



embarcadero conference

The logo for Embarcadero Conference. It features the word "embarcadero" in a white, sans-serif font, with the "e" in red. Below it, the word "conference" is in a large, bold, red, sans-serif font. To the left of the text is a circular graphic with a red background and a white spiral design. Four red spheres are connected to the spiral by thin lines. The background of the slide shows a person's hands holding a smartphone, with a network diagram overlaying the image.

embarcadero conference

The logo for Spring4D, consisting of two thick, parallel red diagonal lines.

Spring4D - Arquitetura OOP para Aplicações Delphi

Cesar Romero

// Apresentação



- Cesar Romero
 - Líder Técnico e Arquiteto de Software
 - Trier Sistemas, Tubarão - SC
 - Pós Graduado em Software Orientado a Objetos - PUC PR
 - Desenvolvedor Delphi desde 1995



// Agenda



- Delphi - Ferramenta de Produtividade
- Problemas Comuns nos Projetos
- RAD x OOP
- Solução - Arquitetura de Sistemas

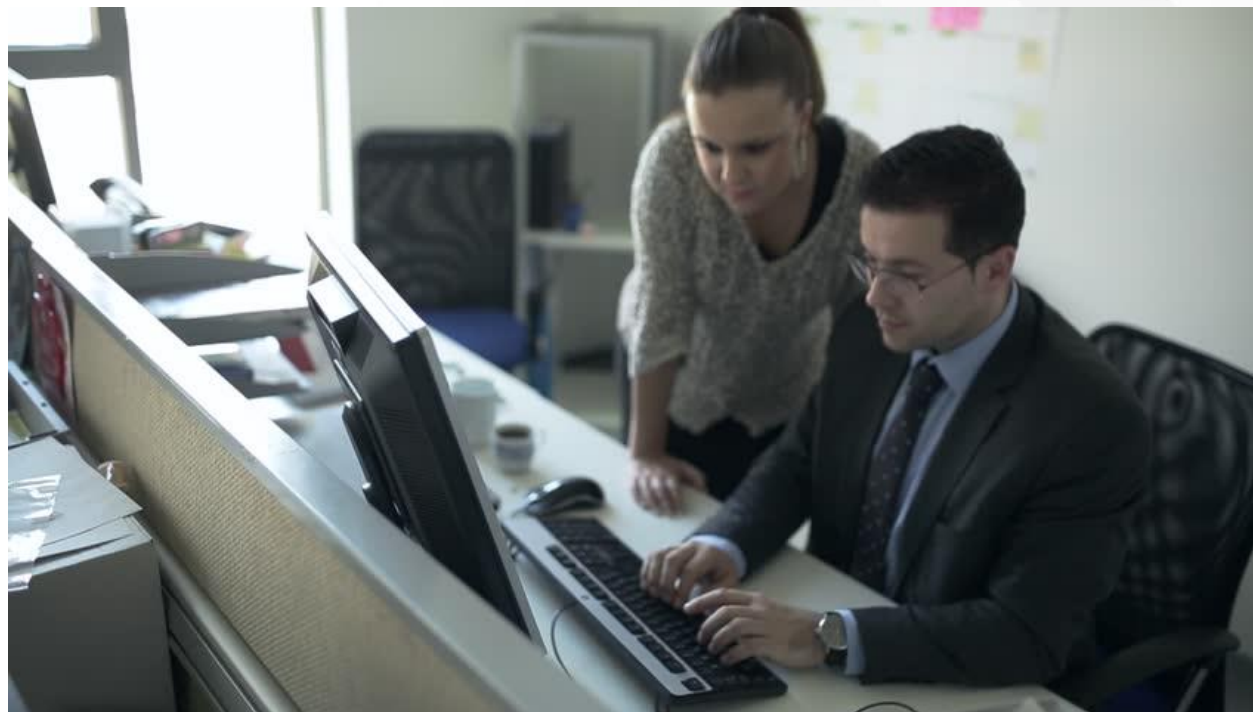




Delphi - Ferramenta de Produtividade



- Pequenas e grandes Empresas criadas do zero
- Ideia ou uma oportunidade de negócio
- Protótipo





