

# Lumicheck

Universidade de Aveiro  
Departamento de Comunicação e Arte  
**Relatório**

Rodrigo Graça [107634], Vânia Morais [102383], Tiago Reis [94643],  
Beatriz Bancaleiro [124411], Henrique Rosa [124445]

Unidade Curricular:  
**Tecnologias e Desenvolvimento Web**  
Prof. Carlos Santos e Prof. David Oliveira

Janeiro 2025

# Conteúdo

<b>1 LumiCheck</b>	<b>2</b>
1.1 Contextualização e apresentação do conceito do produto	2
1.2 Objetivos do produto	2
1.3 Aspetos necessários para o desenvolvimento	2
1.3.1 Tarefas	3
1.3.2 Troféus	3
<b>2 Tecnologias Utilizadas</b>	<b>5</b>
2.1 Frameworks e Bibliotecas Principais	5
2.2 UI e Estilização	5
2.3 Funcionalidades Nativas	5
2.4 Animação e Gestos	6
2.5 Mapas e Localização	6
2.6 Gestão de Código e Colaboração	6
<b>3 Arquitetura de Componentes</b>	<b>7</b>
<b>4 Organização do repositório e fluxo de trabalho no GitHub</b>	<b>8</b>
4.1 Organização LumiCheck-App	8
4.2 Gestão das <i>branches</i>	9
4.3 Gestão de <i>Pull Requests</i>	10
<b>5 Desenvolvimento do <i>Frontend</i></b>	<b>11</b>
5.1 Componentes e Hierarquia	11
5.2 <i>Routing</i> e Navegação	13
<b>6 Instruções de Configuração e Execução do Projeto</b>	<b>16</b>
6.1 Como Executar o Projeto	16
6.2 Como Visualizar o Projeto	16
<b>7 Reflexão crítica e Conclusão</b>	<b>17</b>
7.1 Reflexão Crítica	17
7.2 Conclusão	17
<b>A 30 Tarefas</b>	<b>19</b>
<b>B Possíveis Desafios</b>	<b>21</b>

# Capítulo 1

## LumiCheck

### 1.1 Contextualização e apresentação do conceito do produto

Este mestrado tem como base um elemento de avaliação interdisciplinar, que consiste num projeto de **Challenge Based Learning**. Foi realizada uma parceria com a empresa Blip, que nos proporcionou o desafio de criar uma plataforma que ajude as pessoas com usos excessivos, podendo até correr o risco de serem vícios, mais relacionado com o ambiente digital. Posteriormente o tema sofreu alterações de vícios para *well-being*, passando, assim, a ser mais abrangente.

Com este desafio em mente criamos o projeto, com o nome de **Lumicheck**.

Esta aplicação tem duas vertentes principais, a primeira, a de um utilizador direto, onde no início escolhe que tipo de uso excessivo o preocupa. Podendo ser redes sociais, apostas, fazer demasiadas compras, entre outros, e responde a um questionário sobre o seu estado atual e com que frequência utiliza essas plataformas. Daí irá ser monitorado durante um período que será definido posteriormente. Durante esse período, o utilizador receberá tarefas e perguntas da Lumi, a mascote da aplicação, onde irá receber conquistas após realizar essas tarefas. Na aplicação o utilizador conseguirá ver os seus dados e se tem tido melhorias. Também será fornecida a informação de psicólogos da zona onde se encontra o utilizador e de outros contactos que possam ajudar.

A segunda vertente da aplicação é mais relacionada com um utilizador que está preocupado com outro alguém, onde responde algumas perguntas, daquilo que ele consegue observar sobre a outra pessoa. E a partir daí recebe informações e dicas de como pode ajudar outra pessoa.

### 1.2 Objetivos do produto

Esta aplicação não pode dar um diagnóstico oficial ao utilizador, pois existem vários fatores e o mesmo só pode ser dado por um psicólogo.

Numa primeira instância, a aplicação pretende **realizar uma avaliação inicial**, onde verifica o estado emocional do utilizador, e também serve para consciencializar o utilizador das suas ações que podem ou não ter tendências para comportamentos que geram vícios.

Outro objetivo é **oferecer um apoio interativo e motivacional**, que faça com que o utilizador não desista. E por último, **facilitar na procura de informação** para o utilizador obter a ajuda correta.

### 1.3 Aspetos necessários para o desenvolvimento

Antes de avançar para o desenvolvimento, houve a necessidade de tomar algumas decisões importantes para garantir que o projeto atendesse aos objetivos definidos e fosse relevante para os utilizadores. No

final do semestre, realizámos uma reunião online com a estudante de Mestrado em Psicologia, Joana Dias, que nos forneceu diversos insights valiosos sobre a área da dependência digital.

Entre as informações mais relevantes destacam-se:

- A necessidade de um teste inicial, como o **Teste de Avaliação de Adição à Internet (IAT)**<sup>1</sup> para avaliar as condições do utilizador;
- A criação de **tarefas progressivas** para motivar o utilizador a melhorar os seus hábitos digitais;
- A inclusão de **artigos educativos** sobre literacia digital, como uma ferramenta de apoio e sensibilização.

