REACT NATIVE-ASYNCSTORAGE

AsyncStorage é um componente assíncrono e não encriptado de persistência de dados, isto é, podemos guardar informações no próprio celular enquanto o aplicativo estiver instalado no equipamento.

Vamos usar o formulário para guardar o nome e o sobrenome e possibilitar a recuperação para consulta posterior.

Documentação oficial:

https://docs.expo.dev/versions/v42.0.0/sdk/async-storage/

Este tutorial é continuação do React Native – Hook Form.

Instale o AsyncStorage com a execução da linha abaixo, é um componente nativo, mas será removido, por isso necessitamos da instalação abaixo:

```
expo install @react-native-async-storage/async-storage
```

Abaixo temos o App.js resultante do Tutorial mencionado com o React Hook Form:

```
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,
  },
});
```

Devemos realizar a importação do AsyncStorage com a linha abaixo:

import AsyncStorage from '@react-native-async-storage/async-storage';

O AsyncStorage armazena dados do tipo texto, então para guardar dados de um objeto precisamos transforma-lo em um objeto JSON através do JSON.stringify() para salvar os dados e o método JSON.parse() para recuperá-los.

No nosso caso aqui, quando o botão é clicado o método handleSubmit() envia os dados (data) como objeto, então precisamos usar o JSON.stringify() como no exemplo abaixo, para converter os dados no formato JSON (formato texto):

```
const storeData = async (value) => {
  try {
    const jsonValue = JSON.stringify(value)
    await AsyncStorage.setItem('@storage_Key', jsonValue)
  } catch (e) {
    // saving error
  }
}
```

Vamos adicionar o código abaixo no método onSubmit():

```
// Dados serão armazenados no AsyncStorage
    try {
      // Dados serão transformados em um objeto JSON
      const dadosJSON = JSON.stringify(data);
      // Dados transformados serão guardados no
AsyncStorage
      await AsyncStorage.setItem('@dados', dadosJSON);
    } catch (e) {
      // saving error
      Alert.alert(e.message);
    }
```

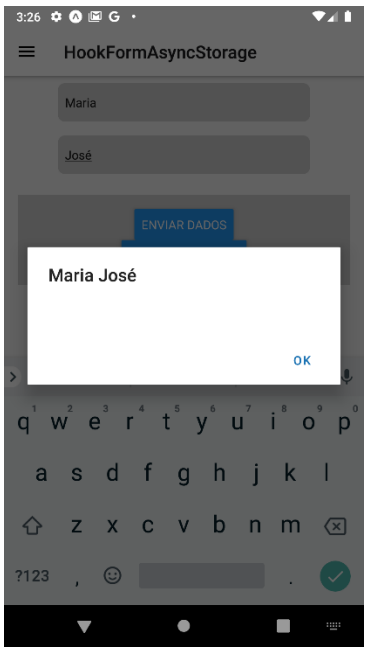
O método setItem() passa dois argumentos, chave '@dados' e valor (data), para recuperar os dados do AsyncStorage precisamos da chave acima definida.

Vamos criar mais um botão no formulário para recuperar os dados gravados e apresentá-los na tela com um Alert:

```
<Button title="Recuperar Dados"
        onPress={async ()=>{
          try {
            const dadosJSONRecuperados = await
AsyncStorage.getItem('@dados');
            if(dadosJSONRecuperados !== null) {
```

```
            // dados gravados no AsyncStorage
            const dados =
JSON.parse(dadosJSONRecuperados);
            Alert.alert(dados.nome + "\n" +
            dados.sobrenome);
          }
        } catch(e) {
          // erro ao ler valores
          Alert.alert(e.message);
        }
      }}
    />
```

Execute o projeto, insira seu nome completo e clique no botão, os dados serão guardados no AsyncStorage:



Abaixo o código completo:

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

```
import AsyncStorage from "@react-native-async-
storage/async-storage";

export default function HookFormAsyncStorage() {
  const {
    control,
    handleSubmit,
    formState: { errors },
  } = useForm({
    defaultValues: {
      nome: "",
      sobrenome: "",
    },
  });
  const onSubmit = async (data) => {
    console.log(data);
    Alert.alert(data.nome + "\n" + data.sobrenome);

    // Dados serão armazenados no AsyncStorage
    try {
      // Dados serão transformados em um objeto JSON
      const dadosJSON = JSON.stringify(data);
      // Dados transformados serão guardados no
AsyncStorage
      await AsyncStorage.setItem("@dados", dadosJSON);
    } catch (e) {
      // saving error
      Alert.alert(e.message);
    }
  };

  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"
```

```
      />

      <View style={styles.caixaInterna}>
        <Button
          style={styles.botao}
          title="Enviar Dados"
          onPress={handleSubmit(onSubmit)}
        />

        <Button
          style={styles.botao}
          title="Recuperar Dados"
          onPress={async () => {
            try {
              const dadosJSONRecuperados = await
AsyncStorage.getItem("@dados");
              if (dadosJSONRecuperados !== null) {
                // dados gravados no AsyncStorage
                const dados =
JSON.parse(dadosJSONRecuperados);
                Alert.alert(dados.nome + "\n" +
dados.sobrenome);
              }
            } catch (e) {
              // erro ao ler valores
              Alert.alert(e.message);
            }
          }}
        />
      </View>
    </View>
  );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#fff",
    alignItems: "center",
    justifyContent: "center",
  },
  caixaInterna: {
    minWidth: "70%",
    backgroundColor: "#ccc",
    alignItems: "center",
    justifyContent: "center",
    padding: 16,
    margin: 16
  },
  input: {
    backgroundColor: "#ccc",
    margin: 8,
    padding: 8,
    minWidth: "70%",
    borderRadius: 8,
    alignSelf: "center",
    justifyContent: "center",
  },
  botao: {
    minWidth: "70%",
    alignSelf: "center",
    justifyContent: "center",
    margin: 8
  },
});
```

Referência Bibliográfica

[1] https://docs.expo.dev/versions/v42.0.0/sdk/async-storage/. Acessado em 24/11/2021.

[2] https://react-native-async-storage.github.io/async-storage/docs/usage/. Acessado em 24/11/2021.