



ARQUITETURA MODULAR E ESCALÁVEL PARA DESENVOLVIMENTO DE APLICATIVOS MÓVEIS NO ESCRITÓRIO DE SOLUÇÕES CRIATIVAS

MODULAR AND SCALABLE ARCHITECTURE FOR MOBILE APPLICATION DEVELOPMENT IN THE CREATIVE SOLUTIONS OFFICE

Rafael Aguiar SILVA

Universidade Estadual do Tocantins (UNITINS)

E-mail: rafaelaguiar@unitins.br

ORCID: <http://orcid.org/0009-0000-8902-2721>

Jeferson Moraes da COSTA

Universidade Estadual do Tocantins (UNITINS)

E-mail: jeferson.mc@unitins.br

ORCID: <http://orcid.org/0000-0001-7605-3174>

780

RESUMO

Arquitetura de software é algo imprescindível para construir qualquer sistema de software escaláveis, testáveis e de fácil manutenção. No entanto nem sempre ela é implementada nos projetos do Escritório de Soluções Criativas (ESC), seja por falta de conhecimento ou por ignorar alguns aspectos importantes do processo de construção. Diante desse cenário, foi idealizado uma proposta para desenvolver uma arquitetura de software para construção de aplicativos moveis. A arquitetura irá conter um esqueleto de organização de pastas e arquivos, assim com um sistema de rotas e módulos embutidos, também virá com as camadas de busca e transformações de dados, tradução e gerenciamento de estados. Todo esse acervo estará disponível no repositório interno do ESC, no qual será disponibilizado para os desenvolvedores. Dessa forma busca-se maximizar a qualidade do desenvolvimento e a qualidade das entregas.

Palavras-chave: Arquitetura de software. Design de software. Componentes de software. Multiplataforma. Desenvolvimento móvel.

ABSTRACT

Software architecture is essential for building any software system that is scalable, testable and easy to maintain. However, it is not always implemented in the projects of

the Office of Creative Solutions (ESC), either because of a lack of knowledge or because it ignores some important aspects of the construction process. Given this scenario, a proposal was drawn up to develop a software architecture for building mobile applications. The architecture will contain a skeleton for organizing folders and files, as well as a system of routes and built-in modules, and will also come with layers for searching and transforming data, translation and state management. All of this will be available in the ESC's internal repository, which will be made available to developers. The aim is to maximize the quality of development and the quality of deliveries.

Keywords: Architecture software. Software design. Software components. Multi-platform development. Mobile development.

INTRODUÇÃO

O conceito de arquitetura de software é amplamente utilizado na atualidade, contudo, suas origens remontam ao período subsequente ao surgimento da engenharia de software, aproximadamente em 1968, quando foi inicialmente proposto por Edsger Dijkstra, esse conceito voltou a ganhar relevância nos anos 1990, impulsionado pelo crescimento dos sistemas empresariais. Diversos conceitos acerca da arquitetura de software foram desenvolvidos ao longo do tempo, e, em última análise, esses conceitos se complementaram, contribuindo para uma compreensão mais abrangente e integrada do tema.

Para IBM (*International Business Machines*), a arquitetura de um sistema de software é definida por um conjunto de decisões essenciais relacionadas à organização do sistema, à escolha dos elementos estruturais e suas interações, bem como ao comportamento detalhado nas interações entre esses elementos. Além disso, a arquitetura envolve a composição desses elementos em subsistemas de escala crescente e a definição de um estilo arquitetural que orienta toda essa estrutura (Kruchten, 2000).

A criação de uma arquitetura robusta e eficiente para o desenvolvimento de projetos de aplicativos móveis é essencial em um cenário tecnológico cada vez mais dinâmico e competitivo, observa-se que os aplicativos móveis têm um papel muito importante, proporcionando soluções inovadoras e acessíveis que atendem às

demandas dos usuários. O Escritório de Soluções Criativas (ESC), que está vinculado ao Núcleo de Inovação Tecnológica – NIT da UNITINS, busca otimizar seus processos através da implementação de uma arquitetura de software específica para seus projetos de aplicativos móveis. Este trabalho se insere junto da crescente necessidade de metodologias e estruturas que facilitem o desenvolvimento de aplicativos eficientes, escaláveis e fáceis de manter.

Este estudo se concentra na criação de uma arquitetura de software específica para o ESC, considerando as particularidades dos projetos de aplicativos móveis desenvolvidos pelo núcleo. A pesquisa aborda desde os conceitos fundamentais de arquitetura de software até a aplicação prática desses conceitos no ambiente do ESC. A delimitação inclui a análise dos fundamentos sobre arquitetura de software, análise sobre tecnologias emergentes, design de software, DevOps para aplicações moveis e UI/UX, buscando as práticas que melhor se adequam ao desenvolvimento de aplicativos móveis no contexto do ESC.

METODOLOGIA

A presente pesquisa é caracterizada do tipo bibliográfica e documental, quanto ao procedimento, e de abordagem de estudo de caso. Em definição a pesquisa bibliográfica, Lakatos e Marconi (2003) descrevem que se trata do tipo de pesquisa que pode ser consultada de forma pública através de publicações em diversos meios de comunicação como monografias, teses, pesquisas, livros, artigos, revistas, jornais, etc.

A pesquisa bibliográfica é um apanhado geral sobre os principais trabalhos já realizados, revestidos de importância, por serem capazes de fornecer dados atuais e relevantes relacionados com o tema. O estudo da literatura pertinente pode ajudar a planificação do trabalho, evitar publicações e certos erros, e representa uma fonte indispensável de informações, podendo até orientar as indagações (Lakatos, Marconi, 2003, p. 158).

