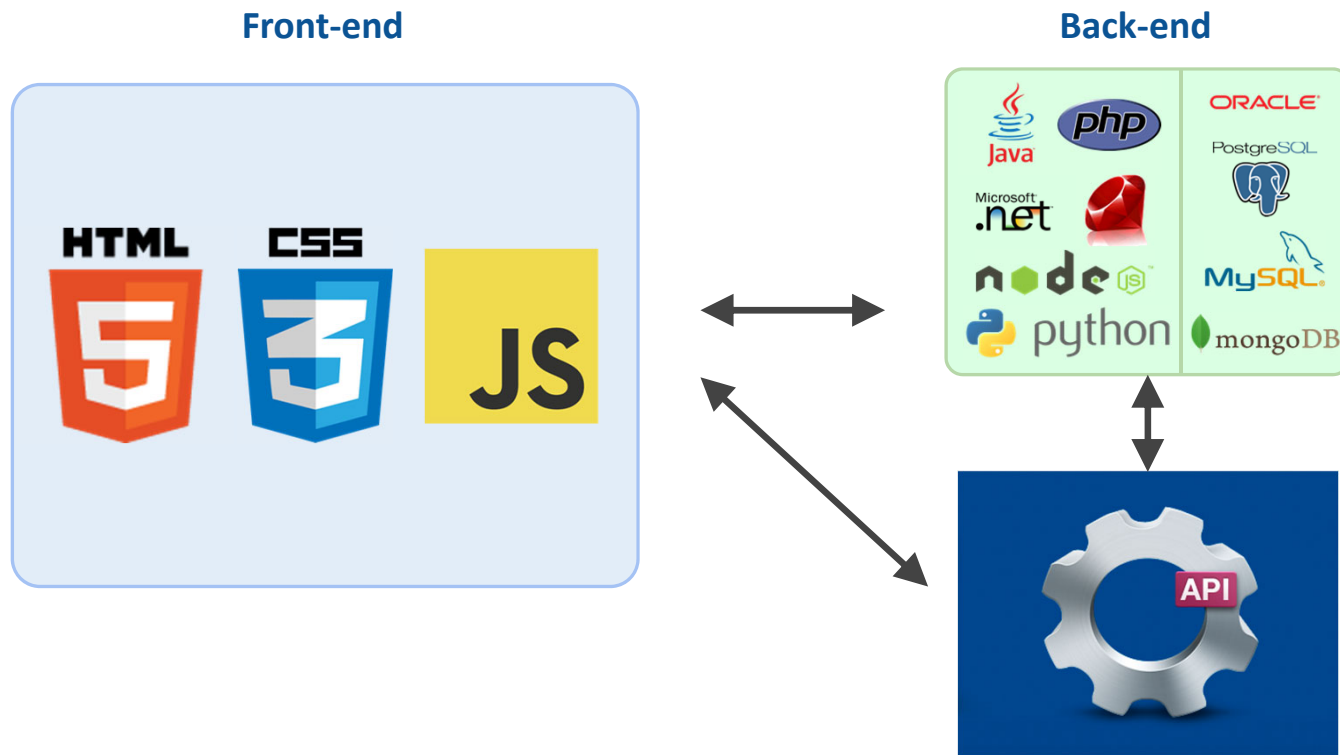


# 3 WEB API

API, JSON, XML

# Herramientas



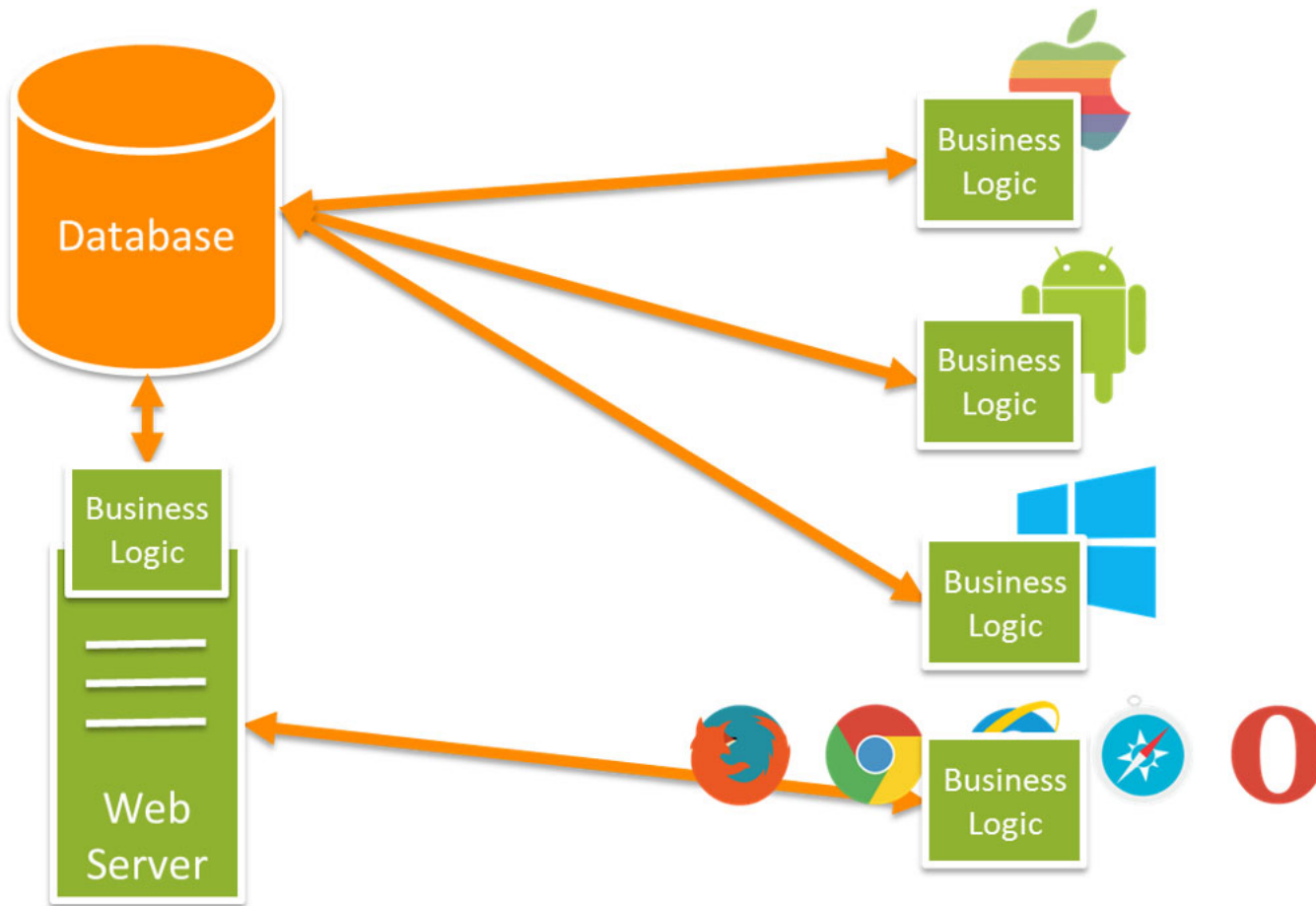
WEB API