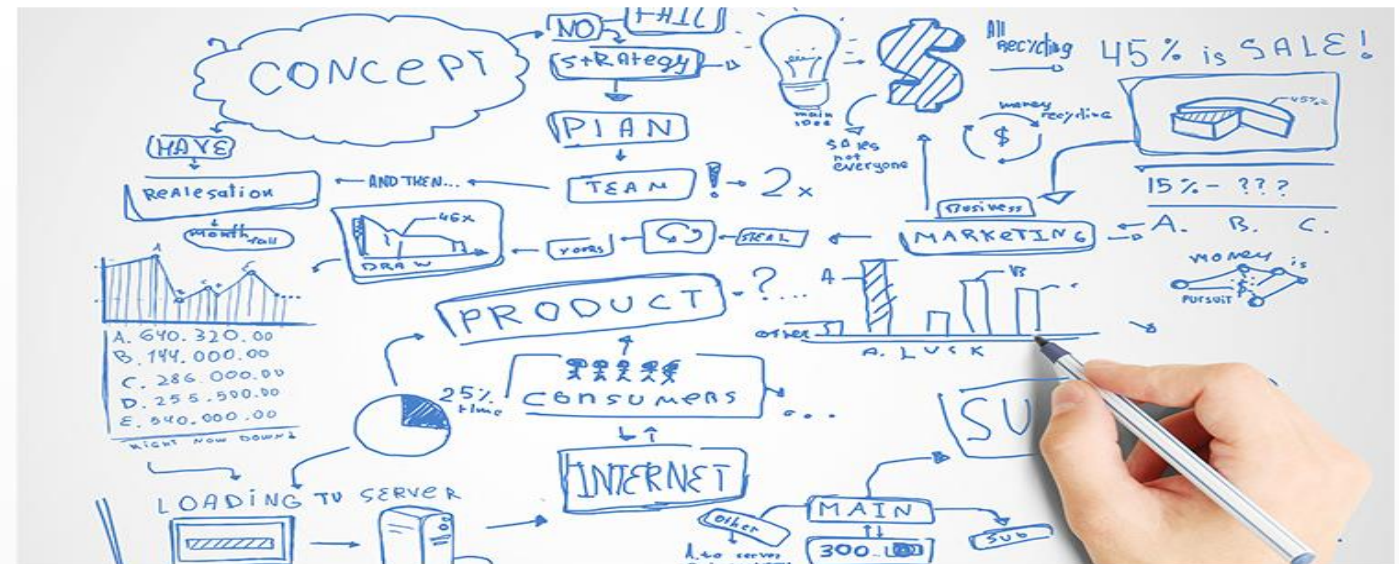
Delphi - Ferramenta de Produtividade



- RAD
- Biblioteca Visual
- Banco de Dados
- Cross Plataforma



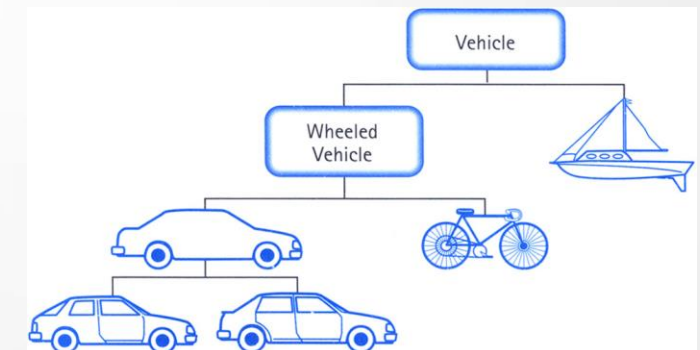
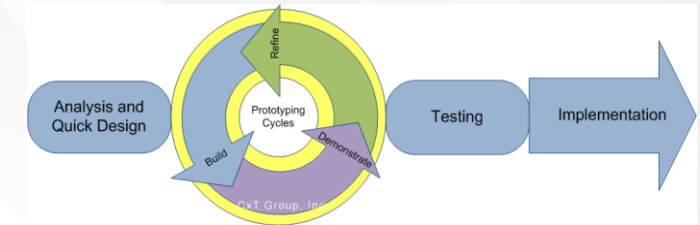
-
- The image shows a hand-drawn mind map on a whiteboard, illustrating a business plan or concept. The central theme is "CONCEPT".
- Flowchart and Key Elements:**
- CONCEPT** (Central cloud)
 - NO ILL** (Box)
 - Strategy** (Box)
 - PLAN** (Box)
 - TEAM** (Box) with **2x** (Text)
 - MARKETING** (Box)
 - BUSINESS** (Box)
 - MONEY RECYCLING** (Text)
 - 45% is SALE!** (Text)
 - 15% - ???** (Text)
 - A. B. C.** (Text)
 - MONEY IS PURSUIT** (Text)
 - CONSUMERS** (Box)
 - INTERNET** (Box)
 - MAIN 300-100** (Box)
 - SUB** (Box)
 - LOAN** (Box)
 - A. to server** (Text)
 - REALISATION** (Box)
 - MONTH WALL** (Text)
 - DRAW** (Text)
 - 46x** (Text)
 - 25% time** (Text)
 - RIGHT NOW DOWN 2** (Text)
 - LOADING TV SERVER** (Text)
- Charts and Graphs:**
- A bar chart with values: A. 640, 320, 0.00; B. 144, 000.00; C. 286, 000.00; D. 255, 500.00; E. 540, 000.00.
 - A pie chart labeled "RIGHT NOW DOWN 2".
 - A line graph labeled "MONTH WALL".
 - A bar chart labeled "DRAW".
 - A bar chart labeled "A. LUCK".
- Handwritten Notes:**
- "AND THEN..." (Text)
 - "SALES not everyone" (Text)
 - "money recycling" (Text)
 - "45% = 15% - ???"
 - "MONEY IS PURSUIT" (Text)
 - "25% time" (Text)
 - "RIGHT NOW DOWN 2" (Text)
 - "LOADING TV SERVER" (Text)



// RAD x OOP



- Paradigmas diferentes
 - Mesmo resultado
- RAD - Maior Produtividade e Menor Custo
 - Componentes, Form Designer, DataSet...
- OOP - Organização
 - Frameworks
 - MVC, MVP, MVVM, ORM, VTF, OPF...



