

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO SUL CAMPUS RESTINGA

LUIZ FELIPE VARGAS DA SILVA

ESTUDO DE PLATAFORMAS LOW-CODE PARA DESENVOLVIMENTO DE
APLICATIVOS PARA DISPOSITIVOS MÓVEIS

Porto Alegre
2023

LUIZ FELIPE VARGAS DA SILVA

ESTUDO DE PLATAFORMAS LOW-CODE PARA DESENVOLVIMENTO DE
APLICATIVOS PARA DISPOSITIVOS MÓVEIS

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Campus Restinga, como requisito parcial para a Obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Jean Carlo Hamerski

Porto Alegre
2023

LUIZ FELIPE VARGAS DA SILVA

ESTUDO DE PLATAFORMAS LOW-CODE PARA DESENVOLVIMENTO DE
APLICATIVOS PARA DISPOSITIVOS MÓVEIS

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Campus Restinga, como requisito parcial para a Obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Porto Alegre, 13 de Setembro de 2023.

BANCA EXAMINADORA

Prof. Dr. Jean Carlo Hamerski (Orientador)
IFRS - Campus Restinga

Prof. Dr. Gleison Samuel do Nascimento (Membro da Banca)
IFRS - Campus Restinga

Prof. Dr. Rafael Pereira Esteves (Membro da Banca)
IFRS - Campus Restinga

Dedico este trabalho aos meus pais e amigos que sempre me incentivaram.

AGRADECIMENTOS

Agradeço a Deus por te chegado ate aqui, e aos meus pais que sempre me incentivaram, e pela oportunidade de estudar na instituição e concluir o ensino superior.

RESUMO

Com o aumento da demanda na pandemia por profissionais capacitados na área da TI, aumentaram a procura por trabalhos remotos. As empresas começaram a contratar pessoas com pouco ou nenhum conhecimento em programação, para suas equipes de desenvolvimento. As plataformas low-code são alternativas para que programadores sem muita experiência de programação consigam desenvolver sistemas de pequeno e médio porte. Elas permitem que as empresas possam automatizar processos e criar novas funcionalidades para sistemas que já utilizam. Este trabalho de conclusão de curso apresenta o estudo realizado sobre plataformas low-code, especificamente as plataformas Kodular e App Inventor, buscando compreender e analisar este paradigma da programação em diferentes aspectos e perspectivas, além de realizar um estudo de caso no desenvolvimento de um aplicativo para dispositivos móveis, permitindo assim comparar as funcionalidades das plataformas estudadas. Ao final deste trabalho de conclusão de curso espera-se ter contribuído para que programadores com pouco conhecimento em linguagem de programação tenham subsídios técnicos para desenvolver um aplicativo mobile de pequeno porte.

Palavras-chave: Low-code. App Inventor, Kodular, Thinkable.

ABSTRACT

<Traduzir novamente> With the increase in demand in the pandemic for trained professionals in the IT area, the demand for remote jobs increased. Companies started hiring people with little or no programming knowledge for their development teams. Low-code platforms are alternatives for programmers without much programming experience to develop small and medium-sized systems. They allow companies to automate processes and create new features for systems they already use. This course conclusion work presents the study carried out on low-code platforms with the objective that programmers with little knowledge in programming language have technical subsidies to develop a small mobile application. As a case study, a Low-code system was developed. Kodular, App Inventor, Thunkable. At the end of this course conclusion work, it was possible to carry out a study on these low-code platforms, seeking to understand and analyze this programming paradigm in different aspects and perspectives, in addition to carrying out a case study in the development of an application for mobile devices, thus allowing to compare the functionalities of the studied platforms.

Keywords: Low-code. App Inventor, Kodular, Thunkable.

LISTA DE FIGURAS

Figura 1 — Número de documentos por área de estudo	17
Figura 2 — Motivos para a adoção do Low code	18
Figura 3 — Diagrama Entidade e Relacionamento	25
Figura 4 — Tela de cadastro de Usuário	28
Figura 5 — Tela de Login do Usuário	29
Figura 6 — Tela de Login Senha Incorreta	30
Figura 7 — Tela Inicial do Aplicativo	31
Figura 8 — Tela de Cadastro de Despesa	32
Figura 9 — Tela do Campo Despesa Valor Categoria	33
Figura 10 — Tela de Cadastro de Receita	34
Figura 11 — Tela Campo Receita Valor Categoria	35
Figura 12 — Tela de Listar Excluir Editar e Cancelar	36
Figura 13 — Tela de Confirmação Excluir	37

LISTA DE TABELAS

Tabela 1 — Benefícios e vantagens do aprendizado Low Code para a formação profissional	16
Tabela 2 — Tabela comparativa entre as ferramentas Low-Code	21
Tabela 3 — Descrição das Histórias de Usuário	22
Tabela 4 — Descrição Requisitos Funcionais do Sistema	24
Tabela 5 — Descrição dos Requisitos Não Funcionais do Sistema	25

LISTA DE ABREVIATURAS E SIGLAS

ROI Retorno Sobre Investimento

SUMÁRIO

1	INTRODUÇÃO	11
1.1	JUSTIFICATIVA	12
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECIFICOS	14
1.4	ORGANIZAÇÃO DO TRABALHO	15
2	TRABALHOS CORRELATOS	16
2.1	A CONTRIBUIÇÃO DO LOW-CODE NO ÂMBITO EDUCACIONAL UM MAPEAMENTO SISTEMÁTICO DA LITERATURA	16
2.2	APLICAÇÃO DE FERRAMENTAS LOW-CODE PARA MELHORIA E AUTOMAÇÃO DE PROCESSOS EM UMA EMPRESA DE CONTABILIDADE	17
2.3	DESENVOLVIMENTO LOW-CODE APLICADA À GESTÃO DE PROJETOS: UM ESTUDO DE CASO DA CONSTRUÇÃO DE UM DASHBOARD PARA A FASE INICIAL DE UM ESCRITÓRIO DE GESTÃO DE PROJETOS	18
3	FERRAMENTAS DE DESENVOLVIMENTO LOW-CODE	19
3.1	APP INVENTOR	19
3.2	KODULAR	20
3.3	THUNKABLE	20
3.4	COMPARATIVO ENTRE AS FERRAMENTAS	21
4	SOLUÇÃO CONCEITUAL	22
4.1	HISTÓRIAS DE USUÁRIO	22
4.2	REQUISITOS FUNCIONAIS	23
4.3	REQUISITOS NÃO FUNCIONAIS	24
4.4	DIAGRAMA ENTIDADE/RELACIONAMENTO	25
5	ESTUDO DE CASO	27
5.1	TELA DE CADASTRO DE USUÁRIO	27
5.2	TELA DE LOGIN DO USUÁRIO	28
5.3	TELA DE LOGIN SENHA INCORRETA	29
5.4	TELA INICIAL DO APLICATIVO	30
5.5	TELA DE DESPESA CATEGORIA	31
5.6	TELA DE DESPESA CAMPO VAZIO CATEGORIA	32
5.7	TELA DE RECEITA CATEGORIA	33
5.8	TELA DE RECEITA CAMPO VAZIO CATEGORIA	34
5.9	TELA DE LISTA EXCLUIR, EDITAR E CANCELAR	35
5.10	TELA DE CONFIRMAR EXCLUIR	36
6	DISCUSSÃO TÉCNICA	38
7	CONCLUSÃO E TRABALHOS FUTUROS	40
	REFERÊNCIAS BIBLIOGRÁFICAS	41

1 INTRODUÇÃO

Com a pandemia de Covid-19, veio a revolução digital e um interesse crescente por parte de empresas, de diversos segmentos e portes, em desenvolver aplicativos e sistemas para atender à alta demanda gerada pelo período de confinamento. Uma das tecnologias que se destacou nesse período foram as plataformas low-code, que ganharam adesão em todo o mundo, pois dispensam conhecimento avançado em codificação.

As empresas começaram a aderir a este tipo de tecnologia de modo a suprir as demandas do mercado de trabalho incorporadas no mundo todo durante o crescimento da pandemia. Tais plataformas concentram-se em apresentar ao profissional de tecnologia da informação uma alternativa de programação em mais alto nível, dispensando conhecimentos aprofundados em codificação propriamente dita.

Neste contexto, considerando em que as plataformas de programação low-code tornaram-se viáveis ao mercado de trabalho, a fim de comportar a demanda exigida durante a pandemia, compreende-se que tratam-se de paradigmas de programação promissores.

As plataformas de programação low-code podem ser oportunidades para quem ainda não domina programação em nível de mercado de trabalho profissional mas deseja atuar na área de tecnologia. As diversas ferramentas que implementam as soluções em low-code mostram-se bem estruturadas facilitando ainda mais o processo de desenvolvimento. Salienta-se ainda que a perspectiva é que hajam resultados com menos tempo e mão de obra especializada, onde um desenvolvedor(a) com conhecimentos específicos de programação restritos ou limitados consiga criar um sistema pequeno em pouco tempo.

Dados publicados pela consultoria Gartner (GARTNER, 2022) projetam que, até 2024, mais de 65% dos softwares e aplicativos serão desenvolvidos em low-code, com um crescimento médio de 40% ao ano. Segundo relatório da Gartner, este mercado movimentou cerca de US\$ 13,3 bilhões (R\$ 72,67 bilhões) em 2021 a nível mundial, uma alta de 22,6% em relação a 2020, quando foi alcançado a cifra de US\$ 11,2 bilhões (R\$ 58,98 bilhões).

Dentre as plataformas low-code mais conhecidas estão as plataformas App Inventor e Kodular. A plataforma App Inventor é uma ferramenta de programação baseada em blocos lógicos, desenvolvida pela Google que permite a criação de aplicativos para smartphones que rodam no sistema operacional Android, sem que seja necessário conhecimentos profundos de programação. A plataforma Kodular é gratuita e oferece ferramentas e suporte para o usuário com o objetivo de facilitar o processo de criação de aplicativo. A partir de blocos lógicos de programação, essa

ferramenta permite a criação de apps sem a necessidade de conhecimentos aprofundados sobre programação. Ele conta com um layout simples e intuitivo, sendo necessário poucos cliques ou arrastar alguns ícones e pronto! Sua própria ideia começa tomar a forma de um aplicativo para sistema Android.

O presente trabalho de conclusão de curso tem por objetivo realizar um estudo sobre essas plataformas low-code, especificamente as plataformas Kodular e App Inventor, buscando compreender seus aspectos e funcionalidades, e, como estudo de caso, desenvolver um aplicativo para dispositivos móveis que permite acessar e apresentar os dados estatísticos de forma gráfica ao usuário, fazendo-se uso das vantagens dessas plataformas.

