

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA  
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

**ÉRICK CÉLIO DE MELO GUIMARÃES**  
**HENRIQUE BERNARDO DA SILVA**  
**MARCUS DANIEL BATISTA E SILVA**

Passaporte Digital do Caminho de Cora Coralina

Anápolis  
Novembro, 2021



UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA  
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

ÉRICK CÉLIO DE MELO GUIMARÃES  
HENRIQUE BERNARDO DA SILVA  
MARCUS DANIEL BATISTA E SILVA

Passaporte Digital do Caminho de Cora Coralina

Trabalho apresentado ao Curso de Engenharia de Computação da Universidade Evangélica de Goiás – UniEVANGÉLICA, da cidade de Anápolis-GO como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Computação.

Orientador (a): Prof. Ms. Natasha Sophie Pereira

Co-Orientador(a): Prof. Dr. Iransé Oliveira Silva

Anápolis  
Novembro, 2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA  
ENGENHARIA DE COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

ÉRICK CÉLIO DE MELO GUIMARÃES  
HENRIQUE BERNARDO DA SILVA  
MARCUS DANIEL BATISTA E SILVA

PASSAPORTE DIGITAL DO CAMINHO DE CORA CORALINA

Monografia apresentada para Trabalho de Conclusão de Curso de Engenharia de Computação da Universidade Evangélica de Goiás - UniEVANGÉLICA, da cidade de Anápolis-GO como requisito parcial para obtenção do grau de Engenheiro(a) de Computação.

**Aprovado por:**

---

**Nome completo do orientador, [Especialista | Mestre | Doutor], Sigla da universidade que atua  
(ORIENTADOR)**

---

**Nome completo do examinador, [Especialista | Mestre | Doutor], Sigla da universidade que atua  
(AVALIADOR)**

**Cidade, dia de mês de ano.**

## **FICHA CATALOGRÁFICA**

GUIMARÃES Érick, SILVA Henrique, SILVA Daniel. **Passaporte Digital do Caminho de Cora Coralina**. Anápolis 2021. (Universidade Evangélica de Goiás – UniEVANGÉLICA).

Monografia. Universidade Evangélica de Goiás, Curso de Engenharia de Computação, da cidade de Anápolis-GO.

1. Caminho de Cora Coralina, Dispositivos móveis, Trilha, Aplicativo para trilha, Aplicativo para o Caminho de Cora Coralina.

## **CESSÃO DE DIREITOS**

NOMES DOS AUTORES: Érick Célio de Melo Guimarães, Henrique Bernardo da Silva, Marcus Daniel Batista e Silva

TÍTULO DO TRABALHO: Passaporte digital do Caminho de Cora Coralina

GRAU/ANO: Graduação /2021

É concedida à Universidade Evangélica de Goiás - UniEVANGÉLICA, permissão para reproduzir cópias deste trabalho, emprestar ou vender tais cópias para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

---

Érick Célio de Melo Guimarães  
Anápolis, 19 de novembro de 2021

---

Henrique Bernardo da Silva  
Anápolis, 19 de novembro de 2021

---

Marcus Daniel Batista e Silva  
Anápolis, 19 de novembro de 2021



## RESUMO

O Caminho de Cora Coralina é uma trilha de longo curso para caminhantes e ciclistas que liga várias cidades turísticas goianas, ela conta com o passaporte do peregrino, que é um registro físico para identificar e registrar os checkpoints de quem percorre o caminho, fornecendo um certificado ao concluir todo o percurso. Este trabalho apresenta o desenvolvimento de um sistema com a principal funcionalidade de automatizar o passaporte, informando também sobre o percurso da trilha, pontos de interesse e estabelecimentos, o sistema também possui rankings para incentivar desafios entre os trilheiros. Os resultados do percurso da trilha são captados através de um celular com sua localização ligada, que são enviados para o servidor onde os dados são tratados e salvos no banco de dados. O desenvolvimento de um painel administrativo onde serão cadastrados os caminhos, estabelecimentos, locais de interesse e checkpoints, além de toda a administração do ranking, usuários e dos certificados.

**Palavras-chave:** Caminho de Cora Coralina. Automatização. Sistema.

## LISTA DE FIGURAS

Figura 1 - Roteiro Caminho de Cora Coralina.	14
Figura 2 - Passaporte do Peregrino.	15
Figura 3 - Passaporte do Peregrino.	15
Figura 4 - Comparação entre PWA x Nativo x Responsivo.	20
Figura 5 - Prototipação da aplicação no Figma.	23
Figura 6 - Mutation de Cadastro de usuário.	25
Figura 7 - Página de controle de permissões	26
Figura 8 - Página de edição da rota	26
Figura 9 - Certificado gerado.	27
Figura 10 - Código da página de perfil de usuário.	28
Figura 11 - Página inicial do app.	29
Figura 12 - Tela de login.	29
Figura 13 - Tela de conquistas.	30



## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>11</b>
<b>Problema</b>	<b>11</b>
<b>Objetivos</b>	<b>11</b>
<b>Objetivo Geral</b>	<b>11</b>
<b>Objetivos Específicos</b>	<b>11</b>
<b>Justificativa</b>	<b>12</b>
<b>1. FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
<b>1.1 Trilhas de Longo Curso</b>	<b>13</b>
<b>1.1.1 Trilhas de Longo Curso no Brasil</b>	<b>13</b>
<b>1.1.2. Trilhas de Longo Curso em Goiás</b>	<b>13</b>
<b>1.1.3. Caminho de Cora Coralina</b>	<b>14</b>
<b>1.2. Gamificação</b>	<b>16</b>
<b>1.3. Dispositivos Móveis</b>	<b>16</b>
<b>1.4. Plataforma de dispositivos móveis</b>	<b>16</b>
<b>1.4.1. Android</b>	<b>17</b>
<b>1.4.2. IOS</b>	<b>17</b>
<b>1.5. Desenvolvimento para dispositivos móveis</b>	<b>17</b>
<b>1.5.1. Nativo</b>	<b>18</b>
<b>1.5.2. Híbrido</b>	<b>18</b>
<b>1.5.3. PWA</b>	<b>19</b>
<b>2. METODOLOGIA DA PESQUISA</b>	<b>21</b>
<b>3. DESENVOLVIMENTO</b>	<b>22</b>
<b>3.1. Modelo cascata</b>	<b>22</b>
<b>3.1.1. Requerimento</b>	<b>22</b>
<b>3.1.2. Projeto</b>	<b>23</b>
<b>3.1.3. Implementação</b>	<b>24</b>
<b>3.1.4. Verificação</b>	<b>30</b>
<b>3.1.5. Manutenção</b>	<b>30</b>
<b>4. RESULTADOS</b>	<b>31</b>
<b>5. CONCLUSÃO E CONSIDERAÇÕES FINAIS</b>	<b>32</b>
<b>REFERÊNCIAS</b>	<b>33</b>



## INTRODUÇÃO

### Problema

O Caminho de Cora Coralina é uma trilha de longo curso com aproximadamente 300 km de extensão, este projeto foi idealizado em 2013 e retomado em 2017 pela Goiás Turismo - Agência estadual de turismo, sendo uma das principais rotas turísticas para caminhantes e ciclistas em Goiás (CAMINHO DE CORA CORALINA, 2019).

