Caminhos para uma Arquitetura Limpa e Testável no Android

+800 congressistas
vagas limitadas

Curadoria:
GDG São Paulo

rafaeltoledo.net

@_rafaeltoledo

Desenv. Android @ Concrete Solutions

Arquitetura
x
Design

@_rafaeltoledo

Arquitetura
x
Design

Polêmico!

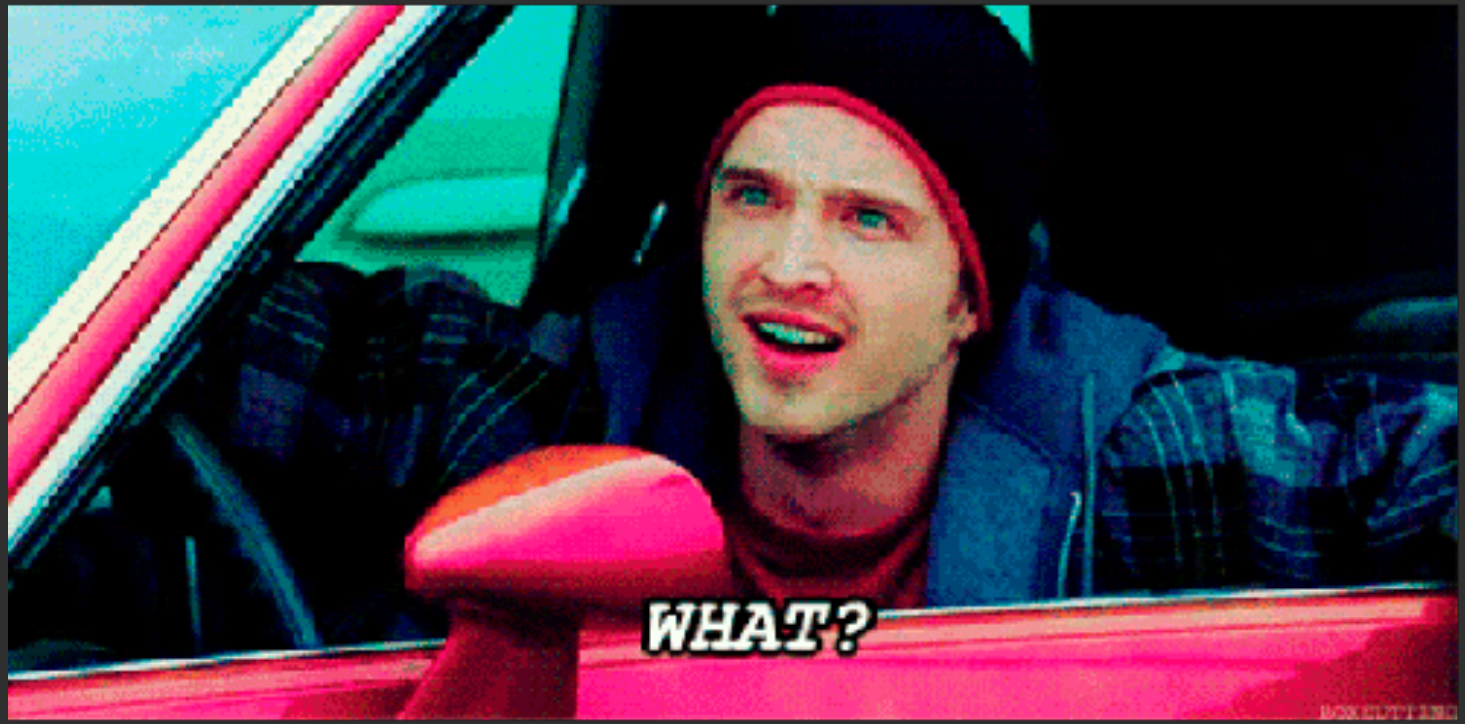
@_rafaeltoledo

"**Principais elementos** do sistema - as peças difíceis de mudar. Uma **fundação** no qual o resto precisa ser construído"

Martin Fowler

"É o conjunto de decisões de *design* que gostaríamos de ter tomado no início do projeto"

Ralph Johnson (GoF)



@_rafaeltoledo

"O design é feito sobre o que foi decidido pela arquitetura, por isso é difícil mudar a arquitetura. Design deve ser flexível ao máximo"

Neal Ford (Thoughtworks)

entendidos?

Arquitetura?


@_rafaeltoledo


@_rafaeltoledo

Arquitetura

por que se preocupar?

Porque ninguém quer mexer em código legado

Por que ninguém quer mexer em código legado?

O que é legado?

