

KAIC DE OLIVEIRA BARROS

**UMA LINHA DE PRODUTO DE SOFTWARE
PARA O TURISMO EM SERGIPE**

ITABAIANA

2021

KAIC DE OLIVEIRA BARROS

UMA LINHA DE PRODUTO DE SOFTWARE PARA O TURISMO EM SERGIPE

Trabalho de Conclusão de Curso submetido ao Departamento de Sistemas de Informação da Universidade Federal de Sergipe, como requisito para a obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal de Sergipe – UFS
Departamento de Sistemas de Informação – DSI

Orientador: Prof. Dr. Raphael Pereira de Oliveira

ITABAIANA

2021

KAIC DE OLIVEIRA BARROS

UMA LINHA DE PRODUTO DE SOFTWARE PARA O TURISMO EM SERGIPE

Trabalho de Conclusão de Curso submetido ao Departamento de Sistemas de Informação da Universidade Federal de Sergipe, como requisito para a obtenção do título de Bacharel em Sistemas de Informação.

Itabaiana, 03 de setembro de 2021.

BANCA EXAMINADORA:

Prof. Dr. Raphael Pereira de Oliveira
Orientador
DSI/UFS

Prof. Dr. Marcos Barbosa Dósea
DSI/UFS

**Prof. Msc. André Vinícius Rodrigues
Passos Nascimento**
DSI/UFS

ITABAIANA
2021

*Dedico esse trabalho aos meus pais, Anailza Alves de Oliveira e Gilson de Jesus Barros
por todo o esforço e estarem sempre presentes.*

Agradecimentos

Esse trabalho não seria possível sem as oportunidades que tive. Nenhuma destas oportunidades seriam possíveis se não fosse pelo esforço e o apoio dos meus pais Ana e Gilson, que se dedicaram ao máximo para garantir o melhor para mim, a vocês, faltam palavras para agradecer.

Aos meus irmãos, Guilbert e Luigi, que sempre torceram por mim e me aturarem durante as noites acordado. À minha tia Ilma, e prima Kika, que foram como mães para mim sempre estendendo a mão nos momentos que precisei. Vocês foram de suma importância para que eu conseguisse alcançar esse objetivo. Muito obrigado por vocês existirem!

Agradeço a todos os meus familiares que sempre estiveram na torcida por mim. Muito obrigado por fazerem parte da minha vida, me ajudando e me apoiando em tudo que foi possível.

Ao Prof. Dr. Raphael Oliveira, pelos ensinamentos, paciência e dedicação, fazendo com que tudo isso se tornasse possível. Fica aqui o meu agradecimento e a minha admiração pelo seu exemplo de competência.

Também agradeço a todos os amigos de graduação, com quais pude compartilhar momentos e experiências, em especial os companheiros Daniel, Jadson e Suellington que contribuíram para este trabalho.

E por fim, quero agradecer a todos que fazem parte do departamento de sistemas de informação, sem exceções, técnicos, secretários e a todos os professores, obrigado por terem contribuído com a minha formação com seus conhecimentos e seus ensinamentos durante as aulas.

*"São as nossas escolhas que revelam
o que realmente somos, muito mais
do que as nossas qualidades."*

Resumo

É comum encontrarmos uma grande variedade de softwares com a finalidade de ajudar o usuário de alguma forma. Contudo, notamos que o estado de Sergipe ainda carece de softwares para a divulgação de seus pontos turísticos. Como o estado conta com diversas cidades turísticas, softwares podem ser construídos de acordo com as atrações de cada cidade. Para atender a essa demanda, o desenvolvimento de software deve levar em consideração as funcionalidades, atrações turísticas e opções, presentes em cada cidade. Uma forma de desenvolvimento de software, levando em consideração as possíveis combinações de funcionalidades para se criar softwares distintos, acontece via Linhas de Produto de Software (LPS). Uma LPS considera funcionalidades que são comuns e variáveis a todos os produtos (softwares) da linha e gerencia a instanciação de produtos individuais. Com base nesses conceitos, este trabalho construiu uma LPS para o turismo em Sergipe. Essa linha permitirá a criação de produtos para as distintas cidades do estado de Sergipe. Os produtos da LPS desenvolvida foram avaliados de acordo com sua facilidade de uso, utilidade e intenção de uso no futuro. Os resultados foram bastante promissores e incentivam a continuidade de inclusão de novas funcionalidades na LPS.

Palavras-chave: Linhas de Produto de Software, Reuso de Software, Turismo em Sergipe.

Abstract

It is common to find a wide variety of software with the purpose of helping the user. However, we noted that the state of Sergipe still lacks of softwares to publicize its sights. Since the state has several tourist cities, softwares can be built according to the attractions of each city. To meet this demand, software development must take into account features, tourist attractions, and options present in each city. One way to tackle this, taking into account possible combinations of features to create different products, is through Software Product Lines (SPL). An SPL considers functionalities that are common and variable to all products (softwares) in the line and manages the instantiation of individual products. Based on these concepts, this work built an SPL for tourism in Sergipe. This line will allow the creation of products for the different cities in the state of Sergipe. The developed SPL products have been evaluated based on their ease of use, usefulness and intention of use in the future. The results were very promising and encourage the inclusion of new features within the SPL.

Keywords: Software Product Lines, Software Reuse, Tourism in Sergipe.

Lista de ilustrações

Figura 1 – Notação gráfica para <i>features</i> opcionais e obrigatórias. Um marcador vazio indica uma <i>feature</i> opcional e um marcador preenchido indica uma <i>feature</i> obrigatória.	18
Figura 2 – Notação gráfica para uma escolha dentre muitas. Essa opção corresponde a um operador xor generalizado.	19
Figura 3 – Notação gráfica para uma escolha entre muitas dentre muitas. Essa escolha corresponde ao operador lógico ou.	19
Figura 4 – Exemplo com <i>features</i> abstratas e concretas.	20
Figura 5 – Feature Model da LPS para o turismo em Sergipe.	24
Figura 6 – Modelo do Banco de Dados da LPS para o turismo em Sergipe.	26
Figura 7 – Protótipos do produto baseado na cidade de Aracaju	28
Figura 8 – Protótipos do produto baseado na cidade de São Cristóvão	35
Figura 9 – Protótipos do produto baseado na cidade de Laranjeiras	36
Figura 10 – Organização das pastas e arquivos da LPS.	37
Figura 11 – Organização da pasta src.	37
Figura 12 – Organização das pastas e arquivos da Web API.	38
Figura 13 – Produto Completo Patrocinado Aracaju	39
Figura 14 – Produto Completo Publicitário Itabaiana	40
Figura 15 – Produto Básico Laranjeiras	41
Figura 16 – Dados demográficos	44
Figura 17 – Boxplot da Percepção da Facilidade de Uso	47
Figura 18 – Boxplot da Percepção de Utilidade	48
Figura 19 – Boxplot da Intenção de Uso	50

Lista de tabelas

Tabela 1 – Comparativo de trabalhos que construíram ferramentas para o turismo	22
Tabela 2 – Afirmativas positivas e negativas do questionário fechado	45
Tabela 3 – Análise das variáveis PEOU, PU e ITU	47
Tabela 4 – Resumo dos dados coletados da variável subjetiva PEOU	48
Tabela 5 – Resumo dos dados coletados da variável subjetiva PU	49
Tabela 6 – Resumo dos dados coletados da variável subjetiva ITU	49
Tabela 7 – Análise valores do Cliff's Delta das variáveis PEOU, PU e ITU	50

Lista de abreviaturas e siglas

API	Application Programming Interface
LPS	Linha de Produto de Software
RAM	Random Access Memory
SIT	Sistemas de Informação Turística
SPL	Software Product Line
SQL	Structured Query Language
TCC	Trabalho de Conclusão de Curso

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	14
1.3	Metodologia	14
1.4	Contribuições	15
1.5	Estrutura do trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Linhas de Produto de Software	16
2.1.1	Feature Model	16
2.1.2	Codificação em LPS	19
2.1.2.1	Implementação de Variabilidade	20
2.2	Trabalhos Relacionados	21
3	UMA LPS PARA O TURISMO EM SERGIPE	23
3.1	Feature Model	23
3.2	Banco de dados e configuração dos produtos	25
3.3	Protótipos	28
3.4	Implementação	30
3.4.1	Implementação da Linha de Produtos	30
3.4.2	Implementação da Web API	32
3.5	A LPS para o Turismo	32
4	AVALIAÇÃO DOS PRODUTOS DA LPS	42
4.1	Planejamento	42
4.1.1	Teste Piloto	43
4.2	Coleta	43
4.3	Análise	46
4.3.1	QP01: O aplicativo é fácil de ser usado?	47
4.3.2	QP02: O aplicativo é útil?	48
4.3.3	QP03: Existe intenção de uso do aplicativo futuramente?	49
4.3.4	Teste Cliff's Delta (Effect Size)	49
4.4	Discussão	51
5	CONSIDERAÇÕES FINAIS	53

REFERÊNCIAS	54
ANEXOS	56
ANEXO A – TERMO DE CONSENTIMENTO	57
ANEXO B – TAREFAS (A)	59
ANEXO C – TAREFAS (B)	61
ANEXO D – QUESTIONÁRIO DE AVALIAÇÃO DE VARIÁVEIS SUBJETIVAS	63

1 Introdução

Vivemos em uma época em que sistemas automatizam muitas tarefas cotidianas, muitas das vezes nem percebidas por seus usuários. Graças ao desenvolvimento de softwares centenas de pessoas estão melhorando sua qualidade de vida. Um exemplo bem simples disso é o fato de que hoje, qualquer pessoa com acesso à Internet e uma conta bancária, não precisa mais sair de casa para ir às compras ou pagar suas contas, pois quase tudo pode ser comprado e pago pela Internet. Utilizando softwares criados para essas finalidades, podemos evitar espera em fila de banco, o stress do trânsito, além dos gastos com combustível ou transporte.

Uma das formas de se desenvolver software é através do reuso. Experiências passadas no reuso de software focavam em reutilizar somente pequenos pedaços de código ou realizar clones de código. Contudo, essas técnicas não se mostraram viáveis (CLEMENTS; NORTHROP, 2002). Projetos de software dos anos 1990 mostraram que sem planejamento, os custos para reuso podem ser mais altos do que desenvolver desde o início (POHL; BÖCKLE; LINDEN, 2005).

Por outro lado, as Linhas de Produto de Software (LPS - do inglês *Software Product Lines - SPL*) surgiram como uma tendência de reuso, onde o reuso é planejado em todas as fases do desenvolvimento do software (CLEMENTS; NORTHROP, 2002). A LPS permite que desenvolvedores de software reusem artefatos de todo o ciclo de desenvolvimento para criar produtos individuais aos seus clientes, sejam produtos na plataforma web, desktop ou móvel (*mobile*).

A plataforma *mobile* revolucionou o mercado de softwares. As aplicações *mobile* possuem vantagens como a facilidade de interação com a interface do usuário (PRESSMAN; MAXIM, 2016). Atualmente, é possível realizar todas as tarefas citadas anteriormente com um simples toque no seu celular inteligente (*smartphone*). Isso acabou trazendo às empresas um novo cenário de desenvolvimento: o dos aplicativos (Apps) *mobile*.

Um dos domínios no qual encontramos aplicativos *mobile* é o de turismo. Esses aplicativos *mobile* dão suporte ao turismo de acordo com diversas necessidades dos usuários, entre elas esses aplicativos podem ajudar a: auxiliar turistas a conhecer e aprofundar seus conhecimentos sobre o local visitado; disseminar informações sobre eventos; exibir hospedagens; e mostrar restaurantes da região. Eles podem ajudar desde a escolha do destino até a reserva de hotéis, aluguéis de carros e definição do roteiro.

Com base nesses conceitos, esse Trabalho de Conclusão de Curso (TCC) desenvolveu uma LPS para o turismo em Sergipe. A LPS construída fornece suporte para o turismo nas diversas cidades do estado de Sergipe (SE), e foi validada através de um estudo empírico.

1.1 Motivação

O nordeste brasileiro é uma região com lindas paisagens, histórias fascinantes, e muito visitada por turistas de todos os lugares. Devido a falta de suporte tecnológico aos turistas no estado de SE e com o intuito de ajudar os turistas e o crescimento da própria comunidade local, encontramos uma oportunidade para desenvolver esse TCC.

Com o intuito de facilitar o acesso à informação foi pensado na construção de uma LPS na plataforma mobile que traga informações em texto, áudio, imagens e vídeos, relacionadas aos pontos turísticos de diversas regiões do estado de SE. Os produtos da LPS também irão apresentar passeios predefinidos passando por pontos turísticos, com o objetivo de mostrar uma rota para o usuário seguir e conhecer a história ou curiosidades desses pontos.

1.2 Objetivos

O objetivo geral desse TCC é construir uma LPS para o turismo no estado de Sergipe. Para alcançar o objetivo geral desse TCC, foram executados os seguintes objetivos específicos:

- Definir as funcionalidades da LPS;
- Implementar os códigos da LPS;
- Contribuir com o acesso a informações relacionadas a pontos e passeios turísticos de cada região do estado de Sergipe;
- Validar a LPS proposta.

1.3 Metodologia

Esse trabalho possui uma metodologia classificada como: de natureza aplicada; com uma abordagem metodológica quantitativa e qualitativa; sendo uma pesquisa exploratória com um estudo empírico (NASCIMENTO; SOUSA, 2015).

A natureza do trabalho é aplicada porque o trabalho gera conhecimentos práticos para o turismo no estado de SE, utilizando diversas cidades e pontos turísticos do estado de SE.

O TCC também utilizou dos métodos quantitativos e qualitativos para apresentar seus resultados. Os métodos quantitativos foram caracterizados pela aplicação de medidas padronizadas e sistemáticas através de medidas estatísticas. Para complementar o método

quantitativo, esse TCC também utilizou o método qualitativo através da análise de opiniões de usuários de produtos da LPS.

Por fim, esse TCC possui o viés exploratório, o qual permitiu o conhecimento gradativo do contexto estudado e a elaboração de hipóteses e questões de pesquisas mais claras.

Com o intuito de avaliar os produtos da LPS construída, um estudo empírico foi realizado avaliando a facilidade de uso, utilidade e intenção de uso no futuro dos produtos da LPS.

1.4 Contribuições

- Divulgação do turismo em Sergipe através dos produtos da LPS;
- Disponibilização de um aplicativo configurável que irá demonstrar e validar os produtos da linha para o turismo no estado de Sergipe;
- Disponibilização da LPS para futuros estudos.

