

PROVA TEÓRICA ANTI PADRÕES

ALUNO: LEONARDO BITTENCOURT ABREU SOARES

URL-GIT: <https://github.com/LeonardoBittencourt97/php>

O que é um Antipattern?

Design Patterns (Padrões de Projeto) são soluções para problemas recorrentes que encontramos no desenvolvimento ou manutenção de um software orientado a objetos.

Segundo os autores de Design Patterns, é preciso que haja pelo menos dois elementos chave para a distinção de um anti-padrão real de uma má ideia, mal hábito ou má prática.

1- Um processo, estrutura ou padrão de ação comumente utilizado que apesar de parecer inicialmente uma resposta adequada e eficaz para determinado problema, tem mais consequências ruins do que boas.

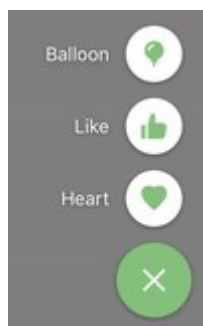
2- Existe outra solução documentada, reproduzível e comprovadamente eficaz.

IOS like-design in Android — Antipattern

Além de deixar o usuário confuso também afeta o desenvolvedor, isso consome muito tempo ou pode vir a fazer com que confie em bibliotecas de terceiros e assim retardar o desenvolvimento.

Fri May 10	11	00	
Sat May 11	12	01	
Sun May 12	1	02	AM
Today	2	03	PM
Tue May 14	3	04	
Wed May 15	4	05	
Thu May 16	5	06	

Outro exemplo é o botão flutuante do IOS:

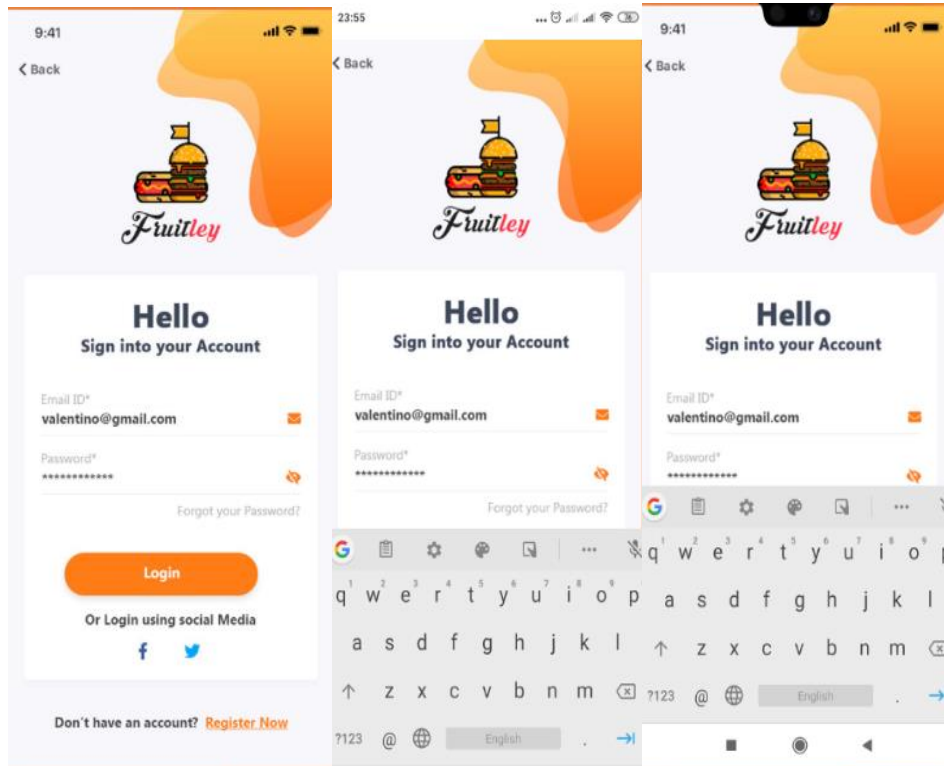


O Android não fornece uma funcionalidade expansível pronta para uso e a mecânica do Android Fab é diferente, de acordo com as diretrizes do Android "Um botão de ação flutuante (FAB) é um botão circular que aciona a ação principal na interface do usuário do seu aplicativo" uma seleção de ações secundárias.

Action buttons and a software sliding keyboard on Android — Antipattern

Dê uma olhada na imagem a seguir:

Então imagine que você precisa digitar algo no teclado



Agora repare na seta azul no canto inferior direito da segunda imagem.

Então vamos imaginar que nosso telefone não tenha apenas botões de hardware, mas também, talvez um recorte para a câmera frontal.

Observe como você tem apenas os campos que realmente precisa, já centralizados.

