



PROF. PEDRO HENRIQUE

---

# DESENVOLVIMENTO PARA IOS 10 COM SWIFT 3



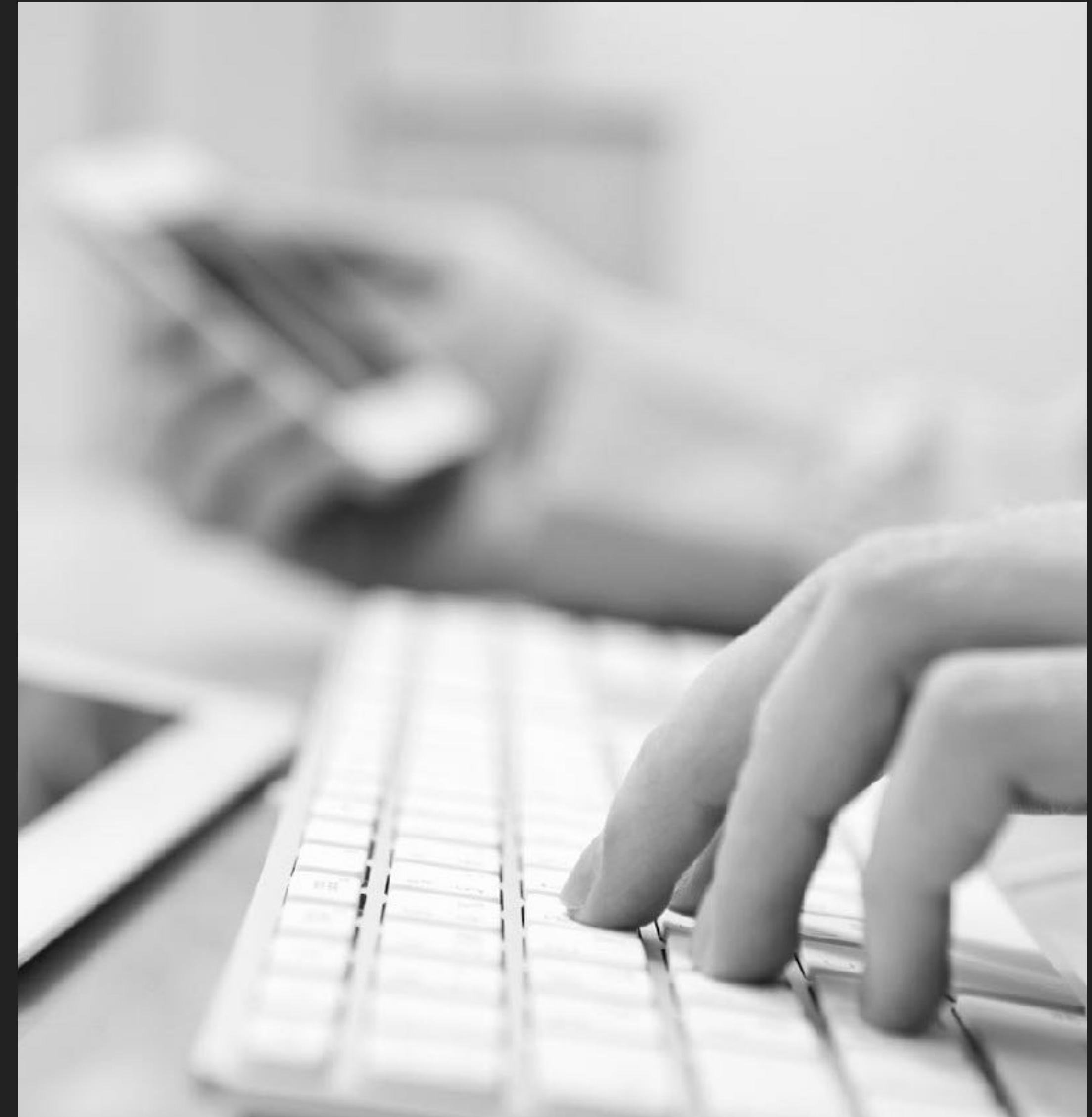
ONDE ENCONTRAR O MATERIAL?

# LEIA O QR CODE

ALÉM DO TRADICIONAL BLACKBOARD DO IESB

## AGENDA

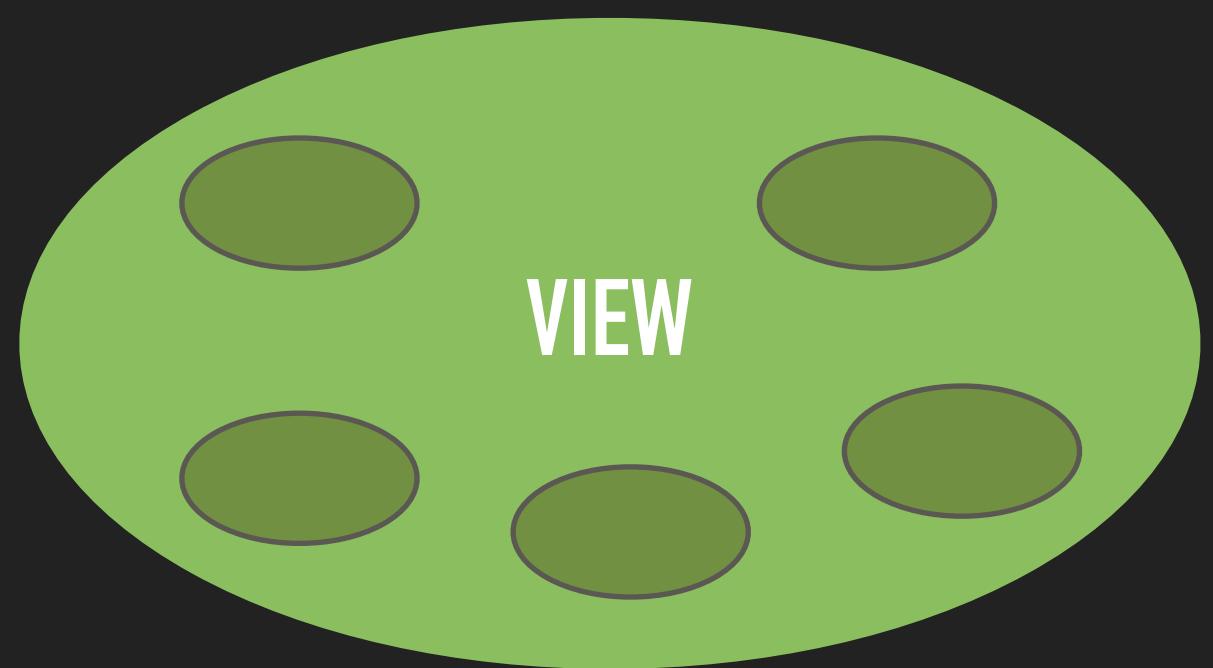
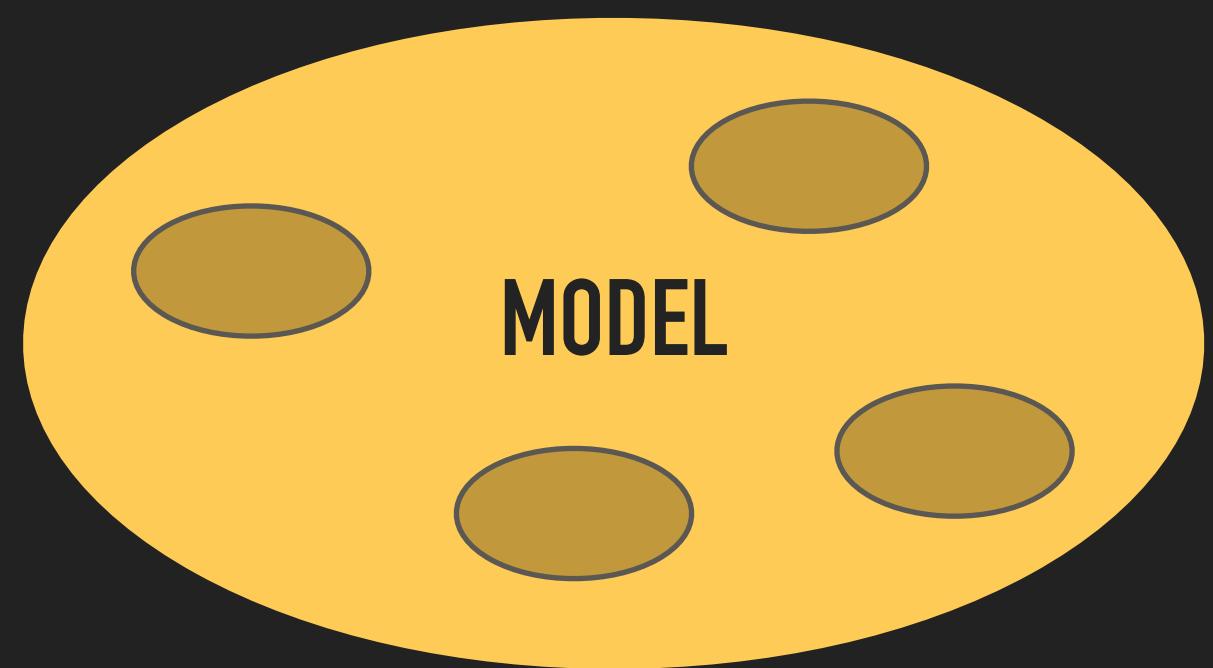
- ▶ MVC aplicado ao iOS;
- ▶ Continuação da Calculadora
- ▶ Exercício para casa



# MODEL VIEW CONTROLLER

---

MVC

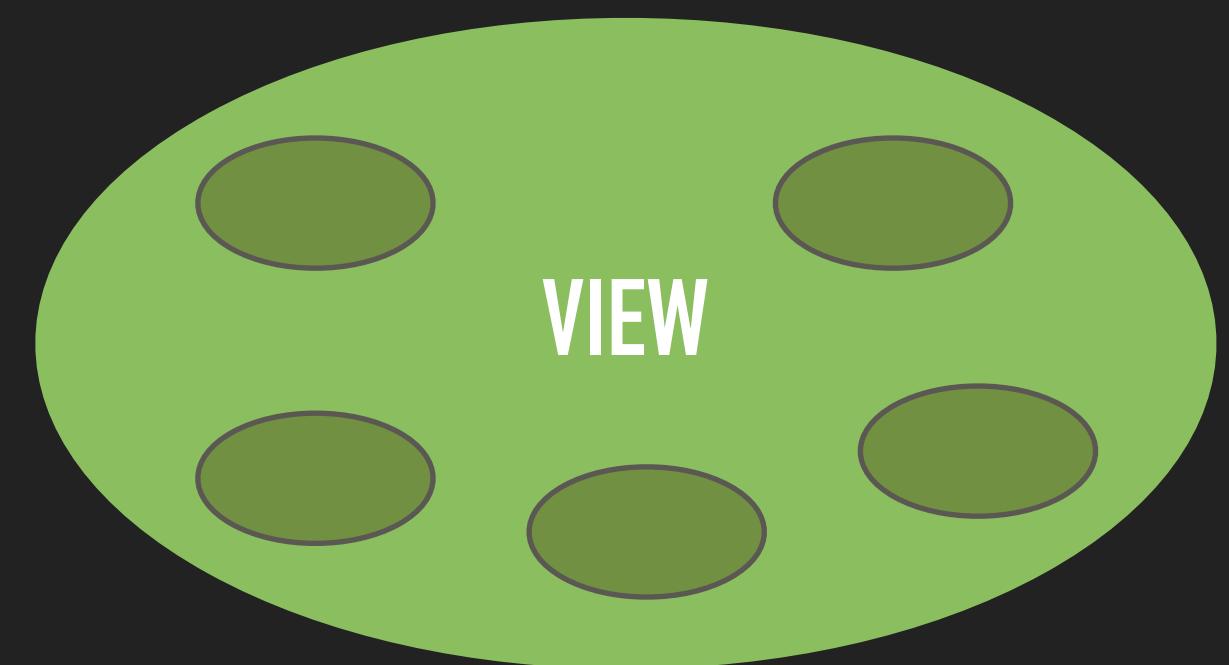
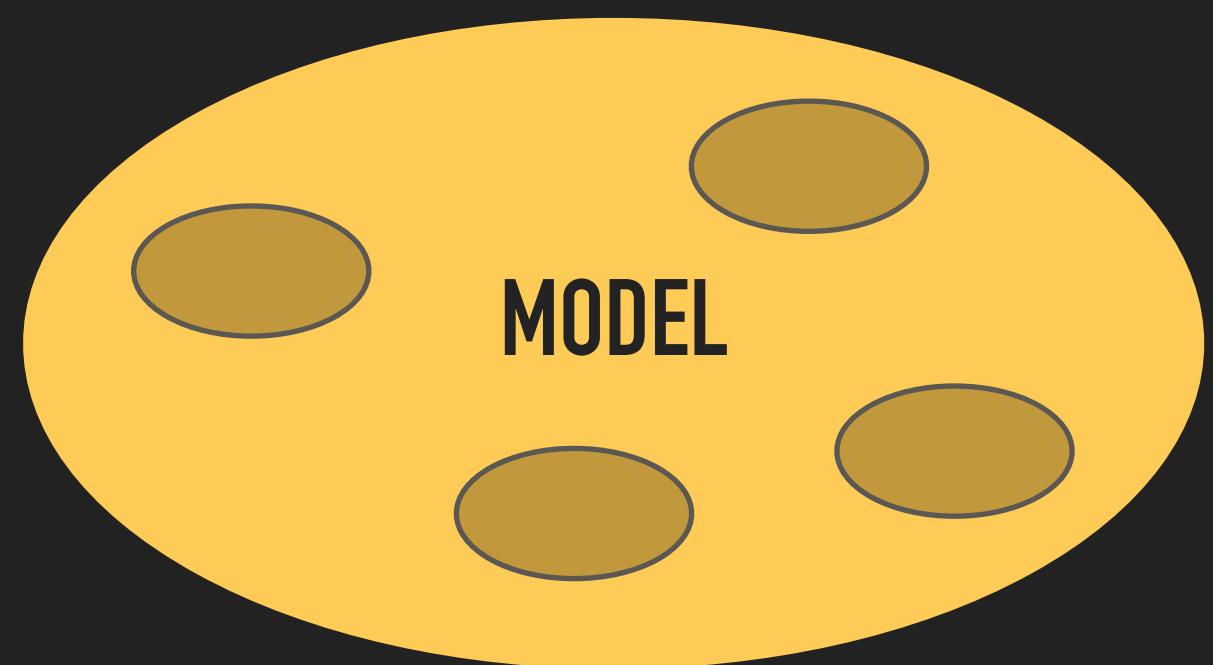


DIVIDE O SEU APLICATIVO EM TRÊS “CAMPOS”.

