

Firestore e React Native

- Desenvolvimento Mobile
 - Prof. Rone Ilídio

O que é o Firebase

- Plataforma que oferece infraestrutura para desenvolvimento de sistemas Web e Mobile
- Possui várias ferramentas para serem utilizadas por esses aplicativos, exemplos:
 - Realtime Database: banco de dados em tempo real que mantém armazenamento local (offline) e sincronismo de dados (online)
 - Analytics: possibilita análise do comportamento dos usuários
 - Authentication: oferece vários métodos para autenticação, como email/senha, Facebook, etc.
 - Cloud Messaging: notificação
 - Admob: publicidade

Utilizando o Realtime Database - Primeiros Passos

- Crie um projeto chamado firebaseok
react-native init firebaseok
- Instale o Firebase no seu projeto
Entre na pasta firebaseok e digite:
yarn add firebase
- Entre no site do firebase: console.firebase.google.com
 - Será necessária uma conta no Google
- Crie um projeto no site com o nome de ReactNativeBasics

Olá! Este é o Firebase

Ferramentas do Google para criar uma infraestrutura de apps, melhorar a qualidade de apps e desenvolver sua empresa

Criar um projeto

☰ Ver documentos

- Dê um nome
- Aceite o Analytics
- Aceite as condições de uso

Configurando o Projeto

- Adicione seu projeto a um plataforma de desenvolvimento, são oferecidas 4:

- iOS
- Android
- Web
- Unit



- Escolha Web, pois Reac Native utiliza Javascript
- Defina um apelido para sua aplicação (pode ser ReactNativeBasics)
- Copie a configuração, a qual dá acesso a sua aplicação para utilizar o Firebase, e cole no seu código
 - Deve ser colado dentro do método `componentWillMount`
 - Os slides a seguir mostram o que deve ser copiado e onde deve ser colado

Copie a configuração

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.2.1/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.2.1/firebase-analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyCPye4p_NjhIEwQRPntSDpYtsS0t2Joras",
    authDomain: "reactnativebasics-9cd25.firebaseio.com",
    databaseURL: "https://reactnativebasics-9cd25.firebaseio.com",
    projectId: "reactnativebasics-9cd25",
    storageBucket: "reactnativebasics-9cd25.appspot.com",
    messagingSenderId: "904353058508",
    appId: "1:904353058508:web:744c17ae319e5356f57c69",
    measurementId: "G-EF81NW2HWK"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

Cole a configuração no seu App.js

```
import React, {Component} from 'react';
import {View, Text} from 'react-native';
import firebase from 'firebase';
export default class ReactNativeBasics extends Component {
  UNSAFE_componentWillMount(){
    // Your web app's Firebase configuration
    var firebaseConfig = {
      apiKey: 'AlzaSyCPye4p_NjhIEwQRPntSDpYtsS0t2Joras',
      authDomain: 'reactnativebasics-9cd25.firebaseio.com',
      databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',
      projectId: 'reactnativebasics-9cd25',
      storageBucket: 'reactnativebasics-9cd25.appspot.com',
      messagingSenderId: '904353058508',
      appId: '1:904353058508:web:744c17ae319e5356f57c69',
      measurementId: 'G-EF81NW2HWK',
    };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);
  }
}
```

1/2

```
render(){  
  return (  
    <View>  
      <Text>Acesso ao Firebase</Text>  
    </View>  
  );  
}
```

1/2

Obs1: esse código corresponde à uma tela básica, simplesmente escrito Acesso ao Firebase, mais a parte que deve ser colada (em vermelho)

Obs2: lembre-se importar o firebase. No slide anterior isso é feito na 3a linha:

```
import firebase from 'firebase';
```

Criação do Bando de Dados

- Na página do firebase, acesso Desenvolver/Database
- Crie um novo banco de dados clicando no botão “Criar Banco de Dados”
- Na tela que aparecerá, escolha “Iniciar no mode de teste” e clique em Próxima

Criar banco de dados ✕

1 Regras seguras para o Cloud Firestore — 2 Defina o local do Cloud Firestore

Depois de definir sua estrutura de dados, será necessário criar regras para proteger seus dados.
[Saiba mais](#)