1.1 JUSTIFICATIVA

Visando justificar este trabalho foi feita uma pesquisa com 14 perguntas e realizada com 53 estudantes dos cursos vinculados à área de informática do IFRS Campus Restinga, buscando entender se os estudantes conhecem essas plataformas de desenvolvimento, e se as utilizam em sala de aula ou no trabalho. Na pesquisa realizada, identificou-se que 44% dos estudantes não conhecem as plataformas de desenvolvimento low-code, o que pode justificar a realização de um TCC que explore este tema. Abaixo seguem alguns dados compilados a partir da pesquisa realizada:

- 54% das pessoas trabalham com tecnologia da informação, e 46% das pessoas não trabalham;
- 22% das pessoas são estagiários, 22% são técnicos em suporte, 22% são desenvolvedores, 19% são estudantes, 10% são técnicos em informática, 5% são desenvolvedores/mobile;
- 38% trabalham de três a quatro anos com TI, 28%, trabalham de um a dois anos no setor de TI, 21% trabalham apenas seis meses na área da TI, de cinco a seis anos 3%, de sete a oito anos 3%, e a mais de dez anos 7%;
- 46% não utilizam plataformas de desenvolvimento, e 44% não conhecem essas plataformas de desenvolvimento, e 4% conhecem e só utilizam plataformas low-code, e 6% conhecem e utilizam os dois tipos de plataforma de desenvolvimento;
- 83% não utilizam nenhum tipo de aplicativo feito por essas plataformas, e os outros 17% já utilizam aplicativos feitos por essas plataformas;
- 89% não utilizam esse tipo de plataforma em ambiente profissional, e os outros 11%, já utilizam;
- dentre os que já utilizam essas plataformas, 56% utilizam há três ou quatro

anos, e os outros 44% das pessoas utilizam há seis meses;

- 76%, das pessoas, acreditam que essas plataformas facilitam no processo de desenvolvimento, e as outras 24%, acreditam que não facilita no processo de desenvolvimento;
- 62%, das pessoas acreditam essas plataformas podem contribuir para falta de mão de obra, e os outros 38% acreditam que essas plataformas não contribuem para falta de mão de obra;
- 73% das pessoas acreditam que essas plataformas podem gerar mais oportunidades de empregos, e os outros 27%, das pessoas acreditam que essas plataformas não vão gerar mais oportunidade de empregos;
- 80% das pessoas acreditam que não vai causar nenhum tipo de desemprego, e os outros 20% das pessoas acreditam que podem causar algum tipo de desemprego;
- 47%, das pessoas não conhecem essas plataformas, 33% das pessoas desenvolvem o próprio código, e outros 20% acham que podem apresentar algum tipo de limitação;
- 69% das pessoas utilizariam aplicativos feitos por plataformas low-code, e os outros 31% das pessoas não utilizariam aplicativos desenvolvidos por plataformas low-code.

As vantagens de se utilizar essas plataformas para desenvolvimento, em comparação com os métodos de programação tradicional, são várias, detalhadas a seguir:

- Agilidade: com a simplificação da quantidade de codificação manual, é possível entregar aplicativos e softwares em um menor tempo. Com isso, as plataformas vão de encontro aos métodos de Desenvolvimento Ágil, como Scrum e Extreme Programming, acelerando a gestão de projetos de TI;
- Flexibilidade: o sistema modular permite uma alteração de código mais flexível, capaz de remover, de forma mais ágil, o que estiver obsoleto ou não corresponder às necessidades atuais da empresa e substituir por inovações mais adequadas;
- Redução de custos: redução de custos em recrutamento e manutenção de profissionais de TI altamente qualificados, mantendo uma equipe menor e focada em funções mais complexas, sem ser interrompida para realizar codificações a cada nova demanda da empresa;
- Menos necessidade de capacitação: os profissionais não precisam ser frequentemente capacitados para acompanhar as inovações (que são constantes quando falamos em tecnologia), já que a programação será mais simples;

- Soluções mais criativas: com low-code, os profissionais de diversas áreas da empresa conseguem testar suas ideias de maneira mais fácil, sem precisar de um desenvolvedor a todo o tempo. As equipes ganham em autonomia para prototipar e experimentar soluções para problemas diários;
- Segurança: processos robustos de segurança já vêm integrados pela codificação inicial dos módulos;
- Risco mínimo e maior ROI: as plataformas são facilmente personalizáveis e fornecem recursos integrados de plataforma cruzada e integração de dados. Como consequência, desenvolvedores e empresas podem focar nas atividades principais do negócio, em vez de dedicar esforços e tempo para identificar problemas técnicos. O investimento, com isso, garante maior retorno financeiro;
- Aumento na produtividade: nas equipes de TI, a possibilidade de utilizar os modelos gráficos predefinidos gera ganho em tempo e aumenta a produtividade;
- Escalabilidade: as plataformas permitem a integração de diferentes funcionalidades, possibilitando a criação de soluções para empresas com estruturas mais complexas;
- Fast deployment: a facilidade no desenvolvimento permite uma implementação das aplicações de forma muito mais rápida. Ajustes após o lançamento podem ser feitos com um único clique;
- Foco no cliente e nos objetivos do negócio: possibilidade de atender mais facilmente os interesses e necessidades do cliente final ou do usuário da plataforma, uma vez que ela já resolve nativamente desafios inerentes a soluções de TI, como escalabilidade da solução, segurança de informação, fast deploy e outros.

1.2 OBJETIVO GERAL

Realizar um estudo sobre plataformas low-code buscando compreender e analisar este paradigma de programação em diferentes aspectos e perspectivas, além de realizar um estudo de caso no desenvolvimento de um aplicativo para dispositivos móveis, permitindo assim comparar as funcionalidades das plataformas estudadas.

1.3 OBJETIVOS ESPECIFICOS

Para alcançar o objetivo citado acima, este trabalho de conclusão de curso

definiu os seguintes objetivos específicos:

- Fazer uma pesquisa com a comunidade escolar para coletar dados sobre o tema;
- Compreender e analisar essas ferramentas em diferentes aspectos e perspectivas;
- Desenvolver um aplicativo para dispositivos móveis nas plataformas low-code Kodular e App Inventor.

1.4 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 apresenta os trabalhos correlatos ao presente trabalho de conclusão de curso. O capítulo 3 discorre sobre tecnologias utilizadas no trabalho e ferramentas. O capítulo 4 apresenta a solução conceitual da modelagem do aplicativo usado como estudo de caso. O capítulo 5 apresenta o estudo de caso realizado sobre as ferramentas App Inventor e Kodular. O capítulo 6 apresenta a discussão sob um ponto de vista técnico quanto à usabilidade das ferramentas de desenvolvimento. Por fim, o capítulo 7 apresenta a conclusão do trabalho.

2 TRABALHOS CORRELATOS

Este capítulo apresenta os trabalhos correlatos ao presente trabalho de conclusão de curso que investigam as plataformas low-code no contexto de desenvolvimento de sistemas e aplicativos mobile.

2.1 A CONTRIBUIÇÃO DO LOW-CODE NO ÂMBITO EDUCACIONAL UM MAPEAMENTO SISTEMÁTICO DA LITERATURA

De Souza (2022, p. 7) apresenta em seu trabalho de conclusão de curso um "mapeamento sistemático sobre o uso do Low Code no âmbito educacional, de forma que possa ser elucidado seu uso nos diferentes níveis educacionais, como é utilizado, quais benefícios para os futuros profissionais". Foram estudados 39 trabalhos, que levaram o autor a defender que as ferramentas low-code possuem utilização não só no ensino superior, mas também no ensino médio. Em cursos de treinamento e inovação, o uso de ferramentas low-code é bem coerente com sua proposta de desenvolvimento de aplicações. Outros usos foram identificados como base de apoio de curso de treinamento profissional, como introdução ao curso de desenvolvimento de algoritmos de Inteligência Artificial e inovação. A Tabela 1 mostra um conjunto de benefícios e vantagens identificados, e a quantidade de trabalhos encontrados no estudo realizado que abordam esses benefícios.

Tabela 1 — Benefícios e vantagens do aprendizado Low Code para a formação profissional

Benefícios e Vantagens	Quantidade de trabalhos
Produtividade no desenvolvimento de software (alto nível de abstração, implantação e modularização facilitada, e automação e otimização de processos)	19
Facilidade em alcançar objetivos educacionais relacionados a área de TIC e Aprendizado da ferramenta Low Code	14
Construção e melhoria de habilidades (colaboração, solução de problemas, construção novas habilidades)	10
Competitividade e treinamento no mercado de trabalho	14
Desenvolvimento de soluções criativas	3
Suporte ao aprendizado em banco de dados (modelagem e scripts)	3
Experiência de usuário	1
Segurança	1

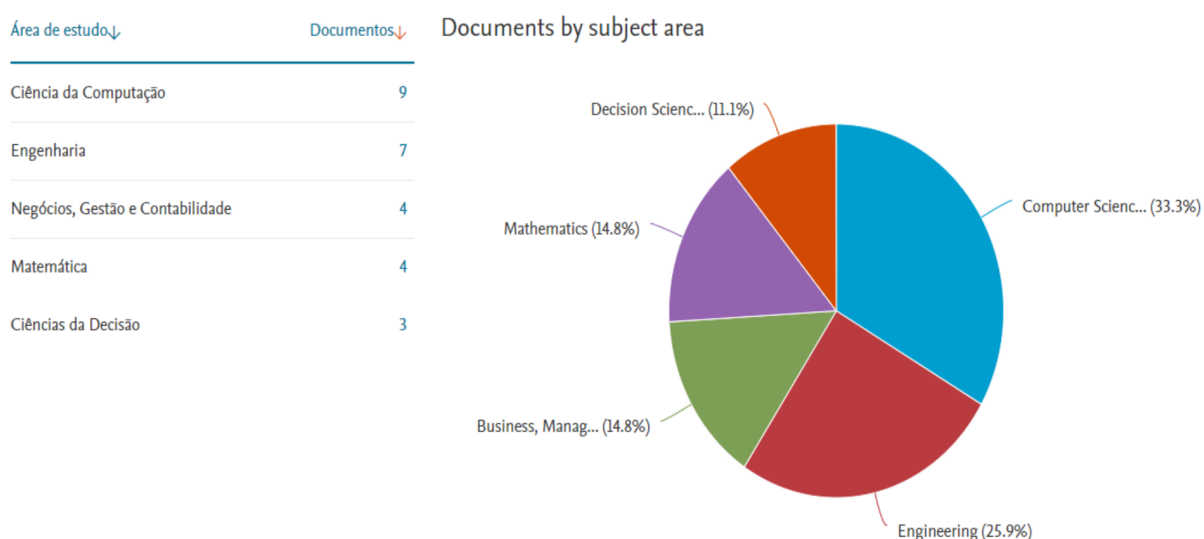
Fonte: Adaptado de De Souza (2022, p. 42).

2.2 APLICAÇÃO DE FERRAMENTAS LOW-CODE PARA MELHORIA E AUTOMAÇÃO DE PROCESSOS EM UMA EMPRESA DE CONTABILIDADE

Moraes (2022, p. 8) apresenta em seu trabalho de conclusão de curso um estudo de caso descrevendo a aplicação de ferramentas low-code para melhoria e automação de processos em uma empresa de contabilidade. Trabalhos de automação de processos de empresas do mercado, em geral, têm uma necessidade latente de adaptação operacional frente à evolução tecnológica que cresce e muda a cada dia. Diante dessa circunstância, se torna indispensável a utilização de métodos e ferramentas que contribuam para a melhorar a gestão e automação dos processos de negócio. O autor apresenta uma análise bibliométrica que buscou analisar a automação de processos e as plataformas low-code, bem como a intersecção entre essas duas áreas de pesquisa. Foi utilizado o banco de dados Scopus, tendo sido encontrados 12 trabalhos que tratavam diretamente dos temas.

A Figura 1 apresenta a distribuição dos trabalhos encontrados na pesquisa por área de estudos. O autor identificou que a área que compreende a maior parte dos artigos encontrados é a ciência da computação com 9 trabalhos, seguido pela área de Engenharia com 7 trabalhos.