RAD X OOP





RAD x OOP



// Solução

- Padronização
- Arquitetura de Sistema
- Produtividade
- Testes

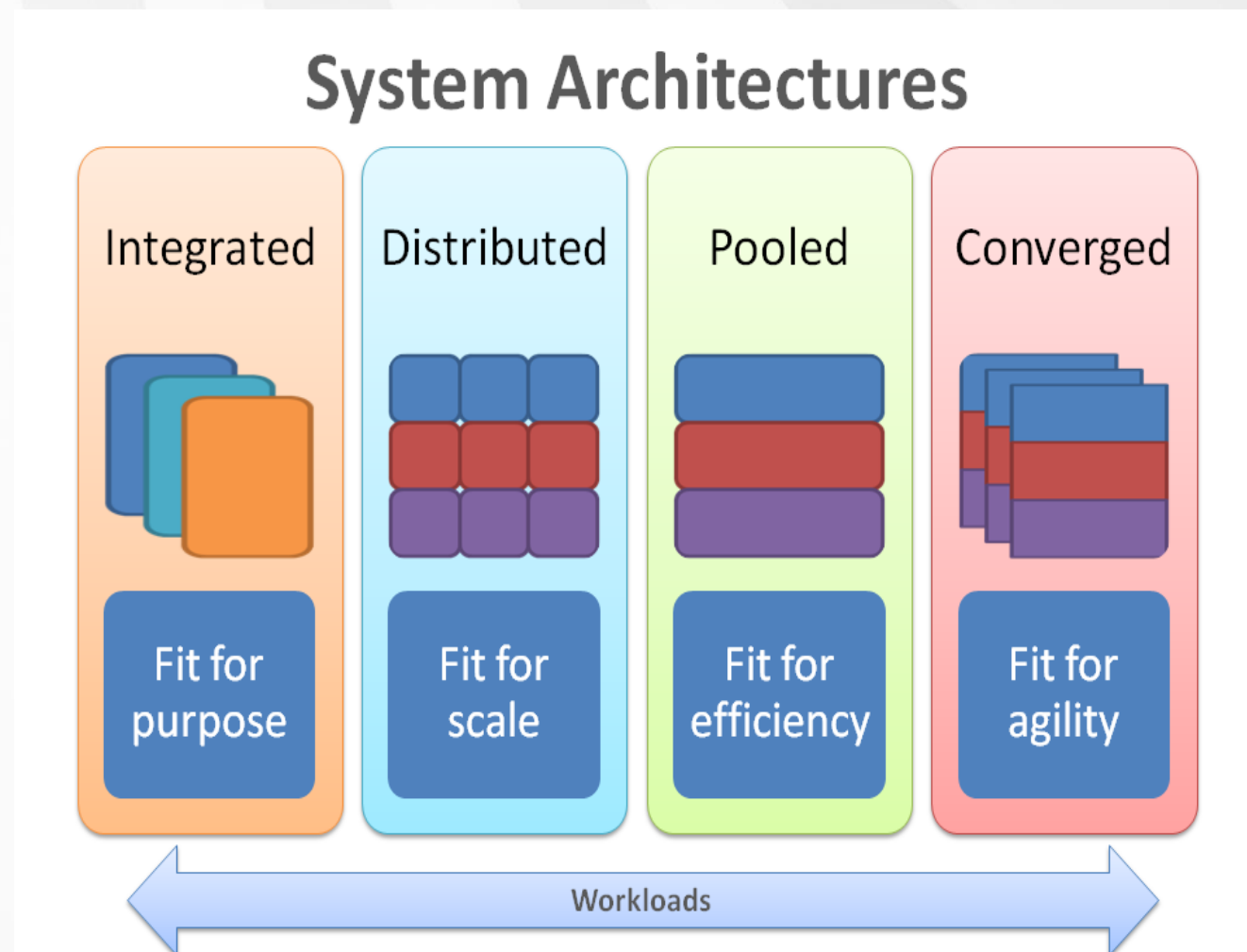


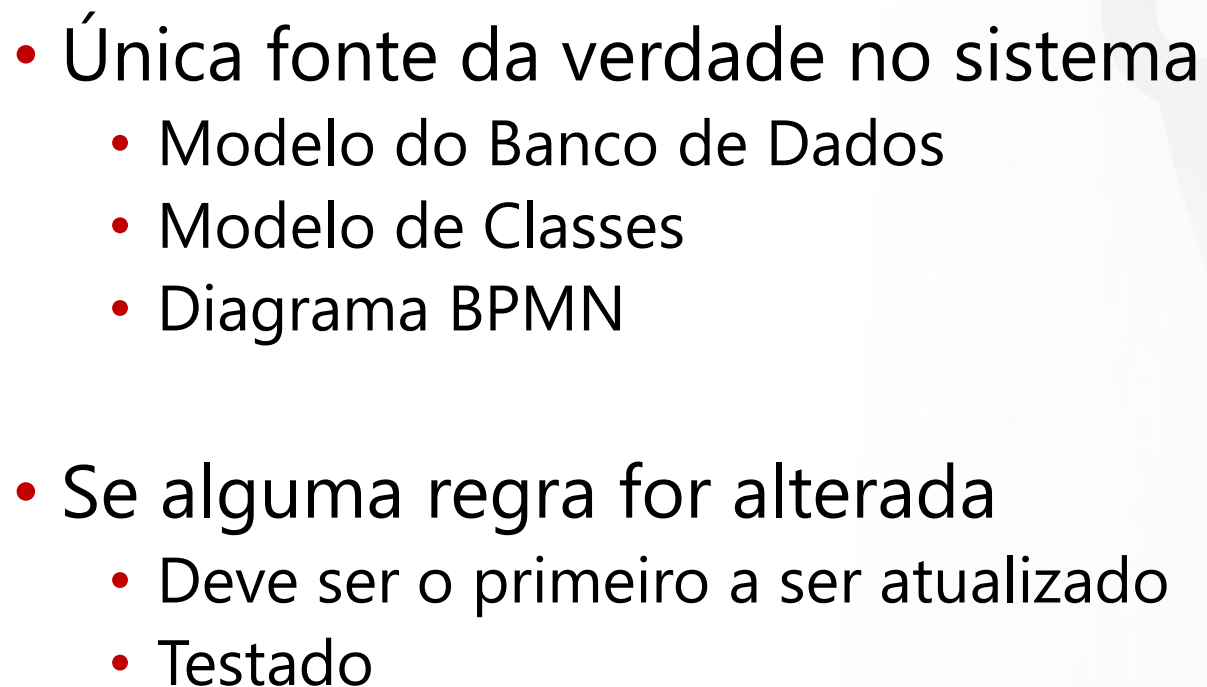
-

// Arquitetura



- Modelagem
- Camada de Persistência
- Regra de Negócios
- Camada de Controle
- Camada de Apresentação
- Interceptação