☐ **Iniciar no modo bloqueado**
Para tornar seu banco de dados privado, basta negar todas as leituras e gravações

☒ **Iniciar no modo de teste**
Configure rapidamente ao permitir todas as leituras e gravações no seu banco de dados

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write;
    }
  }
}
```

⚠ Qualquer pessoa com sua referência de banco de dados poderá fazer leituras ou gravações no seu banco de dados

Ativar o Cloud Firestore impedirá que você use o Cloud Datastore neste projeto, principalmente no aplicativo associado ao App Engine

Cancelar **Próxima**

Criação do Bando de Dados

Na definição do local de armazenamento, mantenha o padrão e clique Concluído

Criar banco de dados ✕

✓ **Regras seguras para o Cloud Firestore** — 2 **Defina o local do Cloud Firestore**

Sua configuração de local é onde os dados do Cloud Firestore serão armazenados.

⚠ Não será possível alterar o local depois de configurá-lo. Além disso, esse será o local do seu intervalo padrão do Cloud Storage.

[Saiba mais](#)

Local do Cloud Firestore

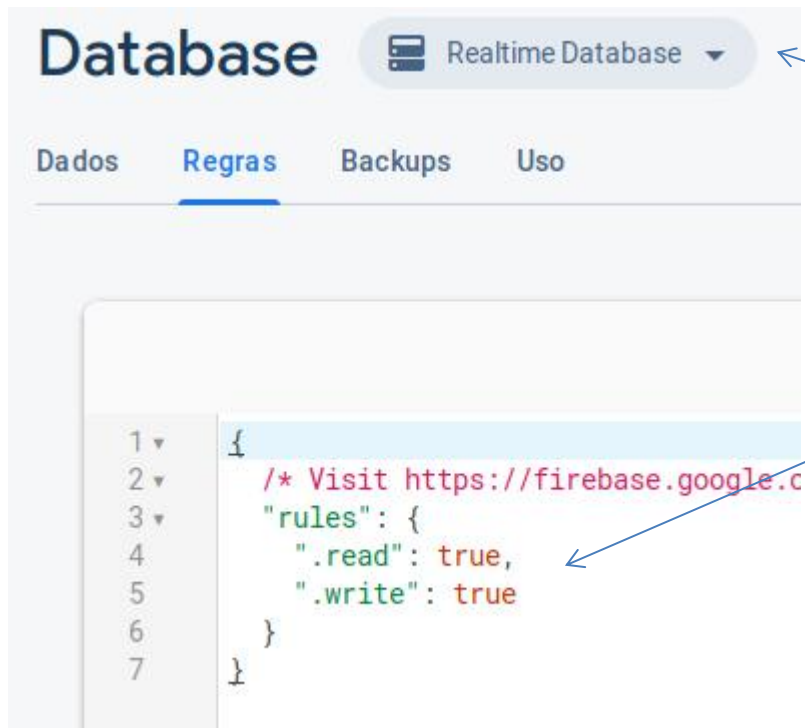
nam5 (us-central) ▼

Ativar o Cloud Firestore impedirá que você use o Cloud Datastore neste projeto, principalmente no aplicativo associado ao App Engine

Cancelar **Concluído**

Criação do Bando de Dados

- Na página do firebase, acesso Desenvolver/Database
- Escolha Realtime Database
- Clique na guia de “Regras”
 - Coloque leitura e escrita como true, de acordo com a figura:



Atenção aqui

e aqui

Inserção de dados

- Modifique o App.js acrescentando o código abaixo no final do UNSAFE_componentWillMount()

```
firebase
  .database()
  .ref('user/0001')  ← Define a estrutura do banco de dados
  .set({
    nome: 'Rone Ilídio',  } ← Dados
    idade: 18,
  })
  .then(() => {
    console.log('Inseriu'); ← Dados inseridos com sucesso
  })
  .catch(() => {
    console.log('Erro'); ← Erro na inserção
  });
```

Obs: o código todo é apresentado a seguir