# MODEL VIEW CONTROLLER

---

MVC

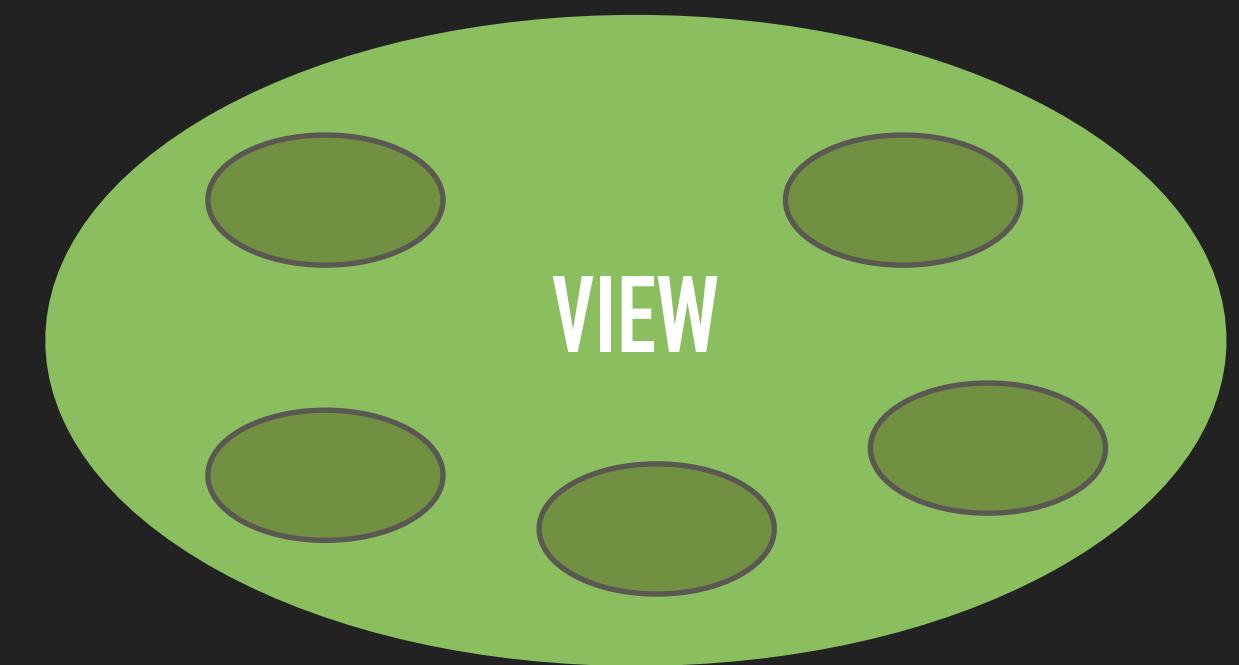
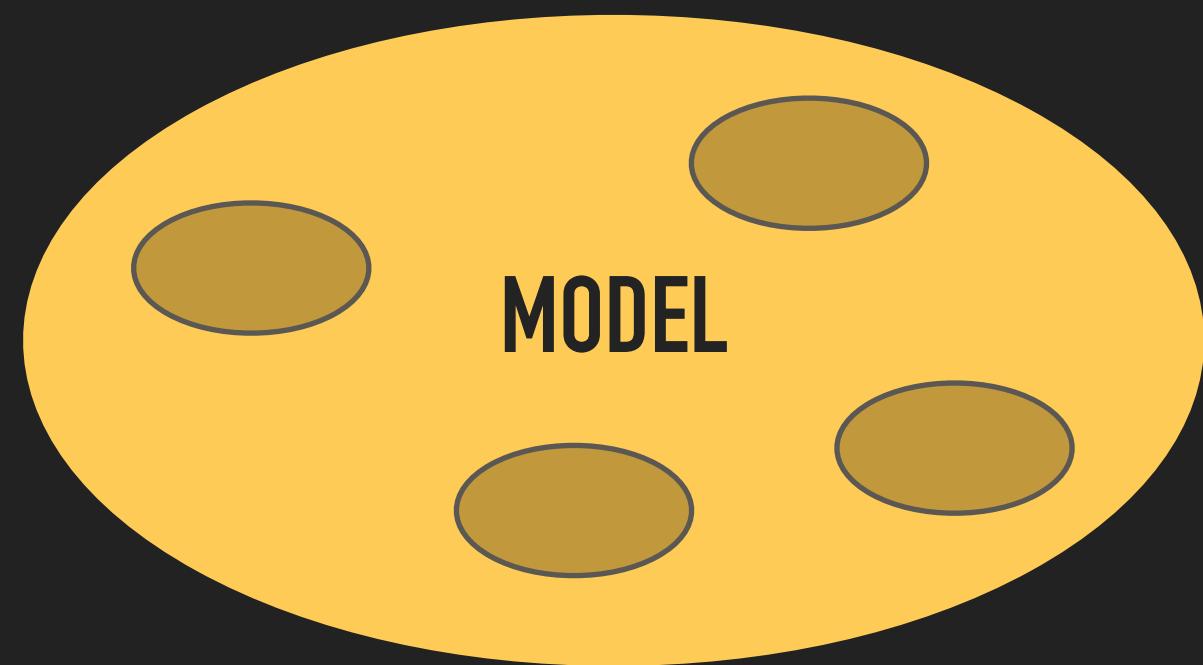


MODEL = O QUE O SEU APLICATIVO É (MAS NÃO COMO ELE É EXIBIDO)

# MODEL VIEW CONTROLLER

---

MVC

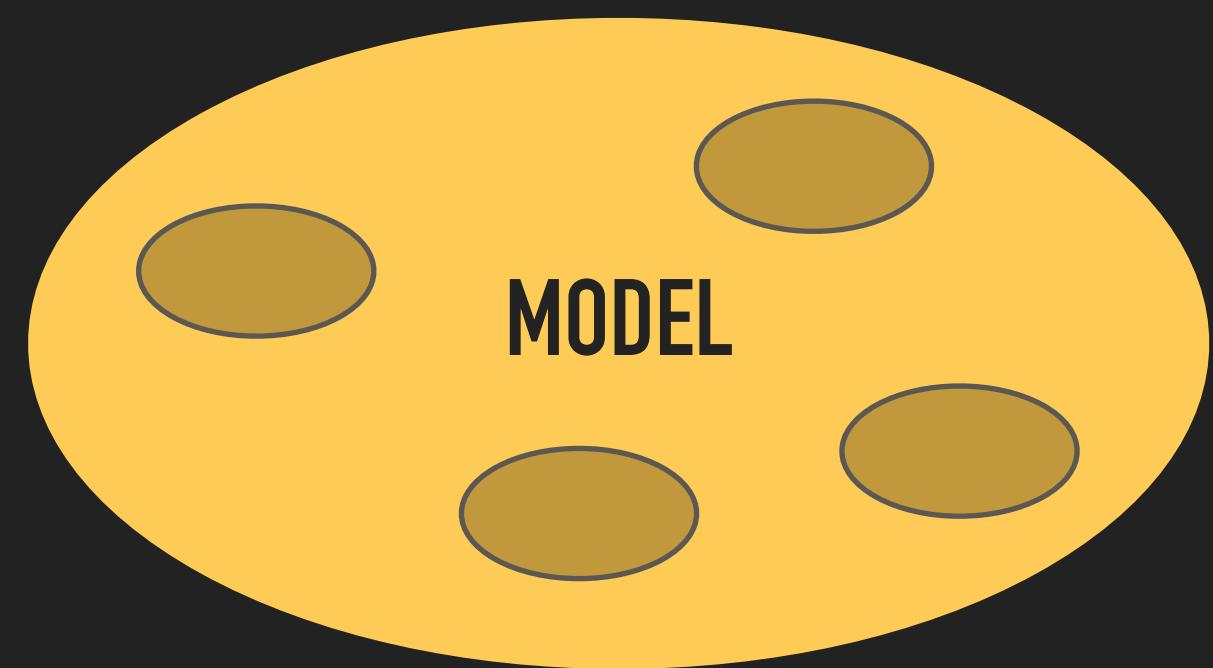


CONTROLLER = COMO A SUA MODEL É APRESENTADA PARA O USUÁRIO (LÓGICA DE INTERFACE)

