# CAPITULO II – SISTEMA WEB

Este capítulo está dedicado a realizar una descripción completa de los elementos conceptuales que se manejan sobre sistemas web. Asimismo, se detallan las características principales de cada elemento. Una vez conceptualizado dichos elementos, se abordara el tema de los servicios en la nube, especialmente los servicios SAAS y una de las formas de implementarlas. Finalmente se describe acerca de las aplicaciones de una sola página o SPA y el diseño web adaptable.

## Sistema web

Un sistema web o también denominado aplicación web según Lujan (2002), es: “Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HyperText Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones”.

Estos tipos de aplicaciones son populares debido a lo práctico del navegador web, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea, etc. que son ejemplos bastante conocidos de aplicaciones web.

Una aplicación web contiene elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la aplicación responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

En vista de que algunas veces existen confusiones sobre páginas web estáticas y dinámicas, a continuación se hace una diferencia entre ellas.

Una **página web estática** tiene las siguientes propiedades:

* Realizadas principalmente en XHTML o HTML.
* Para modificación de contenidos es imprescindible acceder al servidor donde está alojada la página.
* El proceso de actualización es lento y manual.
* No ofrece bases de datos, interacción, etc.
* El usuario no puede seleccionar, ordenar o modificar los contenidos o el diseño de la página.
* Su diseño y desarrollo es más rápido y económico que el de una web dinámica.

Una **página web dinámica** tiene las siguientes propiedades:

* Muchas posibilidades en diseño y desarrollo.
* Utiliza varios lenguajes y técnicas de programación en su desarrollo.
* El proceso de actualización es muy sencillo y no requiere entrar en el servidor.
* Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico, etc.
* El usuario puede alterar el diseño, contenidos o presentación de la página a su gusto.
* Su diseño y desarrollo es más caro que el de una web estática, por su mayor complejidad y sus mayores prestaciones.

Por las características listadas anteriormente, se concluye que una aplicación web o sistema web es un tipo de página web dinámica, ya que concuerda con dichas características. Además, se considera que es un “sistema de información”, porque funciona como cualquier otra aplicación de escritorio, pudiendo almacenar grandes cantidades de información.

### Sistemas web y su diferencia con las aplicaciones de escritorio

En las aplicaciones de escritorio cada cliente debe instalar por separado el software. El cliente realiza peticiones a otro programa (el servidor) que le da respuesta. Una mejora en el servidor, como parte de la aplicación, requiere normalmente una mejora de los clientes instalados en cada computadora personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML que es soportado por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de extensiones tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario.

Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

### 2.1.2 Ventajas

A continuación, se muestra una lista de las ventajas que ofrecen este tipo de sistemas de información.

* **Ahorra tiempo:** se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
* **No hay problemas de compatibilidad**: basta tener un navegador actualizado para poder utilizarlas.
* **No ocupan espacio en nuestro disco duro.**
* **Actualizaciones inmediatas**: como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
* **Consumo de recursos bajo**: dado que toda (o gran parte) de la aplicación no se encuentra en nuestra computadora, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otra computadora.
* **Multiplataforma:** se pueden usar desde cualquier sistema operativo porque solamente es necesario tener un navegador.
* **Portables:** es independiente de la computadora donde se utilice (PC de sobremesa, portátil) porque se accede a través de una página web (solamente es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
* **La disponibilidad** suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
* **Los virus no dañan los datos** porque están guardados en el servidor de la aplicación.
* **Colaboración:** gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones en línea de calendarios u oficina.
* **Los navegadores ofrecen cada vez más y mejores funcionalidades** para crear "aplicaciones web enriquecidas" (Rich Internet application o RIA).

### 2.1.3 Desventajas

* **Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio**. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.
* **La disponibilidad depende de un tercero**. El proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

### Arquitecturas Cliente/Servidor

Un sistema web, tradicionalmente funciona bajo una arquitectura Cliente / Servidor, ya que se despliega en una red de computadoras. Según Lujan (2002), esta arquitectura se define de la siguiente manera.

“Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Normalmente, los servidores son ordenadores potentes dedicados a gestionar unidades de disco (servidor de ficheros), impresoras (servidor de impresoras), tráfico de red (servidor de red), datos (Servidor de bases de datos) o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores”.

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos.



*Figura 2.1: Arquitectura Cliente / Servidor*

*Fuente: Elaboración propia*

### Separación de funciones

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles.

* **Lógica de presentación.** La presentación de los datos es una función independiente del resto.
* **Lógica de negocio (o aplicación).** Los flujos de trabajo pueden cambiarse según las necesidades existentes de un procesador a otro.
* **Lógica de datos.** La gestión de los datos debe ser independiente para poder ser distribuida según las necesidades de la empresa en cada momento.

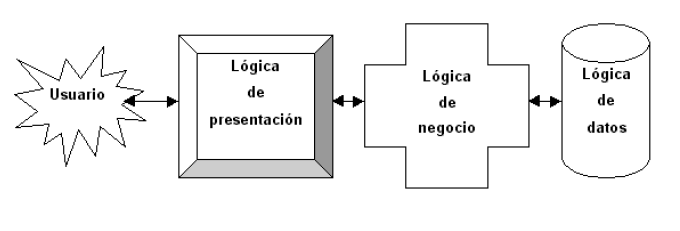


Figura 2.2: Separación de funciones en arquitectura cliente servidor.

Fuente: (Lujan, 2002).

### 2.2.2 Modelos de distribución en aplicaciones cliente / servidor

Según como se distribuyan las tres funciones básicas de una aplicación (presentación, negocio y datos) entre el cliente y el servidor, podemos contemplar tres modelos: presentación distribuida, aplicación distribuida, datos distribuidos.

#### 2.2.2.1 Presentación distribuida

El cliente sólo mantiene la presentación, el resto de la aplicación se ejecuta remotamente. La presentación distribuida, en su forma más simple, es una interfaz gráfica de usuario, a la que se le pueden acoplar controles de validación de datos, para evitar la validación de los mismos en el servidor.

#### 2.2.2.2 Aplicación distribuida

Es el modelo que proporciona máxima flexibilidad, puesto que permite tanto a servidor como a cliente mantener la lógica de negocio realizando cada uno las funciones que le sean más propias, bien por organización, o bien por mejora en el rendimiento del sistema.

#### 2.2.2.3 Datos distribuidos

Los datos son los que se distribuyen, por lo que la lógica de datos es lo que queda separada del resto de la aplicación. Se puede dar de dos formas: ficheros distribuidos o bases de datos distribuidas.

### 2.2.3 Arquitecturas de dos y tres niveles

Una arquitectura de dos niveles está basada en un sistema gestor de bases de datos donde el cliente mantiene la lógica de la presentación, negocio, y de acceso a los datos, y el servidor únicamente gestiona los datos. Suelen ser aplicaciones cerradas que supeditan la lógica de los procesos cliente al gestor de base de datos que se está usando.

En las arquitecturas de tres niveles, la lógica de presentación, la lógica de negocio y la lógica de datos están separadas, de tal forma que mientras la lógica de presentación se ejecutará normalmente en la estación cliente, la lógica de negocio y la de datos pueden estar repartidas entre distintos procesadores.

El objetivo de aumentar el número de niveles en una aplicación distribuida es lograr una mayor independencia entre un nivel y otro, lo que facilita la portabilidad en entornos heterogéneos.

### Elementos de un sistema web

Los elementos más importantes de una aplicación web son: El cliente, el servidor y el protocolo http.

### El cliente

El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HyperText Markup Language (HTML) que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en Java. También se suelen emplear plug-ins3 que permiten visualizar otros contenidos multimedia (como Flash4), aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de incompatibilidad entre distintas plataformas. Por tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.).

