


The background features a city skyline at dusk or dawn, with a body of water in the foreground. Overlaid on this is a network of white lines connecting various nodes. Several nodes are highlighted with red circular icons: a central node with a white gladiator helmet, and other nodes with icons of a laptop, a smartphone, and a tablet. In the bottom right, there is a red wireframe structure resembling a stylized 'e' or a molecular model.

# Embarcadero Conference





# Orientação à Objetos: De uma vez por todas!

Samuel “Muka” David

# [Samuel “Muka” David]

- Instrutor Oficial Delphi
- MVP Embarcadero
- Certificado Delphi [7.. XE Master]
- Colaborador na elaboração da prova Master
- Palestrante Delphi Conference [2009..2016]
- Especialista em Refatoração





[Quem tem medo do OOP?]



# Sou RAD sem OO

- Dilema:

*Por que eu tenho que usar OO se o Delphi me dá tudo pronto nos eventos?*



# Sou RAD sem OO

- Dilema:

*Por que eu tenho que usar OO se o Delphi me dá tudo pronto nos eventos?*



- Resposta:

*O Delphi só é RAD porque é todo orientado a objetos.*



# Programação Orientada a Objetos

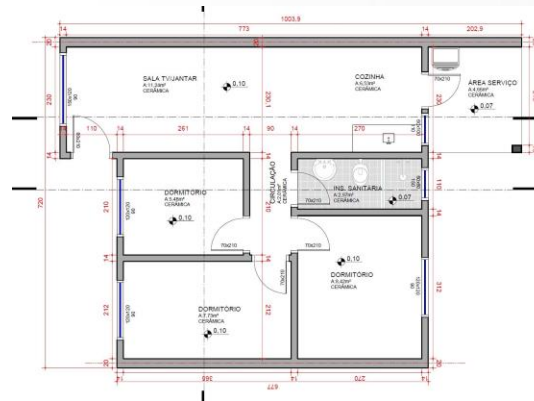
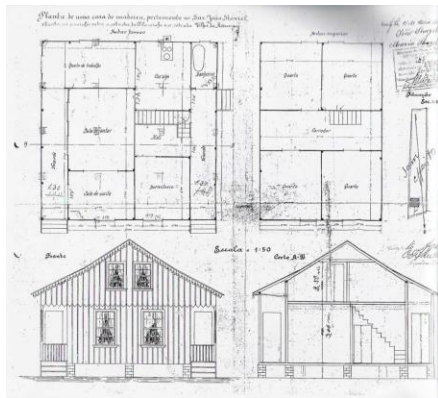
- O que é ?
- De onde vem?
- Do que se alimentam?





# Classe X Objeto

- O Objeto nasce a partir de uma definição, esta definição é chamada de Classe.
- Analogia:
  - Planta de uma Casa (Classe) x Casa construída (Objeto).
  - Receita de Bolo (Classe) x Bolo

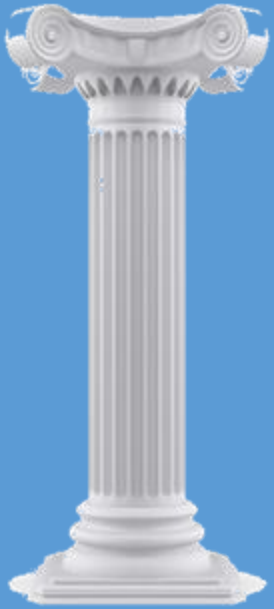




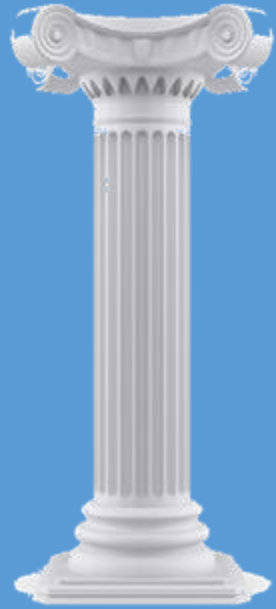
# Quais os pilares da Orientação a Objetos?



