

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Saad Barbosa

Desenvolvimento de Software para Avaliação de Moradia

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arthur Saad Barbosa

Desenvolvimento de Software para Avaliação de Moradia

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2023

Arthur Saad Barbosa

Desenvolvimento de Software para Avaliação de Moradia

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 13 de junho de 2021:

Maria Adriana Vidigal de Lima
Orientador

Júlia Tannús de Souza
Membro da Banca

Paulo Henrique Ribeiro
Membro da Banca

Uberlândia, Brasil
2023

Resumo

Este trabalho tem como principal objetivo a manutenção e o desenvolvimento de uma nova seção no aplicativo *Como Você Mora?*, que coleta de dados sobre avaliação de pós-ocupação (APO) em moradias, considerando casas e apartamentos. O aplicativo permite que qualquer pessoa que possua um aparelho Android, possa responder o questionário desenvolvido pelo laboratório do núcleo [MORA] - Pesquisa em Habitação da Faculdade de Arquitetura e Urbanismo da Universidade Federal de Uberlândia. Ao final do questionário, as respostas são enviadas à uma base de dados de APOs, permitindo que os pesquisadores arquitetos possam analisar os dados e gerar relatórios consolidados. A evolução do *Como Você Mora?* foi necessária para: (i) incluir uma nova seção de perguntas, relacionadas à hábitos sustentáveis no morar, (ii) melhorar o desempenho do aplicativo e das ações relacionadas à obtenção e armazenamento local das respostas, e (iii) aprimorar a comunicação entre o aplicativo e a base de dados de APOs, mantida pelo Sistema APO Digital.

Palavras-chave: Aplicativo, Android, Avaliação de pós-ocupação.

Lista de abreviaturas e siglas

| | |
|-------|--|
| ACID | Atomicidade, Consistência, Isolamento e Durabilidade |
| API | Application Programming Interface |
| APO | Avaliação de pós-ocupação |
| HTML | HyperText Markup Language |
| IDE | Integrated Development Environment |
| JSON | JavaScript Object Notation |
| MVC | Model View Controller |
| MVP | Model View Presenter |
| NoSQL | Not Only SQL |
| SDK | Software Development Kit |
| SQL | Structured Query Language |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| SGBDR | Sistemas de Gerenciamento de Banco de Dados Relacional |
| UML | Unified Modeling Language |
| XML | Extensible Markup Language |

Sumário

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 7 |
| 1.1 | Objetivos | 7 |
| 1.2 | Motivação | 8 |
| 1.3 | Organização do Trabalho | 8 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 10 |
| 2.1 | Java | 11 |
| 2.2 | Arquitetura | 11 |
| 2.3 | Bancos de Dados Relacionais e NoSQL | 12 |
| 2.3.1 | SQLite | 14 |
| 2.3.2 | Banco de Dados Orientado a Documentos | 14 |
| 2.4 | Manutenção de software | 15 |
| 2.4.1 | Ferramenta para Rastreamento de Falhas | 16 |
| 3 | AVALIAÇÃO DE PÓS-OCUPAÇÃO | 17 |
| 3.1 | <i>Como Você Mora?</i> | 19 |
| 3.2 | Histórico da Evolução do <i>Como Você Mora?</i> | 22 |
| 3.2.1 | Primeira Versão | 22 |
| 3.2.2 | Segunda Versão | 22 |
| 3.2.3 | Problemas apontados na Segunda Versão | 23 |
| 4 | EVOLUÇÃO DA APLICAÇÃO <i>COMO VOCÊ MORA?</i> | 25 |
| 4.1 | Ferramentas de desenvolvimento | 25 |
| 4.2 | Organização das Atividades | 26 |
| 4.3 | Inclusão da Seção de Hábitos Sustentáveis | 26 |
| 4.3.1 | Passos para a construção de uma tela | 27 |
| 4.4 | Armazenamento | 30 |
| 4.5 | Envio de respostas | 30 |
| 4.6 | Correções Gerais | 31 |
| 5 | CONCLUSÃO | 34 |
| 5.1 | Considerações Finais | 34 |
| 5.2 | Trabalhos Futuros | 34 |
| | REFERÊNCIAS | 36 |

| | |
|--|-----------|
| ANEXOS | 38 |
| ANEXO A – SEÇÃO DE HÁBITOS SUSTENTÁVEIS DO <i>COMO VOCÊ MORA?</i> | 39 |

1 Introdução

Seja a roupa que você usa, o alimento que você ingere ou o transporte que utiliza, toda escolha de consumo traz alguma consequência para o mundo. A dependência do crescimento econômico para atender às demandas da população é, de certa forma, insustentável e composta por custo social e ambiental [Energia \(2021\)](#).

Para que um software seja considerado funcionalmente bom, é essencial que ele cumpra com os requisitos especificados pelo cliente, em todas as suas fases de implementação. Não só isso, mas que também seja confiável, com a menor quantidade possível de bugs (sabemos que 0 bugs é mito). E, finalmente, que tenha uma boa performance e uma interface intuitiva. Afinal, se a interface não for boa o suficiente, não importa que o código seja o mais modularizado e limpo do mundo: o produto não será utilizável ([SILVEIRA, 2019](#)).

O presente trabalho pretende a manutenção e o desenvolvimento de uma nova seção do aplicativo *Como você Mora?*, parte integrante do Sistema APO Digital, idealizado e construído por professores e alunos dos cursos de graduação em Arquitetura e Urbanismo, Design, Artes Visuais, Ciência da Computação e Sistemas de Informação da Universidade Federal de Uberlândia ([VILLA; ORNSTEIN, 2016](#)). O *Como você Mora?* é uma avaliação de pós-ocupação (APO) composta de um questionário especialmente confeccionado para coletar informações sobre o morar, considerando características socioeconômicas do morador, aspectos gerais da habitação como acessibilidade e segurança, existência de equipamentos de uso coletivo e atividades realizadas nos espaços da moradia.

A atualização do componente *Como você Mora?* tem como propósito introduzir uma nova seção de perguntas, corrigir falhas detectadas e melhorar a qualidade do aplicativo, por meio de testes e correção de erros.

1.1 Objetivos

O objetivo geral deste projeto é atualizar o aplicativo *Como você Mora?* com a inclusão de uma nova seção sobre hábitos sustentáveis do morador e com a introdução de diversas melhorias na interface e na comunicação com os outros componentes do Sistema APO Digital. O aplicativo deve garantir que todo o questionário seja visualizado e respondido pelo usuário morador de forma fluida e eficiente, e que todas as respostas sejam coletadas e devidamente armazenadas em banco de dados local e posteriormente enviadas à base de dados principal, mantida pelo Sistema APO Digital. O Sistema APO Digital se destina a apoiar atividades nas disciplinas da área de Projeto Arquitetônico e a inclusão

de novas seções, como a de hábitos sustentáveis, permite colaboração também em áreas como Conforto Ambiental, Sustentabilidade e Eficiência Energética.

1.2 Motivação

O *Como Você Mora?* permite proporcionar melhores recursos e tecnologias para aumentar a abrangência das aplicações de questionários habitacionais, bem como estender os campos de pesquisas habitacionais e sociais, facilitando o trabalho dos profissionais de arquitetura e afins que aplicam os questionários. Neste contexto, a motivação principal deste projeto é fornecer ao usuário um aplicativo de qualidade, em que ele possa interagir sem falhas, fornecer respostas corretas e chegar ao final do questionário de maneira descomplicada e agradável. Além disso, a inclusão de uma nova seção abordando hábitos sustentáveis permite que a APO apresente pontos informativos e de aprendizado para o morador, como manejo correto do lixo e pontos de coleta de material reciclável durante a aplicação do questionário.

Além do usuário que irá responder ao questionário, existem também os pesquisadores que necessitam que as respostas coletadas sejam comunicadas sem erros através dos componentes do sistema, para que ao final, sejam capazes de compreender com clareza os resultados da pesquisa.

1.3 Organização do Trabalho

O presente trabalho foi desenvolvido em 5 capítulos, organizados e divididos da seguinte forma:

- Capítulo 1: descreve a introdução do trabalho e os principais objetivos do mesmo.
- Capítulo 2: expõe a fundamentação teórica utilizada neste trabalho, abordando Sistema Android, Java, Arquitetura de Projeto, Projeto de Banco de Dados e Metodologia de Evolução de Software.
- Capítulo 3: apresenta o aplicativo *Como Você Mora?*, seu histórico, suas versões e o levantamento de falhas apuradas após a realização de teste em campo.
- Capítulo 4: relata o desenvolvimento das modificações relacionadas à evolução do *Como Você Mora?*, atendendo às correções identificadas e às demandas do grupo [MORA]. O trabalho realizado baseou-se na fundamentação apresentada no Capítulo 2, na documentação das versões anteriores do aplicativo e nas necessidades dos usuários.

- Capítulo 5: comunica as conclusões obtidas a partir do trabalho desenvolvido, indica melhorias para trabalhos futuros e mostra outras perspectivas que podem ser exploradas.

2 Fundamentação Teórica

Este capítulo tem como objetivo apresentar as tecnologias, conceitos e ferramentas que foram utilizadas no desenvolvimento do presente projeto. A aplicação *Como Você Mora?* foi construída utilizando-se o sistema operacional Android, caracterizando-se, portanto, em um aplicativo. A acessibilidade foi uma das principais motivações pois, hoje em dia, a maioria das pessoas possui acesso à internet e utiliza dispositivos móveis, como smartphones. Portanto, desenvolver um aplicativo possibilita alcançar um público amplo e tornar a solução facilmente acessível.

Por ser instalado diretamente nos dispositivos dos usuários, o aplicativo proporciona uma experiência contínua e imediata e pode aproveitar recursos específicos, como câmera, GPS, entre outros, para fornecer funcionalidades avançadas e uma experiência mais personalizada. Ainda, pode aproveitar recursos de armazenamento local, o que permite acesso mais rápido a dados e funcionalidades, mesmo em condições de conectividade limitada.

A escolha do sistema operacional Android foi definida baseando-se na quantidade de usuários que possuem dispositivos Android. Segundo dados da [Statista \(2023\)](#), o sistema operacional Android é utilizado por cerca de 71% dos usuários de dispositivos móveis no mundo. Além disso, o desenvolvimento Android possui um custo menor e mais acessível aos desenvolvedores, visto que para desenvolver para iOS é necessário um Mac. Por outro lado, o desenvolvimento Android pode ser feito em computadores com Windows, Linux e Mac.

O desenvolvimento do aplicativo *Como Você Mora?* foi feito utilizando a plataforma Android Studio, uma IDE para desenvolvimento Android. Algumas das principais vantagens de utilizar o Android Studio são:

- **Integração com o ecossistema Android:** Oferece integração nativa com todas as ferramentas e recursos da plataforma. Isso inclui suporte às diversas versões do Android, APIs, bibliotecas e serviços do Google Play.
- **Editor de layout visual:** Inclui um editor de layout visual que permite criar e visualizar a interface de usuário do aplicativo em tempo real. Isso agiliza o processo de criação e facilita a implementação de interfaces de usuário atraentes e funcionais.
- **Depuração avançada:** Possui um depurador integrado que permite aos desenvolvedores rastrear e corrigir problemas de código durante a execução do aplicativo. Suporta também monitoramento de desempenho e análise de uso de recursos.

- Gerenciador de dependências e build automatizado: Possui suporte integrado ao sistema de gerenciamento de dependências e inclusão de bibliotecas externas de forma eficiente.
- Suporte a emuladores e dispositivos virtuais: Fornece emuladores de dispositivos virtuais que permitem executar e testar o aplicativo em diferentes versões do Android e configurações de hardware. Esta funcionalidade é extremamente útil para testar a compatibilidade do aplicativo e garantir que ele funcione corretamente em uma ampla variedade de dispositivos.

