

# **Arquitectura de aplicaciones web empresariales**

Rai Mengoa Padilla

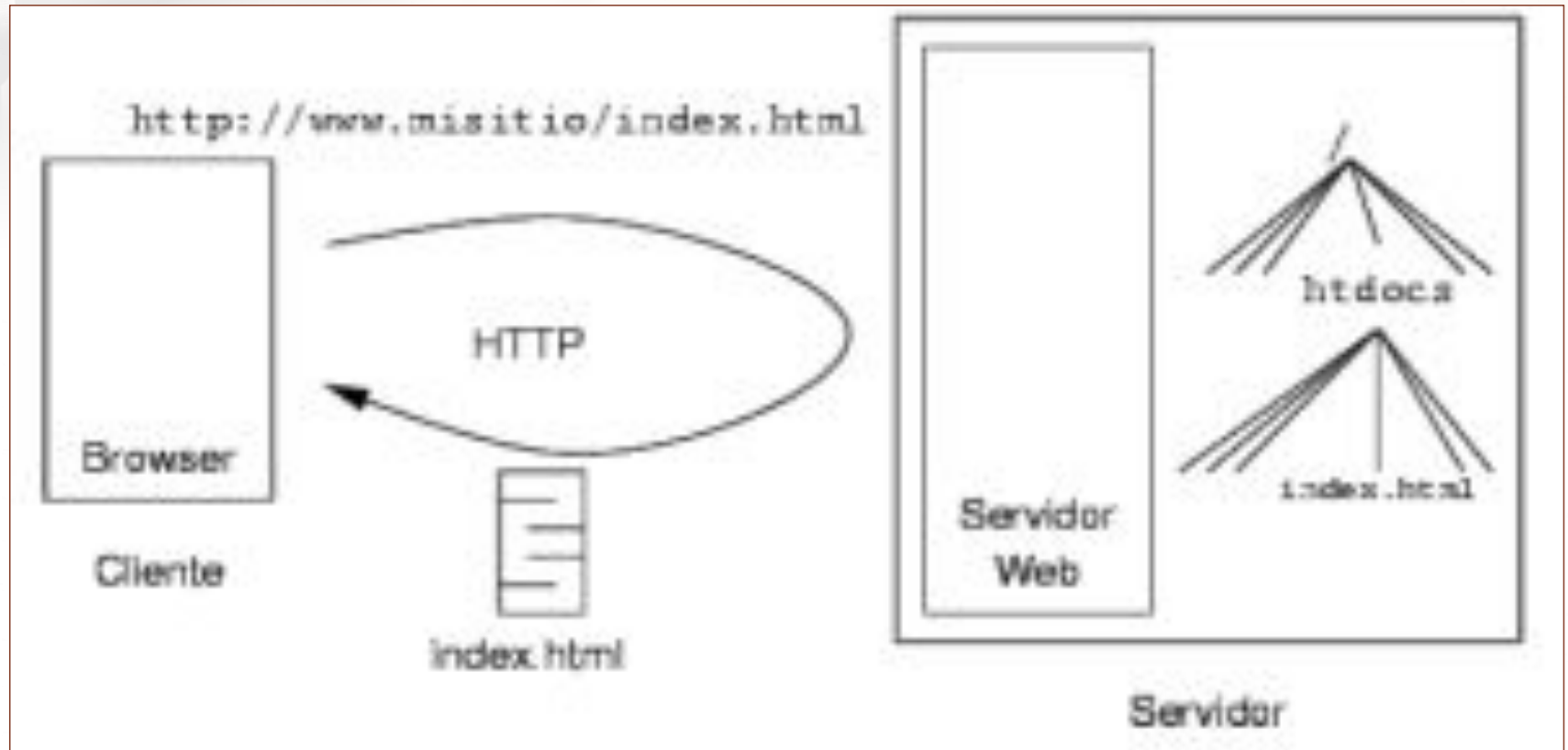
YOUR  
LOGO

您的公司名称写在这里

# Aplicaciones web

- ✚ Una aplicación Web es proporcionada por un servidor Web y utilizada por
- ✚ usuarios que se Conectan desde cualquier punto vía clientes Web (browsers o
- ✚ navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:
- ✚ Un servidor Web
- ✚ Una conexión de red
- ✚ Uno o más clientes

# Aplicaciones web





# Modelos en capas

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas,

Una capa representa un elemento que procesa o trata información. Los tipos son:

**Modelo de dos capas:** La información atraviesa dos capas entre la interfaz y

la administración de los datos.

**Modelo de n-capas:** La información atraviesa varias capas, el más habitual es el

modelo de tres capas.