# MODEL VIEW CONTROLLER

---

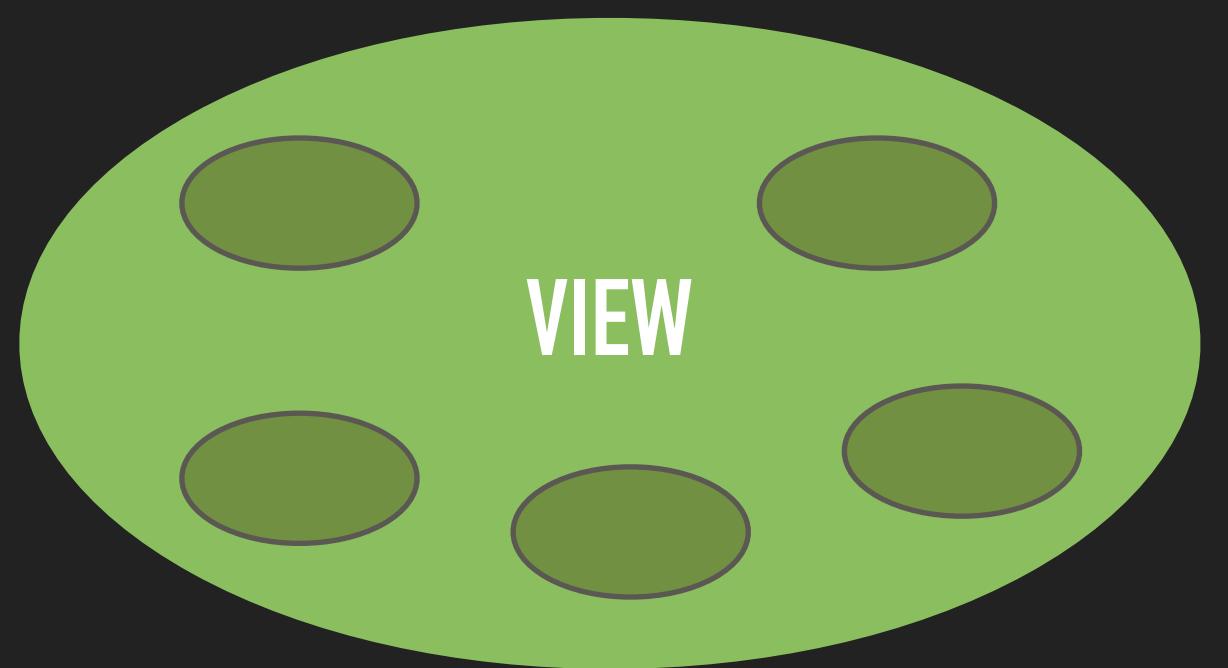
MVC



MODEL



CONTROLLER



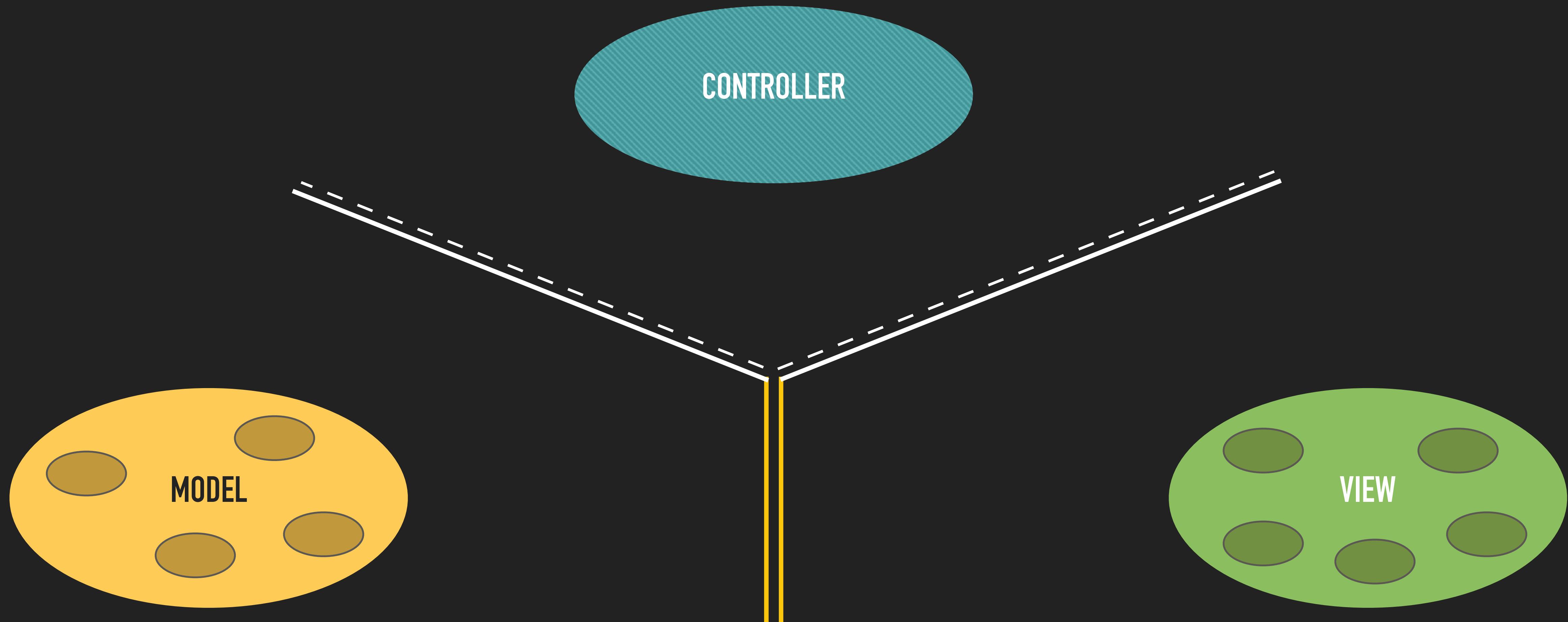
VIEW

VIEW = OS MINIONS DO SEU CONTROLLER

## MODEL VIEW CONTROLLER

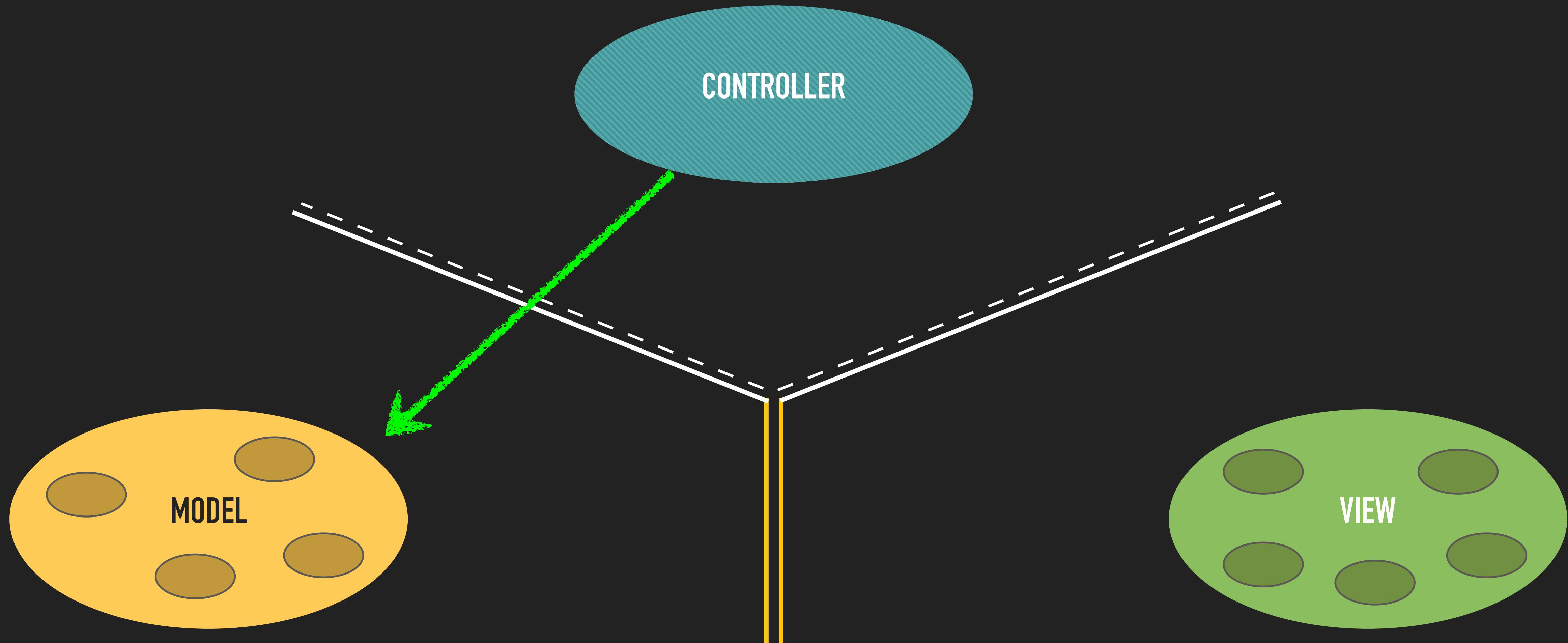
---

MVC



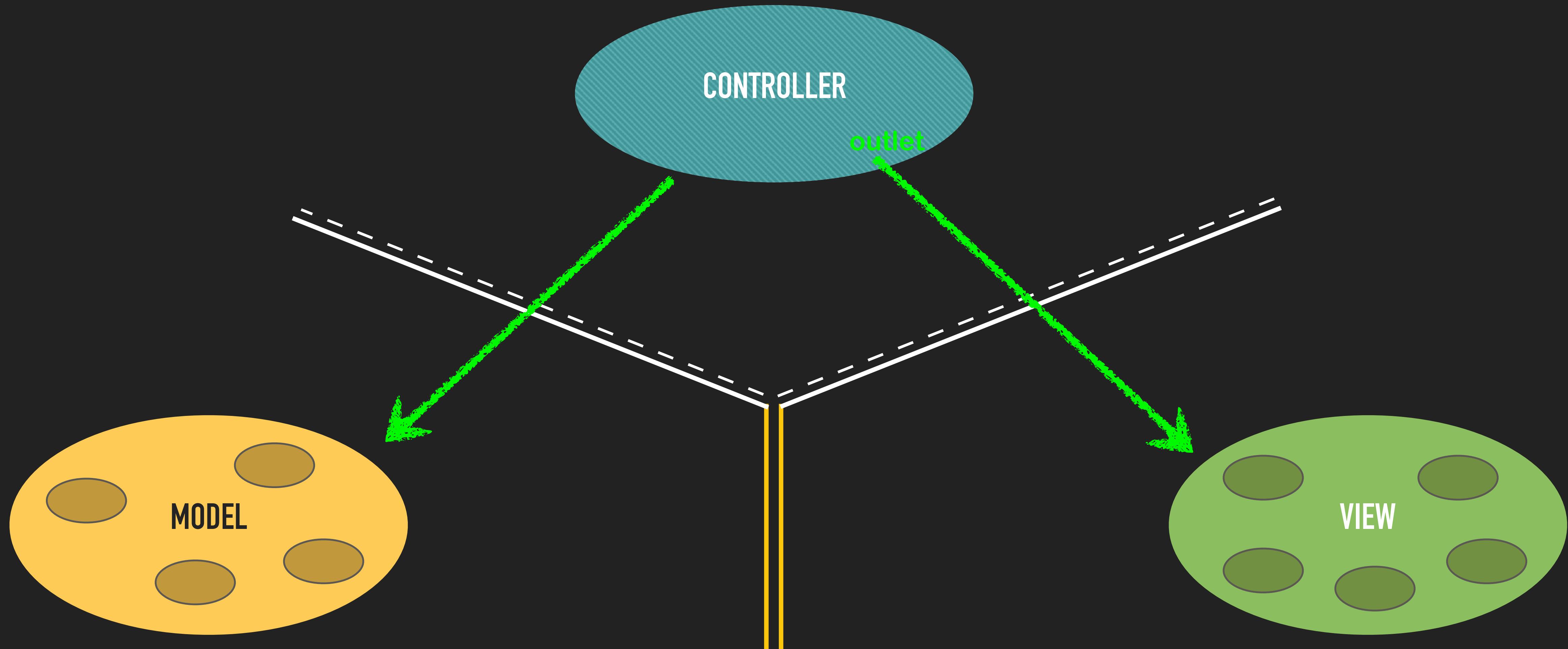
ENTÃO, O MVC PODE SER COMPREENDIDO COMO O TRABALHO DE GERENCIAR A COMUNICAÇÃO ENTRE OS “CAMPOS”.

# MVC



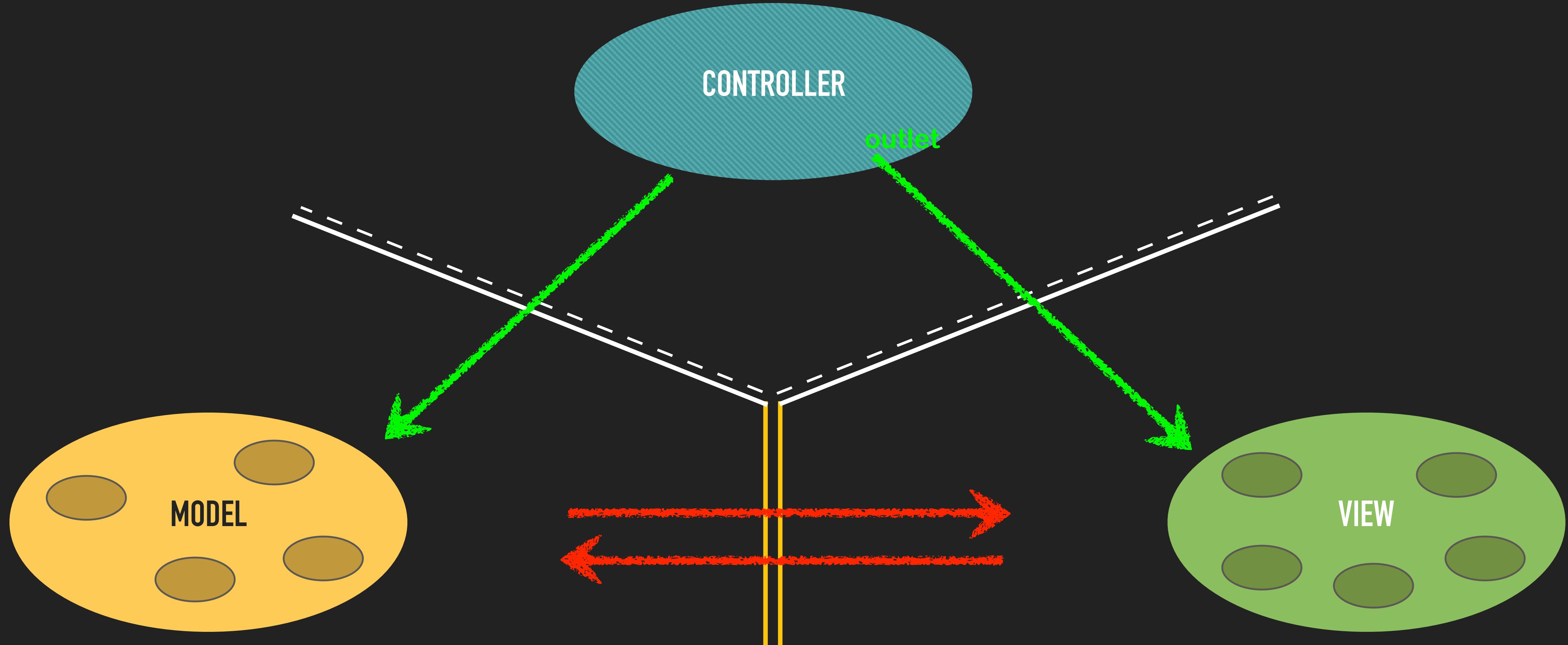
O CONTROLLER SEMPRE PODE CONVERSAR DIRETAMENTE COM SEU MODEL.

# MVC



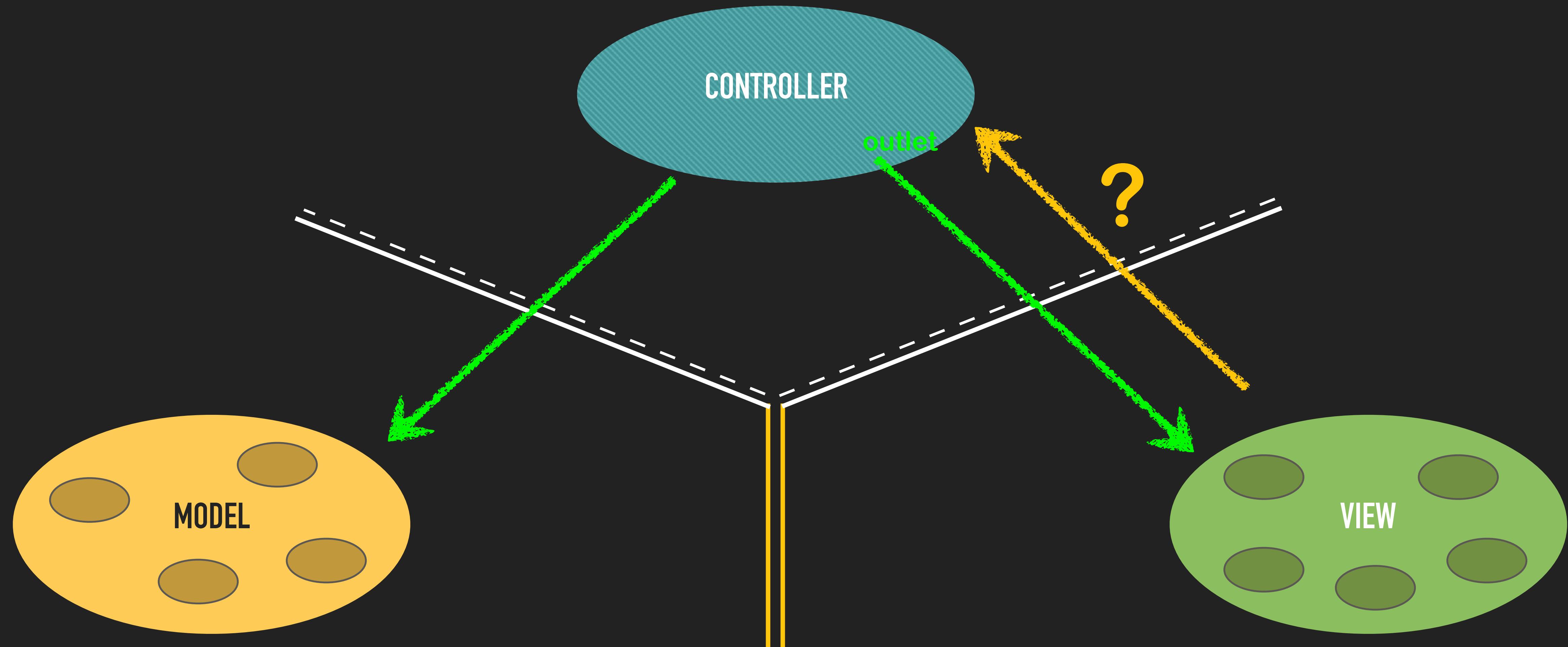
O CONTROLLER TAMBÉM PODE CONVERSAR DIRETAMENTE COM A VIEW.

