CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADO DESENVOLVIMENTO MOBILE

“Aplicativo Mobile SkinQuiz: Diagnóstico de Tipos de Pele para a Clínica La Belle”

Nome: Leticia Ramiro de Abreu - RA: 24001834

Nome: Luana Aparecida Cardoso - RA: 1012023100720

SÃO JOÃO DA BOA VISTA, SETEMBRO - 2025

Relatório Técnico e de Testes – Aplicativo SkinQuiz (React Native)

Relatório técnico

1.Introdução

Este relatório apresenta a especificação do aplicativo **SkinQuiz**, uma solução mobile desenvolvida em **React Native com TypeScript**. O sistema tem como objetivo proporcionar ao usuário uma experiência interativa e intuitiva para a identificação do tipo de pele: oleosa, mista ou seca e a partir desse diagnóstico, sugerir procedimentos estéticos personalizados oferecidos pela **Clínica La Belle**.

---

2. Objetivo do Sistema

O sistema tem como finalidade:

- Disponibilizar um quiz interativo para identificar o tipo de pele do usuário;
- Fornecer recomendações personalizadas de tratamentos estéticos;
- Facilitar o contato com a clínica por meio de um atalho direto para o WhatsApp.

---

3. Tecnologias Utilizadas

- **React Native** – Desenvolvimento mobile android;
- **TypeScript** – Tipagem estática e maior segurança no código;
- **React Navigation (Native Stack)** – Gerenciamento de rotas e navegação entre telas;
- **React Hooks (useState)** – Controle de estados internos;
- **StyleSheet** – Organização e reaproveitamento de estilos.

---

4. Estrutura Funcional

O aplicativo SkinQuiz é organizado em três telas principais, gerenciadas por um navegador de pilha (Stack Navigator) implementado no componente AppNavigator. Essa estrutura garante a navegação sequencial e intuitiva entre a introdução, o quiz e a exibição dos resultados.

4.1 AppNavigator

- Utiliza o NavigationContainer e createNativeStackNavigator da biblioteca React Navigation;
- Define as rotas Intro, Quiz e Result;
- Controla o fluxo de navegação entre as telas;
- Remove cabeçalhos padrão (headerShown: false) para manter a identidade visual personalizada do app.

4.2 IntroScreen

- Exibe o nome da clínica e botão de contato via WhatsApp;
- Mostra imagem ilustrativa de capa;
- Apresenta uma breve descrição do quiz;
- Botão “Começar Quiz” → inicia o questionário e navega para a próxima tela.

4.3 QuizScreen

- Exibe perguntas sequenciais sobre características da pele;
- Utiliza useState para armazenar respostas do usuário;
- Função calculateResult determina o tipo de pele predominante;
- Ao concluir, navega para ResultScreen passando o resultado como parâmetro.

4.4 ResultScreen

- Recebe o resultado (oleosa, mista ou seca) via route.params;
- Função getRecommendation que gera as recomendações personalizadas;
- Exibe o tipo de pele identificado, lista de tratamentos recomendados;
- Botão “Refazer” → retorna para a tela inicial (IntroScreen).

---

5. Práticas Adotadas

- Separação clara de responsabilidades entre telas;
- Uso de tipagem segura (RootStackParamList) no React Navigation;
- Implementação de componentes reutilizáveis (QuestionCard no quiz);
- Tratamento de exceções em chamadas externas (Linking.openURL com catch);

Relatório de Testes

1. Plano de Teste

1.1. Introdução

Objetivo:
Garantir que o aplicativo **SkinQuiz** funcione corretamente, validando a exibição de perguntas, o registro de respostas do usuário e a apresentação da tela de resultados e o redirecionamento para o WhatsApp da clínica estética La Belle para contato direto.

Escopo:

- Testes de **integração** (navegação entre telas).
- Testes **funcionais** (fluxo completo do quiz).
- Testes de **validação de respostas**.

1.2. Objetivos e Tarefas

Objetivos:

- Validar a navegação entre telas (Intro → Quiz → Result).
- Confirmar que as perguntas e alternativas são exibidas corretamente..
- Garantir que o resultado final seja mostrado corretamente.

Tarefas:

- Configurar o ambiente de teste (emulador ou dispositivo físico).
- Executar testes unitários, integração e sistema.
- Registrar resultados e incidentes.

1.3. Recursos, Ambiente e Cronograma

Recursos Humanos:

- QA + desenvolvedores.

Ferramentas:

- Emulador/Dispositivo React Native.
- Jest + React Native Testing Library.