1.5 Estrutura do trabalho

O restante do trabalho está estruturado como segue. No capítulo 2 é apresentada uma revisão bibliográfica e alguns trabalhos encontrados relacionados com o tema do trabalho. No capítulo 3 é apresentado o desenvolvimento da linha de produtos para o turismo em Sergipe. O capítulo 4, trata do experimento que foi realizado nesse TCC. No capítulo 5 são apresentadas as conclusões e as propostas de trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo serão abordados conceitos fundamentais para a compreensão desse TCC. Serão apresentados conceitos sobre Linhas de Produto de Software (LPS) incluindo *feature model* e codificação em LPS com foco na programação para dispositivos móveis. Para finalizar o capítulo, são apresentados os trabalhos relacionados com esse TCC.

2.1 Linhas de Produto de Software

As Linhas de Produtos de Software (LPS) visam capacitar os fornecedores de software a adaptar os produtos de software aos requisitos de clientes individuais (APEL et al., 2016). Dessa forma, é possível construir variações do aplicativo com base nas necessidades do cliente e aumentar a diversidade de produtos em seu portfólio.

Segundo APEL et al. (2016), as linhas de produtos de software seguem um desenvolvimento que surgiu na manufatura industrial. Começando com a fabricação manual de bens individuais, o advento da produção em massa ampliou o processo de produção para grandes quantidades, mas negligenciou o individualismo, pois todos os produtos eram iguais. Como consequência, clientes acabam comprando sistemas com funcionalidades que não necessitam e acabam nunca usando. Evitando esse tipo de problema, os fabricantes planejaram e projetaram sistematicamente linhas de produtos para cobrir todo um espectro de possíveis produtos e variações dos mesmos, atendendo às necessidades e desejos individuais de muitos clientes. As LPS seguem a mesma ideia e consideram a produção em massa e a personalização em massa no desenvolvimento de software.

Nesse TCC, foram elicitadas as variabilidades e implementada uma LPS para o estado de Sergipe. Uma das formas de definir as variabilidades de uma LPS é através da definição de suas possíveis funcionalidades, que podem ser elicitadas através de um *feature model*. A implementação da LPS também deve suportar as variabilidades de acordo com o produto a ser criado. Assim, os conceitos utilizados nesse TCC (*feature model* e implementação) são apresentados a seguir.

2.1.1 Feature Model

A tradução de *Feature Model* é Modelo de Recursos ou Modelo de Funcionalidades. APEL et al. (2016) diz que, o conceito de um recurso é inerentemente difícil de definir com precisão, pois captura, por um lado, as intenções das partes interessadas de uma linha de produtos, incluindo usuários finais e, por outro, conceitos de nível de design

e implementação usados para estruturar, variar e reutilizar artefatos de software. Mas, existem muitas outras definições de *feature models*, tais como:

- (KANG et al., 1990): “aspecto, qualidade ou característica visível ou proeminente do usuário, de um sistema ou sistemas de software”;
- (KANG et al., 1998): “uma abstração funcional distintamente identificável que deve ser implementada, testada, entregue e mantida”;
- (KRZYSZTOF; EISENECKER, 2000): “uma característica distinguível de um conceito (por exemplo, sistema, componente etc.) que é relevante para algumas partes interessadas do conceito”;
- (BOSCH, 2000): “uma unidade lógica de comportamento especificada por um conjunto de requisitos funcionais e não funcionais”;
- (CHEN et al., 2005): “uma característica do produto na visão do usuário ou cliente, que consiste essencialmente em um conjunto coeso de requisitos individuais”;
- (BATORY; SARVELA; RAUSCHMAYER, 2004): “uma característica do produto usada para distinguir programas dentro de uma família de programas relacionados”;
- (CLASSEN; HEYMANS; SCHOBENS, 2008): “um trigêmeo, $f = (R, W, S)$, onde R representa os requisitos que o recurso satisfaz, W as premissas que o recurso adota sobre seu ambiente e S é sua especificação”;
- (ZAVE, 2003): “uma unidade opcional ou incremental de funcionalidade”;
- (BATORY, 2005): “um incremento da funcionalidade do programa”;
- (APEL et al., 2010): “uma estrutura que estende e modifica a estrutura de um determinado programa para satisfazer os requisitos de uma parte interessada, implementar e encapsular uma decisão de projeto e oferecer uma opção de configuração”.

Finalmente, APEL et al. (2016) define recurso como um comportamento característico ou visível ao usuário final de um sistema de software.

Um *feature model* documenta os recursos de uma linha de produtos e seus relacionamentos (APEL et al., 2016). Seu objetivo é a modelagem das funcionalidades pertencentes ao produto incluindo suas interdependências.

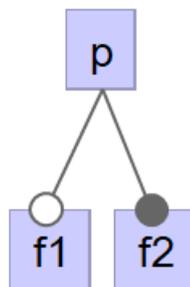
Além disso, podemos definir restrições entre as *features* existentes. Assim podemos garantir que uma determinada *feature* não esteja presente no produto se uma outra *feature* não estiver. Também podemos criar relações entre *features* de uma maneira que as tornem mutualmente excludentes.

Para representar o modelo de *feature* utilizamos um diagrama de *features*, uma notação gráfica em forma de árvore cujos nós são rotulados com nomes de *features*. Assim é possível transmitir várias relações pai-filho entre *features* e suas restrições.

Se uma *feature* f é filha de outra *feature* p , f pode ser selecionada apenas quando p também é selecionada. Normalmente, um diagrama de *features* inclui relações mútuas entre as *features*. Por exemplo, a *feature* pai indica um conceito mais geral e a filha uma especialização (APEL et al., 2016).

As *features* obrigatórias são diferenciadas por um pequeno círculo na *feature* filha, onde: um círculo vazio indica uma *feature* opcional; enquanto o círculo preenchido indica uma *feature* obrigatória. Na Figura 1 podemos ver essa relação. Note que o nó pai é rotulado com p , o nó filho com f . A *feature* $f1$ é opcional e a *feature* $f2$ é obrigatória.

Figura 1 – Notação gráfica para *features* opcionais e obrigatórias. Um marcador vazio indica uma *feature* opcional e um marcador preenchido indica uma *feature* obrigatória.



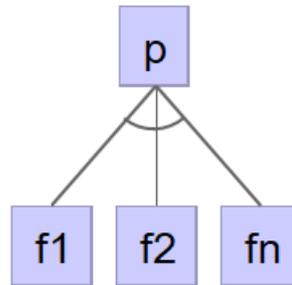
Fonte: APEL et al. (2016).

Na Figura 2, as arestas entre uma *feature* pai p e um grupo de *features* filhas $\{f1...fn\}$ são conectadas através de um arco vazio. Esse elemento gráfico indica a escolha de exatamente uma *feature* dentro um grupo de *features*. Na lógica proposicional, é uma generalização de uma disjunção exclusiva. Essa construção é chamada de escolha alternativa ou mutuamente exclusiva.

A Figura 3 mostra as *features* filhas conectadas através de um arco preenchido. Esse elemento gráfico indica uma escolha irrestrita de uma ou mais *features* de um grupo de *features*. Na lógica proposicional, denota-se como uma disjunção inclusiva.

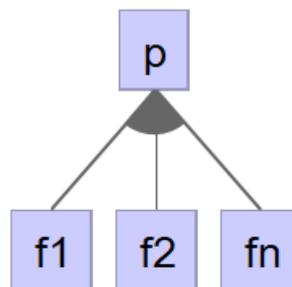
Também podemos ter *features* concretas e *features* abstratas. As *features* abstratas são usadas apenas para fins de estruturação e documentação e não estão relacionadas aos artefatos da LPS. *Features* concretas representam *features* que possuem relacionamento com artefatos da LPS. Na Figura 4 as *features* abstratas são denotadas por caixas com fundo claro e as concretas por caixas fundo escuro.

Figura 2 – Notação gráfica para uma escolha dentre muitas. Essa opção corresponde a um operador xor generalizado.



Fonte: APEL et al. (2016).

Figura 3 – Notação gráfica para uma escolha entre muitas dentre muitas. Essa escolha corresponde ao operador lógico ou.



Fonte: APEL et al. (2016).

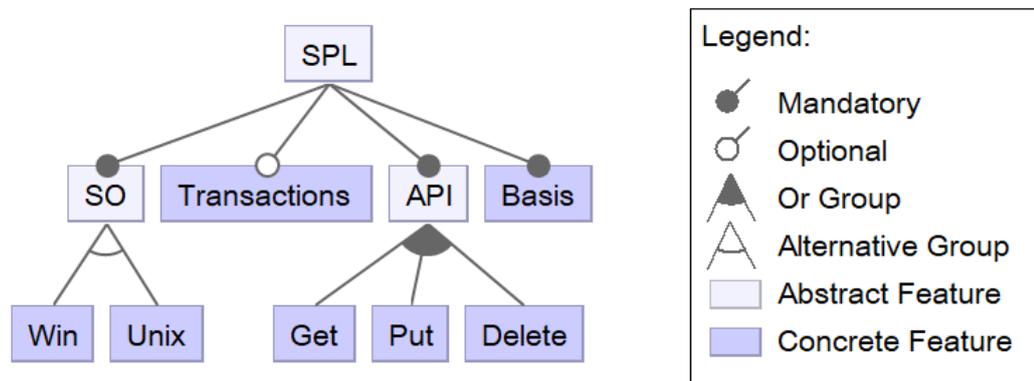
Abaixo da árvore da Figura 4 estão listadas as restrições presentes no modelo. Podemos usar alguns símbolos nestas notações: o símbolo de implicação (\Rightarrow), o operador se e somente se (\Leftrightarrow), o ou lógico (\vee), o e lógico (\wedge) e a negação (\neg). Ainda na Figura 4, podemos ver uma restrição entre as *features* *Transactions* e *Unix*, que expressa uma exclusão entre elas. Logo se escolhermos o sistema operacional *Unix* não poderemos adicionar a *feature Transactions*.

2.1.2 Codificação em LPS

O subprocesso de implementação de domínio lida com a codificação de componentes de software reutilizáveis para toda a LPS.

A entrada para este subprocesso pode ser a arquitetura de referência, incluindo uma lista de artefatos de software reutilizáveis a serem desenvolvidos na implementação do domínio, os próprios requisitos da LPS ou o *feature model* definido para a LPS.

A saída da implementação da LPS abrange os artefatos detalhados de implementa-

Figura 4 – Exemplo com *features* abstratas e concretas.

Fonte: APEL et al. (2016).

ção de componentes de software reutilizáveis.

A implementação em LPS difere da implementação de sistemas simples porquê:

- O resultado da implementação do domínio consiste em componentes configuráveis fracamente acoplados, e não em um aplicativo em execução;
- Cada componente é planejado, projetado e implementado para a reutilização em diferentes contextos, ou seja, nas aplicações da linha de produtos. A interface de um componente reutilizável deve suportar os diferentes contextos;
- A implementação incorpora mecanismos de configuração nos componentes para possibilitar a variabilidade da LPS.

2.1.2.1 Implementação de Variabilidade

No contexto de programação para dispositivos móveis, esse trabalho utilizou o *framework React Native*. Em *React Native*, é possível criar componentes distintos com comportamentos diferentes. Então, pode-se renderizar apenas alguns dos elementos dependendo do estado da sua aplicação. A renderização condicional em *React Native* funciona da mesma forma que condições funcionam em *JavaScript*. Utiliza-se os operadores do *JavaScript* *if* e *else* para criar elementos representando o estado atual, e o *React Native* se encarrega de atualizar a interface do usuário.

Considere a Lista 2.1. Nela temos dois métodos renderizando componentes diferentes. Nesse exemplo é renderizado um dos componentes dependendo do valor do estado **saudacao**, o qual recebe seu valor direto da propriedade **props.saudacao**.

```
1 import React, { useState } from 'react'
2 import {
3   View,
4   Text
5 } from 'react-native'
6
7 export const RenderizacaoCondicional = (props) => {
8
9   const [saudacao, setSaudacao] = useState(props.saudacao)
10
11   const inscrevase = () => {
12     return <Text>Por favor, inscreva-se</Text>
13   }
14
15   const bemVindo = () => {
16     return <Text>Bem vindo de volta</Text>
17   }
18
19   return (
20     <View>
21       { saudacao ? bemVindo() : inscrevase() }
22     </View>
23   )
24 }
25
26 export default RenderizacaoCondicional
```

Lista 2.1 – Exemplo Renderização Condicional

2.2 Trabalhos Relacionados

Nesta seção serão apresentados alguns trabalhos que desenvolveram ferramentas voltadas ao suporte do turismo.

O trabalho de [Daramola, Adigun e Olugbara \(2008\)](#) é uma proposta de arquitetura em camadas de componentes essenciais reutilizáveis que podem ser aproveitados para o desenvolvimento de uma família de produtos de Sistemas de Informação Turística (SIT). A arquitetura apresentada fornece um ponto de partida para a evolução do SIT, capaz de fornecer serviços inteligentes de valor agregado aprimorados. A fim de mostrar que é suficientemente elástica e altamente reutilizável a interação do componente dentro da arquitetura foi modelada como um padrão arquitetônico em camadas. Segundo [Daramola, Adigun e Olugbara \(2008\)](#) a arquitetura é capaz de iniciar o surgimento de famílias TIS confiáveis.

Em [Park e Yim \(2015\)](#) é desenvolvido um aplicativo Android para turismo de

mídia inteligente em Gyeongju. O principal tipo de conteúdo que este aplicativo oferece é o vídeo de alta qualidade. Ele fornece notícias, TV, informações turísticas e informações sobre a vida, ele também permite que os usuários carreguem qualquer informação útil.

De acordo com os autores [Pashtan, Heusser e Scheuermann \(2004\)](#), CATIS é um sistema de informação turística ciente de contexto em dispositivos móveis que utiliza serviços da Web e tecnologias XML para sua implementação. O sistema CATIS incorpora uma série de variáveis de contexto relacionadas à mobilidade, como horário, localização e tipo de dispositivo. Os autores descrevem um algoritmo que se baseia em elementos de mobilidade de contexto, como a viagem do usuário, direção e rapidez, para formar áreas de atendimento pessoal. Seu sistema de informação turística de contexto experimental (CATIS) alavanca tecnologias XML e Web serviços para fornecer informações turísticas a usuários móveis com base nesses áreas de serviço e as preferências dos usuários.

A Tabela 1 abaixo demonstra o comparativo entre os trabalhos relacionados e o presente trabalho:

Tabela 1 – Comparativo de trabalhos que construíram ferramentas para o turismo

Autor	Utiliza LPS?	Suporte para o Turismo?	Plataforma Móvel?	Realizou Avaliação Empírica?
(DARAMOLA; ADIGUN; OLUGBARA, 2008)	X	X		
(PARK; YIM, 2015)		X	X	
(PASHTAN; HEUSSER; SCHEUERMANN, 2004)		X	X	
Presente Trabalho	X	X	X	X

Sendo assim, este trabalho tem como objetivo construir uma LPS para o turismo no estado de Sergipe, onde os produtos estarão disponíveis na plataforma móvel. Ainda, o presente trabalho realizou um estudo empírico para validar os produtos da LPS construída.

