

REACT NATIVE-REACT HOOK FORM

A biblioteca React Hook Form possibilita a criação de formulários tanto em React como em React Native. O problema da criação de formulários nativos é a renderização automática (re-render), esta dependência procura a diminuição do número de renderizações durante a execução do aplicativo.

Documentação oficial:

<https://react-hook-form.com/>

<https://react-hook-form.com/get-started#ReactNative>

Crie um novo projeto:

```
expo init HookForm
```

Escolha a opção blank:

```
Choose a template: » - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
blank (TypeScript) same as blank but with TypeScript
configuration
tabs (TypeScript) several example screens and tabs using
react-navigation and TypeScript
----- Bare workflow -----
minimal bare and minimal, just the essentials to
get you started
```

Abra o projeto no VSCode através do comando abaixo:

```
code HookForm
```

No VSCode, abrir o Terminal com as teclas Alt + F12 e rodar o projeto exemplo, e verificar se tudo foi carregado corretamente.

```
yarn start → servidor expo
```

```
yarn android → para chamar o emulador diretamente
```

Em seguida instale as dependências para realizarmos as navegações:

```
expo install react-hook-form
```

Apague o conteúdo do arquivo App.js pelo o que esta abaixo:

```
import React from "react";
import { Text, View, TextInput, Button, Alert,
StyleSheet } from "react-native";
import { useForm, Controller } from "react-hook-form";

export default function HookForm() {
  const {
    control,
    handleSubmit,
    formState: { errors },
  } = useForm({
    defaultValues: {
      nome: "",
      sobrenome: "",
    },
  });

  const onSubmit = (data) => {
    console.log(data);
    Alert.alert(data.nome + "\n" + data.sobrenome);
  };

  return (
    <View>
      <Controller
        control={control}
        rules={{
          required: true,
        }}
        render={({ field: { onChange, onBlur, value }
      }) => (
        <TextInput
          style={styles.input}
```

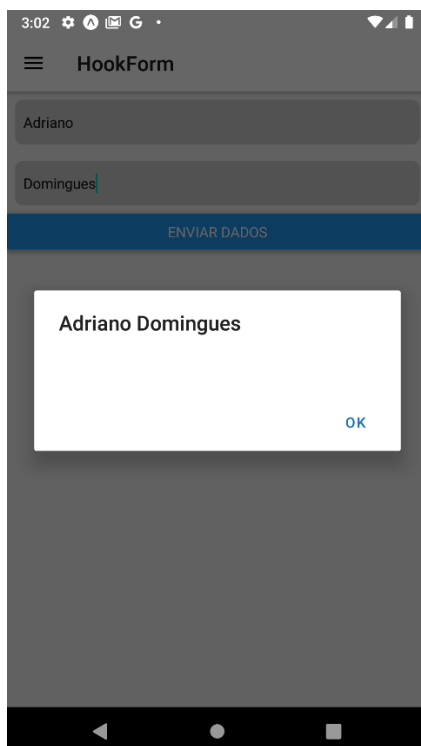
```
        onBlur={onBlur}
        onChangeText={onChange}
        placeholder="Digite seu nome"
        value={value}
      />
    )}
    name="nome"
  />
  {errors.nome && <Text>Campo obrigatório.</Text>}

  <Controller
    control={control}
    rules={{
      maxLength: 100,
    }}
    render={({ field: { onChange, onBlur, value }
  }) => (
    <TextInput
      style={styles.input}
      onBlur={onBlur}
      placeholder="Digite seu sobrenome"
      onChangeText={onChange}
      value={value}
    />
    )}
    name="sobrenome"
  />

  <Button title="Enviar Dados"
onPress={handleSubmit(onSubmit)} />
</View>
);
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#fff",
    alignItems: "center",
    justifyContent: "center",
  },
  input: {
    backgroundColor: "#ccc",
    margin: 8,
    padding: 8,
    minWidth: "70%",
    borderRadius: 8,
  },
});
```

Execute o projeto, insira seu nome completo e clique no botão:



Referência Bibliográfica

[1] <https://react-hook-form.com/> Acessado em 16/11/2021.

[2] <https://www.figma.com/>. Acessado em 16/11/2021.