Assim, a pesquisa bibliográfica realizada neste estudo baseou-se em artigos, monografias e livros pertinentes ao tema abordado, que é a arquitetura de software para aplicações móveis. O método de pesquisa, incluindo o uso de sites de busca e critérios de exclusão, é detalhado a seguir. As buscas foram conduzidas por meio da revisão da literatura disponível em artigos e publicações no Google Acadêmico, visando

a familiarização com os documentos seguida da seleção criteriosa de cada um, em conformidade com o escopo da pesquisa.

Quadro 1: Pesquisa Bibliográfica Realizada.

PESQUISA	CRITÉRIOS	RESULTADOS
“Arquitetura de software” AND “design de software” AND “componentes de software”	Acesso aberto e publicação a partir do ano de 2020.	26
“Arquitetura de software” AND “desenvolvimento móvel”		28
“Arquitetura de software” AND “multiplataforma”		239
"Architecture software" AND “software design” AND “software components”		250
"Architecture software" AND “mobile development”		22
"multi-platform development" AND "mobile development"		37
TOTAL		602

Fonte: Elaborado pelo Autor (2024).

Após realizar a pesquisa e a seleção, a plataforma catalogou os resultados, durante essa busca, os documentos foram pré-selecionados com base na leitura da introdução e do resumo, buscando identificar semelhanças com os objetivos estabelecidos para a pesquisa.

Após a etapa inicial de seleção, foram reunidos 15 documentos de pesquisa. Cada um foi lido minuciosamente para absorver o conhecimento que continha, incluindo 9 em inglês e 6 em português. Do ponto de vista acadêmico, todos os documentos foram relevantes, porém, alguns se destacaram ao influenciar particularmente a formação desta pesquisa.

A pesquisa documental é semelhante à pesquisa bibliográfica, conforme descrito por Matos e Vieira (2001), pode ocasionalmente haver confusão entre ambas, para distingui-las, é importante ressaltar as principais características da pesquisa documental em comparação com a bibliográfica. A diferença está na falta de análise desses materiais, que são provenientes de fontes variadas e dispersas, como jornais, revistas, relatórios, filmes, entre outros.

A pesquisa documental foi conduzida no contexto de um projeto, *Mestres Empreendedores* promovido pelo Escritório de Soluções Criativas (ESC) no ano de 2022. A pesquisa incluiu a análise de documentos relevantes, foram estudados, o

regimento interno do ESC disponível no site da UNITINS, e o regimento do NIT, também disponível no mesmo site. Esses documentos forneceram uma base sólida de entendimento sobre a estrutura e os procedimentos administrativos do ESC, permitindo uma integração mais eficaz da proposta de arquitetura no contexto organizacional.

Esta pesquisa visa não apenas fornecer um produto de qualidade tecnológica, mas também gerar um material significativo para a criação de aplicativos móveis no ESC. Portanto, busca-se oferecer um material tecnológico que destaque e valorize o desenvolvimento de software no núcleo, contribuindo assim para a criação de novos aplicativos, ampliando as possibilidades de aprendizado e enriquecendo o aspecto profissional.

Fundamentos da Arquitetura de Software para Aplicativos Móveis

Arquitetura de Software conforme posto por Prajapati & Geem, (2020), refere-se à estrutura fundamental de um sistema de software, abrangendo componentes de software, suas propriedades visíveis externamente e os relacionamentos entre esses componentes. Para Nouman *et al.*, (2023) ela desempenha um papel crucial em todas as etapas do desenvolvimento do sistema, orientando as decisões de Design e garantindo que o sistema atenda aos requisitos funcionais e não funcionais.

As aplicações móveis integraram-se profundamente ao nosso cotidiano, facilitando uma vasta gama de atividades, podemos observar o funcionamento destas aplicações em atividades rotineiras, desde o momento em que acordamos com o alarme do celular, ou quando fazemos o pedido de um lanche com aplicativos de *delivery*, até tarefas mais complexas como a gestão empresarial.

Conseqüentemente, os desenvolvedores têm uma tarefa difícil de atender às elevadas expectativas dos usuários em termos de usabilidade, acessibilidade, disponibilidade e confiabilidade, esses e outros aspectos que influenciam no desenvolvimento técnico e operacional de um software são influenciados pela estrutura da arquitetura desenvolvida. Partindo desse ponto, considerando o desafio do desenvolvedor, é notável a importância de uma arquitetura de software na afirmação de que, “No contexto de aplicações móveis, onde agilidade, escalabilidade e

experiência do usuário são fundamentais, a escolha da arquitetura de software torna-se ainda mais crucial.” (Bilogub, Skrypchenko & Tytenko. 2023).

De forma a exemplificar, considere a existência de um requisito de software, que dependa de políticas governamentais, leis, decretos, emendas entre outros, ou algo que não foi completamente definido durante o levantamento de requisitos. Nos dois casos não sabemos por quanto tempo a funcionalidade desenvolvida irá atender ao seu propósito, gerando assim o que podemos chamar de cenários voláteis, que por vezes exigem que os arquitetos de software não apenas considerem as funcionalidades presentes desejadas pelo aplicativo, mas também sua capacidade de se ambientar e evoluir em resposta a mudanças imprevistas no ambiente de execução. Para garantir estes cenários, “é necessária uma espinha dorsal de TI estável com uma arquitetura flexível e modular que permita a implementação rápida e fácil de novos módulos” (URBACH; AHLEMANN, 2019, *apud* ALT *et al.*, 2020), assim, a arquitetura de software não deve ser vista como um conjunto improgressivo de diretrizes, mas como uma estrutura dinâmica que guia o desenvolvimento e a manutenção do aplicativo ao longo de seu ciclo de vida.

