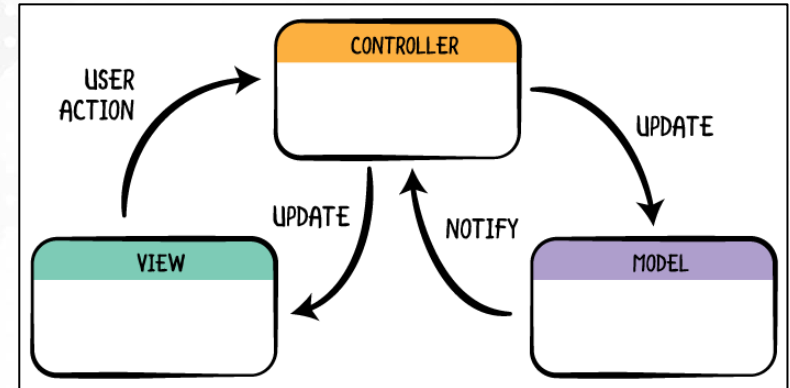


MODEL VIEW CONTROLLER



IMAGINE UM APP COM MUITAS TELAS ...



→ Tela 1



→ Tela 2



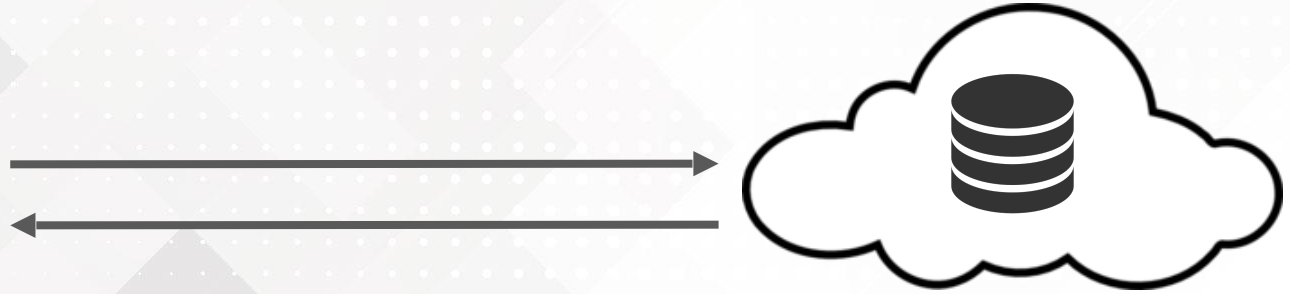
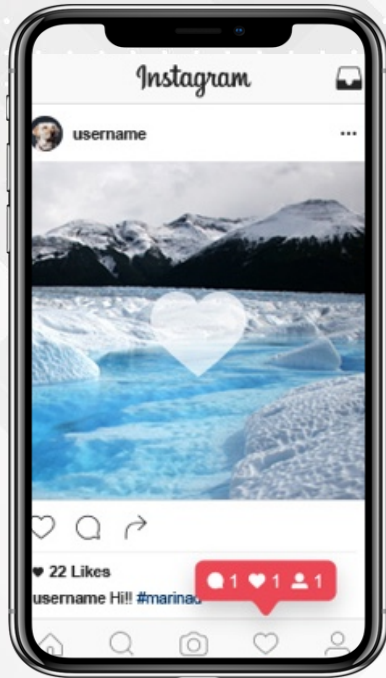
...



