

Interview Challenge

Objetivo

Implementar uma app que consome dados da API pública DummyJSON (<https://dummyjson.com/docs/products>), com foco em Clean Architecture, boas práticas de UI/UX e uso de tecnologias modernas do ecossistema mobile (Android / iOS).
A aplicação deverá conter as funcionalidades descritas a seguir, organizadas de forma modular e com código limpo, bem documentado e legível.

Tecnologias Esperadas

Cabe ao candidato gerir, dentro do tempo que tem, o balanço entre qualidade e completude. Ou seja, é compreensível que o candidato sacrifique qualidade para implementar todos os requerimentos ou não implemente todos os requerimentos para garantir uma melhor qualidade; também é compreensível que o candidato tire proveito de bibliotecas de domínio público e explique somente o que lhe pareça necessário à compreensão do que foi feito.

Cabe ao candidato definir o stack tecnológico a utilizar no desenvolvimento da aplicação sendo valorizada a utilização de MVVM/MVI, programação reativa e dos restantes elementos:

Android

- Compose
- Coroutines/Flow
- Retrofit

iOS

- Swift UI
- Combine

Setup e Entrega

- O projeto deverá conter um ficheiro README .md com:
 - Instruções de instalação, execução e compilação
 - Explicações do que foi implementado
 - Justificativas de decisões técnicas, se necessário
- É aceitável o uso de bibliotecas externas. Indique no README a sua função.
- Pode-se sacrificar partes da implementação para garantir melhor qualidade em outras, se documentado.

Funcionalidades

Funcionalidade 1 – Listagem Local com Paginação

- A app deve recolher a lista de produtos da API:
 - Endpoint: <https://dummyjson.com/products>
 - Pode-se utilizar paginação (`limit` e `skip`) ou obter todos os produtos de uma vez
- Os dados devem ser armazenados localmente
- O download deve ocorrer **apenas no primeiro arranque**
- Em caso de interrupção antes de completar o download, este deverá reiniciar no próximo arranque
- O ecrã de listagem deve exibir:

- Título do produto
- Rating
- Ícone (customizado por categoria de rating: <3, 3—4, >4)

Funcionalidade 2 – Pesquisa Avançada

- O ecrã de listagem deve ser pesquisável em tempo real
- A pesquisa deve permitir:
 - Busca por nome ou descrição
 - Termos fora de ordem
 - Insensibilidade a maiúsculas/minúsculas
 - Ignorar acentuação (diacríticos)
- Exemplo:
 - Produto: "Smartphone Samsung Galaxy"
 - Pesquisa: "galaxy samsung" ou "smart gal" → deve retornar o produto

Funcionalidade 3 – Detalhe do Produto

- Ao selecionar um item da lista, a app deve navegar para o detalhe
- Neste ecrã, deverão ser exibidos:
 - Título
 - Preço
 - Percentual de desconto
 - Stock
 - Rating
 - Imagem principal
- A imagem deve:
 - Estar ancorada no topo
 - Ter altura mínima
 - Ajustar dinamicamente ao scroll

Funcionalidade 4 – Formulário Validado

Criação de um formulário na app

- Criar um formulário com os seguintes campos:
 - Nome do utilizador
 - Email
 - Número (apenas dígitos)
 - Código promocional (apenas letras maiúsculas e hífen)
 - Data de entrega
 - Classificação (dropdown com: Mau, Satisfatório, Bom, Muito Bom, Excelente)
- Validações:
 - Nenhum campo pode estar vazio
 - Email com regex
 - Data:
 - Não pode ser uma segunda-feira
 - Não pode estar no futuro
 - Código promocional:
 - Apenas letras maiúsculas e hífen
 - Mínimo 3 e máximo 7 caracteres
 - Sem acentos

Notas Finais

- O projeto deve conter dois modos de execução:
 - Um com apenas listagem e detalhe
 - Outro com formulário.
- Pode-se usar `build flavors`, múltiplos `Views`, ou qualquer estratégia modular
- Todos os ecrãs devem ser compatíveis com diversos tamanhos de ecrã
- O teclado não pode ocultar campos do formulário