### El Servidor

El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. En los sistemas Unix suele ser un demonio y en los sistemas Microsoft Windows un servicio.

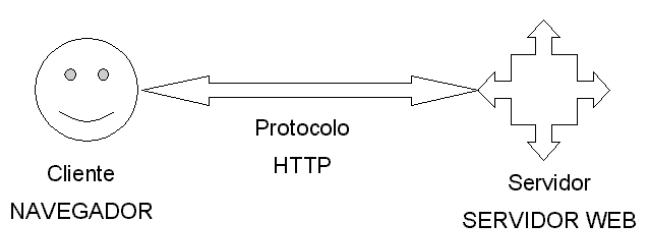
La parte servidor de las aplicaciones web está formada por páginas estáticas que siempre muestran el mismo contenido y por programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas dinámicas.

La programación en la parte del servidor se la realiza mediante tecnologías como ASP, ColdFusion, Java Server Pages (JSP), servlets, PHP, Javascript, etc. Todas ellas son muy similares, se basan en los mismos principios y ofrecen resultados equivalentes.

### Protocolo HTTP

El protocolo HTTP (Hypertext Transfer Protocol) Protocolo de transferencia de hipertexto, forma parte de la familia de protocolos de comunicaciones Transmission Control Protocol/Internet Protocol (TCP/IP), que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores.

En la figura 2.3 se puede observar la relación entre los tres elementos de un sistema web.



*Figura 2.3: Relación entre los elementos de un sistema web*

*Fuente: Elaboracion propia*

## Transferencia de páginas web

El proceso completo, desde que el usuario solicita una página, hasta que el cliente web (navegador) se la muestra con el formato apropiado, es el siguiente:

* El usuario especifica en el cliente web la dirección de la página que desea consultar: el usuario escribe en el navegador la dirección (Universal Resource Locator (URL)) de la página que desea visitar.
* El cliente establece una conexión con el servidor web.
* El cliente solicita la página o el objeto deseado.
* El servidor envía dicha página u objeto (o, si no existe, devuelve un código de error).
* Si se trata de una página HTML, el cliente inicia sus labores de interpretación de los códigos HTML. Si el cliente web encuentra instrucciones que hacen referencia a otros objetos que se tienen que mostrar con la página (imágenes, sonidos, animaciones multimedia, etc.), establece automáticamente comunicación con el servidor web para solicitar dichos objetos.
* Se cierra la conexión entre el cliente y el servidor.
* Se muestra la página al usuario.

Obsérvese que siempre se libera la conexión, por lo que ésta sólo tiene la duración correspondiente a la transmisión de la página solicitada. Esto se hace así para no desperdiciar innecesariamente el ancho de banda de la red mientras el usuario lee la página recibida.

Cuando el usuario activa un enlace de la página, se establece una nueva conexión para recibir otra página o elemento multimedia. Por ello, el usuario tiene la sensación de que está disfrutando de una conexión permanente cuando realmente no es así.

Un detalle importante es que para casa objeto que se transfiere por la red se realiza una conexión independiente. Por ejemplo, si el cliente web solicita una página que contiene dos imágenes integradas, se realizan tres conexiones: una para el documento HTML y dos para los archivos de las imágenes.

## Entornos web

Los sistemas web se emplean en tres entornos informáticos muy similares que suelen confundirse entre sí: Internet, intranet y extranet.

### Internet

Internet es una red global que conecta millones de ordenadores por todo el mundo. Su nacimiento se sitúa en 1969, en un proyecto de investigación del Departamento de Defensa de Estados Unidos. En 1998, la Internet tenía más de 100 millones de usuarios en todo el mundo, en diciembre de 2000 unos 400 millones y el número sigue creciendo rápidamente. Más de 100 países están conectados a este nuevo medio para intercambiar todo tipo de información.