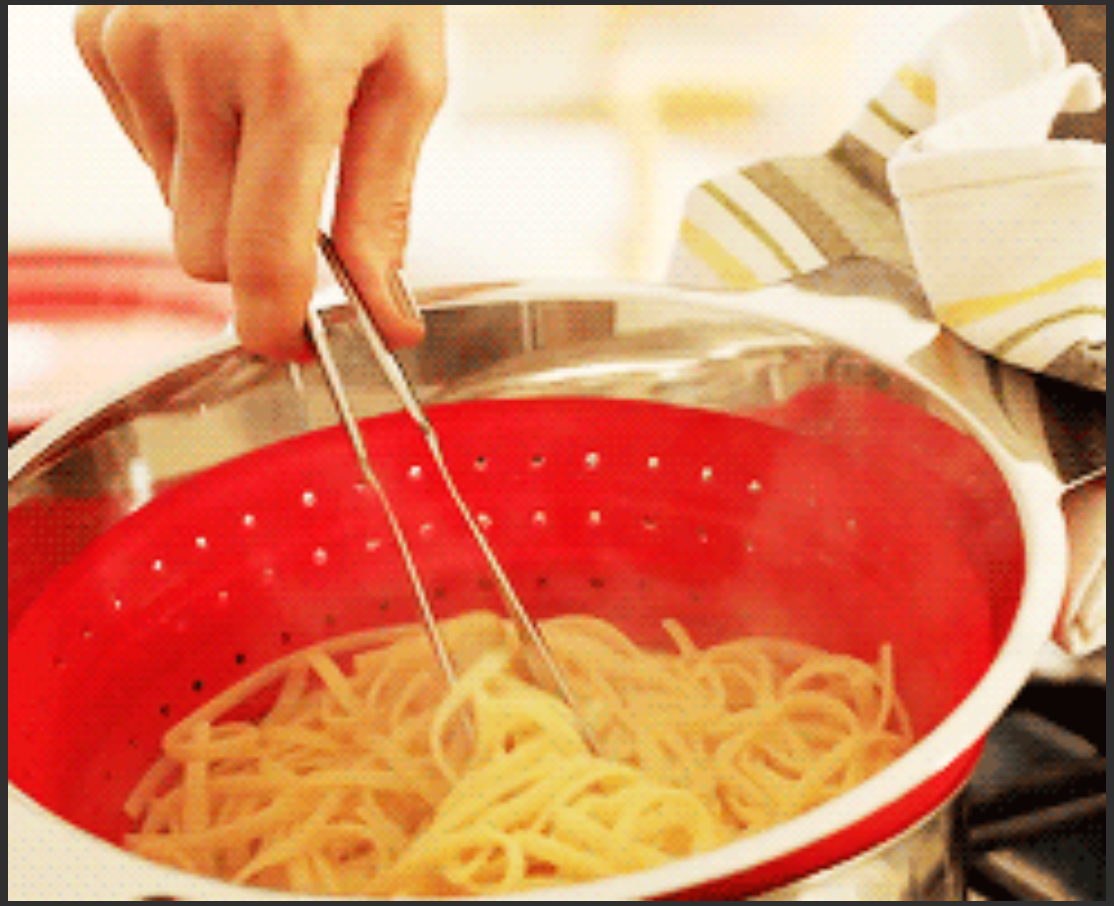

4GIFs.com

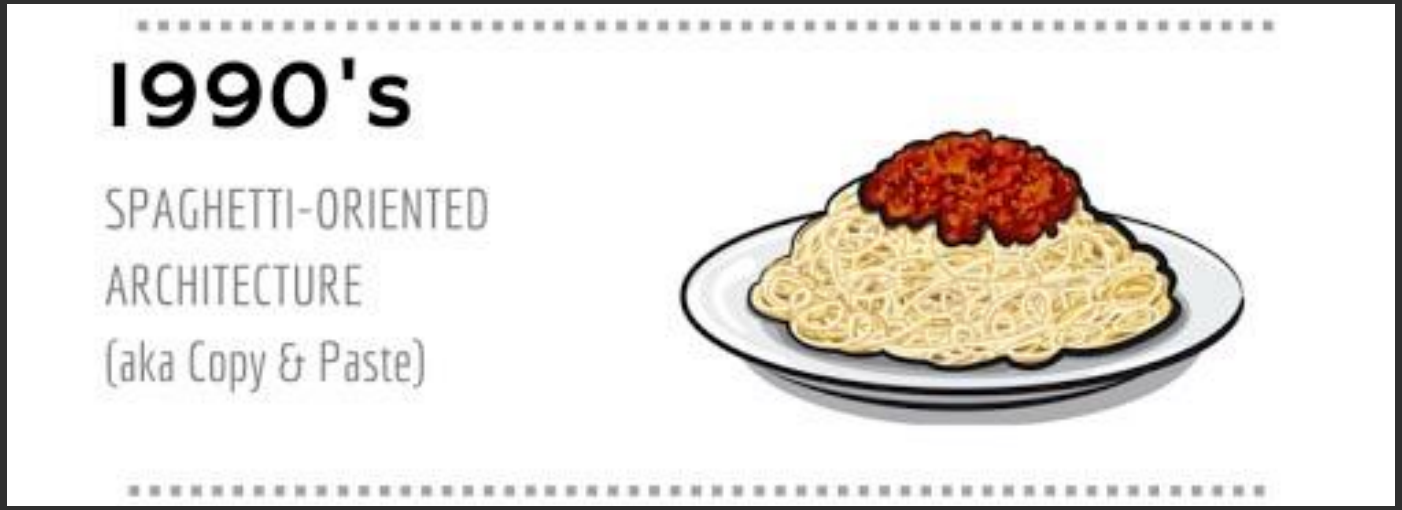
#0 Ausência de testes!