Existe um certificado para quem conclui o percurso a pé ou de bicicleta (Portal Goiás 2021). Segundo o Portal Goiás:

Associação do Caminho de Cora Coralina lança “Certificado do Peregrino”, para caminhante que completar a trilha. Durante a solenidade, o presidente da Goiás Turismo, Fabrício Amaral, foi homenageado com diploma de reconhecimento como “Amigo do Caminho de Cora Coralina”, por sua dedicação na consolidação do atrativo turístico.

Para se obter o certificado de conclusão do percurso é necessário enviar uma foto do passaporte com todos os checkpoints carimbados, que é feito de forma manual por agentes encontrados durante o percurso, segundo o próprio site do projeto. Após estas etapas serem concluídas, a organização do projeto deverá receber essas informações, que serão validadas para que o certificado digital seja gerado para o peregrino (Caminho de Cora Coralina, 2019). Existe alguma forma para melhorar a gestão dos certificados digitais e a experiência vivida na trajetória? E como incentivar a conclusão do Caminho de Cora Coralina?

### Objetivos

#### Objetivo Geral

Automatizar o processo de checkpoints e emissão de passaporte digital para os peregrinos do Caminho de Cora Coralina através de um Aplicativo para dispositivos móveis.

#### Objetivos Específicos

- Desenvolvimento da proposta do projeto;
- Levantamento da pesquisa bibliográfica;
- Desenvolvimento da fundamentação teórica;
- Levantamento de requisitos;
- Prototipação do aplicativo;
- Validação dos requisitos levantados e protótipos.

## **Justificativa**

O turismo tem se tornado uma atividade de geração de empregos e renda, sendo bastante valorizado nas estratégias de desenvolvimento, tornou-se modalidade muito empregada em regiões carentes (ALMEIDA, 2020). Outro ponto que Almeida (2020, p. 2) destaca é:

O setor [de turismo] passou a ser fortemente apoiado pelos Objetivos de Desenvolvimento Sustentável - ODS, ao ser amparado no oitavo objetivo de “promover o crescimento econômico sustentado, inclusivo e sustentável, emprego pleno e produtivo, e trabalho decente para todos.”

O caminho de Cora surge neste contexto, pois foi construído no Brasil central, possuindo paisagens exuberantes, delimitadas em vilarejos e parques, que somados resultam em lindas paisagens históricas (ALMEIDA, 2020).

Atualmente este percurso conta com uma certificação que é obtida ao concluir checkpoints em marcos específicos da rota, para o controle dos checkpoints alcançados pelos peregrinos, os mesmos adquirem um passaporte que é marcado através de carimbos nos devidos marcos, e todo esse processo é realizado manualmente sendo necessário ter um aviso prévio para confirmar a disponibilidade de agentes que irão averiguar a conclusão do checkpoint para obtenção destes certificados (CAMINHO DE CORA CORALINA, 2019).

Devido ao processo manual tanto dos checkpoints, quanto da solicitação do certificado, acaba-se trazendo uma má experiência para o peregrino, assim faz-se necessária a automatização do processo, a fim de facilitar o acesso dos peregrinos, tornando a experiência do caminho mais prazerosa e independente.

## **1. FUNDAMENTAÇÃO TEÓRICA**

### **1.1 Trilhas de Longo Curso**

Em outubro de 1921, o engenheiro florestal norte-americano Benton MacKaye apresentou ao mundo uma das ideias mais originais da história dos esportes de montanha: A Trilha de Longo Curso (MENEZES, 2017a). Sua ideia deu origem à Appalachian Trail, inaugurada em 1932 e, desde então, serviu de inspiração e subsídio para o estabelecimento de mais de 1.000 trilhas de longo curso mundo afora (MENEZES, 2017b).

#### **1.1.1 Trilhas de Longo Curso no Brasil**

No Brasil o projeto teve início em 1999, com a sinalização de 60 km na Floresta da Tijuca, cujo traçado já foi pensado para poder ser expandido para todo o Município do Rio de Janeiro, no que viria a ser a Trilha Transcarioca (MENEZES, 2017a). Em 2018 no Brasil foi instituída a Rede Nacional de Trilhas de Longo Curso e Conectividade - Rede Trilhas, inspirada nas experiências internacionais, a Rede Brasileira de Trilhas de Longo Curso congrega grandes trilhas nacionais compostas por trilhas regionais menores e promove o planejamento integrado e a sinalização padronizada das trilhas em todo território nacional, promovendo recreação, gerando emprego, renda e conectando paisagens (MEYER, 2020). As Trilhas de Longo Curso brasileiras já contabilizam 8.110 km de percurso, por meio de 58 trilhas atualmente cadastradas na Rede (MEYER, 2020), tendo o potencial para contribuir de forma significativa para que possamos transformar nossos atrativos naturais e culturais em produtos turísticos, o que é, de fato, um dos maiores desafios do fortalecimento do turismo no Brasil (MEYER, 2020).

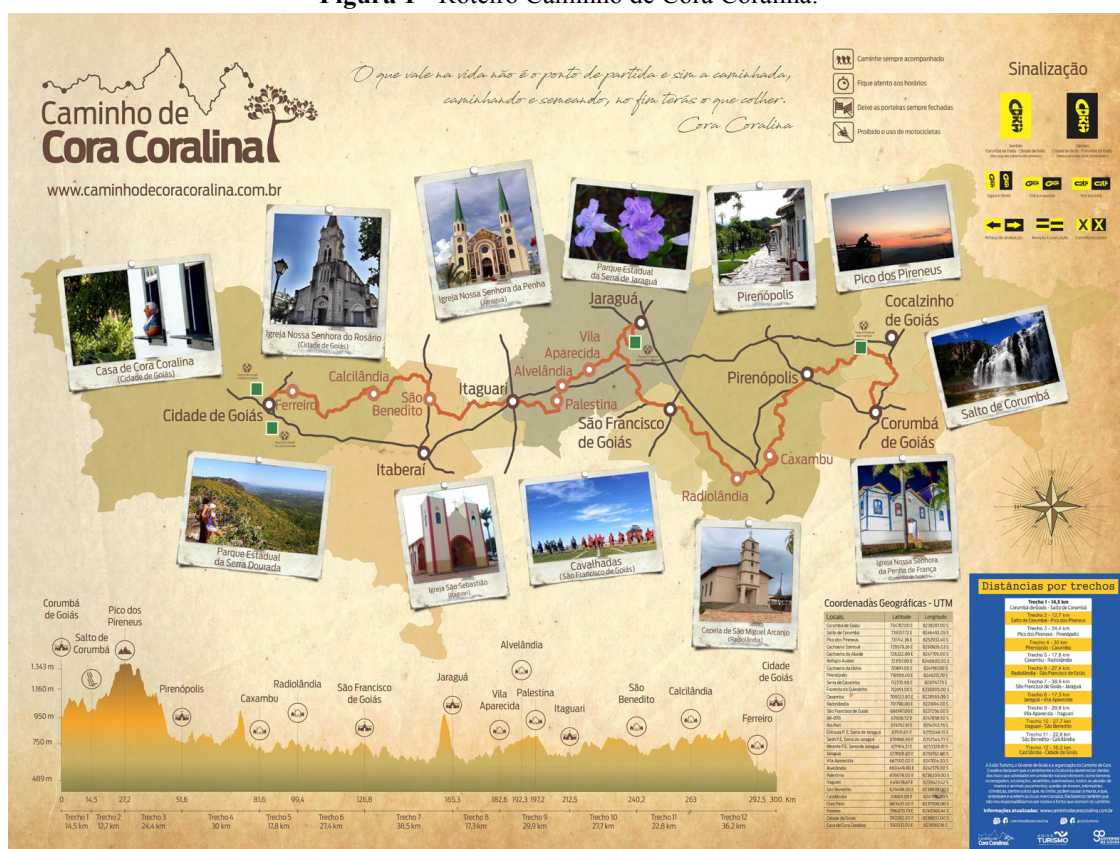
#### **1.1.2. Trilhas de Longo Curso em Goiás**

Dentre as dez trilhas com maior percurso temos duas em Goiás, sendo o Caminho dos Veadeiros e o Caminho de Cora Coralina as duas contando com mais de 300 quilômetros de extensão cada, sendo possível fazer a travessia por todo percurso caminhando ou de bicicleta.