→ Tela N

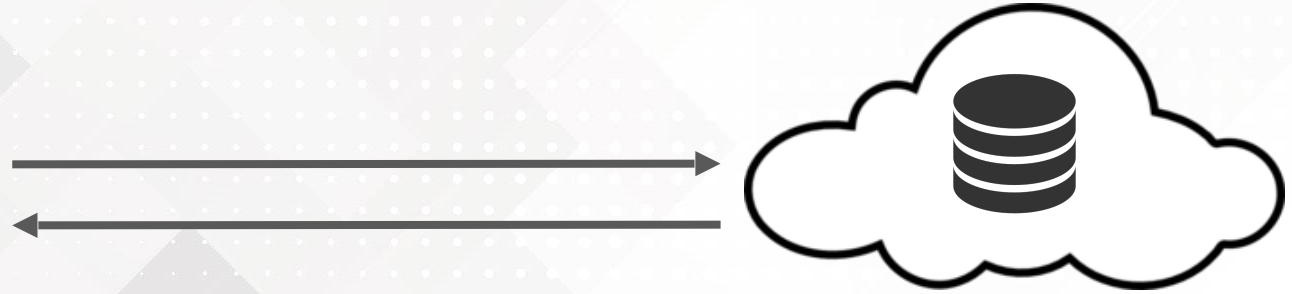
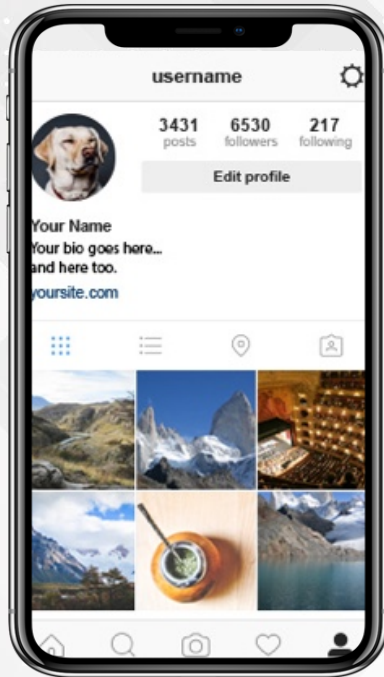


→ Tela 1



- Se conecta com o servidor
- Pede as informações necessárias
- Recebe a resposta
- Processa a respostas em objetos que o app conhece.

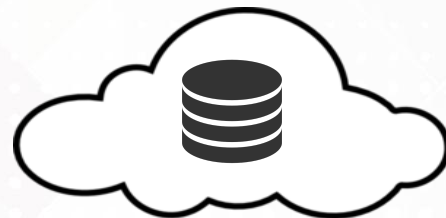
→ Tela 2



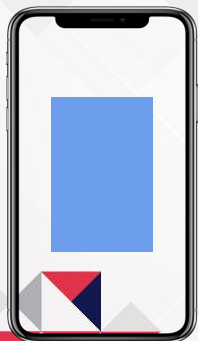
- Se conecta com o servidor
- Pede as informações necessárias
- Recebe a resposta
- Processa a respostas em objetos que o app conhece.

E se for necessária uma modificação? E se o formato em que as informações são retornadas mudar? E se o endereço de onde as obtemos mudar? E se a forma de processá-la mudar?