```
import React, {Component} from 'react';
import {View, Text} from 'react-native';
import firebase from 'firebase';
export default class ReactNativeBasics extends Component {
  UNSAFE_componentWillMount() {
    var firebaseConfig = {
      apiKey: 'AlzaSyCPye4p_NjhIEwQRPNTSDpYtsS0t2Joras',
      authDomain: 'reactnativebasics-9cd25.firebaseio.com',
      databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',
      projectId: 'reactnativebasics-9cd25',
      storageBucket: 'reactnativebasics-9cd25.appspot.com',
      messagingSenderId: '904353058508',
      appId: '1:904353058508:web:744c17ae319e5356f57c69',
      measurementId: 'G-EF81NW2HWK',
    };
    firebase.initializeApp(firebaseConfig);
    firebase.database().ref('user/0001').set({
      nome: 'Rone Ilídio',
      idade: 18,
    }).then(() => {
      console.log('Inseriu');
    }).catch(() => {
      console.log('Erro');
    });
  }
}
```

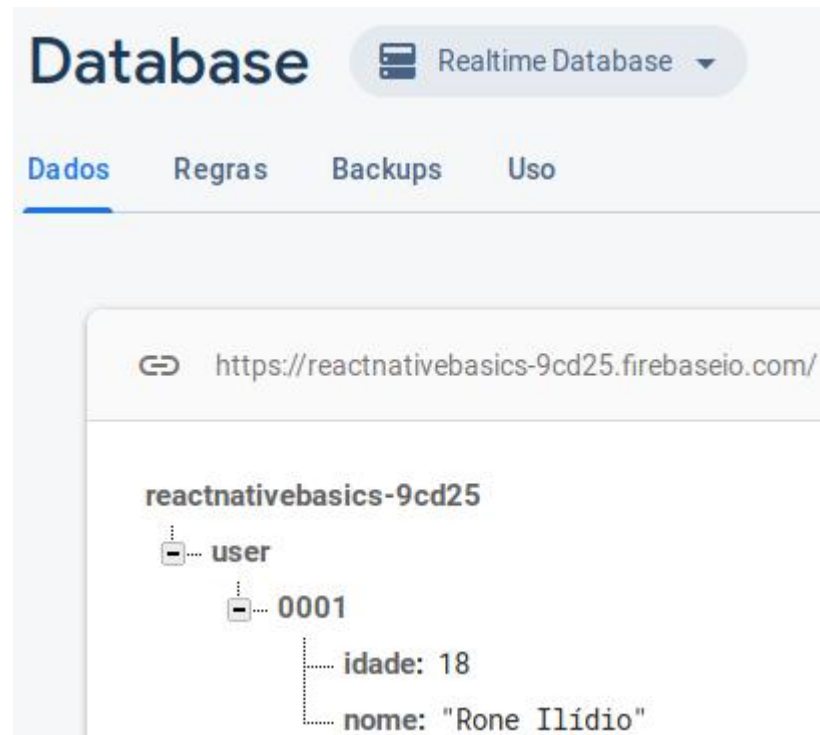
```
render() {  
  return (  
    <View>  
      <Text>Acesso ao Firebase</Text>  
    </View>  
  );  
}
```

- Para inserir mais dados, basta mudar a estrutura e os dados e executar novamente, exemplo:

```
firebase  
  .database()  
  .ref('user/0002')  
  .set({  
    nome: 'Zé Mané',  
    idade: 19,  
  })  
  .then(() => {  
    console.log('Inseriu');  
  })  
  .catch(() => {  
    console.log('Erro');  
  });
```

Execute seu aplicativo

- Após a execução, entre na página do Firebase e acesso Desenvolver/Database guia Dados
- O resultado será como na figura abaixo:



Inserção de Novos Dados

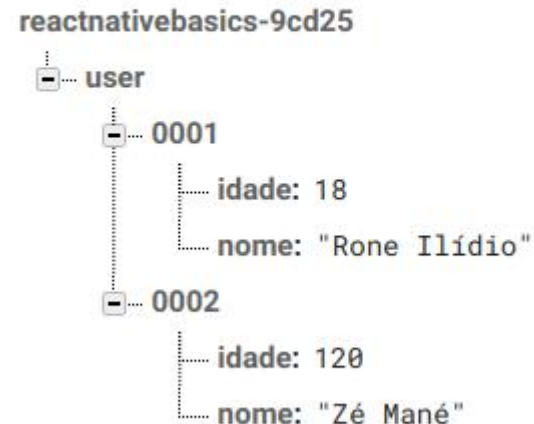
- Modifique os dados a serem inseridos, de acordo com o código a seguir, e execute.
- Modifique quantas vezes quiser, cada execução criará um novo registro

firebase