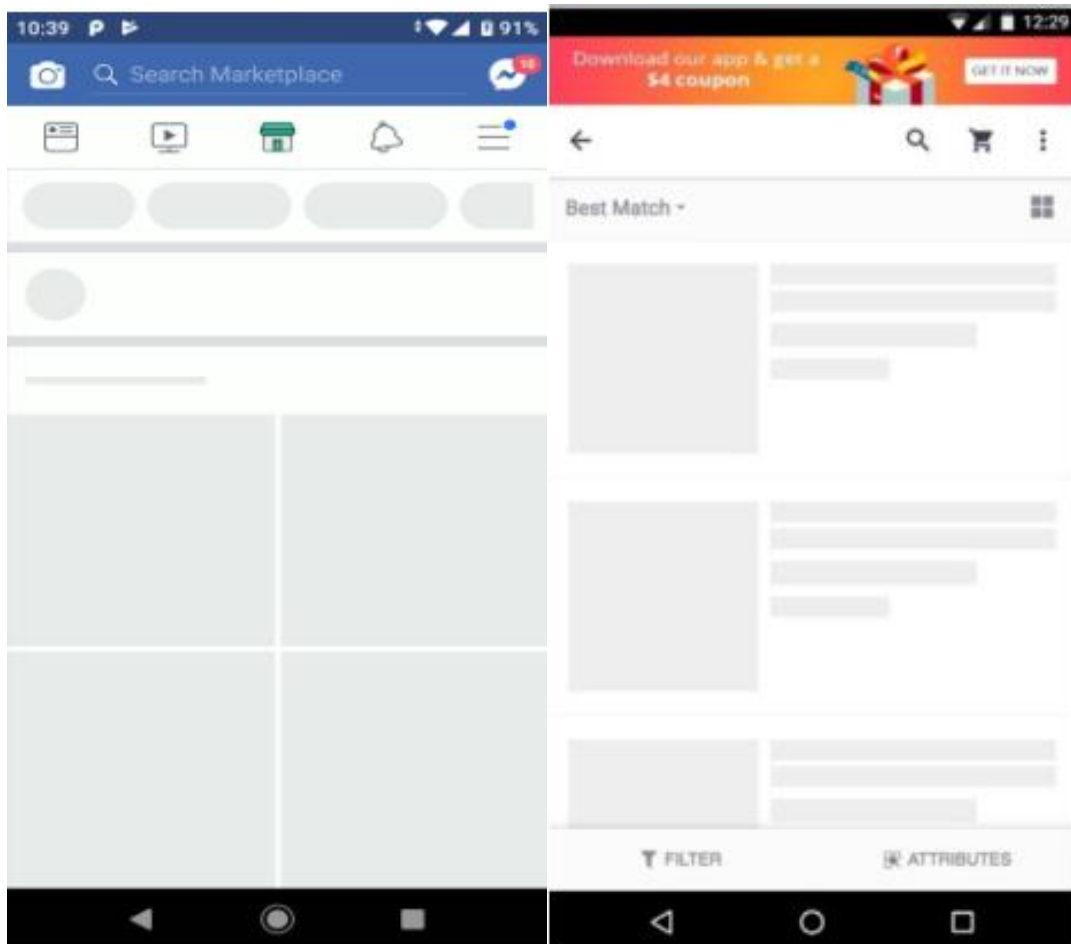
nem todos os telefones Android têm botões de hardware, portanto, qualquer coisa que seja possível clicar ou legível não deve estar diretamente na parte inferior da tela, caso contrário, talvez seja necessário redesenhar tudo em caso de descobrir no final ao entrar na fase de teste.

Um aplicativo que utiliza este antipattern é o Fruitley.

Skeletons instead of progress in Android — Antipattern

A ideia principal de Skeletons é que o usuário perca o interesse em três segundos, mas dê uma olhada no que ele está realmente esperando e ele pode ficar.

Skeletons cresceu partir da web, adotada por gigantes como o Facebook e também Aliexpress.



Então qual é a parte ruim?

Não há suporte nativo no Android, e isso não é um conceito de design de material.

Existem muitas bibliotecas de festas em 3D que permitem implementar o carregamento de skeletons, algumas delas até suportam o AndroidX e são escritas em Kotlin.

No entanto, vamos chegar a um entendimento com o que realmente precisamos, e isso agrupa todas as nossas visualizações para que o sistema desenhe as mesmas visualizações que caixas cinzas antes que os dados sejam carregados. E temos sorte que há uma biblioteca que faz exatamente isto, e funciona se a arquitetura do seu aplicativo seguir a arquitetura padrão “One-Activity-Multiple-

Fragments", que é o componente Navegação, se você use a biblioteca aparentemente otimizada para a memória, ela sempre armazenará o primeiro fragmento em cache e nunca o deixará ir; portanto, você sempre terá o spoiler errado no início de cada carregamento (como mostrar um grande círculo em uma tela em que você finalmente possui quadrados) , e não há nada pior do que enganar o usuário.

Então, o que você pode fazer, é necessário criar um layout duplicado para todos os tipos de tela existentes no seu aplicativo e usar o layout Shimmer do Facebook para animar todo o programa. Para piorar a situação, em caso de alguma mudança, não se esqueça de alterar o layout de carregamento do esqueleto duplicado.

Então pense duas vezes antes de concordar com a tarefa.