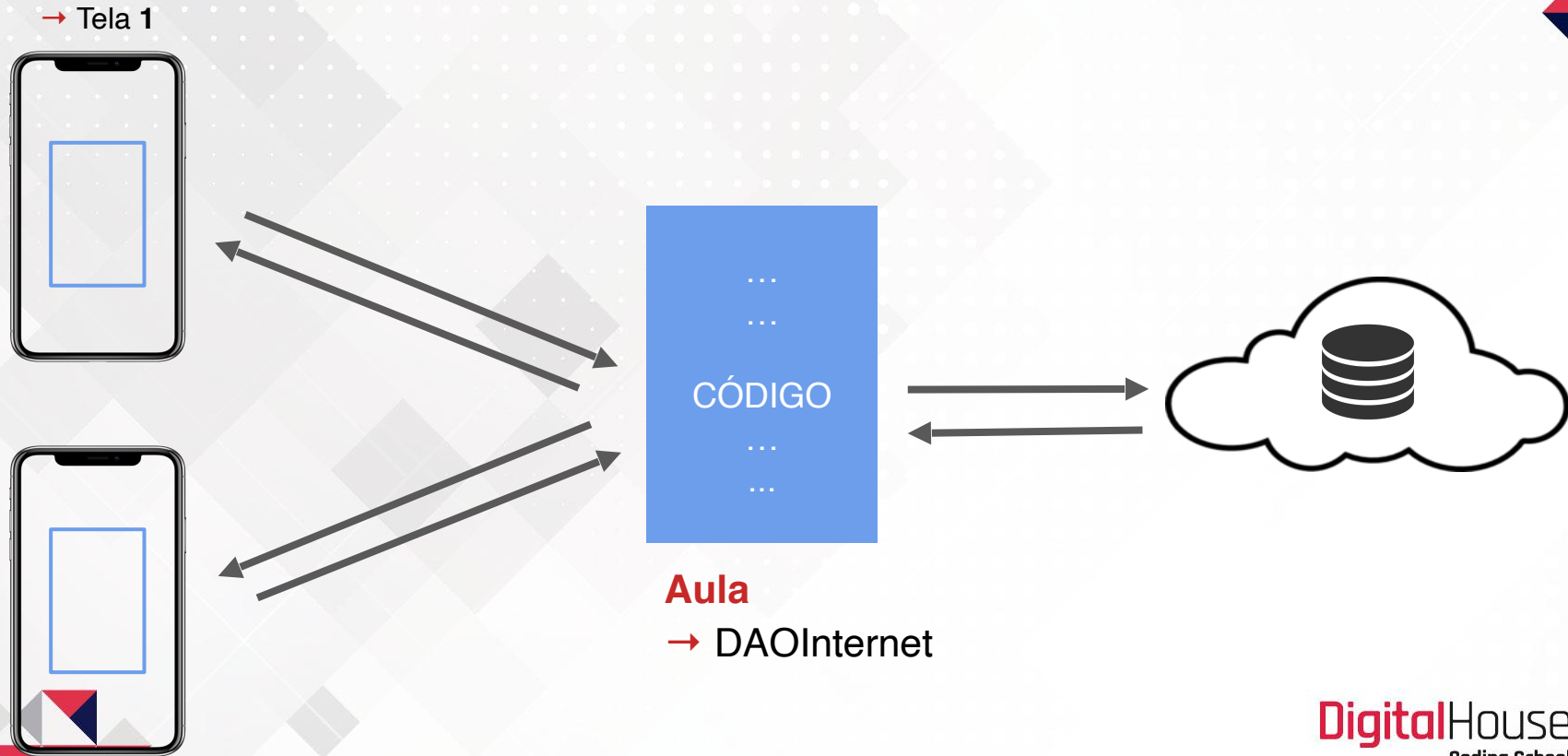
→ Tela 1



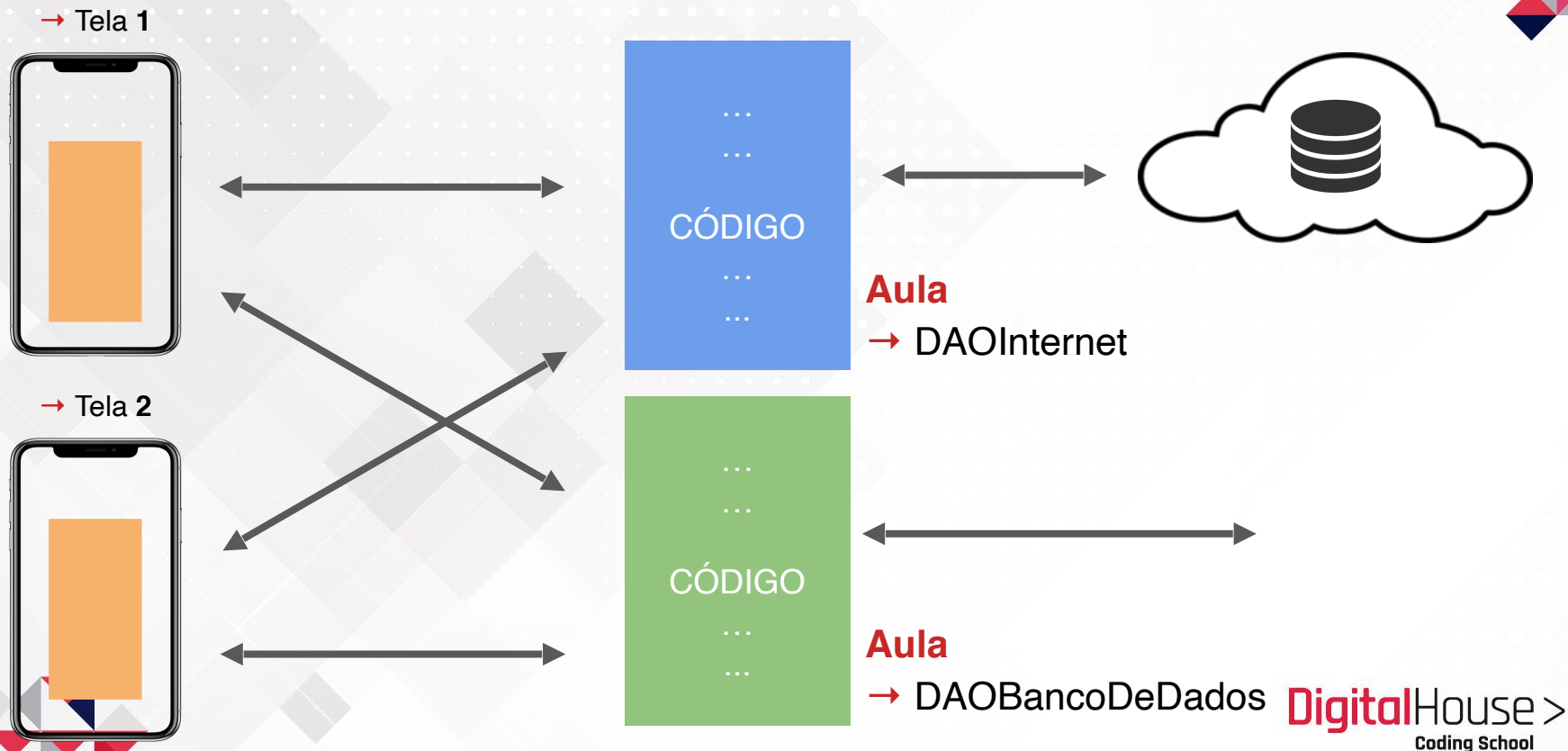
→ Tela 2



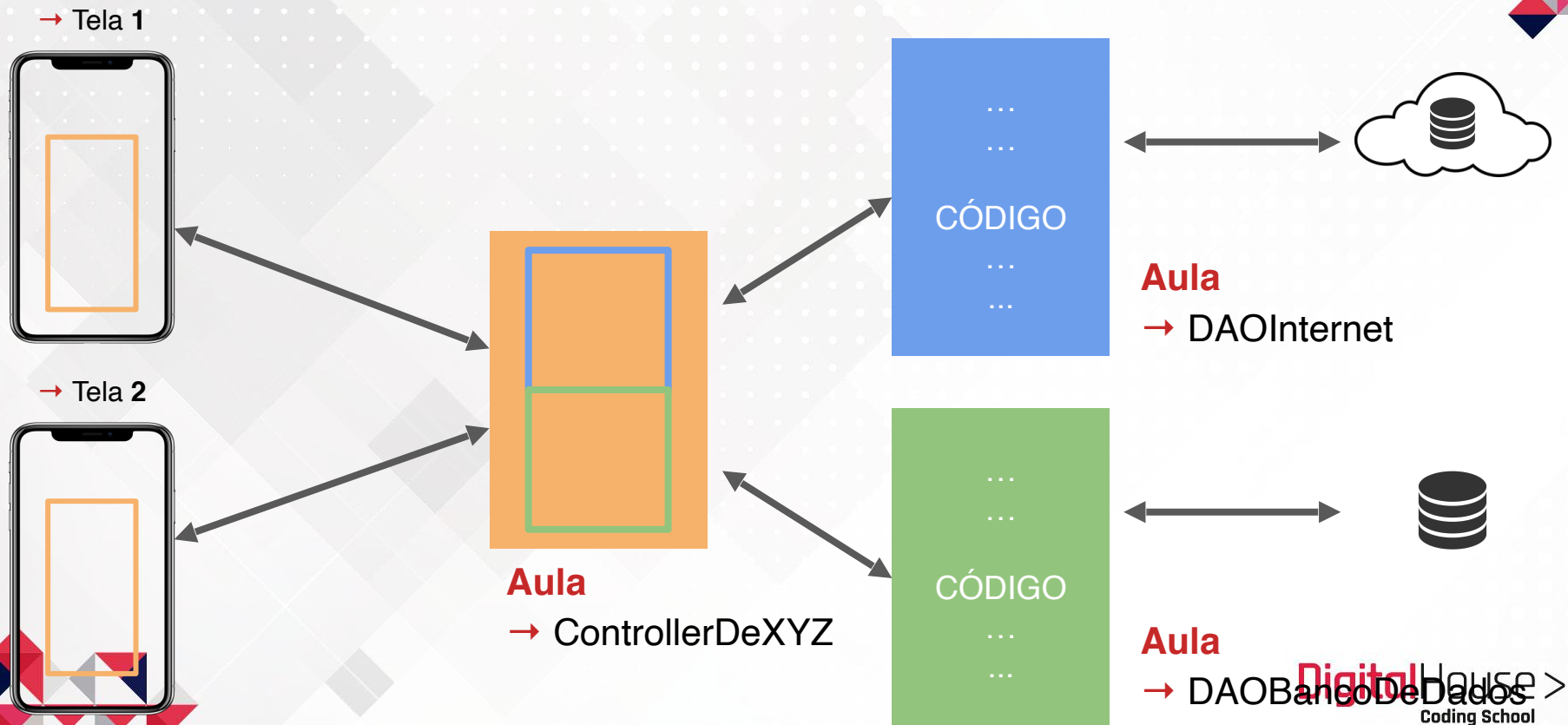
SOLUÇÃO



