

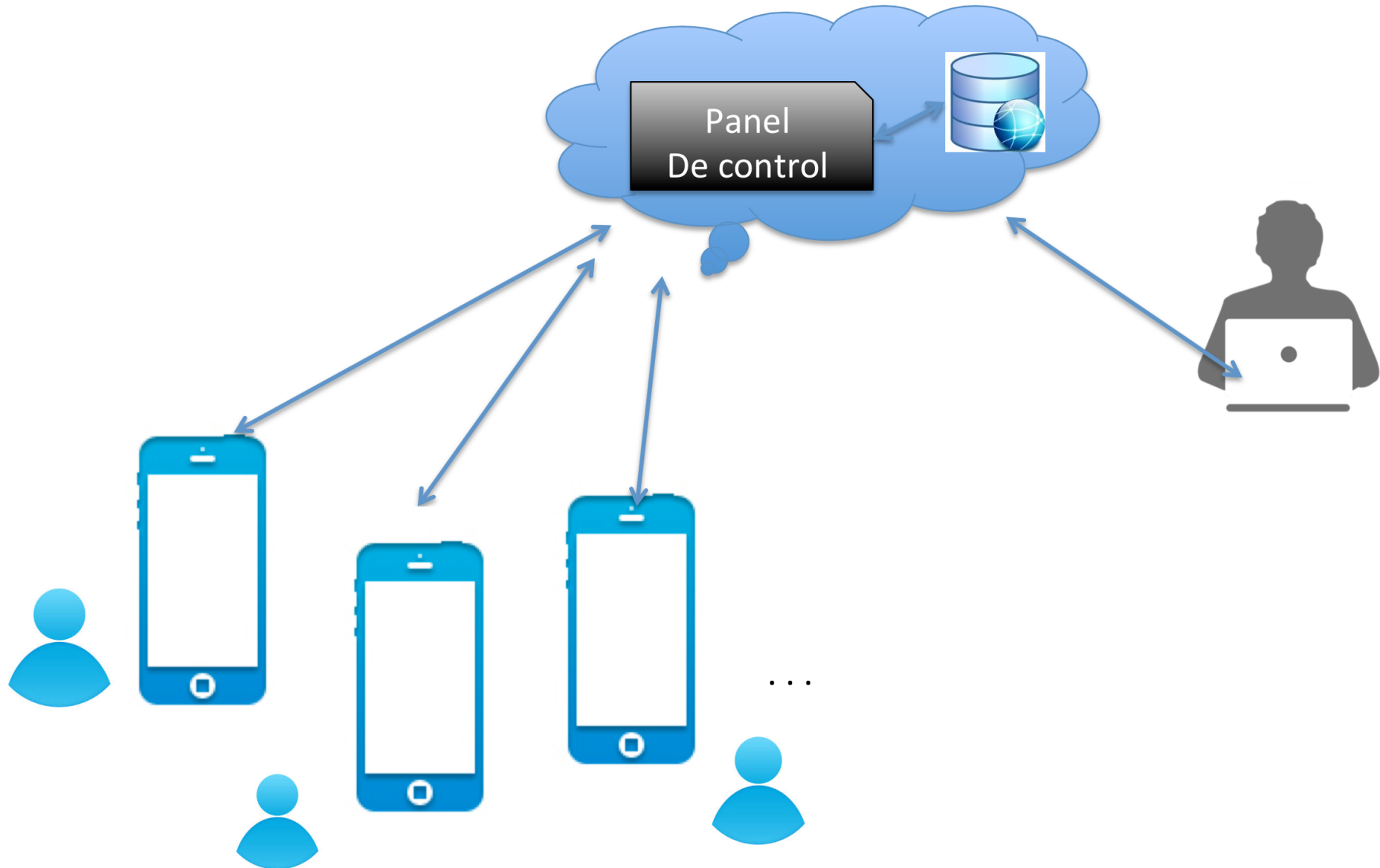


## **SERVICIOS WEB**

# **PARSER Y SERIALIZACIÓN JSON**

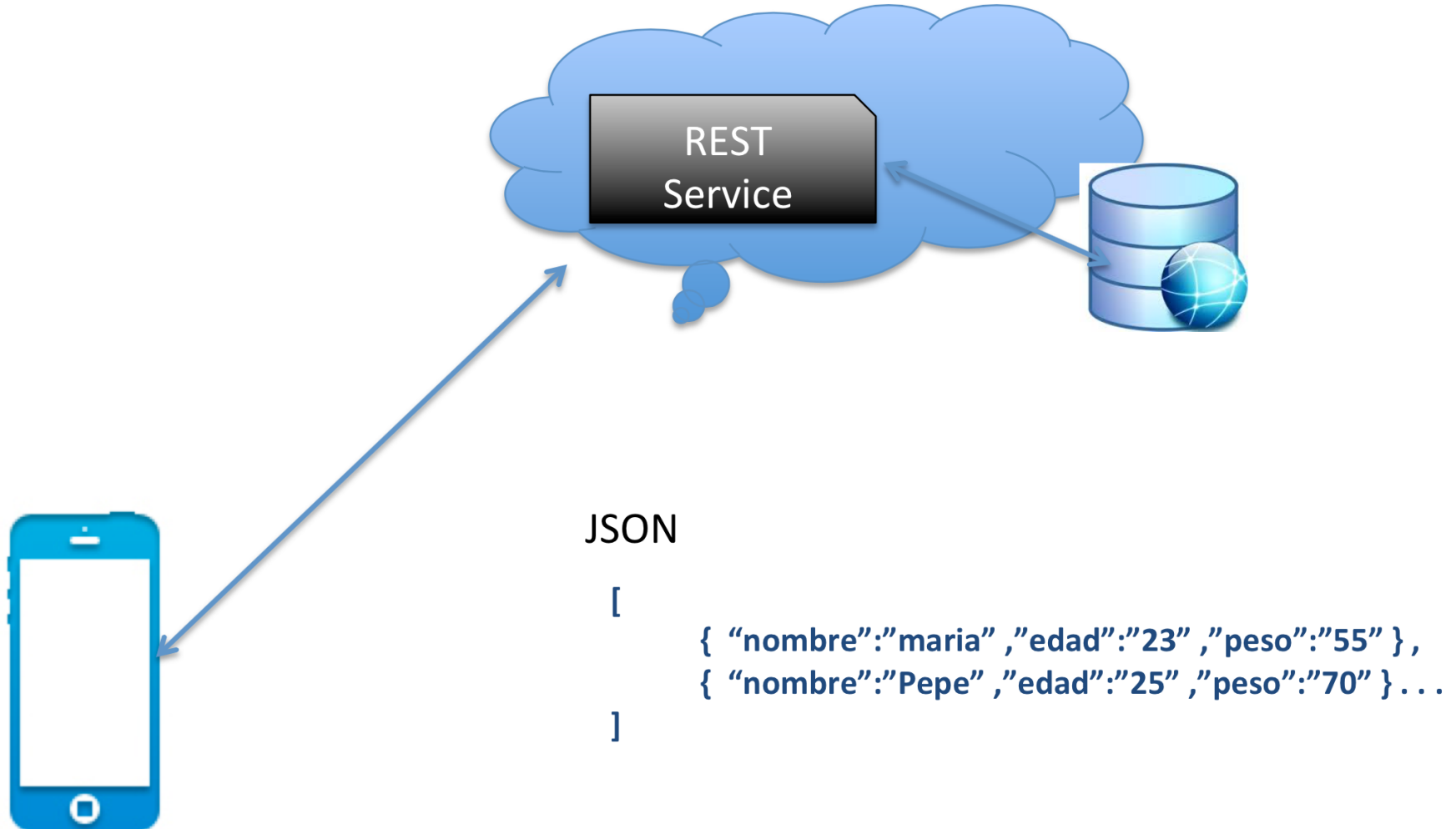


## Arquitectura de información Dinámica, servicio a móvil





## PARSER DE SERIALIZACIÓN JSON y XML





## Servicios Web

El consorcio W3C, define los servicios Web, como sistemas software diseñados para Soportar una interacción interoperable maquina a maquina en una red.

### **SOAP (siglas de Simple Object Access Protocol)**

Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambios de datos XML, el punto identificativo de SOAP es que las operaciones son definidas como puertos WSDL (Web Services Description Language).

### **REST (Representational State Transfer)**

Es un **estilo de arquitectura de software** para sistemas distribuidos tales como la web, a diferencia de SOAP, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. Las operaciones( o funciones) se solicitarán mediante GET, POST, PUT y DELETE. **Además se podrá utilizar JSON en vez de XML** como contenedor de la información, por lo que