Uma arquitetura de software bem planejada é fundamental para o sucesso de projetos de aplicativos móveis, visto que estabelece a organização base no que diz respeito ao qual o software é construído e evolui. A determinação criteriosa de padrões arquitetônicos e design *patterns* influencia de modo direto em aspectos como a usabilidade, desempenho e manutenção do aplicativo, razões estas que são decisivas para a aprovação e satisfação do usuário final. “A arquitetura de software é crucial para determinar o sucesso ou o fracasso em uma variedade de campos de desenvolvimento e design de software” (Baabad *et al.* 2022).

Tecnologias Emergentes e Inovação no Desenvolvimento Mobile

São tecnologias emergentes, aquelas que ainda não foram disseminadas por completo, mais que tenham uma forte possibilidade de mudar seu mercado no futuro, para o desenvolvimento *mobile*, existem diversas soluções interessantes, que podemos classificar em nativas e híbridas.

Aplicativos móveis podem ser desenvolvidos de duas maneiras principais: de forma nativa, específica para cada sistema operacional, ou de forma híbrida, que

permite a coexistência entre diferentes plataformas. O desenvolvimento nativo refere-se à criação de aplicativos especificamente para um único sistema operacional, como Android ou iOS, essa abordagem aproveita ao máximo os recursos e capacidades do sistema, resultando em um desempenho superior e uma experiência de usuário mais integrada e fluída. No entanto, conforme descrito por Bittencourt (2021), essa especialização exige o desenvolvimento de códigos diferentes para cada plataforma, o que pode aumentar os custos e o tempo de desenvolvimento.

Por outro lado, é na desvantagem do desenvolvimento na forma nativa que nasceu o desenvolvimento híbrido, segundo Bittencourt (2021), essa abordagem nos permite desenvolver um aplicativo que irá funcionar nas duas plataformas, isso otimiza o processo de desenvolvimento, todavia é possível que haja dificuldades para implementar funcionalidades específicas de uma plataforma.

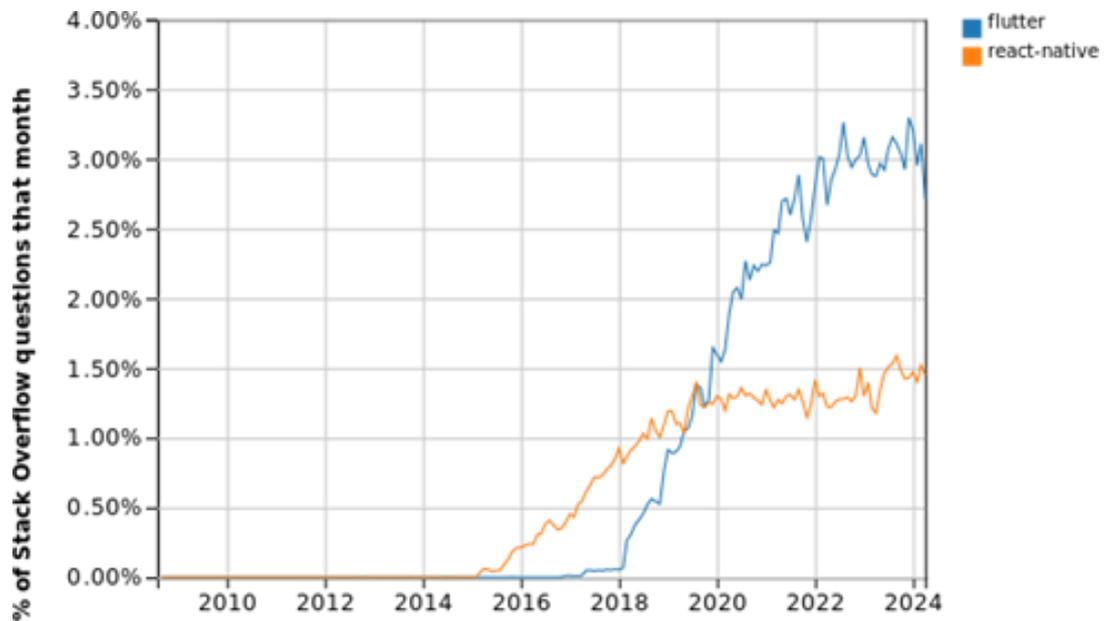
Para Silva e Batista (2023) o fator principal para o surgimento do desenvolvimento híbrido, foi a alta demanda por atualizações e novos aplicativos para novas tecnologias, foi essa brecha no mercado que alavancou projetos como novos frameworks que vêm inovando o desenvolvimento de aplicativos móveis, atendendo às crescentes demandas por soluções mais eficientes e versáteis, entre esses frameworks, destacam-se o Flutter, desenvolvido e mantido pela Google, e o React Native desenvolvido e mantido pela Meta, ambos permitem a compilação para diferentes sistemas operacionais a partir de uma única base de código, otimizando o processo de desenvolvimento.

Essas tecnologias não só facilitam a criação de aplicativos de alta qualidade, como também se tornaram as mais utilizadas no mercado atual, refletindo sua eficácia e popularidade entre os desenvolvedores, essa fomentação pode ser notada no Stack Overflow que é uma plataforma gratuita de perguntas e respostas voltada para comunidade de programadores e entusiastas da tecnologia, criada em 2008 a plataforma conta com mais de 58 milhões de perguntas e respostas, também acumula um total de 51 bilhões de vezes que conhecimentos forma reutilizados, informações disponíveis na plataforma pelo link <https://stackoverflow.co/>.

Na Figura 1 disponibilizada a seguir, que foi elaborada a partir do Stack Overflow Trends, que é uma ferramenta que analisa os interesses em linguagens e tecnologias, baseada no número de perguntas feitas por mês na plataforma Stack

Overflow, foi comparado os dois principais frameworks para desenvolvimento híbrido de aplicativos móveis no período de 2011 a 2024.