- Serviços



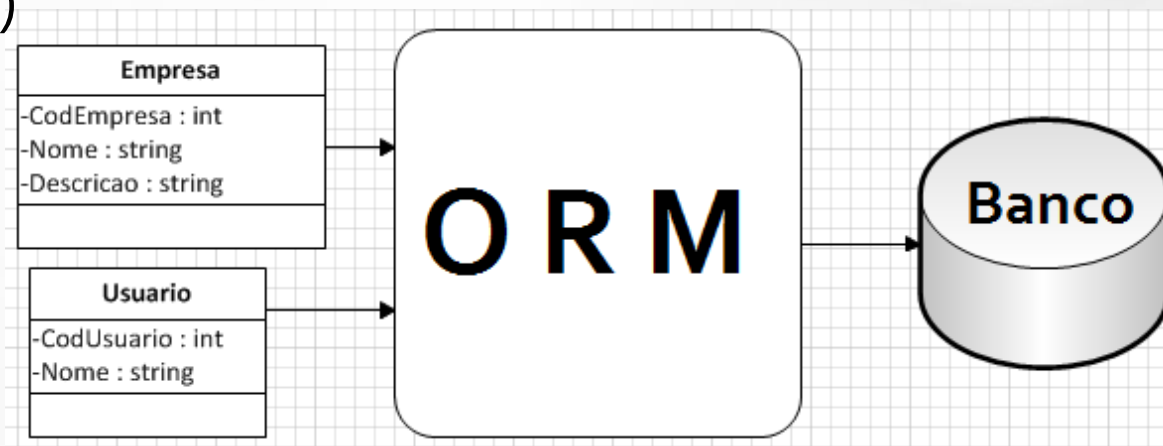




Camada de Persistência



- Entidades
 - Objetos e suas coleções
 - Registros de banco de dados
 - Pode conter ou interagir com outras entidades
- Conteúdo
 - Data Module
 - Conexão com Banco de Dados (**Provider**)
 - Query, DataSet
 - Instruções SQL
 - Classes
 - Metadados

