



# Embarcadero Conference





# Melhores práticas com Datasnap

[Kelter Merlotti]

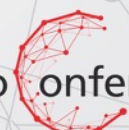
# [Kelver Merlotti]

- Embarcadero MVP
- Coordenador Editorial da Active Delphi
- Consultor e Instrutor independente
- Especialista em Mobile e Multicamadas
- Intitulado “Detentor de todas as respostas” #DeusTaVendo

@ kmerlotti@gmail.com

@kmerlotti

kmerlotti

Embarcadero  conference

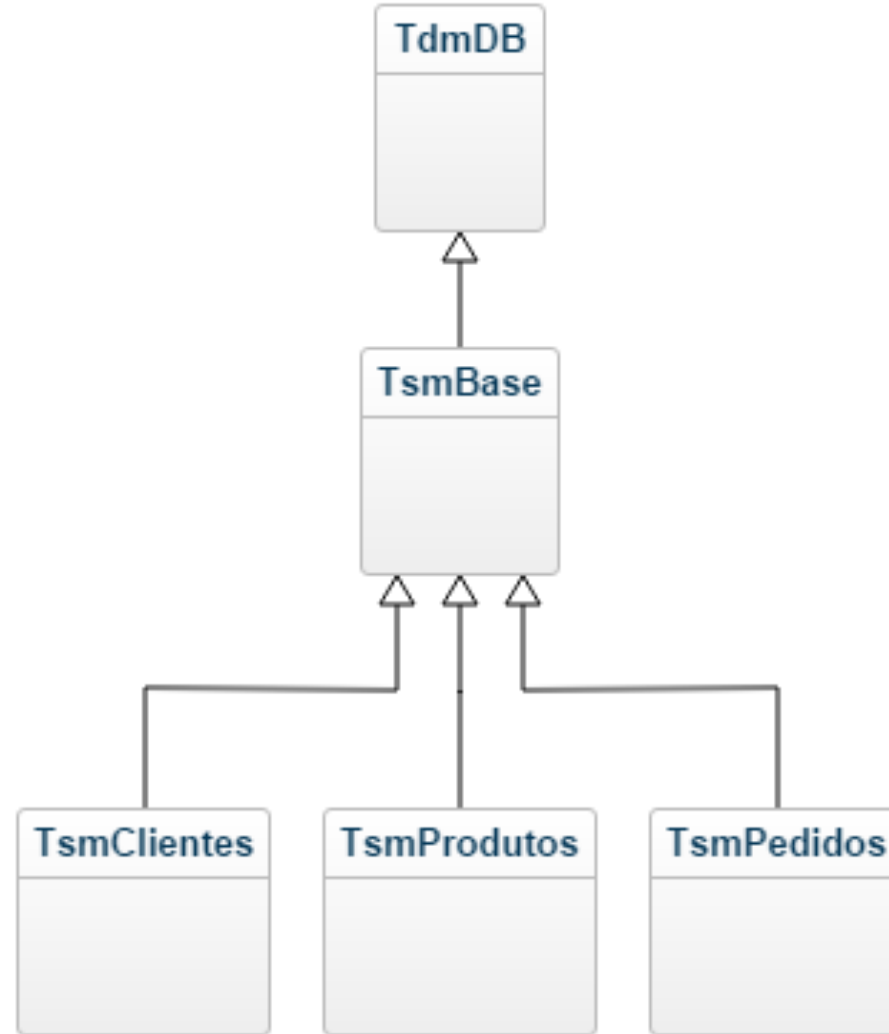
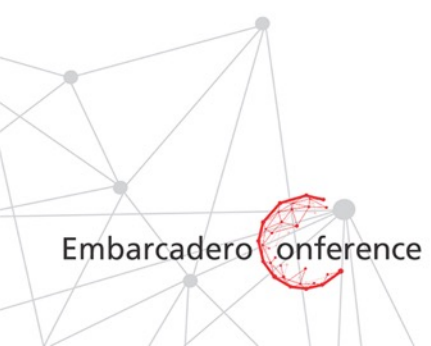


# [Melhores práticas com Datasnap]

- Organização/Estrutura
- Segurança
- Performance

# [Organização/Estrutura]

- Avalie bem a classe base a usar nas suas classes
  - TComponent, TDataModule e TServerModule
- Minimize ao máximo a quantidade de componentes
  - Lembre-se que cada conexão vai criar (e/ou manter) uma cópia do seu módulo em memória
- Separe suas classes por módulos do sistema
  - Mas cuidado com as dependências!
- Utilize Herança



# [Segurança]

- Autenticação
- Autorização

## [Segurança - Autenticação]

- Decide se o servidor aceita ou não o acesso daquele cliente
- Quando aceita, o servidor devolve um “*token*” (ID da sessão) para o cliente, para que este o envie em suas requisições futuras, evitando que o processo de autenticação seja feito novamente.



# [Segurança - Autorização]

- Decide se o servidor aceita ou não a execução de uma requisição
- Decisão baseada no usuário que requisitou, na classe ou no método requisitados.

## [Segurança - Autenticação e Autorização]

- Ambos são gerenciados pelo componente **TDSAuthenticationManager**, respectivamente nos eventos **onUserAuthenticate** e **onUserAuthorize**.

## [Segurança - Autorizando além do onUserAuthorize]

- Atributo TRoleAuth nas classes e métodos
  - Permite especificar qual role pode ou não acessar aquele item
  - Menos flexível, porém fácil configuração
- Coleção Roles do componente TDSAuthenticationManager
  - Útil com DataSetProviders
  - Melhor com muitos perfis (*roles*) ou perfis dinâmicos



# [Performance]

- Quanto menos coisas (objetos e conexões) o servidor precisar manter “vivo”, melhor!
- Muita atenção ao Ciclo de Vida das suas classes
  - Session, Server, Invocation
- Cache: use e abuse!
  - Server side, client side, built-in ou third-party
- Considere o uso de um pool de conexão com seu BD
- TCP x HTTP?
- Cuidado com
  - Autenticação e autorização
  - Log/Monitoramento

[Demo]

Embarcadero  onference

# [Datsnap - onde consigo mais informações?]

- [Documentação - Developing DataSnap Applications](#)
  - [http://docwiki.embarcadero.com/RADStudio/Berlin/en/Developing\\_DataSnap\\_Applications](http://docwiki.embarcadero.com/RADStudio/Berlin/en/Developing_DataSnap_Applications)
- [Secure DataSnap Development \(CodeRage 8 Replay\)](#)
  - <http://www.youtube.com/watch?v=X1CW9A1bCzk>
- [DataSnap REST Messaging Protocol - Session Parameter Caching](#)
  - [http://docwiki.embarcadero.com/RADStudio/Seattle/en/DataSnap\\_REST\\_Messaging\\_Protocol#Session\\_Parameter\\_Caching](http://docwiki.embarcadero.com/RADStudio/Seattle/en/DataSnap_REST_Messaging_Protocol#Session_Parameter_Caching)
- Cache com Redis
  - <http://redis.io/> e <https://github.com/danieleteti/delphiredisclient>
- [FireDAC Connection Pooling in Multithreading Applications](#)
  - [http://docwiki.embarcadero.com/RADStudio/Berlin/en/Multithreading\\_\(FireDAC\)#Connection\\_Pooling](http://docwiki.embarcadero.com/RADStudio/Berlin/en/Multithreading_(FireDAC)#Connection_Pooling)





[OBRIGADO]

@ kmerlotti@gmail.com

@kmerlotti

kmerlotti

Embarcadero conference







# Embarcadero Conference