Figura 1: Evolução ao Longo do Tempo Entre Flutter e React-Native.

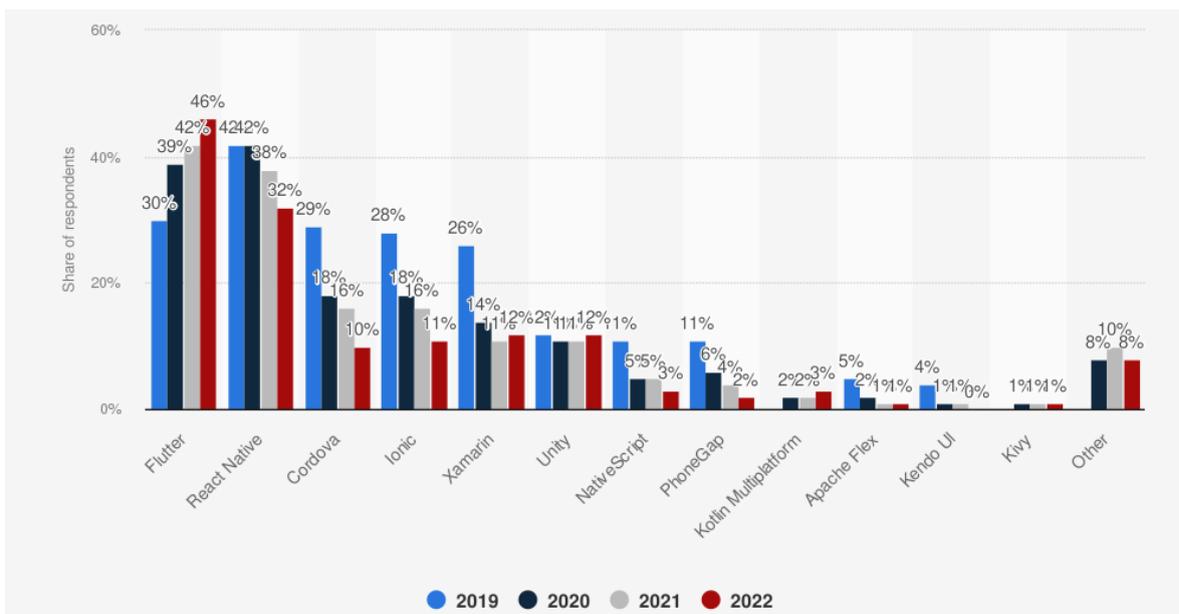


Fonte: Stack Overflow (2024), disponível em: <https://trends.stackoverflow.co/?tags=flutter,react-native>.

Essa análise indica uma crescente popularidade e adoção do Flutter em comparação ao React Native, refletindo as tendências e preferências da comunidade de desenvolvedores. Como mostrado na Figura 1, o gráfico evidencia que, a partir de 2018, o Flutter começou a crescer rapidamente, superando o React Native em termos de perguntas no Stack Overflow. Isso sugere que o Flutter está ganhando maior aceitação e suporte na comunidade, tornando-se uma escolha preferida para o desenvolvimento híbrido de aplicativos móveis.

De acordo com Statista, plataforma global de dados e inteligência empresarial, que conta com 18 milhões de visitas mensais, mais de 4 milhões de usuários registrados, 22.500 fontes renomadas, está presente em 170 indústrias e tem uma carreira de 16 anos, informações disponíveis no link <https://www.statista.com/aboutus/>. O Flutter se destaca como o framework móvel multiplataforma amplamente adotado pelos desenvolvedores ao redor do mundo, conforme constatado na pesquisa realizada entre 2019 e 2022 com a representação gráfica disponível a seguir.

Figura 2: Estruturas Multiplataforma Mais Usadas no Mundo.



Fonte: Statista (2024), disponível em:

<https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>.

Os dados, representados na Figura 2, revelam que 42% dos profissionais de desenvolvimento de software utilizaram o Flutter em 2022, superando o React Native, que foi utilizado por 38% dos desenvolvedores, esse aumento na adoção do Flutter pode ser atribuído à sua eficiência e facilidade de uso, tornando-o uma escolha popular para o desenvolvimento híbrido de aplicativos móveis.

Assim como os frameworks vem ganhando espaço no desenvolvimento mobile a IA (Inteligência Artificial) também tem seu espaço em vários segmentos, segundo Wang (2020) ela tem sido fundamental para melhorar as funções de reconhecimento em dispositivos móveis, como o reconhecimento de imagem e de rosto, aprimorando a experiência do usuário e permitindo interações mais avançadas, como holografia e realidade aumentada.

Embora existam diversas inovações no segmento, faz-se necessária a compreensão de que alguns problemas de segurança e privacidade também ficam prementes no processo de utilização.

No entanto, o setor financeiro digital é caracterizado por um desafio significativo em relação à cibersegurança que surge com as aplicações móveis de fintech. As preocupações discutidas envolvem um conjunto diversificado de ameaças e vulnerabilidades que têm o potencial de

comprometer os dados dos usuários, as transações financeiras e a integridade geral dessas aplicações. Sobre os obstáculos encontrados pelas aplicações móveis de fintech, com um foco particular em violações de dados, roubo de identidade e fraude como preocupações notáveis. Além disso, a pesquisa enfatiza a necessidade de implementar medidas para garantir a segurança dos procedimentos de autenticação do usuário, criptografia de transações e fortes restrições de acesso para reduzir esses riscos. Além disso, o estudo conduzido por [14] examina extensivamente a natureza dinâmica do ambiente de ameaças, destacando especificamente o aumento de formas avançadas de ataques, como malware móvel e operações de phishing que visam especificamente os usuários de aplicações de fintech. Há uma ênfase na necessidade de vigilância contínua, aquisição de informações sobre ameaças e implementação de medidas de segurança proativas para lidar efetivamente com esses perigos em constante mudança. Os estudos juntos ressaltam a natureza complexa e diversificada das preocupações com a cibersegurança no domínio das aplicações móveis de tecnologia financeira, bem como os esforços contínuos necessários para garantir a segurança e a confiabilidade dessas tecnologias financeiras. (MUSTAPHA *et al.*, 2023).

