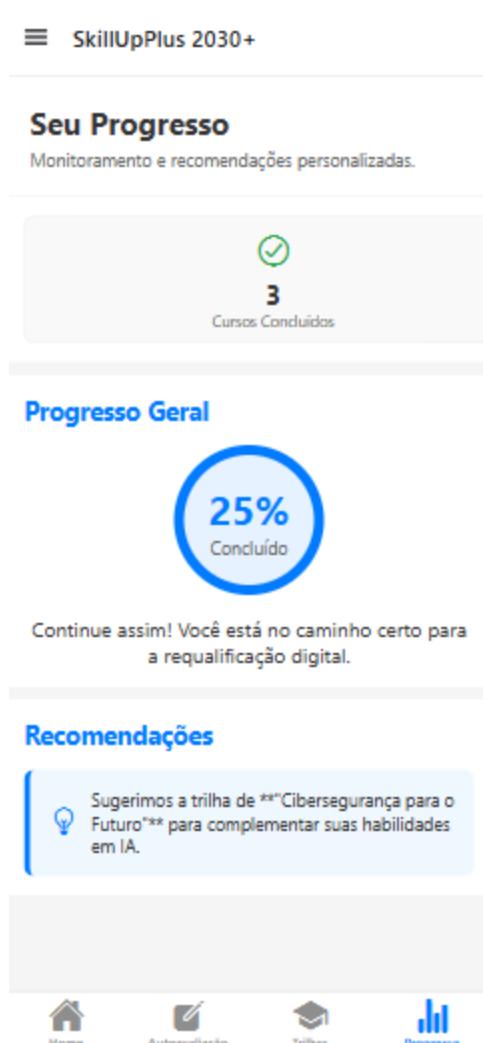
Trilhas de Aprendizado

Escolha seu caminho para o futuro do trabalho.



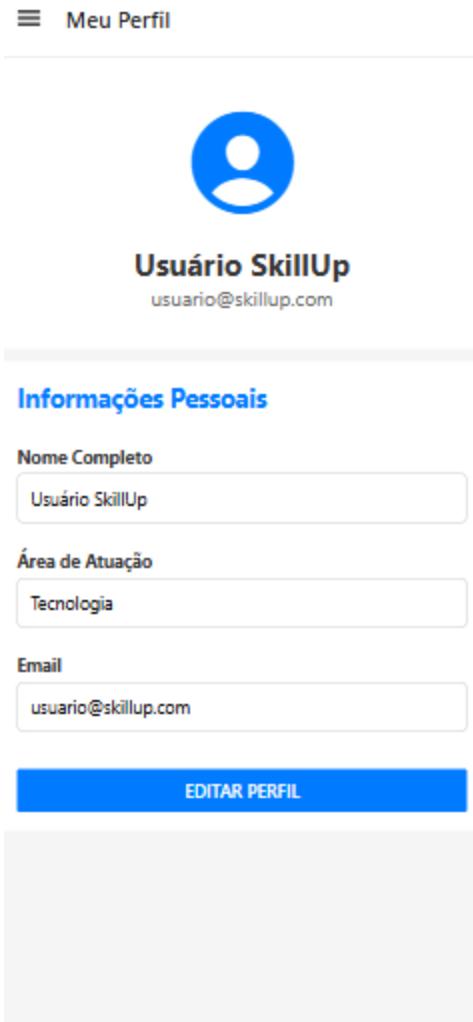
Componentes/Funcionalidades	Aplicação
ScrollView	Utilizado para a lista de trilhas, permitindo a rolagem.
TouchableOpacity	Cria cards clicáveis para cada trilha, simulando a navegação para o detalhe do curso.
View e Text	Utilizados para estruturar o conteúdo e exibir o progresso (barra de progresso simulada).

4.5. Tela de Progresso (app/(drawer)/(tabs)/progress.tsx)



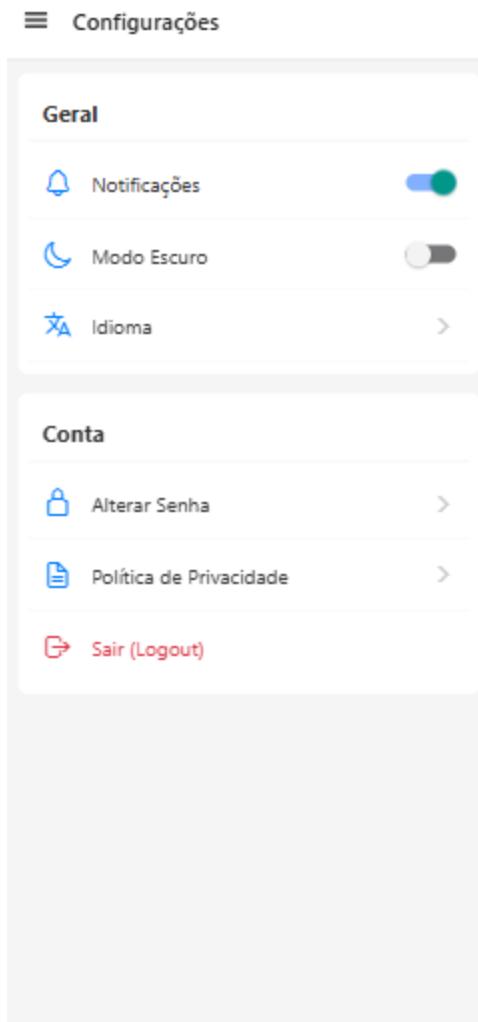
Componentes/Funcionalidades	Aplicação
View e Text	Exibição de métricas e progresso geral (círculo de progresso).
Ionicons	Uso de ícones para representar visualmente as métricas.