Para além destas vantagens citadas, o Android Studio também oferece recursos adicionais, como análise de código estática, suporte a controle de versão, refatoração de código e suporte a várias linguagens de programação.

O sistema Android é usado em uma ampla variedade de dispositivos, tamanhos de tela, versões do sistema operacional e configurações de hardware. Embora este fato possa trazer desafios em termos de compatibilidade e teste, os desenvolvedores têm a oportunidade de atender a uma variedade de dispositivos, sem a necessidade de possuir cada dispositivo para testes.

2.1 Java

Como linguagem de desenvolvimento, foi utilizada a linguagem Java, amplamente usada para codificar aplicações Web. Ela tem sido uma escolha popular entre os desenvolvedores há mais de duas décadas, com milhões de aplicações Java em uso hoje. Java é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si. É uma linguagem de programação rápida, segura e confiável para codificar tudo, desde aplicações móveis e software empresarial até aplicações de big data e tecnologias do servidor ([ORACLE](#), ; [ARNOLD](#); [GOSLING](#); [HOLMES, 2005](#)).

Apesar da atual recomendação que o Google faz pela linguagem Kotlin para o desenvolvimento Android, foi utilizada a linguagem Java para a implementação do aplicativo *Como Você Mora?* devido ao forte suporte da comunidade e quantidade de tutoriais que hoje existem e facilitam o desenvolvimento.

2.2 Arquitetura

Arquitetura de software é um conceito abstrato, que se refere à organização de um sistema. Ela é responsável por definir os componentes que farão parte de um projeto, suas características, funções e a forma como devem interagir entre si e com outros softwares ([PRESSMAN](#); [MAXIM, 2021](#)). Além disso, a arquitetura permite uma maior

escalabilidade do software, como também, permite que diferentes desenvolvedores consigam entender a estrutura que deve refletir o código.

O aplicativo *Como Você Mora?* foi estruturado utilizando a arquitetura Model-View-Presenter (MVP). A arquitetura MVP foi introduzida pela primeira vez em 1990 pela IBM e possui um padrão de desenvolvimento desacoplado que permite que vários desenvolvedores trabalhem e testem simultaneamente. Para tornar o processo de desenvolvimento de aplicativos mais eficiente e o processo de manutenção mais rápido, foi necessário definir camadas separadas de forma adequada. Assim, as características da arquitetura MVP no desenvolvimento de aplicativos são:

- separação de tarefas em segundo plano das atividades,
- divisão de tarefas complexas em tarefas mais simples,
- criação de visualizações independentes da fonte de dados, e
- uso de testes unitários automatizados.

Nesta arquitetura existem 3 separações claras, definidas como camadas:

1. **Model:** camada contendo classes responsáveis por manter os dados que serão apresentados nas interfaces, além de lidar com regras de negócio.
2. **View:** camada responsável por apresentar os dados aos usuários, parte visual.
3. **Presenter:** camada responsável por lidar com interações do usuário na parte visual e por formatar como a view irá apresentar os dados.

O modelo MVP é derivado do padrão MVC (Model, View, Controller), com diferenças fundamentais entre os dois: no MVP, não é recomendado que a View se comunique diretamente com o Model, mas sim através de um intermediário chamado Presenter ou uma classe Contract. A representação do padrão MVP pode ser compreendida pela Figura 1, através de um diagrama de sequência.

2.3 Bancos de Dados Relacionais e NoSQL

Os Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDRs) são amplamente utilizados e considerados um padrão de mercado para o armazenamento e recuperação de dados estruturados. Uma das principais vantagens é a garantia de integridade dos dados, que é assegurada pelas propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

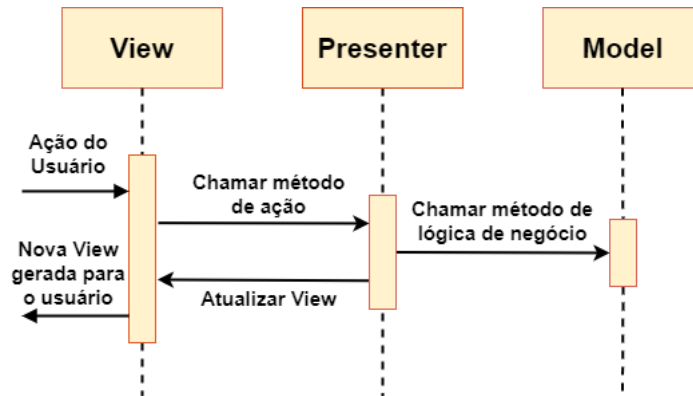


Figura 1 – Diagrama de Sequência do MVP. Adaptado de (ZHANG; LUO, 2010)

No entanto, conforme o volume de dados aumenta, manter a integridade dos dados em um SGBDR pode se tornar uma tarefa onerosa. O processamento eficiente de grandes volumes de dados em tempo hábil usando sistemas relacionais pode ser limitado. Essa limitação levou ao surgimento de estudos na área e ao desenvolvimento de bancos de dados NoSQL, também conhecidos como bancos de dados não-convencionais.

Os bancos de dados NoSQL são uma classe de sistemas de banco de dados que podem renunciar às propriedades ACID e a esquemas pré-definidos em prol do desempenho e da escalabilidade. Eles foram projetados para lidar com grandes volumes de dados e oferecer alta velocidade de leitura e gravação.

Um dos tipos de banco de dados NoSQL é o orientado a documentos. Nesse modelo, os dados são armazenados como documentos, geralmente em formato JSON ou XML, e não em tabelas com esquemas rígidos como em um banco de dados relacional. Os bancos de dados orientados a documentos permitem flexibilidade no esquema dos dados, facilitando a adição e modificação de campos sem a necessidade de alterar a estrutura do banco de dados.

Essa abordagem NoSQL orientada a documentos pode ser vantajosa em casos em que a estrutura dos dados é variável ou não está completamente definida antecipadamente. Além disso, esses bancos de dados são projetados para escalabilidade horizontal, permitindo a distribuição dos dados em vários servidores para melhorar o desempenho.

Os bancos de dados NoSQL, incluindo os orientados a documentos, surgiram como uma alternativa aos SGBDRs tradicionais, oferecendo desempenho e escalabilidade em detrimento das garantias ACID e dos esquemas rígidos. A escolha entre um SGBDR e um banco de dados NoSQL depende das necessidades específicas do projeto, considerando fatores como o volume de dados, o desempenho necessário e a estrutura dos dados.

No projeto do aplicativo *Como Você Mora?* utilizou-se o banco de dados SQLite para armazenamento local e temporário das respostas coletadas no decorrer do questionário no dispositivo. A base de dados de APOs, responsável por armazenar definitivamente

as respostas dos moradores foi construída utilizando-se tecnologia NoSQL, notadamente banco de dados orientado a documentos. A implementação da comunicação entre aplicativo e base de dados do Sistema APO Digital, iniciando-se na formatação e envio das respostas, considera então as diferentes tecnologias de cada componente.

2.3.1 SQLite

O SQLite ([HIPP, 2020](#)) é um sistema de gerenciamento de banco de dados relacional leve e autônomo, amplamente utilizado em aplicativos móveis e embutidos. Opera em um arquivo de banco de dados único com extensão `.sqlite` ou `.db`. Esse arquivo contém todas as tabelas, índices, esquemas e outros elementos do banco de dados. Diferentemente de outros sistemas de gerenciamento de banco de dados, não há um processo de servidor separado em execução.

O SQLite oferece suporte à linguagem SQL (Structured Query Language), que é uma linguagem padrão para interagir com bancos de dados relacionais. Desta forma, pode-se usar comandos SQL para criar, modificar, consultar e manipular dados. É um SGBD altamente portátil e pode ser usado em várias plataformas, incluindo Windows, macOS, Linux, iOS, Android e muitas outras. No desenvolvimento do aplicativo *Como Você Mora?* a interação com o SQLite utilizou uma API compatível com a linguagem Java.

2.3.2 Banco de Dados Orientado a Documentos

No modelo orientado a documentos, os dados são armazenados como documentos encapsulados em pares de chave-valor, geralmente em formato JSON ou XML. Cada documento é único dentro de uma coleção e recebe um identificador. Ao contrário do modelo relacional, os documentos não possuem uma estrutura rígida definida, ou seja, não têm um esquema de dados pré-definido.

Essa flexibilidade é uma das principais características dos bancos de dados orientados a documentos. Não há a necessidade de seguir uma estrutura fixa de tabelas com colunas definidas. Em vez disso, cada documento pode ter sua própria estrutura, contendo diferentes campos e tipos de dados. Essa abordagem é particularmente útil quando os dados possuem estrutura variável ou não estão completamente definidos antecipadamente.

Além disso, ao armazenar os dados em formato JSON, há a vantagem adicional de suportar tipos de dados, o que torna a forma de armazenamento mais amigável para os desenvolvedores. O JSON permite representar diferentes tipos de dados, como strings, números, valores booleanos, objetos aninhados e arrays.

Existem diversos bancos de dados orientados a documentos disponíveis, sendo que alguns dos exemplos mais conhecidos são:

- CouchDB: Banco de dados orientado a documentos de código aberto que segue a arquitetura RESTful e usa o formato JSON para armazenar os documentos. Possui recursos de replicação e sincronização entre diferentes instâncias ([APACHE, 2023](#)).
- MongoDB: Banco de dados NoSQL que utiliza documentos BSON (uma representação binária do JSON) para armazenar os dados. O MongoDB é conhecido por sua escalabilidade e desempenho, além de fornecer recursos avançados de consulta e indexação ([HOWS; MEMBREY; PLUGGE, 2019](#)).

A escolha do banco de dados mais adequado depende dos requisitos específicos do projeto e das necessidades de desempenho, escalabilidade e flexibilidade de modelagem de dados.

2.4 Manutenção de software

O desenvolvimento de um software tem como base uma especificação de requisitos, contendo as necessidades do cliente. Após finalizada a construção do software, o mesmo é liberado aos usuários finais para utilização. Neste momento inicia-se a fase de manutenção, em que os próprios usuários identificam a necessidade de novas funcionalidades e solicitam alterações para que o software se adeque melhor aos propósitos desejados e opere com mais eficiência. A manutenção pode ser adaptativa, corretiva ou evolutiva, garantindo a vida útil do software e permitindo adicionar ou modificar funcionalidades para satisfazer a novos requisitos ou mudanças no ambiente ([PRESSMAN; MAXIM, 2021](#)).

O campo da evolução de software é responsável pela manutenção de um sistema, sendo pela reengenharia, refatoração ou pelo gerenciamento de sistemas legados. A evolução de um software é importante, pois as empresas são dependentes dos sistemas que investiram e utilizam. Todo o processo de evolução de software tem seu início a partir de uma solicitação de mudança. O passo seguinte consiste em realizar uma análise de impacto para identificar os efeitos desta modificação na aplicação. Após esta análise, segue o planejamento da entrega considerando: corrigir defeitos, adaptar à um novo ambiente e incorporar melhorias no produto, conforme Figura 2. A partir do planejamento é realizada a implementação da mudança e, se necessário, pode ser feito o planejamento da entrega ([SOMMERVILLE, 2011](#)).

A atividade de manutenção segue algumas etapas para a obtenção de resultado positivo. A avaliação inicial permite conhecer a documentação existente e validar as solicitações de modificações vindas dos usuários. Em seguida, a análise de impactos e a implementação são executadas e a etapa mais crítica é a realização de testes, para garantir que as modificações foram corretamente codificadas e não introduziram novos problemas.