será ***aconsejable utilizar este protocolo cuando busquemos mejorar el rendimiento, o cuando disponemos de escasos recursos, como sería el caso de los dispositivos móviles.***



## JSON reglas de sintaxis

**<1>** Los objetos **JSON** usan simbolo llaves curl braces) { }, dentro del cual se muestran los datos "Data"

*Ejm: { "nombre":"maria" }*

Estas entidades, pueden ser más pares, entonces estas se separan por comas. Ejm:

*{ "nombre":"maria" ,"edad":"23" ,"peso":"55" }*

**<2>** En caso sean varios **JSON Objects**, es conveniente mostrarlos en un **JSON Array**

El **JSON Array** se muestra con brackets o corchetes []

Ejm:

```
[  { "nombre":"maria" ,"edad":"23" ,"peso":"55" },  
    { "nombre":"Pepe" ,"edad":"25" ,"peso":"70" }  
]
```



## Resumen: JSON reglas de sintaxis

**DATA JSON:** "campo":"valor"

**JSON Object:** {"campo":"valor",..., "campo":"valor" ...}

**JSON Array :**

[

    {"campo":"valor",..., "campo":"valor" ...},  
    {"campo":"valor",..., "campo":"valor" ...},  
    {"campo":"valor",..., "campo":"valor" ...},...