O que acontece se não houver internet e quisermos obter as informações de um banco de dados local?

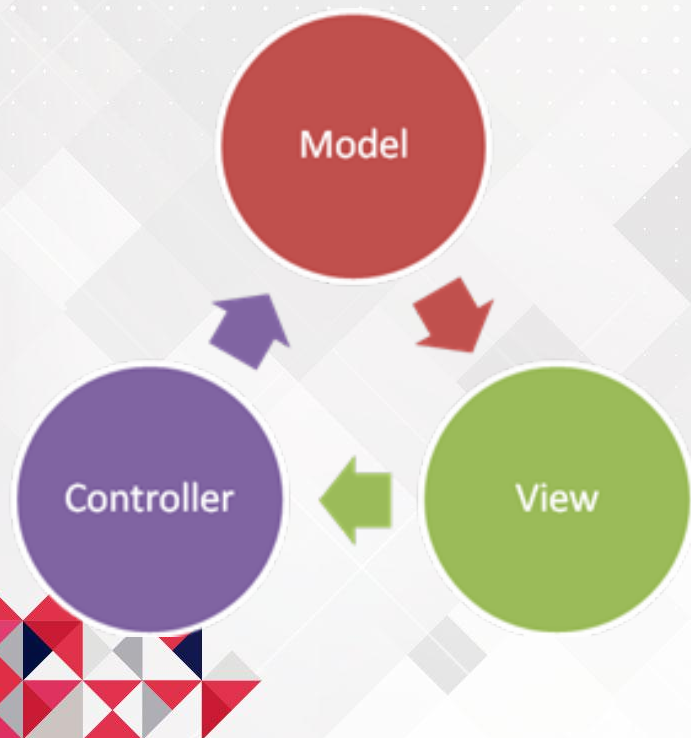


O que acontece se agora for preciso fazer modificações? Esta solução é adaptável a uma mudança de provedor de informações? A uma mudança de formato? A uma mudança de políticas empresariais?