# Modelos en capas / Modelo de 2 Capas

1. Gran parte de la aplicación corre en el lado del cliente (fat client).
2. Las capas son:
3. Cliente (fat client): La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
4. Servidor: Administra los datos.
5. Las limitaciones de este modelo son.
6. Es difícilmente escalable
7. Número de conexiones reducida
8. Alta carga de la red.
9. La flexibilidad es restringida
10. La funcionalidad es limitada.





# Modelos en capas / Modelo de N Capas - 3 Capas

Esta diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas, introduce una capa intermedia (la capa de proceso) Entre presentación y los datos, los procesos pueden ser manejados de forma separada a la interfaz de usuario y a los datos, esta capa intermedia centraliza la lógica de negocio, haciendo la administración más sencilla, los datos se pueden integrar de múltiples fuentes,

las aplicaciones web actuales se ajustan a este modelo.

## 1. Capa de presentación (parte en el cliente y parte en el servidor)

Recoge la información del usuario y la envía al servidor (cliente)

Manda información a la capa de proceso para su procesamiento

Recibe los resultados de la capa de proceso

Generan la presentación

Visualizan la presentación al usuario (cliente)

## 2. Capa de proceso (servidor web)

Recibe la entrada de datos de la capa de presentación

Interactúa con la capa de datos para realizar operaciones

Manda los resultados procesados a la capa de presentación

## 3. Capa de datos (servidor de datos)

Almacena los datos

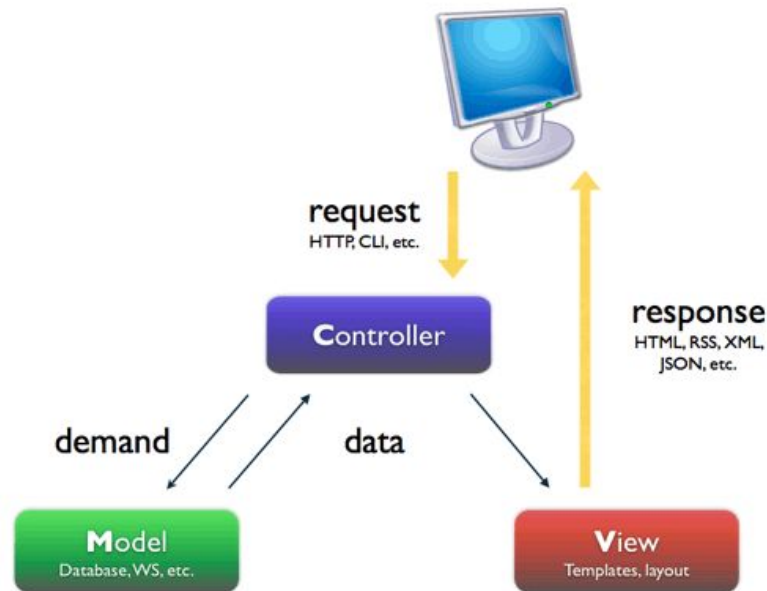
Recupera datos

Mantiene los datos

segura la integridad de los datos

# Uso del patrón MVC

1. El patrón de arquitectura Modelo-Vista-Controlador (MVC) separa una aplicación en tres componentes principales: el modelo, la vista y el controlador.





# Aplicación Web / Front-End Frameworks

El uso de librería en la capa de presentación a evolucionado tanto en hoy en día se