3 Uma LPS para o Turismo em Sergipe

Neste capítulo apresentamos todo o desenvolvimento da linha de produtos para o turismo em Sergipe. Serão apresentados o *feature model*, o banco de dados com a configuração dos produtos da linha, os protótipos, a implementação e os produtos da linha.

3.1 Feature Model

Uma *feature* é uma funcionalidade do sistema que é visível para um *stakeholder*. Ela permite a expressão de partes comuns e variáveis entre as instâncias e representa requisitos reusáveis e configuráveis. Na Figura 5 temos o *Feature Model* proposto pelo presente trabalho e podemos ver suas *features* e respectivas restrições. Esse *feature model* foi criado com a ferramenta FeatureIDE ¹.

Como já vimos, *features* obrigatórias são representadas por um pequeno círculo preenchido. Logo no modelo da Figura 5, o produto obrigatoriamente terá um sistema operacional (ANDROID ou IOS), terá que mostrar pontos turísticos no mapa, apresentar resumos sobre os pontos turísticos e sugerir algum tipo de passeio turístico. As *features* com um pequeno círculo aberto representam funcionalidades opcionais, funcionalidades que podem ser adicionadas a um produto da linha.

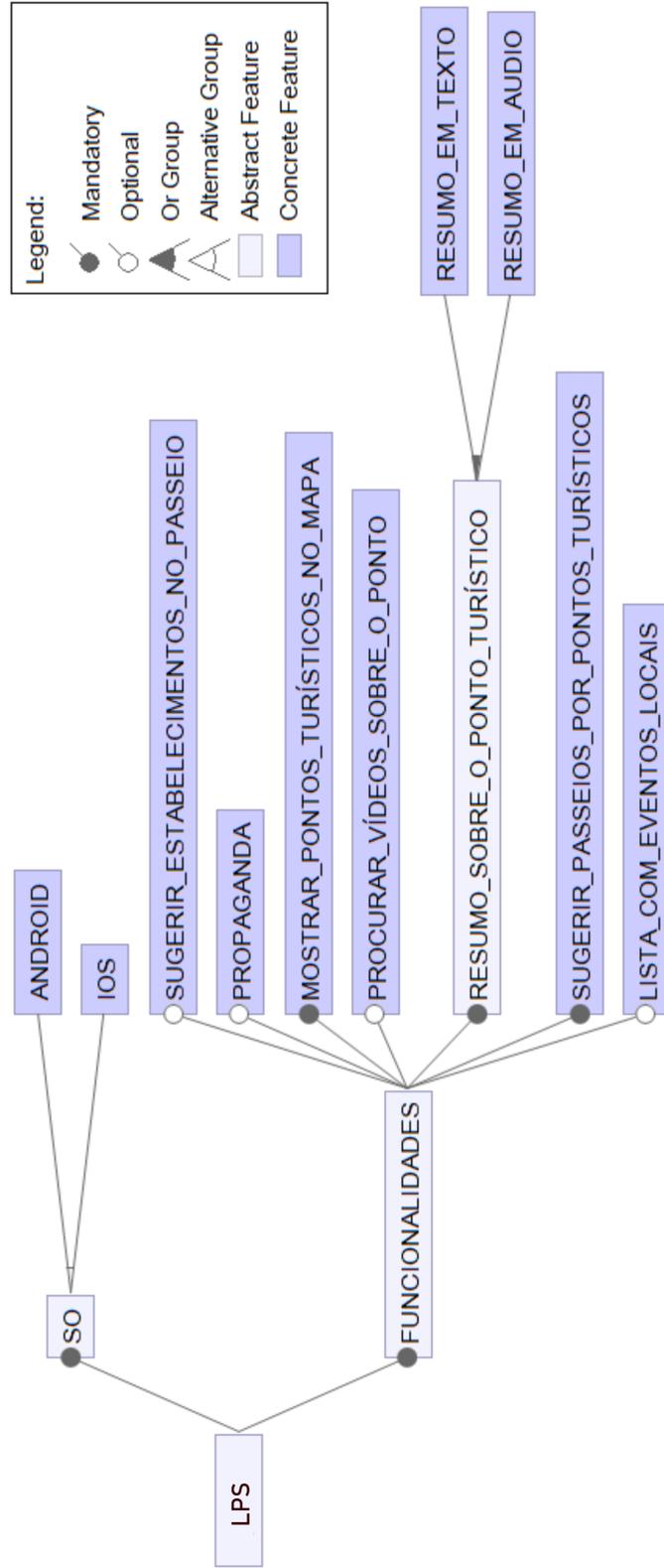
Na *feature SO* vemos uma notação gráfica que representa uma escolha única dentre muitas. Assim, somente poderemos escolher um tipo de sistema operacional para o produto.

Já na *feature RESUMO_SOBRE_O_PONTO_TURÍSTICO*, vemos uma notação gráfica que denota uma escolha irrestrita de uma ou mais *features* de um grupo de *features*. Então se, pelo menos, uma *feature* da coleção tiver sido selecionada, mas não houver outras restrições, o produto é válido. Logo podemos ter um produto com *RESUMO_EM_TEXTO* e *RESUMO_EM_ÁUDIO* ou um produto com uma das duas opções. Abaixo são descritas cada uma das *features* concretas do modelo:

- *ANDROID* - representa que o sistema operacional do produto será o Android;
- *IOS* - representa que o sistema operacional do produto será o IOS;
- *SUGERIR_ESTABELECIMENTO_NO_PASSEIO* - funcionalidade opcional a qual irá sugerir alguns estabelecimentos, que escolherem patrocinar o aplicativo, durante o passeio turístico;

¹ <https://www.featureide.de/>

Figura 5 – Feature Model da LPS para o turismo em Sergipe.



Fonte: Autor.

- *PROPAGANDA* - funcionalidade opcional que mostrará propagandas no aplicativo;
- *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* - funcionalidade obrigatória que irá mostrar os pontos turísticos da cidade no mapa;
- *PROCURAR_VÍDEOS SOBRE O PONTO* - funcionalidade opcional que irá procurar e mostrar vídeos relacionados ao ponto turístico;
- *RESUMO_EM_TEXTO* - funcionalidade que irá mostrar um resumo em texto sobre o ponto turístico;
- *RESUMO_EM_ÁUDIO* - funcionalidade que emitirá um áudio com um resumo sobre o ponto turístico;
- *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* - funcionalidade obrigatória que irá sugerir um percurso passando por alguns pontos turísticos da cidade;
- *LISTA_COM_EVENTOS_LOCAIS* - funcionalidade opcional que trará os eventos locais da cidade em um determinado período de tempo.

No modelo ainda podemos ver uma restrição entre as *features* *PROPAGANDA* e *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO*. Essa restrição tem intenção de criar uma exclusão entre elas, ou seja, se a *feature* *PROPAGANDA* for selecionada, a *feature* *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO* não pode estar selecionada.

3.2 Banco de dados e configuração dos produtos

Nessa seção será apresentado o modelo do banco de dados e como é feita a configuração dos produtos da linha. Para ajudar no entendimento, será utilizada a Figura 6 com o modelo do banco de dados e logo abaixo uma lista com a descrição de cada tabela e seus relacionamentos.

- Tabela **Usuario**: guarda informações para o login e informações do perfil do usuário, ela não tem relacionamento com nenhuma outra tabela;
- Tabela **País**: guarda informações de países, ela tem um relacionamento de um para muitos com a tabela Estado;
- Tabela **Estado**: guarda informações de estados, ela tem um relacionamento de um para muitos com a tabela Cidade;
- Tabela **Cidade**: guarda informações de cidades, ela tem um relacionamento de um para um com a tabela Produto e um relacionamento de um para muitos com a tabela Endereco;

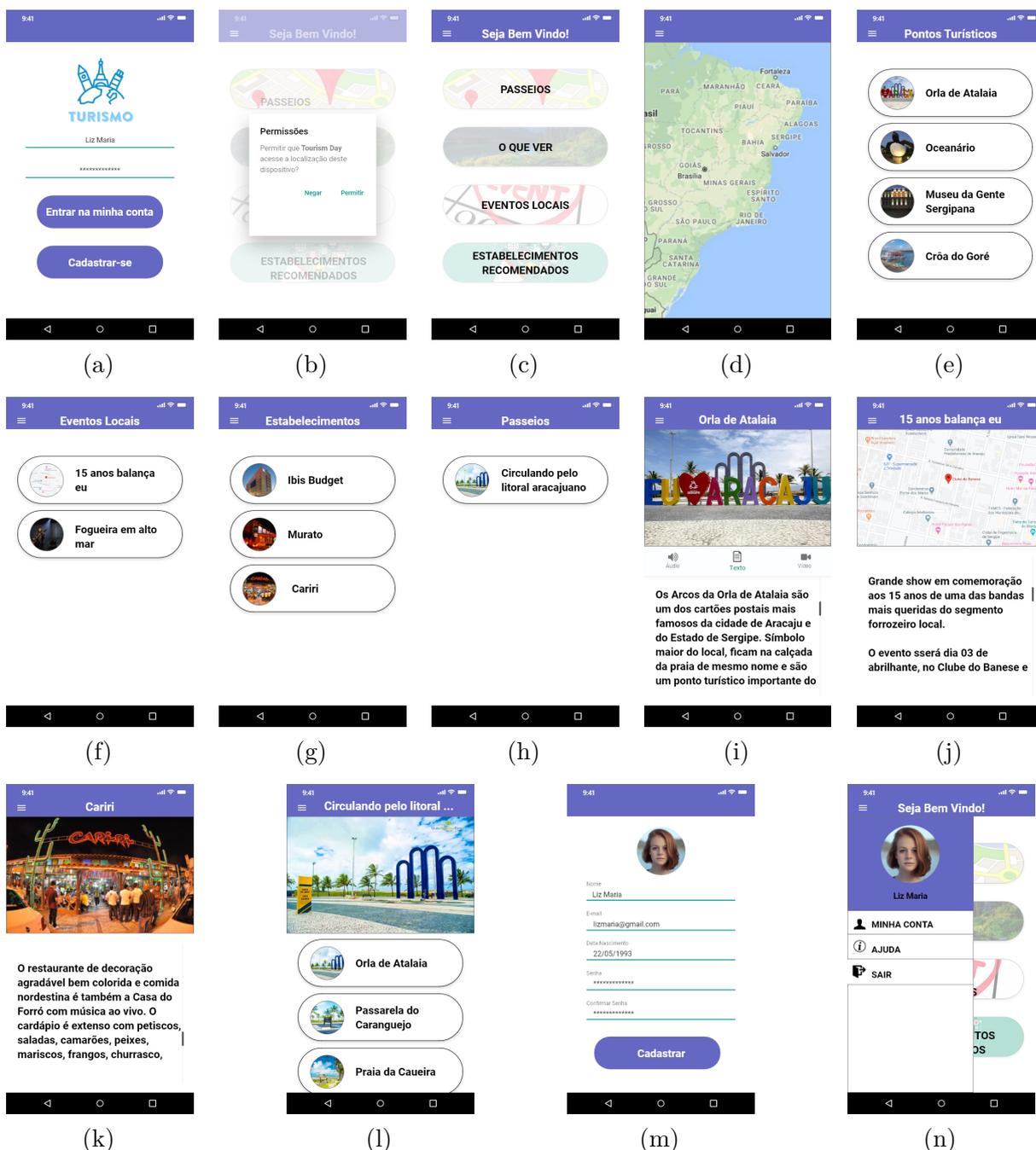
- Tabela **Endereco**: guarda informações de endereços, ela tem um relacionamento de um para muitos com as tabelas Estabelecimento, Evento e PontoTuristico;
- Tabela **Evento**: guarda informações de eventos locais;
- Tabela **Estabelecimento**: guarda informações de estabelecimentos locais, ela tem um relacionamento de muitos para muitos com a tabela Passeio por isso a criação da tabela MapeamentoEstabelecimentoPasseio entre elas;
- Tabela **MapeamentoEstabelecimentoPasseio**: tabela de mapeamento entre as tabelas Estabelecimento e Passeio;
- Tabela **Passeio**: guarda informações de passeios turísticos;
- Tabela **PontoTuristico**: guarda informações de pontos turísticos, ela tem um relacionamento de muitos para muitos com a tabela Paseio por isso a criação da tabela MapeamentoPontoTuristicoPasseio entre elas, a tabela PontoTuristico também tem um relacionamento de um para muitos com a tabela VideoId;
- Tabela **MapeamentoPontoTuristicoPasseio**: tabela de mapeamento entre as tabelas PontoTuristico e Passeio;
- Tabela **VideoId**: guarda informações dos vídeos relacionados aos pontos turísticos;
- Tabela **Produto**: guarda informações dos produtos da linha, ela tem um relacionamento de muitos para muitos com a tabela Feature por isso a criação da tabela MapeamentoProdutoFeature entre elas;
- Tabela **MapeamentoProdutoFeature**: tabela de mapeamento entre as tabelas Produto e Feature;
- Tabela **Feature**: guarda as features presentes no feature model, ela tem um relacionamento de um para um com a tabela TipoVariabilidade;
- Tabela **TipoVariabilidade**: guarda o tipo de variabilidade de cada feature.

As tabelas **Cidade**, **Produto**, **MapeamentoProdutoFeature**, **Feature** e **Ti-povariabilidade** são usadas para fazer a configuração dos possíveis produtos da linha. Basicamente, a cada cidade inserida no banco é criado um produto para essa cidade e cada produto tem suas respectivas *features*. Logicamente, a associação das *features* ao produto deve seguir as regras pertencentes no *feature model* desse trabalho.

3.3 Protótipos

Para validar o modelo e demonstrar algumas variações do produto foram construídos alguns protótipos. Com eles simulamos alguns cenários utilizando informações de algumas cidades de Sergipe e mostramos o relacionamento de cada *feature*. Os protótipos foram criados com a ferramenta Adobe XD².

Figura 7 – Protótipos do produto baseado na cidade de Aracaju



Fonte: Autor.

² <https://www.adobe.com/br/products/xd.html>

Os protótipos da Figura 7 representam o produto com as seguintes *features*: *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 7d), *LISTA_COM_EVENTOS_LOCAIS* (Figura 7f), *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO* (Figura 7g), *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 7h). Por fim, a (Figura 7i) representa a implementação das *features* *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO*, *RESUMO SOBRE O PONTO TURÍSTICO EM ÁUDIO* e *PROCURAR VÍDEOS SOBRE O PONTO*.

Os protótipos da Figura 8 representam o produto com apenas as *features* obrigatórias. Utilizamos a cidade de São Cristóvão para retratar as telas e funcionalidades do aplicativo. Podemos ver que os usuários podem ter: *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 8d), *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 8f). Por fim, a Figura 8g representa a implementação das *features* *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO* e *RESUMO SOBRE O PONTO TURÍSTICO EM ÁUDIO*.

Os protótipos da Figura 9, baseado na cidade de Laranjeiras, representam o produto com as seguintes *features*: *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 9d), *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO* (Figura 9f), *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 9g) e *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO* (Figura 9h).