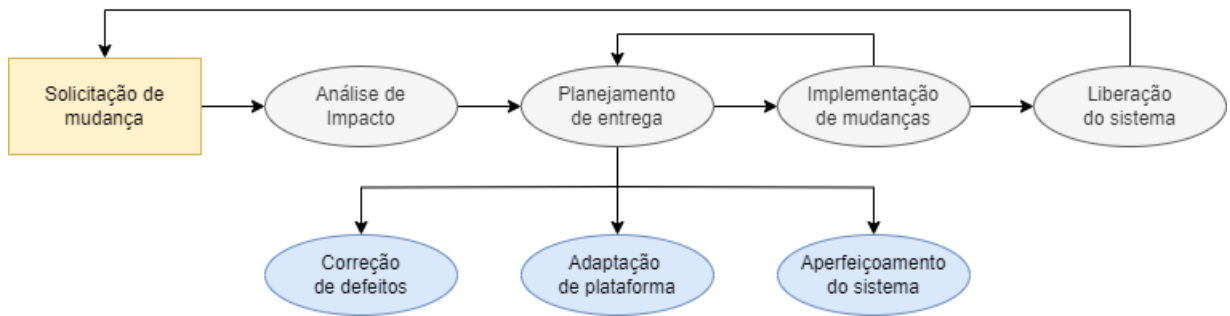


Figura 2 – Processo de Evolução de Software. Adaptado de (SOMMERVILLE, 2011)

Isso pode incluir testes de unidade, testes de integração, testes de sistema e outros tipos de testes relevantes para o contexto do sistema.

Para auxiliar na etapa de testes e rastreamento de falhas na implementação das mudanças referentes ao aplicativo *Como Você Mora?*, foi utilizada uma ferramenta de relatório de falhas, que auxilia no monitoramento e na correção de problemas.

2.4.1 Ferramenta para Rastreamento de Falhas

Um dos grandes desafios do desenvolvimento de aplicativos é acompanhar possíveis problemas que usuários enfrentam ao utilizá-lo. O Firebase Crashlytics é uma ferramenta leve de relatório de falhas que, em tempo real, ajuda a monitorar, priorizar e corrigir problemas de estabilidade que comprometem a qualidade do aplicativo. O Crashlytics economiza tempo na solução de problemas com o agrupamento inteligente das falhas e a exibição das circunstâncias que levam a elas. Além disso, oferece recursos adicionais, como relatórios de estabilidade, que mostram a frequência e a gravidade das falhas ajudando a identificar as áreas problemáticas no código e priorizar as correções.

Outra vantagem do Firebase Crashlytics é a integração com as ferramentas do Firebase, como o Firebase Analytics. Essa integração permite correlacionar eventos do usuário com falhas específicas, fornecendo uma visão abrangente do desempenho e do impacto das falhas (FIREBASE, 2023).

Considerando-se que este trabalho pretende a evolução do aplicativo *Como Você Mora?*, uma ferramenta para identificar e registrar automaticamente as falhas e exceções ocorridas no aplicativo durante o desenvolvimento foi de uso importante para prover uma visão detalhada dos problemas que seriam enfrentados pelo usuário no uso do aplicativo. O registro de uma falha indica linha de código onde ocorreu o erro, a pilha de chamadas e os parâmetros relevantes.

3 Avaliação de Pós-ocupação

Um ato essencial da existência humana é *morar*, e por isso a questão da qualidade da habitação tem sido estudada sob diferentes abordagens (PREISER; WHITE; RABINOWITZ, 2016; VILLA; ORNSTEIN, 2016; HADJRI; CROZIER, 2009). O local da moradia permite ao ser humano sentir segurança física e psicológica, privacidade e conforto; realizar atividades produtivas, de lazer e domésticas, além de estabelecer relações em comunidade. Uma forma de medir a qualidade de habitações, considerando a satisfação dos usuários, é através de avaliação de pós-ocupação (APO). Uma APO pode ser vista como um conjunto de questões, aplicadas ao morador, para avaliação de desempenho do ambiente construído de forma a identificar os aspectos positivos e negativos percebidos sobre a moradia.

Com o objetivo de desenvolver um sistema interativo de avaliação pós-ocupação e desenvolver estudos de caso na cidade de Uberlândia (MG), estabeleceu-se uma parceria entre pesquisadores da Faculdade de Arquitetura e Urbanismo e da Faculdade de Computação da Universidade Federal de Uberlândia, para a construção dos questionários de APOs usando interfaces digitais, ampliando a discussão sobre metodologias e métodos mais eficientes de coleta e análise dos dados.

O trabalho estruturou-se em: (i) pesquisa bibliográfica sobre tipologias, aspectos e avaliação do morar; (ii) categorização e definição dos principais atributos abordados na avaliação; (iii) identificação dos limites e recortes do sistema de avaliação a ser desenvolvido, caracterizando e definindo suas partes, ferramentas, funcionamento e seus objetivos; (iv) desenvolvimento do sistema interativo de APO em meios digitais; (v) aplicação de pré-testes das ferramentas do sistema em protótipos funcionais; (vi) desenvolvimento do sistema de APO interativo em meios digitais (internet, tablets e smartphones); (vii) aplicação de testes do sistema interativo de APO em meios digitais em diferentes tipos de habitações na cidade de Uberlândia (MG); e (viii) disponibilização do sistema de APO em meios digitais para a comunidade.

O Sistema APO Digital, desenvolvido com o intuito de explorar recursos interativos e investigar interfaces digitais mais amigáveis e intuitivas para a aplicação de APOs, pode ser visto na Figura 3. A coleta de informações e a organização das mesmas em uma base de dados central (bancos de dados sobre o morar) permitiu disponibilizar dados estatísticos e científicos aos pesquisadores arquitetos. Diversos aplicativos para coleta de dados em dispositivos móveis foram desenvolvidos, para diferentes aplicações de APOs, utilizando abordagens 2D e 3D, e em diferentes linguagens de programação. (VILLA et al., 2015; PEREIRA et al., 2017; ABREU et al., 2017; MARCELINO, 2021). O Sistema

APO Digital foi registrado no Instituto Nacional de Propriedade Industrial, estando a cotitularidade firmada entre UFU e FAPEMIG¹.

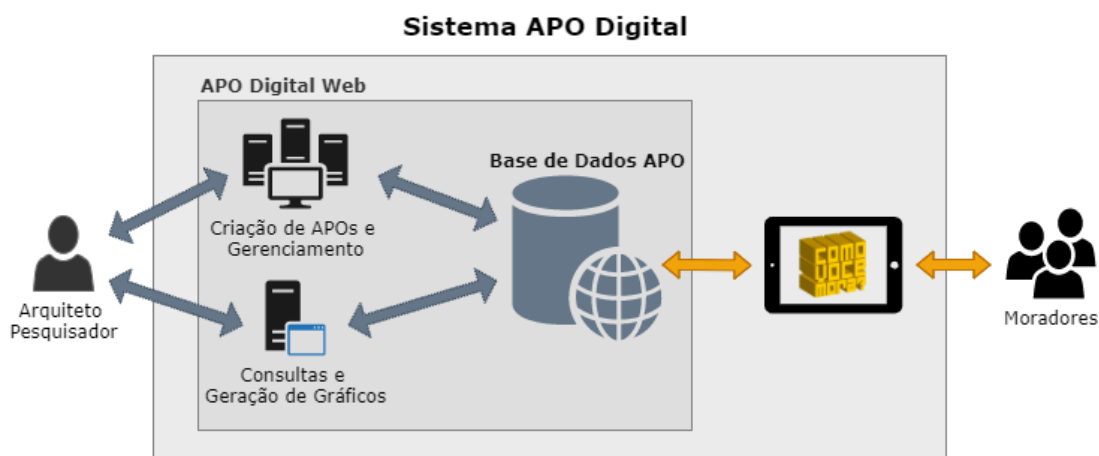


Figura 3 – Sistema APO Digital e seus componentes. Fonte: (MARCELINO, 2021)

A aplicação *APO Digital Web* foi projetada em estrutura cliente-servidor e desenvolvida em linguagem Java para ser executada em plataforma *web*, acessível por qualquer navegador (Chrome, Firefox, Opera, Internet Explorer). A *APO Digital Web* permite a criação de novas APOs, o gerenciamento de APOs já realizadas e o processamento de consultas. O projeto de arquitetura de software da *APO Digital Web* seguiu o padrão Modelo-Visão-Controlador (MVC) com o intuito de separar os dados e a lógica da APO (Modelo). Para implementar o padrão MVC e facilitar a configuração da aplicação *web*, utilizou-se o arcabouço Spring Boot, que é uma solução da Plataforma Spring caracterizada pela convenção sobre configuração, que visa diminuir o número de decisões a serem tomadas pelos desenvolvedores, assim, ganhando simplicidade sem perder flexibilidade (PEREIRA et al., 2017).

O Sistema APO Digital teve seu desenvolvimento norteado pelas seguintes diretrizes (VILLA et al., 2015): (i) ampliação da eficiência e da confiabilidade dos resultados da avaliação obtidos por métodos quantitativos; (ii) possibilidade de maior interação entre o pesquisador e o morador na avaliação, explorando-se formas de interatividade entre recursos gráficos e moradores; (iii) redução dos custos da avaliação e no tempo da pesquisa; (iv) ampliação da eficiência da tabulação dos resultados da avaliação através da utilização de programas auxiliares para a análise e formatação dos dados; (v) capacidade gráfica e multimídia do meio digital e exploração de recursos gráficos na pesquisa avaliativa dos espaços habitacionais; (vi) utilização da tecnologia não somente como equipamento, mas como parte funcional e integral da avaliação, visando aproximar usuários do processo de

¹ Número do registro: BR512017000731-4, data de registro: 15/03/2017, título: "Sistema APO Digital", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.

criação da interface; (vii) capacidade da avaliação se tornar educativa reforçando novas agendas em relação a aplicação de APO como processo e não somente como produto.

3.1 Como Você Mora?

Os aplicativos têm o objetivo de apresentar o questionário da APO desejada, de forma interativa e gamificada, e também de promover informação e conhecimento para os próprios moradores, usuários do sistema, na medida em que, durante a APO, refletem sobre aspectos relativos ao *morar* e seus impactos no ambiente em que residem. O aplicativo *Como Você Mora?* teve sua evolução realizada no presente trabalho, em parceria com o trabalho de [Marcelino \(2021\)](#). Foi concebido para avaliar casas e apartamentos, e um personagem foi criado para guiar o usuário morador durante o percurso da avaliação. O **Dr. Prancheta** está ilustrado na Figura 4 nos seus quatro perfis diferentes. A intenção do uso deste personagem é de proporcionar aos usuários do aplicativo um ambiente mais receptivo, interativo e agradável.



Figura 4 – Personagem Dr. Prancheta. Fonte: ([MARCELINO, 2021](#))

O *Como Você Mora?* foi desenvolvido separando-se as questões em diferentes *seções*. Cada seção trata de um tema específico e de seus conceitos. Foram definidas 7 seções para o *Como Você Mora?*, cada uma delas abordando uma temática específica no aplicativo:

1. **Sobre Você:** questões pessoais sobre o usuário morador para conhecer o gênero, escolaridade, idade, entre outras.
2. **Moradia Anterior:** questões envolvendo o tipo da moradia anterior e o nível de satisfação do morador em relação à mesma.
3. **Moradia Atual:** questões sobre o entorno da moradia atual, endereço, presença de comércio no bairro, equipamentos urbanos na região.
4. **Moradores:** questões sobre as pessoas que habitam na residência: a quantidade, a renda familiar e a existência de funcionários na moradia.