“Prefer simple, direct solutions to problems rather than creating a lot of infrastructure and abstractions to solve those problems.”

Chet Haase

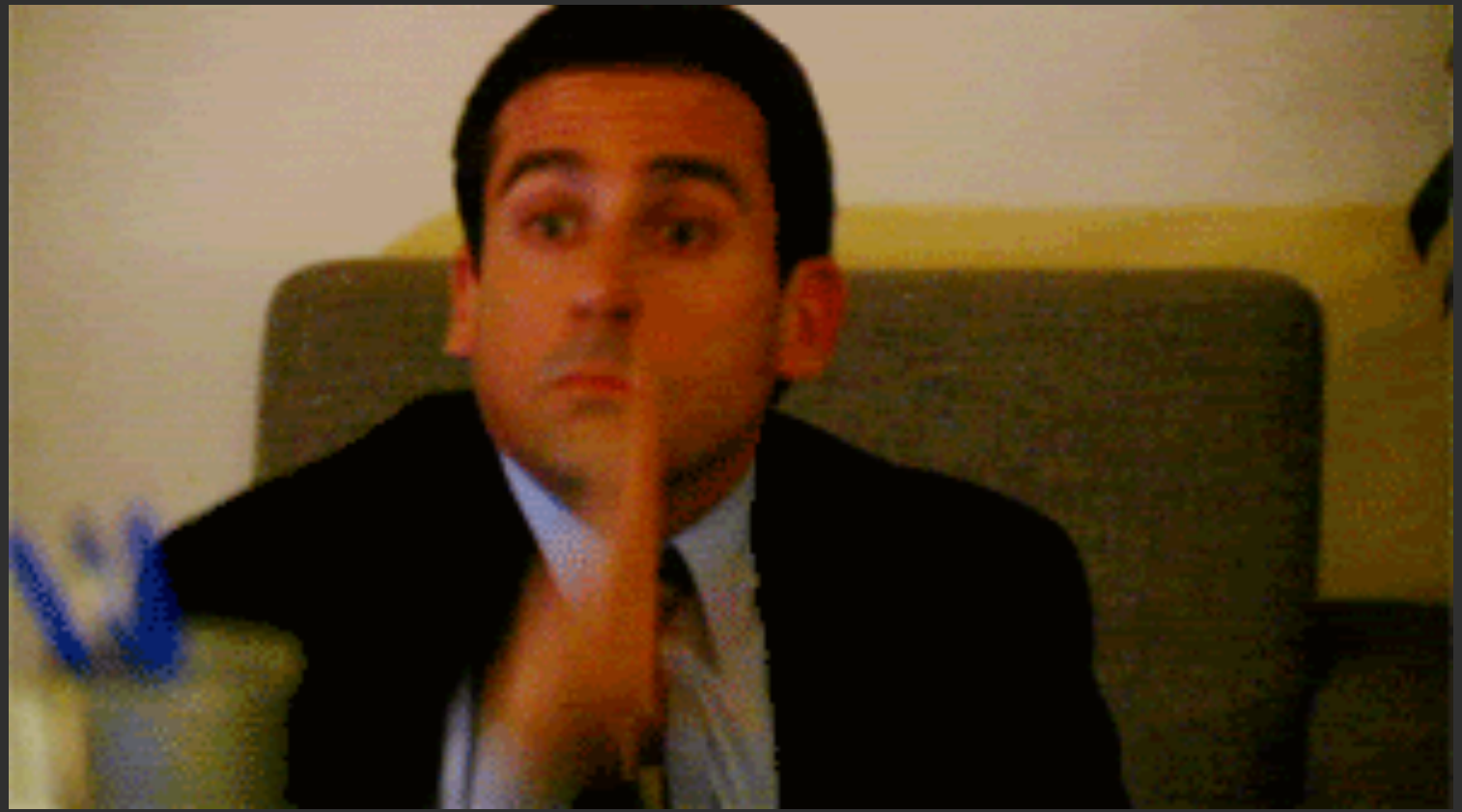
@_rafaeltoledo





@benorama

@_rafaeltoledo



@_rafaeltoledo

Arquitetura

por que se preocupar?