Estas decisões foram cruciais para moldar o desenvolvimento da aplicação, garantindo que ela abordasse os problemas centrais de forma eficaz e amigável para o utilizador.

### 1.3.1 Tarefas

Ao arranjar mecanismos para o utilizador reduzir a sua dependência, deparamo-nos com a ideia de fornecer tarefas ao utilizador para ele as executar, e assim promover hábitos mais saudáveis que o façam passar mais tempo longe dos ecrãs.

Com a ajuda da lista de ações promotoras de bem-estar, “Leve Consigo”, fornecida por ordem dos psicólogos, criamos uma lista de 30 tarefas, que estão presentes no **Anexo A**.

No entanto, recebemos algumas sugestões por parte de uma profissional da área, que se encontram no **Anexo B**, que consiste em haver uma temática por semana e as tarefas estarem de acordo com esse tema.

Estas tarefas são necessárias para a aplicação pois ajudam a promover a gamificação, e mantém o utilizador conectado à mesma.

### 1.3.2 Troféus

A nossa aplicação conta com um elemento de gamificação : Um sistema de troféus que servem como recompensa por certas ações que o utilizador desempenha. Isto foi feito de modo a fazer com que o utilizador tenha uma interação mais interessante e envolvente, cativante e apelativa. Cada troféu é ganho depois de ser executada uma determinada ação - as ações dos troféus foram definidas com base não só nas tarefas diárias, mas também em conquistas que fossem pertinentes e alinhadas com o progresso do utilizador. O nome, o ícone e a cor de cada um dos troféus foram escolhidos deliberadamente para que, de alguma forma, representassem a ação do respetivo troféu. Ao todo, elaborámos 10 troféus, sendo eles os seguintes, com as suas respetivas ações:

- **Início Promissor:** Reduzir o uso de uma *app* considerada viciante em pelo menos 30 minutos durante um dia.
- **Progresso Considerável:** Reduzir o uso médio de uma *app* considerada viciante em 1h por dia durante uma semana.
- **Primeiro Passo:** Completar o teste inicial.
- **Dia Detox:** Passar um dia completo sem redes sociais/jogos/fazer compras.
- **Marco das 20:** Responder a 20 perguntas da Lumi.
- **Dedicado:** Cumprir todas as tarefas diárias durante uma semana.
- **Perfeccionista:** Cumprir todas as tarefas diárias durante um mês.
- **A ”Penantes”:** Sair à rua para fazer uma caminhada em 5 dias distintos.

---

<sup>1</sup>Questionário de 20 perguntas desenvolvido para avaliar o impacto da utilização da internet no comportamento e na rotina do utilizador, classificando o nível de dependência como leve, moderado ou severo.

- **Bom Dia Alegria:** Não usar o telemóvel nos primeiros 30 minutos após acordar durante 3 dias consecutivos.
- **AutoConsciente:** Usar a *app* todos os dias durante uma semana e ver os relatórios semanalmente.

Abaixo estão as representações visuais dos troféus, aquilo que o utilizador vai ver no produto final:



Figura 1.1: Design dos troféus que um utilizador pode conquistar

As cores usadas nos troféus foram escolhidas com atenção a como estas contrastavam com o fundo da interface da aplicação, onde iriam aparecer, de modo a manter uma boa harmonização de cores. Já para o formato dos troféus, foi usada a silhueta da Lumi, a nossa mascote, de modo a tornar o visual dos troféus característico à nossa aplicação.

## Capítulo 2

# Tecnologias Utilizadas

Este capítulo apresenta as tecnologias usadas no desenvolvimento do nosso projeto, detalhando as *frameworks*, bibliotecas e ferramentas que compõem a sua arquitetura e suportam as suas funcionalidades.

### 2.1 *Frameworks* e Bibliotecas Principais

1. **React Native:** Foi escolhida como base para o desenvolvimento da aplicação. É uma tecnologia amplamente adotada para criar aplicações móveis multiplataforma, garantindo desempenho próximo ao nativo.
2. **Expo:** Foi escolhida para simplificar a integração de funcionalidades nativas e o processo de desenvolvimento. O *Expo* é uma plataforma que simplifica o desenvolvimento de aplicações em *React Native*, oferecendo configuração rápida, APIs prontas para funcionalidades comuns (câmara, notificações, etc.) e teste em tempo real com a aplicação **Expo Go**. Permite atualizações OTA (*Over-the-Air*) e reduz a complexidade de lidar com código nativo. Embora tenha limitações em projetos com requisitos nativos avançados, a tecnologia *Expo* é uma boa alternativa para quem procura agilidade e eficiência no desenvolvimento.

### 2.2 UI e Estilização

1. **TailwindCSS e NativeWind:** Ferramentas que proporcionam a estilização baseada em classes utilitárias, promovendo uma abordagem declarativa e eficiente no design de interfaces.
2. **React Native Elements:** Biblioteca com componentes de UI reutilizáveis e personalizáveis, como botões, *inputs* e *cards*.
3. **React Native Vector Icons:** Oferece suporte a ícones vetoriais, amplamente utilizados na interface para melhorar a experiência do utilizador.
4. **React Native SVG:** Permite a manipulação de gráficos vetoriais escaláveis, essenciais para elementos gráficos mais dinâmicos.
5. **Expo Google Fonts:** Integrado para utilizar a fonte Quicksand, fonte esta que faz parte da identidade visual da aplicação.