### 1.1.3. Caminho de Cora Coralina

O Caminho de Cora Coralina cruza as cidades históricas de Corumbá de Goiás, Pirenópolis, São Francisco de Goiás, Jaraguá e a cidade de Goiás, abrangendo também os municípios de Cocalzinho de Goiás, Itaguari e Itaberaí. Idealizado em 2013, o projeto teve como propósito interligar os municípios, povoados, fazendas e atrativos, passando por antigos caminhos, numa rota turística para caminhantes e ciclistas (Caminho de Cora Coralina, 2020).

Figura 1 - Roteiro Caminho de Cora Coralina.



Fonte: Caminho de Cora Coralina, c2019.

Em 2017 foi retomada a implementação do Caminho de Cora Coralina pela Goiás Turismo – Agência Estadual de Turismo que ofereceu apoio em sua estruturação através do Programa Experiências na Natureza, viabilizando a inclusão dos Parques Estaduais e outras Unidades de Conservação no roteiro, mobilizando as comunidades locais e dando os primeiros passos para a organização da Associação Caminho de Cora Coralina (Caminho de Cora Coralina, 2020).

Atualmente o percurso conta com um passaporte físico, onde o viajante poderá registrar os seus checkpoints em marcos predefinidos onde será carimbado, e ao final de todo percurso irá obter um certificado digital após submeter uma foto do passaporte, e o mesmo ser avaliado. Esses checkpoints serão encontrados nas cidades, povoados e no Museu Casa de Cora Coralina, para se obter os carimbos é necessário entrar em contato com os responsáveis do local, para informar data e previsão de horário que irá passar (Caminho de Cora Coralina, 2019).

**Figura 2** - Passaporte do Peregrino.



**Fonte:** Caminho de Cora Coralina, c2019.

**Figura 3** - Passaporte do Peregrino.



**Fonte:** Caminho de Cora Coralina, c2019.

## **1.2. Gamificação**

Gamificação vem do inglês gamification, e é o uso de mecânicas e dinâmicas de jogos para engajar pessoas, resolver problemas e melhorar o aprendizado, motivando ações e comportamentos em ambientes fora do contexto de jogos. Na prática, isso significa oferecer recompensas aos participantes que realizam tarefas pré-estabelecidas, que, geralmente, são voltadas para a recomendação, a divulgação ou a avaliação dos produtos ou serviços da empresa (CARVALHO, 2020).

Com a gamificação é possível atingir alguns resultados, como a estimulação de competição saudável, sentimento de conquista, medição de desempenho, maior acessibilidade e o principal trazer uma boa experiência para seus participantes (CARVALHO, 2020).

## **1.3. Dispositivos Móveis**

O uso de dispositivos móveis tanto no Brasil quanto no mundo só vem crescendo, e hoje é o principal meio de acesso à rede no país, em 2018 o percentual de pessoas de 10 anos ou mais que acessaram a internet pelo celular chegou à 98,1% segundo as informações da Pesquisa Nacional por Amostra de Domicílios Contínua - Tecnologia da Informação e Comunicação (PNAD Contínua TIC) (TOKARNIA, 2020).

Os dispositivos móveis estão sendo utilizados em massa, tanto no ambiente de trabalho quanto lazer, devido a diversidade de aplicativos é possível achar facilidades tanto para o dia a dia quanto para o lazer. E cada vez mais se vê necessário ter um celular para poder se comunicar e interagir dentro de comunidades ou até mesmo com familiares, e diretamente atrelado ao custo de entrada ser realmente mais acessível do que um computador ou notebook podemos notar um grande crescimento de seu uso.

## **1.4. Plataforma de dispositivos móveis**

Pensando nas plataformas de dispositivos móveis que temos no mercado, as mais populares são Android e iOS, cada uma contando com sua SDK e linguagem de programação própria e bem definida. Encontramos o iOS exclusivamente em dispositivos da Apple principalmente no iPhone, já o Android encontramos pelas mais diversas marcas como Samsung, Motorola, LG, Xiaomi e etc. Sendo o Android o mais vendido no Brasil pelo seu



custo ser mais acessível, e seu sistema suportar diversas modificações, já o iOS é bem restritivo às modificações e possui seu preço mais elevado.

#### **1.4.1. Android**

O Android surgiu em 2003, criado pela Android Inc, com o propósito de ser um sistema gratuito para todas as pessoas que quisessem ter acesso a ele e também ser simples aos desenvolvedores, além de claro ser Open Source e baseado no Kernel Linux, já em 2005 o Google adquiriu o Android Inc ganhando ainda mais adoção (MEYER, M. 2020).

Para o desenvolvimento Android é necessário pelo menos um de dois kits de desenvolvimento, primeiramente o NDK (Kit de Desenvolvimento Nativo) esse pode ser manipulado utilizando C/C++ ou Java, e só é indicado em casos específicos como os casos onde é necessário conseguir mais desempenho de um dispositivo para atingir baixa latência ou apps com muita computação, como jogos ou simuladores, e o segundo kit é o Android SDK (Kit de Desenvolvimento de Software ou DevKit), que se trata de uma plataforma de mais alto nível baseado em Java, onde já é incluso ferramentas de desenvolvimento, emulador e outras bibliotecas para auxiliar o desenvolvimento (SILVA, 2020).

#### **1.4.2. IOS**

O iOS surgiu em 2007 lançado junto com o primeiro iPhone em 29 de junho, inicialmente estava disponível exclusivamente para o iPhone, mas hoje também é usado no iPad e iPod Touch, mas ainda é exclusivo para aparelhos da Apple (MARQUES, 2019).

Para o desenvolvimento iOS é utilizado principalmente o Swift e Objective-C, mas temos algumas peculiaridades, por exemplo a compilação para iOS só pode ser feita através do Xcode que é um ambiente de desenvolvimento gratuito da Apple, mas só pode ser instalado em um sistema macOS.

### **1.5. Desenvolvimento para dispositivos móveis**

Ao desenvolver para dispositivos móveis é necessário escolher um formato de desenvolvimento do aplicativo que mais faça sentido, tanto para o propósito do aplicativo quanto para o conforto e experiência da equipe de desenvolvimento, tendo os principais formatos nativo, híbrido e pwa.

### **1.5.1. Nativo**

O desenvolvimento Nativo como o próprio nome já diz, é quando a aplicação é codificada, desenhada e planejada para a plataforma desejada, no caso iOS ou Android, podendo explorar as funcionalidades e proporcionando uma experiência mais alinhada com a plataforma devido à possibilidade de aproveitar todos os recursos oferecidos. Mas devido à complexidade e diferença entre as plataformas caso o projeto necessite ser disponibilizado em mais de uma plataforma é necessário ter dois projetos distintos, e é comum ter equipes distintas para cada projeto, podendo aumentar consideravelmente o custo e tempo de desenvolvimento se compararmos com o desenvolvimento híbrido (Sthima Blog, 2020).

