

# Arquitectura REST

Grupo de Construcción de Sw

# Referencias

- <http://www.infoq.com/resource/minibooks/emag-03-2010-rest/en/pdf/ResteMag.pdf>
- [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)
- <http://www.xfront.com/REST-Web-Services.html>
- <http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api#requirements>

# Qué es REST?

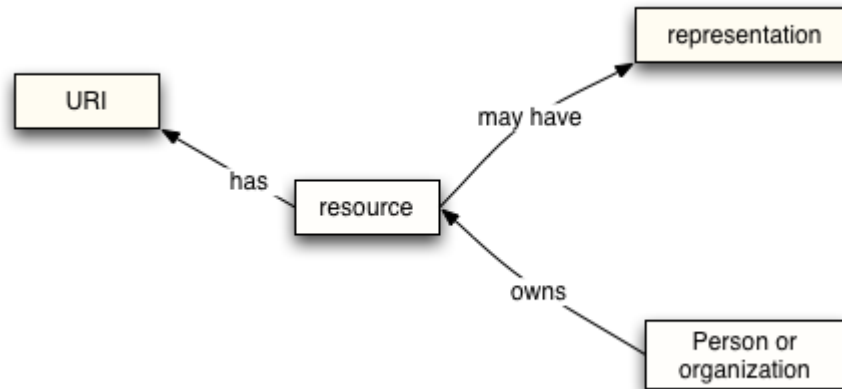
- Roy Fielding acuñó el término REST en su tesis doctoral en el 2000:
  - REST = Representational State Transfer
- REST es un estilo arquitectural
- *HTTP Rest* es una forma de implementarlo

# Estilo arquitectural

- Un estilo arquitectural es un conjunto coordinado de restricciones arquitectónicas que restringen los roles / funciones de los **elementos arquitectónicos y las relaciones** permitidas entre esos elementos dentro de cualquier arquitectura que se ajuste con ese estilo.

