



# DataSnap REST

Fernando Rizzato  
Lead Software Consultant, *Latin America*

# AGENDA

- Arquitectura multicapa
- DataSnap
  - Conceptos Generales
  - Modelos Soportados
- Estándar REST/JSON
- Mapeo de URI (*Uniform Resource Identifier*)
- Demos

# ARQUITECTURA MULTICAPA

## Presentation tier

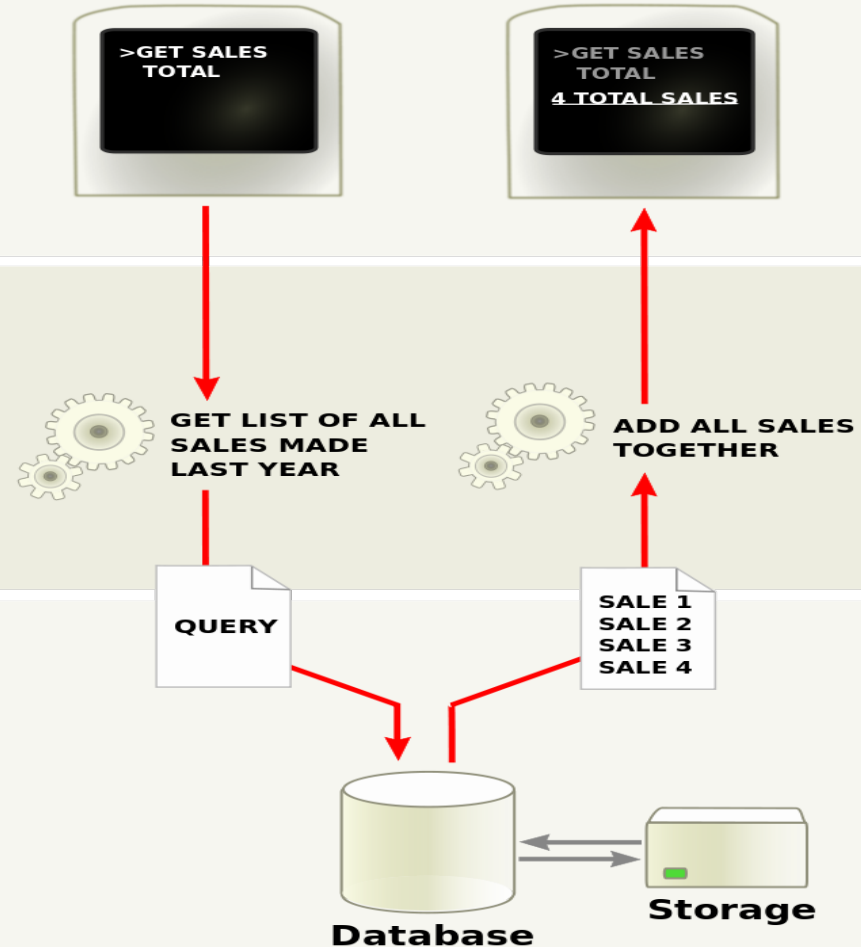
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

## Logic tier

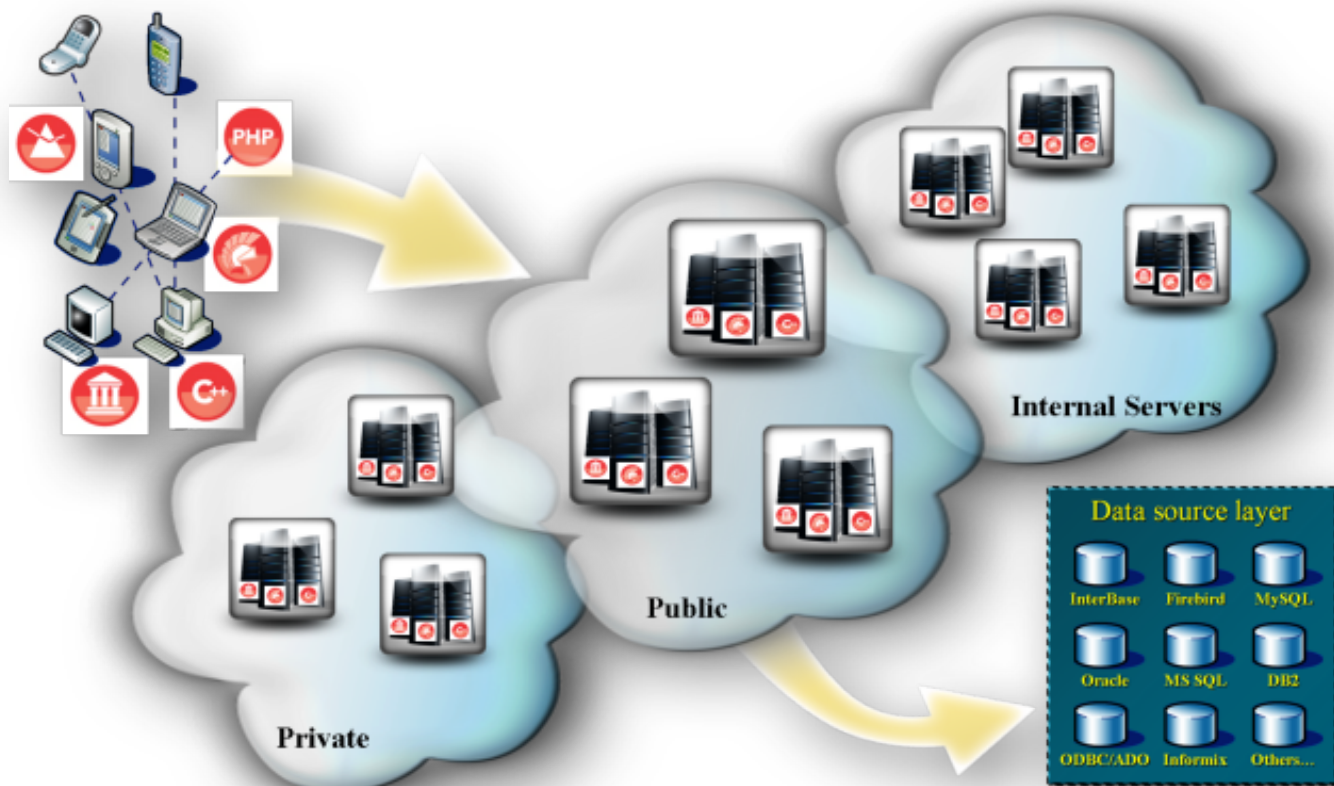
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

## Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



# DATASNAP



# CONCEPTOS GENERALES

- Inicialmente conocido como MIDAS
- Aplicación **Server**
  - Delphi e C++ Builder
- Aplicación **Cliente**
  - Delphi, C++ Builder
  - Cualquier plataforma - de acuerdo con el protocolo elegido
- Diferentes arquitecturas y protocolos a través del tiempo
  - COM/DCOM, TCP/IP (Socket), HTTP/HTTPs, REST/JSON

# MODELOS SOPORTADOS

- MIDAS (IAppServer)
  - COM/DCOM
  - SOCKET (TCP/IP)
- SqlConnection (TSQLConnection)
  - TCP/IP
  - HTTP, HTTPs
- RestConnection (TDSRestConnection)
  - HTTP, HTTPs

# ESTÁNDAR REST/JSON

- Arquitectura (no es un protocolo)
- **R**epresentational **S**tate **T**ransfer
- Consolidado y soportado por todas las plataformas modernas
  - Desktop, Mobile e WEB
- Un protocolo cliente/servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la solicitud
- Un conjunto de operaciones bien definidas
  - POST, GET, PUT e DELETE

# ESTÁNDAR REST/JSON

- Cada recurso (o método) REST tiene su propio identificador
  - **GET** <http://jsonplaceholder.typicode.com/posts/1>
- Notación **JSON** (**J**avaScript **O**bject **N**otation)
  - Más sencillo y ligero en comparación con XML

```
/* will return
{
  id: 1
  title: 'foo',
  body: 'quia et suscipit [...]',
  userId: 1
}
*/
```



# MAPEO DE URI *(UNIFORM RESOURCE IDENTIFIER)*

- El servidor responde a URIs
- La llamada HTTP por lo general tienen el siguiente formato  
*http://my.site.com/datasnap/rest/URIClassName/URIMethodName[/inputParameter]\**

| Command type | Mapping pattern                   | Example               |
|--------------|-----------------------------------|-----------------------|
| GET          | URICclassName.URIMethodName       | Utility.Storage       |
| PUT          | URICclassName.acceptURIMethodName | Utility.acceptStorage |
| POST         | URICclassName.updateURIMethodName | Utility.updateStorage |
| DELETE       | URICclassName.cancelURIMethodName | Utility.cancelStorage |

# MAPEO DE URI *(UNIFORM RESOURCE IDENTIFIER)*

- GET => Utility.Storage(Key: String): TJSONValue;
  - Devuelve un objeto o conjunto de objetos
- POST => Utility.UpdateStorage(Key: String; Data: TJSONValue);
  - Más comúnmente utilizado para actualizar un recurso
  - El contenido de la llamada se asigna a la Data, el postfix URI se asigna a la Key
- PUT => Utility.AcceptStorage(Key: String; Data: TJSONValue);
  - Más comúnmente utilizado para insertar un recurso
  - El contenido de la llamada se asigna al segundo parámetro de entrada (Data)
- DELETE => Utility.CancelStorage(Key: String);
  - Elimina el objeto con la clave dada

DEMOS

# Recursos Adicionales

## ■ Documentación

- [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/DataSnap\\_Overview\\_and\\_Architecture](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/DataSnap_Overview_and_Architecture)
- [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Developing\\_DataSnap\\_Applications](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Developing_DataSnap_Applications)
- [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/REST\\_Overview](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/REST_Overview)
- [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/DataSnap\\_REST](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/DataSnap_REST)
- [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Tutorial:\\_Using\\_a\\_REST\\_DataSnap\\_Server\\_within\\_an\\_Application](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Tutorial:_Using_a_REST_DataSnap_Server_within_an_Application)

## ■ Blogs y Tutoriales

- <https://github.com/ezequieljuliano/DataSetConverter4Delphi>
- <https://www.embarcadero.com/rad-in-action/datasnap-xe>
- <https://www.embarcadero.com/rad-in-action/delphi-labs>
- <https://www.embarcadero.com/rad-in-action/datasnap-rest>

# Recursos Adicionales

## ■ Textos Conceptuales

- <https://spring.io/understanding/REST>
- <https://www.thoughtworks.com/insights/blog/rest-api-design-resource-modeling>
- <http://stackoverflow.com/questions/3105296/if-rest-applications-are-supposed-to-be-stateless-how-do-you-manage-sessions>

# GRACIAS!

## Preguntas?

Me puedes encontrar en:

@FernandoRizzato

fernando.rizzato@embarcadero.com

Síguenos en

*fb.com/EMBTLatAm*