4.6. Tela de Perfil (app/(drawer)/profile.tsx)



Componentes/Funcionalidades	Aplicação
TextInput	Exibição e edição de dados do usuário (Nome, Área, Email).
Hooks (useState, useEffect)	Gerenciamento do estado de edição e carregamento inicial de dados.
Persistência de Dados	Simula o carregamento e salvamento do nome do usuário via AsyncStorage.

4.7. Tela de Configurações (app/(drawer)/settings.tsx)



Componentes/Funcionalidades	Aplicação
Switch	Componentes para ligar/desligar funcionalidades (Notificações, Modo Escuro).
TouchableOpacity	Botões para ações como Sair (Logout), que utiliza o AsyncStorage para remover o token e redirecionar para a tela de Login.
useRouter	Hook do Expo Router para navegação programática (redirecionamento no Logout).

5. Validação e Considerações sobre o Ambiente de Execução

O aplicativo **SkillUpPlus 2030+** teve sua lógica de negócios, estrutura de navegação híbrida (Stack, Drawer e Tabs) e persistência de dados validados com sucesso na plataforma **Web**, conforme evidenciado nas capturas de tela apresentadas na seção anterior. O funcionamento na Web comprova a integridade do código JavaScript/TypeScript e a correta implementação dos componentes do React Native e Expo Router.

No entanto, durante os testes em ambiente nativo simulado (**Emulador Android - Pixel 5 / API 33**), identificou-se uma limitação técnica relacionada ao ambiente de execução do **Expo Go**.

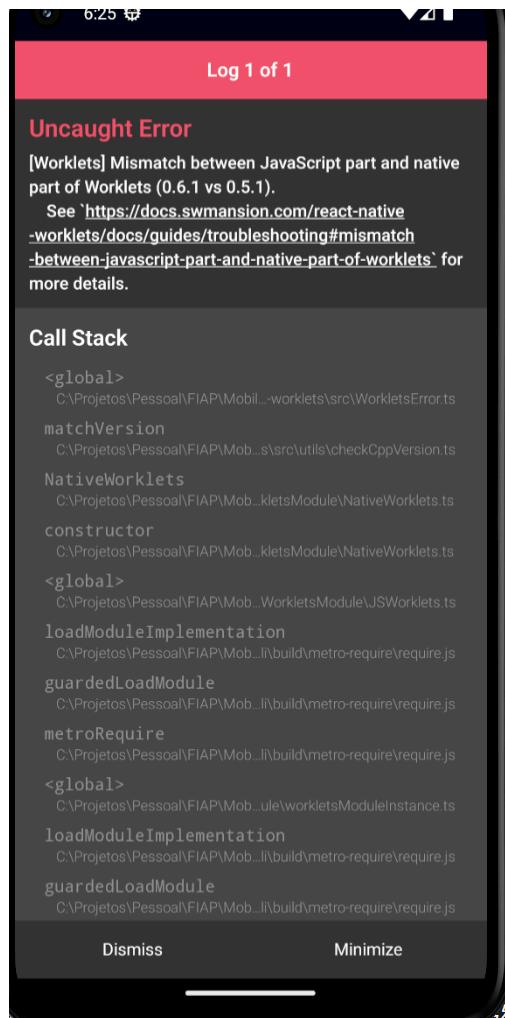
Diagnóstico do Problema: Na primeira execução do aplicativo foi detectada uma inconsistência crítica de versões na biblioteca `react-native-reanimated` (dependência essencial para o Drawer Navigator), gerando o erro "*Worklets Mismatch*". Este erro ocorre devido a um descompasso entre a versão da biblioteca JavaScript instalada no projeto (v0.6.1) e a versão do binário nativo pré-compilado presente no aplicativo Expo Go do emulador (v0.5.1).

ERROR [WorkletsError: [Worklets] Mismatch between JavaScript part and native part of Worklets (0.6.1 vs 0.5.1).

See

`<https://docs.swmansion.com/react-native-worklets/docs/guides/troubleshooting#mismatch-between-javascript-part-and-native-part-of-worklets>` for more details.]

Comportamento da Aplicação: É importante ressaltar que este erro **não bloqueia a lógica da aplicação**. Ao interagir com a interface de depuração do Expo Go e selecionar as opções "**Minimize**" ou "**Dismiss**", o erro é ocultado e é possível prosseguir com a navegação normalmente entre as telas de Login, Home, Trilhas e Perfil. Isso corrobora que a estrutura de rotas e a lógica de negócios estão funcionais, sendo a falha isolada exclusivamente na camada de animação nativa do Drawer.





Medidas de Contorno Realizadas: Foram executados todos os procedimentos padrão de manutenção de ambiente, incluindo:

- Limpeza profunda de cache do Metro Bundler (`npx expo start --clear`).
- Limpeza de dados do aplicativo via ADB (`adb shell pm clear host.exp.exponent`).
- Sincronização de versões do React e reconstrução da pasta `node_modules`.

Apesar dessas medidas, a persistência do cache nativo no emulador impactou a renderização visual inicial, embora a aplicação permaneça funcional na Web e operante no Android após

minimizar o alerta. Abaixo, apresentamos os logs técnicos e as evidências para fins de documentação.