### **1.5.2. Híbrido**

O desenvolvimento híbrido pode ser uma solução quando não se deseja gastar muito tempo e/ou quando não se tem a necessidade de utilizar recursos exclusivos de uma plataforma em específico, pois podemos utilizar um projeto para atender todas as plataformas só sendo necessário a compilação para a desejada, e para tal desenvolvimento temos algumas tecnologias no mercado.

No mercado temos o framework Ionic que roda uma camada semelhante a de um navegador, assim todo código pode ser desenvolvido usando Javascript, que é uma linguagem amplamente usada na web, mas que deixa a desejar no quesito de performance e experiência do usuário em comparação à experiência da plataforma, temos também framework é o React Native que também utiliza o Javascript, mas que usa uma estratégia bem diferente do Ionic, pois ele converte todo código para a linguagem nativa, e cria só uma camada extra para conseguir comunicar com os recursos específicos de cada plataforma, assim tendo um custo de performance um pouco maior do que um aplicativo nativo devido esta camada extra, mas com uma experiência para o usuário bem melhor se comparado ao Ionic devido a possibilidade de seguir o padrão de cada plataforma, e com o tempo bem menor se compararmos com o desenvolvimento nativo (Sthima Blog, 2020).

### 1.5.3. PWA

O desenvolvimento de um PWA (Progressive Web App) é um aplicativo utilizando as tecnologias da web (HTML, CSS , JavaScript, WebAssembly), ou seja é praticamente um site que pode aparecer para o usuário como aplicativos tradicionais e com algumas vantagens que deixam a experiência muito próxima da oferecida por apps nativos, podemos ter acesso às funcionalidades de Push Notification, Ícone na tela home do smartphone, Splash screen, Processos rodando em background, Suporte a funcionamento em modo Offline, Acesso à câmera e galeria, Acesso à geolocalização e Acesso aos contatos. O conceito de pwa foi introduzido em 2015 por Berriman e Alex Russell, em 2019 já estava disponível para a maioria dos browsers e continua sendo atualizado, e podemos observar alguns casos populares de aplicativos que utilizam PWA três deles são o Twitter, Uber, e Pinterest (NEILPATEL, 2021).

As grandes vantagens de um app utilizando PWA são, praticidade e agilidade pois não precisa ser baixado tornando-se relativamente mais leve para o usuário e acaba não ocupando espaço nem tempo para ser baixado, acessibilidade e responsividade pois pode ser acessado por qualquer dispositivo que tenha um browser instalado e quaisquer tipos de conexões, também é possível converter uma aplicação web para PWA sem quase nenhum esforço, e o desenvolvimento se torna relativamente mais rápido baseado na experiência do time pois não é preciso se preocupar com diferentes plataformas, e podendo utilizar qualquer tecnologia que é usada no desenvolvimento web, como o principais frameworks usados no mercado como React, Vue, Svelte, etc. As principais desvantagens são a incompatibilidade de alguns browsers como o Firefox, a ausência em algumas lojas de aplicativos, e a interface não ficar tão fluida e performática como um app nativo para aplicações que necessitam de alta performance (NEILPATEL, 2021).

**Figura 4** - Comparação entre PWA x Nativo x Responsivo.

FEATURES	Progressive Web App	Native App	Responsive Web
Multi Platform Capability	✓	✗	✓
Low Cost to build	✓	✗	✗
Installation Required	✗	✓	✗
Updates Required	✗	✓	✗
Push Notifications	✓	✓	✗
Easy Sharing	✓	✗	✓
Low Data Consumption	✓	✗	✓
Offline Usability	✓	✗	✗
Faster UI	✓	✗	✗

Fonte: WARREN, 2020.

## 2. METODOLOGIA DA PESQUISA

Para alcançar o objetivo deste trabalho, seguimos a metodologia científica clássica proposta por Dias-Neto et al. onde é composta por três principais estágios. **Concepção**, onde efetuamos toda a pesquisa bibliográfica sobre o caminho de Cora Coralina e uma pesquisa ad-hoc sobre as tecnologias que poderíamos usar. **Construção**, onde usamos o modelo de desenvolvimento de software em cascata (PRESSMAN). O modelo possui as seguintes etapas, (i) requerimento - levantamento de requisitos e prototipação do sistema, (ii) projeto - definição das tecnologias e da arquitetura do sistema, (iii) implementação - etapa em que o sistema é desenvolvido utilizando as tecnologias definidas, (iv) verificação - após finalizarmos a implementação, serão realizados testes que verificam se o sistema cumpre todos seus objetivos, e (v) manutenção - atualização e evolução do sistema. **Validação**, após finalizarmos a etapa de construção, serão realizados testes de aceitação com os usuários finais do sistema, em que os mesmos usarão o aplicativo ao percorrer o Caminho de Cora, e então verificar se o sistema atende à necessidade encontrada na análise de requisitos.

### **3. DESENVOLVIMENTO**

Esse capítulo apresentará os passos realizados para o desenvolvimento do software utilizando o modelo cascata. Será apresentado o processo de desenvolvimento que inclui o processo de controle de versão do código, as arquiteturas desenvolvidas e os padrões de desenvolvimento.