**A**PPPLICATION

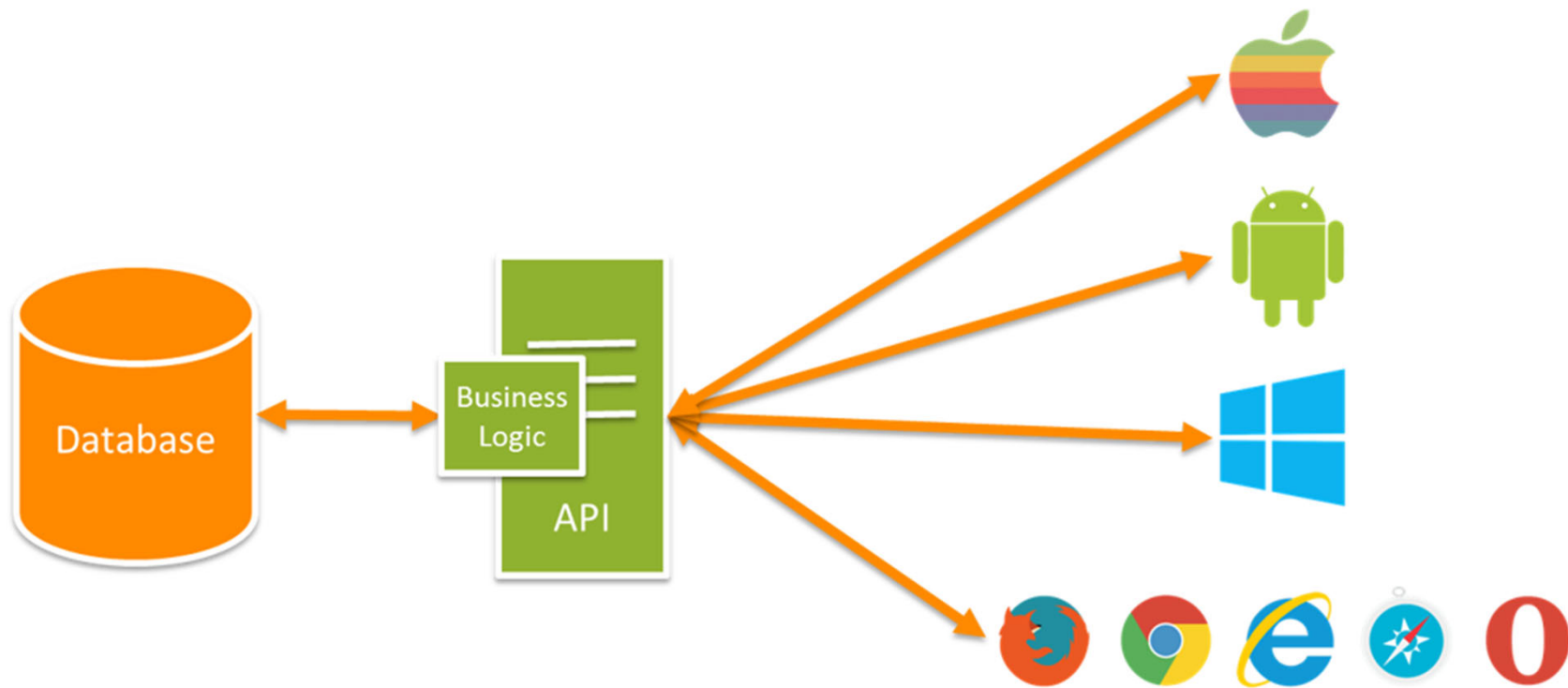
**P**ROGRAMMING

**I**NTERFACE

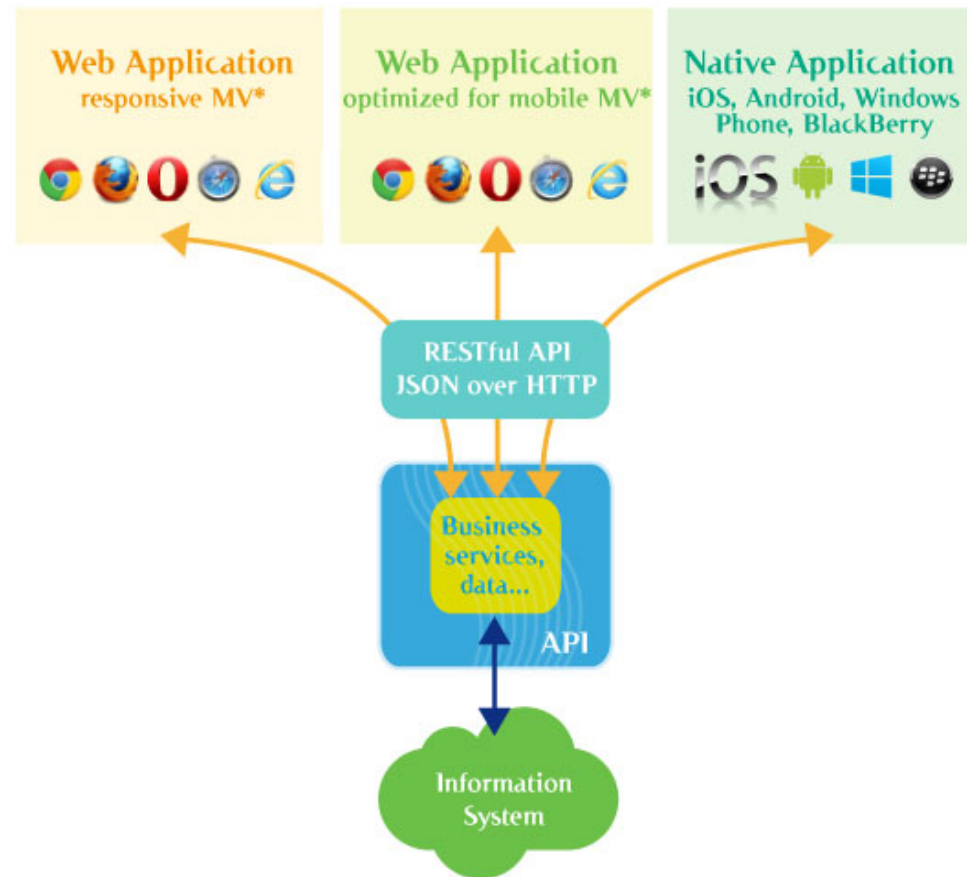
# Sin API



## Con