Tendo validado os possíveis produtos da LPS e suas funcionalidades através dos protótipos, iniciou-se a fase de implementação da LPS, apresentada a seguir.

3.4 Implementação

Os produtos foram desenvolvidos usando o *Framework React Native* em um ambiente cliente-servidor. Para o desenvolvimento do servidor foi utilizado *Node.js* mantendo assim a linguagem *JavaScript* em todo o desenvolvimento do projeto. Esta seção foi dividida em duas partes, uma com a implementação da linha e seus produtos (cliente) e a outra com a implementação da Web API (servidor).

O projeto da API³ e o projeto do aplicativo⁴ estão disponíveis on-line.

3.4.1 Implementação da Linha de Produtos

Para o desenvolvimento dos produtos foi utilizado o *Visual Studio Code*⁵, um editor de código-fonte desenvolvido pela *Microsoft* para *Windows*, *Linux* e *macOS*.

O desenvolvimento seguiu alguns passos: criação do projeto; instalação dos pacotes necessários para a projeto; configuração da arquitetura do projeto; e realizar a codificação do mesmo. A Figura 10 ilustra a organização das pastas e arquivos da LPS. Focaremos mais especificamente na pasta *src*, onde está a implementação do projeto. A Figura 11 apresenta a organização da pasta *src*, ela contém subpastas e arquivos que estão explicados logo abaixo:

- **components**: pasta com os componentes que são reutilizados em vários locais do projeto;
- **configs**: pasta com arquivos de configuração do projeto, por exemplo o arquivo com a conexão com a Web API;
- **screens**: pasta com os arquivos de telas do projeto;
- **store**: pasta onde está a implementação de *Redux*, uma biblioteca feita para gerenciar estados em uma aplicação;
- **commonStyles**: arquivo de estilos compartilhados por todo o projeto;
- **navigator**: arquivo com a configuração das rotas do projeto;
- **utils**: arquivo com funções reutilizadas em vários locais do projeto.

O projeto contou com o uso de alguns recursos do *Google Cloud Plataform*⁶, mais especificamente as API's *Directions API*, *Maps SDK for Android* e *YouTube Data API v3*.

³ <https://github.com/KaicBarros/AppTccWs>

⁴ <https://github.com/KaicBarros/AppTCC>

⁵ <https://code.visualstudio.com/>

⁶ <https://cloud.google.com/>

As API's *Directions API* e *Maps SDK for Android*, foram utilizadas para o uso do mapa e criação da rota do passeio turístico e com a API *YouTube Data API v3* foi possível obter vídeos do *Youtube* relacionados aos pontos turísticos.

Para a funcionalidade com as propagandas foi utilizado o pacote *React Native Firebase*⁷ que se integra à plataforma *Google AdMob*⁸, uma plataforma de publicidade móvel para gerar receita com seu aplicativo.

Na construção dos diferentes produtos, o projeto conta com a capacidade de renderizar apenas as funcionalidades (*features*) pertencentes ao produto. Essas funcionalidades são capturadas do banco de dados de acordo com a cidade informada pelo usuário, assim sendo, a configuração dos produtos está no banco de dados.

Essa capacidade de renderizar apenas algumas funcionalidades é chamada de renderização condicional. Na Lista 3.1 temos o método responsável por verificar se a funcionalidade (*feature*) será renderizada. Ele recebe como parâmetro um *array* de funcionalidades e um *id* da funcionalidade que deverá ou não ser renderizada. Esse método basicamente faz um teste verificando se o *id* da funcionalidade está presente no *array* de funcionalidades, caso esteja, o método retorna verdadeiro e a funcionalidade será renderizada para o usuário, caso contrário o método retorna falso e essa funcionalidade não será renderizada.

```
1 const verificaFeature = (idFeature, features) => {
2   let existe = false
3   for (let index = 0; index < features.length; index++) {
4     if(features[index].IdFeature === idFeature) {
5       existe = true
6     }
7   }
8   return existe
9 }
```

Lista 3.1 – Método responsável por verificar se a funcionalidade será renderizada.

Já na Lista 3.2 temos a chamada do método citado acima na tela que será renderizada a funcionalidade. O parâmetro `SUGERIR_ESTABELECIMENTO_NO_PASSEIO` é uma constante com o *id* da respectiva funcionalidade e o parâmetro *features* é o *array* com as funcionalidades do produto que será renderizado, logo, o componente *Touchableopacity* só será renderizado se o retorno do método for verdadeiro.

```
1 {
2   verificaFeature(SUGERIR_ESTABELECIMENTO_NO_PASSEIO, features) &&
3   <TouchableOpacity onPress={() => abrirEstabelecimentos()}>
4     <Image
```

⁷ <https://rnfb-docs.netlify.app/admob/usage>

⁸ <https://admob.google.com/home/>

```
5         source={imgEstabelecimentos}
6         style={styles.imgBackground}
7     />
8     <Text style={styles.button}>
9         ESTABELECEMENTOS
10    </Text>
11 </TouchableOpacity>
12 }
```

Lista 3.2 – Chamada do método onde a funcionalidade será renderizada.

3.4.2 Implementação da Web API

A Web API ou somente API é responsável por manter os produtos atualizados. Ela realiza a busca das informações no banco de dados e fornece aos produtos. Logo, é nela que encontraremos as consultas SQL executadas no banco de dados.

Para o desenvolvimento da API também foi utilizado o *Visual Studio Code*. A API foi construída em *Node.js*, um software de código aberto, multiplataforma, baseado no interpretador V8 do *Google* e que permite a execução de códigos *JavaScript* fora de um navegador web. Na Figura 12 apresenta a organização dos arquivos e pastas da Web API.

A API utiliza o *Express*⁹ versão 4.17.1, um *framework* para aplicativo da web mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel. Além do *Express* foi preciso usar alguns pacotes, tais como, o *Nodemon*¹⁰ versão 2.0.4 para monitorar os ajustes e restartar o servidor, o *Body Parser*¹¹ versão 1.19.0 para converter os dados da requisição para o formato *json* e o *Knex*¹² versão 0.21.1 para criar consultas SQL.

Na Figura 12 temos a pasta *api*, onde estão os arquivos com os métodos do projeto. Na pasta *config* estão os arquivos de configuração de conexão com o banco de dados. Por fim, na pasta *migrations* temos os arquivos com o código que definem a estrutura do banco de dados.

3.5 A LPS para o Turismo

Depois de apresentado como é feita a renderização dos diferentes produtos da linha, esta seção apresenta alguns dos possíveis produtos. Serão utilizadas imagens capturadas da tela de um dispositivo durante o uso dos produtos da linha.

⁹ <https://www.npmjs.com/package/express>

¹⁰ <https://www.npmjs.com/package/nodemon>

¹¹ <https://www.npmjs.com/package/body-parser>

¹² <https://www.npmjs.com/package/knex>

Como explicado na Seção 3.4.1, após a escolha da cidade de Aracaju as seguintes features são selecionadas:

- *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 13e);
- *LISTA_COM_EVENTOS_LOCAIS* (Figura 13i);
- *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO* (Figura 13e);
- *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 13d);
- *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO* (Figura 13g);
- *RESUMO SOBRE O PONTO TURÍSTICO EM ÁUDIO* (Figura 13g);
- *PROCURAR_VÍDEOS SOBRE O PONTO* (Figura 13h);

Já escolhendo a cidade Itabaiana as seguintes features são selecionadas:

- *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 14e);
- *LISTA_COM_EVENTOS_LOCAIS* (Figura 14i);
- *PROPAGANDA* (Figura 14e);
- *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 14d);
- *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO* (Figura 14g);
- *RESUMO SOBRE O PONTO TURÍSTICO EM ÁUDIO* (Figura 14g);
- *PROCURAR_VÍDEOS SOBRE O PONTO* (Figura 14h);

Após o usuário escolher a cidade Laranjeiras as seguintes features são selecionadas:

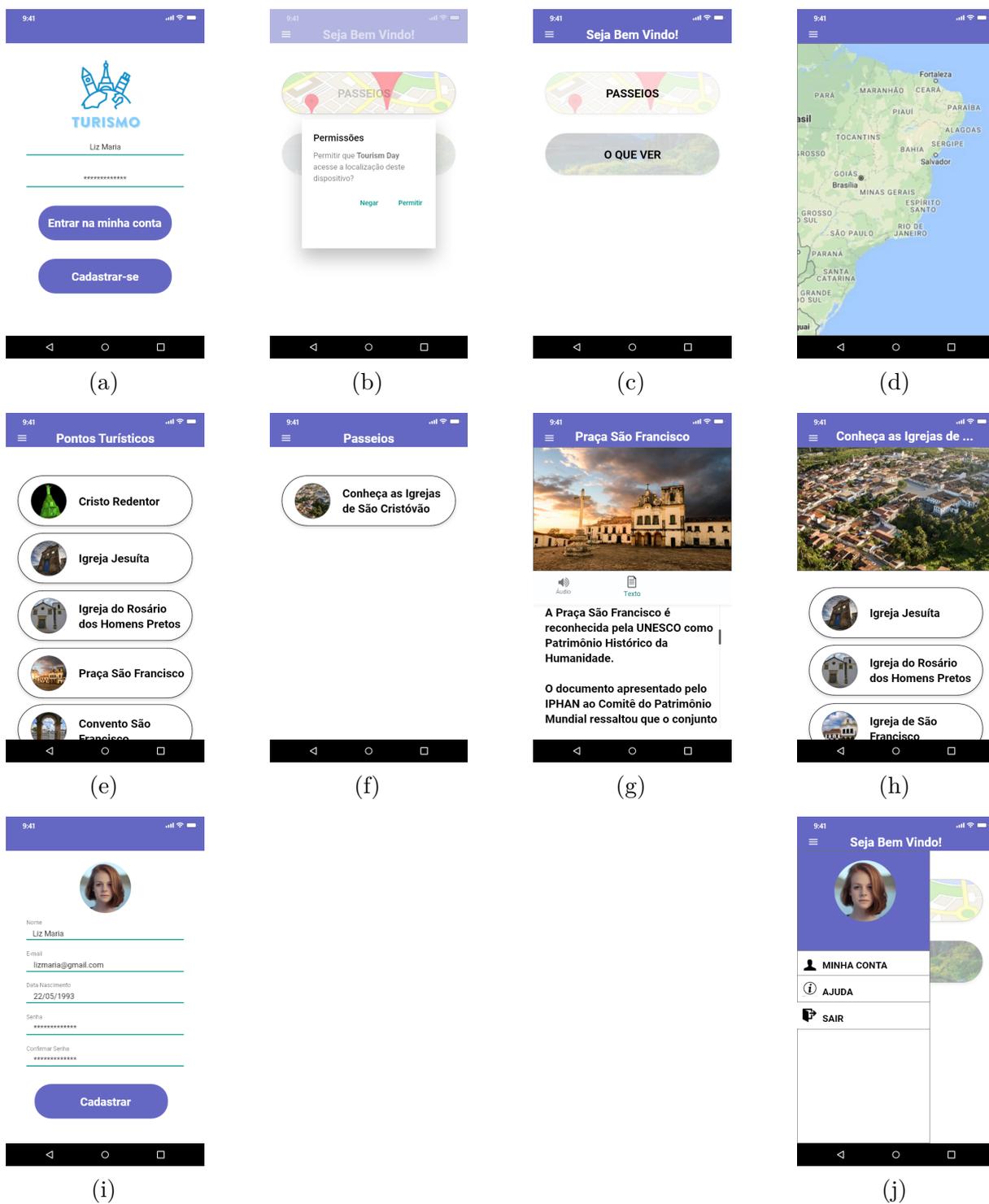
- *MOSTRAR_PONTOS_TURÍSTICOS_NO_MAPA* (Figura 15e);
- *PROPAGANDA* (Figura 15e);
- *SUGERIR_PASSEIOS_POR_PONTOS_TURÍSTICOS* (Figura 15d);
- *RESUMO SOBRE O PONTO TURÍSTICO EM TEXTO* (Figura 15g);
- *RESUMO SOBRE O PONTO TURÍSTICO EM ÁUDIO* (Figura 15g);

Entre as Figuras 13 e 14 podemos notar um caso de restrição do *feature model* (Figura 5) desse trabalho, que é a relação de exclusão entre as *features* *SUGERIR_ESTABELECIMENTOS_NO_PASSEIO* e *PROPAGANDA* (relação explicada na Seção 3.1). Logo, a diferença entre os dois produtos é a existência de uma das *textitfeatures*.

Já na Figura 15 é ilustrado o produto com apenas as *features* obrigatórias do *feature model* (Figura 5). Também podemos notar a existência da *feature* *PROPAGANDA*.

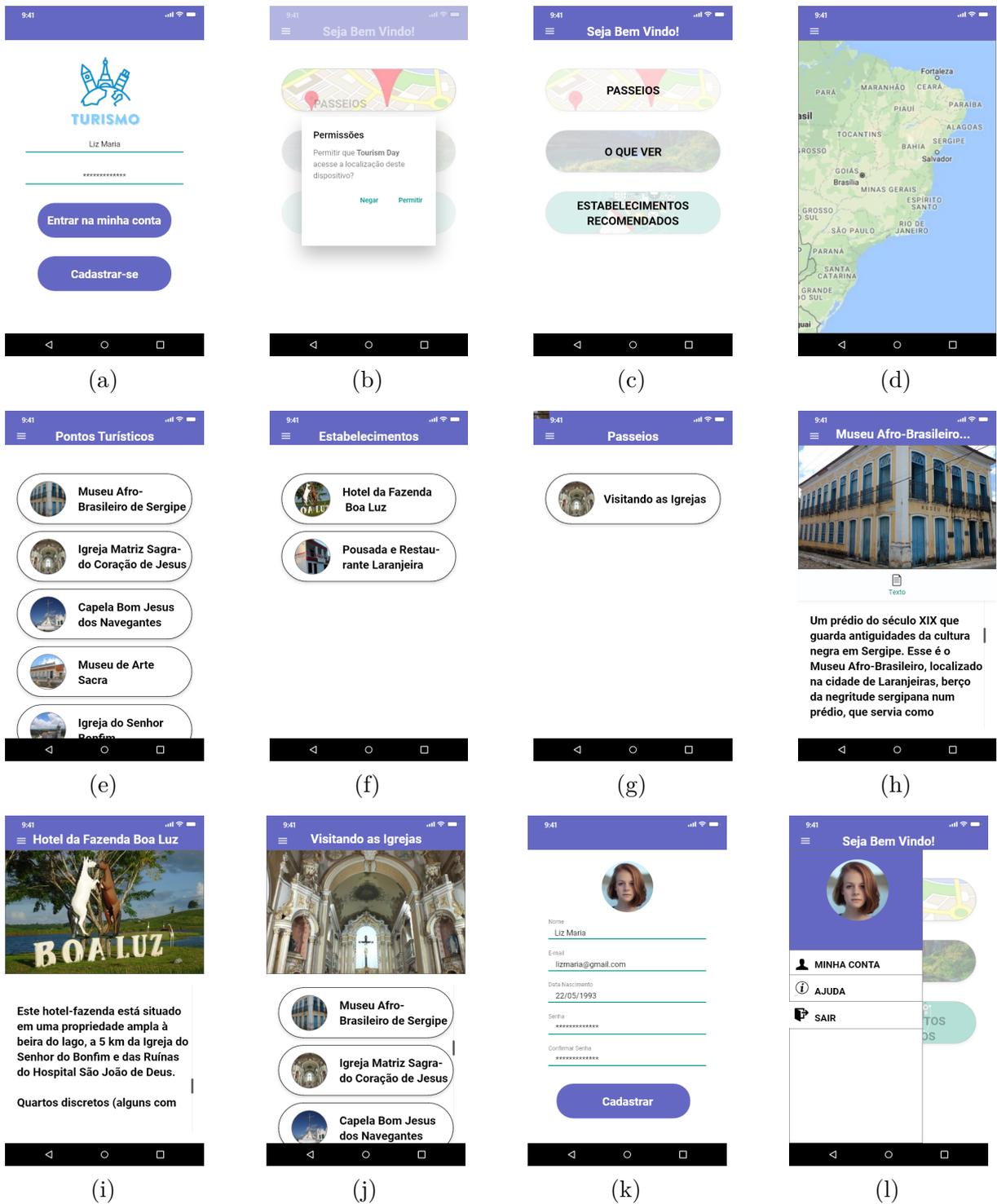
Uma vez implementada a LPS para o turismo em Sergipe, seus devidos produtos foram avaliados empiricamente conforme apresentado no próximo capítulo.