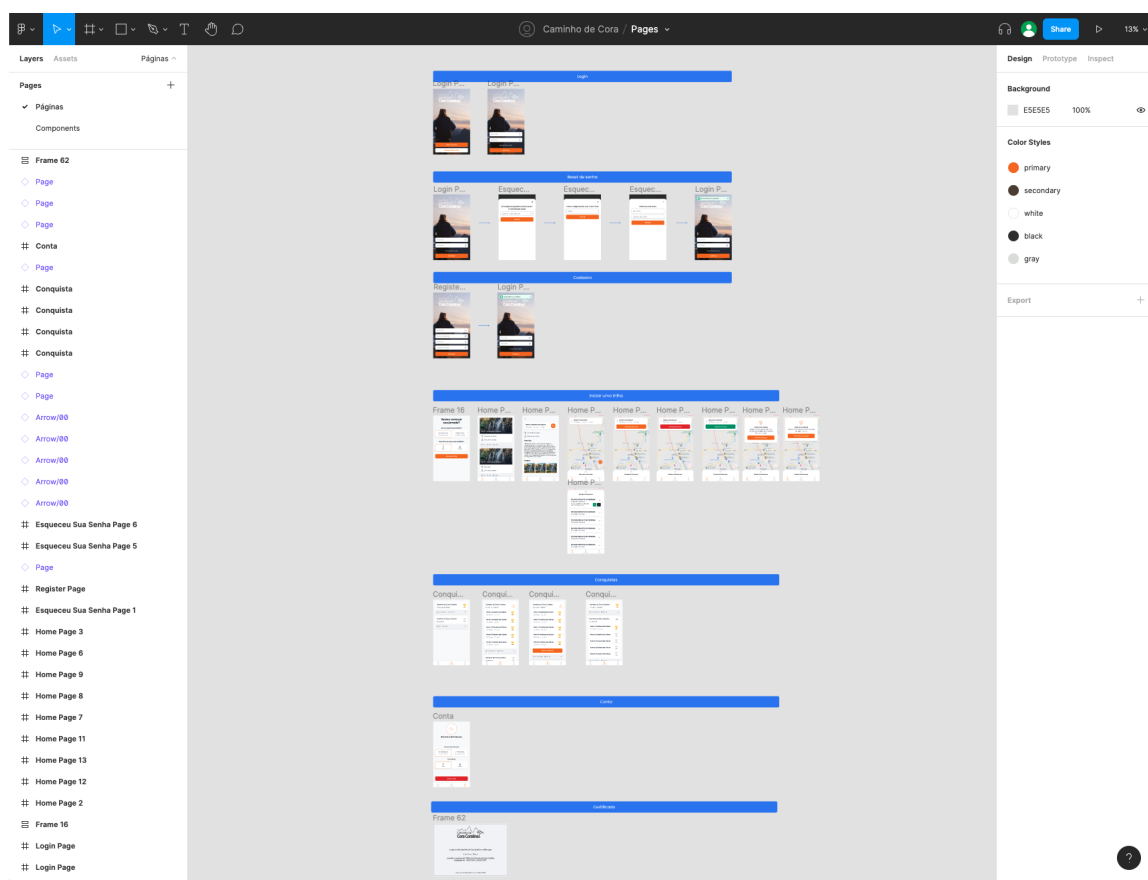
#### **3.1. Modelo cascata**

A aplicação do modelo cascata se deu pelo escopo e objetivos bem definidos no início do projeto, então como não teríamos nenhuma mudança brusca percebemos que para otimizar a velocidade do desenvolvimento poderíamos utilizar esta metodologia, que se baseia nos seguintes passos:

##### **3.1.1. Requerimento**

Nesta etapa fizemos o levantamento de requisitos e para documentação utilizamos o template DMS (Documento de Modelagem de Sistema), que segue as recomendações do livro UML na Prática Do Problema ao Sistema e do modelo PRISM do MPDS (Modelo Prático para Desenvolvimento de Sistemas Orientados a Objeto baseado na UML), onde mapeamos as regras de negócio, tabela de especificação do requisito com sua descrição, risco e prioridade, diagrama geral de caso de uso e o detalhamento de cada caso de uso com suas condições e o seu respectivo diagrama de atividade, em apêndice segue a documentação completa. Para o desenvolvimento dos protótipos, foi utilizada uma ferramenta online específica para prototipagem, chamado Figma, que por meio dela possibilitou a definição de diversos elementos gráficos do aplicativo, sendo eles: o padrão de cores, espaçamentos, ícones, layout dos componentes e o fluxo da aplicação com suas respectivas telas.

**Figura 5** - Prototipação da aplicação no Figma.



Fonte: AUTORES, 2021.

### 3.1.2. Projeto

Nesta etapa fizemos algumas pesquisas ad-hoc para definirmos as tecnologias a serem utilizadas no desenvolvimento que atendessem os requisitos e os objetivos de um bom software seguindo a ISO 25010, chegamos a conclusão de criar dois sistemas distintos, um seria o painel de administrador onde será possível fazer toda a gerência do caminho, e um sistema para o usuário onde terá toda experiência de usuário pensada para sua usabilidade ao percorrer o Caminho de Cora, ambos serão sistemas web se comunicando através de uma *API GraphQL* utilizando as seguintes tecnologias, como principal framework para criação de interface o *React*, no backend usando *NodeJS* e o banco de dados *PostgreSQL*.

### 3.1.3. Implementação

Antes do início do desenvolvimento foi decidido pela equipe para que todo artefato do sistema fosse inserido sobre o controle de versão utilizando a ferramenta Git e a plataforma GitHub. A escolha do mesmo se deu por ser um software gratuito e de fácil utilização e de grande abrangência no mercado de trabalho. Foi criado dois repositórios, um para comportar a aplicação front-end e outro o back-end, foi decidido por utilizar duas *branches* padrões para serem as principais, a primeira denominada *Master* na qual conterà a última versão estável do software, a segunda denominada *Develop* que será a responsável pela parte do código em que está em desenvolvimento e testes, e ambas se perpetuam até a finalização do projeto.

Iniciamos com o desenvolvimento do sistema do painel administrativo onde temos toda a gerência de usuários, perfil de acesso, certificados, checkpoints, rotas, estabelecimentos e pontos de interesse. No desenvolvimento foi utilizado o *Strapi* para melhor controle do conteúdo e segurança da aplicação, também foi utilizado *GraphQL* como meio para consumir os dados. A figura 2 retrata a utilização do *GraphQL* para cadastro de um usuário.



**Figura 6** - Mutation de Cadastro de usuário.

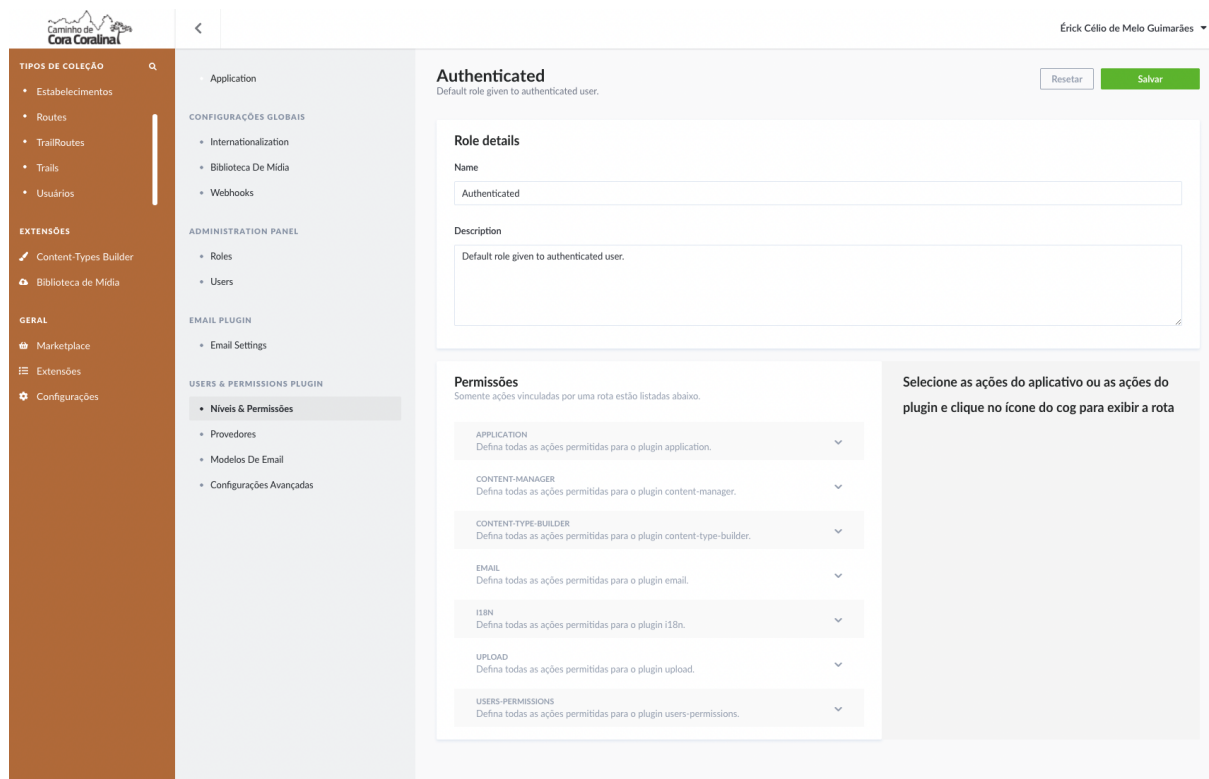
```
export const CREATE_USER = gql`
  mutation createUser(
    $name: String!
    $email: String!
    $password: String!
    $birthdate: Date!
    $sex: ENUM_USERSPERMISSIONSUSER_SEX!
  ) {
    createUser(
      input: {
        data: {
          name: $name
          username: $email
          email: $email
          password: $password
          birthdate: $birthdate
          sex: $sex
        }
      }
    ) {
      user {
        id
      }
    }
  }
`;
```

**Fonte:** AUTORES, 2021.

No *GraphQL* está mapeada toda a entidade de usuário a qual será armazenada no banco de dados pelo *Strapi*. Uma das vantagens do uso do *GraphQL* é a sua organização dos dados e tornando as queries ainda mais manutenível e provendo uma forma de consumo onde só pegamos as informações necessárias melhorando a performance de requisições.