O autor também discorre sobre as tecnologias como criptografia avançada, autenticação biométrica e detecção de anomalias impulsionada por IA observando que estas são recomendadas para mitigar tais riscos.

Design e Modularidade em Arquiteturas de Software

A modularidade em arquiteturas de software é um princípio essencial que proporciona maior flexibilidade e compreensão no desenvolvimento de sistemas. Este design permite a divisão de um sistema em módulos independentes, facilitando a manutenção, a escalabilidade e a reutilização de componentes, além disso, a modularidade contribui para a redução da complexidade e para a melhoria da qualidade do software desenvolvido.

Para Rogozov et al., (2020) uma arquitetura modular de software envolve a estruturação de componentes de software em módulos independentes e intercambiáveis, sendo cada módulo responsável por executar um aspecto específico da funcionalidade desejada. A modularização de responsabilidades oferece muitos benefícios para o desenvolvimento de software, conforme demonstrado a seguir no Quadro 2.

Quadro 2: Benefícios da Modularização.

Compreensibilidade	Módulos bem definidos são mais fáceis de entender, pois cada um tem um propósito claro e limitado. Isso facilita a leitura e a análise do código.
Manutenção	Mudanças e atualizações podem ser feitas em módulos individuais sem afetar o restante do sistema. Isso reduz o risco de introduzir erros inesperados e torna a manutenção mais eficiente.
Reutilização	Módulos que realizam funções específicas podem ser reutilizados em diferentes partes do sistema ou mesmo em outros projetos, economizando tempo e esforço no desenvolvimento.
Testabilidade	A modularidade permite testar módulos isoladamente, tornando os testes de unidade e a depuração mais eficazes.
Escalabilidade	Adicionar novos recursos ou funcionalidades ao sistema é mais fácil quando o software é modular. Novos módulos podem ser criados e integrados conforme necessário.
Colaboração	Equipes de desenvolvimento podem trabalhar em módulos independentes ao mesmo tempo, facilitando a colaboração em projetos maiores.

Fonte: Evangelista (2023).

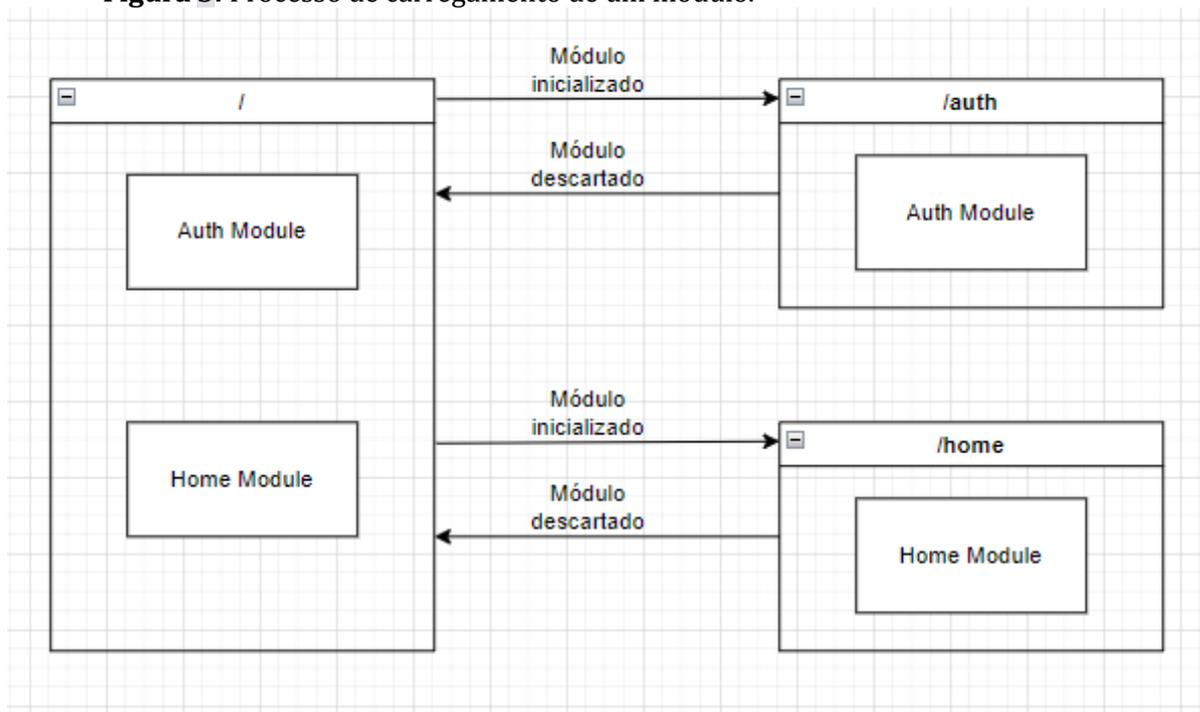
Em um sistema de compras online, onde o usuário é autenticado por um processo de acesso próprio, as funcionalidades podem ser divididas em módulos distintos. Cada módulo representa um caso de uso específico, como a adição de itens ao carrinho e a finalização da compra. Essa modularidade permite que cada componente seja reutilizado conforme necessário, facilitando a manutenção e a escalabilidade do sistema.