Por que se preocupar?

- Código mais fácil de manter!
- Rápido e fácil de testar
- Fácil de entender
- Desacoplado


Caminhos?
@_rafaeltoledo

MVP?

Viper?

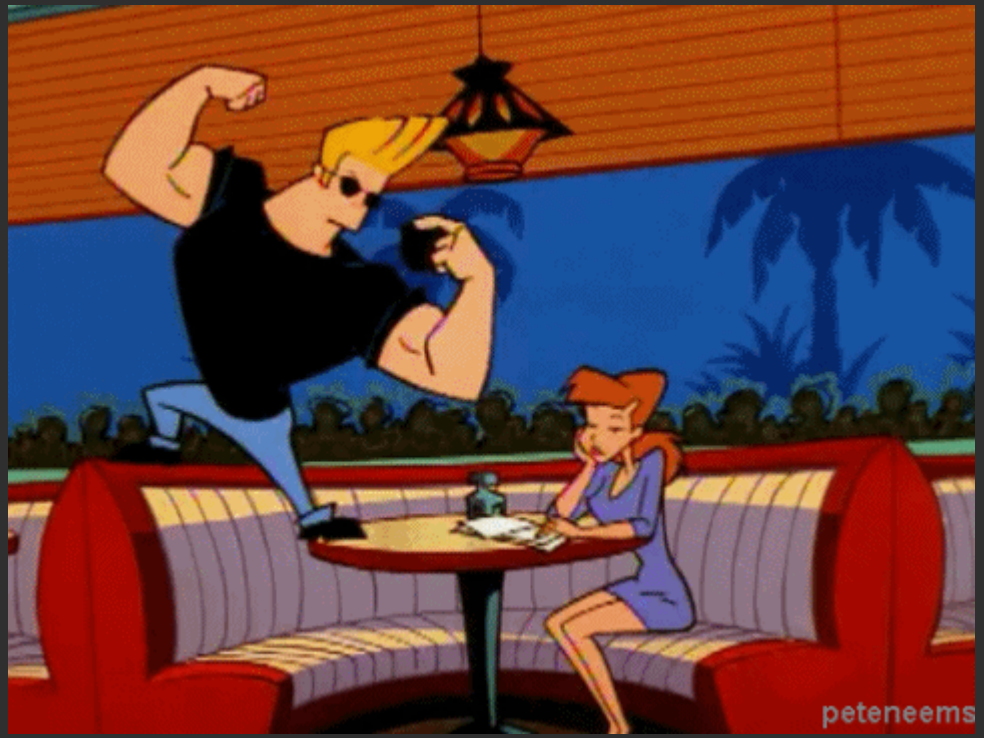
Clean?

MVC?

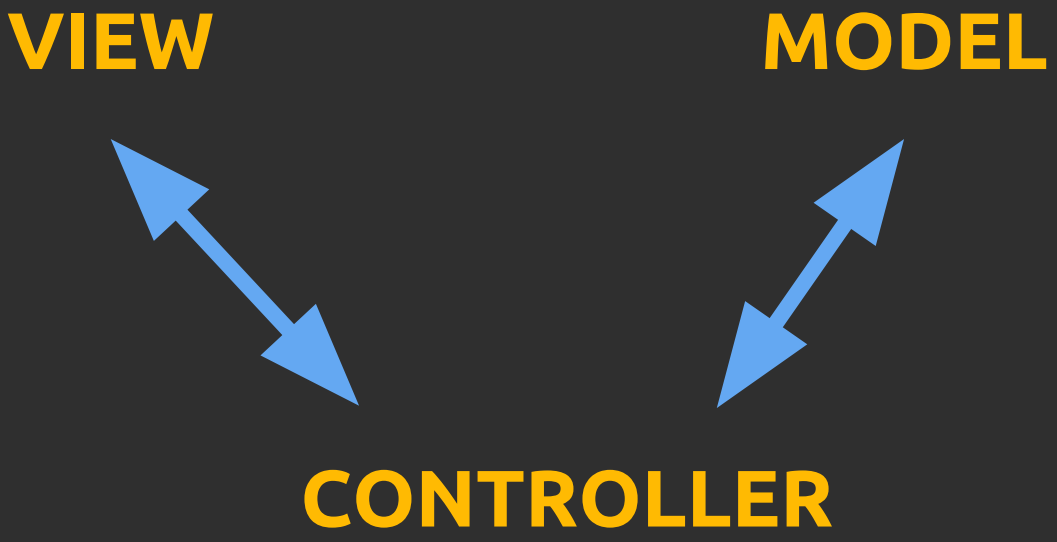
MVVM?