Devido a utilização do *Strapi*, toda a camada de controle de usuários e permissões são feitas através do mesmo.

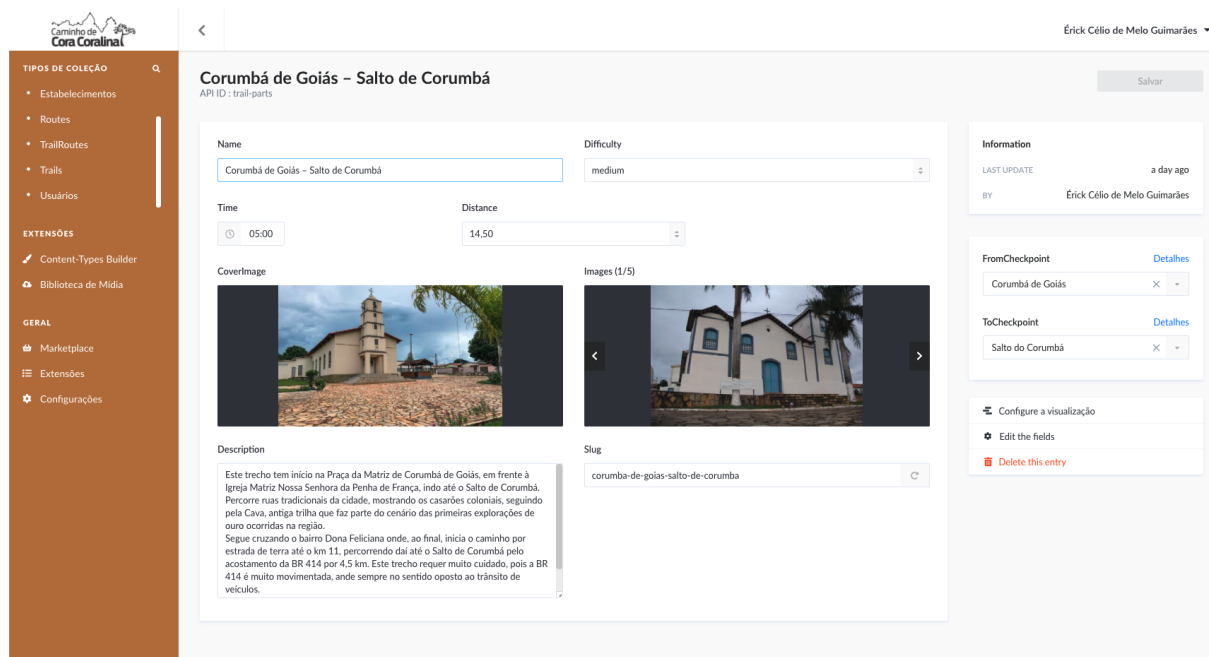
Figura 7 - Página de controle de permissões



The screenshot shows the 'Authenticated' role configuration page. The left sidebar contains navigation menus for 'TIPOS DE COLEÇÃO', 'EXTENSÕES', and 'GERAL'. The main content area is titled 'Authenticated' and includes a 'Role details' section with fields for 'Name' (Authenticated) and 'Description' (Default role given to authenticated user.). Below this is a 'Permissões' section listing various permissions such as APPLICATION, CONTENT-MANAGER, CONTENT-TYPE-BUILDER, EMAIL, ISBN, UPLOAD, and USERS-PERMISSIONS. A right-hand panel prompts the user to 'Selecione as ações do aplicativo ou as ações do plugin e clique no ícone do cog para exibir a rota'. At the top right, there are 'Resetar' and 'Salvar' buttons.

Fonte: AUTORES, 2021.

Figura 8 - Página de edição da rota

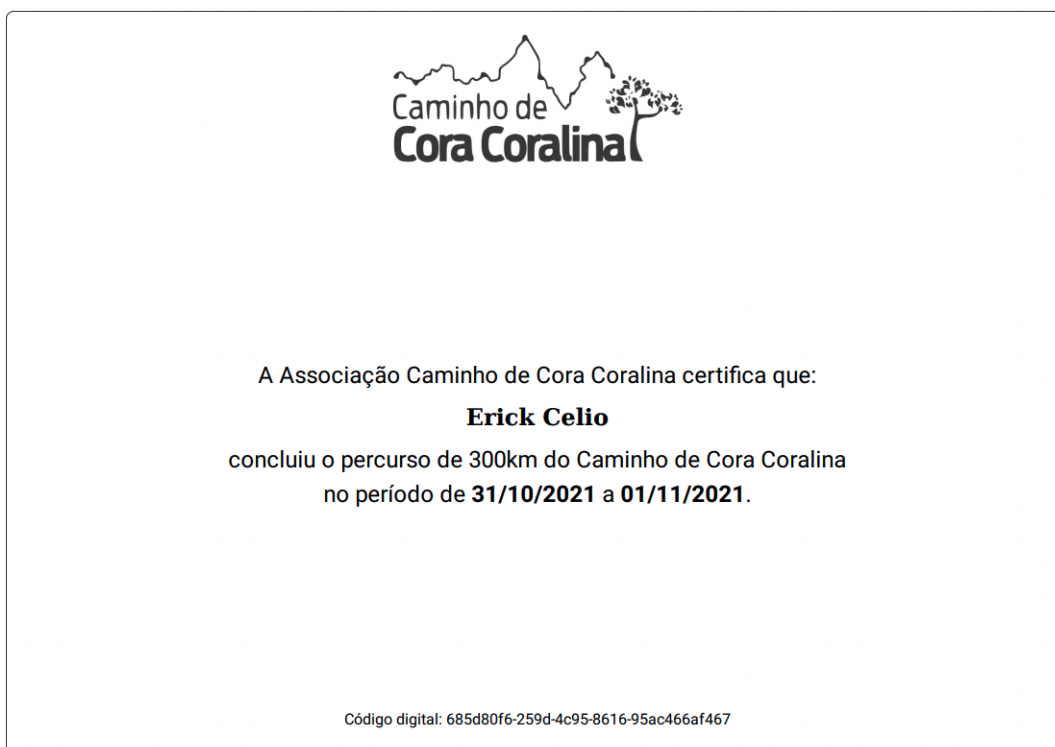


The screenshot shows the 'Corumbá de Goiás - Salto de Corumbá' route editing page. The left sidebar is identical to the previous screenshot. The main content area is titled 'Corumbá de Goiás - Salto de Corumbá' and includes a 'Name' field with the route name, a 'Difficulty' dropdown set to 'medium', and 'Time' and 'Distance' fields set to '05:00' and '14,50' respectively. There are two 'CoverImage' fields, each with a photo of a church. A 'Description' field contains text about the route starting at the Matriz de Corumbá de Goiás and passing through the Cava. A 'Slug' field is set to 'corumba-de-goias-salto-de-corumba'. On the right, there is an 'Information' sidebar showing 'LAST UPDATE' as 'a day ago' and 'BY' as 'Érick Célio de Melo Guimarães'. Below this are 'FromCheckpoint' and 'ToCheckpoint' sections, each with a dropdown menu. At the bottom right, there are options to 'Configure a visualização', 'Edit the fields', and 'Delete this entry'. A 'Salvar' button is located at the top right.