Cronograma:

- Preparação: 15 dias
- Execução: 7 dias
- Relatório: 3 dias

1.4. Critérios de Início e Conclusão

Início:

- Código estável entregue.
- Ambiente configurado.

Conclusão:

- Todos os casos de teste executados.
- Incidentes documentados.
- Aprovação final da equipe.

1.5. Riscos e Mitigações

Riscos:

- Perguntas não renderizarem corretamente.
- Erros na contagem de resultado.
- Navegação pode falhar entre telas.

Mitigações:

- Testes unitários para lógica de pontuação.
- Testes manuais em dispositivos.

---

2. Especificação de Design de Teste (Test Design Specification)

2.1. Objetivos do Teste

Verificar se o quiz entrega a experiência correta ao usuário: perguntas → respostas → resultado.

2.2. Estratégia de Teste

- **Unitários:** lógica de cálculo da pontuação.
- **Integração:** navegação entre telas.
- **Sistema:** fluxo completo do quiz.
- **Critérios de Aceitação:**
- Perguntas e alternativas exibidas corretamente.
- Tela de resultado exibida ao final.

2.3. Dados de Teste

Exemplo: “Você sente sua pele repuxar ou ressecar após a lavagem?”

Alternativas:

- Sim, com frequência
- Apenas em algumas áreas
- Não, minha pele permanece hidratada

Resposta esperada:

A) Pele seca

B) Pele mista

C) Pele oleosa

Esses dados de teste simulam as respostas que o usuário daria no Quiz, permitindo validar:

- Se as perguntas e alternativas são exibidas corretamente.
- Se o app mapeia as respostas para o tipo de pele correto.
- Se o resultado final exibido corresponde às respostas escolhidas.

---

3. Especificação de Casos de Teste

CT-QUIZ-001 – Integração da Tela Inicial (IntroScreen)

Objetivo:

- Garantir que a tela inicial (IntroScreen) exiba os elementos principais e redirecione corretamente para o quiz.
- Renderizar a IntroScreen.
- Validar se o nome da clínica e a descrição do quiz aparecem.
- Clicar no botão “Começar Quiz”.

Resultado Esperado:

- O botão “Começar Quiz” navega para a tela QuizScreen.

Status: Sucesso.

CT-QUIZ-002 – Integração do Quiz (QuizScreen)

Objetivo:

- Validar a exibição de perguntas, alternativas e navegação entre elas.
- Renderizar a QuizScreen.
- Verificar se a primeira pergunta e suas alternativas são exibidas corretamente.
- Selecionar uma alternativa e validar se a próxima pergunta é exibida.
- Repetir até a última pergunta e verificar se a função de cálculo do resultado é chamada.

Resultado Esperado:

- As perguntas e alternativas aparecem corretamente.
- O fluxo de seleção percorre todas as perguntas até a finalização.

Status: Sucesso.

CT-QUIZ-003 – Integração da Tela de Resultado (ResultScreen)

Objetivo:

- Validar a exibição do resultado final, recomendações e navegação ao refazer.
- Renderizar a ResultScreen com um parâmetro de resultado simulado (ex.: “oleosa”).
- Verificar se o tipo de pele e recomendações correspondentes são exibidos.

- Clicar no botão “Refazer” e validar se navega para IntroScreen.

Resultado Esperado:

- O resultado e recomendações são exibidos corretamente.
- O botão “Refazer” retorna à tela inicial.

Status: Sucesso.

CT-QUIZ-004 – Integração entre Telas (Intro → Quiz → Result)

Objetivo:

- Validar que a navegação entre as telas funciona corretamente no fluxo principal.
- Iniciar o app na IntroScreen.
- Clicar em “Começar Quiz”.
- Responder às perguntas até o fim.
- Confirmar que a ResultScreen é exibida com o resultado.

Resultado Esperado:

- O fluxo completo de navegação ocorre sem falhas, mantendo os parâmetros corretos entre telas.

Status: Sucesso.

CT-QUIZ-005 – Teste Unitário da Função calculateResult

Objetivo:

- Validar a lógica da função calculateResult de forma isolada, sem depender de navegação ou interface.
- Cenários Testados:
    1. Maioria das respostas é “oleosa” → retorna “oleosa”.
    2. Maioria das respostas é “seca” → retorna “seca”.
    3. Empate entre tipos → retorna “mista”.

Resultado Esperado:

- A função retorna corretamente o tipo de pele predominante.
- Status: Sucesso.

Status: Sucesso.