```
.database()  
.ref('user/0002')  
.set({  
  nome: 'Zé Mané',  
  idade: 120,  
})  
.then(() => {  
  console.log('Inseriu');  
})  
.catch(() => {  
  console.log('Erro');  
});
```

Resultado

<https://reactnativebasics-9cd25.firebaseio.com/>



Busca por Dados

- No exemplo a seguir, o app busca os dados que estão no caminho 'user/0001' após o clique do botão.
- Obs: o código a seguir exibe um warning, por isso ele foi ignorado com a importação de YellowBox, que permite ignorar o warning de “Setting a timer”.

```
import React, {Component} from 'react';
import {View, Text, Button} from 'react-native';
import firebase from 'firebase';
import { YellowBox } from 'react-native';
export default class ReactNativeBasics extends Component {
  construct() {
    YellowBox.ignoreWarnings(['Setting a timer']);
  }
  state = { dados: {nome: '', idade: 0} };
  UNSAFE_componentWillMount() {
    // Your web app's Firebase configuration
    var firebaseConfig = {
      apiKey: 'AlzaSyCPye4p_NjhlEwQRPNTSDpYtsS0t2Joras',
      authDomain: 'reactnativebasics-9cd25.firebaseio.com',
      databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',
      projectId: 'reactnativebasics-9cd25',
      storageBucket: 'reactnativebasics-9cd25.appspot.com',
      messagingSenderId: '904353058508',
      appId: '1:904353058508:web:744c17ae319e5356f57c69',
      measurementId: 'G-EF81NW2HWK',
    };
    firebase.initializeApp(firebaseConfig);
  }
}
```

```
buscarDados = () => {  
  firebase  
    .database()  
    .ref('user/0001')  
    .once('value', data => {  
      this.setState({dados: data.toJSON()});  
      console.log(this.state.dados.nome);  
    });  
};  
render() {  
  return (  
    <View>  
      <Text>Acesso ao Firebase</Text>  
      <Button  
        title="Buscar"  
        onPress={() => {  
          this.buscarDados();  
        }}  
      />  
      <Text>Nome: {this.state.dados.nome}</Text>  
      <Text>Idade: {this.state.dados.idade}</Text>  
    </View>  
  );  
}
```



Busca os dados no
banco de dados

Dados em Realtime

- O exemplo a seguir exhibe os dados do banco de dados na tela
- Se os dados alterarem, o aplicativo também altera automaticamente.

```
import React, {Component} from 'react';
import {View, Text, Button} from 'react-native';
import firebase from 'firebase';
import {YellowBox} from 'react-native';
export default class ReactNativeBasics extends Component {
  construct() {
    YellowBox.ignoreWarnings(['Setting a timer']);
  }
  state = {
    dados: {nome: '', idade: 0},
  };
  UNSAFE_componentWillMount() {
    // Your web app's Firebase configuration
    var firebaseConfig = {
      apiKey: 'AlzaSyCPye4p_NjhIEwQRPNtSDpYtsS0t2Joras',
      authDomain: 'reactnativebasics-9cd25.firebaseio.com',
      databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',
      projectId: 'reactnativebasics-9cd25',
      storageBucket: 'reactnativebasics-9cd25.appspot.com',
      messagingSenderId: '904353058508',
      appId: '1:904353058508:web:744c17ae319e5356f57c69',
      measurementId: 'G-EF81NW2HWK',
    };
```