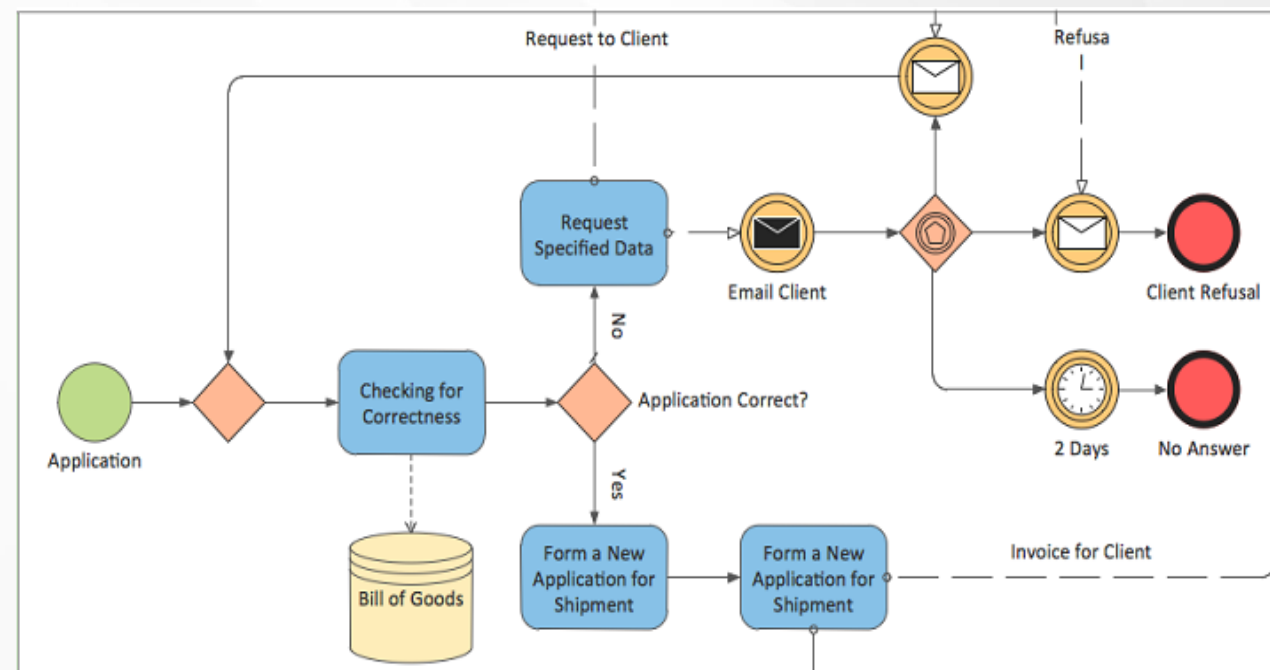




Regra de Negócios - Model



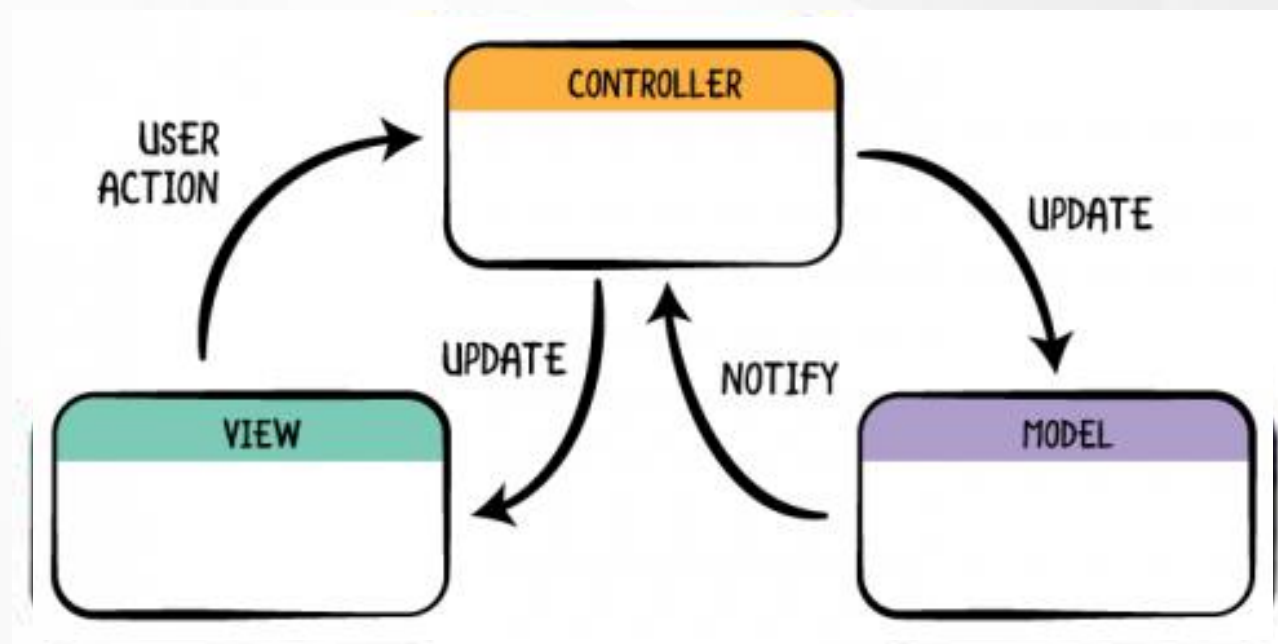
- Camada de conhecimento do sistema
 - Contém as rotinas de regra de negócio
 - Deve ter como objetivo a Camada de Persistência
 - Pode conter e interagir com outros Models
 - Envia notificação das alterações
- Conteúdo
 - Data Module
 - Query, DataSet
 - Classes
 - ObjectDataSet