Flux?

@_rafaeltoledo



MVC


VIEW
MODEL
CONTROLLER

Model

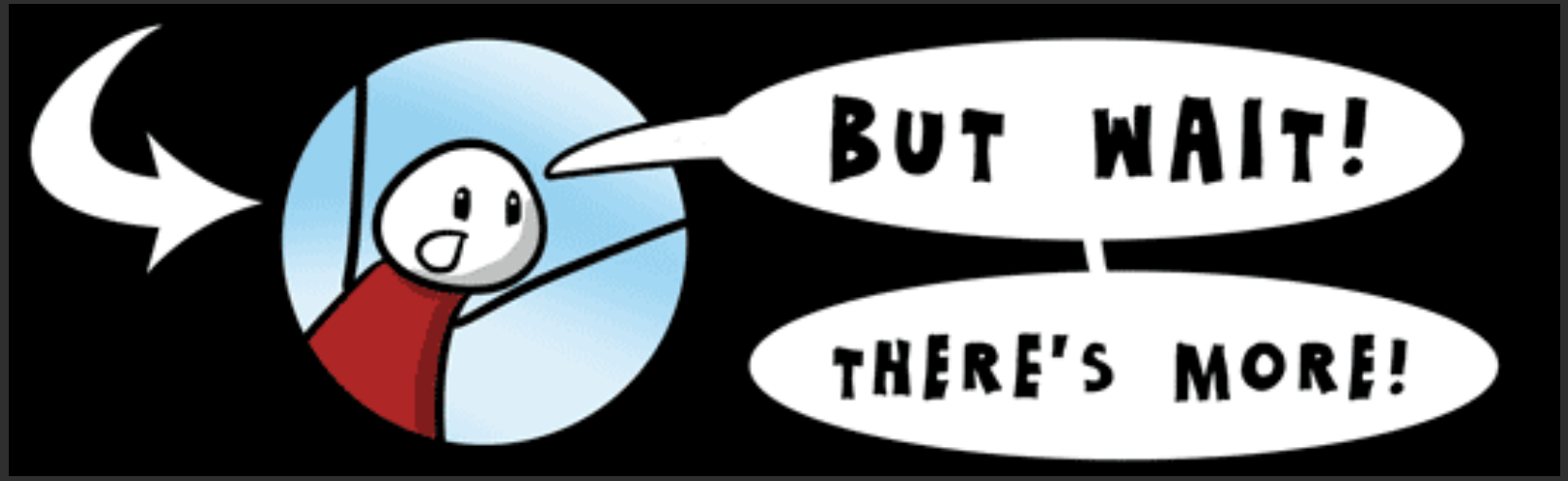
- Representar / Interagir com os dados

View

- Renderizar e apresentar os dados

Controller

- Tratar eventos vindos da View
- Atualizar o Model
- Gerenciar a navegação


BUT WAIT!
THERE'S MORE!

Contexto Mobile!

há mais a ser feito

@_rafaeltoledo

Model

- Representar / Interagir com os dados

- Interagir com a rede - API?

View

- Renderizar e apresentar os dados
- Gerenciar os estados das Views

Controller

- Tratar eventos vindos da View
- Atualizar o Model
- Gerenciar a navegação
- Interagir com componentes do sistema
- Gerenciar eventos do sistema
- Atualizar a View de acordo com os eventos do sistema

MVC no Android

- Pode ocorrer acúmulo de funções no Controller

MVC?

MVP!

MVP

- Não chega a ser um padrão arquitetural
- Adendo ao MVC
- Mostra uma maneira de estruturarmos a camada de apresentação de forma desacoplada

