



CODERUSH



QUIZ

Discentes:
Kauany Miranda
William Matsunaga

Proposta

- Aprendizado com quiz;
- Interação com exemplos práticos: perguntas, respostas e feedback;
- Reforço do aprendizado mostrando erros ao final, permitindo refazê-los;
- Aumento na dificuldade do conteúdo a cada fase.

Por que um Quiz?

- Forçar o cérebro a recuperar informações ativamente da memória;
- Repetições de testes ajudam a fixar conteúdos;
- Identificação de conteúdo que ainda não dominou a partir do feedback imediato;
- Consciência do seu próprio conhecimento e compreensão.

O que o app ensina?

Conteúdo:

- Estrutura e Layout;
- texto e Interação;
- Estilos (StyleSheet e inline);
- Navegação;

- JSX;
- Hooks;
- AsyncStorage;
- Requisições API.

Conceitos aplicados

- useState: utilizado em cada pergunta para armazenar a opção selecionada;
- Screens: HomeScreen, LevelsScreen e QuizScreen;
- NavigationContainer: navegação stack usada na troca de telas ;
- Animated: usado nos modais.

Animated

```
const fadeAnim = useRef(  
  new Animated.Value(0)  
)  
.current;  
  
function openModal(setter) {  
  setter(true);  
  Animated.timing(fadeAnim, {  
    toValue: 1,  
    duration: 200,  
    useNativeDriver: true,  
  }).start();  
}  
  
function closeModal(setter) {  
  Animated.timing(fadeAnim, {  
    toValue: 0,  
    duration: 150,  
    useNativeDriver: true,  
  }).start(() => setter(false));  
}
```

- A constante fadeAnim inicia totalmente transparente;
- O modal é exibido pelo setter(true);
- A opacidade vai de 0 - 1 para o modal abrir
- A opacidade vai de 1 - 0 para o modal fechar;
- O modal é escondido pelo setter(false).

ScrollView e TouchableOpacity

```
return (  
  <ScrollView contentContainerStyle={  
    [ styles.container, { flexGrow: 1 } ] >  
    <View style={styles.header}>  
      <Text style={styles.title}>Níveis</Text>  
      <TouchableOpacity  
        style={styles.backButton}  
        onPress={() => navigation.goBack()}  
      >  
        <Image  
          source={require('../assets/home2.png')}  
          style={{ width: 40, height: 40 }}  
          resizeMode="contain"  
        />  
      </TouchableOpacity>  
    </View>  
  )
```

- ScrollView permite rolar para cima e para baixo caso o conteúdo ultrapasse o tamanho da tela e organiza o conteúdo para ocupar o espaço completo.
- TouchableOpacity cria um botão clicável, respondendo ao toque e executando a navegação.

Navigation

```
const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen
          name="Home"
          component={HomeScreen}
        />
        <Stack.Screen
          name="Levels"
          component={LevelsScreen}
        />
        <Stack.Screen
          name="Quiz"
          component={QuizScreen}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

- `createNativeStackNavigator()` cria um navegador de pilha, onde as telas funcionam como páginas empilhadas;
- `<NavigationContainer>` envolve toda a estrutura da navegação, sem ele a navegação não funciona;
- `<Stack.Navigator />` define todas as telas do app.



OBRIGADO



GITHUB: [CODERUSH](#)

Discentes:
Kauany Miranda
William Matsunaga