

FIAP 3SIS

Sistemas de Informação
Desenvolvimento Cross Platform
Professor Paulo Sabra

Checkpoint 06

Pokémon

VISÃO GERAL

O Flutter é uma poderosa ferramenta para o desenvolvimento cross-plataforma, oferecendo eficiência, reutilização de código e a capacidade de criar aplicativos de alta qualidade em várias plataformas. Com sua comunidade ativa e suporte contínuo do Google, o Flutter continua a ser uma escolha atraente para desenvolvedores que buscam uma solução abrangente e moderna para atender às demandas do mercado de aplicativos multiplataforma.

ESPECIFICAÇÕES

Com base no projeto "Pokédex", que foi desenvolvido durante as aulas, propõe-se a elaboração do projeto denominado "Pokémon". Para a identificação de cada Pokémon no projeto, os dois últimos dígitos do número de RM de cada participante devem ser utilizados.

Para a simplificação e otimização da arquitetura do projeto, recomenda-se o aproveitamento das bibliotecas, já disponíveis no projeto, as quais demonstraram ser eficazes para o desenvolvimento. Além disso, é crucial fazer uso da Profile "Flutter" disponibilizada no ambiente de desenvolvimento Visual Studio Code (VS Code) a fim de garantir a qualidade e desempenho do projeto.

Link do Repositório <https://github.com/paulosabra/pokedex>

Exemplo: **RM 00012 resulta em um Pokémon com o identificador 12**

ATIVIDADES

Desenvolvimento do Projeto

A fim de garantir uma abordagem sistemática no desenvolvimento do projeto, é recomendável seguir a seguinte sequência de etapas:

1. Primeiramente, proceda à atualização das classes pertencentes ao domínio Models, implementando as funções toJson() e fromJson().
Pode-se fazer uso da anotação @JsonSerializable(), semelhante à aplicada no projeto GitHub Explorer.
2. Em seguida, promova a atualização da função do serviço PokemonService, configurando a requisição à URL na API.
Nesse ponto, é possível realizar testes para verificar o funcionamento da requisição.
3. Posteriormente, crie os eventos (Events), estados (States) e a lógica associada no padrão Bloc, de modo a estruturar e gerenciar de forma eficiente o fluxo de dados do aplicativo.
4. Por fim, proceda à atualização da interface de usuário (Screen), substituindo o uso do componente FutureBuilder pelo Bloc, a fim de otimizar a manipulação e exibição dos dados.

A desenvolvimento seguindo essas etapas contribuirá para o desenvolvimento organizado e eficaz do projeto em questão.