API



# WEB API Architecture



# WEB API

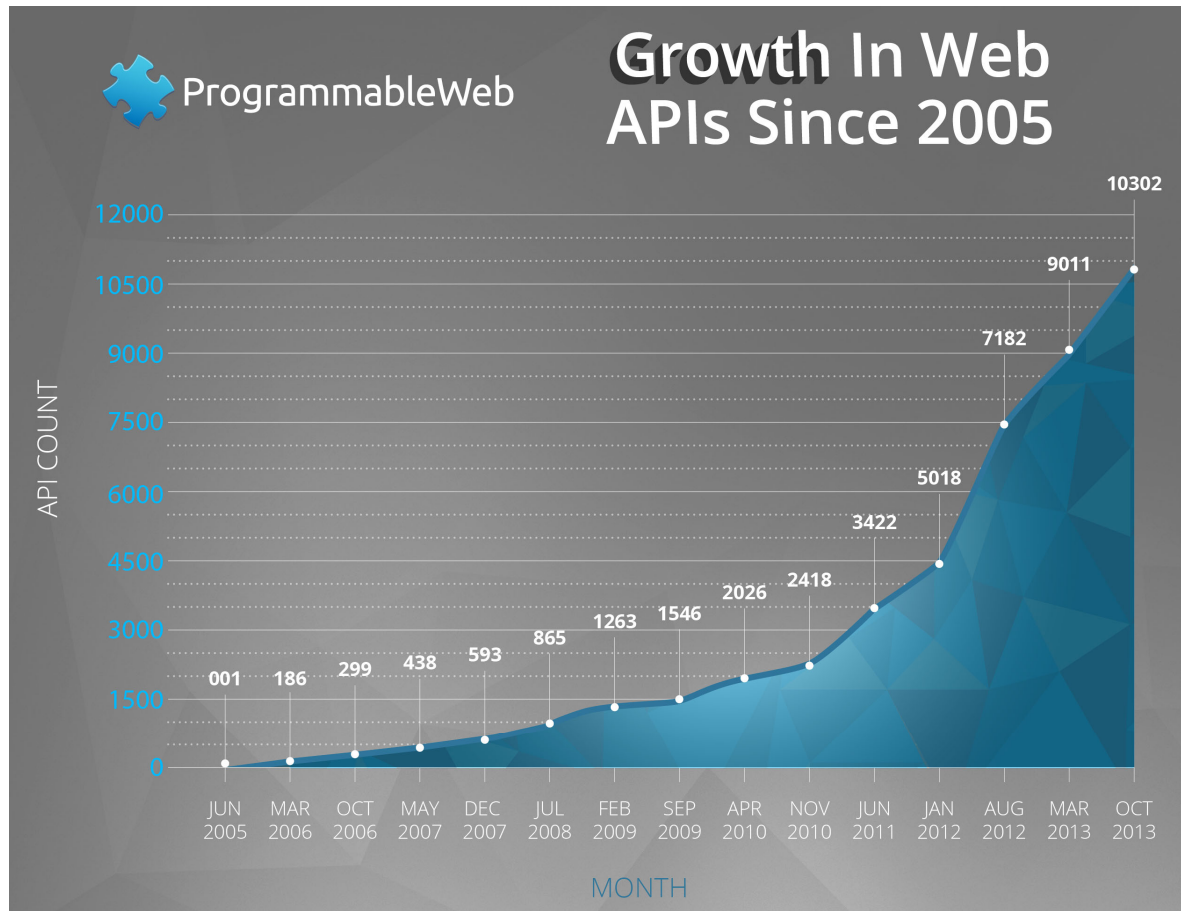
- Conectividad por medio de HTTP requests
- Lenguajes de intercambio de datos como XML y JSON
- Estructura común en el mundo web 2.0 es REST (Representational State Transfer)
- Las que usan REST son RESTful APIs