# MVC



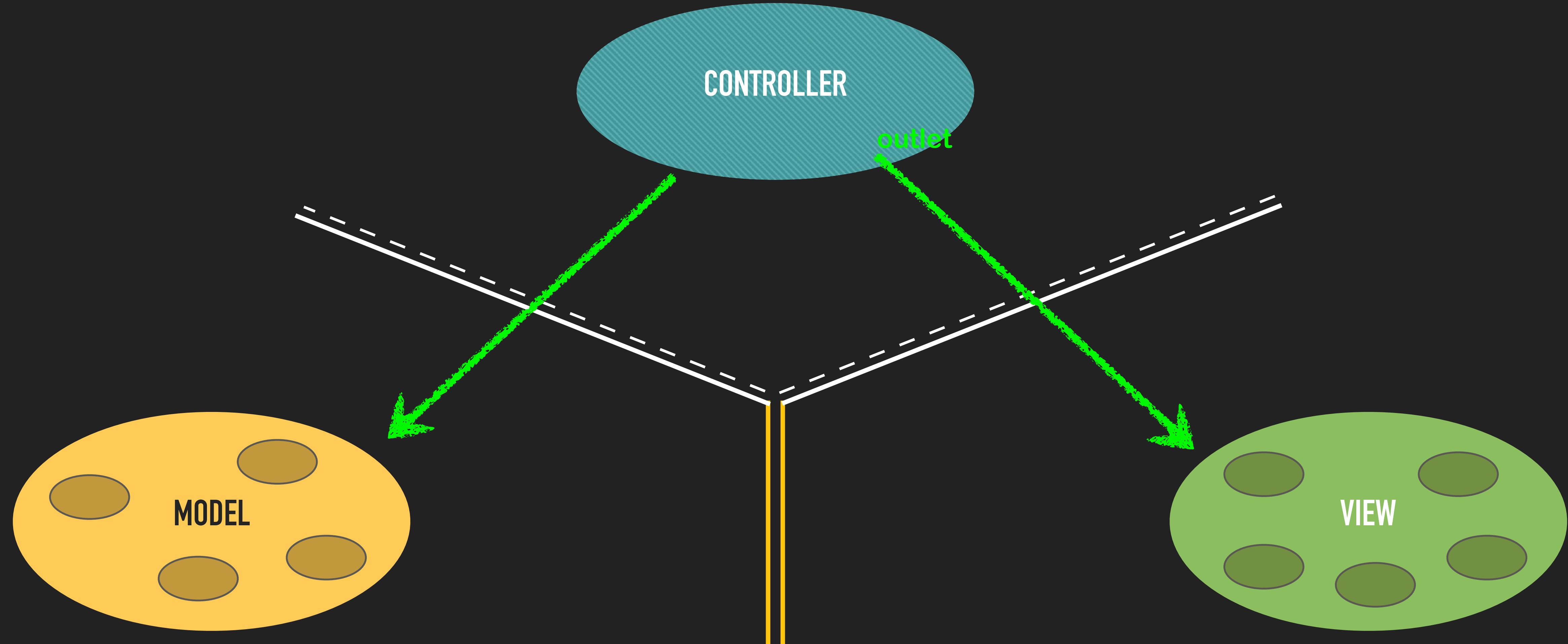
A MODEL E A VIEW JAMAIS DEVEM SE COMUNICAR.

# MVC



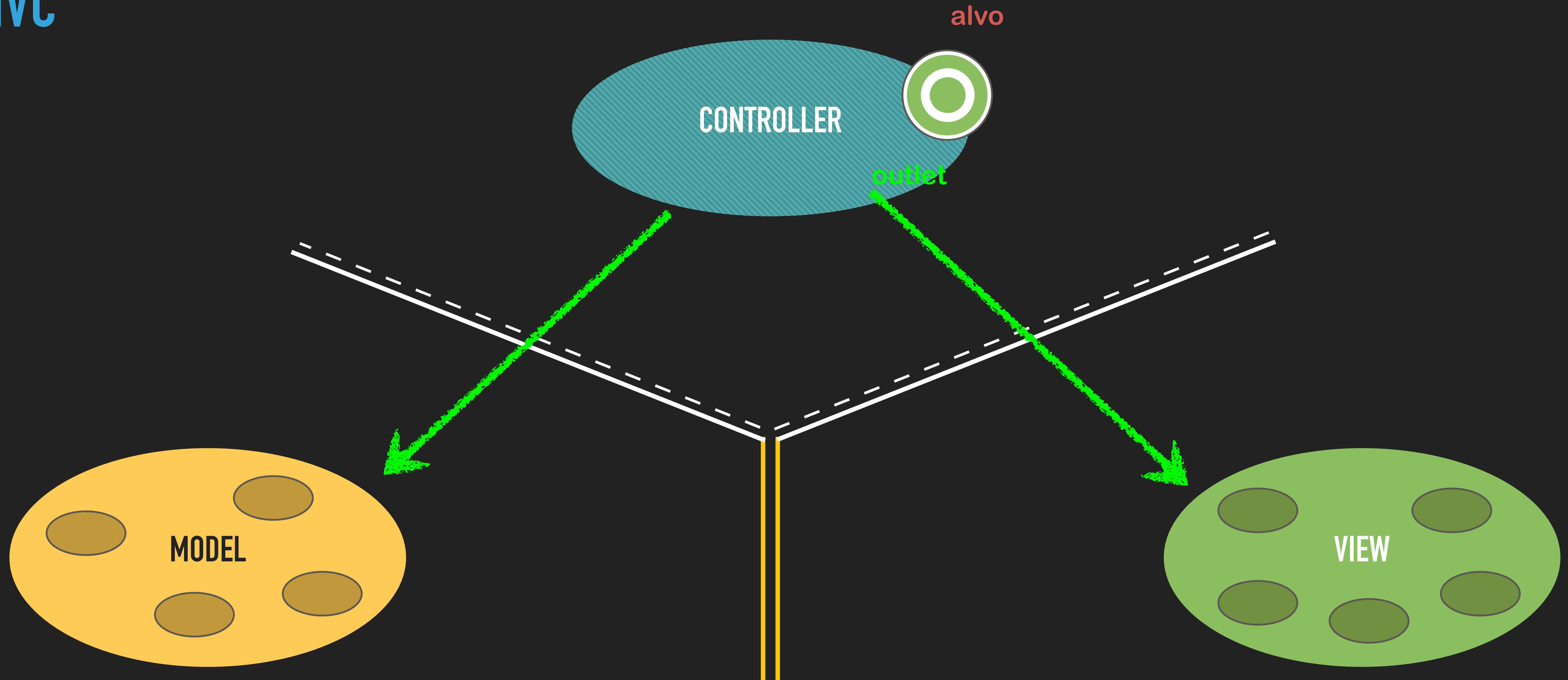
A VIEW PODE CONVERSAR COM O CONTROLLER?

## MVC



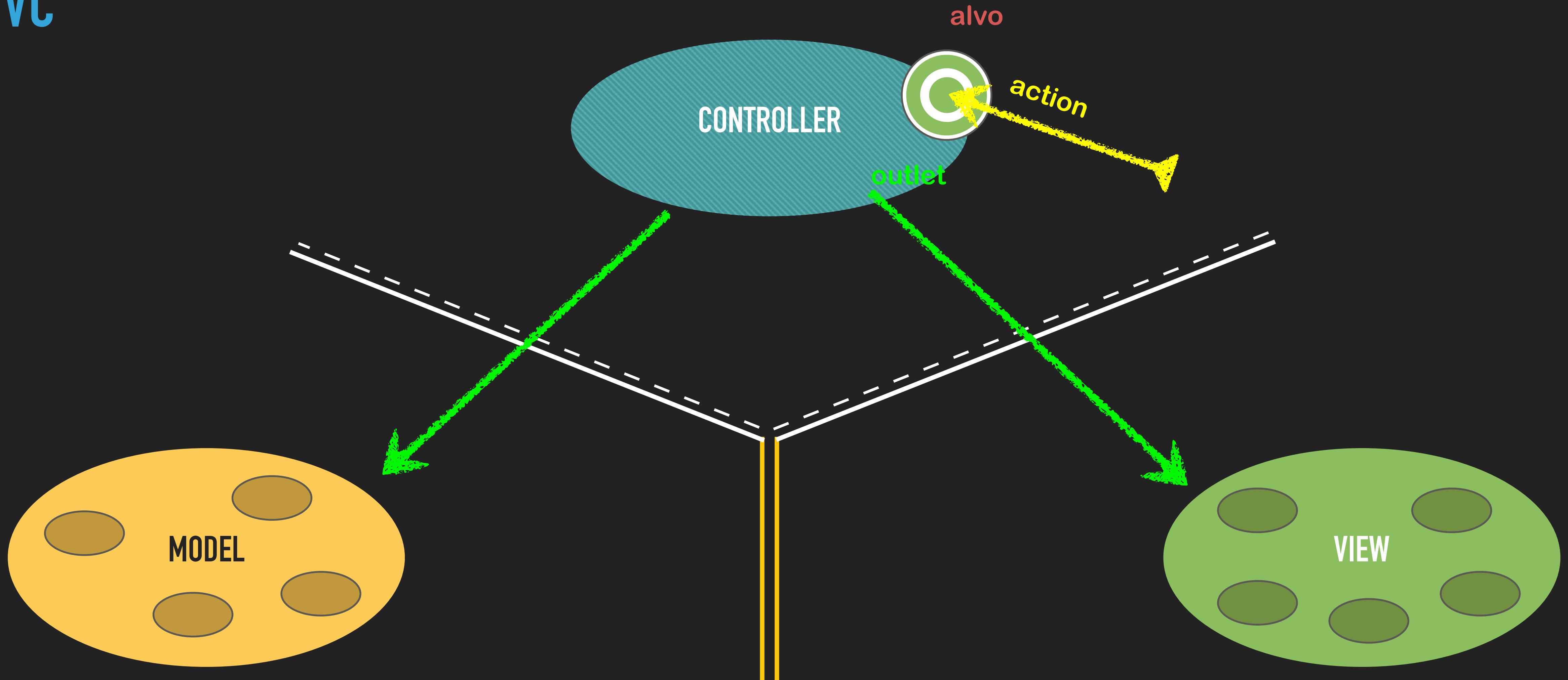
MAIS OU MENOS... A COMUNICAÇÃO SE DÁ ATRAVÉS DE UMA MANEIRA “CEGA” E ESTRUTURADA AO MESMO TEMPO.

## MVC



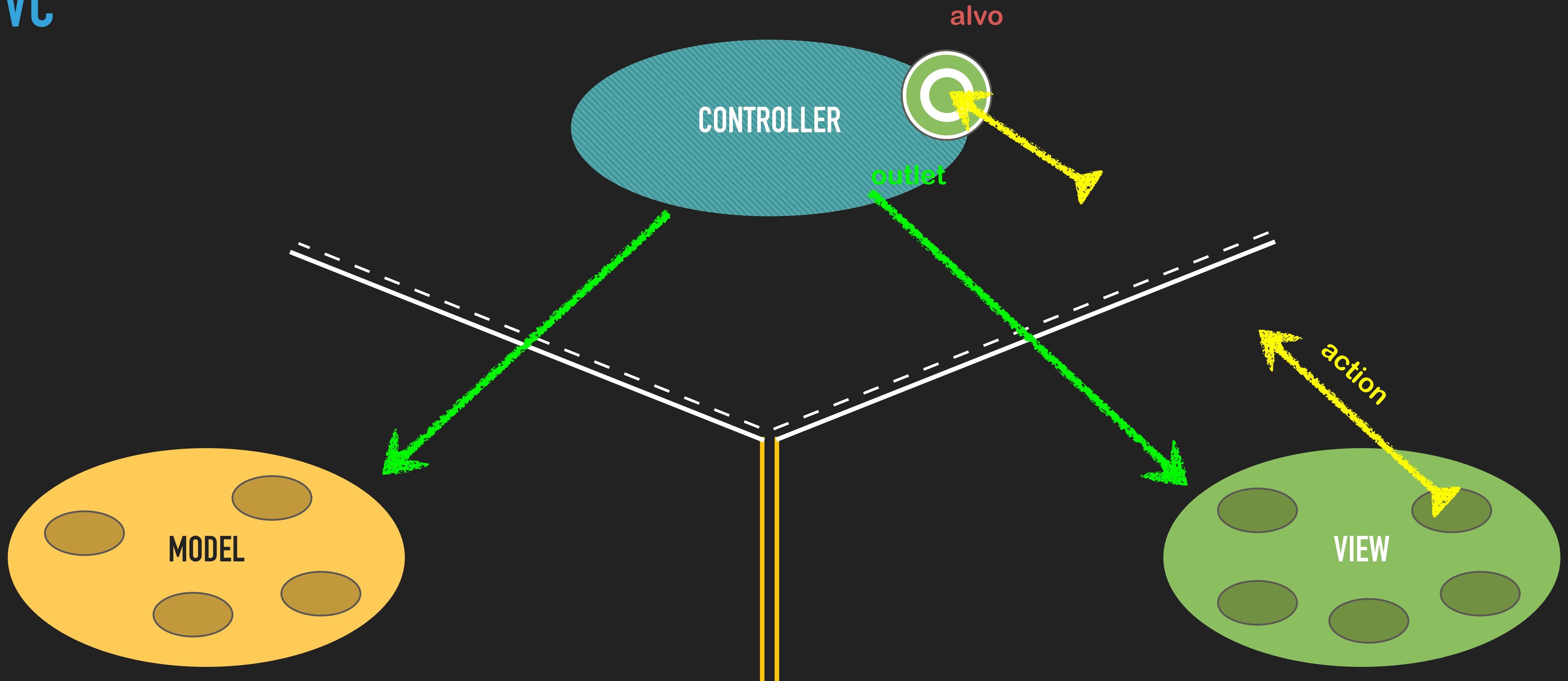
O CONTROLLER PODE COLOCAR UM ALVO EM SI MESMO.