Figura 1 — Número de documentos por área de estudo

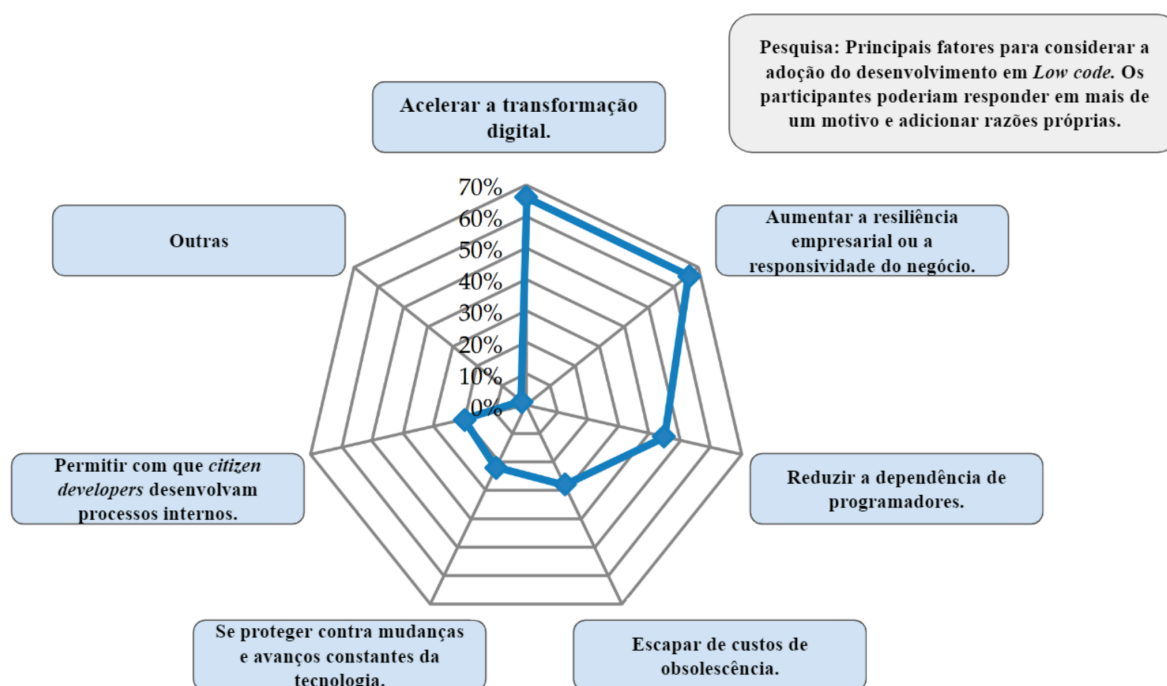


Fonte: Moraes (2022, p. 53).

2.3 DESENVOLVIMENTO LOW-CODE APLICADA À GESTÃO DE PROJETOS: UM ESTUDO DE CASO DA CONSTRUÇÃO DE UM DASHBOARD PARA A FASE INICIAL DE UM ESCRITÓRIO DE GESTÃO DE PROJETOS

Nogueira (2023) apresenta em seu trabalho de conclusão de curso que as plataformas low-code ganharam grande evidência após a onda de digitalização impulsionada pela pandemia do SARS-CoV-2. No contexto em que as empresas necessitavam competir em um mercado inteligente, as plataformas low-code podem proporcionar soluções rápidas, eficientes e de baixo custo para os gestores que necessitam de um maior grau de automação nos seus processos. Nesse contexto, o trabalho do autor tem como objetivo realizar um estudo de caso sobre o tema, que mostra na prática o contexto de desenvolvimento em low-code. Um excelente quadro é apresentado pelo autor, exibido na Figura 2, que aborda os motivos para as empresas adotarem as plataformas de desenvolvimento low-code.

Figura 2 — Motivos para a adoção do Low code



Fonte: Nogueira (2023, p. 52).

3 FERRAMENTAS DE DESENVOLVIMENTO LOW-CODE

Este capítulo apresenta algumas ferramentas gratuitas para o desenvolvimento de aplicativos mobile. Essas ferramentas também podem ser usadas para modelagem do sistema. As ferramentas estudadas foram: App Inventor¹, Kodular²e Thunkable³. A pesquisa foi restringida a essas três ferramenta por que são ferramentas completas com bastantes componentes visuais de clicar e arrastar para a tela e não precisam de extensão para funcionar, e pela facilidade de uso e são mais utilizadas por profissionais no setor de tecnologia da informação.

3.1 APP INVENTOR

O MIT App Inventor é um ambiente de desenvolvimento integrado de aplicativos da Web originalmente fornecido pelo Google e agora mantido pelo Instituto de Tecnologia de Massachusetts (MIT). Ele permite que os recém-chegados à programação de computador criem aplicativos de software para o sistema operacional Android. Ele usa uma interface gráfica, muito semelhante ao do zero e da interface do usuário StarLogo TNG, que permite aos usuários arrastar e soltar objetos visuais para criar um aplicativo que pode ser executado em dispositivos Android e IOS.

Ele usa uma interface gráfica do usuário (GUI) muito semelhante às linguagens de programação Scratch e StarLogo , que permite aos usuários arrastar e soltar objetos visuais para criar um aplicativo que pode ser testado em dispositivos Android e iOS e construído para rodar como um aplicativo Android. Ele usa um aplicativo móvel complementar que permite testes e depuração instantâneos ao vivo. Ao criar o App Inventor, o Google baseou-se em pesquisas anteriores significativas em computação educacional e no trabalho realizado no Google em ambientes de desenvolvimento online.

O App Inventor e os outros projetos são baseados e informados por teorias de aprendizado construcionistas, que enfatizam que a programação pode ser um veículo para engajar ideias poderosas por meio do aprendizado ativo. Como tal, faz parte de um movimento contínuo em computadores e educação que começou com o trabalho de Seymour Papert e do MIT Logo Group na década de 1960, e também se manifestou com o trabalho de Mitchel Resnick em Lego Mindstorms e StarLogo .O App Inventor também oferece suporte ao uso de dados em nuvem por meio de seu componente CloudDB.

1 <https://appinventor.mit.edu/>

2 <https://www.kodular.io/>

3 <https://thunkable.com/#/>

3.2 KODULAR

Entre as ferramentas estudadas estão o Kodular, que permite converter aplicativos Android utilizando uma plataforma online gratuita de programação visual e intuitiva de arrastar e soltar. Você não precisa dominar uma linguagem de programação, pois ele já traz os blocos os códigos necessários para desenvolvedores na construção do projeto, app inventor é uma ferramenta de programação baseada em blocos lógicos, desenvolvida pela Google que permite a criação de aplicativos para smartphones que rodam no sistema operacional Android, sem que seja necessário conhecimentos profundos de programação.

O Kodular é uma plataforma que oferece ferramentas e suporte para o usuário com o objetivo de facilitar o processo de criação de aplicativo Isso significa criar um aplicativo do zero sem precisar saber de programação. A partir de blocos lógicos de programação, essa ferramenta permite a criação de apps sem a necessidade de conhecimentos aprofundados sobre programação. Ele conta com um layout simples e intuitivo, sendo necessário poucos cliques ou arrastar alguns ícones e pronto! Sua própria ideia começa tomar a forma de um aplicativo para sistema Android.

Inspirado na ideia de um sistema de software desenvolvido pelo Instituto de Tecnologia de Massachusetts (MIT), o AppInventor, o Kodular oferece ainda uma App Store autodesenvolvida, onde se encontram aplicativos feitos a partir dele, um ID de extensões e um sistema para controle de contas.

3.3 THUNKABLE

Thunkable é um construtor drag-and-drop (clique e soltar) para criação de aplicativos Android com aparência nativa e recursos interativos. Tudo é feito através de uma interface visual com os componentes e conexões disponíveis. Este editor possui dois componentes principais: Designer e Blocos. ele é baseado no app inventor, Os blocos lógicos são basicamente uma abstração dessa lógica. Ao invés de ter que escrever o código correspondente, o bloco lógico faz isso por debaixo dos panos. Cada bloco representa uma ação específica e o usuário vai ligar estes blocos para montar um algoritmo completo.

O primeiro passo é entrar no site do Thunkable e se cadastrar. Feito isso, clique no botão “Create New App” para criar um projeto do zero. Ao clicar, um modal abrirá. No campo disponível, insira qual é o nome do seu projeto.

No quadro seguinte, temos duas abas: Design e Blocks . Na primeira aba é onde encontraremos todos os elementos visuais que podemos colocar dentro da tela do nosso aplicativo (representado no quadro a seguir). Por exemplo, vamos

adicionar um botão (o primeiro elemento). Para isso, basta arrastar e soltar.

Agora, vamos inserir uma interação a este botão com os blocos lógicos. Para isso, entre na aba Blocks . Ao abrir, você irá se deparar com uma boa quantidade de menus. Estes são todos os blocos de ações correspondentes ao nosso botão.

3.4 COMPARATIVO ENTRE AS FERRAMENTAS

Dentre as ferramentas estudadas, é possível identificar que todas tem o mesmo propósito, porém cada uma tem seus aspectos, funcionalidades e objetivos. A Tabela 2 apresenta um comparativo entre as ferramentas descritas. As características destacadas nessa tabela consideram as funcionalidades e objetivos aplicáveis ao presente trabalho de conclusão de curso.

Tabela 2 — Tabela comparativa entre as ferramentas Low-Code

Funcionalidades		
App Inventor	Kodular	Thunkable
<p>App Size (tamanho do app .aia/.apk): 5 MB / 10 MB;</p> <p>Screens (número de telas): 10 telas;</p> <p>Minimum Android version (versão mínima do Android suportada): 1.6 (donut, API level 4);</p> <p>Não é cross plataforma, ou seja, não é multiplataforma o que implica dizer que os aplicativos feitos aqui não poderão rodar num sistema IOS;</p> <p>Aceita o uso de extensões;</p> <p>Possui galeria de apps e guias;</p> <p>Live Testing;</p> <p>100% gratuito;</p>	<p>App Size (tamanho do app .aia/.apk): 5 MB / 20 MB;</p> <p>Screens (número de telas): ilimitado;</p> <p>Minimum Android version (versão mínima do Android suportada): 4.1 (16) JellyBean;</p> <p>Aceita o uso de extensões;</p> <p>Possui galeria de apps e guias;</p> <p>Live Testing;</p> <p>É 100% gratuito;</p>	<p>App Size (tamanho do app .aia/.apk): 5 MB / 50 MB;</p> <p>Screens (número de telas): ilimitado;</p> <p>Minimum Android version (versão mínima do Android suportada): 5.0 (21) Lollipop;</p> <p>Minimum iOS version: 10.0;</p> <p>É Cross Plataforma;</p> <p>Não aceita o uso de extensões;</p> <p>Possui galeria de apps e guias;</p> <p>Live Testing;</p> <p>Não é 100% gratuito;</p>

Fonte: O autor (2023).

4 SOLUÇÃO CONCEITUAL

O presente trabalho de conclusão de curso apresenta, como estudo de caso, o desenvolvimento de um aplicativo para dispositivos móveis através das plataformas low-code, baseado em blocos de código e com funções de clicar e arrastar componentes na tela. O aplicativo tem como objetivo gerenciar gastos pessoais como despesas e receitas, através de uma interface intuitiva que apresente dados estatísticos para o usuário na tela.