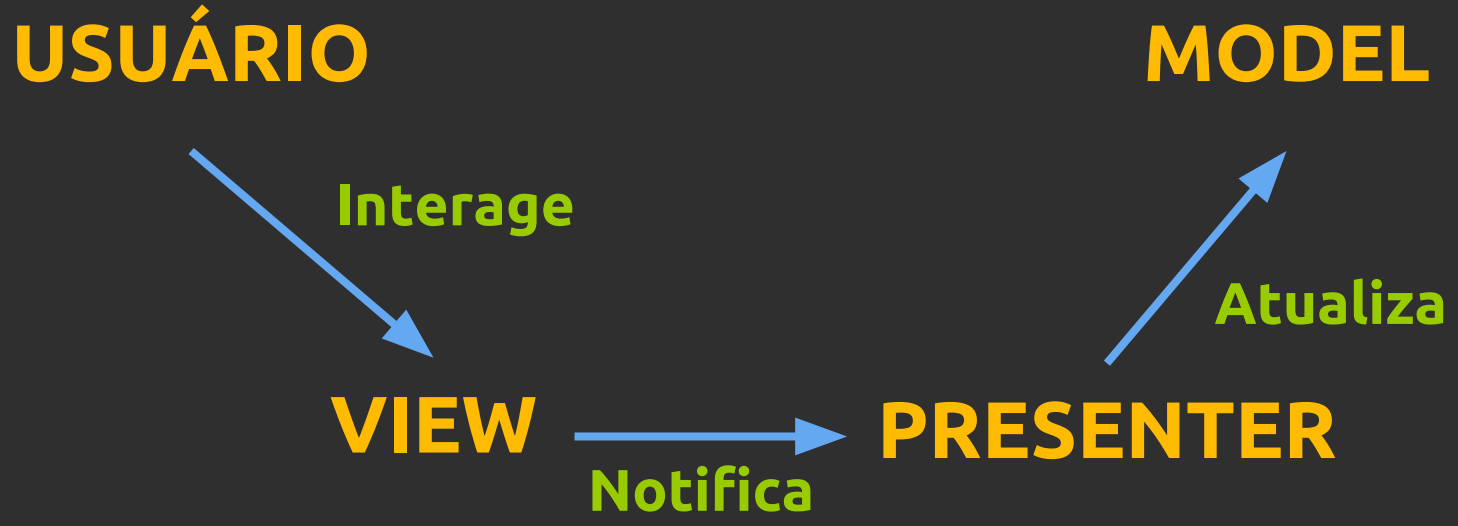
Presenter

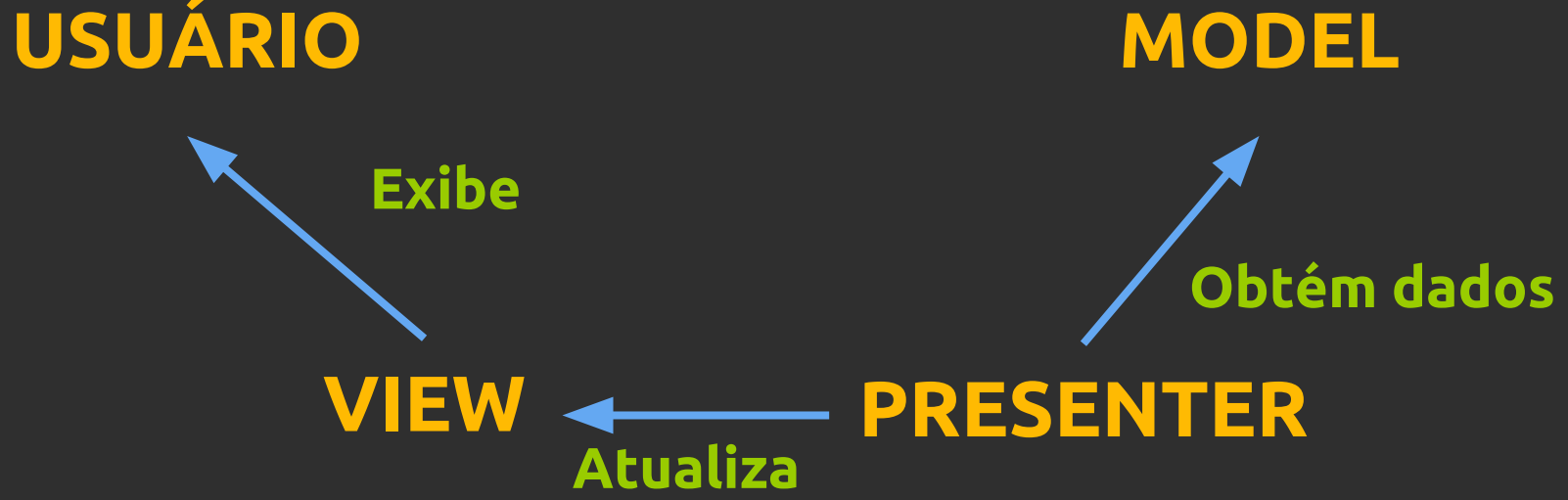
- Android costumeiramente mistura Model + View. Ex: CursorAdapter

- Precisamos de uma divisão clara de responsabilidades

Presenter

- Ponto de ligação entre o Model e as Views
- Mais fácil de testar as interações - interface
- De maneira geral, 1:1 View / Presenter - exceção views complexas


