

The Proffy logo is displayed in a large, white, sans-serif font against a solid purple background. The background is decorated with various green geometric shapes, including circles, squares, and crosses, some of which are semi-transparent.

## Versão 2.0 Proffy

Nessa versão nós vamos implementar novas funcionalidades para nossa aplicação que acabamos de desenvolver durante a Next Level Week 2.0 da [Rocketseat](#), a **Proffy**.

Esse vai ser um desafio para elevar ainda mais o seu nível como programador, onde você irá implementar novas funcionalidades para tornar a sua aplicação ainda melhor!

[Layout da aplicação](#)

[Autenticação de usuários](#)

[Recuperação de senhas](#)

[Perfil do proffy](#)

[Splash Screen no React Native com Expo](#)

[Paginação na listagem de proffys](#)

[Exibindo horários disponíveis dos proffys](#)

[Salvando seus proffys favoritos](#)

[Logout da aplicação](#)

[Deploy da aplicação](#)

# Layout da aplicação

Durante o desenvolvimento de todos os desafios aqui propostos, você sempre terá algo para fazer no layout das suas aplicações.

Por exemplo, no desafio de autenticação de usuários você terá que criar a página de login nas versões Web e Mobile.

Então para saber como acessar o Layout dos desafios, veja na página abaixo:



[Layout dos desafios Proffy](#)

## Autenticação de usuários

A primeira nova funcionalidade da sua aplicação será a autenticação de usuários nas aplicações Web e Mobile. Para isso, nós primeiramente devemos criar, no nosso servidor, a funcionalidade para o usuário se cadastrar, inserindo seu e-mail e senha.



**Dica:** Antes de salvar a senha do usuário no banco de dados, lembre-se sempre de criptografar a senha para manter ela segura. Para isso você pode utilizar bibliotecas como o [bcrypt](#).

Depois de permitir a criação de uma senha para todos os usuários cadastrados, você precisa de uma rota para o login do usuário. Essa rota deve validar a senha do usuário no banco de dados e uma boa estratégia para manter o usuário logado por um tempo determinado é utilizar tokens JWT.



**Dica:** Nós temos um post no nosso blog sobre autenticação no Node.js utilizando bcrypt e JWT, deixamos no final dessa sessão em conteúdo extra.

O usuário deve poder logar em sua conta para acessar a aplicação para escolher se ele quer dar aulas, ou mesmo visualizar professores disponíveis, tanto no web quanto no mobile.

💡 **Dica:** Para manter o usuário logado no mobile, caso ele clique na opção "Lembrar-me", você pode salvar o token JWT no LocalStorage do navegador (Web) ou AsyncStorage da sua aplicação mobile. Caso ele não selecione para se Lembrar, você pode apenas mantê-lo autenticado durante uma sessão (até fechar a aplicação).

#### 📖 Conteúdo extra:

- **Vídeo:** [Autenticação no React Native / ReactJS com Context API & Hooks | Masterclass #12](#)
- **Blog:** [Autenticação JWT no React Native com API REST em NodeJS](#)
- **Blog:** [Fluxo de autenticação com Token JWT no React Native](#)

## Recuperação de senhas

Nem sempre um usuário vai se lembrar da sua senha para logar na aplicação, principalmente se há muito tempo que ele acessou pela última vez, então uma funcionalidade essencial é a funcionalidade de recuperar a senha do usuário.

💡 **Dica:** Nos conteúdos extras deixamos alguns conteúdos que você pode utilizar para entender como funcionaria o fluxo de recuperação de senhas no seu servidor.

#### 📖 Conteúdo extra:

- **Vídeo:** [Recuperação de senha e envio de e-mail no Adonis | Behind the Code #02](#)
- **Vídeo:** [\[API NodeJS + Express + Mongo\] Recuperação de senha com NodeMailer | Diego Fernandes](#)

## Perfil do proffy

Após a autenticação de um professor na aplicação WEB, você deve permitir que ele possa exibir o seu perfil e permitir também que ele edite algumas informações do seu cadastro.

Para isso, no seu backend implemente uma rota para alteração do perfil. Ele deve poder alterar as seguintes informações:

- Alterar horário/dias de disponibilidade
- Alterar o número de WhatsApp
- Alterar sua biografia
- Alterar seu valor por hora de aula



**Dica:** No perfil do usuário irá sempre aparecer os dados que ele cadastrou na página "Dar aulas", então sempre ao entrar no seu perfil, busque essas informações do banco de dados para exibi-las nos campos.

## Splash Screen no React Native com Expo

Aqui você deverá configurar a splash screen do app mobile do Proffy. Para isso, nós temos um conteúdo mais que especial sobre Deploy da aplicação mobile com expo, onde ensinamos a configurar a splash screen.



**Conteúdo extra:**

- **Vídeo:** [Gerando APK \(Android\) e IPA \(iOS\) com React Native & Expo | Code/Drops #15](#)

## Paginação na listagem de proffys

O usuário logado na aplicação pode visualizar todos os professores disponíveis na plataforma, mas exibir todos os professores de uma só vez pode ser bastante prejudicial para a performance da sua aplicação.

Para resolver esse problema, implemente uma paginação na listagem de professores disponíveis. Essa paginação deve ser feita ao buscar os itens da sua api, e você pode mandar uma informação pelo corpo da requisição, como por exemplo `"page": 1` para que você saiba quais itens retornar do banco de dados.



**Conteúdo extra:**

**Blog:** [Scroll infinito no React Native](#)

## Exibindo horários disponíveis dos proffys

Na sua listagem de proffys, se você olhar no layout, perceberá que é exibido todos os horários disponíveis dos professores logo abaixo da bio dele. Seu trabalho é implementar essa funcionalidade.

Para isso, você apenas precisa trazer junto aos dados do proffy as informações de seus dias disponíveis fazendo um join com a tabela `class_schedule` para retornar esses dados junto com a listagem.



**Conteúdo extra:**

**Blog:** [SQL no Node.js com Knex.js | Masterclass #13](#)

## Salvando seus proffys favoritos

Agora, você deve salvar seus proffys favoritos no banco de dados! Para isso, você pode criar uma tabela `favorites` que irá salvar o id do seu usuário logado e o id do seu proffy favorito.

Assim, toda vez que você entrar na página de proffys favoritos, você irá ter atualizado os seus proffys que você mais adora!

## Logout da aplicação

Como última funcionalidade, mas não menos importante, você deve implementar uma função de logout na aplicação, para que o usuário possa trocar de conta caso ele deseje.

Aqui, você pode aproveitar o conteúdo sobre o fluxo de autenticação que deixamos na parte de autenticação para entender o funcionamento do Logout.



**Conteúdo extra:**

- **Vídeo:** [Autenticação no React Native / ReactJS com Context API & Hooks | Masterclass #12](#)

## Deploy da aplicação

Agora é a hora de se desafiar a publicar a sua aplicação na WEB para todo mundo poder acessar! Nós criamos um guia **exclusivo** para essa edição da NLW ensinando do zero o processo de deploy da nossa aplicação, e o melhor, você ainda vai aprender a configurar o PostgreSQL na sua aplicação!

 **Conteúdo extra:**

**Blog:** [Deploy NLW](#)