5. **Avaliando o Edifício:** questões gerais sobre a construção, a qualidade da construção, segurança e privacidade em relação à rua, entre outras.
6. **Avaliando a Unidade:** questões sobre os cômodos: tamanhos, atividades realizadas nos mesmos, acomodação de móveis, eventuais reformas.

As questões da APO podem ser abertas ou fechadas. A Figura 5 mostra os elementos de apresentação e coleta das respostas dos moradores. O item 1 apresenta uma barra deslizante para que o morador escolha sua resposta. À medida em que o usuário movimenta o pino da barra as opções são apresentadas e ele pode definir sua escolha. O item 2 ilustra o uso de botões contendo as respostas possíveis. Os botões podem ser usados para questões com escolha única ou múltipla escolha: basta que o usuário pressione o botão com a resposta desejada (ou respostas desejadas). O item 3 mostra a seleção de prioridades com um pódio de quatro lugares. A opção de maior prioridade fica na posição mais alta de pódio, e assim sucessivamente. O item 4 mostra a caixa de texto, que deve ser associada às questões abertas.

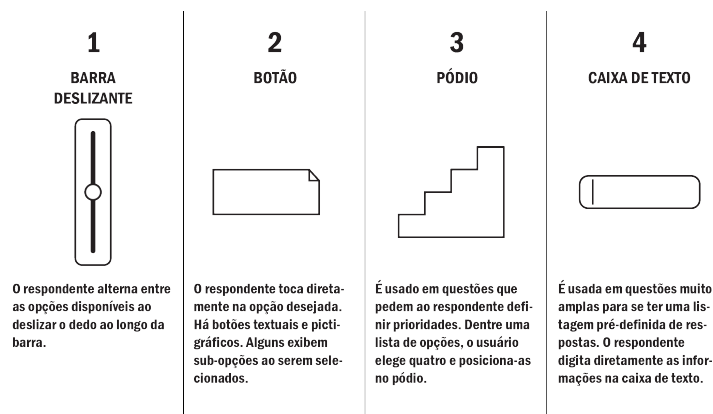


Figura 5 – Modelo de respostas para questões da APO. Fonte: (MARCELINO, 2021)

Conforme as questões vão sendo apresentadas ao usuário morador, o *Como Você Mora?* organiza dinamicamente o fluxo das perguntas. Esta organização é necessária pois, dependendo de uma resposta, outras questões podem aparecer ou não. Por exemplo, se a questão apresentada é “Você sempre morou no mesmo local?” e a resposta do morador for “Sim”, então não devem ser apresentadas as questões sobre a moradia anterior, que interpelam sobre o tipo, a localização, o grau de satisfação geral, o padrão de acabamento, entre outros aspectos. As respostas fornecidas pelo usuário morador são armazenadas local e temporariamente, nos respectivos dispositivos móveis, utilizando-se o SQLite (HIPPI, 2020).

Um exemplo dessa organização dinâmica do fluxo de perguntas acontece no cenário da pergunta da Figura 6. Caso o usuário responda que já morou em outro local, ele é redirecionado para a próxima pergunta (Figura 8). Caso a resposta seja que sempre

morou no mesmo local, ele é redirecionado para a próxima seção, conforme mostrado na Figura 7.

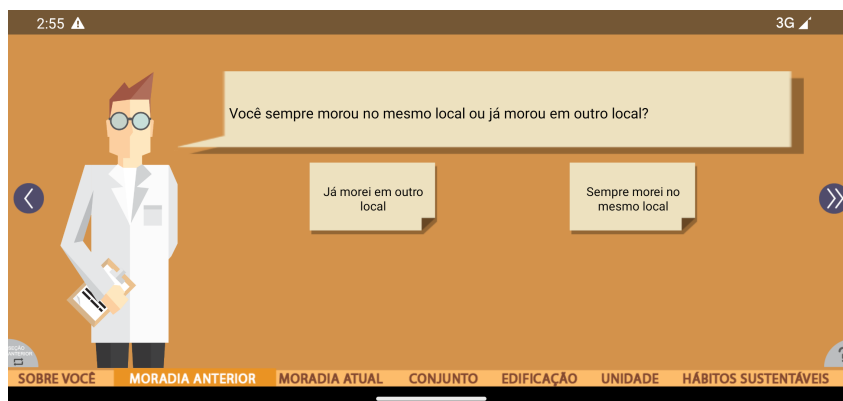


Figura 6 – Pergunta sobre moradia anterior

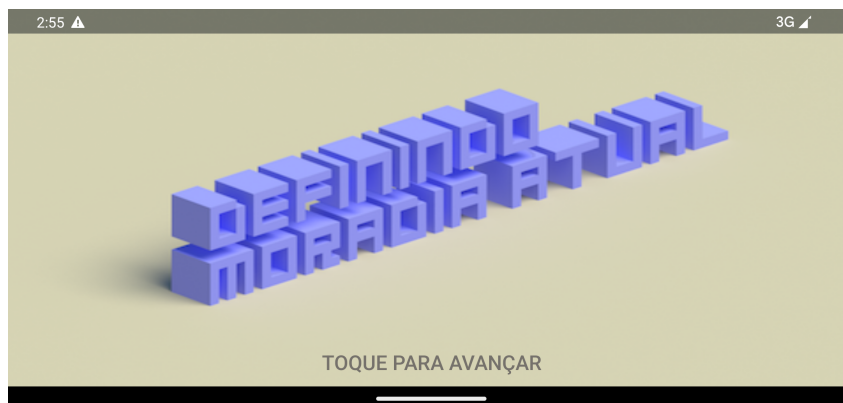


Figura 7 – Seção moradia atual

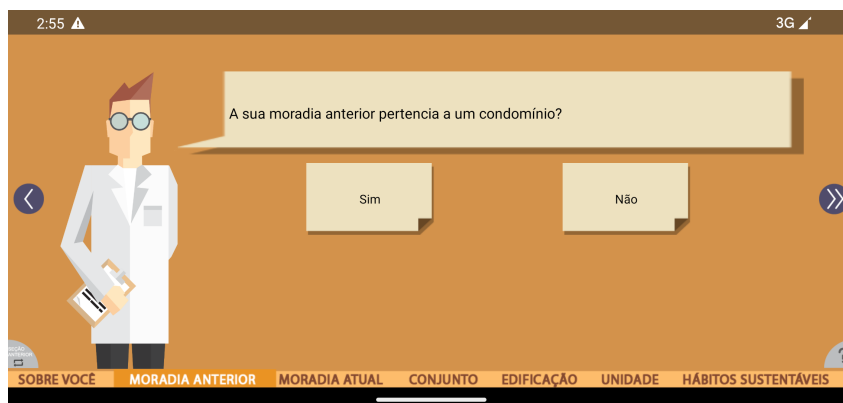


Figura 8 – Pergunta sobre moradia anterior

3.2 Histórico da Evolução do *Como Você Mora?*

O aplicativo *Como Você Mora?* teve duas versões iniciais construídas e testadas pelo grupo [MORA], e uma terceira versão foi demandada, objeto deste trabalho, baseando-se nos resultados da aplicação da segunda versão.

3.2.1 Primeira Versão

Em todo o processo de desenvolvimento do aplicativo *Como Você Mora?* foram estudadas possibilidades e metodologias adequadas de experiência do usuário (UX Design) e de gamificação para propiciar à interface a competência para não só cumprir seu objetivo principal de coletar dados sobre o morar, mas para fazer desse processo de avaliação uma experiência mais positiva, criativa, educativa e interativa para os moradores respondentes. A primeira versão do aplicativo em 2D utilizou o framework Corona SDK e linguagem Lua tendo sido uma experiência bastante satisfatória em relação à imersão dos usuários no questionário, mas em relação ao tempo de resposta o aplicativo se mostrou não tão eficaz (VILLA et al., 2015).

Após ser rodada uma pesquisa utilizando o questionário, foram levantados vários pontos de melhoria a serem realizados antes de ser rodada a pesquisa novamente. Um destes pontos foi a exclusão de algumas perguntas e a transferência de algumas perguntas entre as seções existentes. Essas mudanças nos questionários foram pensadas para ajudar o usuário, diminuindo o questionário e também melhorando o fluxo das perguntas, aproximando aquelas que fizessem sentido estarem juntas.

Além da mudança do questionário foram encontrados problemas de usabilidade de alguns componentes que estão na Figura 5. Dentro da plataforma em que o aplicativo estava desenvolvido, que era a Corona SDK, não eram encontradas muitas soluções de problemas por não existir uma comunidade muito ativa nos dias atuais. Um dos maiores problemas da plataforma antiga era a linguagem utilizada, que era a linguagem Lua, que apesar de ser fácil de compreender, ela não é muito utilizada por outras plataformas, o que acaba tornando a comunidade muito pequena e difícil de colaboração. Por isto a equipe de desenvolvimento chegou à conclusão que era uma boa hora de migrar a plataforma de todo o aplicativo.

3.2.2 Segunda Versão

A plataforma escolhida para a migração do aplicativo, foi o Android Studio, que é uma plataforma atual, lançada em 2013, e bastante utilizada no mundo todo, possuindo uma comunidade robusta que auxilia bastante no enfrentamento de problemas no desenvolvimento. Além da plataforma ser atual, a mesma é recomendada pela própria Google

que acompanha os aplicativos desenvolvidos para Android. Com o Android Studio também foi alterada a linguagem de programação utilizada para este novo desenvolvimento.

A linguagem escolhida foi a linguagem Java que possui uma grande comunidade, além de ser ensinada na faculdade, o que pôde alavancar as alterações necessárias ao projeto. Tendo em vista que ela é ensinada no próprio curso, qualquer outro estudante do curso de computação consegue ter um norte para compreender o que foi desenvolvido, sem precisar apreender toda uma nova linguagem. Esta linguagem é muito utilizada por ser usada não apenas na comunidade do Android Studio, mas também em várias áreas da computação, tornando a comunidade muito ativa no mundo todo.

Além destes pontos, algo que estava sendo muito desejado pela equipe era a transferência das respostas fornecidas pelos moradores durante o uso do aplicativo para o Sistema APO Digital de forma padronizada e correta. Inicialmente as respostas vão sendo coletadas e mantidas temporariamente no armazenamento local do dispositivo móvel, utilizando-se o SGBD SQLite. A transferência foi planejada para ser realizada no padrão JSON, que é um padrão universal de transferência de dados, sendo de fácil manutenção e compreensão, considerando futuras manutenções. Outro motivo para o uso do padrão JSON, é o fato da base de dados do Sistema APO Digital utilizar a tecnologia NoSQL com SGBD MongoDB. Neste caso, os dados já estariam no formato nativo do armazenamento.

Para utilizar o banco de dados SQLite e fazer a transferência dos dados locais o Sistema APO Digital, foram utilizadas as bibliotecas suportadas pelo Android Studio. Para cada dispositivo, o envio das respostas do questionário foi planejado ser realizado a cada final de questionário. Ao finalizar um questionário, o morador teria a opção de enviar ao Sistema APO Digital suas respostas. Caso uma conexão com a Internet não estivesse disponível, este questionário poderia ser enviado em um momento posterior.