# Web API - Usos

- Extender el contenido de un sitio web
- Procesamiento de datos
- Interconexión entre sitios/servicios web
- Construir un sitio completamente con datos de diferentes sitios (Mashups)



# Web API - Usos



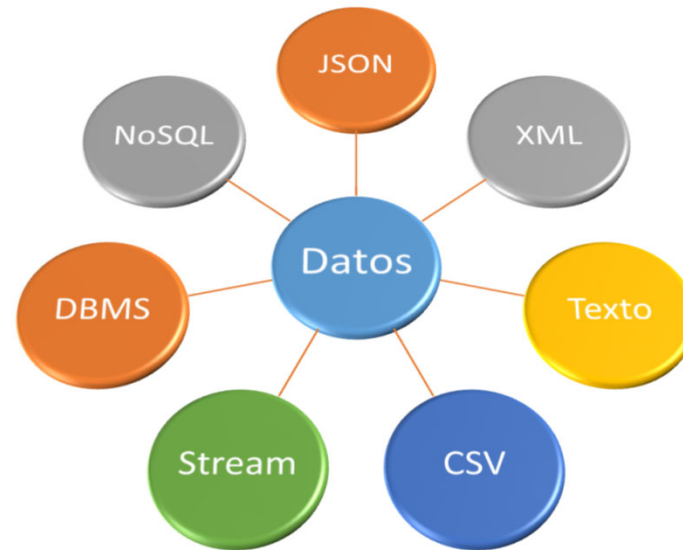
# Web API - Usos

API Name	Description	Category	Submitted
<a href="#">Google Maps</a>	The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface. It is designed to work on both mobile...	<a href="#">Mapping</a>	12.05.2005
<a href="#">Twitter</a>	The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user...	<a href="#">Social</a>	12.08.2006
<a href="#">YouTube</a>	The Data API allows users to integrate their program with YouTube and allow it to perform many of the operations available on the website. It provides the capability to search for videos, retrieve...	<a href="#">Video</a>	02.08.2006

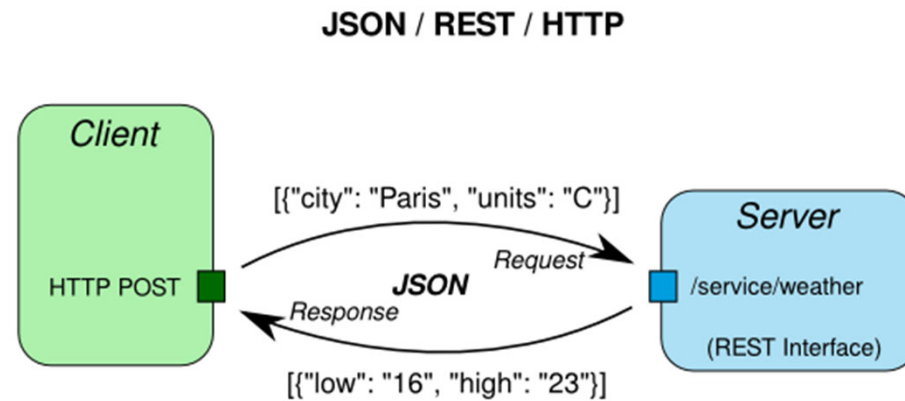
# Web API - Usos

<a href="#">Flickr</a>	The Flickr API can be used to retrieve photos from the Flickr photo sharing service using a variety of feeds - public photos and videos, favorites, friends, group pools, discussions, and more. The...	<a href="#">Photos</a>	09.04.2005
<a href="#">Facebook</a>	The Facebook API is a platform for building applications that are available to the members of the social network of Facebook. The API allows applications to use the social connections and profile...	<a href="#">Social</a>	08.16.2006
<a href="#">Amazon Product Advertising</a>	What was formerly the ECS - eCommerce Service - has been renamed the Product Advertising API. Through this API developers can retrieve product information. The API exposes Amazon's product data...	<a href="#">eCommerce</a>	12.02.2005
<a href="#">Twilio</a>	Twilio provides a simple hosted API and markup language for businesses to quickly build scalable, reliable and advanced voice and SMS communications applications. Twilio provides a telephony...	<a href="#">Telephony</a>	01.09.2009

# Estructura de Datos



# Estructura de Datos



# ¿Qué es JSON?

## JavaScript Object Notation

Es un **estándar abierto** basado en texto para representar e intercambiar datos legibles por humanos.