## MVC



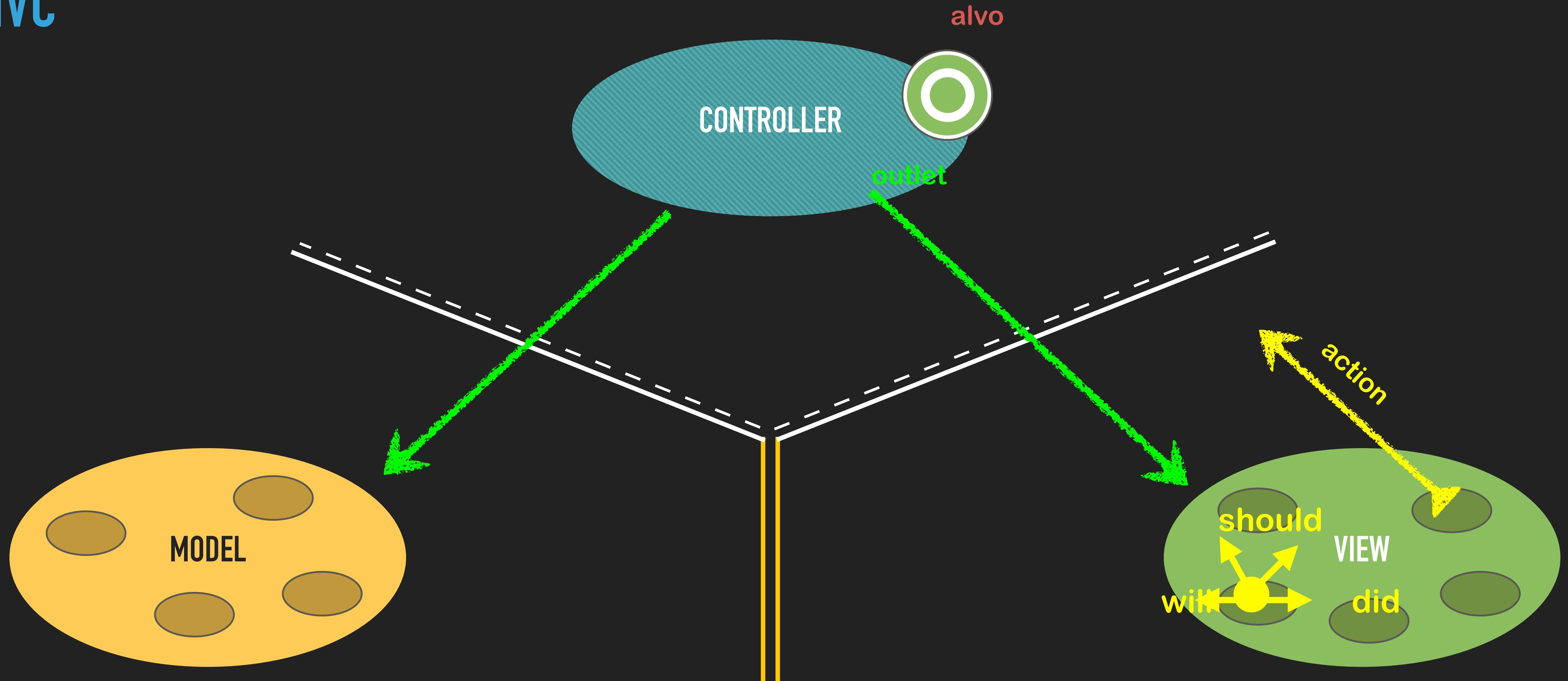
E ENTÃO, ACEITAR UMA ACTION QUE VEM DA VIEW.

### MVC



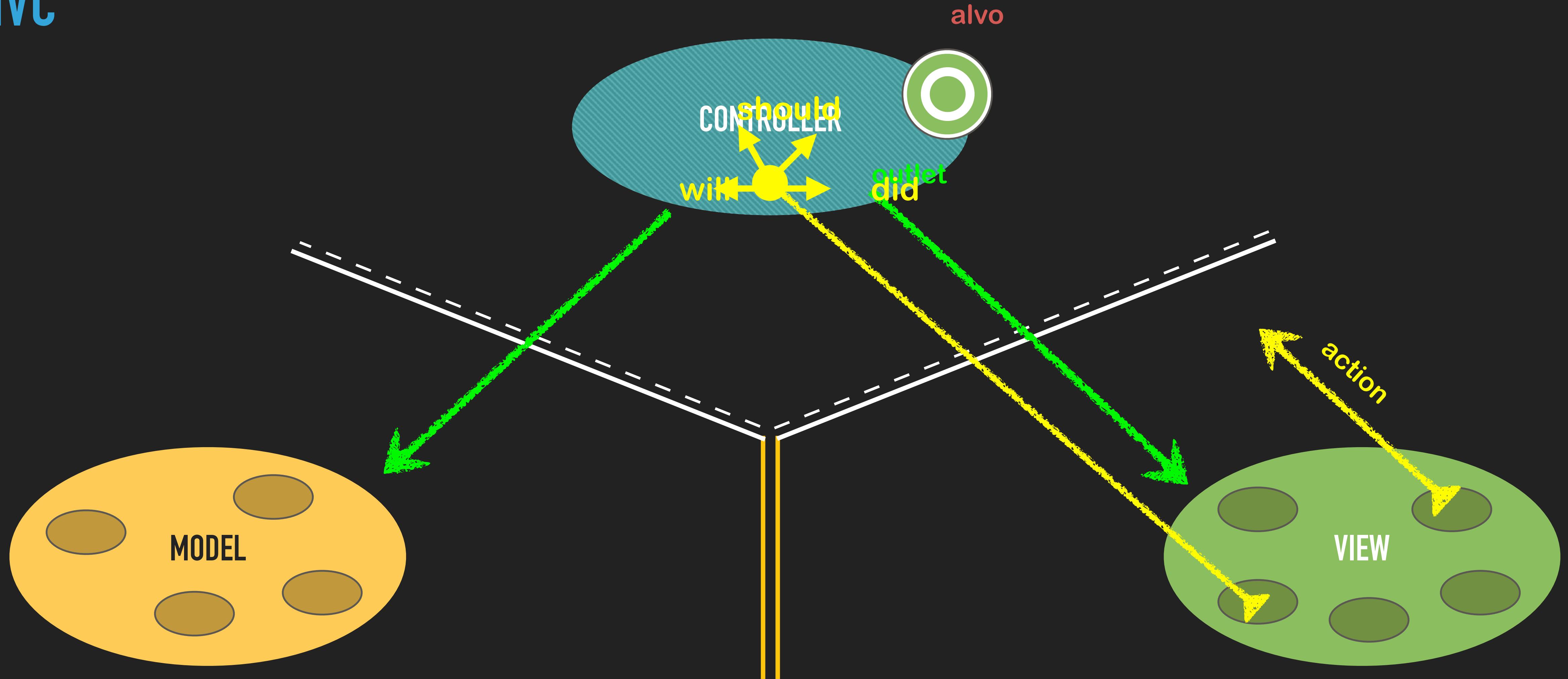
A VIEW ENVIA A ACTION QUANDO COISAS ACONTECEM NA INTERFACE DO USUÁRIO.