O que é MVC ?

Modelo - Visualização - Controlador



Um padrão de **arquitetura** de software.

Que permite **estruturar** o nosso código.

Dividindo-o em **camadas** interconectadas.

Mantendo **responsabilidades** separadas entre:

- ❑ Os dados do sistema (**Modelo**)
- ❑ A interface de usuário (**Visualização**)
- ❑ A lógica de negócio (**Controlador**)

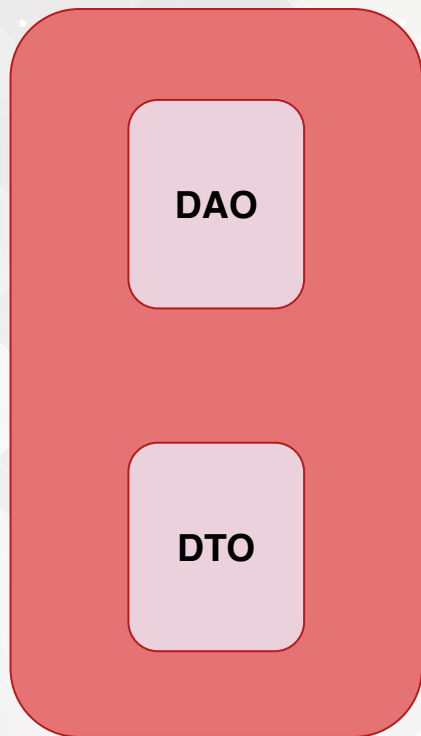
Modelo

O **Modelo** é formado pelos objetos que “modelam o nosso negócio”. Geralmente, **representam as informações** e encapsulam os “dados” do nosso aplicativo.

Também contém os objetos que permitem obter esses dados de diferentes fontes de informação.

Exemplos:

- Em um sistema bancário, as contas e os clientes.
- Em um jogo, os personagens, as peças e as regras.
- Em uma agenda, os contatos.
- Em um calendário, os eventos e as datas.



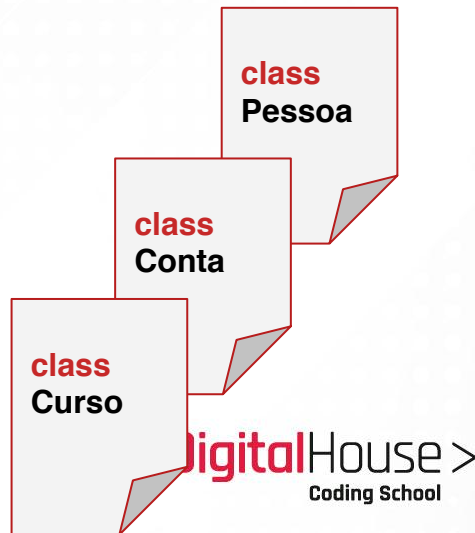
DAO (Data Access Object):

Responsável por gerenciar e administrar o acesso a dados em bancos de dados, arquivos, internet, etc.



DTO (Data Transfer Object):

É o modelo de dados que o aplicativo usa.





VISUALIZAÇÃO

A **Visualização** é a interface de usuário, a camada responsável por apresentar o aplicativo. Sua responsabilidade é a de interagir com o usuário, mostrando coisas e reagindo às suas ações.

Exemplos:

As classes que compõem essa camada são (geralmente) independentes do nosso domínio:

- UIButton
- UILabel
- UITableView
- etc...



CONTROLADOR

O **Controlador** é a camada que age como intermediária entre o **Modelo** e a **Visualização**.

Os objetos que a compõem são os responsáveis por comunicar à **Visualização** quando uma mudança no **Modelo** acontece, assim ela pode se atualizar.

Também são responsáveis por modificar o **Modelo** quando alguma mudança é feita na **Visualização**, pela ação de um usuário (que inseriu um texto em um TextField, fez "Tap" em um botão, moveu um slider, etc...)