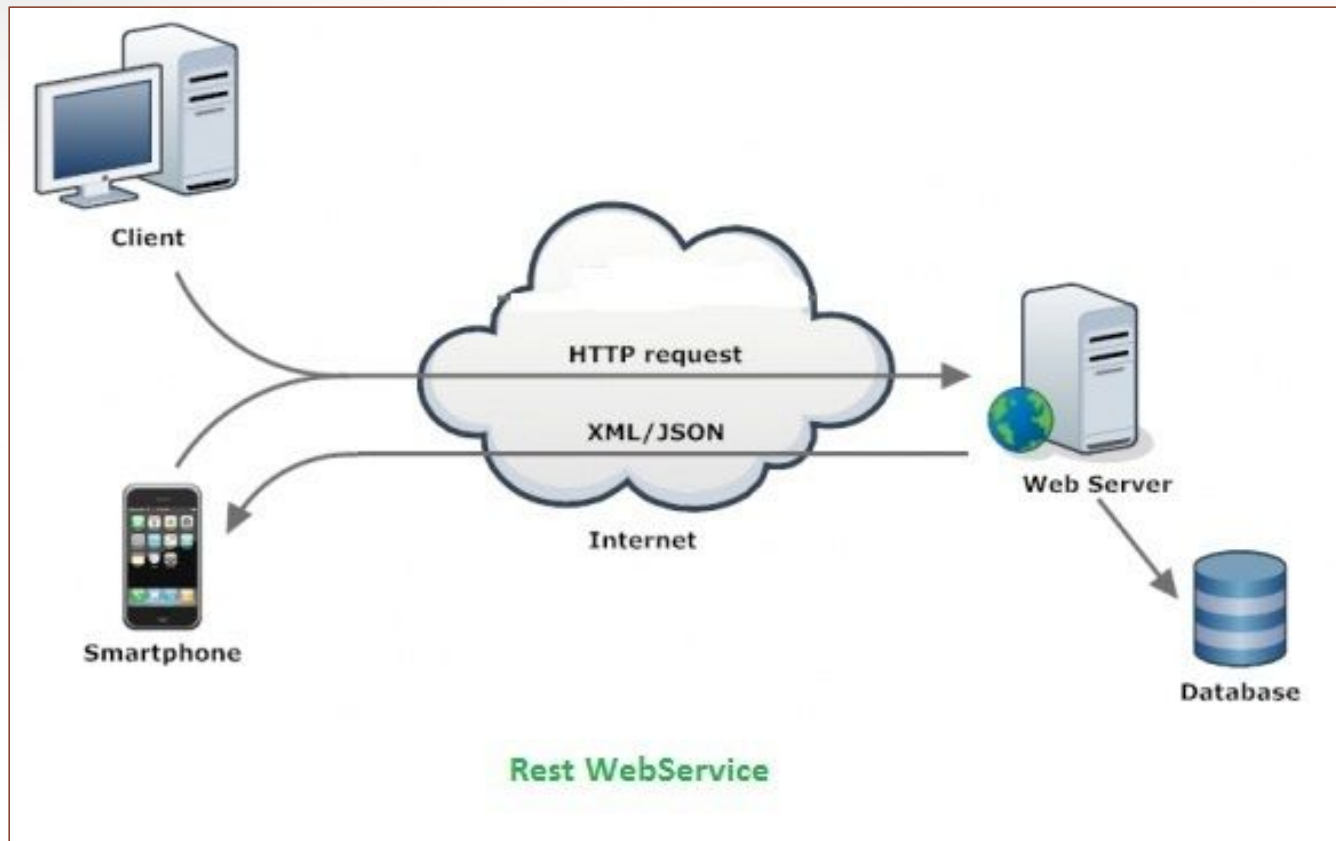
pueden encontrar un cantidad considerable de framework de desarrollo

principalmente desarrollando en javascript, así como embedidos en el lenguaje o

una mezcla de ambos.



# Aplicación Web / Front-End Frameworks



# Aplicación Web / Front-End Frameworks

## Principales características:

**Data Binding (Enlace de datos)** – permite el mapeo de datos en base de un modelo de datos, permite tanto la carga, modificación y actualización de datos, de manera muy practica y sencilla.

**UI widget(controles personalizados)** – otra característica de los framework es el manejo de una gama de controles personalizados, generalmente orientados a formularios de datos, reportes y estadísticos.

**AJAX(Asynchronous JavaScript And XML)** – generalmente la comunicación con los componentes back-end se realiza mediante ajax, sea llamando a web services o recursos en json.

# Framework más conocidos basados en Javascript

1. Angular JS
2. EXT JS
3. jQWidgets

# Framework más conocidos basados en Javascript

## 1. Angular JS

### Ejemplo

```
angular
  .module('appInventario', [])
  .controller('ListaController', ListaController);

function ListaController($scope, $http) {
  $http
    .get("inventario.json")
    .success(function(data) {
      $scope.productos = data;
    });
}
```

JS

Controlador

# Framework más conocidos basados en Javascript

## 2. EXT JS

```
var form = Ext.create('Ext.form.Panel', {  
    fullscreen: true,  
    items: [  
        {  
            xtype: 'textfield',  
            name: 'name',  
            label: 'Name'  
        },  
        {  
            xtype: 'emailfield',  
            name: 'email',  
            label: 'Email'  
        },  
        {  
            xtype: 'passwordfield',  
            name: 'password',  
            label: 'Password'  
        }  
    ]  
});
```

**Name**

**Email**

**Password**



# Framework más conocidos basados en Javascript

## 3. JQWidgets

jqxChart Chart with Grid example

Day	Keith	Erica	George
Select Filter			
Tuesday	25	25	30
Wednesday	30	20	25
Thursday	35	25	45
Friday	20	20	25
Saturday	30	20	30
Sunday	60	45	90