## MVC



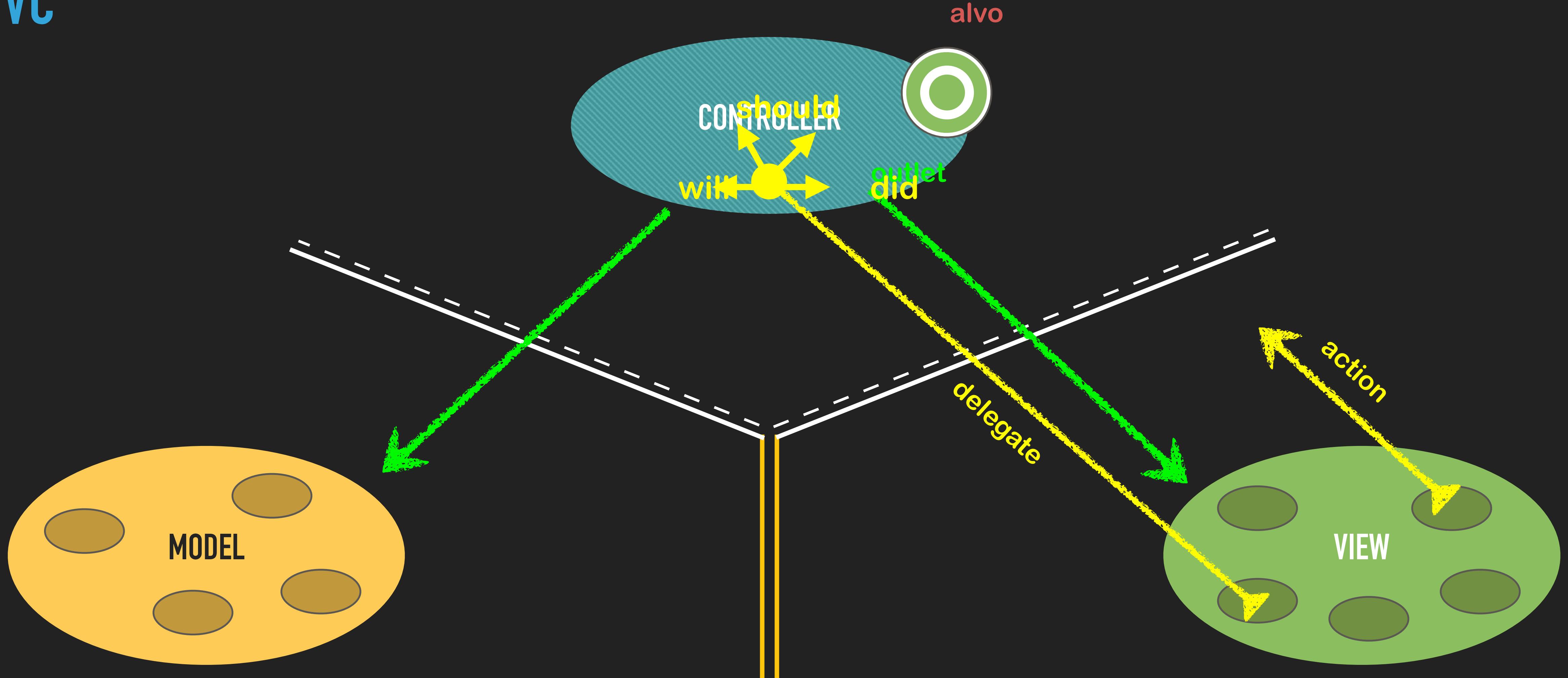
ÀS VEZES A VIEW PRECISA SE SINCRONIZAR COM O CONTROLLER

## MVC



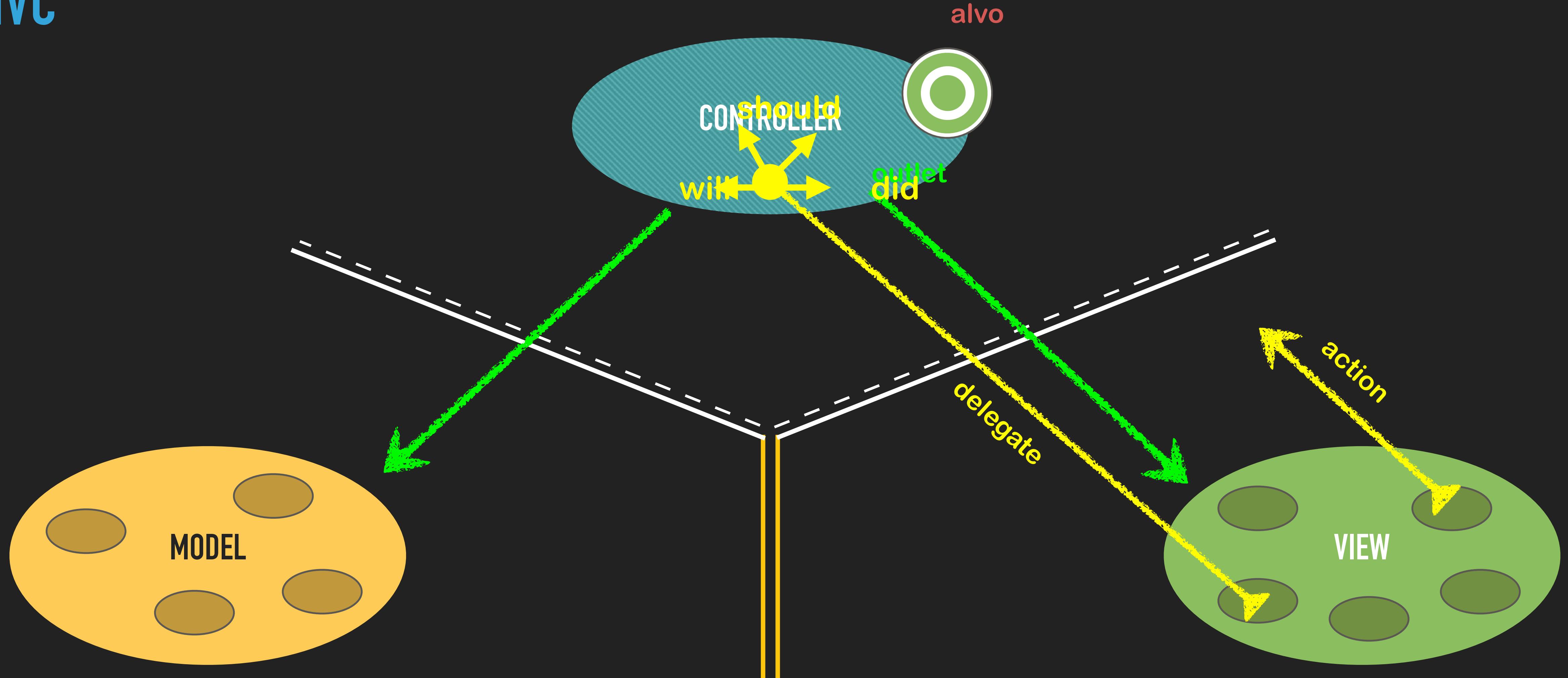
ENTÃO O CONTROLLER SETA A SI MESMO COMO DELEGATE DA VIEW

## MVC



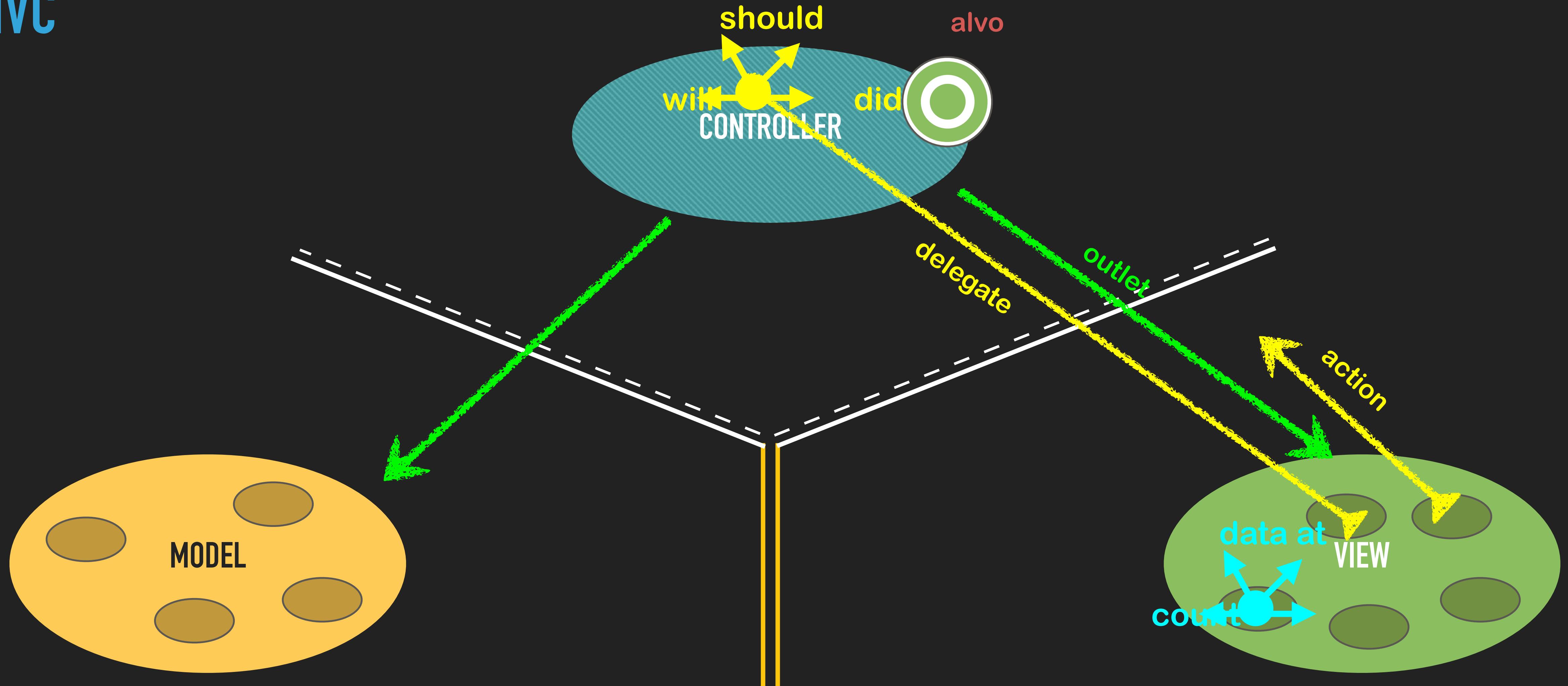
ESTE PROCESSO DE SETAR O DELEGATE É FEITO ATRAVÉS DE UM PROTOCOLO (QUE É ALGO CEGO EM RELAÇÃO À CLASSE)

## MVC



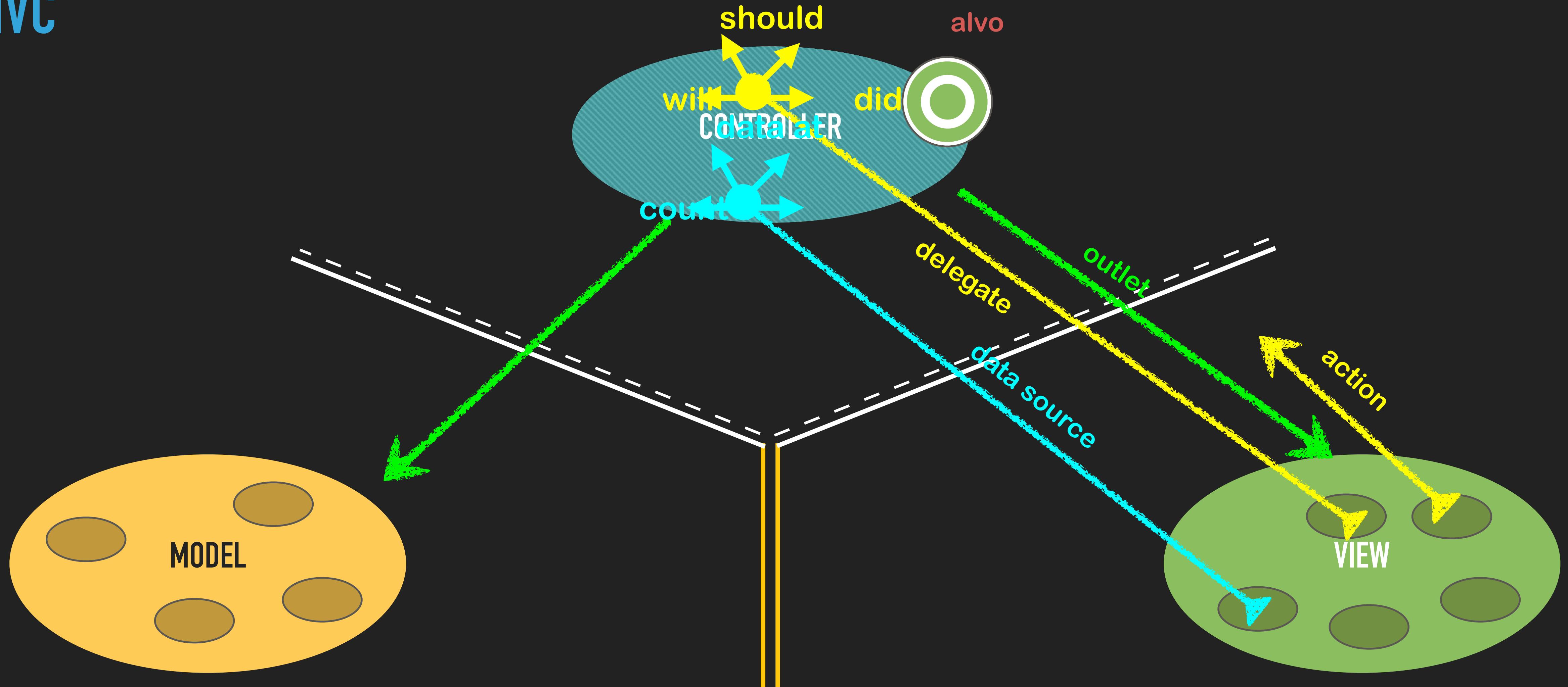
A VIEW NÃO É DONA DO DADO QUE ELA EXIBE!!!

