

# Aplicativo Via Cep Mobile/Web

Tiago Lima de Melo

07/06/2024

## Introdução

Este relatório apresenta o desenvolvimento de um aplicativo para consulta de CEP e endereços utilizando a API aberta ViaCEP. O projeto foi proposto como um teste técnico pela empresa 7Pay, destinado à seleção de candidatos. O objetivo do aplicativo é fornecer uma interface eficiente e amigável para a busca de informações de endereços a partir dos dados de CEP. Além da consulta por CEP, o aplicativo também permite a busca de informações fornecendo Unidade Federativa (UF), cidade e logradouro, contribuindo para a praticidade e rapidez no acesso a essas informações.

## Recursos para desenvolvimento

Conforme solicitado no exercício, foi utilizado o Flutter 3, em sua versão stable, para a implementação do projeto. A escolha do Flutter 3 deve-se à sua robustez e à capacidade de criar aplicações multiplataforma de forma eficiente.

Para estruturar o projeto, optei por utilizar a arquitetura limpa (Clean Architecture). Esta abordagem visa um maior desacoplamento entre as camadas do aplicativo, facilitando tanto a manutenção quanto a expansão futura do projeto. A Clean Architecture permite que cada camada tenha responsabilidades bem definidas e que o código seja mais modular e testável.

Além disso, para melhorar a organização do código, adotei a prática de encapsular métodos que dependem de pacotes externos. Essa estratégia tem como objetivo reduzir o impacto e o esforço de manutenção preventiva ou substituição desses pacotes, tornando o código mais flexível e resiliente a mudanças externas.

Embora opcional, também utilizei o MobX como gerenciamento de estado, conforme solicitado. O MobX proporciona um gerenciamento de estado reativo e eficiente, simplificando a sincronização entre o estado da aplicação e a interface do usuário.

## Recursos utilizados

Para a construção do aplicativo, foram utilizados diversos pacotes do Flutter, cada um com funcionalidades específicas que contribuíram para a robustez e eficiência do projeto. A seguir, detalho os motivos para a escolha de cada pacote:

- **flutter\_modular (versão 6.3.4):** Utilizado para a injeção de dependências e navegação modular no projeto. O Flutter Modular facilita a organização do código

em módulos independentes, promovendo um melhor desacoplamento e manutenção do código. Além disso, simplifica a navegação entre diferentes partes da aplicação, tornando-a mais intuitiva e escalável.

- **lottie (versão 3.1.2):** Utilizado para adicionar animações Lottie na aplicação. O pacote Lottie permite a integração de animações complexas e personalizadas de maneira eficiente, melhorando a experiência do usuário com animações suaves e atraentes.
- **mask\_text\_input\_formatter (versão 2.9.0):** Utilizado para formatação de entradas de texto. Este pacote é essencial para garantir que os dados de entrada do usuário sigam um formato específico, como CEP, números de telefone, entre outros. Isso melhora a usabilidade e a confiabilidade dos dados inseridos na aplicação.
- **mocktail (versão 1.0.3):** Utilizado para criar mocks nos testes unitários. O Mocktail é uma ferramenta importante para simular o comportamento de dependências externas durante os testes, permitindo que os desenvolvedores isolem e testem partes específicas do código de maneira eficaz e sem dependências reais.
- **responsive\_builder (versão 0.7.0):** Utilizado para construir layouts responsivos. O Responsive Builder facilita a criação de interfaces que se adaptam a diferentes tamanhos de tela e dispositivos, garantindo uma experiência de usuário consistente e otimizada tanto em dispositivos móveis quanto em telas maiores.