### 2.3 Funcionalidades Nativas

1. **Expo Location:** Possibilita o acesso à localização do dispositivo, necessário para a funcionalidade do mapa com profissionais da área da psicologia.
2. **Expo Linear Gradient:** Utilizado para criar gradientes nos elementos visuais, de modo a apresentar uma estética mais acolhedora e de acordo com os protótipos *Hi-Fi*.

3. ***Expo Splash Screen:*** Garante uma tela de carregamento inicial personalizada, melhorando a primeira impressão do utilizador.
4. ***Expo Status Bar:*** Configuração e personalização da barra de status do dispositivo para alinhamento com a interface.

## 2.4 Animação e Gestos

1. ***React Native Reanimated:*** Implementado para criar animações fluídas e responsivas, essenciais para as funções de *swipe* na parte inicial da aplicação.
2. ***React Native Gesture Handler:*** Suporte para gestos, como deslizar e tocar, ampliando a interatividade da aplicação.
3. ***React Native Safe Area Context:*** Garante que os elementos da interface respeitam as margens seguras dos dispositivos mais recentes/modernos.

## 2.5 Mapas e Localização

1. ***React Native Maps:*** Integrado para exibir mapas interativos e informações geográficas relevantes aos utilizadores.

## 2.6 Gestão de Código e Colaboração

Usamos a plataforma **GitHub** para armazenar, versionar e colaborar no código do projeto. O **GitHub** facilitou o trabalho em equipa, permitindo que diferentes elementos do grupo desenvolvessem funcionalidades de forma simultânea, com a gestão eficiente de *branches* e fusão de *pull requests*. Além disso, foi fundamental para manter um histórico detalhado do desenvolvimento e resolver conflitos de código durante o progresso desta parte.

## Capítulo 3

# Arquitetura de Componentes

A arquitetura de componentes foi projetada para garantir modularidade, escalabilidade e organização no desenvolvimento do nosso projeto.

A aplicação foi estruturada utilizando uma hierarquia de componentes que segue os princípios de *Atomic Design*<sup>1</sup> dividindo-os em diferentes níveis de granularidade:

1. **Componentes de Página (Pages):** Representam as páginas principais da aplicação, como a página inicial (HomePage), página de perfil (ProfilePage) e outras. Estes componentes são responsáveis por consolidar os elementos da interface e lidar com a lógica de navegação.
2. **Componentes de UI (Organismos e Moleculares):** Incluem elementos visuais maiores, como formulários, gráficos e outros blocos funcionais. São compostos por átomos menores (botões, ícones, etc.), mas funcionam como unidades independentes dentro das páginas.
3. **Ícones e Arquivos SVG:** Arquivos SVG, como *check.svg* e *trophygold.svg*, são usados para elementos gráficos estáticos ou dinâmicos na interface, contribuindo para a consistência visual e identidade da app.

O diagrama em anexo (de modo a ser visível devido ao seu tamanho), presente no final do documento, ilustra a hierarquia dos componentes. Este facilita a visualização da estrutura do projeto, o que traz uma maior organização para a equipa.

---

<sup>1</sup>**Atomic Design** é uma metodologia de design que organiza componentes em diferentes níveis de granularidade, como átomos, moléculas, organismos, *templates* e páginas, facilitando a escalabilidade e a manutenção do projeto.



## Capítulo 4

# Organização do repositório e fluxo de trabalho no GitHub

### 4.1 Organização LumiCheck-App

Como já falado anteriormente, usamos a plataforma **GitHub** para fazer o controlo de versões e partilhar o nosso projeto. Primeiro começamos por criar uma organização que está dividida em três repositórios: *Frontend*, *Backend* e *Documentation*, como presente na figura 4.1