## MVC



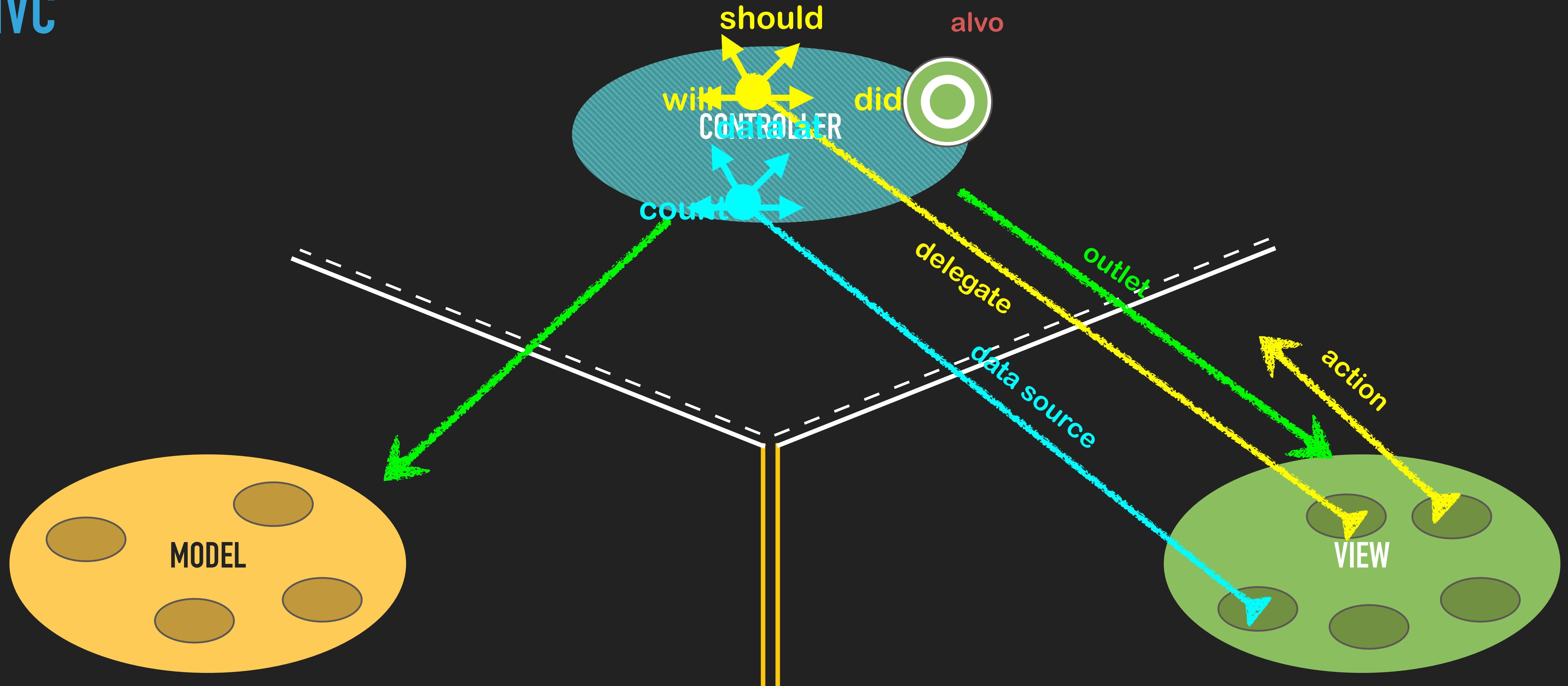
ENTÃO, QUANDO É NECESSÁRIO, ELA USA UM PROTOCOLO PARA OBTER OS DADOS

## MVC



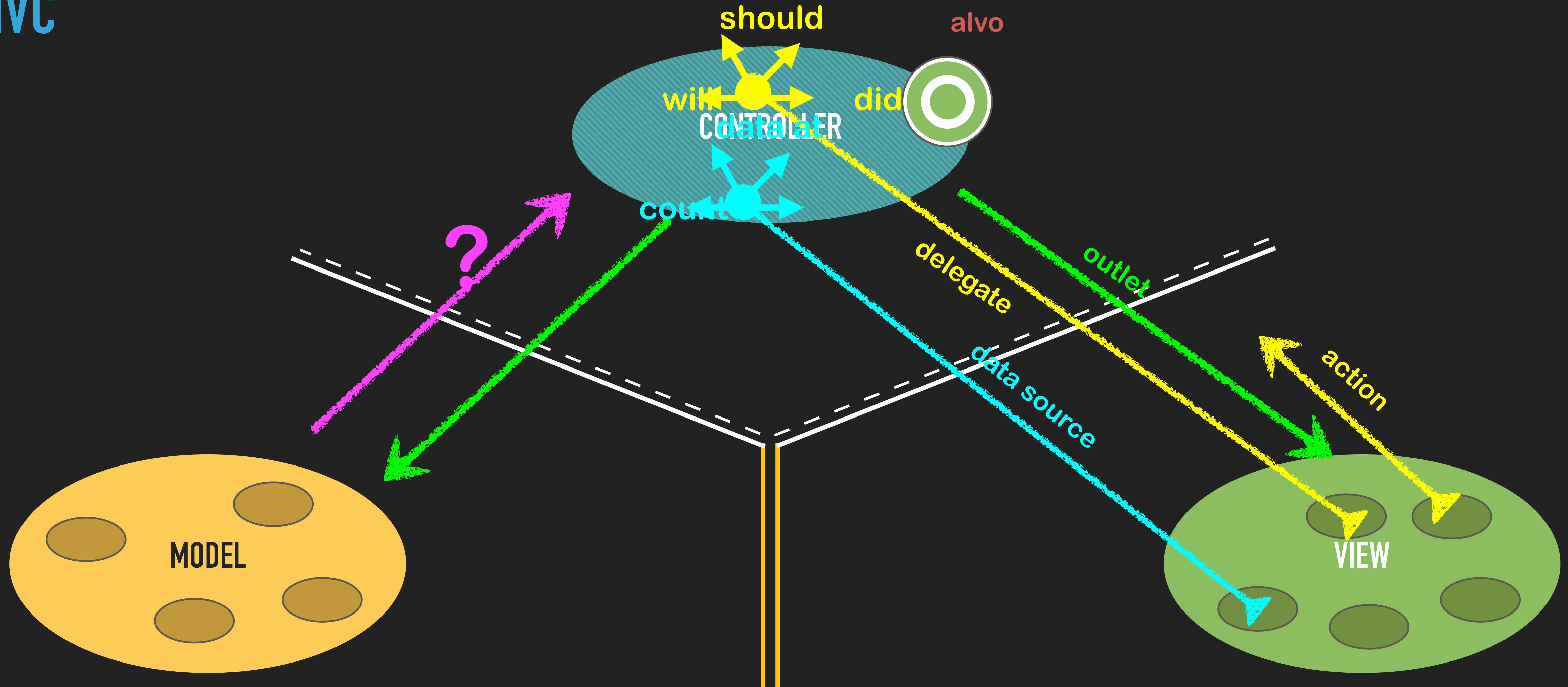
ENTÃO, QUANDO É NECESSÁRIO, ELA USA UM PROTOCOLO PARA OBTER OS DADOS

## MVC



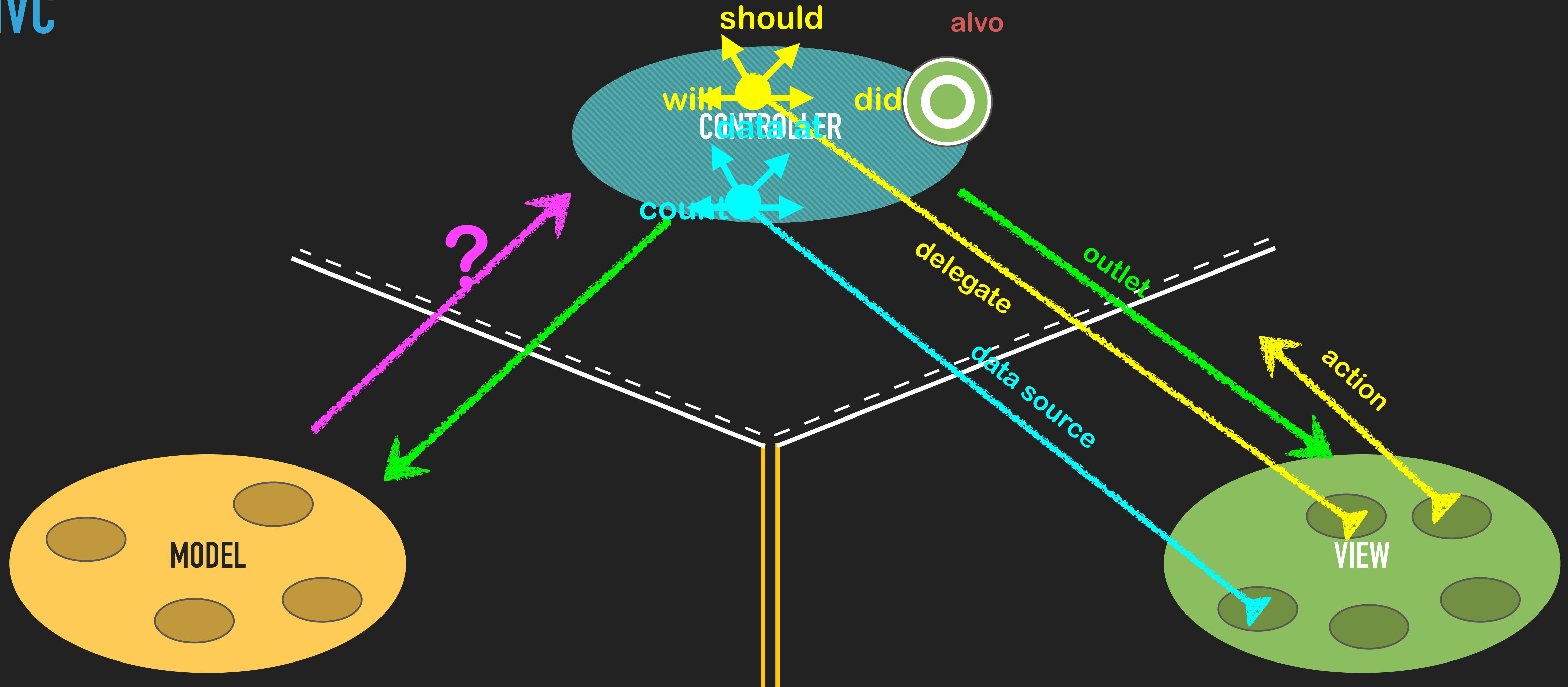
O PAPEL DO CONTROLLER, PORTANTO, É INTERPRETAR A MODEL E FORMATAR A INFORMAÇÃO PARA A VIEW

## MVC



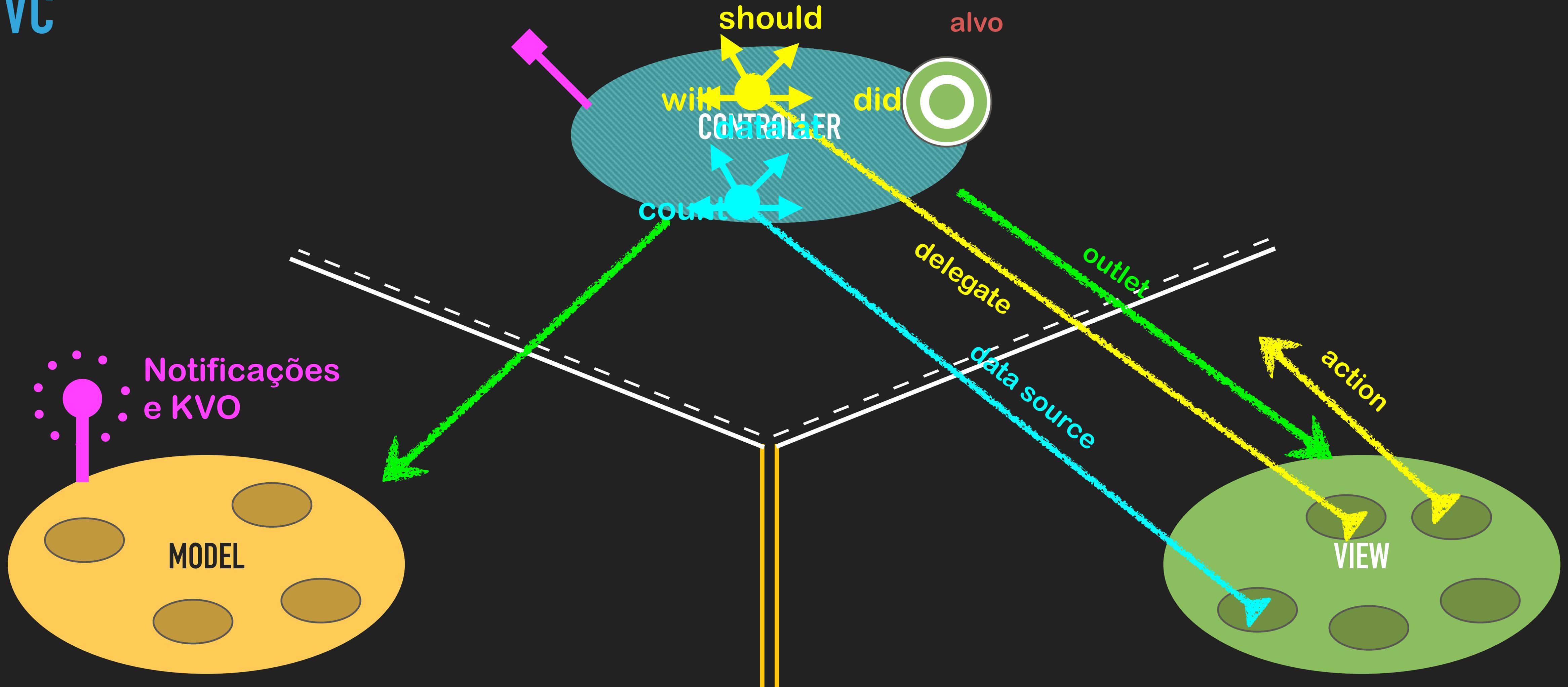
E A MODEL? ELA PODE FALAR DIRETAMENTE COM O CONTROLLER?

## MVC



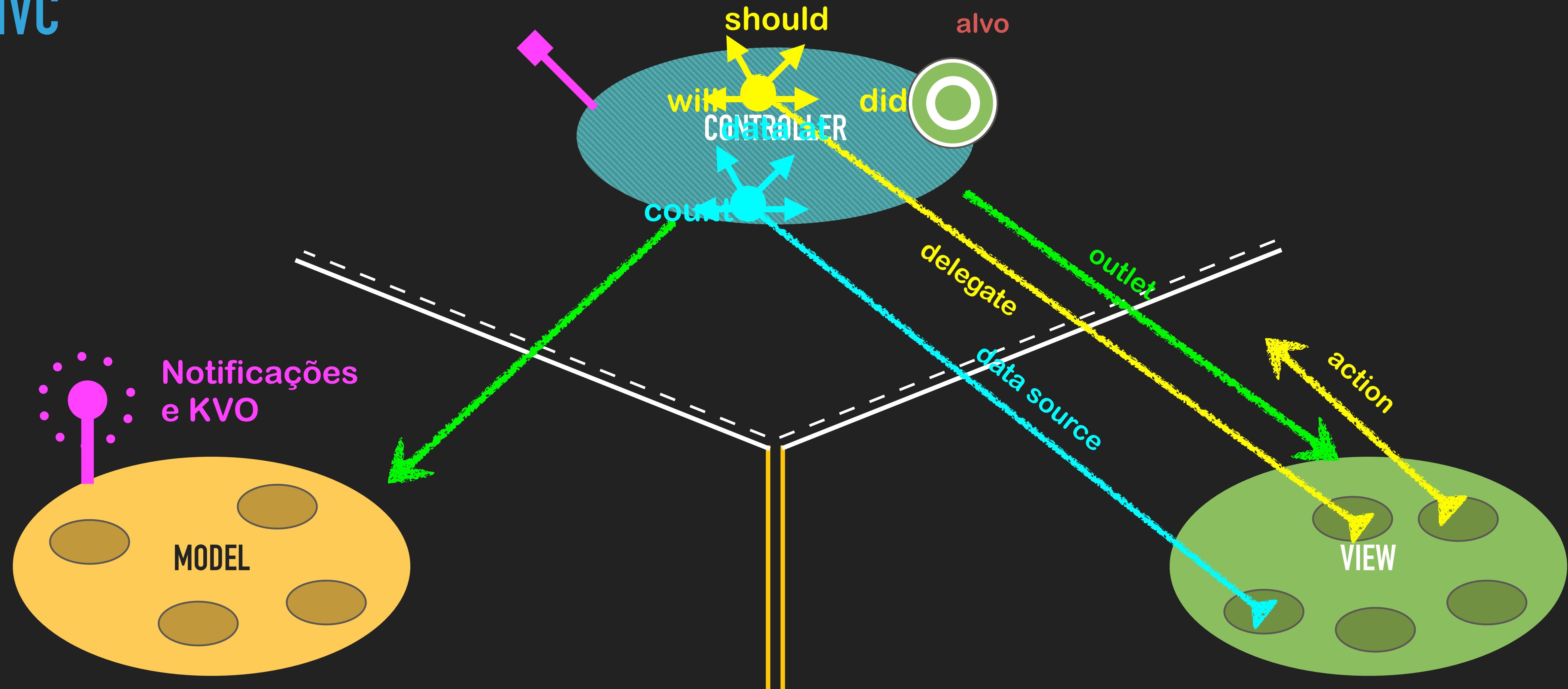
NÃO! A MODEL É (DEVERIA SER) COMPLETAMENTE INDEPENDENTE DA INTERFACE DE USUÁRIO.