Figura 8 – Protótipos do produto baseado na cidade de São Cristóvão



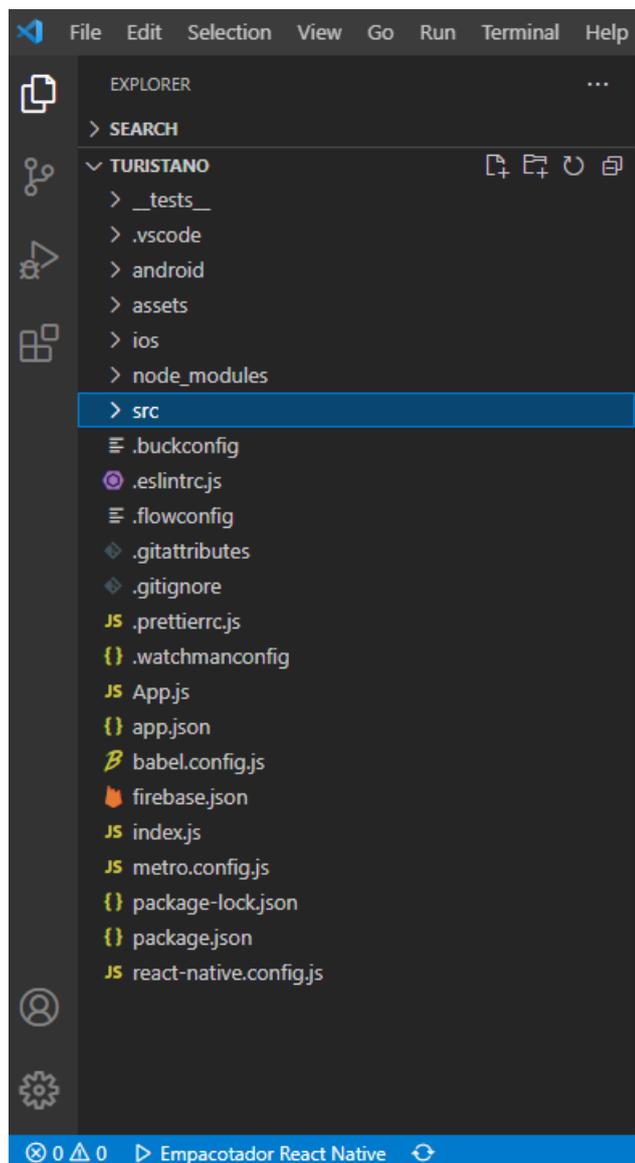
Fonte: Autor.

Figura 9 – Protótipos do produto baseado na cidade de Laranjeiras



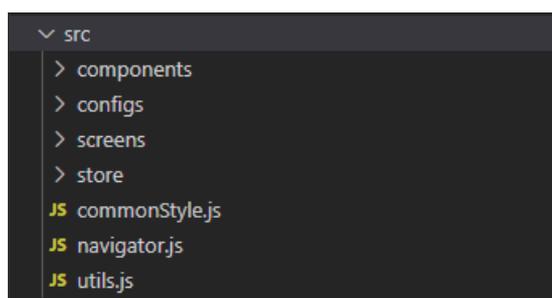
Fonte: Autor.

Figura 10 – Organização das pastas e arquivos da LPS.



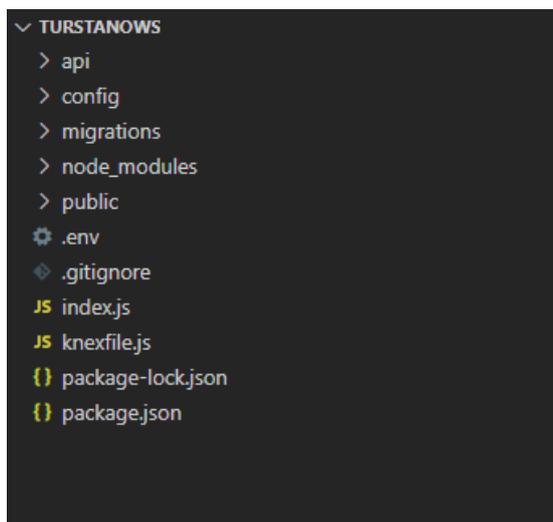
Fonte: Autor.

Figura 11 – Organização da pasta src.



Fonte: Autor.

Figura 12 – Organização das pastas e arquivos da Web API.



Fonte: Autor.

Figura 13 – Produto Completo Patrocinado Aracaju

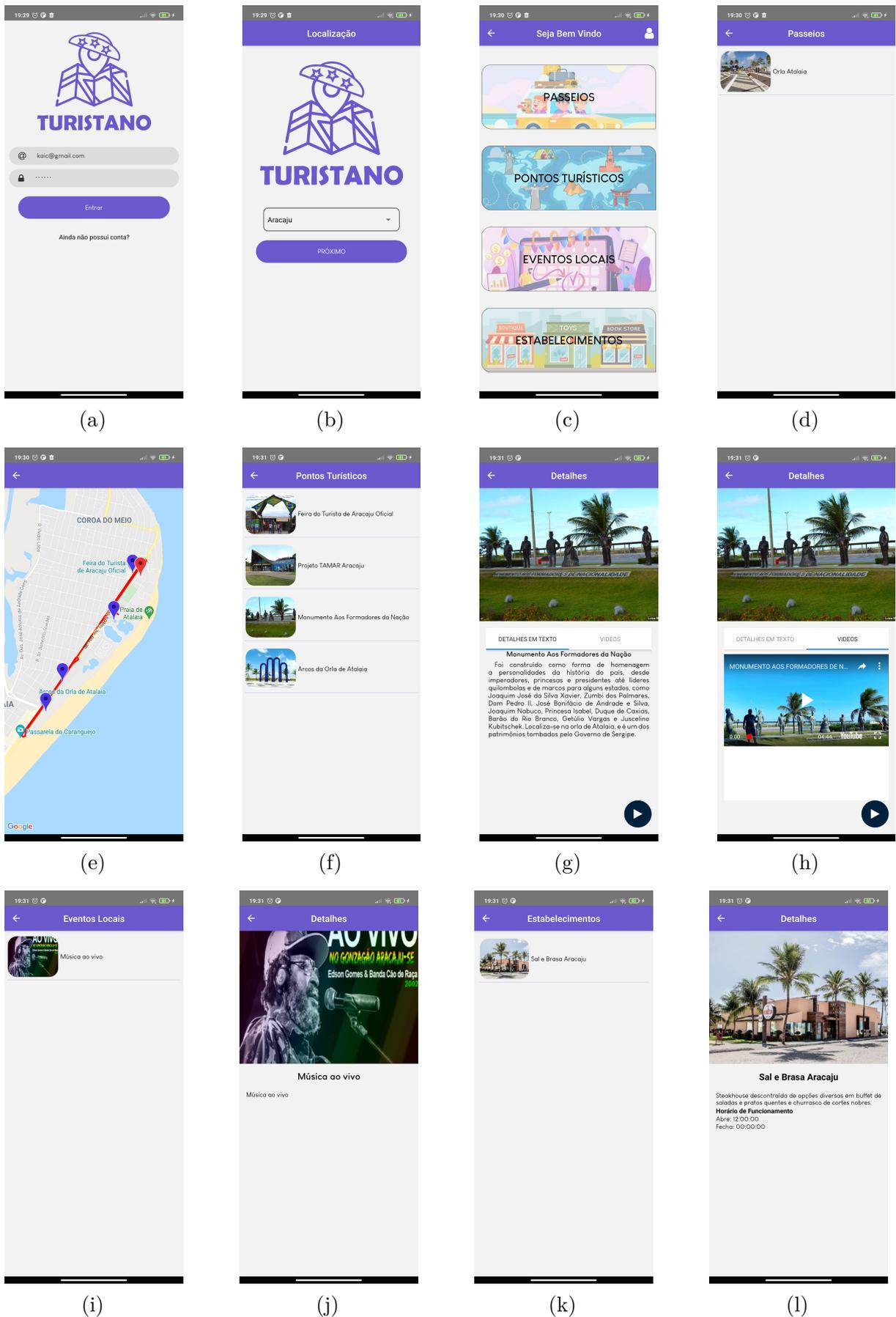
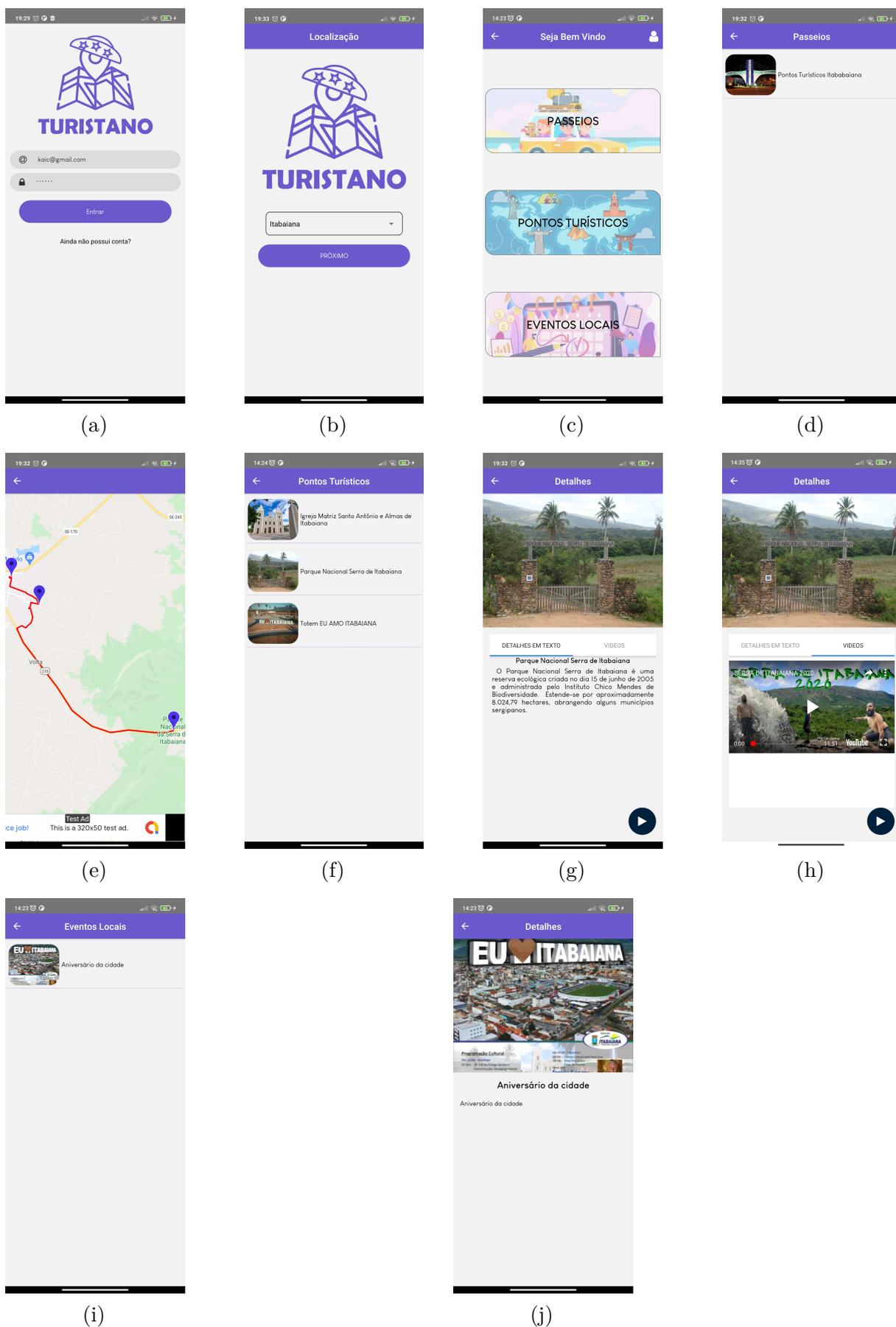
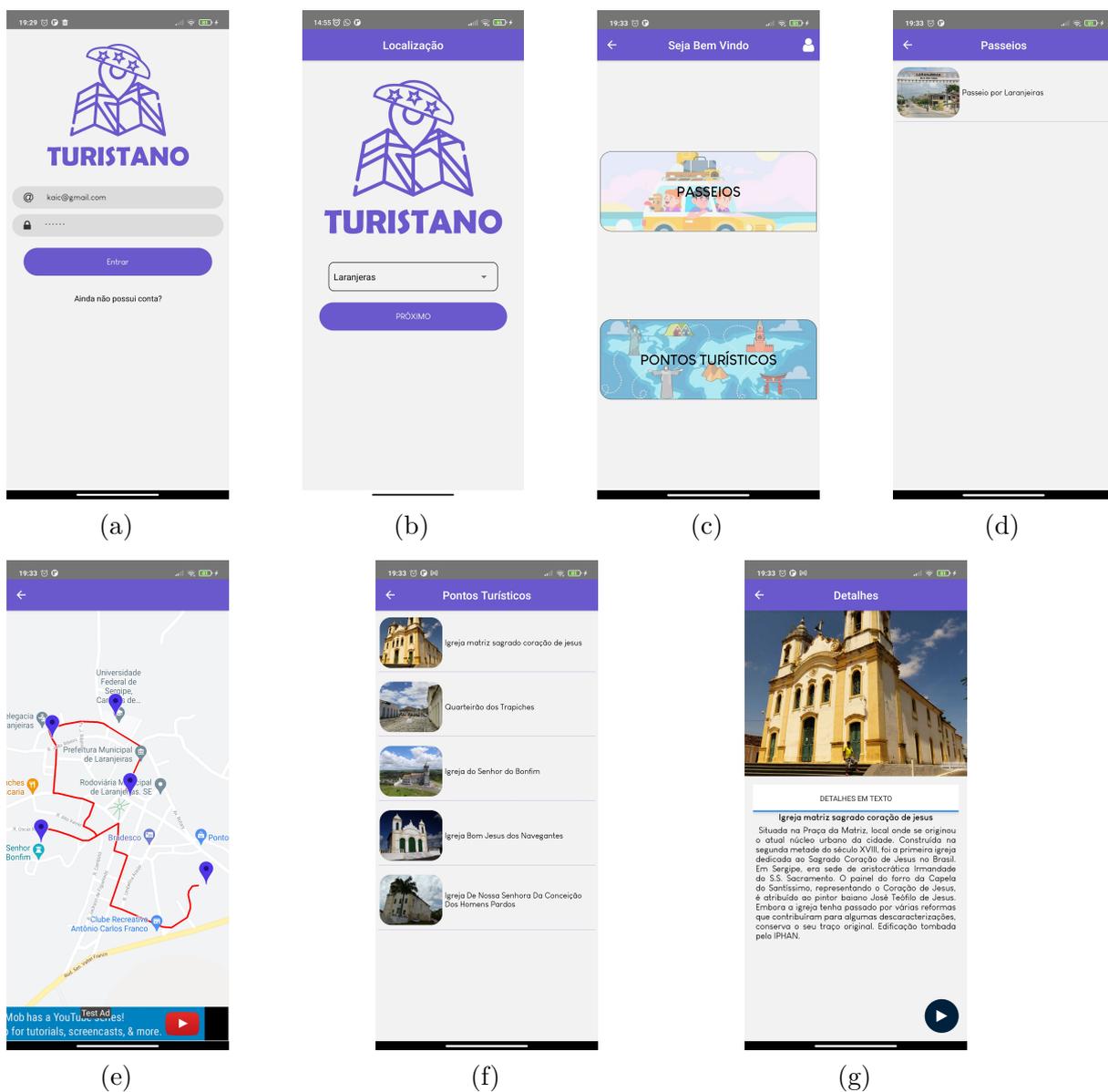