A principal funcionalidade do aplicativo é o gerenciamento de despesas e receitas. O aplicativo permite o cadastro e listagem de despesas e receitas.

A seguir são apresentadas as seções de histórias de usuário, requisitos funcionais e não funcionais e o diagrama entidade-relacionamento do aplicativo desenvolvido.

4.1 HISTÓRIAS DE USUÁRIO

Na Tabela 3 podemos visualizar as histórias de usuário projetadas para o aplicativo. As histórias de usuário estão organizadas por ordem de execução iniciando com história de cadastro de usuário, onde o usuário digita seu nome e cria uma senha. Na história de usuário número 2, o usuário pode utilizar o aplicativo fazer login com usuário e senha. A história número 3 permite que o usuário possa cadastrar uma despesa, sendo necessário estar logado no aplicativo. Na história de usuário número 4 o usuário pode cadastrar uma receita, sendo necessário estar logado no aplicativo. A história de usuário número 5 permite que o usuário possa excluir, editar e cancelar despesas e receitas, sendo necessário estar logado no aplicativo e ter uma despesa ou receita cadastrado.

Tabela 3 — Descrição das Histórias de Usuário (continua)

Identificador	História	
US1	Como	Usuário
	O que	Cadastro do usuário no aplicativo
	Para que	Para o usuário logar no aplicativo
	Critério de aceitação	Necessário digitar o nome e criar uma senha
US2	Como	Usuário
	O que	Login no aplicativo

Tabela 3 — Descrição das Histórias de Usuário (conclusão)

Identificador	História	
US3	Para que	Para que o usuário possa acessar as demais funcionalidades do aplicativo
	Critério de aceitação	Necessário possuir usuário e senha
	Como	Usuário
	O que	Cadastro de despesa
US4	Para que	O usuário possa cadastrar uma despesa
	Critério de aceitação	1. Necessário estar logado no aplicativo 2. Necessário selecionar uma categoria
	Como	Usuário
	O que	Cadastro de receita
US5	Para que	O usuário possa cadastrar uma receita
	Critério de aceitação	1. Necessário estar logado no aplicativo 2. Necessário selecionar uma categoria
	Como	Usuário
	O que	Listagem de despesas e receitas
	Para que	O Usuário possa excluir, editar e cancelar despesas e receitas
	Critério de aceitação	1. Necessário estar logado no aplicativo 2. Necessário ter uma despesa ou receita cadastrado

Fonte: O autor (2023).

4.2 REQUISITOS FUNCIONAIS

Os recursos viáveis para o funcionamento deste aplicativo são descritos nesta seção, que tem por objetivo identificar as funcionalidades ou serviços que aplicação disponibilizará ao usuário para que atinja os objetivos propostos.

Na Tabela 4 são apresentados os requisitos funcionais identificados no aplicativo. RF1 trata sobre cadastro de usuário no aplicativo. RF2 trata sobre login no aplicativo, sendo que o usuário só pode fazer login com senha criada no aplicativo. RF3 trata sobre cadastro de despesa no aplicativo. RF4 trata sobre cadastro de receita no aplicativo. RF5 trata sobre listagem de despesas e receitas no aplicativo.

Tabela 4 — Descrição Requisitos Funcionais do Sistema

Id	Nome	Prioridade	Status	Descrição	Responsável
RF1	Cadastro de usuário	Alta	Identificado	O aplicativo deve disponibilizar a funcionalidade de cadastro de usuário e senha	Usuário
RF2	Login no aplicativo	Alta	Identificado	O aplicativo deve disponibilizar a funcionalidade de fazer login com a senha	Usuário
RF3	Cadastro de despesa	Alta	Identificado	O aplicativo deve disponibilizar a funcionalidade de cadastro de despesa	Usuário
RF4	Cadastro de receita	Alta	Identificado	O aplicativo deve disponibilizar a funcionalidade de cadastro de receita	Usuário
RF5	Listagem de despesas e receitas	Alta	Identificado	O aplicativo deve disponibilizar a funcionalidade de listar despesas e receitas, permitindo que o usuário exclua, edite e cancele despesas e receitas	Usuário

Fonte: O autor (2023).

4.3 REQUISITOS NÃO FUNCIONAIS

A segurança do aplicativo é um requisito não funcional identificado. O cadastro de uma despesa ou receita deve ser restrito ao usuário, podendo garantir a integridade dos dados e a privacidade do usuário quando estiver utilizando o aplicativo.

Para instalar e utilizar o aplicativo é necessário ter um telefone com sistema operacional Android.

Tabela 5 — Descrição dos Requisitos Não Funcionais do Sistema

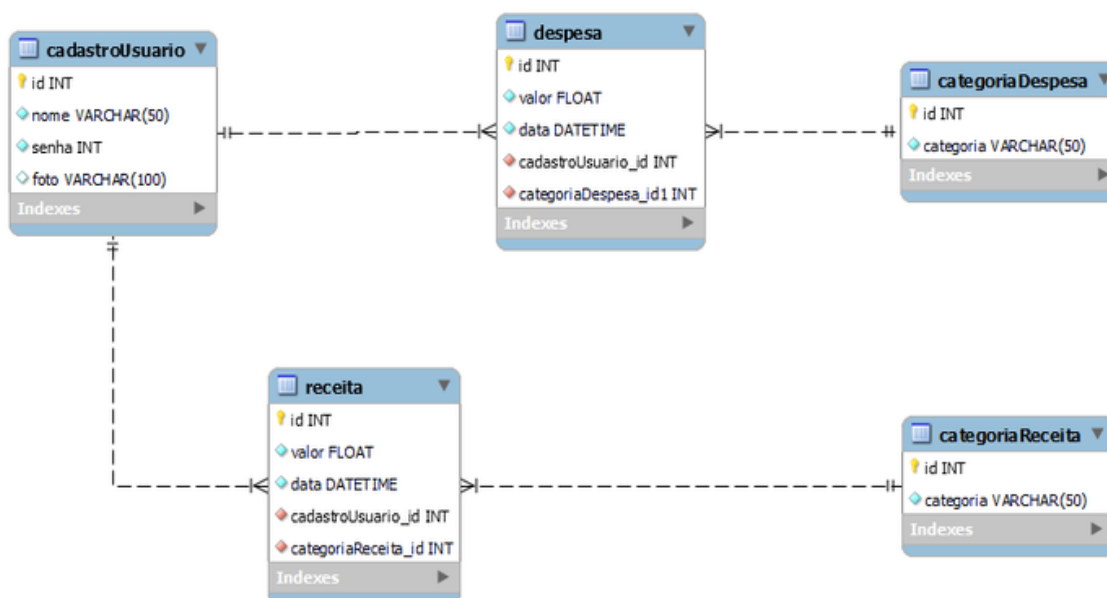
Id	Nome	Prioridade	Status	Descrição	Responsável
RNF1	Segurança do aplicativo	Alta	Identificado	O aplicativo deve garantir a segurança dos dados de usuário através da senha e login de acesso.	Analista
RNF2	Sistema operacional	Alta	Identificado	O aplicativo deve rodar no sistema operacional Android	Analista

Fonte: O autor (2023).

4.4 DIAGRAMA ENTIDADE/RELACIONAMENTO

A Figura 3 apresenta o diagrama entidade/relacionamento para fins de projeto do banco de dados do aplicativo.

Figura 3 — Diagrama Entidade e Relacionamento



Fonte: O autor (2023).

Conforme visto na Figura 3, a tabela cadastro de usuário armazena nome e senha e a foto do usuário. A tabela despesa armazena os valores das despesas cadastradas. A tabela categoria despesa armazena as categorias de despesas. A tabela receita armazena os valores da receitas cadastradas. A tabela categoria receita armazena as categorias de receitas.

5 ESTUDO DE CASO

Como estudo de caso do presente trabalho de conclusão de curso, foi criado um aplicativo de controle gastos pessoais para demonstrar as diferenças entre as plataformas Kodular e App Inventor. O aplicativo não foi desenvolvido na plataforma Thunkable porque é uma plataforma paga e também porque os aplicativos ficam abertos na plataforma para que qualquer pessoa possa utilizar, não apresentando segurança aos usuários.

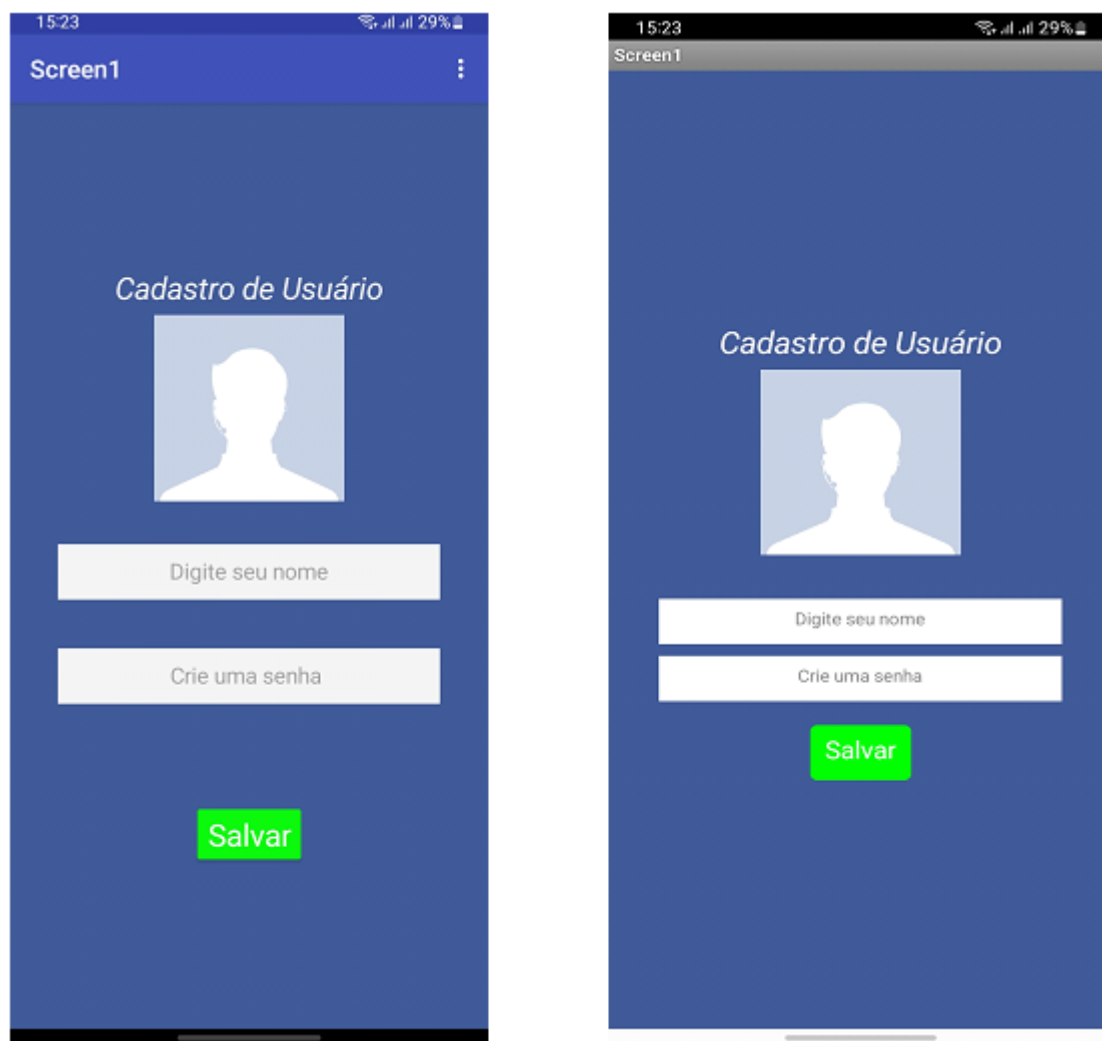
O aplicativo desenvolvido tem o objetivo de gerenciar o controle de gastos pessoais. Para o usuário fazer cadastro no aplicativo ele deve digitar seu nome e criar uma senha, depois para cadastrar uma despesa ou receita é só clicar no botão e selecionar uma categoria para adicionar na lista.