Es derivado de la programación JavaScript para representar **estructuras de datos**.

# Características

- Independiente de un lenguaje específico
- Basado en texto
- De Formato ligero
- Fácil de parsear
- NO Define funciones
- NO tiene estructuras invisibles
- NO tiene espacios de nombres (Namespaces)
- NO tiene validador
- NO es extensible

# JSON – Valores

- Texto
  - Una cadena de caracteres con ""
- Numérico
  - Números entero o real sin comillas
- Lógico
  - True, False
- Objeto
- Arreglos
- Vacío
  - Null

{

```
"titulo": "SOA",  
"autor": "Emilio",  
"paginas": 125,  
  "stock" : true,  
  "formatos" : ["ebook", "fisico"]
```

}



## JSON – Objeto / Clase

- Es un conjunto de propiedades con su valor.
- Empieza con “{” y termina con “}”
- Las propiedades se separan con “,”
- El nombre y valor se separan con “:”

```
{  
  "titulo": "SOA",  
  "autor": "Emilio",  
  "paginas": 125  
}
```

## JSON – Arreglo

- Colección ordenada de valores u objetos
- Empieza con “[” y termina con “]”
- Los valores se separan con “,”

```
[  
  {  
    "titulo": "SOA",  
    "autor": "Virginia Fox",  
    "paginas": 125  
  },  
  {  
    "titulo": "Cloud",  
    "autor": "Ana Porter",  
    "paginas": 125  
  }  
]
```



## Ejercicio

- Crear un documento JSON para representar un registro de notas de un estudiante. El estudiante tiene matricula, nombre, apellido. Además de materias (1 o más). Cada materia tiene nombre, créditos y calificación.

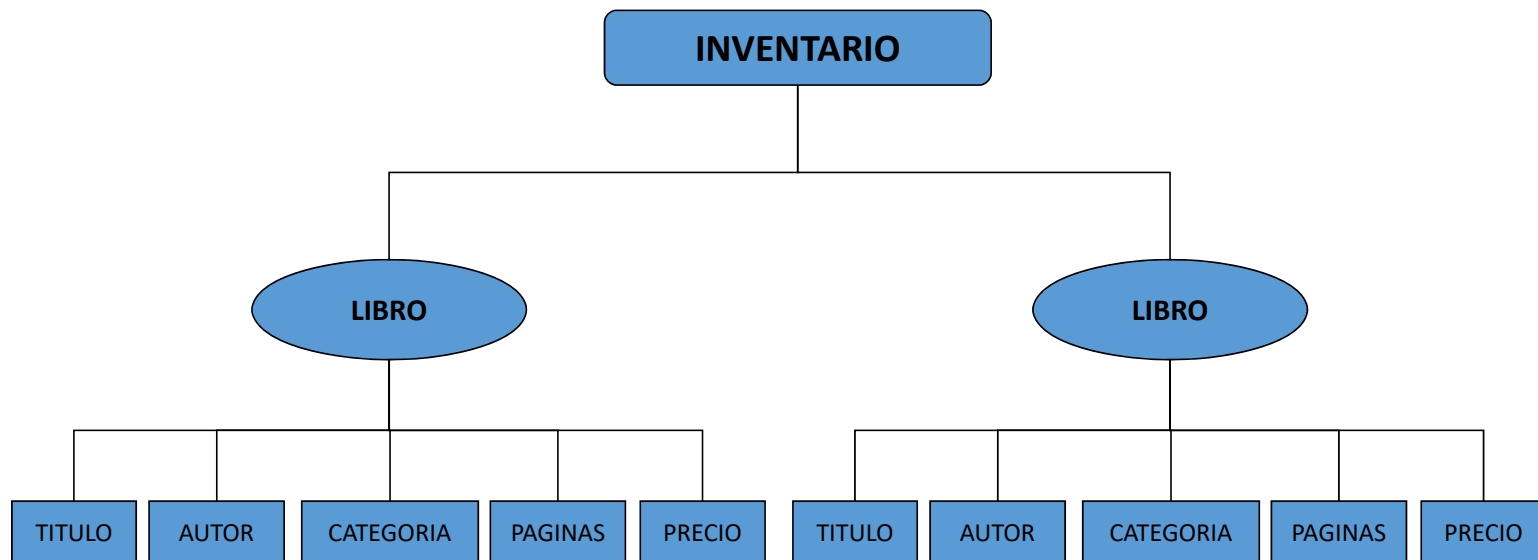
# ¿Qué es XML?