## MVC



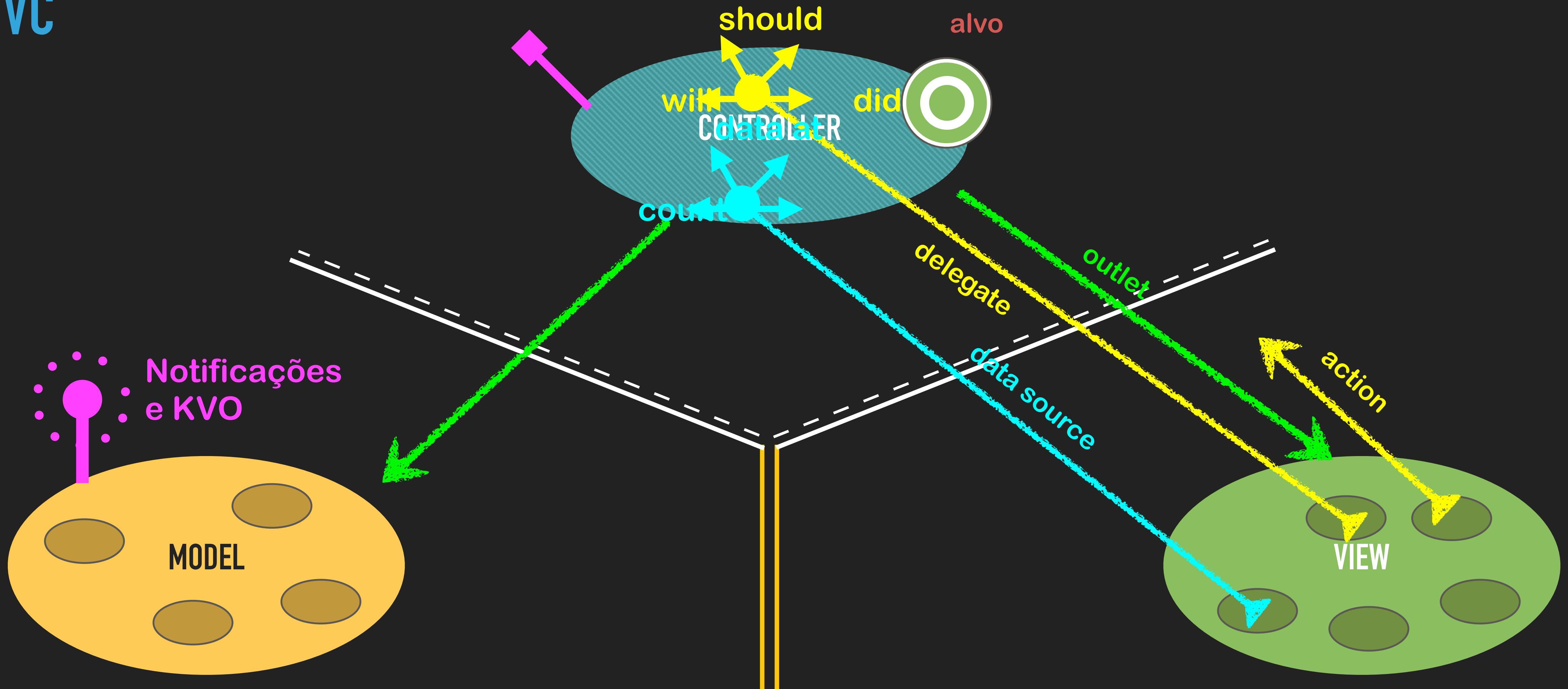
ELA USA UM “SISTEMA DE ESTAÇÕES DE RÁDIO” E COLOCA A NOTÍCIA EM BROADCAST.

## MVC



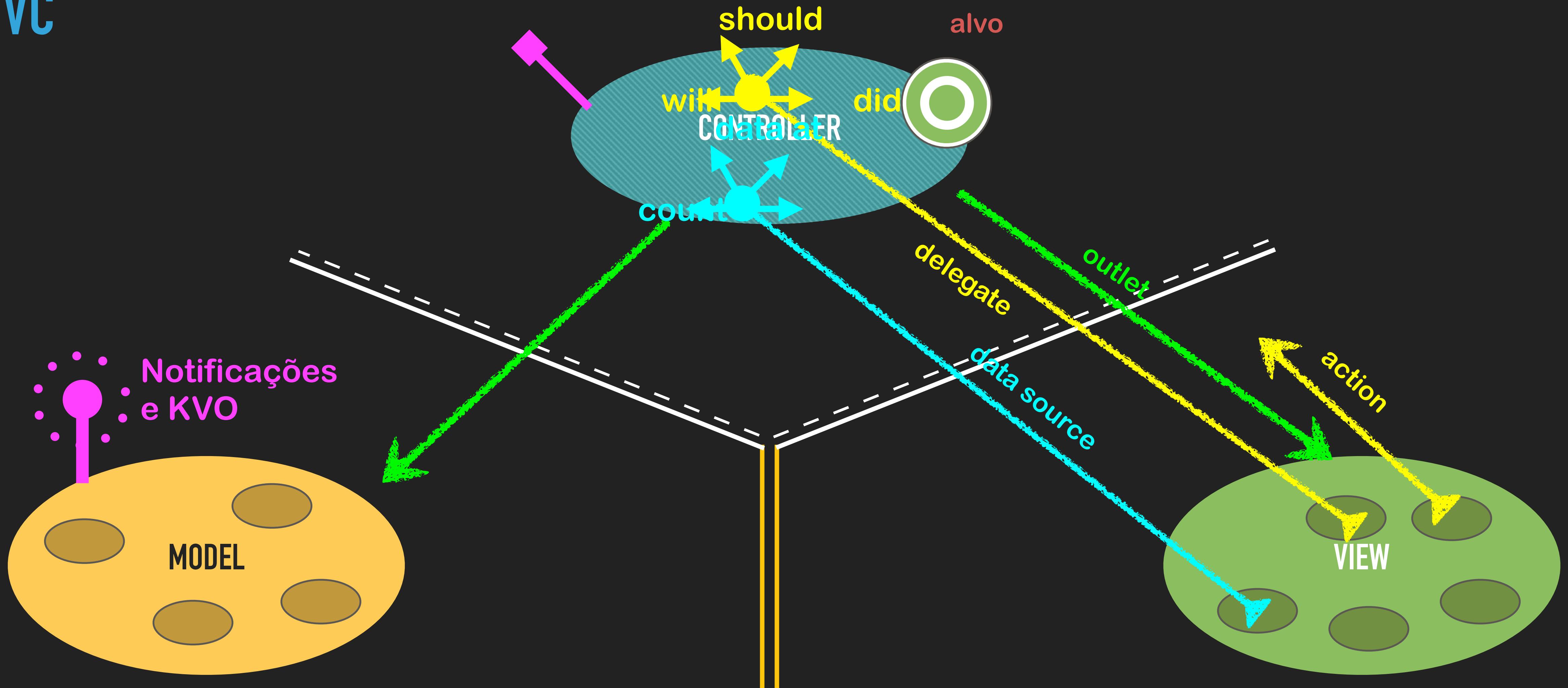
O CONTROLLER (OU MESMO OUTRAS CLASSES MODEL) "SINTONIZAM" NESTAS ESTAÇÕES DE RÁDIO PARA OUVIR ALGO INTERESSANTE.

## MVC



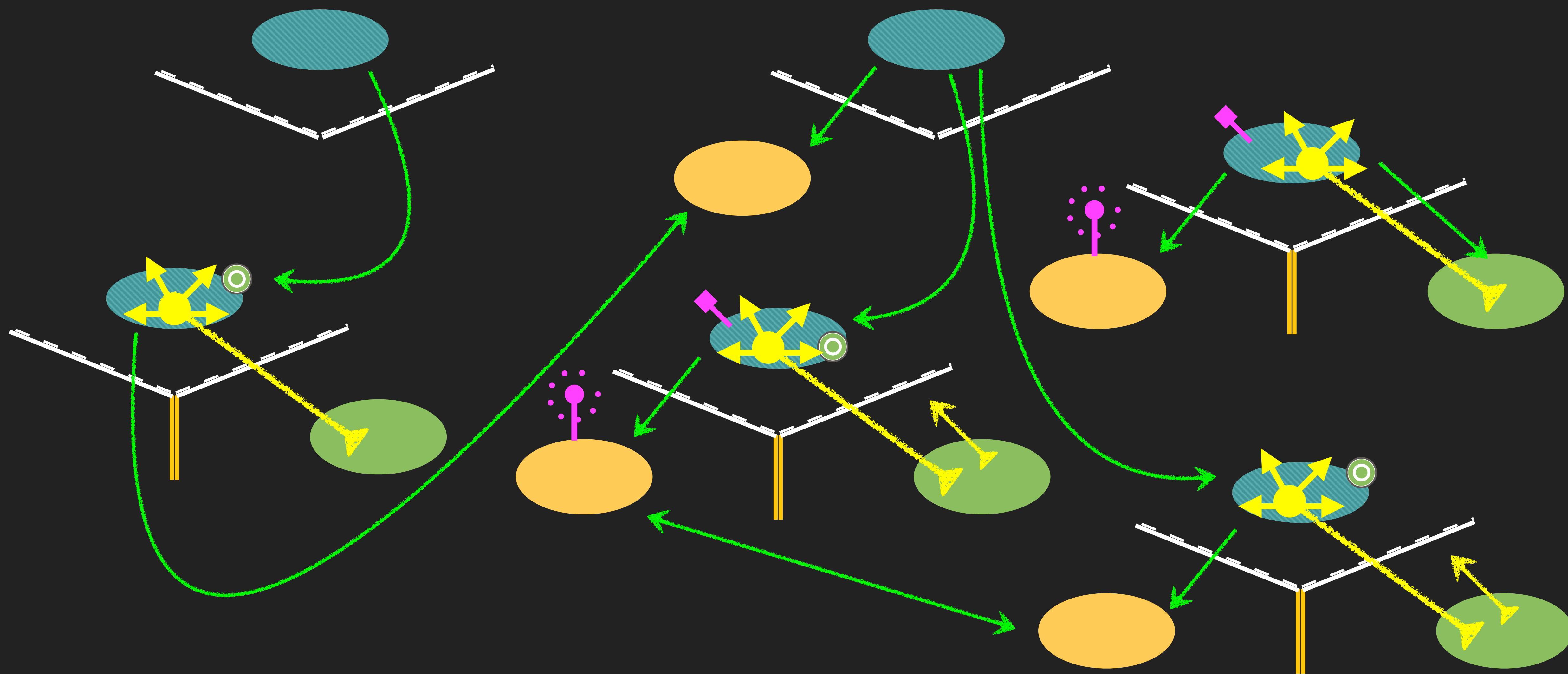
UMA VIEW PODE ATÉ “SINTONIZAR” ALGUMA RÁDIO, MAS QUE NÃO SEJA DE UMA ESTAÇÃO MODEL.

## MVC

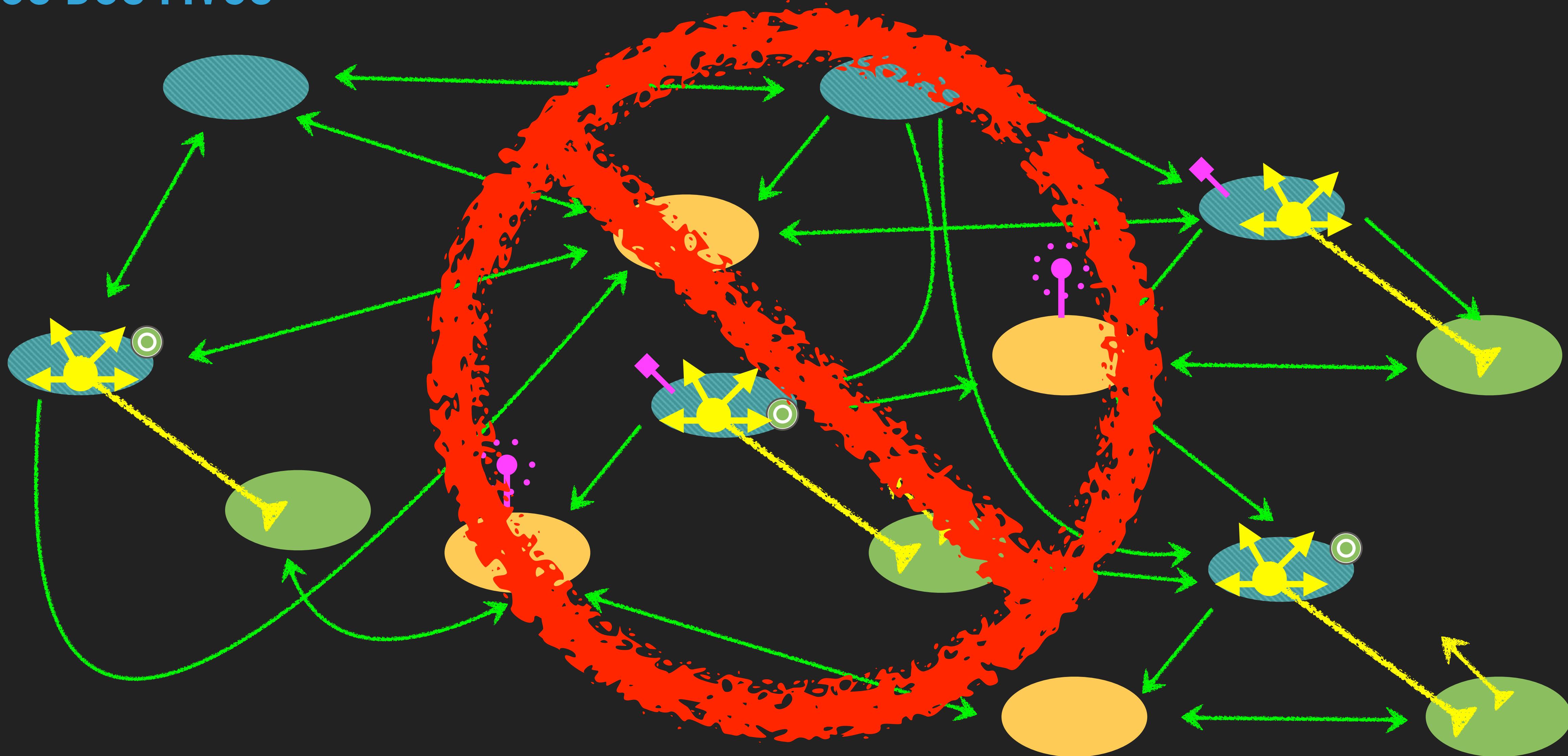


AGORA, COMBINEMOS VÁRIOS MVCS PARA FAZER APLICATIVOS COMPLEXOS...

# MVCS TRABALHANDO JUNTOS



## O CAOS DOS MVCS



HORA DE BRINCAR!!!

---

## CONTINUAÇÃO DA CALCULADORA

- ▶ Aplicando o MVC;
- ▶ struct versus class (capacidade de mutação e etc);
- ▶ Mais exemplos de Optional;
- ▶ Dictionary<TipoDaChave, TipoDoValor>
- ▶ Enumerações;
- ▶ Valores associados;
- ▶ Funções como tipos;
- ▶ Closures (sintaxe para definir funções on the fly);
- ▶ UIStackView
- ▶ Primeiro contato com o Autolayout (vamos “pregar” as coisas nos cantos)





**EXERCÍCIO PARA CASA  
ENTREGA ATÉ O DIA  
25/05 ÀS 19:09**

**Descrição completa no Blackboard**

**ELE NÃO IDENTAVA O CÓDIGO !!!**



**E VOCÊ, IDENTOU?**