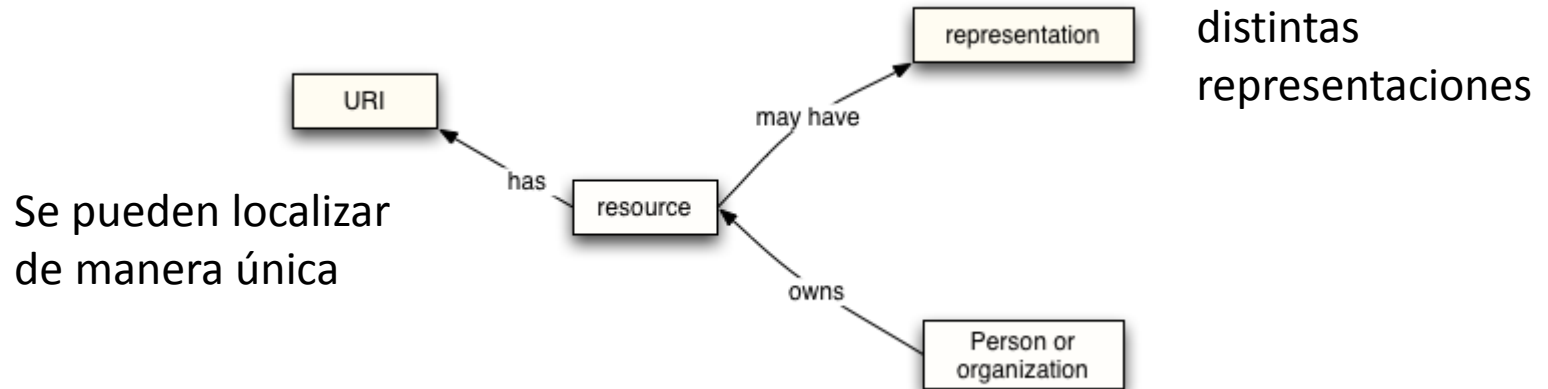
# Elementos

- Lo más importante: recursos



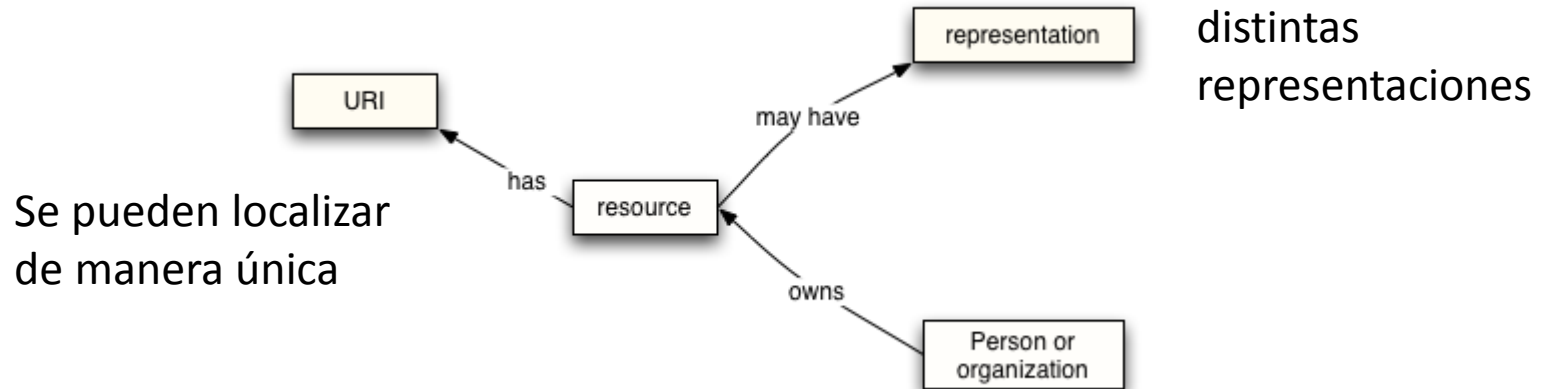
# Elementos

- Lo más importante: recursos



# Elementos

- Lo más importante: recursos



<http://example.com/customers/1234>

<http://example.com/orders/2007/10/776654>

<http://example.com/products/4554>

# Acciones (u operaciones)

- Transferir una representación del recurso de un lado para otro en la red
- Crear, Retornar, Modificar, Suprimir un recurso
- Las operaciones son sin estado (no se sabe qué pasó antes)



# En el protocolo HTTP:

- Protocolo de red sincrónico de request/response usado por aplicaciones en la WEB.
- Un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores.

# En el protocolo HTTP:

- Define los verbos que indican la acción que desea que se efectúe sobre el recurso identificado.
  - HTTP POST envía y **crea** el recurso en el servidor
  - HTTP GET **retorna** un recurso en una representación
  - HTTP PUT le envía al servidor representaciones de recursos para **modificarlo**
  - HTTP DELETE **suprime** un recurso identificado con una URI

Resources	POST
Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a>	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.
Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a>	Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.

Tabla tomada de wikipedia

Resources	GET
Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a>	List the URIs and perhaps other details of the collection's members.
Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a>	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.

Tabla tomada de wikipedia

Resources	PUT
Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a>	Replace the entire collection with another collection.
Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a>	Replace the addressed member of the collection, or if it doesn't exist, create it.

Tabla tomada de wikipedia

Resources	DELETE
Collection URI, such as <a href="http://example.com/resources">http://example.com/resources</a>	Delete the entire collection.
Element URI, such as <a href="http://example.com/resources/item17">http://example.com/resources/item17</a>	Delete the addressed member of the collection.

Tabla tomada de wikipedia

# Principios de diseño REST

- Identificar las entidades conceptuales que se quieren exhibir como servicios
- Crear una URL para cada recurso. Los recursos son sustantivos NO VERBOS
  - Ejemplo de una mala definición:
    - <http://www.parts-depot.com/parts/getPart?id=00345>
  - Corrección:
    - <http://www.parts-depot.com/parts/00345>

# Principios de diseño REST

- Categorizar los recursos de acuerdo con:
  - El cliente recibirá una representación del recurso (HTTP GET)
  - El cliente modificará o creará el recurso (HTTP POST, PUT, y/o DELETE).
- Todos los recursos que se accedan vía HTTP GET no deben tener efecto de borde (solo hace traer una representación del recurso) .