```
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
firebase
  .database()
  .ref('user/0001')
  .on('value', data => {
    this.setState({dados: data.toJSON()});
    console.log(this.state.dados.nome);
  });
}
render() {
  return (
    <View>
      <Text>Acesso ao Firebase</Text>
      <Text>Nome: {this.state.dados.nome}</Text>
      <Text>Idade: {this.state.dados.idade}</Text>
    </View>
  );
}
```

Dados em Realtime

- `on('value', data =>` Faz a busca periódica no banco de dados
- Se os dados forem alterados, a tela altera automaticamente.
- Para alterar, acesse o banco de dados

<https://reactnativebasics-9cd25.firebaseio.com/>

reactnativebasics-9cd25

user

0001

idade: 20

nome: "Rone Ilídio da Silva"

0002

idade: 120

nome: "Zé Mané"

Clique no campo e altere um dos campos do usuário 0001.

Atualizar Dados

- O exemplo a seguir modifica o nome do usuário 0002 com os dados fornecidos pelo usuário
- O resultado pode ser visto diretamente no banco de dados

```
firebase
  .database()
  .ref('user/0002')
  .update({
    nome: 'Novo nome',
  });
```

```
import React, {Component} from 'react';
import {View, Text, Button} from 'react-native';
import firebase from 'firebase';
export default class ReactNativeBasics extends Component {
  state = {
    dados: {nome: '', idade: 0},
  };
  UNSAFE_componentWillMount() {
    // Your web app's Firebase configuration
    var firebaseConfig = {
      apiKey: 'AlzaSyCPye4p_NjhIEwQRPNtSDpYtsS0t2Joras',
      authDomain: 'reactnativebasics-9cd25.firebaseio.com',
      databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',
      projectId: 'reactnativebasics-9cd25',
      storageBucket: 'reactnativebasics-9cd25.appspot.com',
      messagingSenderId: '904353058508',
      appId: '1:904353058508:web:744c17ae319e5356f57c69',
      measurementId: 'G-EF81NW2HWK',
    };
  };
```

```
firebase.initializeApp(firebaseConfig);
```

```
firebase
```

```
  .database()
```

```
  .ref('user/0002')
```

```
  .update({
```

```
    nome: 'Novo nome',
```

```
  });
```

```
}
```

```
render() {
```

```
  return (
```

```
    <View>
```

```
      <Text>Atualização</Text>
```

```
    </View>
```

```
  );
```

```
}
```

```
}
```

Atualiza somente o campo especificado, os demais são mantidos

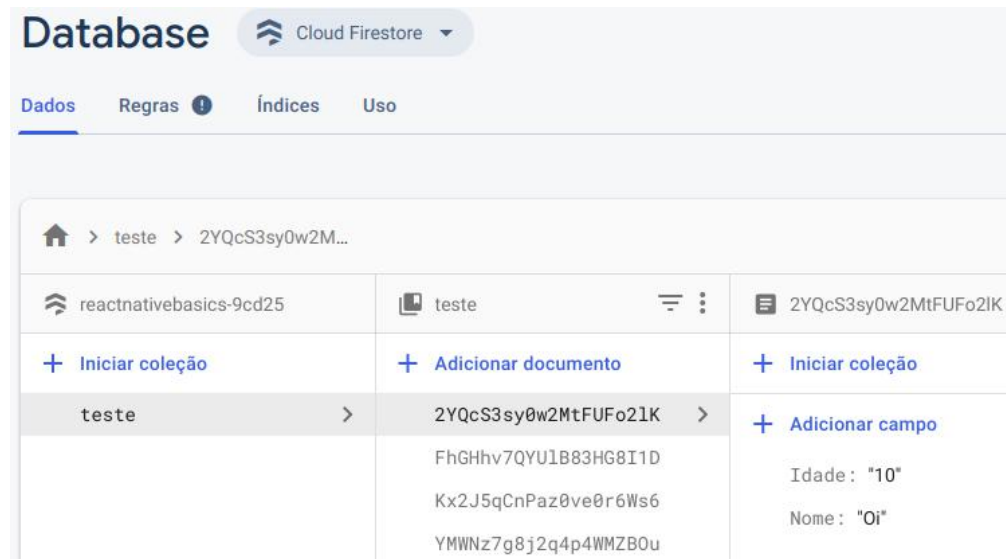
Excluindo

- Utilizar o método remove
- Remove o caminho, sendo possível especificar o caminho
- Somente um campo:
 - `firebase.database().ref('user/0002/nome').remove();`
- O registro todo:
 - `firebase.database().ref('user/0002').remove();`

Firestore

Firestore

- O Firebase Firestore é um banco de dados NoSQL
- Ele armazena documentos, divididos em coleções
- Na tela inicial vemos:
 - as coleções à esquerda
 - cada coleção possui um conjunto de documentos (ao centro)
 - cada documento possui vários dados (à direita)



Firestore

- O projeto a seguir armazena dados de pessoas em uma coleção chamada teste.
- Crie um projeto chamado firestore
react-native init firestore
- Instale o Firebase no seu projeto
Entre na pasta firestore e digite:
yarn add firebase
- No console do Firebase, clique em Database e escolha Cloud Firestore
- Crie o arquivo ./config/firebase.js com o código a seguir:

