Tasks com Navigation 5

Versões utilizadas

- React-Native 0.62.2
- React Navigation 5.x

Ambiente

- Seguindo a versão oficial do React Native CLI Quickstart
- Dependências adicionais a serem instaladas:
  - Para o navigation:

```
npm install @react-navigation/native react-native-reanimated react-native-gesture-handler
  react-native-screens react-native-safe-area-context @react-native-community/masked-view
  @react-navigation/stack @react-navigation/drawer
```

  - Demais dependências:

```
npm install @react-native-community/async-storage @react-native-community/datetimepicker
  axios moment react-native-gravatar react-native-vector-icons
```

Movendo código

Trazer os arquivos do projeto oficial que estava na versão 4 do Navigation, para esse novo projeto (Colando todos na raiz)

- index.js
- react-native.config.js
- pasta "src"
- pasta "assets"

Executar o comando ***npx react-native link***

Caso queira os arquivos já na versão final, aqui estão eles.

Arquivos alterados

Navigator

Esse foi o arquivo mais impactado uma vez que ele é o responsável pela definição da navegação. A forma de uso é praticamente a mesma (exceto pela remoção do SwitchNavigator), mas a sintaxe e imports mudaram bastante.

```
import React from 'react'
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import { createDrawerNavigator } from '@react-navigation/drawer';

import Auth from './screens/Auth'
import TaskList from './screens/TaskList'

import AuthOrApp from './screens/AuthOrApp'
import Menu from './screens/Menu'
import commonStyles from './commonStyles'

const Stack = createStackNavigator();
const Drawer = createDrawerNavigator();

const menuConfig = {
    labelStyle: {
        fontFamily: commonStyles.fontFamily,
        fontWeight: 'normal',
        fontSize: 20
    },
    activeTintColor: '#080',
}

const DrawerNavigator = props => {
    const { email, name } = props.route.params
    return (
        <Drawer.Navigator drawerContentOptions={menuConfig}
              drawerContent={(props) => <Menu {...props} email={email} name={name} />}>
            <Drawer.Screen name="Today" options={{ title: 'Hoje' }}>
                {props => <TaskList {...props} title='Hoje' daysAhead={0} />}
            </Drawer.Screen>
            <Drawer.Screen name="Tomorrow" options={{ title: 'Amanhã' }}>
                {props => <TaskList {...props} title='Amanhã' daysAhead={1} />}
            </Drawer.Screen>
            <Drawer.Screen name="Week" options={{ title: 'Semana' }}>
                {props => <TaskList {...props} title='Semana' daysAhead={7} />}
            </Drawer.Screen>
            <Drawer.Screen name="Month" options={{ title: 'Mês' }}>
                {props => <TaskList {...props} title='Mês' daysAhead={30} />}
            </Drawer.Screen>
        </Drawer.Navigator>
    );
};

const AuthNavigator = () => {
    return (
        <Stack.Navigator screenOptions={{ headerShown: false }}>
            <Stack.Screen name="AuthOrApp" component={AuthOrApp} />
            <Stack.Screen name="Auth" component={Auth} />
```

```
            <Stack.Screen name="Home" component={DrawerNavigator} />
        </Stack.Navigator>
    );
};

const Navigator = () => {
    return (
        <NavigationContainer>
            <AuthNavigator />
        </NavigationContainer>
    );
};

export default Navigator;
```

