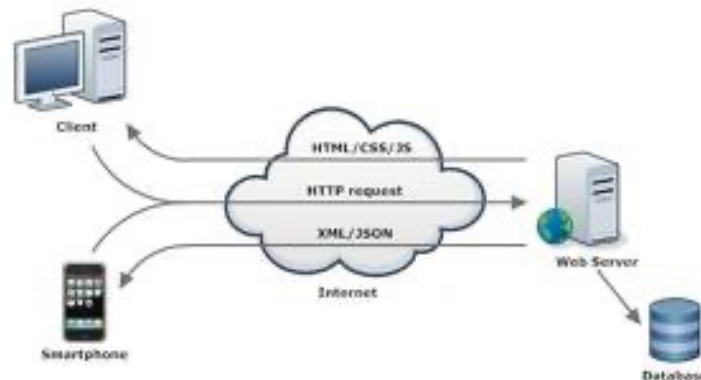


Intro

En los últimos años se ha popularizado un estilo de arquitectura de software conocido como REST (Representational State Transfer).

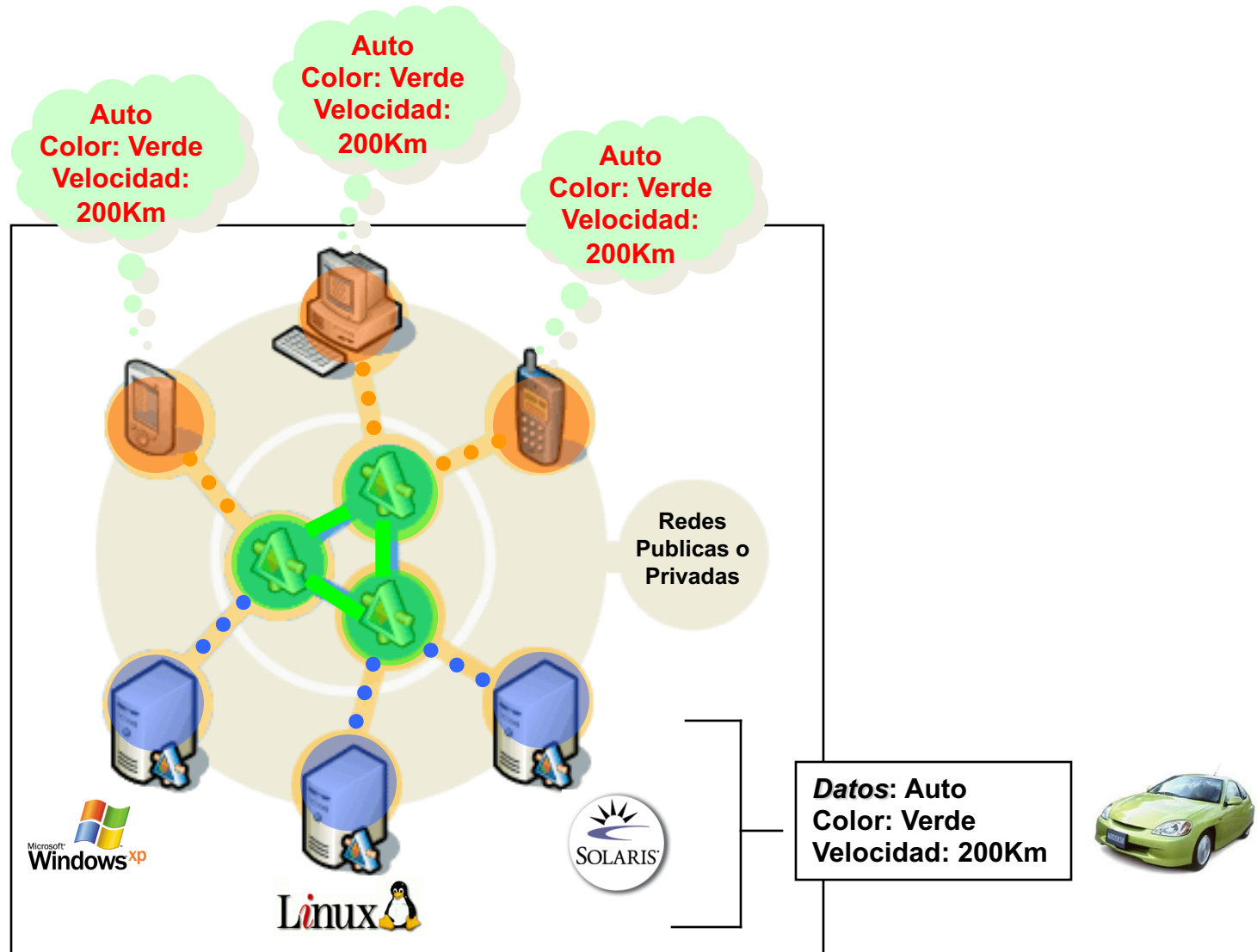
Este nuevo estilo ha supuesto una nueva opción a la hora de desarrollar aplicaciones orientadas a la Web.