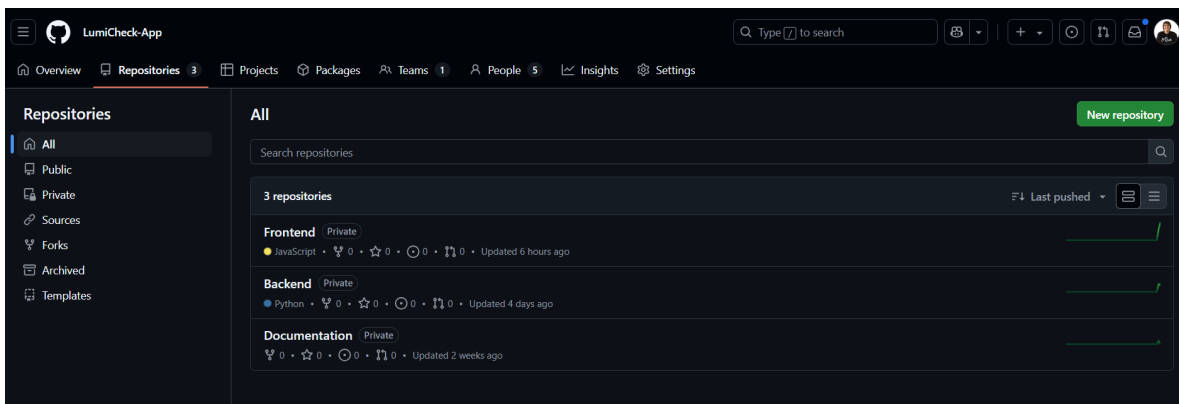


Figura 4.1: Repositórios presentes na organização do projeto

Focando no repositório *Frontend*, temos a organização demonstrada pela figura 4.2

assets	help page done	2 days ago
src	help page minor changes	yesterday
.gitignore	endpoint tarefas testing	2 days ago
README.md	setup completo	2 weeks ago
app.json	routes basicas feitas	2 weeks ago
babel.config.js	try put tailwind	2 weeks ago
metro.config.js	navbar done	2 weeks ago
package-lock.json	Tirar contactos smepre bugados	5 days ago
package.json	Tirar contactos smepre bugados	5 days ago
postcss.config.js	try put tailwind	2 weeks ago
tailwind.config.js	trophy page almost complete	last week

Figura 4.2: Organização das pastas no repositório do frontend

A pasta **assets** armazena recursos estáticos como imagens, ícones e até ficheiros *json*. Já a pasta **src** contém o código fonte da aplicação e está dividida em duas partes: **components** e **pages**. Estes dois diretórios contêm os componentes e páginas, respetivamente, que serão explicados em detalhe mais à frente. Os restantes ficheiros presentes no repositório **Frontend** são ficheiros de configuração, como *package.json* onde são geridas as dependências; o *.gitignore*, que diz ao *GitHub* quais ficheiros ou pastas este deve ignorar; e o *README.md*, onde está descrito como correr o projeto.

## 4.2 Gestão das *branches*

A gestão de *branches* no repositório de *frontend* foi cuidadosamente estruturada para facilitar o desenvolvimento colaborativo e garantir a qualidade do código. Na figura 5.4, estão presentes todas as *branches* usadas até agora.

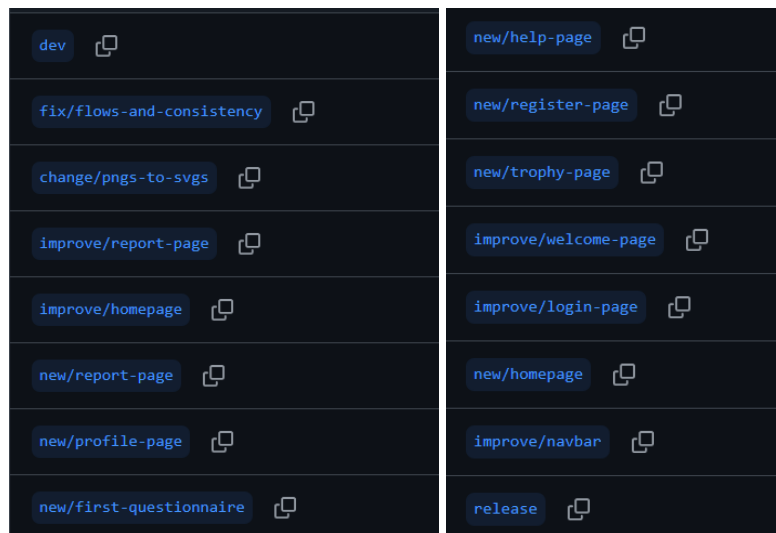


Figura 4.3: Branches usadas durante esta fase do projeto

### Estrutura Principal das *Branches*:

- **dev:** É a *branch* principal de desenvolvimento do repositório. É nela que todas as novas funcionalidades, correções de *bugs* e melhorias são integradas antes de serem promovidas para a *branch* de lançamento (*release*). As mudanças realizadas pelos membros da equipa são desenvolvidas em *branches* específicas e, após revisão e aprovação, são fundidas na *dev*. Este fluxo garante que a *branch* principal de desenvolvimento esteja sempre atualizada, permitindo a integração contínua e o trabalho simultâneo de diferentes membros da equipa sem conflitos.
- **release:** É destinada à criação de versões estáveis do projeto. É nesta *branch* que consolidamos as alterações aprovadas e finalizamos o desenvolvimento antes de disponibilizar uma versão para os utilizadores finais. Esta abordagem permite que sejam feitos os últimos ajustes, garantindo que as versões lançadas estejam funcionais e livres de erros críticos.

### *Branches* de Funcionalidades:

Para cada nova página ou conjunto de páginas, uma *branch* específica é criada a partir da *branch dev*. Exemplos incluem:

- **new/report-page:** Implementação da nova página de relatórios.
- **new/profile-page:** Desenvolvimento da página de perfil do utilizador.
- **new/trophy-page:** Implementação da página para exibir troféus e conquistas.