Além disso, é importante que, ao iniciar um aplicativo, não sejam carregadas todas as funcionalidades de uma vez, mas apenas aquelas que serão utilizadas. Essa abordagem não só otimiza o desempenho do aplicativo, mas também melhora a experiência do usuário, garantindo que os recursos sejam disponibilizados de forma eficiente e conforme a demanda. A modularidade, portanto, é essencial para o desenvolvimento de sistemas flexíveis e robustos, capazes de se adaptar a diferentes necessidades e contextos de uso.

Um sistema de navegação compatível é essencial, pois permite a correlação adequada com os módulos do aplicativo. A modularidade facilita o carregamento e descarregamento dos módulos à medida que o usuário utiliza o aplicativo, garantindo uma experiência mais eficiente e personalizada.

À medida que o usuário navega pelo aplicativo, os módulos necessários são carregados conforme a demanda, enquanto aqueles que não são mais necessários podem ser descartados. Alguns módulos, devido à sua importância e frequência de uso, podem permanecer carregados para uso imediato. Essa abordagem garante que os recursos estejam sempre disponíveis quando necessários, proporcionando uma experiência de usuário mais fluida e responsiva, conforme ilustrado na Figura 3.

Figura 3: Processo de carregamento de um módulo.



Fonte: Elaborada pelo autor (2024).

Segundo Kude *et al.*, (2023), ao dividir os componentes de software em módulos independentes, os desenvolvedores podem projetar e implementar cada módulo de forma independente e simultaneamente com outros componentes, aumentando a eficiência do desenvolvimento. A modularização facilita a incorporação de novas funcionalidades sem comprometer os módulos existentes, garantindo que o sistema permaneça estável e adaptável.

O desacoplamento proporcionado por essa abordagem permite a implementação de novas funcionalidades sem a necessidade de alterar códigos já desenvolvidos, o que contribui significativamente para a manutenção e evolução do software. Além disso, a modularidade tem um impacto direto na usabilidade, uma vez

que as melhorias podem ser implementadas de forma incremental e contínua, beneficiando diretamente a experiência do usuário final.

RESULTADOS E DISCUSSÃO

A implementação da arquitetura proposta resultou em uma estrutura de diretórios bem definida, facilitando a criação de funcionalidades concisas e modulares. Além disso, estabeleceu-se uma base de código genérica que incorpora métodos fundamentais de CRUD (*Create, Read, Update, Delete*). Esta organização promove maior clareza no desenvolvimento e manutenção do código, permitindo que os desenvolvedores identifiquem e modifiquem componentes específicos com maior facilidade. A padronização dos métodos CRUD proporciona uma abordagem consistente para operações de dados em toda a aplicação, reduzindo a duplicação de código e aumentando a eficiência no processo de desenvolvimento.

Para a realização deste projeto, o framework Flutter foi selecionado com base em um estudo comparativo prévio, que analisou diversas opções de desenvolvimento mobile multiplataforma. O Flutter se destacou como a escolha mais adequada, considerando fatores como eficiência no desenvolvimento, desempenho das aplicações resultantes e capacidade de atender às demandas específicas de múltiplas plataformas móveis. Sua adoção possibilitou a criação de uma base sólida para a implementação da arquitetura proposta, aproveitando suas características de desenvolvimento ágil e a capacidade de gerar aplicativos nativos para diferentes sistemas operacionais móveis a partir de um único código-base. Esta decisão reforçou a eficácia da estrutura organizacional e dos métodos padronizados, contribuindo para um processo de desenvolvimento mais eficiente e coeso.

A arquitetura implementada torna eficaz as práticas de desenvolvimento ágil e colaborativo. A integração de práticas Git no processo de desenvolvimento permite um controle de versão robusto e facilita a colaboração entre membros da equipe. Esta abordagem alinha-se com os princípios ágeis de desenvolvimento iterativo e entrega contínua, possibilitando uma resposta mais rápida às mudanças de requisitos e uma gestão eficiente do código-fonte.

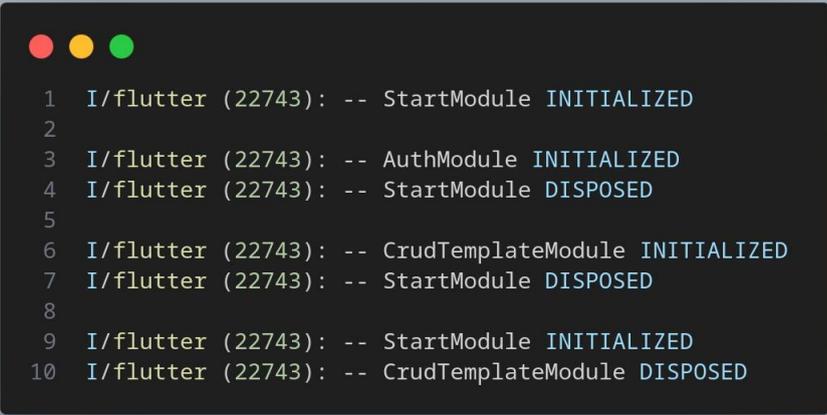
Um aspecto determinante da arquitetura é a adoção do sistema de modularização Flutter_modular. Esta escolha proporciona uma estrutura que facilita o

trabalho conjunto de equipes de desenvolvimento, permitindo a decomposição de tarefas em módulos distintos. Conseqüentemente, cada desenvolvedor pode trabalhar em sua feature de forma isolada, minimizando conflitos de código e promovendo um desenvolvimento paralelo mais eficiente. Esta modularização não apenas melhora a organização do código, mas também aumenta a reutilização de componentes e simplifica a manutenção a longo prazo.