3.2.3 Problemas apontados na Segunda Versão

Com o objetivo de validar o aplicativo gamificado *Como Você Mora? - Segunda Versão* foram realizados testes experimentais com enfoque na aferição da qualidade na exibição dos diversos tipos de perguntas, do armazenamento local das respostas, do desempenho do aplicativo e da verificação de possíveis erros. Os testes foram realizados pelos alunos responsáveis pela construção do aplicativo, por alunos responsáveis pelo design gráfico e confecção do questionário e por outros membros envolvidos na pesquisa.

Os aplicativos foram instalados nos tablets disponíveis no [MORA] e o questionário foi aplicado em uma área próxima à Universidade Federal de Uberlândia, em uma amostragem de 50 aplicações, somando-se casas e apartamentos.

A partir da aplicação do questionário, da observação dos alunos participantes do projeto e de sugestões recebidas pelos moradores que responderam à APO usando

o *Como Você Mora?*, foram identificados alguns pontos a serem melhorados visando a otimização e o bom funcionamento do aplicativo. Considerando os problemas relacionados às funcionalidades e as sugestões, podem-se citar:

- Problemas relacionados às telas:
 - Tempo da troca de tela entre perguntas superior a 3 segundos;
 - Barra deslizante apresentando travamento, a cada deslize as opções se modificam bem como as imagens correspondentes às opções;
 - Dificuldade de encontrar a opção utilizando a barra deslizante quando o conjunto de opções é superior a 6;
 - Teclado cobrindo a caixa de texto utilizada para a entrada de dados;
 - Pódio sem a possibilidade de arrastar componentes;
- Tempo total de resposta do questionário entre 20 min e 40 min (a depender da idade do entrevistado);
- Necessidade de uma nova seção para coletar dados sobre hábitos sustentáveis do morador;
- Possibilidade de encerrar o questionário e continuar posteriormente na próxima questão, sem ter que reiniciar o questionário;
- Problemas no envio do questionário ao Sistema APO Digital:
 - Erro na coleta e envio de respostas à perguntas contendo sub-perguntas (as respostas às sub-perguntas estavam sendo perdidas);
 - Diversas respostas ao questionário, sendo de moradores diferentes, poderiam estar armazenadas localmente e não estavam sendo corretamente diferenciadas no Sistema APO Digital.

No intuito de endereçar estas questões, foram realizadas modificações e atualizações no aplicativo, para a produção de uma nova versão. O trabalho de produção da nova versão será abordado no Capítulo 4.

4 Evolução da aplicação *Como Você Mora?*

4.1 Ferramentas de desenvolvimento

Conforme citado no Capítulo 2, o projeto foi desenvolvido seguindo-se o padrão de arquitetura MVP. Esse tipo de arquitetura possui a vantagem de separar as lógicas de negócio e persistência de dados das lógicas de visualização, gerando assim um código com maior facilidade de manutenção e adição de novos comportamentos. Basicamente, o MVP se divide em 3 camadas principais:

- **Model:** É a camada responsável por armazenar os dados. Além disso, lida com as regras de negócio e comunicação com as camadas de persistências de dados e rede.
- **View:** É a camada responsável por prover a visualização dos dados e todos os componentes visuais necessários. Além disso, acompanha a ação do usuário para para notificar o *Presenter*.
- **Presenter:** É a camada responsável por fornecer os dados de visualização e lidar com as ações realizadas na camada de *View*.

Além disso, foram utilizadas algumas bibliotecas externas, a fim de facilitar o desenvolvimento do projeto:

- **ButterKnife:** Facilita e otimiza o processo de recuperar a referência de um componente visual. Como por exemplo:

```
String teste = (String) findViewById(R.id.teste);
```

O uso do *ButterKnife* permite que este código seja reduzido a:

```
@BindView(R.id.teste) String teste
```

Assim, a *ButterKnife* permite vincular automaticamente as visualizações definidas no arquivo de layout XML do Android com as variáveis Java correspondentes no código-fonte. Isso elimina a necessidade de chamar manualmente o método `findViewById()` para cada visualização, tornando o código mais limpo e legível.

- **Retrofit:** Facilita a realização de solicitações de rede de forma simples e eficiente, sendo usada para realizar chamadas de API HTTP e converter as respostas em

objetos Java de forma automática. Simplifica o processo de transmitir e receber dados entre o app e um servidor, que faz o uso do *JSON*. O envio das respostas do questionário é feito através de uma *API*, sendo assim, esta biblioteca facilita a comunicação entre componentes.

- **Crashlytics:** permite a monitoração do app em caso de falhas, sendo uma ferramenta de relatório e rastreamento de erros para aplicativos Android. Fornece recursos para o registro de falhas e análise de exceções para auxiliar desenvolvedores a identificar e solucionar problemas em seus aplicativos.

4.2 Organização das Atividades

Para organizar o processo de evolução do software e priorizar o desenvolvimento da nova seção e das modificações e correções solicitadas, seguiu-se uma abordagem baseada em requisitos e valor agregado. Neste sentido, as atividades realizadas seguiram a seguinte ordem:

1. Inclusão da nova seção: codificação das telas sobre Hábitos Sustentáveis no aplicativo *Como Você Mora?*.
2. Armazenamento: programação da retomada do questionário após encerramento do aplicativo, sem perda das respostas já fornecidas.
3. Comunicação: aprimoramento do envio das respostas via API e criação de um identificador de usuário obtido via serviço POST.
4. Correções gerais: envio de respostas e sub-respostas, melhorias no fluxo.

Considerando-se o impacto de cada uma das atividades supracitadas no usuário final, na funcionalidade geral do sistema e nos objetivos da APO, a atividade de início, que foi a inclusão da nova seção de Hábitos Sustentáveis, permitiu que os pesquisadores pudessem validar as questões, o formato como foram criadas e o fluxo. Permitiu também a validação dos objetos visuais desenhados para a composição das telas.

Em seguida, as atividades de armazenamento, comunicação e correções gerais foram programadas de forma a entregar o máximo valor agregado com os recursos disponíveis e considerando-se as dependências entre as modificações.

4.3 Inclusão da Seção de Hábitos Sustentáveis

A nova seção do *Como Você Mora?* permite identificar o nível de consciência ambiental dos moradores, conhecer seus hábitos e atitudes na economia de água e energia,

| | Perguntas |
|----|--|
| 1 | Você economiza água? → <i>sim</i> – O que você faz para economizar água? → <i>sim</i> – Quais tipos de equipamentos economizadores de água você possui? → <i>não</i> – Por que você não usa esse tipo de equipamento? |
| 2 | Você economiza energia elétrica? → <i>sim</i> – O que você faz para economizar energia elétrica? |
| 3 | Você conhece a diferença entre as bandeiras tarifárias de energia elétrica? |
| 4 | Na sua habitação, há algum equipamento que faz proveito da energia solar? → <i>sim</i> – Quais? |
| 5 | Você separa o lixo entre reciclável e comum? |
| 6 | Você sabe para onde o lixo reciclável da sua cidade vai? |
| 7 | Você conhece algum Ecoponto próximo da sua residência? |
| 8 | Você separa o óleo dos outros tipos de resíduos? |
| 9 | O que você faz com resíduos de construção civil? |
| 10 | Como você realiza o descarte de baterias e equipamentos eletrônicos? |
| 11 | Como você realiza o descarte de medicamentos vencidos? |
| 12 | Você consome alimentos, aqueles produzidos com agricultura orgânica? |
| 13 | Quando você vai ao supermercado, como transporta as mercadorias compradas? |
| 14 | Você possui plantas em sua moradia? |
| 15 | Qual meio de transporte você utiliza no dia-a-dia? |

Tabela 1 – Principais questões da seção de Hábitos Sustentáveis

na coleta e destinação de resíduos, no consumo de alimentos orgânicos e na mobilidade urbana. A Tabela 1 apresenta as principais questões que compõem a seção.

No sistema de avaliação proposto, os tipos de respostas: podem ser dicotômicas (*sim/ não*), tricotômicas (*sim/ não/ não sei*), de múltipla escolha e com escala de diferencial semântico (uso de uma escala de valores de 5 pontos). Além das questões, foram propostos pontos informativos aos usuários respondentes, contendo esclarecimentos quanto aos hábitos questionados. O questionário completo correspondente à seção de Hábitos Sustentáveis pode ser encontrado no Anexo A.

Para cada uma das questões contidas no Anexo A, foi construída a tela correspondente. A cor de fundo para esta seção é a cor verde, e o personagem Dr. Prancheta foi utilizado como guia apresentando o enunciado da questão e informando como podem ser respondidas. As respostas foram coletadas utilizando-se o modelo de botões. A subseção seguinte mostra os passos para a construção de cada tela contendo uma questão.

4.3.1 Passos para a construção de uma tela

O primeiro passo para a construção de uma tela do questionário é construir seu layout em um arquivo *XML*, que descreve a estrutura e os elementos visuais. A Figura 9 mostra o exemplo de uma das telas do aplicativo:

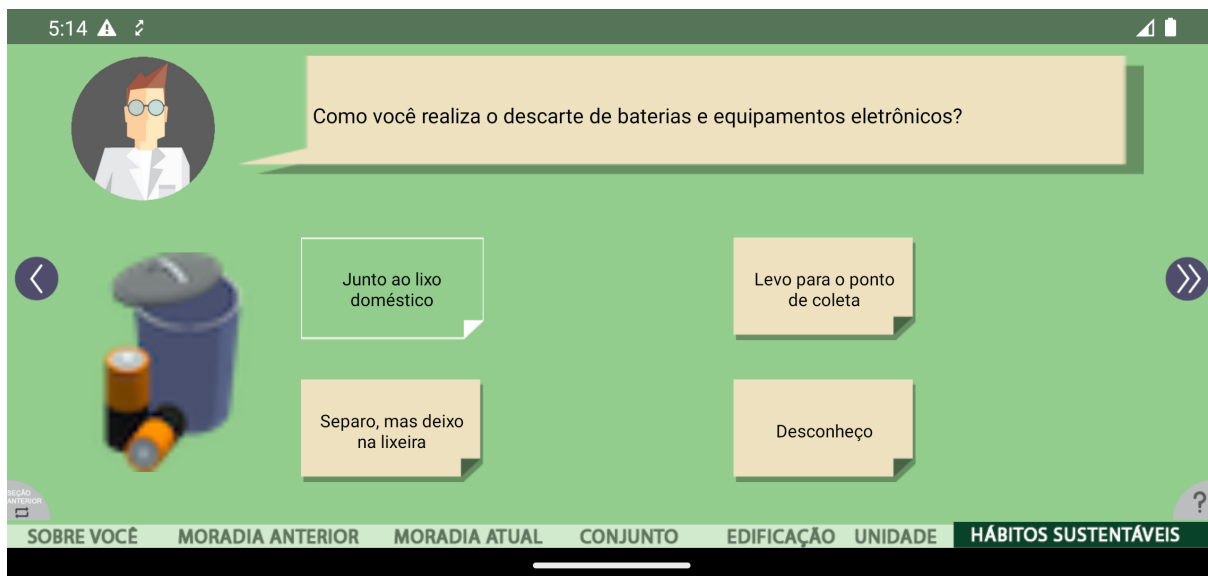


Figura 9 – Exemplo de resposta selecionada

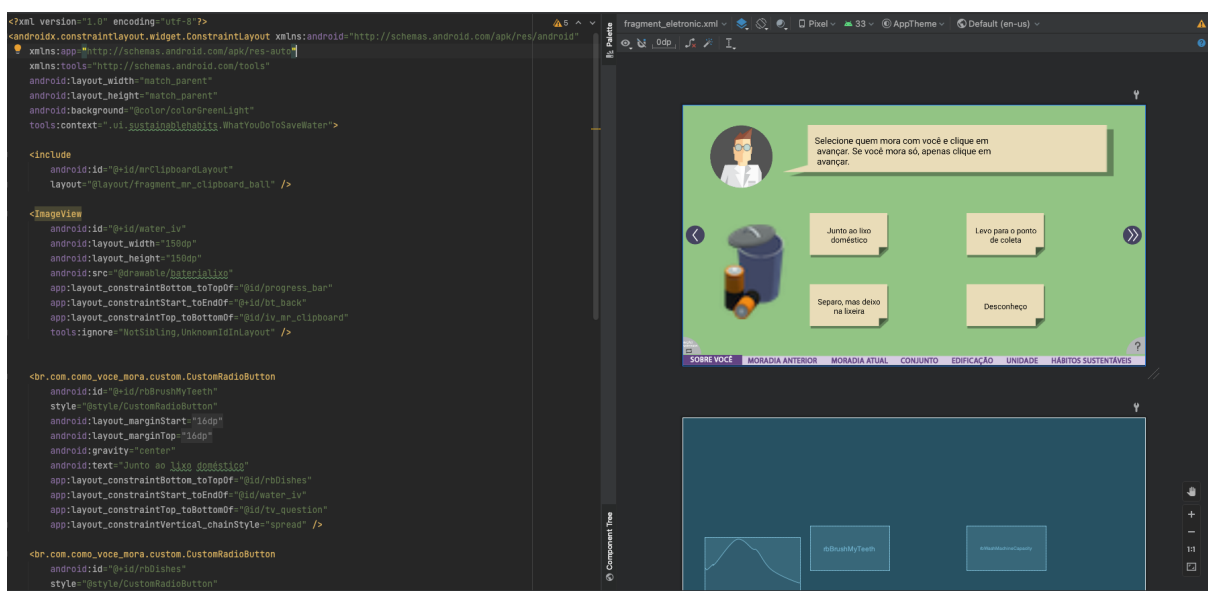


Figura 10 – Exemplo de layout de uma das telas do questionário

Como pode-se verificar na Figura 10, a construção do layout é feita utilizando componentes já disponibilizados pelo próprio framework Android. Neste exemplo, o layout começa com a tag *Constraint Layout*. Um layout é composto por um ou mais elementos visuais e é responsável por organizá-los. No caso do *Constraint Layout* os elementos são organizados por meio de *constraints*. Cada elemento visual possui 4 *constraints*: top, bottom, left, right que podem se posicionar da seguinte forma:

- layout_constraintLeft_toLeftOf : Alinha o lado esquerdo de um elemento com o lado esquerdo de outro
- layout_constraintLeft_toRightOf : Alinha o lado esquerdo de um elemento com o lado direito de outro
- layout_constraintRight_toLeftOf: Alinha o lado direito de um elemento com o lado esquerdo de outro
- layout_constraintRight_toRightOf: Alinha o lado direito de um elemento com o lado direito de outro
- layout_constraintTop_toTopOf: Alinha o lado de cima de um elemento com o lado de cima de outro

layout_constraintTop_toBottomOf: Alinha o lado de cima de um elemento com o lado de baixo de outro
layout_constraintBottom_toTopOf: Alinha o lado de baixo de um elemento com o lado de cima de outro
layout_constraintBottom_toBottomOf: Alinha o lado de baixo de um elemento com o lado de baixo de outro

Seguindo o *Constraint Layout* tem-se uma tag de *include* para definir um layout para interfaces de usuário flexíveis e responsivas. As restrições definem relações de posicionamento entre os elementos visuais, em vez de depender de hierarquias de layout aninhadas. Como o projeto do *Como você Mora?* possui inúmeras telas e alguns layouts se repetem, faz sentido criar apenas um e reutilizar em diferentes telas.

Seguindo nossa estrutura visual, temos uma tag de *ImageView* que é responsável por renderizar arquivos de imagens no nosso layout. Neste exemplo, ela é responsável por renderizar a imagem das pilhas com o lixo. E por fim, temos mais 4 tags do tipo *CustomRadioButton*, que são responsáveis por renderizar as perguntas. No Android, temos diversos *widgets* para montarmos um layout, mas quando surge a necessidade de um *widget* especial, é necessário a criação de uma *Custom View*, onde conseguimos estender um *widget* padrão e personalizarmos da nossa forma. Neste exemplo, o layout das perguntas é definido da seguinte forma:

```
public class CustomRadioButton extends CheckBox implements
    CompoundButton.OnCheckedChangeListener {
    public CustomRadioButton(Context context) {
        super(context);
        init();
    }

    public CustomRadioButton(Context context, AttributeSet attrs) {
        super(context, attrs);
        init();
    }

    public void init() {
        setBackground(getResources().getDrawable(R.drawable.bg_radio_button_normal));
        setOnCheckedChangeListener(this);
    }

    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        updateView();
    }

    public void updateView() {
        setBackground(getResources().getDrawable(isChecked() ?
            R.drawable.bg_radio_button_selected :
            R.drawable.bg_radio_button_normal));
    }
}
```

Neste caso, quando uma resposta é selecionada (clorada pelo usuário), seu background se altera, conforme mostrado na Figura 9, em que a resposta “Junto ao lixo

doméstico” foi marcada.

Após a criação do arquivo *XML* contendo o layout, é criado o componente *Fragment*, que é responsável por exibir esse layout. Além disso, definem-se ações, lógicas e valores que os elementos visuais terão. No exemplo a seguir, é mapeada a ação de clique no componente visual baseado no identificador definido no arquivo *XML*.

```
@OnClick(R.id.bt_next)
public void onBtNextClicked() {
    if (anyOptionChecked) {
        if (getActivity() != null) {
            ResearchFlow.addAnswer(answerRequest, fragment: this);
            ((AboutYouActivity) requireActivity()).addFragment(mNextFrag);
        }
    }
}
```

Figura 11 – Exemplo de mapeamento de ações

Na Figura 11 pode-se verificar que as respostas de cada tela são armazenadas em um *Singleton* chamado de *ResearchFlow*. Esse *Singleton* é responsável por armazenar as respostas de cada tela. No final do questionário, são enviadas para um *API* todas as respostas obtidas durante o questionário.

4.4 Armazenamento

Um dos desafios do projeto era a possibilidade do usuário começar a responder o questionário e terminá-lo em outro momento. Para isso, foi utilizado o *SQLite* para o armazenamento local e temporário das respostas. Sendo assim, a cada pergunta que o usuário passa, a resposta é salva no banco de dados local.

Quando o usuário fecha o aplicativo e retorna, identifica-se no qual foi a última resposta e à qual pergunta ela pertencia. Assim, é possível redirecionar o usuário para a pergunta em que parou anteriormente.

4.5 Envio de respostas

O envio das respostas é a parte mais essencial do projeto, pois ela permite que a pesquisa seja realizada por qualquer pessoa em qualquer local que ela esteja, sem a necessidade de uma pessoa estar indo coletar os dados de pesquisas.

O envio das respostas foi feito através de uma *API*. Uma *API* é um conjunto de definições e protocolos usado no desenvolvimento e na integração de aplicações. Às vezes,

as APIs são descritas como um contrato entre um provedor e um usuário de informações, estabelecendo o conteúdo exigido pelo consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta) (RED-HAT, 2020).

Além do envio das respostas, é necessário uma forma de identificar a que usuário pertencem tais respostas. Sendo assim, para identificar um usuário foi utilizada uma API em que recupera-se um identificador para o usuário quando este abre o aplicativo e seleciona a opção de iniciar questionário. O identificador será utilizado para criar um bloco de respostas e enviar no momento final ao Sistema APO Digital, responsável pela manutenção de diversas APOs e respostas fornecidas por usuários moradores. Dessa forma, é possível analisar as respostas de um determinado usuário.

4.6 Correções Gerais

Após a finalização do desenvolvimento das perguntas dos hábitos sustentáveis, o grande desafio foi corrigir os problemas apontados pela equipe de pesquisadores da Faculdade de Arquitetura e Urbanismo. Um dos principais problemas que o aplicativo tinha era na comunicação das respostas para o Sistema APO Digital. Algumas perguntas, como exemplo aquelas das figuras 12 e 13, requerem que sejam enviadas mais de uma resposta para o servidor, visto que é necessário enviar a resposta afirmativa ou negativa acrescida das respostas complementares.

Para a resolução desse problema, foi necessária a utilização de uma estrutura de dados, chamada *Map*. Essa estrutura de dados serve para armazenar pares (chave, valor). Nesse caso, a chave seria (sim ou não) e os valores seriam as respostas posteriores.