### ***Branches* de Melhorias:**

Estas *branches* são utilizadas para refinamentos de funcionalidades ou correção de problemas:

- **improve/homepage:** Melhorias na página inicial para corrigir problemas ou adicionar novas funcionalidades.
- **improve/report-page:** Ajustes na página de relatórios para otimizar a experiência do utilizador.
- **improve/navbar:** Alterações na barra de navegação para torná-la mais funcional e visualmente atraente.

### ***Branches* Temporárias:**

Além das *branches* principais, algumas *branches* são criadas para tarefas específicas:

- **fix/flows-and-consistency:** Correções de fluxos e inconsistências no código.
- **change/pngs-to-svgs:** Substituição de imagens PNG por SVG para melhorar a escalabilidade e o desempenho.

## **4.3 Gestão de *Pull Requests***

Para facilitar o trabalho colaborativo e manter um histórico claro das mudanças, todas as novas funcionalidades e correções de problemas foram implementadas por meio de pull requests.

Após o desenvolvimento numa *branch* específica, o membro da equipa que a desenvolveu cria um *pull request* (PR) com o nome da *branch*. Outro elemento da equipa analisa o PR e aprova-o caso este vá de acordo com o que se pretende. Por fim, após aprovação, o PR é integrado na *branch dev*. Caso haja conflitos, o autor do PR é responsável por resolvê-los antes da integração.

De modo a organizar os PRs, foram usadas *tags* como *fix*, *improve* e *new*. Estas etiquetas permitem que a equipa identificasse rapidamente o propósito de cada alteração submetida, como correções de *bugs* (*fix*), melhorias incrementais (*improve*) ou novas funcionalidades adicionadas ao projeto (*new*). Esta categorização ajudou não só a priorizar revisões, mas também a planear de forma mais eficiente as integrações nas *branches* principais.

## Capítulo 5

# Desenvolvimento do *Frontend*

### 5.1 Componentes e Hierarquia

No projeto, a estrutura é iniciada a partir do ficheiro *index.js*, que regista e carrega o *App.js*. Este ficheiro funciona como o ponto de partida para toda a aplicação, onde é configurada a navegação, incluindo a barra de navegação (*navbar*) e os fluxos de navegação definidos através de *stacks*. A aplicação divide-se em páginas com e sem *navbar*. As páginas com *navbar* estão agrupadas num *stack* chamado *Home Tabs*, que contém cinco separadores principais acessíveis através da barra de navegação.

Além destas cinco páginas principais, existem outras páginas acessíveis a partir de links dentro das páginas da *navbar*, como, por exemplo, a página de edição de perfil, que é alcançada a partir do separador *Perfil*. Esta página é gerida dentro do *stack Home Tabs* e também está associada a um *stack* específico do perfil.

#### Organização dos Componentes:

Os componentes e as páginas estão organizados em pastas separadas dentro da pasta *src*:

- **components:** Contém os componentes reutilizáveis e específicos, que foram modularizados para evitar a sobrecarga de código nos ficheiros das páginas.
- **pages:** Contém as diferentes páginas da aplicação.

Embora a reutilização de componentes seja limitada em algumas partes, a modularização foi aplicada sempre que possível para garantir um código mais organizado e legível. Componentes como *Card* das *Tasks*, *Achievements* (troféus), Progresso dos Troféus e *Cards* de Seleção (usados no perfil e subpáginas) foram criados para manter uma estrutura modular e funcional.

#### Principais Componentes e suas Funções:

- **Task:** Responsável por listar todas as tarefas, permitindo a reutilização do código para diferentes cenários onde tarefas precisam de ser exibidas.

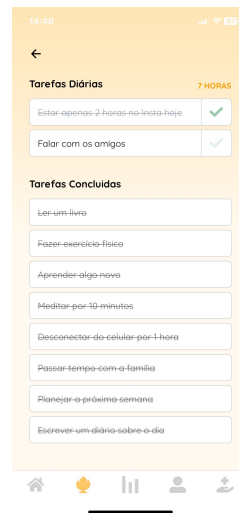


Figura 5.1: Componente *Task*

- **Achievements:** Exibe os troféus, incluindo informações como imagem, nome e descrição, presentes na sala de troféus ou associados ao progresso do utilizador.



Figura 5.2: Componente *Achievements*

- **Progresso dos Troféus:** : Exibe o progresso visual em relação à conquista dos troféus.



Figura 5.3: Componentes de Progresso de Troféus

- **Cards de Seleção:** Utilizados para interações específicas dentro do perfil e das suas subpáginas, permitindo uma navegação mais visual e intuitiva.



Figura 5.4: Componentes *Cards* de Seleção

Os nomes dos componentes foram definidos de forma autoexplicativa, facilitando a compreensão rápida da sua finalidade e promovendo a legibilidade do código.

## Estilização dos Componentes

Para a estilização, foi utilizado o **Tailwind CSS** em conjunto com o **Nativewind**, uma ferramenta adaptada para *React Native*. Esta abordagem foi escolhida devido à sua eficiência na criação de estilos de forma rápida e consistente, sem a necessidade de adicionar extensas linhas de código de estilização repetitiva em cada ficheiro. Com o uso de classes utilitárias, tornou-se mais fácil ajustar os estilos diretamente nos componentes, permitindo um desenvolvimento ágil e uma manutenção simplificada.