Figura 14 – Produto Completo Publicitário Itabaiana



Fonte: Autor.

Figura 15 – Produto Básico Laranjeiras



Fonte: Autor.

4 Avaliação dos Produtos da LPS

A avaliação dos aplicativos da Linha de Produto de Software (LPS) segue as guias de Wohlin et al. (2012) para reportar experimentos. Destaca-se a seguir o planejamento, a coleta e a análise. Por fim, é feita uma discussão dos resultados desse experimento.

4.1 Planejamento

Para realizar a avaliação, primeiramente foi definido o planejamento do experimento. Para isso, foi utilizado o *Goal, Question, Metrics (GQM - Objetivo, Questão e Métrica)* (BASILI; ROMBACH, 1988). Essa abordagem é definida em três etapas:

1. Objetivo (*Goal*): Responsável pela definição dos objetivos para um objeto (produto, processo ou recurso) ao qual se quer medir;
2. Questão (*Question*): Definição de perguntas referentes ao objetivo, traçando vertentes para alcançar tais objetivos;
3. Métrica (*Metric*): Definição de quais dados serão coletados para responder as questões definidas.

De acordo com o GQM, o objetivo (*Goal*) desse estudo é: **Analisar** o aplicativo desenvolvido para dar suporte ao turismo em Sergipe, **com o propósito de avaliação em relação** a sua facilidade de uso, utilidade e intenção de uso **do ponto de vista** de usuários (leigos, estudantes e profissionais na área de computação) **no contexto** de um trabalho de conclusão de curso.

Definimos as seguintes *Questões de Pesquisa (QP)*:

- QP₀₁: O aplicativo é fácil de ser usado?
- QP₀₂: O aplicativo é útil?
- QP₀₃: Existe intenção de uso do aplicativo futuramente?

Como métrica, definimos três variáveis dependentes subjetivas para a avaliação do aplicativo: facilidade de uso percebida (*Perceived Easy of Use - PEOU*), utilidade percebida (*Perceived Usefulness - PU*) e intenção de uso (*InTention of Use - ITU*). Essas variáveis foram extraídas do modelo *Technology Acceptance Model (TAM)* (DAVIS, 1989).

Foram definidas hipóteses, nulas e alternativas, para para métrica utilizada, apresentadas a seguir:

- H1₀: O aplicativo é percebido como difícil de ser usado.
H1_a: O aplicativo é percebido como fácil de ser usado.
- H2₀: O aplicativo é percebido como não útil.
H2_a: O aplicativo é percebido como útil.
- H3₀: Não existe intenção de uso do aplicativo.
H3_a: Existe intenção de uso do aplicativo.

4.1.1 Teste Piloto

Com o objetivo de refinar e aperfeiçoar os documentos e etapas elaborados para o experimento, antes da execução do experimento foi realizado um teste piloto.

O teste piloto foi realizado com quatro participantes, que são alunos do curso de Sistemas de Informação da Universidade Federal de Sergipe (UFS), cursando à partir do quinto período. Após a aplicação do teste piloto foram identificadas as seguintes melhorias:

- Foi necessário aumentar a memória RAM do servidor em que a Web API estava ativa;
- O Termo de consentimento foi alterado para coletar os dados demográficos dos participantes;
- Na Avaliação do produto uma pergunta sobre a variável PU foi ajustada para deixar mais clara a comparação do aplicativo com outras ferramentas.

Essas melhorias foram analisadas e corrigidas para a coleta de dados do experimento.

4.2 Coleta

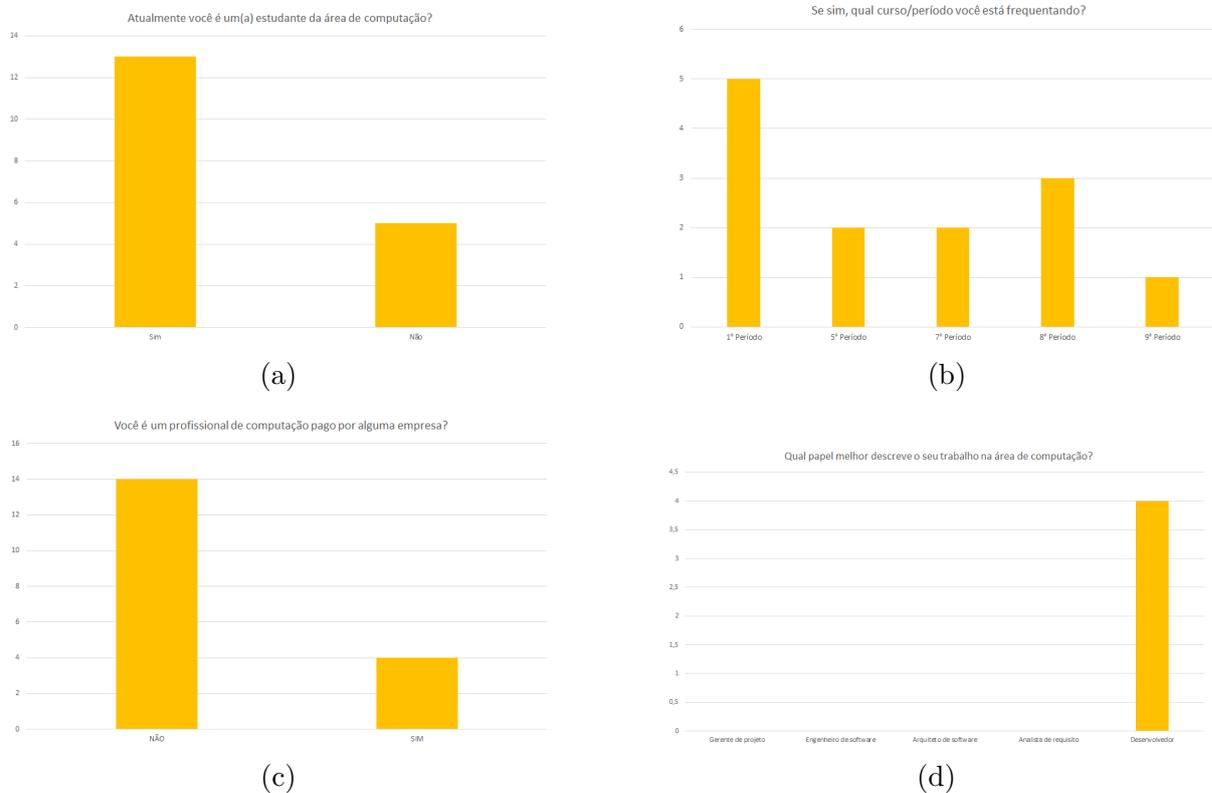
Uma vez executado o teste piloto e corrigidas as devidas melhorias, iniciou-se a coleta de dados para o experimento.

O experimento foi realizado com um grupo de 18 participantes, na maioria, alunos do departamento de Sistemas de Informação da UFS e teve como objetivo coletar a opinião sobre a facilidade de uso, utilidade e sua intenção de usar o aplicativo com os produtos da linha no futuro.

Foram coletados alguns dados demográficos visto na Figura 16. Dentre os participantes, 13 são estudantes da área de computação como pode ser visto na Figura 16a e a maioria está cursando o primeiro período (Figura 16b). Os participantes também foram

questionados sobre a área profissional, onde apenas quatro estão trabalhando no momento na área da computação (Figura 16c) e todos como desenvolvedores (Figura 16d).

Figura 16 – Dados demográficos



Fonte: Autor.

Para a coleta dos dados, utilizou-se o *Google Forms*¹, um aplicativo de gerenciamento de formulários lançado pelo *Google*². Foram definidos quatro formulários apresentados à seguir:

1. **Termo de consentimento:** formulário onde o participante declara-se ciente do experimento, aceita participar e informa alguns dados demográficos. O termo completo encontra-se no Anexo A.
2. **Tarefas (A):** formulário que contém um conjunto de tarefas (A) que o participante deve executar nos produtos da linha. Esse formulário está disponível no Anexo B.
3. **Tarefas (B):** formulário que contém um conjunto de tarefas (B) que o participante deve executar nos produtos da linha. Esse formulário está disponível no Anexo C.
4. **Avaliação dos produtos da linha:** formulário onde o participante responde um conjunto de perguntas fechadas e abertas que permitem expressar sua opinião sobre

¹ <https://www.google.com/intl/pt-BR/forms/about/>

² <https://www.google.com/>

a facilidade de uso, utilidade e sua intenção de usar os produtos no futuro. Esse formulário encontra-se no Anexo D.

O formulário com a avaliação dos produtos da linha segue o modelo TAM. Ele possui 13 questões fechadas (relacionadas à PEOU, PU e ITU) que foram estruturadas de forma a evitar viés no momento da sua aplicação. O questionário foi estruturado com base no modelo da escala Likert de cinco pontos, onde cada afirmativa (positiva ou negativa) está posicionada de um lado da escala. O participante deve responder as afirmações expressando o seu grau de conformismo com cada afirmação. Esse grau é medido de um à cinco, em que, "um" o participante discorda totalmente e "cinco", concorda totalmente com a afirmativa. Para finalizar o participante pode compartilhar sugestões, críticas, elogios e melhorias na última questão aberta do questionário, totalizando assim 14 perguntas. As afirmações (positivas e negativas), e sua relação com as variáveis do TAM, são apresentadas na Tabela 2.

Tabela 2 – Afirmativas positivas e negativas do questionário fechado

Avaliação dos produtos da linha			
Nº	Positivas	Negativas	TAM
1	O aplicativo Turistano é simples e fácil usar.	O aplicativo Turistano é complexo e difícil de usar.	PEOU
2	Em geral, o aplicativo Turistano é fácil de entender.	Em geral, o aplicativo Turistano é difícil de entender.	PEOU
3	O aplicativo Turistano é fácil de aprender.	O aplicativo Turistano é difícil de aprender.	PEOU
4	Acredito que o aplicativo Turistano reduziria o tempo e o esforço necessários para obter informações sobre pontos turísticos e passeios.	Acredito o aplicativo Turistano aumentaria o tempo e o esforço necessários para obter informações sobre pontos turísticos e passeios.	PU
5	Em geral, o aplicativo Turistano é útil.	Em geral, o aplicativo Turistano é inútil.	PU
6	Acredito que o aplicativo Turistano seja útil para a obtenção de informações sobre pontos turísticos e passeios turísticos.	Acredito que o aplicativo Turistano é inútil para a obtenção de informações sobre pontos turísticos e passeios turísticos.	PU
7	Acredito que o aplicativo Turistano incorpora os mecanismos necessários para obter informações sobre pontos turísticos e passeios.	Acredito que o aplicativo Turistano carece dos mecanismos necessários para obter informações sobre pontos turísticos e passeios.	PU

Table 2 continuação da tabela anterior

8	Em geral, acredito que o aplicativo Turistano apóia de forma eficiente a obtenção de informações sobre pontos turísticos e passeios.	Em geral, acredito que o aplicativo Turistano não apóia de forma eficiente a obtenção de informações sobre pontos turísticos e passeios.	PU
9	O uso do aplicativo Turistano melhorará meu conhecimento sobre pontos turísticos e passeios.	O uso do aplicativo Turistano não vai melhorar meu conhecimento sobre pontos turísticos e passeios.	PU
10	Se eu precisar usar o aplicativo Turistano no futuro, acredito que vou considerar seu uso.	Se eu precisar usar o aplicativo Turistano no futuro, acredito que não irei considerar seu uso.	ITU
11	Acredito que seria fácil se tornar hábil no uso do aplicativo Turistano.	Acredito que seria difícil se tornar hábil no uso do aplicativo Turistano.	ITU
12	Pretendo usar o aplicativo Turistano no futuro.	Não tenho intenção de usar o aplicativo Turistano no futuro	ITU
13	Eu recomendaria usar o aplicativo Turistano.	Eu não recomendaria usar o aplicativo Turistano.	ITU

Foi realizada uma análise quantitativa e uma discussão qualitativa dos dados, a fim de verificar os resultados da percepção de facilidade de uso (PEOU), utilidade percebida (PU), intenção de uso (ITU) dos participantes, apresentados a seguir.