---

4. Procedimentos de Teste

- Rodar o app no emulador/dispositivo.
- Navegar entre as telas (Intro → Quiz → Result).
- Responder às perguntas com diferentes combinações.
- Validar se o resultado e recomendações correspondem às respostas.

---

5. Relatório de Transmissão

- Itens transmitidos: Código do Quiz (React Native).
- Versão: 1.0.0
- Responsáveis: Dev + QA.

ID	Data/Hora	Responsável	Resultado	Observações
CT-QUIZ-001	14/09/2025 10h	QA1	Sucesso	IntroScreen exibiu corretamente nome da clínica e botão “Começar Quiz”.
CT-QUIZ-002	16/09/2025 10h30	QA1	Sucesso	Pergunta inicial e alternativas renderizadas corretamente.
CT-QUIZ-003	17/09/2025 11h	QA1	Falha	Resultado exibiu “mista” quando respostas eram predominantemente “oleosa”.
CT-QUIZ-004	17/09/2025 11h30	QA1	Sucesso	Fluxo completo Intro → Quiz → Result validado.
CT-QUIZ-005	17/09/2025 12h	QA1	Sucesso	Função calculateResult retornou tipos de pele conforme cenários de teste.

---

6. Relatório de Incidentes

INC-001 – Erro de Renderização de Ícones no Teste de integração

- Descrição: Durante a execução dos testes com Jest, foi registrado o aviso "An update to Icon inside a test was not wrapped in act(...)".

- Impacto: O aviso não impediu a execução dos testes, todos passaram com sucesso. Contudo, pode gerar ruído nos logs e dificultar a análise de falhas reais.
- Causa: O componente FontAwesome do pacote @expo/vector-icons utiliza setState internamente ao ser renderizado, o que dispara a mensagem de aviso.
- Solução aplicada: Os testes foram ajustados para mockar o componente de ícone, evitando atualizações de estado desnecessárias e eliminando o warning.
- Status: Resolvido.

---

7. Relatório de Teste Funcional

7.1 Objetivo

O objetivo dos testes funcionais foi validar se o aplicativo SkinQuiz atende aos requisitos definidos, verificando o comportamento do sistema em situações reais de uso.

Foram avaliadas as seguintes funcionalidades principais:

- Exibição da tela inicial (IntroScreen).
- Execução do quiz (QuizScreen).
- Cálculo e exibição do resultado (ResultScreen).
- Navegação entre telas (Intro → Quiz → Result).
- Redirecionamento para WhatsApp da clínica.

7.3 Procedimentos Realizados

- O aplicativo utilizado foi o Expo Go
- Testes realizados com diferentes combinações de respostas no quiz.
- Simulação de cenários de empate e predominância de tipos de pele.
- Verificação da navegação entre telas por meio dos botões disponíveis.
- Validação da abertura de link externo para contato via WhatsApp.

7.4 Resultados Obtidos

- IntroScreen: exibiu corretamente nome da clínica, descrição do quiz e botão “Começar Quiz”.
- QuizScreen: perguntas renderizadas conforme esperado; navegação entre questões funcional.
- ResultScreen: apresentou o tipo de pele correspondente às respostas, exibiu recomendações e permitiu retorno à tela inicial.
- Função calculateResult: em cenários de maioria, retornou o tipo de pele corretamente.

- Integração com WhatsApp: atalho abriu o aplicativo externo corretamente, permitindo contato direto com a clínica.

Os testes funcionais demonstraram que o aplicativo SkinQuiz atende às necessidades de uso, garantindo a navegação e exibição de resultados esperados pelo usuário. A única limitação identificada foi a inconsistência no cálculo do resultado em situações de empate, que será ajustada em versões futuras.

---

8. Conclusão

- O aplicativo SkinQuiz atendeu aos requisitos funcionais principais, garantindo o fluxo Intro → Quiz → Resultado sem falhas críticas.
- O cálculo do tipo de pele apresentou inconsistências em cenários de respostas majoritariamente “oleosa”, apontando necessidade de revisão da lógica de contagem.
- O incidente de renderização de ícones foi resolvido com a aplicação de mock nos testes, garantindo maior estabilidade nos relatórios.

Melhorias futuras:

- Adicionar integração com banco de dados para salvar o histórico de resultados dos usuários, permitindo acompanhar evolução da pele ao longo do tempo.
- Ajustar a função calculateResult para cenários de empate/maioria.
- Permitir login opcional (Google/WhatsApp) para personalizar a experiência do usuário.
- Ampliar a base de perguntas para maior precisão diagnóstica.
- Implementar acessibilidade (voice over, contraste de cores) para atender usuários com necessidades especiais.