A combinação destas práticas com a estrutura modular da arquitetura resulta em um ambiente de desenvolvimento que favorece a escalabilidade do projeto e a produtividade da equipe. Ao permitir que diferentes membros da equipe trabalhem simultaneamente em módulos separados, a arquitetura contribui para um fluxo de trabalho mais fluido e para uma distribuição mais eficaz das tarefas de desenvolvimento. Isso não apenas acelera o processo de criação de novas funcionalidades, mas também facilita a integração contínua e a entrega de incrementos de software de forma regular e consistente, alinhando-se assim com os princípios fundamentais das metodologias ágeis.

A modularidade, cujos benefícios foram previamente discutidos neste trabalho, foi efetivamente incorporada no código-fonte do projeto. Esta implementação se manifesta de forma prática na iniciação e descontinuação dinâmica dos módulos durante o tempo de execução da aplicação. Para ilustrar este processo, foram capturadas telas de log e da interface da arquitetura, que demonstram claramente o ciclo de vida dos módulos.

Figura 4: Logs do sistema de módulos da arquitetura.



```
1 I/flutter (22743): -- StartModule INITIALIZED
2
3 I/flutter (22743): -- AuthModule INITIALIZED
4 I/flutter (22743): -- StartModule DISPOSED
5
6 I/flutter (22743): -- CrudTemplateModule INITIALIZED
7 I/flutter (22743): -- StartModule DISPOSED
8
9 I/flutter (22743): -- StartModule INITIALIZED
10 I/flutter (22743): -- CrudTemplateModule DISPOSED
```

Fonte: Elaborado pelo autor (2024).

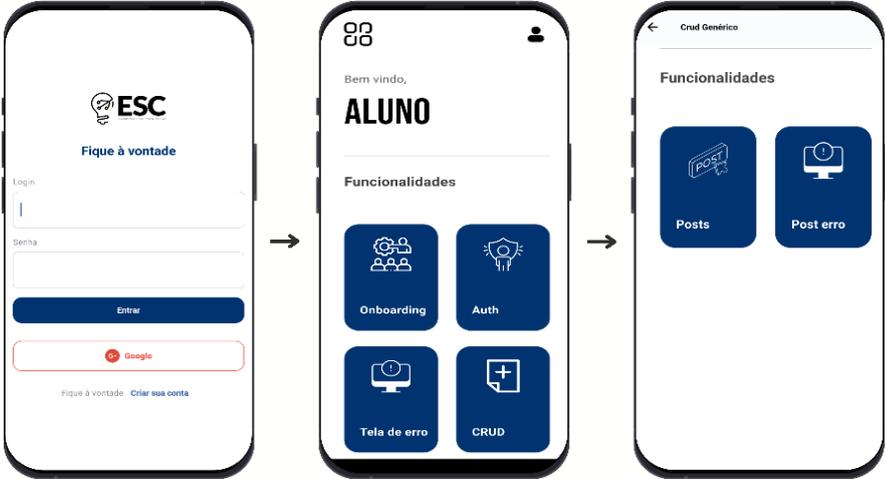
Estas evidências visuais mostram como os módulos são inicializados quando necessários e descartados (dispose) quando não mais utilizados, otimizando assim o uso de recursos do sistema. Este mecanismo de gerenciamento dinâmico de módulos está intrinsecamente ligado ao sistema de roteamento da aplicação. Quando uma nova rota é solicitada, o módulo correspondente é carregado, e quando o usuário navega para fora do escopo desse módulo, ele é adequadamente finalizado.

A integração entre o sistema de módulos e o roteamento da aplicação resulta em uma arquitetura altamente eficiente e responsiva. Esta abordagem não apenas melhora o desempenho geral do aplicativo, reduzindo a carga de memória, mas também facilita a manutenção e a escalabilidade do projeto. Os desenvolvedores podem adicionar, modificar ou remover funcionalidades de forma isolada, sem afetar o funcionamento global da aplicação.

Além disso, esta implementação modular proporciona uma separação clara de responsabilidades entre diferentes partes do aplicativo, alinhando-se com princípios de design de software como o Single Responsibility Principle. Isso resulta em um código mais limpo, mais fácil de entender e de manter, contribuindo para a longevidade e adaptabilidade do projeto em face de futuras mudanças ou expansões.

A arquitetura implementada exerce um impacto significativo na experiência do usuário final (UI/UX) do aplicativo. A modularização e o gerenciamento eficiente de recursos se traduzem em benefícios tangíveis para os usuários, como tempos de carregamento mais rápidos e uma navegação mais fluida entre as funcionalidades.

Figura 5: Principais telas do aplicativo.



Fonte: Elaborado pelo autor (2024).

A integração entre o sistema de módulos e o roteamento da aplicação proporciona transições suaves, resultando em uma experiência mais coesa e intuitiva. A estrutura modular também favorece a consistência da interface, permitindo a padronização e reutilização de elementos de UI comuns em diferentes partes do aplicativo. Além disso, a eficiência no uso de recursos garante um desempenho otimizado em diversos dispositivos, aspecto importante no desenvolvimento multiplataforma com Flutter.

Esta abordagem arquitetural não apenas facilita o desenvolvimento iterativo e responsivo às necessidades dos usuários, mas também contribui para uma experiência de usuário mais robusta, consistente e adaptável, alinhada com as expectativas modernas de interação móvel.

CONCLUSÃO

A implementação da arquitetura proposta para o desenvolvimento de aplicativos móveis no Escritório de Soluções Criativas (ESC) demonstrou resultados significativos em termos de eficiência, modularidade e experiência do usuário. A adoção do framework Flutter, combinada com práticas de desenvolvimento ágil e um sistema de modularização eficaz, resultou em uma estrutura que não apenas facilita o processo de desenvolvimento, mas também melhora a qualidade e a manutenibilidade do código produzido. Esta abordagem permitiu uma separação clara de responsabilidades entre os componentes do aplicativo, alinhando-se com princípios modernos de design de software e promovendo uma colaboração mais eficiente entre os membros da equipe.