En 1998 se alcanzo la especificación XML 1.0

- XML es un lenguaje de marcado extensible, estructurado, validable que permite transmitir información en la WEB de una forma organizada.
- e**X**tensible **M**arkup **L**anguage

# XML es estructurado

- Es estructurado
  - Permite una jerarquía de tipo árbol con elementos anidados



# Sintaxis de un documento XML

- Extensible
  - Podemos crear nuestras propias etiquetas

```
<?xml version="1.0"?>
```

```
<INVENTARIO>
```

```
<LIBRO>
```

```
<TITULO>SQL SERVER 2005</TITULO>
```

```
<AUTOR>JUAN CARLOS GARCIA</AUTOR>
```

```
<CATEGORIA>BASE DE DATOS</CATEGORIA>
```

```
<PAGINAS>300</PAGINAS>
```

```
<PRECIO>$25</PRECIO>
```

```
</LIBRO>
```

```
<LIBRO>
```

```
<TITULO>MARKETING INTERNACIONAL</TITULO>
```

```
<AUTOR>KOTTER</AUTOR>
```

```
<CATEGORIA>ADMINISTRACION</CATEGORIA>
```

```
<PAGINAS>500</PAGINAS>
```

```
<PRECIO>$38</PRECIO>
```

```
</LIBRO>
```

```
</INVENTARIO>
```

# Aplicaciones de XML

- Además de las aplicaciones XML para describir clases específicas, se han definido diversas aplicaciones
- XSL, Extensible Stylesheet Language
  - Nos permite crear potentes hojas de estilo de documentos utilizando sintaxis XML
- Almacenamiento de Base de Datos
- Descripción de presentaciones multimedia
- Comunicación de aplicaciones en la WEB
- Intercambio de información financiera
- Intercambio de scripts de voz
- Intercambio de información legal

# JSON vs XML

```
<empinfo>
  <employees>
    <employee>
      <name>Scott Philip</name>
      <salary>£44k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Tim Henn</name>
      <salary>£40k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Long yong</name>
      <salary>£40k</salary>
      <age>28</age>
    </employee>
  </employees>
</empinfo>
```

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "Scott Philip",
        "salary" : £44k,
        "age" : 27,
      },
      {
        "name" : "Tim Henn",
        "salary" : £40k,
        "age" : 27,
      },
      {
        "name" : "Long Yong",
        "salary" : £40k,
        "age" : 28,
      }
    ]
  }
}
```



# Obtener JSON desde el servidor

- jQuery permite invocar al servidor y retorna el contenido en formato JSON
- Se realiza mediante una llamada AJAX

```
1 $.ajax({  
2   dataType: "json",  
3   url: url,  
4   data: data,  
5   success: success  
6 });
```

```
1 $.getJSON( "ajax/test.json", function( data ) {  
2   var items = [];  
3   $.each( data, function( key, val ) {  
4     items.push( "<li id='" + key + "'>" + val + "</li>" );  
5   });  
6  
7   $( "<ul/>", {  
8     "class": "my-new-list",  
9     html: items.join( " " )  
10  }).appendTo( "body" );  
11 });
```

## Taller JSON vs XML

- Con JQUERY puede tomar las fotos públicas de la API de Flickr. Usando el browser consulte las dos direcciones:
  - JSON
    - [https://api.flickr.com/services/feeds/photos\\_public.gne?format=json](https://api.flickr.com/services/feeds/photos_public.gne?format=json)
  - XML
    - [https://api.flickr.com/services/feeds/photos\\_public.gne](https://api.flickr.com/services/feeds/photos_public.gne)
  - Determinar el tamaño en bytes de la respuesta de cada tipo de llamada.
- Crear una página HTML con JS y JQUERY que muestre las fotos. Use el parámetro tags para traer fotos relacionadas a monumentos.
  - [http://api.flickr.com/services/feeds/photos\\_public.gne?jsoncallback=?](http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?)

# Taller JSON vs XML

- ```
(function() {  
    var flickerAPI = "http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?";  
  
    $.getJSON( flickerAPI, {  
        tags: "",  
        tagmode: "any",  
        format: "json"  
    })  
    .done(function( data ) {  
        $.each( data.items, function( i, item ) {  
            $( "<img>" ).attr( "src", item.media.m ).appendTo( "#images" );  
        });  
    });  
})();
```

