



Embarcadero Delphi Conference

Técnicas de
desenvolvimento
utilizando mapeamento
objeto-relacional
(TMS Aurelius)



Wagner Rafael Landgraf



CONCEITOS



**Embarcadero
Delphi Conference**



- Object Relational Mapping
 - Mapeamento Objeto <-> Relacional
 - Tabelas <-> Classes
 - Registros <-> Objetos
 - Colunas <-> Atributos (propriedades/fields)



- Vantagens
 - Regras de negócio com objetos
 - Sem uso de SQL, fácil alternância de bancos de dados
 - Fácil migração para multi-camadas (REST server)
 - Fácil construção do client (sem datasets, sem conexões, uso de objetos em JavaScript, etc.)



- TMS Aurelius
 - Framework ORM disponibilizado pela TMS
 - TMS Software (<http://www.tmssoftware.com>)
 - Multi-banco, multi-plataforma, multi-biblioteca
 - Suporte a servidores REST (serialização JSON)
 - Data binding (dataset ou live bindings)
 - Arquitetura preparada para eventual suporte a Mobile Studio (iOS, Android)



PASSO-A-PASSO APLICAÇÃO COM ORM



**Embarcadero
Delphi Conference**



- APLICAÇÃO DE TÉCNICAS PARA CONSTRUÇÃO DE UM SOFTWARE UTILIZANDO ORM (TMS AURELIUS)
- SOFTWARE PARA CONTROLE DE INCIDENTES (BUGS)



- Passo 01
 - Criação de entidades simples
 - PODO (Plain Old Delphi Objects)
 - Tela simples pra alteração das propriedades do objeto (exemplo com TUsuario)



- Passo 02
 - Uso de Repositório como camada adicional para interface do usuário
 - Repositório abstrai onde e como os dados estão sendo gravados
 - Gerenciamento da memória no repositório
 - Exemplo: repositório em memória (TRepositorioMemoria)



- Passo 03
 - USO DO AURELIUS para persistir objetos
 - Mapeamento das classes (objeto <-> relacional)
 - Novo repositório para salvar em banco de dados (TRepositorioDatabase)
 - Banco de dados arquivo local usando SQLite
 - Opcional: uso de factories pra facilitar chaveamento entre repositórios



- Passo 04
 - Uso de TAureliusDataset pra desenvolvimento RAD (edição/visualização de objetos)
 - Exemplo com listagem de projetos (TProjeto)
 - Uso de package separado com as classes das entidades para facilitar uso em design-time
 - (Novos métodos no repositório pra manipulação de TProjeto)



- Passo 05
 - Tela para edição de incidentes
 - Uso de campos especiais do TAureliusDataset
 - TAureliusEntityField
 - Lookup fields
 - Subpropriedades (campo “Projeto.Nome”)
 - (Novos métodos no repositório pra manipulação de Tincidente)



- Passo 06
 - Listagem de incidentes
 - Mais campos especiais
 - Sufixo “.EnumName” (para tipos enumerados)
 - Conceito de `TAureliusDataset.Current<T>`
 - Uso de queries com Aurelius (busca de incidentes por projeto)



- Passo 07
 - Uso de listas de objetos (coleções) – comentários do incidente
 - Dataset com mestre-detalle (campo do tipo dataset) para edição dos comentários
 - Exemplo de lógica nas entidades (data/hora do comentário)



- Passo 08
 - Alteração para uso de Firebird client/server
 - Exemplo da portabilidade – software não altera



- Passo 09
 - Criação de um servidor REST usando DataSnap
 - Uso do TJsonManager para converter entidades Aurelius em objetos JSON e vice-versa
 - Servidor usa conceito de repositório idêntico à aplicação anterior (reutilização de código)
 - Criação dos métodos do servidor
 - Teste via browser (recebimento de objetos JSON via URL)



- Passo 10
 - Finalizando servidor REST com todos os métodos (mesmos métodos do repositório)
 - Criação de repositório para acessar servidor REST
 - Alteração da aplicação cliente/servidor – basta mudar repositório
 - Exemplo de portabilidade – software cliente continua sem maiores alterações



- Passo 11
 - Exemplo de utilização dos objetos em outros clientes (JavaScript, HTML)
 - Representação JSON dos objetos Aurelius são simples e portáteis



RESUMO DAS TÉCNICAS



**Embarcadero
Delphi Conference**



- Classes de domínio simples (PODO – Plain Old Delphi Objects)
- Uso de repositório (portabilidade)
- Interface visual simples, manipulando objetos e sem conhecimento da camada de acesso a dados
- Uso de datasets para facilitar edição via interface visual
- Servidor REST retornando JSON para uso em clientes de outras plataformas (JavaScript, etc.)
- Padrões de projetos (Abstract Factory, Dependency Injection, etc.)



PERGUNTAS



**Embarcadero
Delphi Conference**

Obrigado!



Embarcadero
Delphi Conference

Wagner Rafael Landgraf

Gerente de Produtos – TMS Software

Graduação em Engenharia Eletrônica, Mestre em Informática Industrial

info@tmssoftware.com

<http://www.tmssoftware.com>

<http://facebook.com/tmssoftware>

<http://twitter.com/tmssoftwarenews>

Página TMS Aurelius:

<http://www.tmssoftware.com/site/aurelius.asp>



- Portal de Treinamentos e Vagas – <http://www.edobrasil.net/treinamentos>
- Embarcadero Developer Network - <http://edn.embarcadero.com>
- Diretório de MVP's - <http://www.embarcadero.com.br/mvp-directory>
- Documentação dos Produtos - <http://docs.embarcadero.com>
- CodeRage 7 - <http://www.embarcadero.com/coderage>
- YouTube - <http://youtube.com/user/embarcaderodobrasil>
- Twitter - <https://twitter.com/EmbarcaderoBR>
<http://twitter.com/EmbarcaderoTech>
- Blogs: <http://blogs.embarcadero.com>
- Facebook:
<https://www.facebook.com/pages/Embarcadero-Delphi-Brasil/399151510134179>
- atendimento@embarcadero.com.br
- (11) 5643-1333