USUÁRIO
MODEL
Interage
Atualiza
VIEW
PRESENTER
Notifica


USUÁRIO
MODEL
Exibe
Obtém dados
VIEW
PRESENTER
Atualiza

MVP - Interactors

- Controla as interações do usuário e com as fontes de dados

- Modelo de callback / troca de mensagens

```java
public interface MyPresenter {

  void onResume();

  void onItemClicked(int position);
}
```

```java
public interface MyView {

  void showProgress();

  void hideProgress();

  void setItems(List<String> items);

  void showMessage(String message);
}
```

```java
public interface LoadItemsInteractor {

  void loadItems(OnItemsLoadedListener listener);

  interface OnItemsLoadedListener {

    void onLoaded(List<String> items);
  }
}
```

```
public class MyActivity implements MyView {

  MyPresenter presenter;
  ...
  @Override
  public void showProgress() {
    progressBar.setVisibility(View.VISIBLE);
  }

  @Override
  public void setItems(List<String> items) {
    adapter.addAll(items);
    adapter.notifyDataSetChanged();
  }
}
```

```java
public class MyActivity implements MyView {

  MyPresenter presenter;
  ...
  @Override
  protected void onResume() {
    super.onResume();
    presenter.onResume();
  }
}
```

```
public class MyPresenterImpl implements MyPresenter,
                        LoadItemsInteractor.OnItemsLoadedListener {

  public MyPresenter(MyView view, LoadItemsInteractor interactor) {
    this.view = view;
    this.interactor = interactor;
  }

  @Override
  public void onResume() {
    view.showProgress();
    interactor.findItems(this);
  }
}
```

MVP

- Como são interfaces, podem ser facilmente testados!

```
presenter.onResume();
verify(view, atLeast(1)).showProgress(); // Mockito

SomeInteractor mocked = mock(SomeInteractor.class);
when(mocked.loadFromNetwork()).thenReturn(Arrays.asList("object");
```

então MVP é só alegria?

@_rafaeltoledo

MVP no Android

- Use com cuidado, pois pode acabar se tornando overengineered!

Mas e o MVVM?

MVVM

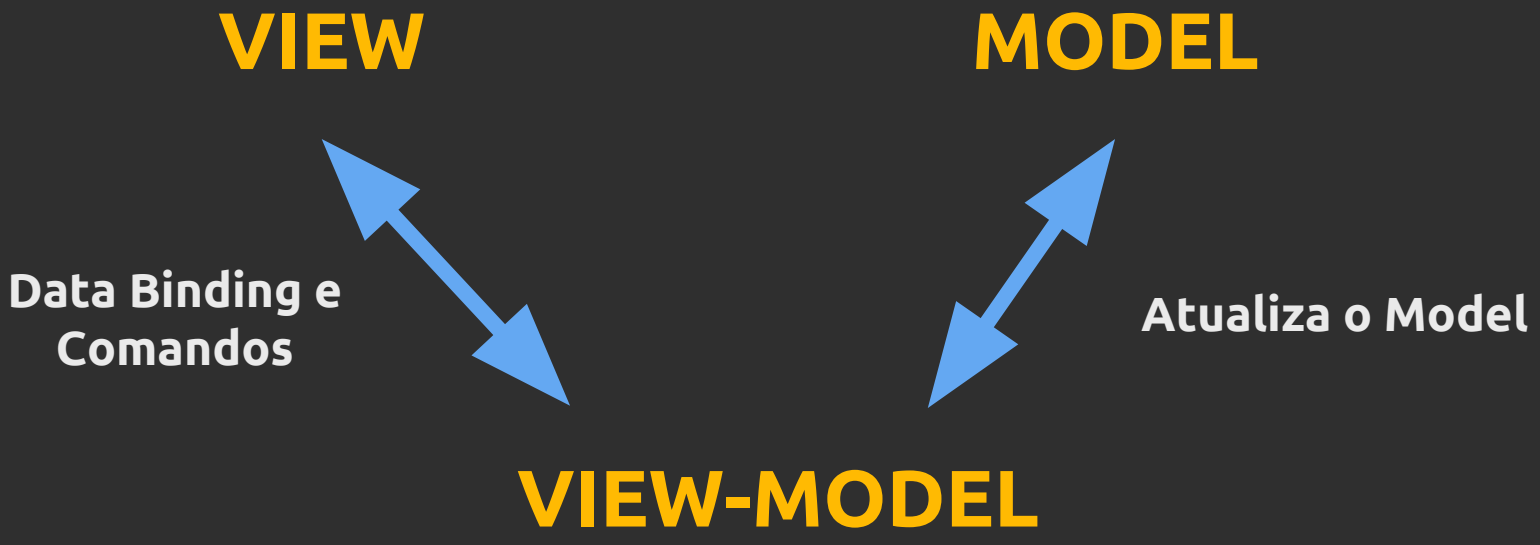
- Apareceu com mais força após o anúncio do DataBinding no I/O

- Existem outras libs de binding: RoboBinding, AndroidBinding, ...