A seguir, será exibido o conjunto de telas gerado em cada plataforma, lado a lado para fins de comparação. Nas figuras, a imagem à esquerda apresenta a tela gerada na plataforma Kodular , enquanto a imagem à direita apresenta a tela gerada na plataforma App Inventor.

5.1 TELA DE CADASTRO DE USUÁRIO

Conforme pode ser visualizado na Figura 4, a tela de cadastro de usuário exige que o usuário digite seu nome e crie uma senha para acesso ao aplicativo. Esta tela será solicitada no primeiro acesso do usuário ao aplicativo.

Figura 4 — Tela de cadastro de Usuário

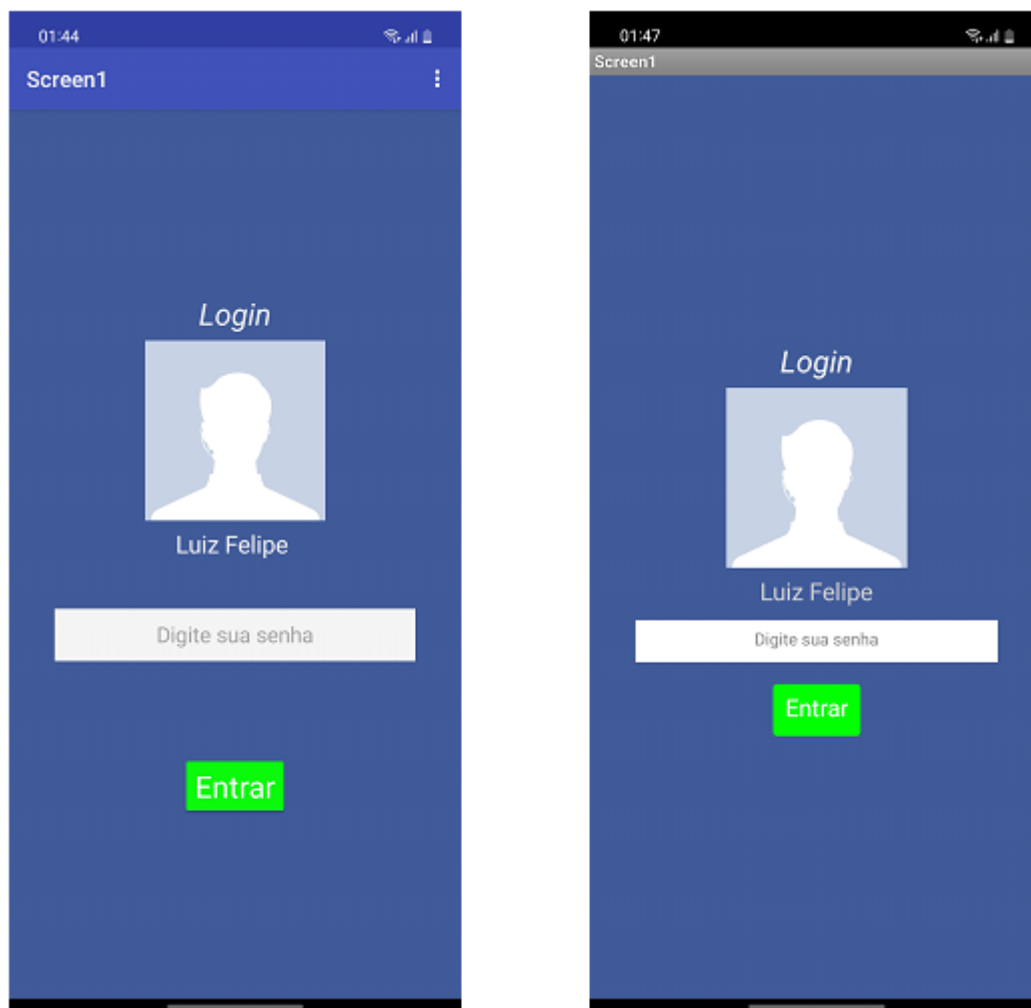


Fonte: O autor (2023).

5.2 TELA DE LOGIN DO USUÁRIO

A Figura 5 apresenta a tela de login. Na tela de login, o usuário não precisa mais informar seu nome, pois ele já está salvo no banco de dados do aplicativo. Agora é só informar a senha que o usuário criou e fazer login para tela inicial.

Figura 5 — Tela de Login do Usuário

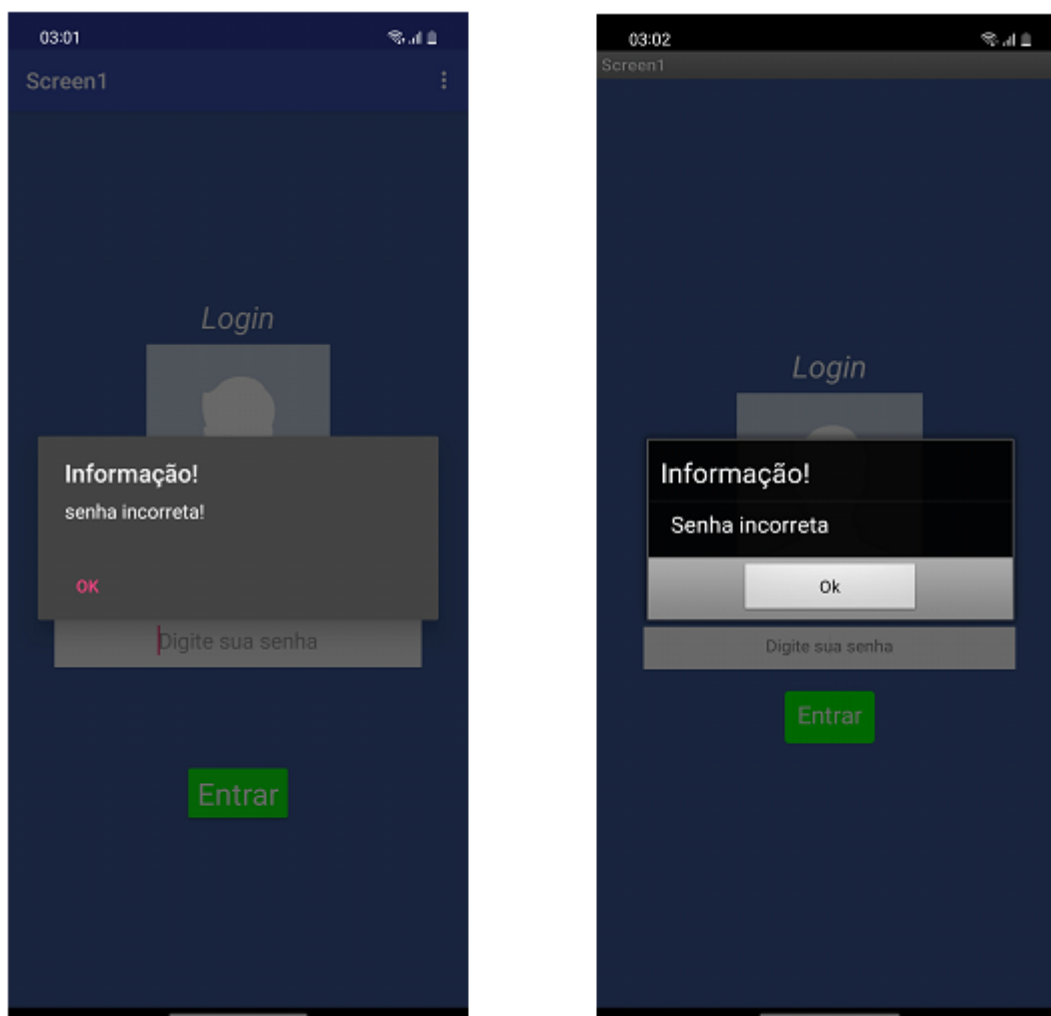


Fonte: O autor (2023).

5.3 TELA DE LOGIN SENHA INCORRETA

Como pode ser visualizado na Figura 6, se o usuário digitar sua senha errado, irá aparecer uma mensagem informando que a senha está incorreta.

Figura 6 — Tela de Login Senha Incorreta

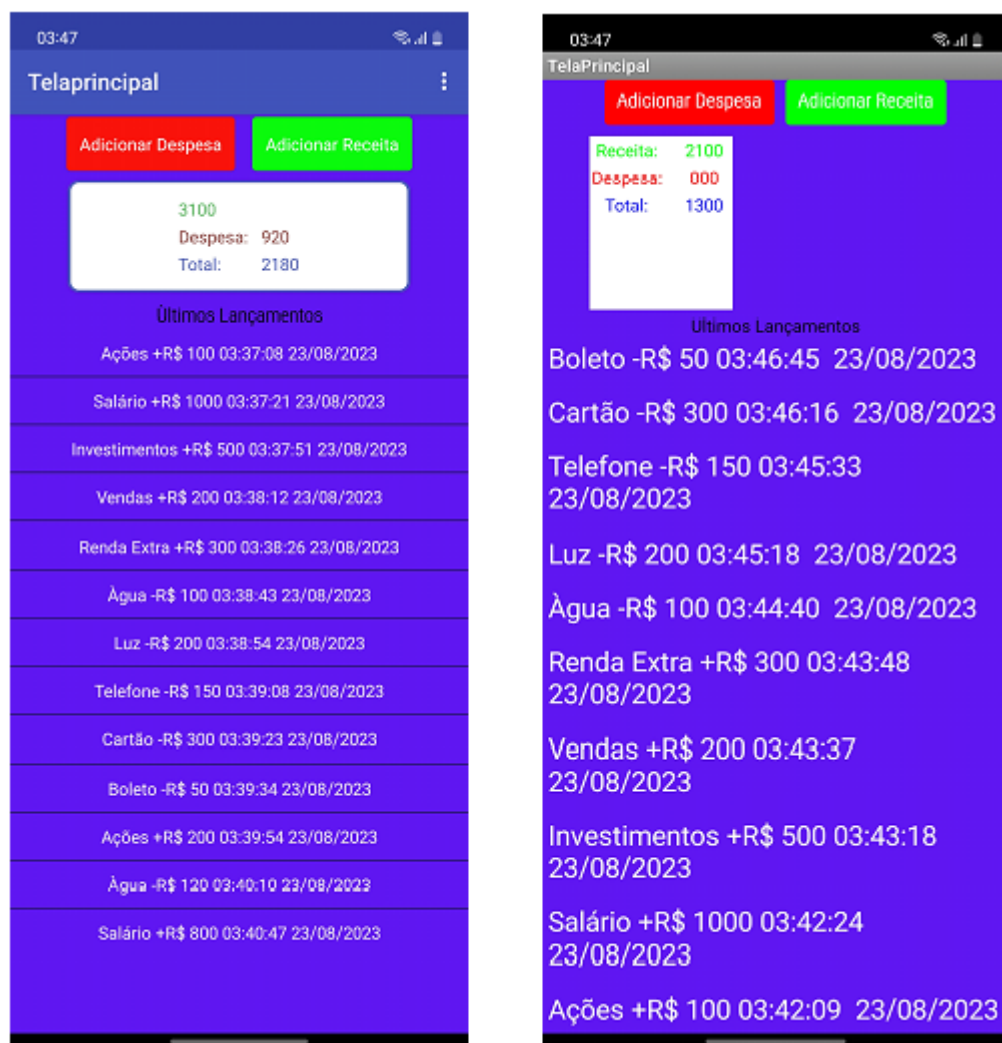


Fonte: O autor (2023).