```
import firebase from 'firebase/app';  
import 'firebase/firestore';
```

```
// Your web app's Firebase configuration  
var firebaseConfig = {  
  apiKey: 'AlzaSyCPye4p_NjhIEwQRPntSDpYtsS0t2Joras',  
  authDomain: 'reactnativebasics-9cd25.firebaseio.com',  
  databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',  
  projectId: 'reactnativebasics-9cd25',  
  storageBucket: 'reactnativebasics-9cd25.appspot.com',  
  messagingSenderId: '904353058508',  
  appId: '1:904353058508:web:744c17ae319e5356f57c69',  
  measurementId: 'G-EF81NW2HWK',  
};  
// Initialize Firebase  
var app = firebase.initializeApp(firebaseConfig);  
//export default firebase;  
const db = firebase.firestore(app);  
export default db;
```

Chave do
Firebase

Objeto associado ao Firestore

Firebase Firestore

- No arquivo App.js, crie a interface gráfica
- Essa tela terá dois TextInput e um botão, que ao ser pressionado cria um novo documento no banco de dados
- Veja o código a seguir

```
import React, {Component} from 'react';
import db from './config/firebase';
import {StyleSheet, SafeAreaView, View, TextInput, Button} from 'react-native';
export default class App extends Component {
  state = {
    Nome: "",
    Idade: "",
  };
  constructor(props) {
    super(props);
    this.ref = db.collection('teste');
  }
  render() {
    console.disableYellowBox = true;
    return (
      <SafeAreaView style={styles.container}>
        <View style={styles.inputContainer}>
          <TextInput
            value={this.state.Nome}
            placeholder="Nome"
            style={styles.textInput}
            onChangeText={value => this.setState({Nome: value})}
          />
        </View>
      </SafeAreaView>
    );
  }
}
```

```
<TextInput
  value={this.state.Idade}
  placeholder="Idade"
  style={styles.textInput}
  onChangeText={value => this.setState({Idade: value})}
/>
<Button
  onPress={() =>
    this.ref.add({Nome: this.state.Nome, Idade: this.state.Idade})}
  title="Adicionar Pessoa"
  color="#841584"
/>
</View>
</SafeAreaView>
);
}
}
const styles = StyleSheet.create({
  container: { flex: 1, backgroundColor: '#F5FCFF', },
  inputContainer: { margin: 30, },
  textInput: { height: 40, textAlign: 'center', color: '#333333',
    marginBottom: 10, fontSize: 14, borderWidth: 1,
    borderBottomColor: '#111111',
  },
});
```

Firestore

- `import db from './config/firebase';` --> importa o objeto que manipula o banco de dados
- `this.ref = db.collection('teste');` --> Obtém o objeto para manipular a coleção 'teste'
 - Obs: o nome de uma coleção inexistente leva à criação dessa coleção
- `this.ref.add({Nome: this.state.Nome, Idade: this.state.Idade})` --> adiciona um novo documento.
- Obs: o ID dos Objetos são gerados automaticamente

Busca no Firestore

- A aplicação a seguir busca todos os documentos de uma coleção
- Vamos colocar a configuração do banco de dados em um arquivo separado
 - Crie a pasta `./config` (obs: uma pasta na raiz de seu projeto)
 - Crie o arquivo `./config/firebase.js`
 - Coloque o código que está no próximo slide
- Após isso, modifique o arquivo `App.js`
 - Coloque o código que está dois slides a seguir