```
img {  
    height: 100px;  
    float: left;  
}
```

● Front-End

# Mejorando estilo programación en JS

- Handlebars (<https://handlebarsjs.com/>)
- Proporciona la potencia necesaria para permitirle crear **plantillas semánticas** de manera efectiva.
- <http://builds.handlebarsjs.com.s3.amazonaws.com/bucket-listing.html?sort=lastmod&sortdir=desc>



- Front-End

# Mejorando estilo programación en JS




```
<script type="text/x-handlebars-template" id="template">
<ul>
  {{# each beers}}
    <li>Name: {{name}}</li>
    <li>Style: {{style}}</li>
  {{/each}}
</ul>
</script>
```

```
var data = {};
data.beers = [{"name": "Estrella", "style": "Euro Lager"},
{"name": "Voll Damm", "style": "Bock"},
{"name": "Devil's", "style": "Indian Pale Ale"}];
var beerTemplate = Handlebars.compile($("#template").html());
var html = beerTemplate(data);
$("#content").append(html);
```



## Taller JSON vs XML Versión 2

```
.done(function( data ) {  
  $.each( data.items, function( i, item ) {  
    $( "<img>" ).attr( "src", item.media.m ).appendTo( "#images" );  
  });  
});
```



Reemplazar con Handlebars, y usar mas atributos del JSON, como descripción de la imagen, autor, fecha, categoría, etc.

- Mejor estilo de programación

```
$(function() {  
    var flickerAPI = "http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?";  
  
    $.getJSON( flickerAPI, {  
        tags: "",  
        tagmode: "any",  
        format: "json"  
    })  
    .done(function( data ) {  
        var source = $("#flirk-template").html();  
        var template = Handlebars.compile(source);  
        var result = template(data);  
        $('#images').html(result);  
    });  
});
```

```
<script id="flirk-template" type="text/x-handlebars-template">  
  
</script>
```

## • JSON vs XML

| Características     | XML                                                                                                                                   | JSON                                                                                                      |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Tipos de Datos      | No define tipo de datos. Necesita de XML Schema.                                                                                      | Provee tipos de datos escalares y la habilidad de estructurar información a través de arreglos y objetos. |
| Soporte de Arreglos | Puede soportar arreglos.                                                                                                              | Soporte nativo de arreglos.                                                                               |
| Soporte de Objetos  | Puede representar objetos. Se necesitan convenciones.                                                                                 | Soporte nativo de objetos.                                                                                |
| Soporte Nulo        | Requiere de <b>xsi:nil</b> .                                                                                                          | Reconoce nativamente valores <b>null</b> .                                                                |
| Comentarios         | Soporte nativo                                                                                                                        | No es soportado.                                                                                          |
| Namespaces          | Soporta namespace el cual elimina el riesgo de colisiones.                                                                            | No tiene el concepto de namespace.                                                                        |
| Mapeo               | Complejo.                                                                                                                             | Simple.                                                                                                   |
| Tamaño              | Tienden a ser muy grandes                                                                                                             | Remueve los literales ahorrando espacio.                                                                  |
| Soporte JavaScript  | Requiere XML DOM.                                                                                                                     | No se requiere de una aplicación adicional. Se usa una simple función <b>eval</b> .                       |
| Curva aprendizaje   | Requiere el uso de: <a href="#">XPath</a> , XML Schema, <a href="#">XSLT</a> , <a href="#">XML Namespaces</a> , <a href="#">DOM</a> . | Tecnología simple.                                                                                        |



## Prototipo Aplicación WEB.

# Proyecto

- Máximo grupo de 4 personas.
- Definir diagrama de arquitectura WEB basado en el que revisamos la primera clase. Aún cuando no implemente todos los componentes debe indicar como seria la solución completa
- Uso de mínimo una 1 API, adicional a la de mapas en caso de usarla. Puede crear su propia API o una existente.
- Usar uno o mas marcos de trabajo en cada capa de la arquitectura adicional a jQuery y Bootstrap.
- Usar un KVS
- Usar Persistencia SQL y/o noSQL (Exposición: **Viernes 4 Oct 2019 a las 18:00, y Examen mismo día a las 20:00**)

### CLIENTE



### SERVIDOR WEB

ASP.NET Core



### BASE DE DATOS



PostgreSQL



mongoDB



redis



Microsoft SQL Server



Firebase

ORACLE



amazon  
DynamoDB