¿Qué es REST realmente?

- Se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red.
- Estos principios resumen cómo los recursos son definidos y consumidos.
- El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional.

<https://api.ventadeautos.com.ar/autos/1>



¿Qué es REST realmente?

```
curl http://api.billysbilling.com/v2/invoices \
-u '33yQLVz7YGsDRSufjAnZGnlrjqprAhd2:' \
-d '{
  "contactId":"595-RkHZFe4iuIzyNhkD",
  "currencyId":USD,
  "entryDate":"2012-01-02",
  "lines":[
    {
      "productId":"592-qlEPoJbMc1Uo3c
      "description":"Chameleons",
      "quantity":12,
      "unitPrice":123
    }
  ]
}'
```

¿Cómo sería un ejemplo de diseño basado en REST?

Los métodos HTTP más importantes son POST, GET, PUT y DELETE. Ellos suelen ser comparados con las operaciones asociadas a la tecnología de base de datos, operaciones CRUD: CREATE, READ, UPDATE, DELETE.

Acción	HTTP	SQL
Create	POST	INSERT
Read	GET	SELECT
Update	PUT	UPDATE
Delete	DELETE	DELETE

Ejemplo de un Api REST simple

<http://www.xyz.com/autos> (listado - GET)

<http://www.xyz.com/autos/1> (consulta - GET)

<http://www.xyz.com/autos/1/borrar> (GET)

<http://www.xyz.com/autos/1> (actualizar - POST)

<http://www.xyz.com/autos/nuevo> (POST)

Ejemplo de un Api REST Full

<http://www.xyz.com/autos> (listado - GET)

<http://www.xyz.com/autos/1> (consulta - GET)

<http://www.xyz.com/autos/1> (DELETE)

<http://www.xyz.com/autos/1> (PUT)

<http://www.xyz.com/autos> (POST)

Ejemplo de un Api REST Full

<http://www.xyz.com/autos/1/conductores>
(consulta - GET)

<http://www.xyz.com/autos/1/conductores/45>
(consulta - GET)

Características de REST

- poco acoplados
- encapsulados
- simples
- rápidos
- reutilizables
- sin información de estado (“stateless”)
- autónomos e independientes de una plataforma o lenguaje
- localizables
- solución orientada a la Web 2.0
- basados en estándares simples HTTP (GET – POST – PUT –DELETE)

Pautas a seguir en el diseño de un API REST

- Identificar todas las entidades conceptuales que se desean exponer como servicio.
- Crear una URL para cada recurso. Los recursos deberían ser nombres
- Definir el formato de salida (XML, JSON, YAML, RSS)

¿Quiénes usan REST?

- GOOGLE
- AMAZON
- EBAY
- MERCADOLIBRE
- FACEBOOK
- TWITTER
-

Ejemplo

<https://dev.twitter.com/rest/public>

<https://developers.facebook.com/docs/graph-api/reference/>

<http://developers.mercadolibre.com/api-directory/>

<https://code.google.com/apis/console>

<https://apiary.io/>

Buenas practicas

- HTTPS siempre
- JSON sobre XML
- HTTP status codes para los errores + descripción de errores
- Versionar el API
- Paginar, filtrar, ordenar

Otros temas

Seguridad

Descripción / Documentación

¿Preguntas?