Camada de Controle - Controller

- É a ligação entre as camadas de apresentação e de conhecimento
 - Organizado em Classes com Comandos
 - Cria e utiliza outros Controllers, Models, Entidades, Middleware, Serviços (DI)
 - Disponibiliza os dados para a camada de apresentação
- Conteúdo
 - Data Module
 - Action List e Actions
 - Classes

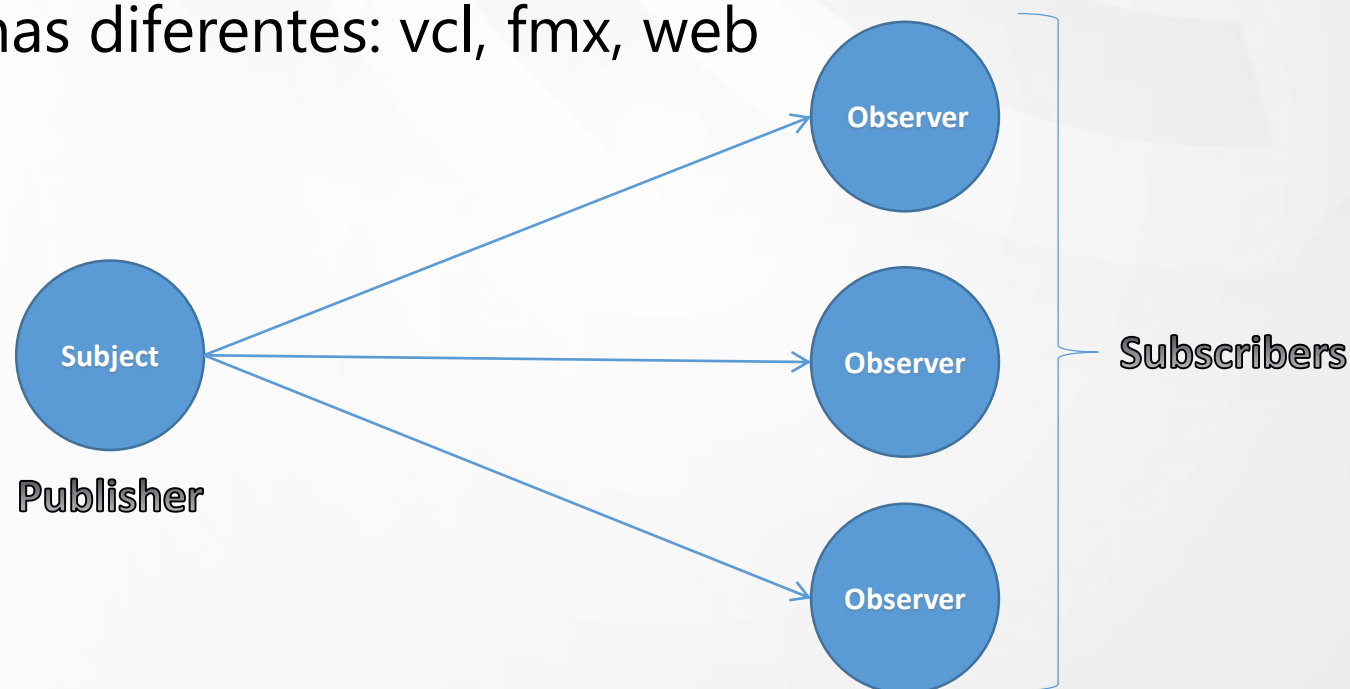




Camada de Apresentação - Views



- Representação visual da camada de conhecimento
 - Recebe notificação das alterações (**Observer**)
 - Pode conter código para atualizar o Estado da View
 - Validação dos dados recebidos (**Interceptação**)
 - Executa comandos no Controller a partir das ações dos usuários
 - Não é reutilizada entre plataformas diferentes: vcl, fmx, web
- Conteúdo
 - Classes
 - Formulários e Frames
 - Componentes Visuais
 - Data Source - Binding