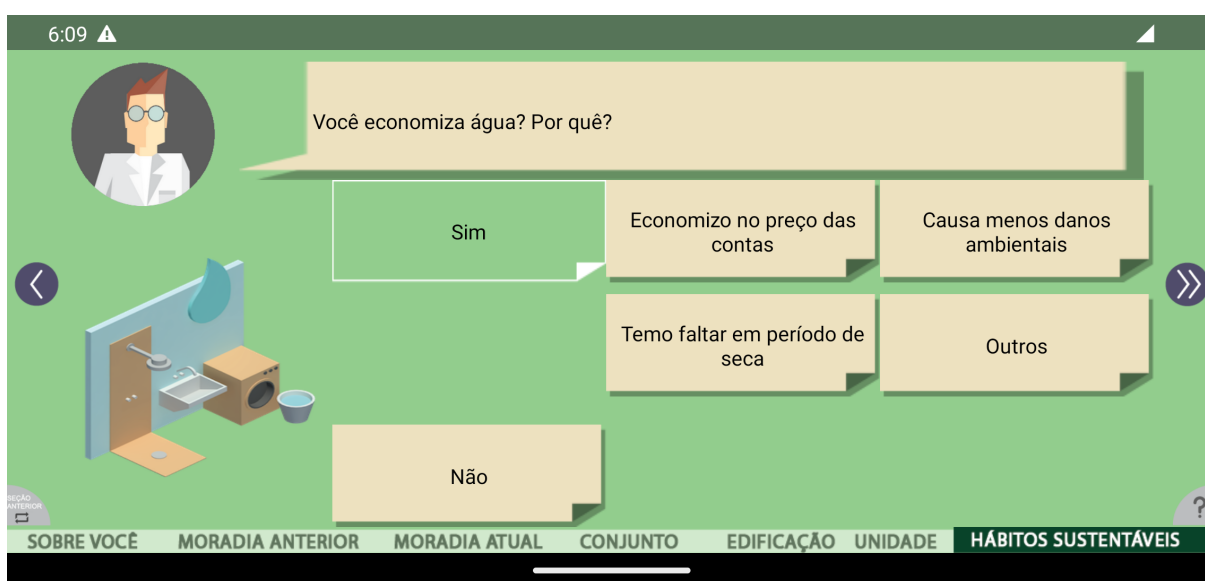


Figura 12 – Exemplo de múltipla resposta

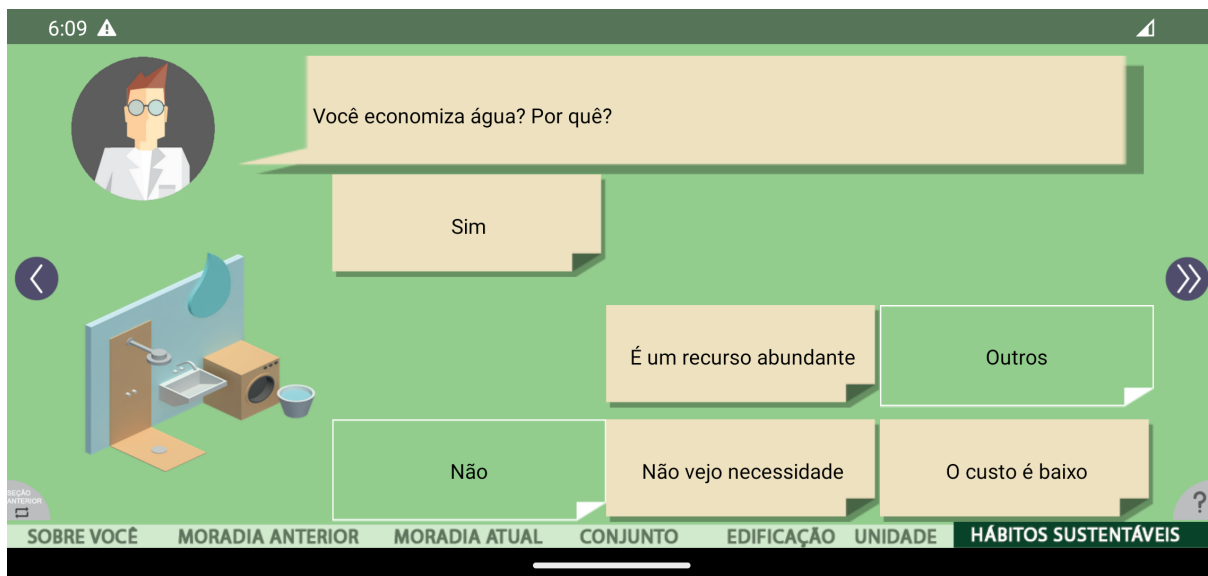


Figura 13 – Exemplo de múltipla resposta

Além das correções de comunicação do aplicativo com o servidor e de partes visuais, foi necessária a correção de problemas que fazem o aplicativo fechar inesperadamente. Esse tipo de falha, que denominamos de *crash*, podem ser visualizadas através da ferramenta *Crashlytics*, conforme mostrado na figura 14.

A ferramenta do *Crashlytics* permite que identifiquemos diferentes falhas que o aplicativo está tendo, de diferentes sessões e além disso, gera dados e informações sobre as versões que estão com problema, conforme mostra a Figura 14. Um exemplo disso, é o problema demonstrado na Figura 15, na qual conseguimos visualizar que o mesmo acontece na tela *UnityActivitiesByRoom*, mais especificamente na linha 97 e que estamos tentando acessar uma posição nula de uma lista. Após identificar o problema, conseguimos através da ferramenta de depuração do *Android Studio* corrigi-lo.

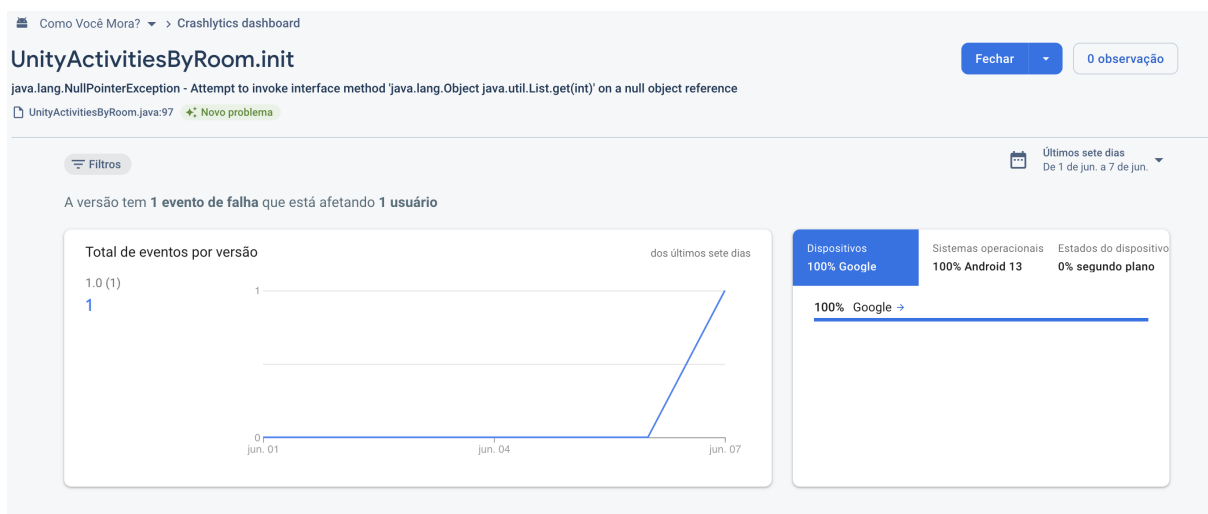


Figura 14 – Painel Crashlytics

Resumo do evento 1.0 (1) 13 Sdk_gphone64_arm64 7 de jun. de 2023, 10:04:56

Stack trace Chaves Registros Dados

TXT

Fatal Exception: java.lang.NullPointerException
Attempt to invoke interface method 'java.lang.Object java.util.List.get(int)' on a null object reference

br.com.como_voce_mora.ui.unity.UnityActivitiesByRoom.init (UnityActivitiesByRoom.java:97)
br.com.como_voce_mora.ui.BaseFragemnt.onCreateView (BaseFragemnt.java:28)
androidx.fragment.app.Fragment.performCreateView (Fragment.java:2963)
androidx.fragment.app.FragmentManager\$StateManager.createView (FragmentManager.java:518)
androidx.fragment.app.FragmentManager.moveToExpectedState (FragmentManager.java:282)
androidx.fragment.app.FragmentManager.executeOpsTogether (FragmentManager.java:2189)
androidx.fragment.app.FragmentManager.removeRedundantOperationsAndExecute (FragmentManager.java:2100)
androidx.fragment.app.FragmentManager.execPendingActions (FragmentManager.java:2002)
androidx.fragment.app.FragmentManager\$5.run (FragmentManager.java:524)
android.os.Handler.handleCallback (Handler.java:942)
android.os.Handler.dispatchMessage (Handler.java:99)
android.os.Looper.loopOnce (Looper.java:201)
android.os.Looper.loop (Looper.java:288)
android.app.ActivityThread.main (ActivityThread.java:7872)
java.lang.reflect.Method.invoke (Method.java)
com.android.internal.os.RuntimeInit\$MethodAndArgsCaller.run (RuntimeInit.java:548)
com.android.internal.os.ZygoteInit.main (ZygoteInit.java:936)

Firebase Background Thread #2 jdk.internal.misc.Unsafe.park

CleanupReference java.lang.Object.wait

Firebase Background Thread #3 jdk.internal.misc.Unsafe.park

Figura 15 – Painel Crashlytics

5 Conclusão

5.1 Considerações Finais

Conclui-se que o propósito de transferir o código do aplicativo *Como Você Mora?*, desenvolvido inicialmente na plataforma Corona SDK, em linguagem Lua, para a plataforma do Android Studio, que é mais utilizada atualmente, foi uma decisão acertada. Além de ser transferido para uma plataforma mais atualizada, os mesmos componentes começaram a ter um desempenho muito melhor.

Com uma nova versão do aplicativo desenvolvida com o auxílio do Android Studio, a mesma pode ser publicada na Google Play, sem nenhum problema, por ter sido gerada na arquitetura de 64 bits, que é usada nos celulares atualmente, e exigida pela loja. Foi obtida uma melhor fluidez em toda a navegação pelas telas do aplicativo em qualquer celular, ao contrário da primeira versão, que em celulares mais antigos, travava muito ao final do questionário.

O objetivo deste trabalho foi atender às demandas de manutenção e complementação da segunda versão do *Como Você Mora?*. A inclusão da nova seção de Hábitos Sustentáveis permitiu que a APO ficasse mais completa podendo contribuir com a compreensão das ações que podem reduzir impactos ambientais por parte dos pesquisadores. O armazenamento local e a comunicação dos resultados dos questionários com o Sistema APO Digital foram aperfeiçoados, garantindo um funcionamento eficiente da APO.

Além disso, diversas correções gerais foram efetuadas, como alteração na ordem das perguntas do questionário para melhorar a navegação do usuário e o envio das respostas coletadas. Algumas perguntas também passaram a ser representadas por formatos diferentes, para incluir novas possibilidades de respostas que não estavam presentes na primeira versão.

5.2 Trabalhos Futuros

Para uma continuação deste projeto, seria interessante a migração do projeto para a linguagem *Kotlin*. A *Kotlin* foi definida pela Google como a linguagem principal para Android. Sendo assim, as novas bibliotecas e tutoriais serão todos baseados nessa nova linguagem, deixando por assim a linguagem Java obsoleta para Android.

Caso o projeto queira se expandir para além da plataforma Android, seria interessante uma nova versão do projeto em uma plataforma híbrida, como por exemplo, o *Flutter*. O *Flutter* é um framework open source desenvolvido pela Google para criar apli-

cativos multiplataforma compilados nativamente a partir de uma única base de código ([FLUTTER, 2020](#)). Sendo assim, com apenas um desenvolvimento teríamos um aplicativo tanto para Android quanto para iOS, expandindo o alcance de pessoas que a pesquisa conseguiria atingir.

Referências

- ABREU, A. de S. L. et al. Avaliação de pós-ocupação em ambientes residenciais utilizando interface interativa e dispositivos móveis. In: SBSI. *IV Workshop de Iniciação Científica em Sistemas de Informação (WICSI)*. Lavras, MG, Brasil, 2017. p. 9–12. Citado na página 17.
- APACHE, C. *Apache couchdb*. 2023. Online: visitado em 01/05/2023. Disponível em: <<https://couchdb.apache.org>>. Citado na página 15.
- ARNOLD, K.; GOSLING, J.; HOLMES, D. *The Java programming language*. Reading: Addison Wesley Professional, 2005. Citado na página 11.
- ENERGIA, O. *Consumo sustentável: o que é e qual sua importância para o futuro?* 2021. Disponível em: <<https://origoenergia.com.br/blog/consumo-consciente/consumo-sustentavel-o-que-e-e-qual-a-importancia/>>. Citado na página 7.
- FIREBASE. *Firebase Crashlytics*. 2023. Online: visitado em 01/05/2023. Disponível em: <<https://firebase.google.com/docs/crashlytics?hl=pt-br>>. Citado na página 16.
- FLUTTER. *Flutter*. Flutter Framework Community, 2020. Disponível em: <<https://flutter.dev/>>. Citado na página 35.
- HADJRI, K.; CROZIER, C. Post-occupancy evaluation: purpose, benefits and barriers. *Facilities*, v. 27, n. 1/2, p. 21–33, 2009. Disponível em: <<http://dx.doi.org/10.1108/02632770910923063>>. Citado na página 17.
- HIPP, R. D. *SQLite*. 2020. Online: visitado em 10/05/2022. Disponível em: <<https://www.sqlite.org/index.html>>. Citado 2 vezes nas páginas 14 e 20.
- HOWS, D.; MEMBREY, P.; PLUGGE, E. *Introdução ao MongoDB*. [S.l.]: Novatec Editora, 2019. Citado na página 15.
- MARCELINO, R. A. *Aplicativo Android "Como Você Mora?"*. 2021. <https://repositorio.ufu.br/handle/123456789/32203>. Online: visitado em 30/03/2023. Citado 4 vezes nas páginas 17, 18, 19 e 20.
- ORACLE. *Java Documentation*. <<https://docs.oracle.com/en/java/index.html>>. Online: visitado em 02/05/2023. Citado na página 11.
- PEREIRA, M. H. de B. et al. Aplicação e gerenciamento de avaliações de pós-ocupação usando interfaces digitais. In: TECHWEEK UFU. *Anais da IV FACOM Techweek*. Uberlândia, MG, Brasil, 2017. p. 30–35. Citado 2 vezes nas páginas 17 e 18.
- PREISER, W.; WHITE, E.; RABINOWITZ, H. *Post-Occupancy Evaluation (Routledge Revivals)*. Routledge, 2016. (Routledge Revivals). ISBN 9781138888326. Disponível em: <<https://books.google.com.br/books?id=mn9NvgAACAAJ>>. Citado na página 17.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software-9*. [S.l.]: McGraw Hill Brasil, 2021. Citado 2 vezes nas páginas 11 e 15.