Menu

Essa classe precisou de ajustes para receber os dados do usuário via props. Outro ajuste necessário foi a navegação pois sem o Switch foi necessário resetar a pilha para não permitir voltar à tela da aplicação depois de fazer logout. Exibindo apenas os pontos que devem ser inseridos/atualizados.

```
import { DrawerContentScrollView, DrawerItemList } from '@react-navigation/drawer'
import { CommonActions } from '@react-navigation/native';


const logout = () => {
    delete axios.defaults.headers.common['Authorization']
    AsyncStorage.removeItem('userData')
    props.navigation.dispatch(
        CommonActions.reset({
            index: 0,
            routes: [
                {
                    name: 'Auth',
                },
            ],
        })
    )
}

return (
    <DrawerContentScrollView>
        <View style={styles.header}>
            <Text style={styles.title}>Tasks</Text>
            <Gravatar style={styles.avatar}
                options={{
                    email: props.email,
                    secure: true
                }} />
            <View style={styles.userInfo}>
                <Text style={styles.name}>
                    {props.name}
                </Text>
```

```
                <Text style={styles.email}>
                    {props.email}
                </Text>
            </View>
            <TouchableOpacity onPress={logout}>
                <View style={styles.logoutIcon}>
                    <Icon name='sign-out' size={30} color='#800' />
                </View>
            </TouchableOpacity>
        </View>
        <DrawerItemList {...props} />
    </DrawerContentScrollView>
)
```

AuthOrApp

Também precisou de ajuste para limpar a pilha, evitando poder retornar até a tela de loading. Exibindo apenas os pontos que devem ser inseridos/atualizados.

```
import { CommonActions } from '@react-navigation/native';

componentDidMount = async () => {
    const userDataJson = await AsyncStorage.getItem('userData')
    let userData = null

    try {
        userData = JSON.parse(userDataJson)
    } catch (e) {
        // userData está inválido
    }

    if (userData && userData.token) {
        axiodefaults.headers.common['Authorization'] = `bearer ${userData.token}`
        // this.props.navigation.navigate('Home', userData)
        this.props.navigation.dispatch(
            CommonActions.reset({
                index: 0,
                routes: [
                    {
                        name: 'Home',
                        params: userData,
                    },
                ],
            })
        );
    } else {
        // this.props.navigation.navigate('Auth')
        this.props.navigation.dispatch(
            CommonActions.reset({
                index: 0,
                routes: [
                    {
                        name: 'Auth',
                    },
                ],
```

```
            })
        )
    }
}
```

Auth

Também precisou de ajuste para limpar a pilha, evitando poder retornar até a tela de login sem realizar logout. Exibindo apenas os pontos que devem ser inseridos/atualizados.

```
import { CommonActions } from '@react-navigation/native';

signin = async () => {
        try {
            const res = await axios.post(`${server}/signin`, {
                email: this.state.email,
                password: this.state.password
            })

            AsyncStorage.setItem('userData', JSON.stringify(res.data))
            axios.defaults.headers.common['Authorization'] = `bearer ${res.data.token}`
            // this.props.navigation.navigate('Home', res.data)
            this.props.navigation.dispatch(
                CommonActions.reset({
                    index: 0,
                    routes: [
                        {
                            name: 'Home',
                            params: res.data,
                        },
                    ],
                })
            )
        } catch (e) {
            showError(e)
        }
    }
```

O projeto completo está compartilhado no repositório oficial do curso.

Para uso no iOS

Para executar o projeto no iOS, alguns passos adicionais podem ser necessários:

```
sudo gem install cocoapods
cd ios
pod install
cd ..
react-native run-ios
```

Possíveis problemas

Duplicated Files

Se aparecer uma mensagem do tipo

```
error: Multiple commands produce '/Users/.../Debug-iphonesimulator/tasks2020.app/AntDesign.ttf':
```

Abrir o projeto no XCode e, clicando na raiz do projeto, em `Build Phases`, abrindo a opção `[CP] Copy Pods Resources`, apagar os arquivos das fonts no `Input Files` e `Output Files`

RTCBridge required dispatch_sync...

Se aparecer esse erro, editar o arquivo `AppDelegate.m`, adicionando esse trecho nos imports

```
#if RCT_DEV
  #import <React/RCTDevLoadingView.h>
#endif
```

e esse trecho abaixo da linha que começa `RCTBridge *bridge...`

```
#if RCT_DEV
    [bridge moduleForClass:[RCTDevLoadingView class]];
#endif
```

Error building: 'React/RCTBridgeDelegate.h' file not found; 'React/RCTBridge.h' file not found

1. Apagar a pasta `ios/build`
2. No XCode, entrar no menu `File -> Project Settings -> Build System`, e em `Build System` selecionar `Legacy Build System`
3. Executar `react-native run-ios`