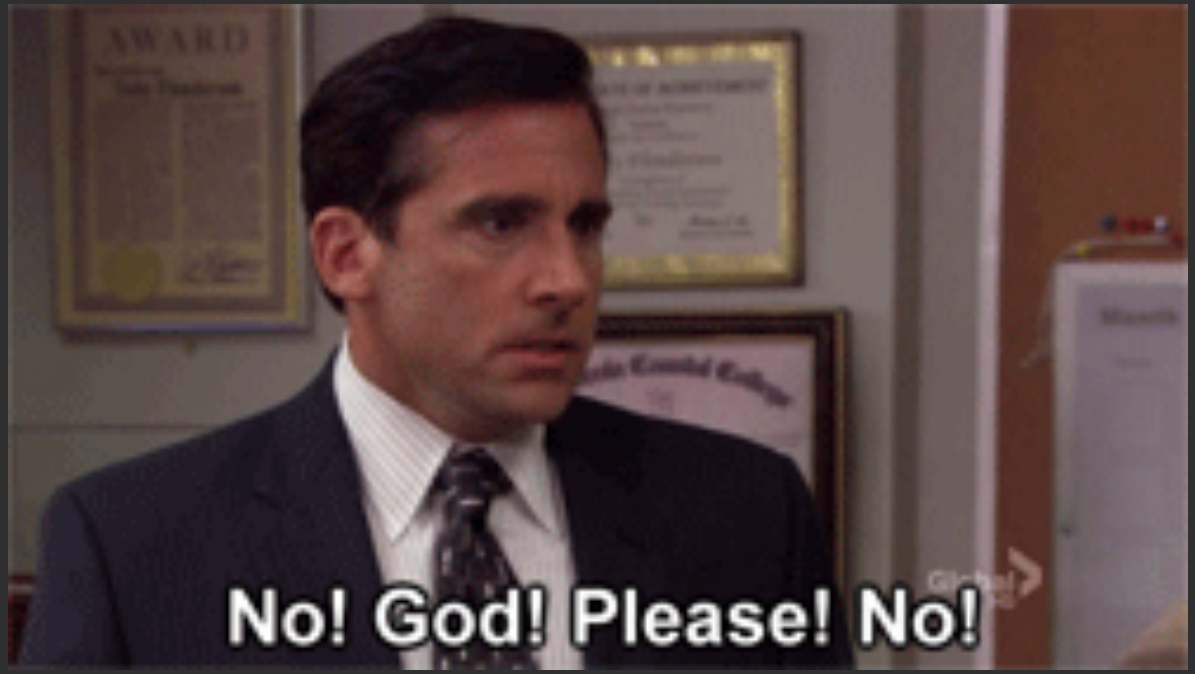
MVVM

- Comumente utilizado em ambientes Microsoft

- É quase um requisito o uso de alguma biblioteca, caso contrário haverá muito boilerplate


VIEW
MODEL
Data Binding e
Comandos
Atualiza o Model
VIEW-MODEL

Sinceramente?



@_rafaeltoledo

Nossa... por quê?

MVVM

- Códigos mais complexos tendem a ficar extremamente acoplados

- Testes? #comofas

Jake Wharton @JakeWharton · 21 de jul

Data binding is the fragments of 2015.

Ver tradução

RETWEETS	CURTIRAM
96	120

01:48 - 21 de jul de 2015 · Detalhes

Outras abordagens

- Clean Architecture
- Flux
- Viper (adaptado do iOS)

Arquitetura

algumas dicas finais

@_rafaeltoledo

Dicas

- Não seja extremista
- Sempre pense e escreva testes
- Um código difícil de testar é um indicador de que a arquitetura está meio capenga...

Dicas

- Desacople do framework

- Use Robolectric para testes unitários, Espresso / Robotium para navegação

No final das contas...

estamos buscando um código bacana!

@_rafaeltoledo

Links bacanas!

- antonioleiva.com/mvp-android
- fragmentedpodcast/episodes/11
- github.com/chiuki/friendspell
- github.com/googlesamples/android-topeka
- fernandocejas.com/2014/09/03/architecting-android-the-clean-way/

OBRIGADO!

rafaeltoledo.net

@_rafaeltoledo

Desenv. **Android** @ **Concrete** **Solutions**

estamos contratando!

concretesolutions.com.br/carreira