]

**El Archivo debe tener extension .json**

**El MIME Type debe es "application/json"**



## ESTRUCTURA DE JSON

Parent:

```
{"geonames": [
```

```
{  
  "countryName": "Peru",  
  "adminCode1": "LMA",  
  "fclName": "city, village,...",  
  "countryCode": "PE",  
  "lng": -77.0282363891602,  
  "fcodeName": "capital of a political entity",  
  "toponymName": "Lima",  
  "fcl": "P",  
  "name": "Lima",  
  "fcode": "PPLC",  
  "geonameId": 3936456,  
  "lat": -12.0431818993027,  
  "adminName1": "Provincia de Lima",  
  "population": 7737002  
},...
```

```
] }
```





## Ejm Servicio Web Json

<http://www.iphonecursoonline.com/cursoWS/web-service/listado.php>

The screenshot shows a web browser window with the address bar displaying `www.iphonecursoonline.com/cursoWS/web-service/listado.php`. Below the address bar, there is a navigation bar with several folders: `marketing Mo...`, `Más visitados`, `News`, `Noticias`, `Popular`, `Populares`, and `Últimas no`. The main content area of the browser displays a JSON array of three objects, each representing a menu item. The JSON is formatted with line numbers on the left side of the code editor.

```
1[
2  {
3    id: "3",
4    nombre: "Ensalada de Palta",
5    descripcion: "descripcion de la ensalada etc etc",
6    precio: "12.50",
7    archivo: "ensalada.jpg",
8    categoria: "1"
9  },
10 {
11   id: "4",
12   nombre: "Ceviche de Mero",
13   descripcion: "descripcion del ceviche etc etc",
14   precio: "30.30",
15   archivo: "ceviche.jpg",
16   categoria: "1"
17 },
18 {
19   id: "5",
20   nombre: "Anticuchos",
21   descripcion: "descripcion anticuchos etc etc",
22   precio: "15.25",
23   archivo: "anticuchos.jpg",
24   categoria: "1"
25 }
26]
```





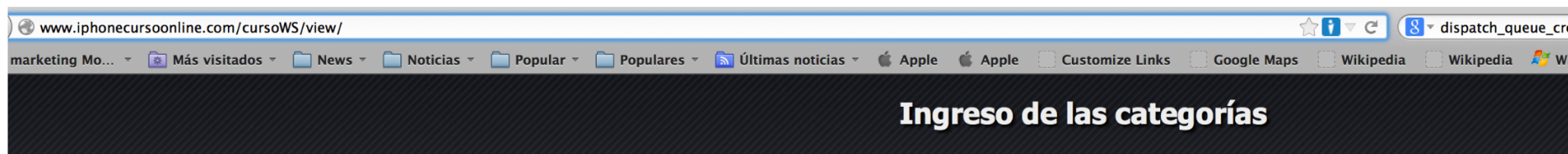
academiaMóviles.com

CURSO

Desarrollo  
de aplicaciones móviles

# Ejm backEnd

<http://www.iphonecursoonline.com/cursoWS/view/>



## Ingreso de las categorías

NOMBRE:

Plato nuevo Peru

DESCRIPCIÓN:

etc

PRECIO:

11

ARCHIVO:

ceviche.jpg

CATEGORIA:

1

REGISTRAR



## Pasos para PARSEAR de Json desde un Url

```
//1- parser Json serializar a objeto iOS
//se requiere link http://www.iphonecursoonline.com/cursoWS/web-service/listado.php

//2-preparando parametros
let linkDatos = "http://www.iphonecursoonline.com/cursoWS/web-service/listado.php"
let url = NSURL(string: linkDatos)
let datosUrl = NSData(contentsOfURL: url!)

//3-creando funcion Parser

func decodeJSON(data: NSData) -> [AnyObject]? {
    return NSJSONSerialization.JSONObjectWithData(data,
        options: .allZeros, error: nil) as? Array
}

//4-llamando ala funcion decode
var arrayDatos = decodeJSON(datosUrl!) as Array!
```

**arrayDatos** contiene los objetos NSDictionary que podrán ser utilizados en iOS



Ejemplo