## 5.2 *Routing* e Navegação

A navegação entre as diferentes páginas da aplicação foi implementada utilizando a biblioteca *React Navigation*, uma solução robusta e popular para aplicações *React Native*. Esta biblioteca permite definir a estrutura de navegação da aplicação, gerindo as transições entre as diferentes telas e fornecendo uma experiência de utilizador fluida e intuitiva.

Foram utilizadas as seguintes extensões do *React Navigation*:

- **@react-navigation/native:** Este pacote contém as implementações comuns a todos os outros pacotes de navegação e define as estruturas base para o funcionamento da biblioteca. É um pré-requisito para utilizar as outras extensões.
- **@react-navigation/native-stack:** Esta extensão implementa um navegador baseado em *stack* (pilha). A navegação em *stack* funciona como uma pilha de cartas: cada nova tela é "empilhada" sobre a anterior, e o utilizador pode voltar para as telas anteriores utilizando o botão de voltar nativo do dispositivo ou uma ação específica. Este tipo de navegação é ideal para fluxos lineares, como a navegação entre detalhes de um produto ou a sequência de passos num formulário. No código, é utilizado para gerir a navegação entre páginas como *WelcomePage*, *LoginPage*, *RegisterPage* e outras, permitindo o fluxo de autenticação e o acesso à aplicação principal. Também

é usado para navegação dentro de secções específicas, como a secção de Troféus (*TrophiesPage*, *AllTasks*, *AllTrophies*) e a secção de Perfil (*ProfilePage*, *Settings*, *EditProfile*, *AllLumiQuestions*).

- **@react-navigation/bottom-tabs:** Esta extensão cria um navegador com separadores na parte inferior do ecrã (*navbar* inferior). Cada separador corresponde a uma tela diferente, permitindo ao utilizador alternar rapidamente entre as principais secções da aplicação. No código, o componente *HomeTabs* utiliza esta extensão para criar os cinco separadores principais: "Home", "Troféus", "Report", "Perfil" e "Ajuda".

## Principais Rotas Configuradas:

A estrutura de navegação pode ser resumida da seguinte forma:

### 1. Fluxo Inicial (*Stack Navigator* Principal):

- **Welcome:** Página de boas-vindas.
- **Login:** Página de *login*.
- **Register:** Página de registo.
- **FirstQuestionnaire:** Página do primeiro questionário (após o registo).
- **HomeTabs:** Navegador com separadores inferiores (*Tab Navigator*).
- **TrophyDetail:** Tela de detalhe de um troféu específico.

### 2. Navegação com Separadores Inferiores (*Tab Navigator* - *HomeTabs*):

- **Home:** Página principal.
- **Troféus:** Página de troféus (abre o *TrophiesPageStack*).
- **Report:** Página de relatórios.
- **Perfil:** Página de perfil (abre o *ProfilePageStack*).
- **Ajuda:** Página de ajuda.

### 3. *Stack Navigator* para Troféus (*TrophiesPageStack*):

- **TrophiesPage:** Página principal de troféus.
- **AllTasks:** Página com todas as tarefas.
- **AllTrophies:** Página com todos os troféus.

### 4. *Stack Navigator* para Perfil (*ProfilePageStack*):

- **ProfilePage:** Página principal de perfil.
- **Settings:** Página de definições.
- **EditProfile:** Página de edição de perfil.
- **AllLumiQuestions:** Página com todas as perguntas Lumi.

## Conexão entre as Páginas:

As rotas conectam as páginas através dos componentes *Stack.Screen* e *Tab.Screen*. O *Stack.Navigator* gere a navegação em pilha, permitindo a transição entre telas através de ações como *navigation.navigate('NomeDaTela')*. O *Tab.Navigator* permite a troca entre as telas principais através dos separadores inferiores. A combinação de ambos permite uma estrutura de navegação hierárquica e organizada. Por exemplo, a partir da tela Perfil (dentro do *HomeTabs*), o utilizador pode navegar para a tela *Settings* (dentro do *ProfilePageStack*).

## Contribuição para a Experiência do Utilizador:

O uso do *React Navigation* contribui significativamente para uma boa experiência do utilizador, proporcionando:

- **Transições Fluidas:** As animações e transições entre as telas são suaves e consistentes com as convenções das plataformas móveis.
- **Navegação Intuitiva:** A estrutura hierárquica e o uso de separadores inferiores tornam a navegação na aplicação fácil de entender e utilizar.
- **Gestão do Histórico de Navegação:** A navegação em *stack* permite que o utilizador volte facilmente para as telas anteriores utilizando o botão de voltar do dispositivo.