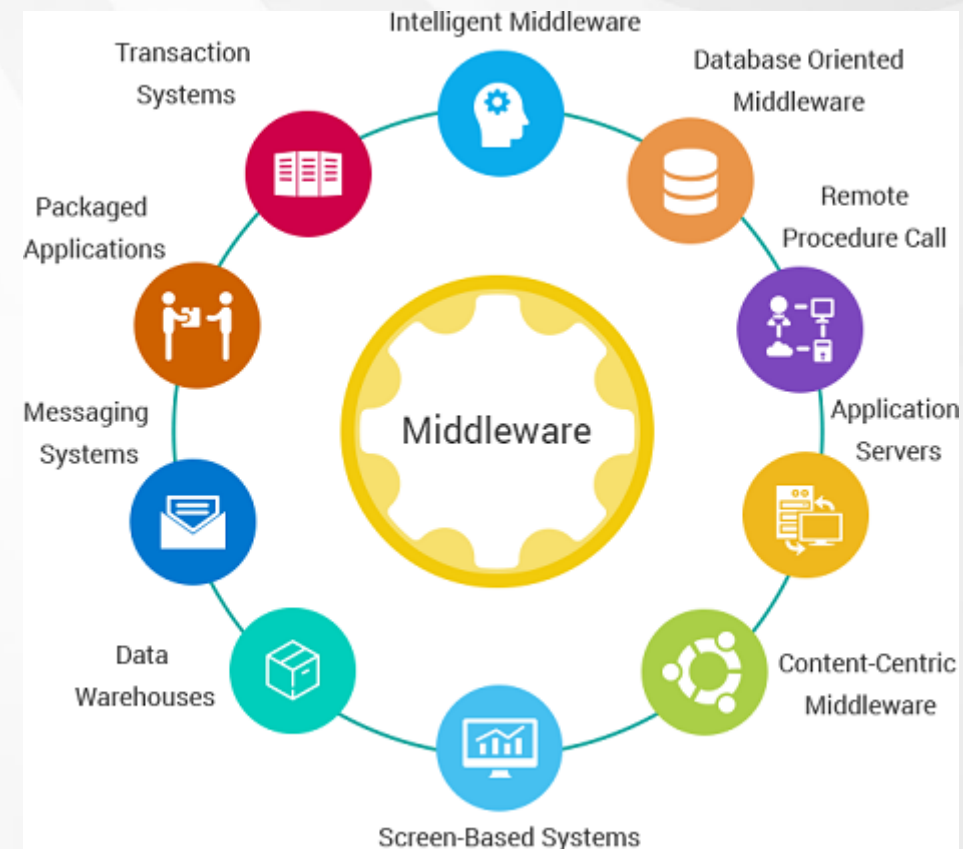




Interceptação - Middleware



- Camada Intermediária (DI)
 - Geralmente dinâmica - plugável
 - Intercepta os comandos e ações
 - Pode ser encadeado
- Conteúdo
 - Interfaces, Classes
 - DataModules e Componentes não visuais
- Exemplos
 - Tratamento de Exceções
 - Autenticação e Autorização
 - Filtro e validação da entrada de dados
 - Log das operações





Serviços - Providers



- Camada de Integração
 - Ligação entre a aplicação e recursos externos
 - Integração com equipamentos: Impressoras fiscais e balanças
 - Dados: Arquivos, Zip, Banco de Dados, REST
 - Enviar E-mail, SMS
 - Configuração: Registro do Windows, Arquivos
- Conteúdo
 - Interfaces
 - Classes
 - DataModules
 - Componentes não visuais



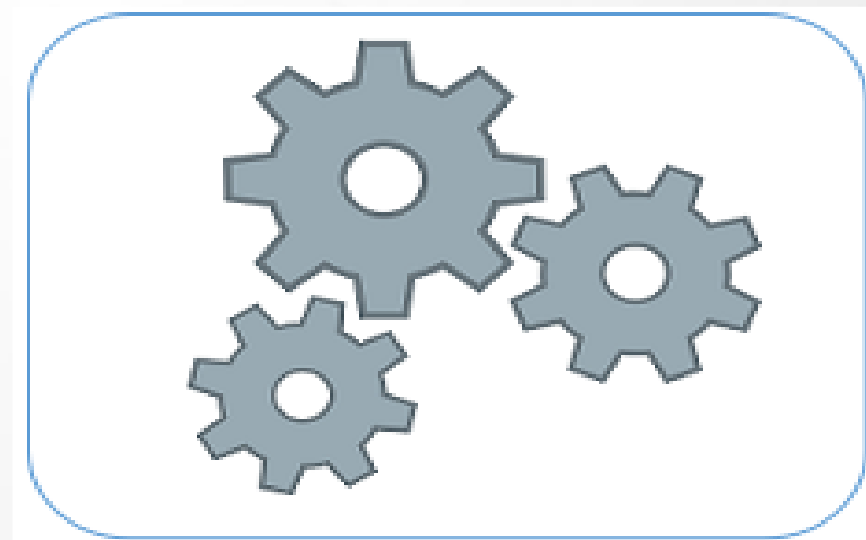


Produtividade - Gerar Código



- Evitar código boilerplate

“Código boilerplate se refere a seções de código que devem ser incluídas em muitos lugares com pouca ou nenhuma alteração.” [wikipedia](#)
- A geração de código pode chegar até a 80% da aplicação
 - Estática
 - Dinâmica
- O Código gerado não deve ser alterado
- Personalização
- Templates