Fonte: AUTORES, 2021.

O *Strapi* é um CMS de código aberto baseado no Node.js para todas as necessidades de API e gerenciamento de conteúdo. Uma api funcional que pode se conectar ao desenvolvimento de software sem o incômodo de complexidades do CMS, sendo bem extensível, com possibilidade de desenvolver fluxos alternativos, estendendo ou não as suas próprias funcionalidades e tudo usando o JavaScript como principal linguagem tanto para a parte de back-end quanto do front-end, o painel administrativo é feita usando React e totalmente customizável. Todos os dados são disponibilizados através de uma API personalizável, podendo ser consumida através do padrão *REST* ou *GraphQL* para atender as necessidades do projeto, como por exemplo na geração do certificado do peregrino tivemos que desenvolver um endpoint totalmente customizado, utilizando a biblioteca *puppeteer* que possibilitou a geração do pdf baseado em uma página da própria web, onde podemos customizar e criar um certificado para cada usuário.

**Figura 9** - Certificado gerado.



**Fonte:** AUTORES, 2021.

Após finalizarmos o desenvolvimento do painel admin começamos a desenvolver o front-end do app usado pelo peregrino, usamos como principal tecnologia o React com typescript, com algumas bibliotecas amplamente usadas no mercado como o Redux, TailwindCSS e Formik que nos ajudam com a organização do projeto, produtividade e boas práticas já adotadas.

Para o desenvolvimento das funcionalidades inicialmente é criado o serviço na qual a funcionalidade irá buscar e consumir os dados do back-end. Realizada a criação do serviço então é produzida a página da funcionalidade, página essa a qual o usuário irá interagir através de formulários, botões entre outros elementos gráficos. Cada página é composta por diversos componentes dentro do React. O componente por sua vez inclui templates e o estilo do componente que são responsáveis pelos elementos HTML e CSS que serão mostrados no navegador do usuário.

Figura 10 - Código da página de perfil de usuário.

```
const Perfil = () => {
  // You, a month ago • feat: some adjustments at login user
  const user = useSelector(userSelector);
  const queryClient = useQueryClient();
  const userMutation = useUpdateUserMutation({
    onSuccess: () => {
      queryClient.invalidateQueries('getUser');
    },
  });
  const trailMutation = useUpdateTrailMutation({
    onSuccess: () => {
      queryClient.invalidateQueries('getUser');
    },
  });

  const [citiesDistance, setCitiesDistance] = useState<number[]>([]);
  const userCurrentPosition = useSelector(userPositionSelector);
  const dispatch = useDispatch();
  const trail = useSelector(trailSelector);
  const [activeCity, setActiveCity] = useState(trail?.inversePaths ? 1 : 0);
  const [activeModality, setActiveModality] = useState<'bike' | 'foot'>(
    trail?.modality || 'bike'
  );

  const getCitiesDistances = useCallback(async (userPosition) => {
    const distance = await getDistanceFrom([
      userPosition,
      ...cities.map((city) => city.coordinates),
    ]);

    setCitiesDistance(distance);
  }, []);

  const handleChangeAvatar = useCallback(
    (avatar: UploadMutation) => {
      if (user.id) {
        userMutation.mutate({
          userId: user.id,
          avatar: avatar.upload.id,
        });
      }
    }
  );
}
```

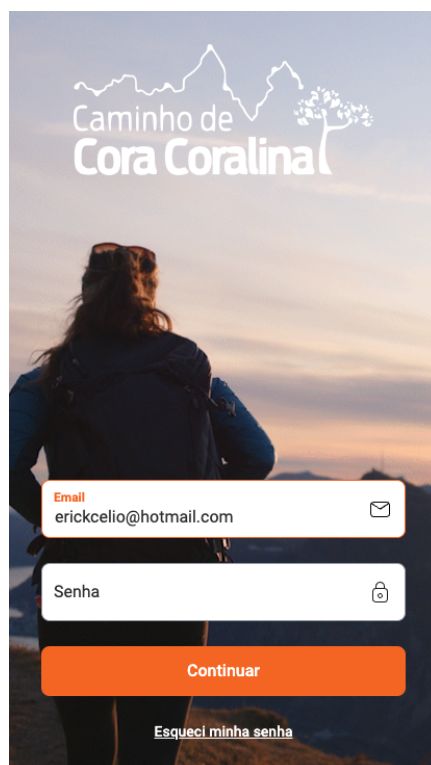
Fonte: AUTORES, 2021.

**Figura 11** - Página inicial do app.

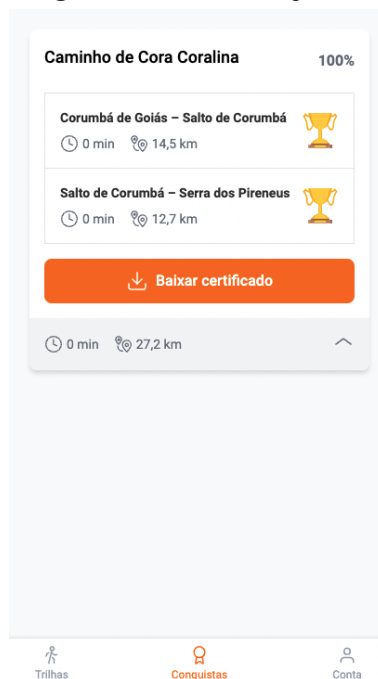


**Fonte:** AUTORES, 2021.

**Figura 12** - Tela de login.



**Fonte:** AUTORES, 2021.

**Figura 13** - Tela de conquistas.

**Fonte:** AUTORES, 2021.

Após a finalização do desenvolvimento, efetuamos as configurações de CI/CD para deploy da aplicação, para o Strapi utilizamos o *Heroku*, e para o front-end utilizamos a *Vercel* ambos *PaaS* com planos gratuitos onde foi possível disponibilizar para outras pessoas testarem sem nenhuma preocupação com toda a organização de infraestrutura, e também sendo possível validar simulando um ambiente de produção.

### 3.1.4. Verificação

Após a finalização do desenvolvimento foi efetuado uma bateria de testes funcionais realizados em um ambiente controlado, para garantir que todas as funcionalidades desenvolvidas estão de acordo com o estabelecido no escopo do projeto, portanto ainda não foram aplicados testes com usuários reais.

### 3.1.5. Manutenção

Não tivemos nenhuma evidência de falha e nenhum ponto de quebra da aplicação, sendo assim, não foi necessário nenhum tipo de manutenção na aplicação.

## **4. RESULTADOS**

Como resultados alcançados neste trabalho, foi possível aplicar conhecimentos e técnicas de desenvolvimento de software, que possibilitou na realização de toda uma documentação com base no levantamento de requisitos, prototipação e design de aplicativos, desenvolvimento de um backoffice e de uma aplicação baseada nas rotinas de usuários que necessitam de ter acesso às trilhas do Caminho de Cora, podendo assim, ver seus trajetos, imagens, informações e estabelecimentos próximos e ao concluir todas as trilhas é possível fazer o download do seu certificado e visualizar suas conquistas.

## 5. CONCLUSÃO E CONSIDERAÇÕES FINAIS

A realização desse projeto possibilitou aos integrantes do grupo a aplicação de toda a teoria adquirida no decorrer da graduação, tanto na pesquisa bibliográfica quanto na aplicação de ferramentas para o desenvolvimento da solução para o problema em questão.