RED-HAT. *API REST*. Red Hat Enterprise, 2020. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Citado na página 31.

SILVEIRA, I. *Os pilares da qualidade de software e como garanti-los*. [S.l.], 2019. Disponível em: <<https://medium.com/@isabellasilveira/os-pilares-da-qualidade-de-software-e-como-garanti-los-620e9c626e8c>>. Citado na página 7.

SOMMERVILLE, I. *Engenharia de software*. Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>. Citado 2 vezes nas páginas 15 e 16.

STATISTA. *Global market share held by mobile operating systems from 2009 to 2023*. 2023. Disponível em: <<https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>>. Acesso em: 13 jul. 2023. Citado na página 10.

VILLA, S. B. et al. Tecnologia e interfaces digitais para avaliação pós-ocupação de edifícios de apartamentos: Possibilidades metodológicas. *Blucher Engineering Proceedings*, v. 2, n. 2, p. 316 – 328, 2015. ISSN 2357-7592. Disponível em: <www.proceedings.blucher.com.br/article-details/20539>. Citado 3 vezes nas páginas 17, 18 e 22.

VILLA, S. B.; ORNSTEIN, S. W. *Qualidade ambiental na habitação - avaliação pós-ocupação*. Oficina de Textos, 2016. ISBN 9788579752070. Disponível em: <<https://books.google.com.br/books?id=MrgWDAAAQBAJ>>. Citado 2 vezes nas páginas 7 e 17.

ZHANG, Y.; LUO, Y. An architecture and implement model for model-view-presenter pattern. In: *2010 3rd International Conference on Computer Science and Information Technology*. [S.l.: s.n.], 2010. v. 8, p. 532–536. Citado na página 13.

Anexos

ANEXO A – Seção de Hábitos Sustentáveis do *Como Você Mora?*

O conceito de sustentabilidade é complexo, pois atende a um conjunto de variáveis interdependentes, em diferentes dimensões: sustentabilidade social, sustentabilidade econômica, sustentabilidade ecológica, sustentabilidade espacial e a sustentabilidade cultural. Este anexo apresenta a seção de Hábitos Sustentáveis da APO *Como Você Mora?*, considerando casas e apartamentos. O intuito desta seção é o de propor ao usuário morador perguntas que permitam refletir e coletar informações para que se identifique o papel dos moradores na redução dos impactos ambientais, analisando-se seus hábitos e ações na separação do lixo, uso dos recursos de energia, água, alimentos orgânicos, emissão de gases na atmosfera, reciclagem entre outros.

Considerando o recurso água, as questões avaliam como o usuário faz uso da água em sua moradia para identificar o nível de consciência ambiental das pessoas e levantar as estratégias adotadas para economia e uso racional da água. Deseja-se conhecer quais dos equipamentos que auxiliam na economia de água são mais usados (torneiras econômicas, restritores de vazão de água, válvulas de bacia sanitária com duplo acionamento, entre outros). No uso da energia elétrica, são colocadas questões que coletam informações sobre o uso de fonte de energia alternativa, procedimentos de instalação, custo e vantagens percebidas.

Quanto à emissão de gases estufa, as questões abordam os meios de transporte usados no dia-a-dia (como ônibus, carro, metrô e outros), considerando distâncias de até de cinco quarteirões, entre 5 e 10 quarteirões e acima de 10 quarteirões. Para o consumo de alimentos, as questões focam no consumo de alimentos orgânicos, e no cultivo de plantas em casa. As páginas a seguir contêm a descrição do questionário que foi confeccionado para a atualização do aplicativo.

6. HÁBITOS SUSTENTÁVEIS

a. Você economiza água? Por quê? () Sim () Não

Se sim: () Economizo no preço das contas () Temo faltar água em períodos de seca () Causa menos danos ambientais () Outros

Se não: () Não vejo necessidade () O custo é baixo () É um recurso abundante () Outros

Você sabia? Conteúdo: aparelhos economizadores de água.

b. O que você faz para economizar água? (Se necessário, marque mais de uma opção)

| | | |
|--|---|---------------------------------|
| <input type="checkbox"/> Escovo os dentes com a torneira fechada | <input type="checkbox"/> Utilizo a máquina de lavar com a capacidade máxima | |
| <input type="checkbox"/> Ensaboo a louça com a torneira fechada | <input type="checkbox"/> Reutilizo água da máquina de lavar | |
| <input type="checkbox"/> Tomo banhos rápidos | <input type="checkbox"/> Posso aparelhos que economizam água | <input type="checkbox"/> Outros |

c.1. (Se marcar que possui aparelhos que economizam água) Quais tipos de equipamentos economizadores de água você possui?

| | | |
|---|---|---|
| <input type="checkbox"/> Máquina de lavar com reuso | <input type="checkbox"/> Descarga com duplo acionamento | <input type="checkbox"/> Torneira econômica |
| <input type="checkbox"/> Caixa d'água para reuso | <input type="checkbox"/> Vaso com caixa acoplada | <input type="checkbox"/> Outros |

c.2 (Se não marcar que possui aparelhos que economizam água) Por que você não usa esse tipo de equipamento (painéis fotovoltaicos ou placas solares)?

| | | |
|-------------------------------------|---|---|
| <input type="checkbox"/> Desconheço | <input type="checkbox"/> Instalação depende de reformas | <input type="checkbox"/> Não vejo necessidade |
| <input type="checkbox"/> Custo alto | <input type="checkbox"/> Dificil manutenção | |

Você sabia? Conteúdo: Desperdício de água

Você sabia? Conteúdo: Utilização da máquina de lavar roupas na capacidade máxima

Você sabia? Conteúdo: Gasto de água de uma máquina de lavar roupas

d. Você economiza energia elétrica? Por quê? () Sim () Não

Se sim: () Economizo no preço das contas () Temo faltar energia () Causa menos danos ambientais () Outros

Se não: () Não vejo necessidade () O custo é baixo () É um recurso abundante na natureza () Outros

Você sabia? Conteúdo: eletrodomésticos com selo A

Você sabia? Conteúdo: Desligar os aparelhos que não estão em uso

e. O que você faz para economizar energia elétrica? Pode marcar mais de uma opção.

| | | |
|---|---|---------------------------------|
| <input type="checkbox"/> Apago as luzes ao sair do ambiente | <input type="checkbox"/> Compra eletrodomésticos com selo A | <input type="checkbox"/> Outros |
| <input type="checkbox"/> Desligo os aparelhos sem uso | <input type="checkbox"/> Utilizo lâmpadas econômicas (como fluorescentes e LED) | |

f. Você sabe qual é a diferença entre as bandeiras tarifárias na conta de energia elétrica?

() Sim () Ouvi falar, mas não sei () Desconheço

Você sabia? Conteúdo: bandeiras tarifárias

Você sabia? Conteúdo: placas e painéis solares

g. Na sua habitação, há algum equipamento que faz proveito da energia solar?

Placas solares Painéis fotovoltaicos Ambos os sistemas Nenhum dos dois sistemas

h. Se não ou apenas um dos sistemas: Por qual motivo você não faz uso desse tipo de equipamento (painéis e placas)?

| | | |
|--|---|-------------------------------------|
| <input type="checkbox"/> Custo alto | <input type="checkbox"/> Não vejo vantagem | <input type="checkbox"/> Desconheço |
| <input type="checkbox"/> Depende de reformas | <input type="checkbox"/> Dificil manutenção | <input type="checkbox"/> Outros |

i. Você separa o lixo entre reciclável e comum? Sim Não

Se sim: Para causar menos problemas ao meio ambiente Pois há coleta seletiva no meu bairro Para doar aos catadores de lixo Outros

Se não: Pois não há coleta seletiva no meu bairro Não vejo necessidade Pois acho difícil separar no dia-a-dia Pois não sei que tipo de lixo é reciclável Outros

j. Você sabe para onde o lixo reciclável da sua cidade vai? Sim Não

INFORMAÇÃO: conheça os pontos de coleta seletiva em sua cidade

k. Você conhece algum Ecoponto próximo da sua residência? Sim Não

INFORMAÇÃO: conheça os ecopontos em sua cidade

l. Você separa o óleo dos outros tipos de resíduos? Por quê? Sim Não

Se sim: Causa menos danos ambientais Reutilizo o óleo para outros fins Levo até os locais de coleta Outros

Se não: Não vejo necessidade Dificil separar no dia-a-dia Não sei onde posso levar Não Utilizo Outros

m. O que você faz com resíduos de construção civil?

Jogo em algum terreno vazio Levo para um ecoponto Jogo em aterro sanitário Não faço nada Não gero resíduos Uso caçamba

n. Como você realiza o descarte de baterias e equipamentos eletrônicos?

Junto com o lixo doméstico Separo, mas deixo na lixeira Levo para o ponto de coleta Desconheço

Você sabia? Conteúdo: descarte correto de pilhas e baterias

o. Como você realiza o descarte de medicamentos vencidos?

Junto com o lixo doméstico Separo, mas deixo na lixeira Levo para o ponto de coleta Não utilizo Doação antes do vencimento

Você sabia? Conteúdo: descarte correto de medicamentos vencidos

p. Você consome alimentos, aqueles produzidos sem substâncias químicas (como agrotóxicos ou hormônios)?

Sim Não

p1. Se sim: Preocupação saúde Reduzir danos ambientais São mais saborosos

p2. Se não: Preços altos Dificuldade de encontrar Não vejo necessidade

n. Quando você vai ao supermercado, como transporta as mercadorias compradas?

Bolsa reutilizável Caixa de papelão Sacola plástica Outros

Universidade Federal de Uberlândia - Faculdade de Arquitetura e Urbanismo e Design
[MORA] PESQUISA EM HABITAÇÃO

o. Você possui plantas em sua moradia? () Sim () Não

Se sim: () Decoração (acho bonito) () Gosto de cuidar () Consumo próprio () Melhora a qualidade do ar () Outros

Se não: () Não tenho tempo () Não sei cuidar () Não acho necessário () Não tenho espaço () Outros

p. Se Sim: Quais tipos de plantas você possui? () Decorativa () Hortaliça/ Comestível () Medicinal

q. Qual meio de transporte você utiliza nos casos abaixo?

| | A pé | Bicicleta | Carro ou moto | Transporte coletivo |
|--|------|-----------|---------------|---------------------|
| No dia-a-dia, até cinco quarteirões | | | | |
| No dia-a-dia, de cinco a dez quarteirões | | | | |
| No dia-a-dia, acima de dez quarteirões | | | | |
| Mais utilizado no geral | | | | |

r. Em relação ao meio de transporte que você mais utiliza, explique sua escolha indicando se necessário mais de uma resposta.

| | | |
|--------------------------------------|-----------------------------------|------------------------|
| () Rapidez de locomoção | () Consigo ir a todos os lugares | () Divido com outras |
| () Custo é mais acessível | () Preocupação com a saúde | () Necessidade física |
| () Causa menos problemas ambientais | () Outros | |