## Desafios na Implementação:

Um desafio específico na implementação do *routing* foi garantir que, ao clicar novamente num separador que já está ativo, o utilizador fosse levado para o início do *stack* correspondente. Isso foi resolvido com o uso do *listeners* e do evento *tabPress* nos componentes *Tab.Screen* do *HomeTabs*. Ao detetar um novo clique num separador ativo, o código impede o comportamento padrão e utiliza *navigation.navigate* para redirecionar o utilizador para a tela inicial do *stack* correspondente, garantindo uma melhor experiência de utilizador. Por exemplo, se o utilizador está na página *EditProfile* e clica novamente no separador "Perfil", será redirecionado para a página *ProfilePage*.

Em resumo, a implementação do *routing* com *React Navigation* e suas extensões permitiu criar uma estrutura de navegação robusta, organizada e intuitiva, contribuindo para uma experiência de utilizador positiva.



## Capítulo 6

# Instruções de Configuração e Execução do Projeto

O repositório do *frontend* do projeto **LumiCheck** está hospedado na plataforma GitHub e pode ser acessado através do seguinte link: [Repositório Frontend Lumicheck](#)

Este repositório contém todos os arquivos necessários para clonar, instalar dependências e executar o projeto localmente.

### 6.1 Como Executar o Projeto

Para executar o projeto localmente, as instruções abaixo devem ser seguidas:

1. Ter o **Node.js** instalado e atualizado no computador;
2. Dar clone do repositório;
3. Instalar as dependências através do comando *npm install*;
4. Por fim iniciar o projeto com o comando *npm start*

### 6.2 Como Visualizar o Projeto

**No Telemóvel:**

- Depois de executar o comando *npm start*, um **QRCode** será gerado no terminal;
- Com a app **Expo Go** (disponível na App Store e Play Store), digitaliza o QRCode.

**Versão Web:**

- Depois de executar o comando *npm start*, pressiona a tecla **w** no terminal para abrir a versão web;
- No painel de ferramentas (f12), selecciona o modo "Dispositivo Móvel" para visualizar a interface como se estivesse num telemóvel.

## Capítulo 7

# Reflexão crítica e Conclusão

### 7.1 Reflexão Crítica

Este capítulo apresenta uma análise crítica sobre o desenvolvimento do projeto, destacando as dificuldades enfrentadas ao longo do processo, as estratégias adotadas para superá-las, e os próximos passos necessários para concluir e melhorar o projeto.

#### Dificuldades Enfrentadas

1. **Integração de Fontes e Estilização:** Configurar a fonte *Quicksand* e o *TailwindCSS* para funcionar corretamente no ambiente do *React Native* foi um desafio técnico inicial, devido à necessidade de integrar ferramentas não nativas para estilização.
2. **Mapas sem a API do Google:** Implementar um mapa funcional sem usar a API do Google, devido ao custo associado, foi um obstáculo significativo. Foi necessário investigar alternativas gratuitas para atender às necessidades do projeto.
3. **Animações na Interface:** Desenvolver animações para as páginas **Home** e **Report** foi desafiador, especialmente para manter a fluidez e a responsividade. Isso exigiu o uso intensivo de bibliotecas como *React Native Reanimated* e otimizações no código.
4. **Animações de Slides e Cards:** Implementar animações para transições entre *slides* e movimentação de *cards*, mantendo a experiência de utilizador fluida e atrativa, foi outra área que exigiu pesquisa e ajustes constantes.

#### Próximos Passos

1. **Implementação do Backend:** O próximo grande passo no desenvolvimento da app LumiCheck é a implementação de um *backend* robusto que possibilite o armazenamento de dados, autenticação de utilizadores e outras funcionalidades dependentes de um servidor.
2. **Integração com APIs Nativas:** Planeia-se abandonar o uso do **Expo** para buscar dados de *screen time* através da API nativa dos sistemas Android, permitindo maior precisão e acesso a funcionalidades avançadas.
3. **Melhorias na Interface:** Continuar a melhorar a experiência do utilizador, otimizando animações e elementos visuais baseados no *feedback* recebido.

### 7.2 Conclusão

O desenvolvimento do *frontend* da app LumiCheck foi uma experiência enriquecedora, permitindo à equipa adquirir conhecimentos técnicos e experiência prática no uso de ferramentas modernas de desenvolvimento. Apesar das dificuldades, o projeto evoluiu significativamente, e a equipa conseguiu implementar as principais funcionalidades planeadas. Os próximos passos incluem ajustes finais e

melhorias, assegurando que o produto final seja funcional, esteticamente agradável e impactante para os utilizadores.