5.4 TELA INICIAL DO APLICATIVO

A Figura 7 apresenta a tela inicial do aplicativo após login. Ela tem dois botões: um para adicionar uma despesa e outro para adicionar uma receita. Logo abaixo é apresentado um espaço com a soma e o total de receita e despesa. Abaixo, são listados os últimos lançamentos cadastrados de despesas e receitas, assim como suas respectivas categorias.

Figura 7 — Tela Inicial do Aplicativo

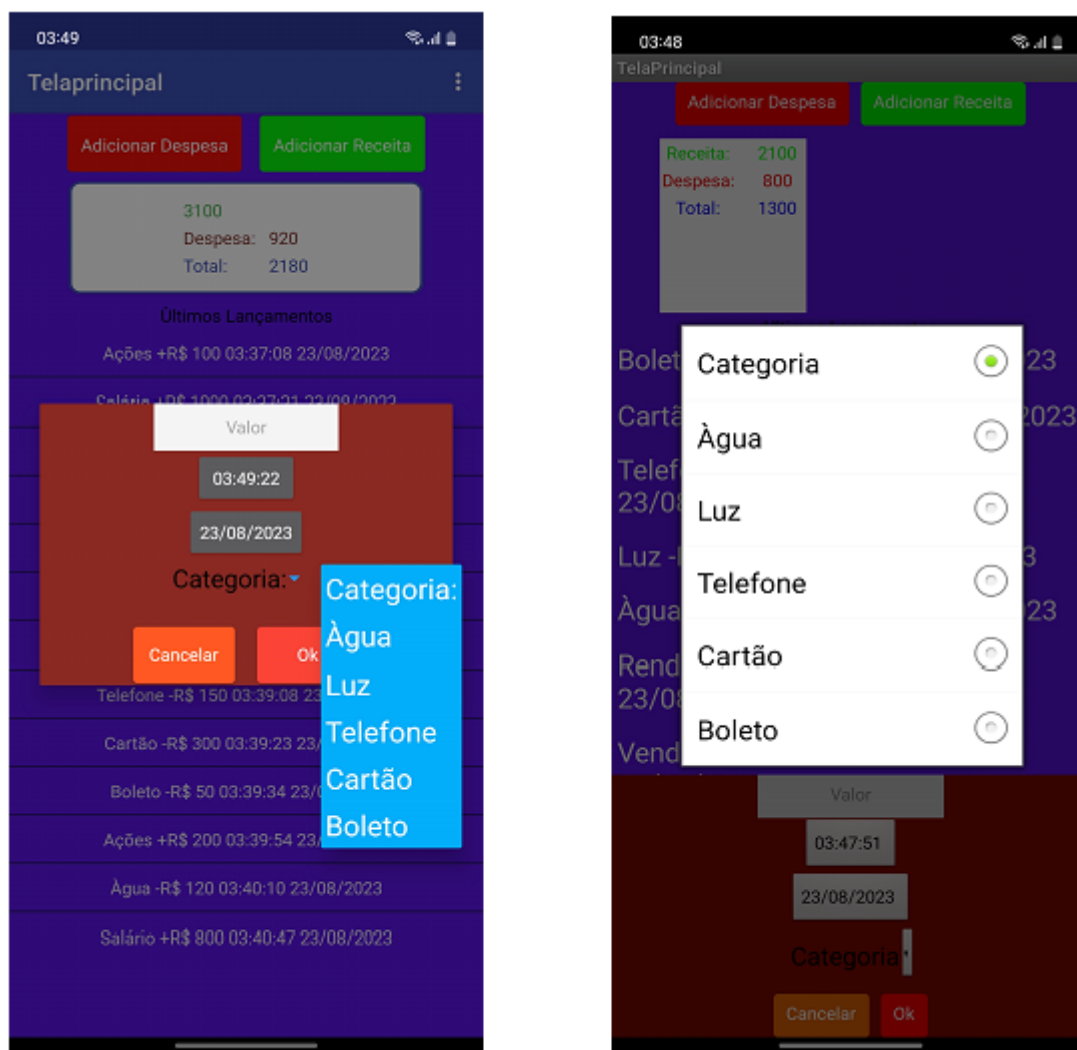


Fonte: O autor (2023).

5.5 TELA DE DESPESA CATEGORIA

A Figura 8 apresenta a tela de cadastro de despesa. Ao clicar no botão adicionar despesa, irá abrir um card solicitando que o usuário informe o valor e a categoria da despesa, podendo ser água, luz, telefone, cartão e boleto.

Figura 8 — Tela de Cadastro de Despesa

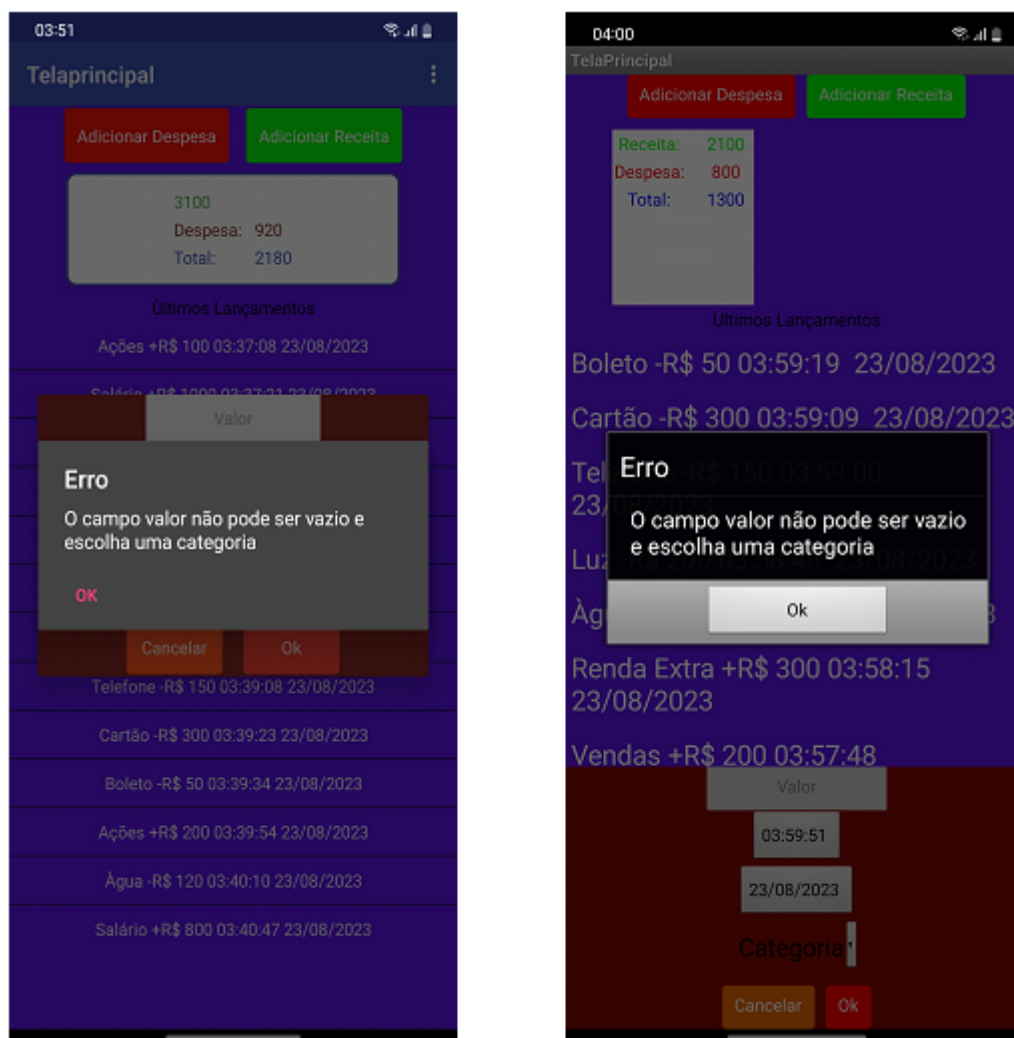


Fonte: O autor (2023).

5.6 TELA DE DESPESA CAMPO VAZIO CATEGORIA

Como pode ser visualizado na Figura 9, o campo valor deve ser preenchido e deve-se escolher uma categoria da despesa, caso contrário, uma mensagem aparecerá para o usuário.

Figura 9 — Tela do Campo Despesa Valor Categoria

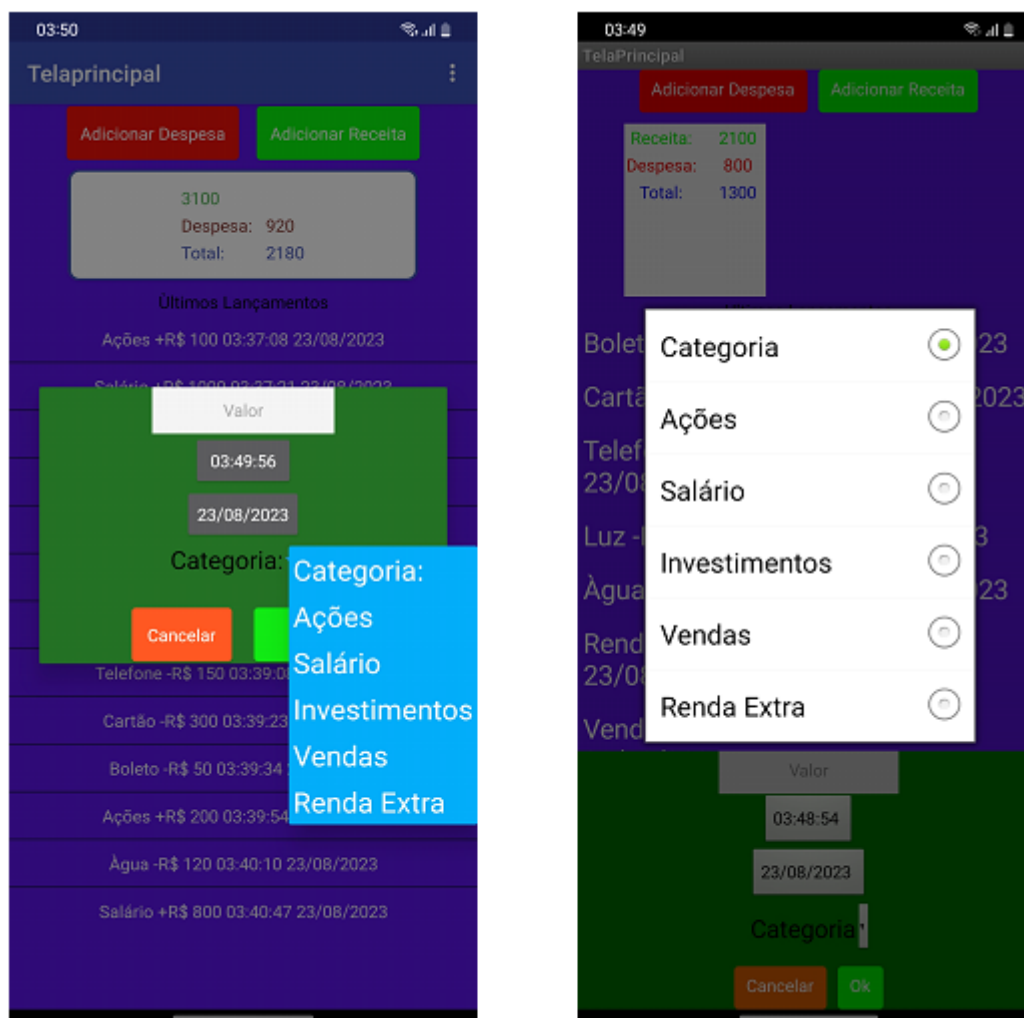


Fonte: O autor (2023).