# Pilares da OO



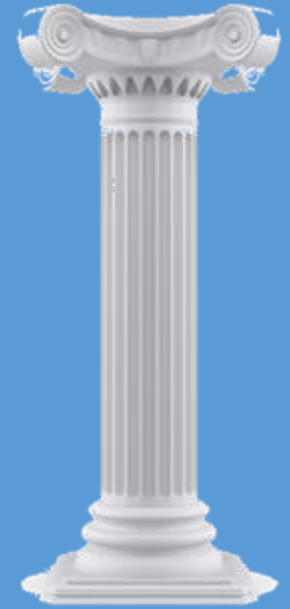
Abstração



Herança



Encapsulamento



Polimorfismo



# Abstração

- Identidade (Instância)
- Características (Propriedades)
- Executar ações (métodos)



# O Objeto e a Memória

- Cada objeto tem sua própria residência com seu endereço.
- Objeto1 := TMinhaClasse.Create;
- Objeto2 := TMinhaClasse.Create;
- Objeto3 := TMinhaClasse.Create;



# O Objeto e a Memória

- Cada objeto tem sua própria residência com seu endereço.
- Objeto1 := TMinhaClasse.Create;
- TMinhaClasse.Create;
- Objeto2 := TMinhaClasse.Create;
- TMinhaClasse.Create;
- Objeto3 := TMinhaClasse.Create;

# Valor x Referência

- Cada objeto tem sua própria residência com seu endereço.
- Objeto1 := TMinhaClasse.Create;
- Objeto2 := Objeto1;
- Objeto3 := Objeto2;



# Herança

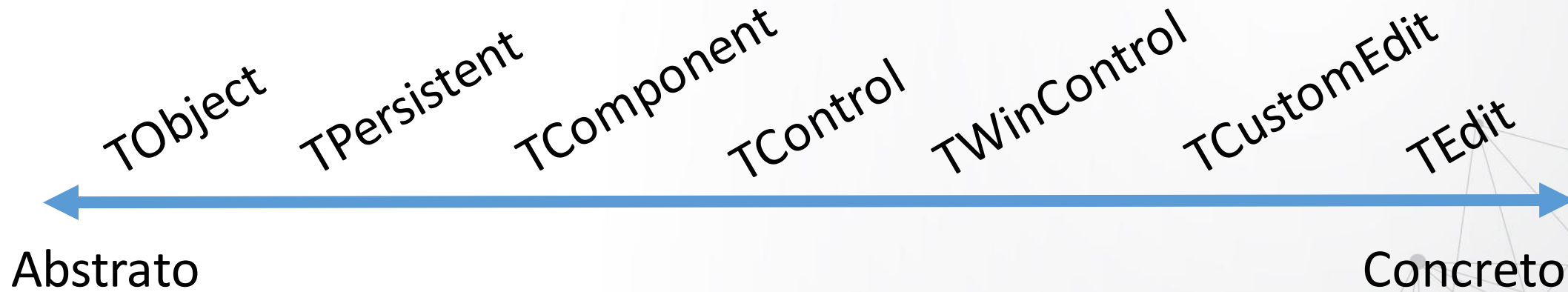
- Receber e estender características



lizclimo.tumblr.com

# Abstrato X Concreto

- Quanto mais Abstrato, mais próximo do início de uma hierarquia estamos, neste ponto tentamos ser o mais genérico possível;
- Quanto mais Concreto, mais longe do início de uma hierarquia estamos, neste ponto especificamos a classe para atender demandas mais específicas.





# Encapsulamento

- Ocultar funcionamento;
  - Isolar funcionalidade;
  - Distribuir responsabilidades.
- 
- Um Carro é composto de vários objetos encapsulados, como rodas, radio, motor... Vc conhece os objetos, mas não tem necessidade de entender sua arquitetura para poder utiliza-los



# Polimorfismo

- Classes concretas podem ser representadas por classes mais abstratas, mantendo as funcionalidades do objeto atribuído.



# O que não é OO

- Declarar métodos e variáveis diretamente na unit
- Programar apenas orientado a eventos



# Coisas legais do OO

- Componentes
- Threads
- Atributos
- Generic
- RTTI
- ORM



@ mukadavid@gmail.com

f www.facebook.com/mukadavid

[OBRIGADO]



Embarcadero conference





# Embarcadero Conference