Os resultados obtidos indicam um possível avanço na forma como os projetos de aplicativos móveis são desenvolvidos no ESC. A estrutura modular e o gerenciamento dinâmico de recursos não só otimizaram o desempenho das aplicações, mas também proporcionaram uma base sólida para o desenvolvimento iterativo e responsivo às necessidades dos usuários. Isso representa um passo importante na evolução das práticas de desenvolvimento de software no escritório, potencialmente servindo como um modelo para outras organizações que enfrentam desafios similares no desenvolvimento de aplicativos móveis multiplataforma.

As implicações práticas deste estudo são substanciais para o campo do desenvolvimento de software móvel. A arquitetura implementada oferece uma solução viável para os desafios comuns enfrentados no desenvolvimento de aplicativos, como a necessidade de escalabilidade, manutenibilidade e desempenho otimizado em diferentes plataformas. Além disso, a abordagem modular adotada facilita a incorporação de novas tecnologias e práticas à medida que surgem, permitindo que o ESC permaneça na vanguarda do desenvolvimento de aplicativos móveis.

Para futuras pesquisas, sugere-se uma investigação mais aprofundada sobre o impacto a longo prazo desta arquitetura na produtividade da equipe e na qualidade dos aplicativos desenvolvidos. Seria valioso explorar como esta abordagem pode ser adaptada para incorporar tecnologias emergentes, como inteligência artificial e realidade aumentada, mantendo a eficiência e a modularidade do sistema. Adicionalmente, um estudo comparativo entre esta arquitetura e outras abordagens de desenvolvimento móvel poderia fornecer insights valiosos para a comunidade de desenvolvimento de software, contribuindo para o avanço contínuo das práticas neste campo em rápida evolução.

REFERÊNCIAS

BAABAD, A. et al. Characterizing the architectural erosion metrics: A systematic mapping study. **IEEE access: practical innovations, open solutions**, v. 10, p. 22915–22940, 2022.

BILOGUB, D.; SKRYPCHENKO, M.; TITENKO, S. Quality attributes and architectural patterns of modern mobile apps. **Modern engineering and innovative technologies**, n. 29– 01, p. 33–38, 2023.

BITTENCOURT, JENIFFER. **Desenvolvimento de Apps Mobile: por onde começar?**. Alura, 29 dez. 2021. Disponível em: <<https://www.alura.com.br/artigos/desenvolvimento-apps-mobile-por-onde-comecar>>. Acesso em: 2 jun. 2024.

EVANGELISTA, M. B. **Arquitetura de software: Fundamentos e princípios - Marcos Barbosa Evangelista**. Disponível em: <<https://medium.com/@evangelistamarcos99/arquitetura-de-software-fundamentos-e-princ%C3%ADpios-3d9a0a969eb3>>. Acesso em: 31 maio. 2024.

KRUCHTEN, P. **The rational unified process: An introduction**. 2. ed. Boston, MA, USA: Addison Wesley, 2000.

ARQUITETURA MODULAR E ESCALÁVEL PARA DESENVOLVIMENTO DE APLICATIVOS MÓVEIS NO ESCRITÓRIO DE SOLUÇÕES CRIATIVAS. Rafael Aguiar SILVA; Jeferson Moraes da COSTA. JNT Facit Business and Technology Journal. QUALIS B1. ISSN: 2526-4281 - FLUXO CONTÍNUO. 2024 – MÊS DE OUTUBRO - Ed. 55. VOL. 01. Págs. 780-797. <http://revistas.faculdefacit.edu.br>. E-mail: jnt@faculdefacit.edu.br.

KUDE, T. et al. How deadline orientation and architectural modularity influence software quality and job satisfaction. **Journal of operations management**, v. 69, n. 6, p. 941–964, 2023.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

MATOS, K. S. L.; VIEIRA, S. L. **Pesquisa educacional: o prazer de conhecer**. Fortaleza: EDUECE/Demócrito Rocha, 2001.143p.

MUSTAPHA, I. et al. Cybersecurity challenges and solutions in the fintech mobile app ecosystem. **International Journal of Interactive Mobile Technologies (ijIM)**, v. 17, n. 22, p. 100–116, 2023.

NOUMAN, M. et al. A systematic review of non-functional requirements mapping into architectural styles. **Bulletin of Electrical Engineering and Informatics**, v. 12, n. 2, p. 1226–1236, 2023.

PRAJAPATI, A.; GEEM, Z. W. Harmony search-based approach for multi-objective software architecture reconstruction. **Mathematics**, v. 8, n. 11, p. 1906, 2020.

ROGOZOV, Y. I. et al. Method of designing the modular structure of the information system. **Journal of physics. Conference series**, v. 1457, n. 1, p. 012014, 2020.

SILVA, Lucas Morais; BATISTA, Messias Rafael. Análise de desenvolvimento de aplicativos usando plataformas nativas e multiplataformas. **Revista do IESP**, João Pessoa, v. 1, n. 1, p. 1-20, 2023. Disponível em: <https://www.iesp.edu.br/sistema/uploads/arquivos/publicacoes/analise-de-desenvolvimento-de-aplicativos-usando-plataformas-nativas-e-multiplataformas-autor-silva-lucas-morais-.pdf>. Acesso em: 16 maio. 2024.

URBACH; AHLEMANN, 2019, apud ALT et al. Software-defined business: Implications for IT management. **Business & information systems engineering**, v. 62, n. 6, p. 609–621, 2020.

WANG, Z. Using Artificial Intelligence to improve camera’s recognition function on mobile phone. **IOP conference series. Materials science and engineering**, v. 768, n. 5, p. 052023, 2020.