5.7 TELA DE RECEITA CATEGORIA

A Figura 10 apresenta a tela de cadastro de receita. Ao clicar no botão adicionar receita, irá abrir um card solicitando que o usuário informe o valor e a categoria da receita, podendo ser ações, salário, investimentos, vendas ou renda extra.

Figura 10 — Tela de Cadastro de Receita

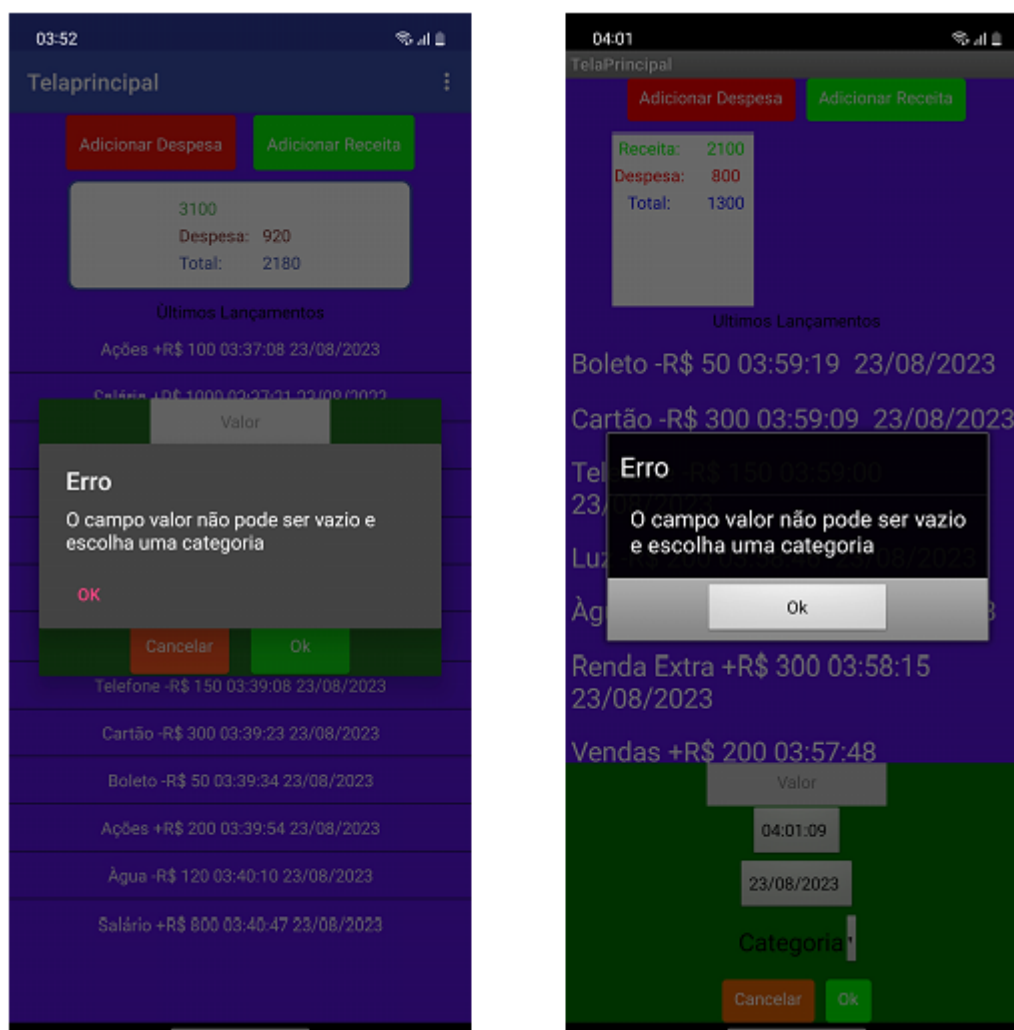


Fonte: O autor (2023).

5.8 TELA DE RECEITA CAMPO VAZIO CATEGORIA

Como pode ser visualizado na Figura 11, o campo valor deve ser preenchido e deve-se escolher uma categoria da receita, caso contrário uma mensagem aparecerá para o usuário.

Figura 11 — Tela Campo Receita Valor Categoria

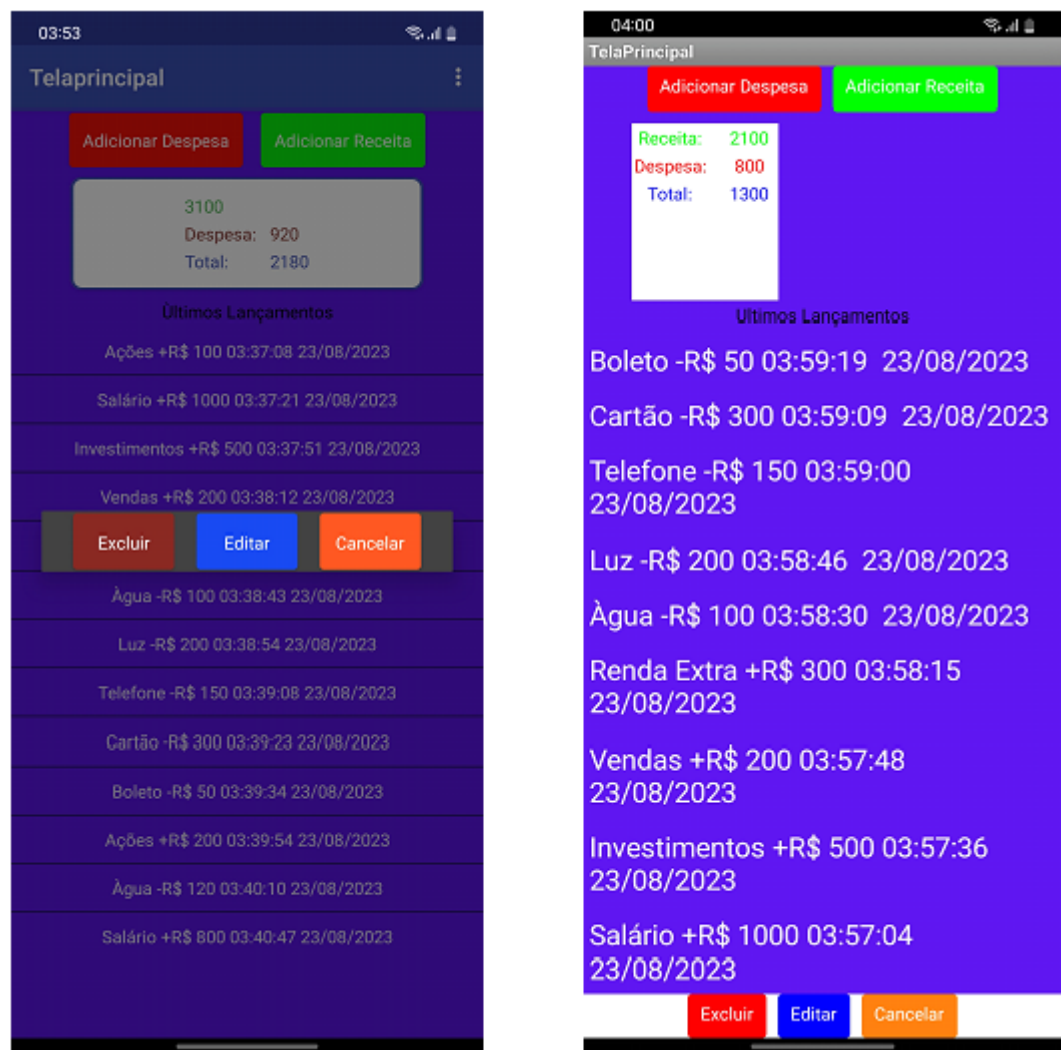


Fonte: O autor (2023).

5.9 TELA DE LISTA EXCLUIR, EDITAR E CANCELAR

A Figura 12 apresenta a tela gerada quando o usuário clica em algum item da lista de despesa e receita que estão cadastrados. O usuário poderá excluir, editar e cancelar valores e a categoria selecionada.

Figura 12 — Tela de Listar Excluir Editar e Cancelar

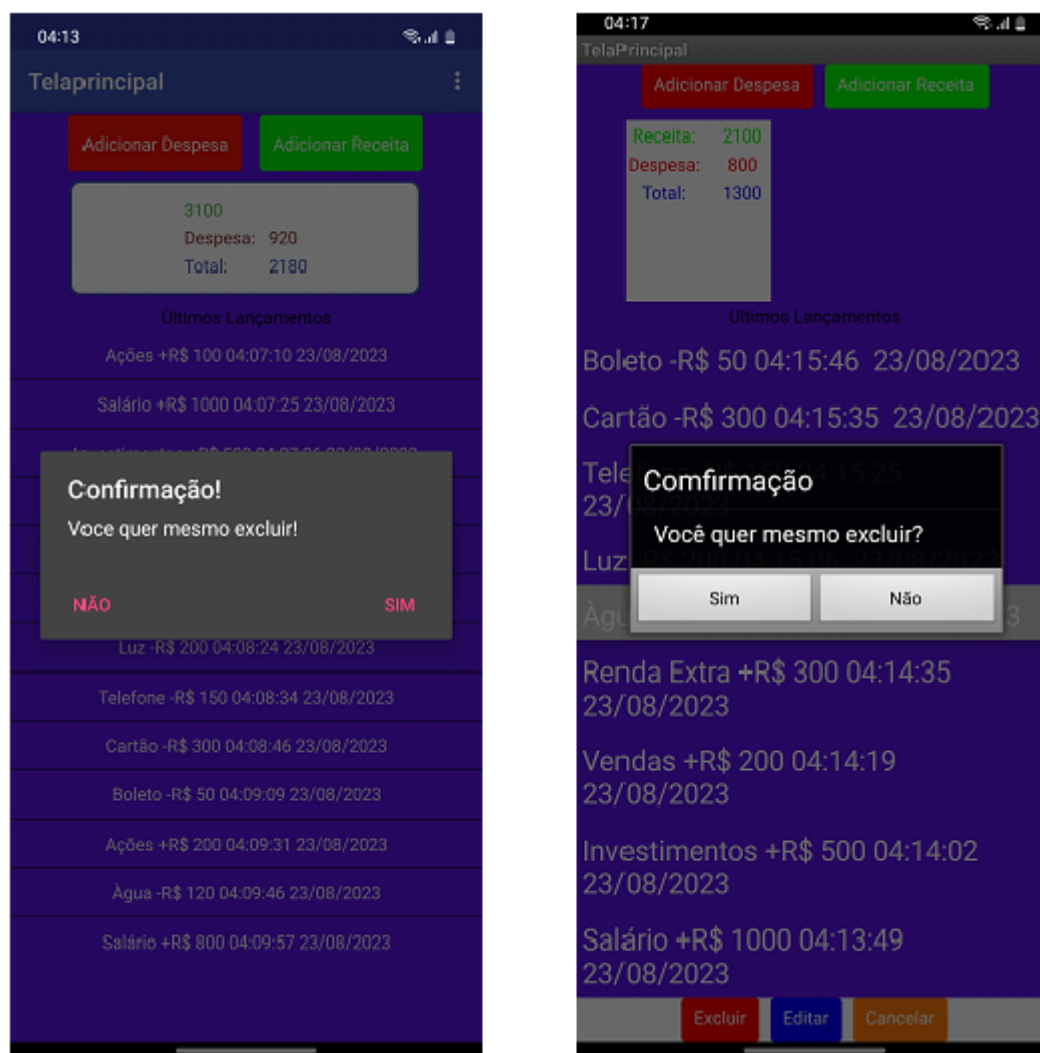


Fonte: O autor (2023).