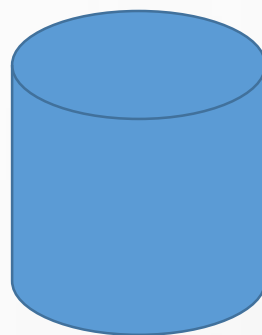


Produtividade - Gerar Código Estático



- Banco de dados
 - Instruções SQL
- Persistência
 - Dicionário de Dados - Metadados
 - Mapeamento ORM
- Apresentação
 - View
 - Validação de Input
- Testes

Metadados



Template

```
1 unit {($ namespace)}.Persistence.Entity;  
2 interface  
3   Entity;  
4   use  
5     System.Classes,  
6     Spring.Persistence.Mapping.Attributes,  
7     {($ namespace)}.Persistence.Commits;  
8   type [[TableClass]] {($ common_class)}  
9     {($ entity_class)}  
10    {($ table)};  
11  implementation  
12  end;  
13  {($ class_base)} {($ common_class)}  
14  TCommon {($ entity_name)} = class(TObject)  
15    strict private {($ columns)}  
16    {($ table)} {($ columns)}  
17    protected {($ columns)}  
18    {($ class_property)} {($ columns)}  
19    end {($ common_class)}  
20  {($ field_de_class_base)}  
21  {($ nullable_class_field)} {($ attribute_name)}  
22  {($ class_property)} {($ property)} {($ attribute_name)}
```

Código

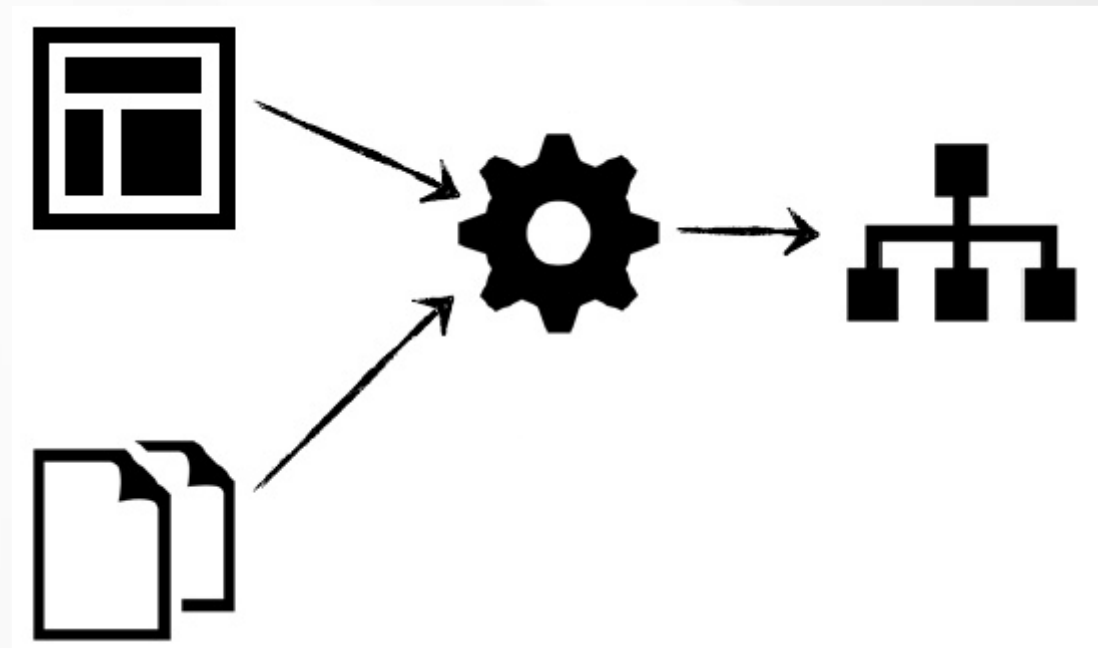
```
1 type  
2   TComercio = class(TObject)  
3     strict private  
4       FIdade: Integer;  
5       FNome: string;  
6       FEndereco: nullable Integer;  
7     protected  
8       property Idade: Integer read FIdade write FIdade;  
9       property Nome: string read FNome write FNome;  
10      property Endereco: nullable Integer read FEndereco;  
11    end;  
12  {($ entity)}  
13  {($ table)} {($ columns)}  
14  {($ class_base)} {($ common_class)}  
15  {($ field_de_class_base)} {($ nullable_class_field)} {($ attribute_name)}  
16  {($ class_property)} {($ property)} {($ attribute_name)}
```



Produtividade - Gerar Código Dinâmico



- Configurável
- Utiliza Framework
- Em tempo de execução
 - View
 - Interceptação
 - Ativar ou desativar Serviços





cesarliws@gmail.com



/cesarliws



@cesarliws

embarcadero
conference

// **obrigado**



embarcadero conference