4.3 Análise

Nesta Seção, analisaremos quantitativamente os dados de acordo com as hipóteses estabelecidas. As variáveis subjetivas (PEOU, PU e ITU) foram analisadas comparando a mediana das respostas das questões com o valor neutro da escala Likert de 5 pontos, onde o valor neutro é igual a 3. Utilizou-se o RStudio para realizar as análises estatísticas e gerar os gráficos do experimento (WALTER et al., 2012).

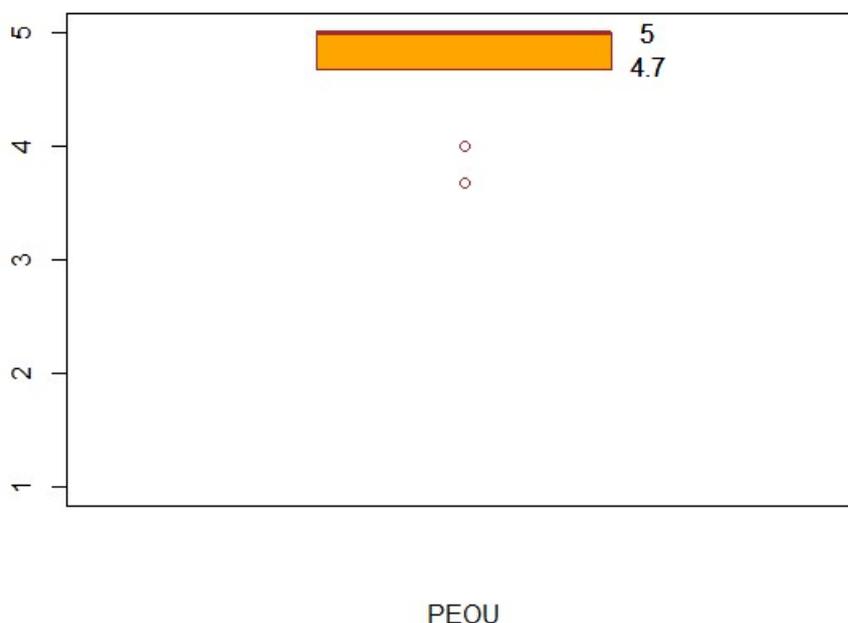
A Tabela 3 apresenta os resultados dos testes aplicados para cada variável de pesquisa. Os resultados do teste de normalidade mostram que todas as variáveis não estão normalmente distribuídas, uma vez que os resultados do teste *Shapiro-Wilk* são menores do que 0.05 ($p\text{-value} < 0.05$). Assim, foi aplicado o teste *One-Sample Wilcoxon Signed Rank*, com um valor de mediana 3 (uma amostra de teste não-paramétrico).

A seguir, apresenta-se a análise estatística e a análise dos boxplots para cada QP desse estudo.

Tabela 3 – Análise das variáveis PEOU, PU e ITU

Variáveis subjetivas dependentes	Média	Mediana	Desvio padrão	Shapiro-Wilk	Wilcoxon test
PEOU	4.76	5.00	0.4222	0.00008	0.0007
PU	4.45	4.42	0.4910	0.01867	0.0010
ITU	4.46	4.75	0.4953	0.01476	0.0010

Figura 17 – Boxplot da Percepção da Facilidade de Uso



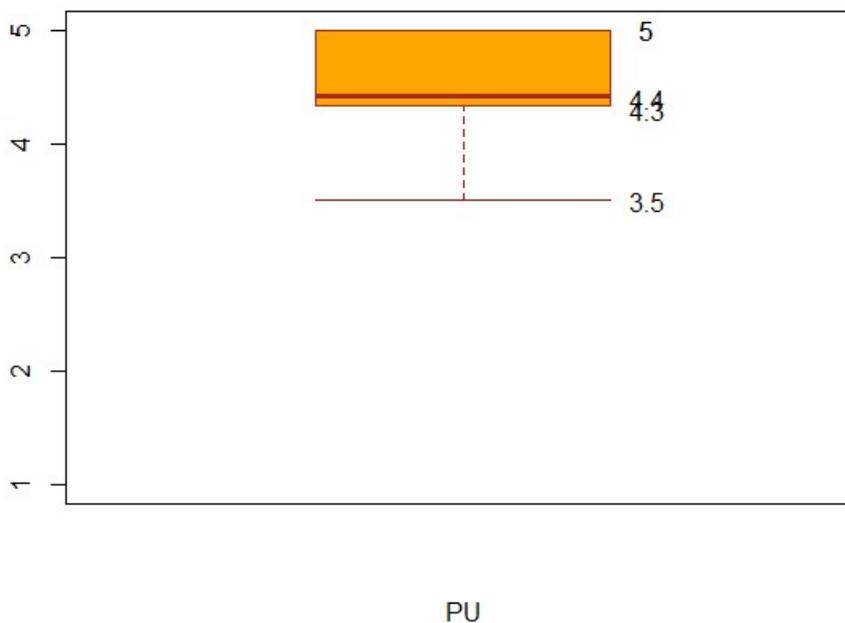
PEOU
Fonte: Autor.

4.3.1 QP01: O aplicativo é fácil de ser usado?

Para responder a essa questão, coletou-se as respostas dos participantes em relação ao grau de satisfação dos possíveis produtos da linha, afim de mensurar sua Facilidade de Uso Percebida (PEOU). O teste de normalidade *Shapiro-Wilk* retornou um *p-value* de 0.00008, conforme apresentado na Tabela 3. Sendo não normal a distribuição dos dados, foi utilizado o teste *One-Sample Wilcoxon Signed Rank*, cujo *p-value* foi 0.0007. Como o valor é menor que 0.05 (5%), foi rejeitada a hipótese nula H_{10} (Os produtos da linha são percebidos como difíceis de serem usados), ou seja, os participantes concordaram que o aplicativo é percebido como fácil de ser usado.

De acordo com *boxplot* da variável PEOU (Figura 17), a mediana (linha mais escura da caixa) tem o valor de 5. As linhas abaixo e a cima da mediana representam o quartil Q1 e quartil Q3, gerando o comprimento da caixa. O valor de Q1 é 4.7 e de Q3 é 5. O ponto mínimo é corresponde ao valor 3.8 e o máximo corresponde ao valor 5. Há a

Figura 18 – Boxplot da Percepção de Utilidade



Fonte: Autor.

presença de dois *outliers* localizados próximo do valor 4 da caixa, os dois são alunos do primeiro período do curso de Sistemas de Informação da Universidade Federal de Sergipe, não possuem experiência profissional remunerada em computação e um deles errou itens dos formulários Tarefas A (Anexo B) e Tarefas B (Anexo C). Como a mediana é igual ao valor de Q3, os dados coletados são negativamente assimétricos. Observa-se, pelo gráfico de caixa, que todos os respondentes pontuaram seus valores acima do valor neutro, que é três (3). A Tabela 4 apresenta os dados coletados para esse *boxplot*.

Tabela 4 – Resumo dos dados coletados da variável subjetiva PEOU

Mínimo	1º Quartil	Mediana	Média	3º Quartil	Máximo
3.8	4.7	5.00	4.76	5.0	5.0

4.3.2 QP02: O aplicativo é útil?

Para responder a essa questão, coletamos as respostas dos participantes em relação ao grau de satisfação dos possíveis produtos da linha, afim de mensurar sua utilidade percebida (PU). O teste de normalidade *Shapiro-Wilk* retornou um *p-value* de 0.01867, conforme apresentado na Tabela 3. Sendo não normal a distribuição dos dados, foi utilizado o teste *One-Sample Wilcoxon Signed Rank*, cujo *p-value* foi 0.0010. Como o valor é menor que 0.05 (5%), foi rejeitada a hipótese nula H_{20} (os produtos da linha são percebidos como não úteis), ou seja, os participantes concordaram que o aplicativo é percebido como útil.

De acordo com *boxplot* da variável PU (Figura 18), a mediana (linha mais escura da caixa) tem o valor de 4.4. As linhas abaixo e a cima da mediana representam o quartil Q1 e quartil Q3, gerando o comprimento da caixa. O valor de Q1 é 4,3 e de Q3 é 5. O ponto mínimo é corresponde ao valor 3,5 e o máximo corresponde ao valor 5. Como a mediana está mais próxima de Q1, os dados coletados são positivamente assimétricos. Observa-se, pelo gráfico de caixa, que todos os respondentes pontuaram seus valores bem acima do valor neutro, que é três (3). A Tabela 5 apresenta os dados coletados para esse *boxplot*.

Tabela 5 – Resumo dos dados coletados da variável subjetiva PU

Mínimo	1º Quartil	Mediana	Média	3º Quartil	Máximo
3.5	4.3	4.42	4.45	5.0	5.0

4.3.3 QP03: Existe intenção de uso do aplicativo futuramente?

Para responder a essa questão, coletamos as respostas dos participantes em relação ao grau de satisfação dos possíveis produtos da linha, com o objetivo de mensurar sua intenção de uso (ITU). O teste de normalidade *Shapiro-Wilk* retornou um *p-value* de 0.01476, conforme apresentado na Tabela 3. Sendo não normal a distribuição dos dados, foi utilizado o teste *One-Sample Wilcoxon Signed Rank*, cujo *p-value* foi 0.0010. Como o valor é menor que 0.05 (5%), foi rejeitada a hipótese nula H_{30} (não existe intenção de uso dos produtos da linha.), ou seja, os participantes concordam que existe intenção de uso do aplicativo.

De acordo com *boxplot* da variável ITU (Figura 19), a mediana (linha mais escura da caixa) tem o valor de 4.8. As linhas abaixo e a cima da mediana representam o quartil Q1 e quartil Q3, gerando o comprimento da caixa. O valor de Q1 é 4 e de Q3 é 5. O ponto mínimo é corresponde ao valor 3.2 e o máximo corresponde ao valor 5. Como a mediana está mais próxima de Q3, os dados coletados são negativamente assimétricos. Observa-se, pelo *boxplot*, que todos os respondentes pontuaram seus valores bem acima do valor neutro, que é três (3). A Tabela 6 apresenta os dados coletados para esse *boxplot*.

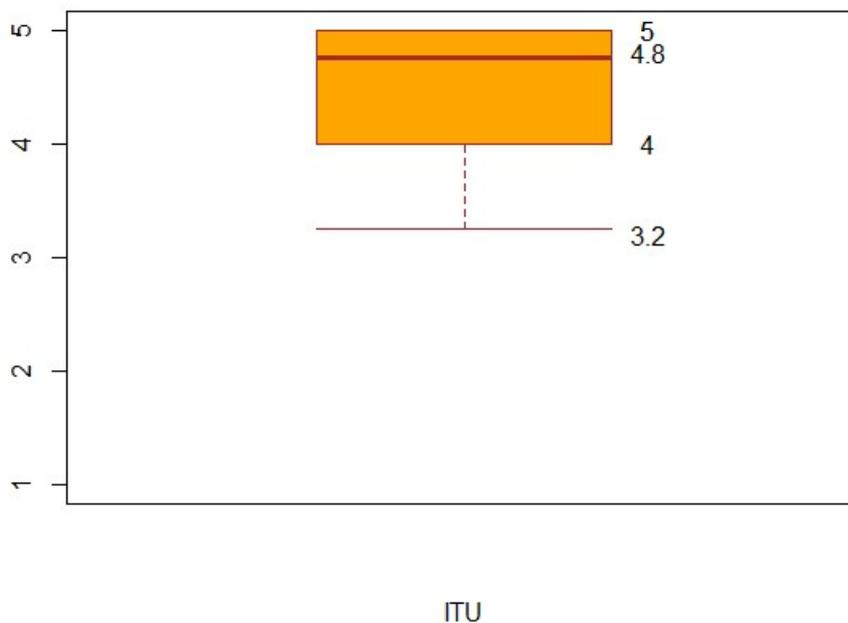
Tabela 6 – Resumo dos dados coletados da variável subjetiva ITU

Mínimo	1º Quartil	Mediana	Média	3º Quartil	Máximo
3.2	4	4.8	4.46	5.0	5.0

4.3.4 Teste Cliff's Delta (Effect Size)

Com o intuito de reforçar os resultados obtidos através dos testes estatísticos aplicados às variáveis PEOU, PU e ITU, foi aplicado um teste de *Effect Size* denominado *Cliff's Delta* (GIBBONS; CHAKRABORTI, 2020). O teste *Cliff's Delta* pode ser definido

Figura 19 – Boxplot da Intenção de Uso



Fonte: Autor.

como a diferença entre a probabilidade de uma observação aleatória de um grupo 01 ser maior que uma observação aleatória de um grupo 02, ou, a probabilidade de uma observação aleatória de um grupo 01 ser menor que uma observação aleatória de um grupo 02. O teste *Cliff's Delta* é limitado, significando que um *effect size* de 1 ou -1 indica a ausência de sobreposição entre os grupos, e, 0 indica que as distribuições dos grupos são equivalentes. Ainda, o sinal indica a direção do efeito. Um sinal positivo revela uma direção favorável às variáveis PEOU, PU, ITU analisadas nesse experimento. No experimento realizado, verificamos o resultado de cada variável com a média 3, obtendo o valor *effect size* large em todas as variáveis. Na tabela 7 é possível verificar os dados obtidos no teste.

Tabela 7 – Análise valores do Cliff's Delta das variáveis PEOU, PU e ITU

Variáveis subjetivas dependentes	<i>Cliff's Delta</i>	<i>Effect Size</i>
PEOU	0.68 [-0.08, 0.94]	Large
PU	0.60 [-0.25, 0.92]	Large
ITU	0.55 [-0.23, 0.90]	Large

Como consequência, foram rejeitadas todas as hipóteses nulas; aceitando que o aplicativo é percebido como fácil de usar, útil e existe uma boa intenção de usá-lo no futuro.

4.4 Discussão

Os resultados encontrados na avaliação do aplicativo mostraram que os entrevistados perceberam o aplicativo como fácil de ser usado, útil e que possui a intenção de utilizá-lo futuramente. Esses dados encontrados corroboram com o desenvolvimento do aplicativo que desde do seu início levou em consideração como os usuários iriam interagir com o mesmo. Apesar do resultado positivo para todas as variáveis analisadas (PEOU, PU e ITU), futuras avaliações poderão ser realizadas para manter o aplicativo em perfeito funcionamento.

Através do último formulário preenchido pelos participantes, os mesmo podiam deixar sugestões, críticas, elogios e melhorias na avaliação do aplicativo (Anexo D). A seguir serão apresentados alguns comentários dos participantes.