5.10 TELA DE CONFIRMAR EXCLUIR

Como pode ser visualizado na Figura 13, ao clicar em algum item da lista irá aparecer essa mensagem perguntado se o usuário quer mesmo excluir algum valor e categoria selecionada.

Figura 13 — Tela de Confirmação Excluir



Fonte: O autor (2023).

6 DISCUSSÃO TÉCNICA

As plataformas low-code são aquelas que permitem o desenvolvimento de um programa a partir de uso mínimo de código possível. Elas foram criadas quando desenvolvedores perceberam uma alta taxa de repetição em linhas de código. Apresentam um visual diferente das linhas de código tradicionais, com uma interface mais intuitiva e agradável para quem utiliza, com botões de atalho, módulos pré-configurados ou até mecanismos de arrastar e posicionar elementos na tela.

O foco das plataformas low-code são é evitar serviços manuais desnecessários, sendo especialmente útil em projetos longos ou empresas que repetem os mesmos processos ao longo do tempo.

Ao final deste trabalho de conclusão de curso percebe-se que as plataformas low-code vieram para suprir uma necessidade na área da TI, já que falta programadores no mercado com conhecimento. Essas ferramentas são uma opção para quem está iniciando sua carreira na área de programação, já que qualquer pessoa com pouco conhecimento pode criar um aplicativo do zero.

As plataformas estudadas, App Inventor e Kodular, são bem similares. O App inventor, permite desenvolver aplicativos para Android usando um navegador da Web e um telefone ou emulador conectado. O processo para criar o aplicativo é bem simples, bastando você selecionar os componentes que deseja na interface e depois montar os blocos que especificam como cada componente deve se comportar. Funcionalidades relacionadas à interface de usuário contém 15 componentes, mídia contém 10 componentes, Desenho e Animações contém 3 componentes, Mapa contém 7 componentes, Sensores contém 9 componentes, Sociais contém 7 componentes, Arrumação contém 4 componentes, Conectividade contém 4 componentes, Lego Mindstorms contém 15 componentes, e Experimental contém 2 componentes (Firebase, CloudDB).

O Kodular é uma versão mais atualizada do App Inventor, apresentando mais componentes que auxiliaram na hora do desenvolvimento. fornecem vários serviços para a criação de tipos de aplicativos Android sem codificação. O Kodular Creator está hospedado no Google Cloud Platform, logo seus projetos serão mantidos com segurança. Quanto às funcionalidades, a Interface de usuário contém 29 componentes, Layout contém 10 componentes, Media contém 17 componentes, Desenho e animações contém 6 componentes, Mapa contém 7 componentes, Sensores contém 16 componentes, Armazenamento contém 7 componentes, Social contém 10 componentes, e Utilidades contém 9 componentes.

Os componentes listados são tudo aquilo que o desenvolvedor pode clicar ou arrastar para a tela do aplicativo, sendo que esses elementos já vem pré configurados, permitindo ao desenvolvedor alterar no blocos de código.

Ao utilizar a plataforma low-code Kodular, percebeu-se que ela tem alguns métodos de criação que segue o padrão de arrastar componentes começando com uma tela em branco. Esses componentes podem ser caixas de texto, botões laterais ou flutuantes em um menu. Os componentes já vem pré-configurados, auxiliando bastante no desenvolvimento do aplicativo. O desenvolvedor clica na interface e arrasta um componente, como os botões de despesa e receita do aplicativo desenvolvido. Para mostrar quanto entrou e saiu na tela, foram escolhidos os seguintes componentes, card view e organização horizontal, para mostrar para o usuário as despesas e receitas e o total delas. E para o usuário poder escolher uma categoria foram escolhidos o componente de uma lista suspensa que serve para selecionar as categorias de despesa e receita. Para o usuário do aplicativo poder visualizar suas despesas e receita na tela do aplicativo foi utilizado uma list view ou visualizador de lista.

A plataforma low-code App inventor também segue alguns padrões de clicar e arrastar e soltar componentes na tela. para criação do layout do aplicativo, foram utilizados os seguintes componentes: a) uma organização horizontal para posicionar os elementos na tela; b) dois botões um para adicionar uma despesa e outra para adicionar uma receita; c) lista suspensa para mostrar na tela as categorias de despesas e receita; d) visualizar de lista para mostrar para o usuário suas despesas e receita por categoria

No desenvolvimento do aplicativo foram encontradas algumas dificuldades tais como os componentes que são de interface de usuário como espaços entre os botões, sensores como um notificador. O Kodular, apesar de ser semelhante ao App Inventor, tem mais componentes, como interface de usuário e espaços entre os botões.

Tecnicamente, essas plataformas de desenvolvimento tendem a trazer um aumento na produtividade, escalabilidade, agilidade, e ganho de tempo ao poder utilizar componentes de clicar arrastar e soltar na tela, e usar apenas blocos de código na construção do aplicativo.

7 CONCLUSÃO E TRABALHOS FUTUROS

Esse trabalho foi desenvolvido com objetivo de mostrar que qualquer pessoa com pouco conhecimento em programação consegue criar um aplicativo do zero utilizando essas plataformas como base.

Durante o processo de pesquisa para o desenvolvimento desse trabalho foi observado que essas plataformas vem crescendo muito, sendo uma boa oportunidade para quem quer apreender a desenvolver aplicativos sem precisar saber codificar. É por isso que essas ferramentas tem chamado a atenção.

As plataformas low-code oferecem um grande potencial na hora de ajudar os desenvolvedores na criação de aplicativos para dispositivos móveis. além de serem fáceis de se utilizar, também aceleram o processo de desenvolvimento, já que você não precisa escrever todo o código de um projeto manualmente.

Os software desenvolvidos por plataformas low-code permitem a produção de aplicativos para dispositivos móveis de forma mais rápida e com um processo de codificação manual mínimo. Os blocos de construção abstraem o código por trás das ações e comandos, tornando possível para o desenvolvedor montar fluxos de trabalho e aplicativos de negócios sem a necessidade de trabalhar de forma manual. Desta forma, é como se fosse um quebra-cabeça: já existem blocos ou peças prontos para ir encaixando para entregar o resultado final que espera.

Essas plataformas também apresentam algumas vantagens como agilidade. Elas permitem integração com APIs e serviços web, e oferece também produtividade ao simplificar o processo de desenvolvimento e aumentar o uso da automação, simplificando o ciclo de vida dos processos.

De maneira geral, como trabalhos futuro, diversas melhorias poderiam ser desenvolvidas, como uma funcionalidade no aplicativo para filtrar despesas e receitas a partir de sua categoria, cadastro de novas categorias de receitas e despesas, e também uma tela para os valores totais de despesa e receita de cada mês. Também fica como trabalho futuro a realização de uma pesquisa com empresas que utilizam essas plataformas low-code como desenvolvimento, buscando saber seus pontos de vista, visando conhecer se essas plataformas aumentam a produtividade, agilidade e automação dos processos na prática.

REFERÊNCIAS BIBLIOGRÁFICAS

App inventor: Mit App inventor. wikipédia. Disponível em:
https://en.wikipedia.org/wiki/MIT_App_Inventor. Acesso em: 4 set. 2023.

Bem paraná. **Apps e softwares em low-code devem crescer 65% até 2024**: Pandemia da covid19 e consultoria Gartner. Blog Mettzer. Florianópolis, 2021. Disponível em: <https://www.bemparana.com.br/tecnologia/apps-e-sofwarees-em-low-code-devem-crescer-65-ate-2024-270825/>. Acesso em: 10 mai. 2021.

DE SOUZA, JHONATA LIMA . **A Contribuição do Low Code no Âmbito Educacional: Um Mapeamento Sistemático da Literatura**. 2022. 65 slides. Disponível em: https://www.cin.ufpe.br/~tg/2021-2/tg_SI/TG_jls3.pdf. Acesso em: 4 set. 2023.

MEDIUM. **Kodular, App Inventor ou Thunkable?**: Diferenças entre kodular-app-inventor-ou-thunkable. Mettzer. Florianópolis, 2016. Disponível em: <https://medium.com/stefany-s%C3%A1/kodular-app-inventor-ou-thunkable-484865945b29>. Acesso em: 21 ago. 2016.

Medium. **Thunkable**: criando aplicativos low code. medium. Disponível em: <https://medium.com/code-prestige/thunkable-criando-aplicativos-sem-c%C3%B3digo-3d408c4c7e80>. Acesso em: 4 set. 2023.

MORAES, PEDRO HENRIQUE MACENA MONTEIRO. **APLICAÇÃO DE FERRAMENTAS LOW-CODE PARA MELHORIA E AUTOMAÇÃO DE PROCESSOS EM UMA EMPRESA DE CONTABILIDADE**. 2022. 93 slides. Disponível em: https://repositorio.ufc.br/bitstream/riufc/65241/1/2022_tcc_pmoraes.pdf. Acesso em: 4 set. 2023.

NOGUEIRA, João Pedro Barbosa Bochi. **DESENVOLVIMENTO LOW CODE APLICADO À GESTÃO DE PROJETOS: Um Estudo de Caso da Construção de um Dashboard para a fase inicial de um Escritório de Gestão de Projetos**. 2023. 87 slides. Disponível em: <http://www.repositorio.poli.ufrj.br/monografias/projpoli10039582.pdf>. Acesso em: 4 set. 2023.

ProfAntenado. **Kodular**: o que é kodular. profAntenado. Disponível em: <https://blog.profantenado.com/o-que-e-kodular>. Acesso em: 4 set. 2023.

SYDLE. **Plataforma low code: o que é e quais as suas vantagens?**: Vantagens do Low Code . Blog Mettzer. Florianópolis, 2020. Disponível em: <https://www.sydle.com/br/blog/plataforma-low-code-6007088adbedb32b132e28a4/>. Acesso em: 10 mai. 2021.