# Apêndice A

## 30 Tarefas

1. Convidar um amigo para irem a um café.
2. Fazer uma caminhada.
3. Fazer alongamentos (até pode haver uma sugestão de um vídeo para seguir).
4. Ler um capítulo de um livro.
5. Desenhar ou pintar algo simples (até pode haver uma sugestão do que tentar desenhar, um objeto simples, ex: um lápis).
6. Ouvir uma música sem distrações e prestar atenção à letra.
7. Jogar um jogo de tabuleiro ou cartas.
8. Planear uma pequena viagem imaginária.
9. Fazer uma lista de filmes ou livros que gostavas de ver ou ler.
10. Fazer um origami.
11. Resolver um sudoku.
12. Experimentar uma receita nova.
13. Limpar ou organizar algo que tens estado a adiar.
14. Fazer o plano do dia seguinte.
15. Fazer uma chamada a um amigo.
16. Fazer uma chamada a um familiar.
17. Cantar uma música que gostes em voz alta.
18. Tirar 5 minutos para meditar.
19. Plantar uma semente.
20. Fazer 10 flexões, abdominais ou agachamentos.
21. Escrever uma carta para o teu eu do futuro daqui a 10 anos.
22. Andar descalço a sentir a natureza, para sentir a relva ou areia.
23. Ouvir um *podcast*.
24. Fazer um álbum digital com as fotos dos teus momentos favoritos.
25. Criar uma *playlist* no *Spotify* com o teu *mood* atual.

26. Ver roupa que tenhas a mais e já não uses e vender ou doar.
27. Fazer uma lista de lugares que gostaria de visitar.
28. Escrever um poema.
29. Ir a um parque e identificar 3 seres vivos.
30. Jogar Pedra/Papel/Tesoura com um amigo.

## Apêndice B

# Possíveis Desafios

### Semana 1 – Introdução ao “*Offline*”

- **Ficar 1 hora *offline* por dia:** Definir um horário fixo (por exemplo, das 20h às 21h) e desligar completamente do telemóvel.
- **Evitar o telemóvel durante as refeições principais:** ao almoço e ao jantar, deixa o telemóvel de lado e aproveita para conversar, caso estejas acompanhado.
- **Descobrir uma atividade de 20 minutos que realmente gostes:** Experimentar caminhar, ler, ouvir música, fazer palavras cruzadas ou algo que te relaxe e motive.

### Semana 2 – Foco no Presente

- **Ficar 2 horas sem redes sociais:** Escolher um período (manhã, tarde ou noite) em que costumavas usar muito o telemóvel e evita abrir redes sociais.
- **Fazer uma caminhada de 15 minutos sem o telemóvel:** Concentra-te no ambiente à tua volta e aproveita o momento.
- **Reservar 10 minutos para a respiração consciente:** Antes de adormecer, dedica-te apenas a observar a tua respiração, sem interrupções digitais.

### Semana 3 – Criatividade e Conexões

- **Desenhar ou escrever algo novo todos os dias:** Pode ser um desenho simples ou um pequeno texto. O importante é deixar a criatividade fluir.
- **Conversar com alguém sem distrações digitais:** Garantir que tens pelo menos uma conversa genuína, sem o telemóvel por perto.
- **Evitar o telemóvel no quarto antes de dormir:** Ficar pelo menos 30 minutos sem o telemóvel antes de te deitares. Deixa-o longe da tua cama.

### Semana 4 – Movimento e Habilidades

- **Aprender algo novo:** Dedicar 20 minutos por dia a um *hobby* ou habilidade que sempre quiseste explorar, como desenhar, cozinhar ou tocar um instrumento.
- **Praticar 30 minutos de atividade física sem o telemóvel:** Escolher uma atividade como yoga, exercício ou caminhada.
- **Ler um capítulo ou 15 páginas de um livro:** Fazer isso diariamente, sem interrupções digitais.

## Semana 5 – Redução do Tempo Online

- **Reduzir o tempo de uso do telemóvel pela metade:** Usa esta *app* para monitorizar o teu tempo e ajusta os teus hábitos.
- **Evitar o telemóvel durante o transporte:** Se possível, utiliza esse tempo para observar o mundo ou pensar.
- **Ver uma série ou um filme sem distrações:** Desligar o telemóvel e foca-te apenas no que estás a assistir.

## Semana 6 – Explorar o Mundo Real

- **Assistir ao nascer ou pôr do sol todos os dias:** Fazer isso sem o telemóvel e absorver a beleza do momento.
- **Escrever bilhetes ou cartas:** Dedicar um momento do teu dia para escrever uma mensagem ou carta simples para alguém especial.
- **Sair com um amigo ou familiar sem o telemóvel:** Passar tempo de qualidade com alguém sem usar o telemóvel durante o encontro.

## Semana 7 – Mente e Corpo

- **Fazer 10 minutos de meditação ou *mindfulness*:** Reservar um momento para relaxares, sem *apps* ou guias digitais.
- **Desativar todas as notificações de redes sociais e jogos:** Mantém-te sem distrações durante toda a semana.
- **Evitar multitarefas com o telemóvel:** Fazer uma coisa de cada vez, como ouvir música, ler ou trabalhar, sem alternar entre *apps* ou redes.

## Semana 8 – Consolidando o Hábito

- **Ficar 2 horas por dia sem o telemóvel em casa:** Deixa-o noutra sala durante esse tempo e concentra-te noutras atividades.
- **Organizar um espaço da tua casa:** Dedicar tempo a arrumar ou reorganizar um espaço todas as semanas (ex.: uma gaveta, prateleira ou armário).
- **Fazer um resumo da tua experiência:** Escrever ou refletir sobre como reduzir o tempo no telemóvel mudou o teu dia a dia.