```
import firebase from 'firebase/app';  
import 'firebase/firestore';
```

```
// Your web app's Firebase configuration  
var firebaseConfig = {  
  apiKey: 'AlzaSyCPye4p_NjhIEwQRPNTSDpYtsS0t2Joras',  
  authDomain: 'reactnativebasics-9cd25.firebaseio.com',  
  databaseURL: 'https://reactnativebasics-9cd25.firebaseio.com',  
  projectId: 'reactnativebasics-9cd25',  
  storageBucket: 'reactnativebasics-9cd25.appspot.com',  
  messagingSenderId: '904353058508',  
  appId: '1:904353058508:web:744c17ae319e5356f57c69',  
  measurementId: 'G-EF81NW2HWK',  
};
```

```
// Initialize Firebase  
var app = firebase.initializeApp(firebaseConfig);  
//export default firebase;  
const db = firebase.firestore(app);  
export default db;
```

```
import React, {Component} from 'react';
import db from './config/firebase';
import { StyleSheet, SafeAreaView, View, TextInput, Button, Text, } from
'react-native';
export default class App extends Component {
  constructor(props) {
    super(props);
    this.ref = db.collection('teste');
  }
  busca = () => {
    this.ref
      .get()
      .then(snapshot => {
        snapshot.docs.forEach(element => {
          var nome = element.data().Nome;
          var idade = element.data().Idade;
          console.log(nome + ": " + idade);
        });
      });
  };
};
```

```
render() {  
  console.disableYellowBox = true;  
  return (  
    <SafeAreaView style={styles.container}>  
      <View style={styles.inputContainer}>  
        <Button  
          onPress={() => this.busca()}  
          title="Buscar Todas as Pessoas"  
          color="#841584"  
        />  
      </View>  
    </SafeAreaView>  
  );  
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#F5FCFF',
  },
  inputContainer: {
    margin: 30,
  },
  textInput: {
    height: 40,
    textAlign: 'center',
    color: '#333333',
    marginBottom: 10,
    fontSize: 14,
    borderWidth: 1,
    borderBottomColor: '#111111',
  },
});
```

Busca Com Condição Where

- A aplicação a seguir estabelece condições de busca: buscará os documentos cujo nome é informado pelo usuário
- Copie o código a seguir no arquivo App.js
- Importante: essa aplicação também utiliza o arquivo ./config/firebase.js apresentado posteriormente

```
import React, {Component} from 'react';
import db from './config/firebase';
import { StyleSheet, SafeAreaView, View, TextInput, Button, Text, } from
'react-native';
export default class App extends Component {
  state = {
    Nome: "",
  };
  constructor(props) {
    super(props);
    this.ref = db.collection('teste');
  }
  busca = () => {
    this.ref
      .where('Nome', '==', this.state.Nome)
      .get()
      .then(snapshot => {
        snapshot.docs.forEach(element => {
          var nome = element.data().Nome;
          var idade = element.data().Idade;
          console.log(nome + ": " + idade);
        });
      });
  };
};
```

```
render() {  
  console.disableYellowBox = true;  
  return (  
    <SafeAreaView style={styles.container}>  
      <View style={styles.inputContainer}>  
        <TextInput  
          value={this.state.Nome}  
          placeholder="Nome"  
          style={styles.textInput}  
          onChangeText={value => this.setState({Nome: value})}  
        />  
        <Button  
          onPress={() => this.busca()}  
          title="Buscar Pessoas por Nome"  
          color="#841584"  
        />  
      </View>  
    </SafeAreaView>  
  );  
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#F5FCFF',
  },
  inputContainer: {
    margin: 30,
  },
  textInput: {
    height: 40,
    textAlign: 'center',
    color: '#333333',
    marginBottom: 10,
    fontSize: 14,
    borderWidth: 1,
    borderBottomColor: '#111111',
  },
});
```