Conseguimos construir um *software* performático e que atende aos requisitos levantados inicialmente e com alto nível de manutenibilidade, pois foi feito com a linguagem de desenvolvimento *JavaScript* a qual é amplamente difundida no mercado de trabalho atual.

Com a utilização dos *frameworks Strapi* e *React* foi possível obter maior produtividade durante o desenvolvimento do *software* devido ao fato de que os mesmos disponibilizam diversas funcionalidades já implementadas e testadas, tornando mais simples o desenvolvimento dos requisitos, fazendo com que a equipe mantenha o foco nas regras de negócio do sistema.

A aplicação da metodologia cascata foi suficiente para o desenvolvimento do projeto, mas acreditamos que para atualizações e novas *features* futuras, sendo possível tirar proveito de outras metodologias mais flexíveis e que permitam processos em paralelo, para maior produtividade da equipe caso seja desenvolvido por mais pessoas.

Ao decorrer do projeto tivemos várias ideias, mas decidimos focar em entregar as principais funcionalidades que fariam diferença para o usuário, resolvendo os problemas levantados e ressaltados pelo mesmo, mas sendo válido ser implementado futuramente novas *features* para incentivar o uso do aplicativo por meio da gamificação, como por exemplo uma adição de um ranking, onde seria incentivado os usuários percorrerem cada trajeto com o menor tempo, podendo ter algum tipo de prêmio ou bonificação, o desenvolvimento de uma interface acessível, para os usuários que acessam a aplicação em um *Desktop* ou *Tablet*, pois focamos essencialmente na experiência para dispositivos móveis e também um teste de campo com usuários, para detectar pontos de melhoria e *feedback* para novas *features*, que façam sentido implementar no aplicativo.



## REFERÊNCIAS

ALMEIDA, Maria. *O caminho de cora coralina - turismo literário ou marketing do turismo?*. Revista Sapiência: Sociedade, Saberes e Práticas Educacionais, 2020. Disponível em: <<https://www.revista.ueg.br/index.php/sapiencia/article/view/10083>>. Acesso em: 25 de fev. de 2021.

*Aplicativo Nativo x Híbrido: Qual escolher?*. Sthima Blog, 2020. Disponível em: <<https://sthima.com.br/blog/aplicativo-nativo-x-hibrido-qual-escolher/>>. Acesso em: 08 de mai. de 2021.

*Caminho de Cora Coralina. Caminho de Cora Coralina*, c2019. Página. Disponível em: <<https://caminhodecoracoralina.com.br>>. Acesso em: 20 de fev. de 2021.

*Caminho de Cora Coralina. Rede Trilhas*, c2020. Página. Disponível em: <<http://www.redetrilhas.org.br/w3/index.php/as-trilhas/trilha-regional/cora-coralina>>. Acesso em: 01 de mai. de 2021.

*Caminho de Cora lança o Certificado do Peregrino*. Portal Goiás, 2021. Disponível em: <<https://www.goias.gov.br/servico/39-turismo/124209-caminho-de-cora-lan%C3%A7a-o-certificado-do-peregrino.html>>. Acesso em: 25 de fev. de 2021.

*Caminho dos Veadeiros. Rede Trilhas*, c2020. Página. Disponível em: <<http://www.redetrilhas.org.br/w3/index.php/as-trilhas/trilha-regional/caminhos-dos-veadeiros>>. Acesso em: 01 de mai. de 2021.

CARVALHO, Rafael. *O que é a gamificação e como ela funciona?*. edools, 2016. Disponível em: <<https://www.edools.com/o-que-e-gamificacao/>>. Acesso em: 20 de jun. de 2021.

DIAS-NETO, Arilo Claudio; SPINOLA, R.; TRAVASSOS, Guilherme Horta. Developing software technologies through experimentation: experiences from the battlefield. In: *XIII Ibero-American Conference on Software Engineering*. 2010.

MARQUES, Julia. *O que significa iOS? Conheça cinco fatos sobre o sistema do iPhone*. Techtudo, 2019. Disponível em: <<https://www.techtudo.com.br/listas/2019/07/o-que-significa-ios-conheca-cinco-fatos-sobre-o-sistema-do-iphone.ghtml>>. Acesso em: 05 de mai. de 2021.

MENEZES, Pedro. *O aprendizado brasileiro das trilhas de longo curso no mundo*. ((o)) ecco, 2017b. Disponível em:

<<https://www.oeco.org.br/colunas/o-aprendizado-brasileiro-das-trilhas-de-longo-curso-no-mundo/>>. Acesso em: 01 de mai. de 2021.

MENEZES, Pedro. *O Brasil no caminho das trilhas de longo curso*. ((o)) ecco, 2017a. Disponível em:

<<https://www.oeco.org.br/colunas/o-brasil-no-caminho-das-trilhas-de-longocurso/#:~:text=Parque%20Nacional%20da%20Serra%20do,a%20trilha%20de%20longo%20curso.>>. Acesso em: 01 de mai. de 2021.

MEYER, Julio. *As Trilhas de Longo Curso e as unidades de conservação: sinergia pela natureza*. ((o)) ecco, 2020. Disponível em:

<<https://www.oeco.org.br/analises/as-trilhas-de-longo-curso-e-as-unidades-de-conservacao-sinergia-pela-natureza/>>. Acesso em: 01 de mai. de 2021.

MEYER, Maximiliano. *A história do Android*. Oficina da Net, 2020. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>. Acesso em: 02 de mai. de 2021.

NEVES, Rodrigo. *Nativo ou Híbrido: Qual solução adotar?*. VitaminaWeb, 2017. Disponível em: <<https://www.vitaminaweb.com.br/nativo-ou-hibrido-qual-solucao-adotar/>>. Acesso em: 20 de jun. de 2021.

PRESSMAN, Roger; MAXIM, Bruce. *Engenharia de Software-8ª Edição*. McGraw Hill Brasil, 2016.

*PWA (Progressive Web App): O Que É e Como Criar o Seu (+3 Exemplos)*. NEILPATEL, c2021. Disponível em: <<https://neilpatel.com/br/blog/pwa-o-que-e/>>. Acesso em: 15 de nov. de 2021.

SILVA, Eduardo. *5 melhores linguagens de programação para Android*. geekhunter, 2020. Disponível em:

<[https://blog.geekhunter.com.br/linguagens-de-programacao-para-android/#Como\\_funciona\\_o\\_sistema\\_Android](https://blog.geekhunter.com.br/linguagens-de-programacao-para-android/#Como_funciona_o_sistema_Android)>. Acesso em: 02 de mai. de 2021.

TOKARNIA, Mariana. *Celular é o principal meio de acesso à internet no país*. Agência Brasil, 2020. Disponível em:

<<https://agenciabrasil.ebc.com.br/economia/noticia/2020-04/celular-e-o-principal-meio-de-acesso-internet-no-pais#:~:text=Os%20dados%20mostram%20que%2079,88%2C5%25%20em%202018.>>. Acesso em: 01 de mai. de 2021.

WARREN, Jennifer. *PROGRESSIVE WEB APPS – PROS, CONS, COST AND POPULAR TECHNOLOGIES USED FOR DEVELOPMENT*. softwarebrothers, 2020. Disponível em: <<https://softwarebrothers.co/blog/progressive-web-apps-pros-cons-cost-and-popular-technologies-used-for-development/>>. Acesso em: 15 de nov. de 2021.