- P₀₁: *Eu achei desnecessário o usuário precisar entrar com e-mail e senha, poderia facilitar o uso do usuário acessar o aplicativo sem e-mail e senha e posteriormente fazer a autenticação, por exemplo;*
- P₀₂: *Um bom aplicativo, bastante informativo e fácil de compreender no geral. Além de mostrar localidades turísticas ainda conta sobre suas histórias e ainda mostra trajetórias para aquele determinado ponto. Talvez seria interessante colocar esta função do guia do mapa a partir de onde a pessoa está localizada para se guiar melhor, mas é só uma ideia não muito bem pensada e não sei se daria certo;*
- P₀₃: *O aplicativo tem uma ideia muito boa, e é de simples uso, se receber mais atualizações, com mais locais, ficaria ainda melhor;*
- P₀₄: *O aplicativo é de fácil utilização, além de ser uma ferramenta valiosa para quem está fazendo algum passeio turístico ou na percepção de pontos de importantes em uma simples passagem, recomendo o uso;*
- P₀₅: *O Aplicativo é muito pratico. poderia futuramente incluir uma parte sobre culinária da região;*
- P₀₆: *Design simples e bonito, possui uma boa usabilidade;*
- P₀₇: *Acredito que as principais melhorias que poderia realizar seria na interface, está muito simples parecendo um app dos anos 2000, poderia utilizar cards ao invés de lista, ter uns destaques na home.*

Os comentários são muito pertinentes já que tratam de melhorias de uso e novas funcionalidades.

A sugestão P₀₁, P₀₂, P₀₃ e P₀₅ tratam-se de funcionalidades que irão demandar muito tempo para o desenvolvimento, já que terão que serem feitas alterações no Feature Model, banco de dados, produtos e Web API, por isso ficou decidido que as sugestões serão analisadas melhor futuramente.

Também foram deixados comentários positivos como podemos ver em P₀₄ e P₀₆ e negativos (comentário P₀₇) que serão levados em consideração para possíveis melhorias.

5 Considerações Finais

O mercado de aplicativos *mobile* vem se tornando um ótimo campo para os desenvolvedores. Com a grande popularização de *smartphones* é muito fácil ter informação em qualquer lugar com o auxílio da internet.

Mas, com o grande número de usuários, também teremos um grande número de necessidades diferentes. Com o objetivo de atender uma boa parte destas necessidades verificamos que a técnica de desenvolvimento focada no reuso de funcionalidades, mais especificamente utilizando Linhas de Produto de Software, pode ser útil para atender essas diferentes necessidades.

Atualmente não foi identificado nenhum aplicativo *mobile* que traga informações sobre pontos turísticos e também sugira alguns passeios turísticos em Sergipe. Assim, o presente trabalho teve como objetivo construir uma LPS para o turismo em Sergipe, validar a mesma construindo os produtos com as funcionalidades da LPS e realizar um estudo avaliando os produtos da linha. Vale ressaltar que os produtos construídos pela LPS para o turismo em Sergipe foram considerados fáceis de se utilizar, úteis e possuem a intenção de serem utilizados por seus usuários no futuro.

Como trabalho futuro tem-se a inclusão das funcionalidades sugeridas pelos participantes durante a avaliação da linha de produtos (por exemplo, as sugestões dos participantes P₀₂ e P₀₅). Ainda, como futura contribuição deste trabalho, poderá ser desenvolvida uma aplicação para a realização da configuração dos produtos. Essa aplicação ficará responsável por realizar o cadastro de cidades, pontos turísticos e produtos por exemplo.

Referências

- APEL, S. et al. *Feature-oriented software product lines*. [S.l.]: Springer, 2016. Citado 3 vezes nas páginas 16, 17 e 18.
- APEL, S. et al. An algebraic foundation for automatic feature-based program synthesis. *Science of Computer Programming*, Elsevier, v. 75, n. 11, p. 1022–1047, 2010. Citado na página 17.
- BASILI, V. R.; ROMBACH, H. D. The tame project: Towards improvement-oriented software environments. *IEEE Transactions on software engineering*, IEEE, v. 14, n. 6, p. 758–773, 1988. Citado na página 42.
- BATORY, D. Feature models, grammars, and propositional formulas. In: SPRINGER. *International Conference on Software Product Lines*. [S.l.], 2005. p. 7–20. Citado na página 17.
- BATORY, D.; SARVELA, J. N.; RAUSCHMAYER, A. Scaling step-wise refinement. *IEEE Transactions on Software Engineering*, IEEE, v. 30, n. 6, p. 355–371, 2004. Citado na página 17.
- BOSCH, J. *Design and use of software architectures: adopting and evolving a product-line approach*. [S.l.]: Pearson Education, 2000. Citado na página 17.
- CHEN, K. et al. An approach to constructing feature models based on requirements clustering. In: IEEE. *13th IEEE International Conference on Requirements Engineering (RE'05)*. [S.l.], 2005. p. 31–40. Citado na página 17.
- CLASSEN, A.; HEYMANS, P.; SCHOBGENS, P.-Y. What's in a feature: A requirements engineering perspective. In: SPRINGER. *International Conference on Fundamental Approaches to Software Engineering*. [S.l.], 2008. p. 16–30. Citado na página 17.
- CLEMENTS, P.; NORTHROP, L. M. *Software product lines - practices and patterns*. [S.l.]: Addison-Wesley, 2002. (SEI series in software engineering). ISBN 978-0-201-70332-0. Citado na página 13.
- DARAMOLA, J.; ADIGUN, M. O.; OLUGBARA, O. O. A product line architecture for evolving intelligent component services in tourism information systems. In: ENTER. [S.l.: s.n.], 2008. p. 117–129. Citado 2 vezes nas páginas 21 e 22.
- DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, JSTOR, p. 319–340, 1989. Citado na página 42.
- GIBBONS, J. D.; CHAKRABORTI, S. *Nonparametric statistical inference*. [S.l.]: CRC press, 2020. Citado na página 49.
- KANG, K. C. et al. *Feature-oriented domain analysis (FODA) feasibility study*. [S.l.], 1990. Citado na página 17.

- KANG, K. C. et al. Form: A feature-; oriented reuse method with domain-; specific reference architectures. *Annals of software engineering*, Springer, v. 5, n. 1, p. 143, 1998. Citado na página 17.
- KRZYSZTOF, C.; EISENECKER, U. W. *Generative Programming: Methods, Tools and Applications*. [S.l.]: Addison-Wesley, 2000. Citado na página 17.
- NASCIMENTO, F. P.; SOUSA, F. L. L. Metodologia da pesquisa científica. teoria e prática. In: THESAURUS. *Metodologia da Pesquisa Científica. Teoria e Prática*. [S.l.], 2015. p. 384. Citado na página 14.
- PARK, B.; YIM, J. Design of an android app for gyeongju smart tourism. In: *Computer science and its applications*. [S.l.]: Springer, 2015. p. 991–995. Citado 2 vezes nas páginas 21 e 22.
- PASHTAN, A.; HEUSSER, A.; SCHEUERMANN, P. Personal service areas for mobile web applications. *IEEE Internet computing*, IEEE, v. 8, n. 6, p. 34–39, 2004. Citado na página 22.
- POHL, K.; BÖCKLE, G.; LINDEN, F. J. van D. *Software product line engineering: foundations, principles and techniques*. [S.l.]: Springer Science & Business Media, 2005. Citado na página 13.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 13.
- WALTER, O. et al. Uma visão geral do rstudio aplicado ao ensino de controle estatístico do processo. In: *XL CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA*. [S.l.: s.n.], 2012. Citado na página 46.
- WOHLIN, C. et al. *Experimentation in software engineering*. [S.l.]: Springer Science & Business Media, 2012. Citado na página 42.
- ZAVE, P. An experiment in feature engineering. In: *Programming methodology*. [S.l.]: Springer, 2003. p. 353–377. Citado na página 17.

Anexos

ANEXO A – Termo de Consentimento

Prezado(a) participante. O objetivo deste experimento é coletar a sua opinião sobre a facilidade de uso, utilidade e sua intenção de usar o aplicativo Turistano no futuro. Sua participação é voluntária e confidencial. Você pode desistir a qualquer momento. Esta pesquisa faz parte do Trabalho de Conclusão de Curso (TCC) de Kaic Oliveira Barros, do Departamento de Sistemas de Informação (DSI) da Universidade Federal de Sergipe (UFS). Caso esteja interessado em ser informado sobre o resultado deste estudo ou de qualquer publicação resultante, você terá a oportunidade de indicar isso e nos fornecer seu endereço de e-mail. Entre em contato com Raphael Oliveira (orientador do TCC) através do e-mail raphael.oliveira@academico.ufs.br, caso tenha dúvidas ou comentários relacionados ao estudo. Agradecemos desde já!

E-mail: _____

CONSENTIMENTO ELETRÔNICO. Selecione suas opções abaixo. Selecionando a opção "Sim" indica que: i) você leu e entendeu as informações acima, ii) você voluntariamente concorda em participar, e iii) você tem pelo menos 18 anos de idade. Se você não deseja participar desse experimento, recuse a participação selecionando "Não".

- Sim
- Não

Dados demográficos

Você é um profissional de computação pago por alguma empresa?

- Sim
- Não

Dados demográficos profissionais

Qual papel melhor descreve o seu trabalho na área de computação?

- Engenheiro de software
- Gerente de projeto
- Arquiteto de software
- Analista de requisito

Gerente de projeto

Outros

Quantos anos de experiência profissional você tem (Ex.: 0, 1, 2, 3, ..., 10, ...)?

Resposta: _____

Dados demográficos educacionais

Atualmente você é um(a) estudante da área de computação? Se sim, qual curso/-
período você está frequentando?

Resposta: _____

ANEXO B – Tarefas (A)

A seguir serão apresentadas as tarefas a serem realizadas no aplicativo Turistano.

E-mail: _____

Tarefa 1 - Criar conta (caso não tenha) e realizar o login. Após o login, o que você vê?

- () Uma tela com o título “Seja Bem Vindo” e alguns botões.
- () Uma tela com o título Localização onde posso escolher uma cidade e clicar em próximo.
- () Uma tela para informar o e-mail e senha.
- () Uma tela com informações do meu perfil e o botão sair.

Tarefa 2 - Após realizar os passos da Tarefa 1, escolha a cidade “ARACAJU” e clique no botão “PASSEIOS”. Depois de clicar no botão, o que você vê?

- () Uma tela com uma lista de passeios.
- () Uma tela com o título “Seja Bem Vindo” e alguns botões.
- () Uma tela com uma lista de eventos.
- () Uma tela com detalhes de um estabelecimento.

Tarefa 3 - Após realizar os passos da Tarefa 2, clique no item “Orla Atalaia”. Depois de clicar no item, o que você vê?

- () Uma tela com informações do meu perfil e o botão sair.
- () Uma tela com o título “Seja Bem Vindo” e alguns botões.
- () Uma tela com uma rota e alguns pontos marcados no mapa.
- () Uma tela para informar o e-mail e senha.

Tarefa 4 - Agora volte até a tela com o título "Seja Bem Vindo" e clique no botão "EVENTOS". Depois de clicar no botão, o que você vê?

- () Uma tela com o título “Detalhes” e algumas informações sobre um ponto turístico.
- () Uma tela com informações do meu perfil.

- () Uma tela com uma lista de eventos.
- () Uma tela com detalhes de um evento.

ANEXO C – Tarefas (B)

A seguir serão apresentadas as tarefas a serem realizadas no aplicativo Turistano.

E-mail: _____

Tarefa 1 - Criar conta (caso não tenha) e realizar o login. Após o login, o que você vê?

- () Uma tela com o título “Seja Bem Vindo” e alguns botões.
- () Uma tela com o título Localização onde posso escolher uma cidade e clicar em próximo.
- () Uma tela para informar o e-mail e senha.
- () Uma tela com informações do meu perfil e o botão sair.

Tarefa 2 - Após realizar os passos da Tarefa 1, escolha a cidade “ARACAJU” e clique no botão “PONTOS TURÍSTICOS”. Após clicar no botão, o que você vê?

- () Uma tela com o título “Localização” onde posso escolher uma cidade e clicar em próximo.
- () Uma tela com o título “Seja Bem Vindo” e alguns botões.
- () Uma tela com o título “Detalhes” e algumas informações sobre um ponto turístico.
- () Uma tela com uma lista com pontos turísticos.

Tarefa 3 - Após realizar os passos da Tarefa 2, clique no item “Arcos da Orla de Atalaia”. Depois de clicar no item, o que você vê?

- () Uma tela com o título “Detalhes” e algumas informações sobre um ponto turístico.
- () Uma tela com informações do meu perfil.
- () Uma tela com uma lista de estabelecimentos.
- () Uma tela com detalhes de um evento.

Tarefa 4 - Agora volte até a tela com o título "Seja Bem Vindo" e clique no botão "Estabelecimentos". Depois de clicar no botão, o que você vê?

- () Uma tela com o título “Detalhes” e algumas informações sobre um ponto turístico.

- () Uma tela com informações do meu perfil.
- () Uma tela com uma lista de estabelecimentos.
- () Uma tela com detalhes de um evento.

ANEXO D – Questionário de avaliação de variáveis subjetivas

Questionário com um conjunto de perguntas fechadas e abertas que permitem aos participantes expressar sua opinião sobre a facilidade de uso, utilidade e sua intenção de usar o aplicativo Turistano no futuro.

E-mail: _____

Avaliação da Ferramenta

Para cada afirmativa abaixo, marque a opção que mais representa sua percepção sobre o aplicativo Turistano.

1 - O aplicativo Turistano é simples e fácil usar.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

2 - Em geral, o aplicativo Turistano é difícil de entender.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

3 - O aplicativo Turistano é fácil de aprender.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

4 - Acredito que o aplicativo Turistano aumentaria o tempo e o esforço necessários para obter informações sobre pontos e passeios turísticos, comparado com outros aplicativos.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

5 - Em geral, o aplicativo Turistano é útil.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

6 - Acredito que o aplicativo Turistano é inútil para a obtenção de informações sobre pontos e passeios turísticos.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

7 - Acredito que o aplicativo Turistano incorpora os mecanismos necessários para obter informações sobre pontos e passeios turísticos.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

8 - Em geral, acredito que o aplicativo Turistano não apoia de forma eficiente a obtenção de informações sobre pontos e passeios turísticos.

Discordo Totalmente 1 2 3 4 5 Concordo Totalmente

9 - O uso do aplicativo Turistano melhorará meu conhecimento sobre pontos e passeios turísticos.

Discordo Totalmente ()1 ()2 ()3 ()4 ()5 Concordo Totalmente

10 - Se eu precisar usar o aplicativo Turistano no futuro, acredito que não irei considerar seu uso.

Discordo Totalmente ()1 ()2 ()3 ()4 ()5 Concordo Totalmente

11 - Acredito que seria fácil se tornar experiente no uso do aplicativo Turistano.

Discordo Totalmente ()1 ()2 ()3 ()4 ()5 Concordo Totalmente

12 - Não tenho intenção de usar o aplicativo Turistano no futuro.

Discordo Totalmente ()1 ()2 ()3 ()4 ()5 Concordo Totalmente

13 - Eu recomendaria usar o aplicativo Turistano.

Discordo Totalmente ()1 ()2 ()3 ()4 ()5 Concordo Totalmente

14 - Caso queira compartilhar sugestões, críticas, elogios e melhorias, comente abaixo por favor.

Resposta: _____