### Intranet

Una intranet es una red de ordenadores basada en los protocolos que gobiernan Internet (TCP/IP) que pertenece a una organización y que es accesible únicamente por los miembros de la organización, empleados u otras personas con autorización.

Una intranet puede estar o no conectada a Internet. Un sitio web en una intranet es y actúa como cualquier otro sitio web, pero los cortafuegos (firewall) lo protegen de accesos no autorizados.

### Extranet

Una extranet es una intranet a la que pueden acceder parcialmente personas autorizadas ajenas a la organización o empresa propietaria de la intranet.

Mientras que una intranet reside detrás de un cortafuego y sólo es accesible por las personas que forman parte de la organización propietaria de la intranet, una extranet proporciona diferentes niveles de acceso a personas que se encuentran en el exterior de la organización.

## HTML Hypertext Transfer Protocol

HTML es un lenguaje de marcas (etiquetas) que se emplea para dar formato a los documentos que se quieren publicar en la WWW. Los navegadores pueden interpretar las etiquetas y muestran los documentos con el formato deseado.

Las páginas web o páginas HTML son unos cheros escritos en el lenguaje HTML. El desarrollo de estas páginas abarca un amplio grupo de tecnologías, desde las páginas más sencillas que sólo usan el lenguaje HTML hasta las más complejas que usan Dynamic HTML (DHTML), JavaScript, applets realizados en Java o componentes ActiveX.

## Cloud Computing

Según la UAM (2011), cloud computing o computacion en la nube es: “Cuando se hace referencia a la nube, se está aludiendo a un termino con algunos años de historia y que es una forma metafórica de nombrar a Internet. Básicamente la computación en la nube consiste en los servicios ofrecidos a través de la red tales como correo electrónico, almacenamiento, uso de aplicaciones, etc., los cuales son normalmente accesibles mediante un navegador web”.

### Historia

El desarrollo de la computación en la nube comenzó a través de grandes empresas de servicios de

Internet como Google y Amazon los cuales construyeron su propia infraestructura. A partir de allí surgió una arquitectura: un sistema de recursos distribuidos de manera horizontal, introducidos como servicios virtuales de tecnologías de información (TI) escalados masivamente y manejados como recursos agrupados y configurados continuamente.

Aunque la implementación es reciente, la idea no es nueva ya que se ha discutido en el medio desde hace algunos años con distintos nombres tales como: “utility computing”, computación en demanda, computación elástica, o “grid computing” (no confundir con el procesamiento en red mencionado anteriormente). Haciendo una comparación de ideas y tecnologías entre las décadas de 1960 y 1970 con la época actual se tiene lo siguiente:

**Década 1960**

* Uso de “terminales tontas” que dependían de un sistema central más potente.
* La información se guardaba en el servidor.
* Se necesitaba conexión constante con el sistema central para funcionar correctamente.
* Imposibilidad de instalar aplicaciones.

**Siglo XXI**

* Terminales poco potentes pero autosuficientes(Netbooks, tablets, smartphones)
* La información se aloja en los servidores del proveedor de servicios aunque hay posibilidad de guardar información en la terminal del usuario.
* Se necesita conexión constante con el sistema central para hacer uso de todos los recursos.
* Dependiendo de la terminal, es posible instalar aplicaciones, aunque la idea es ejecutarlas a través de Internet.

### Características

* No es necesario disponer de un equipo potente, tan solo de un aparato con conexión a internet; esto debido a que el dispositivo del usuario no realizara ningún proceso complejo y los ficheros puede guardarse en la nube. Los servidores en donde se hallan los programas que se utilicen son los encargados de las tareas complicadas que antes se realizaba localmente.
* Con el uso del Cloud Computing no hay necesidad por parte del usuario de conocer la infraestructura detrás de esta, ya que pasa a ser una abstracción, “una nube” donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y con pocas fallas.
* Es auto reparable: En caso de surgir un fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de ´esta se genera uno nuevo.
* Posee un alto nivel de seguridad: El sistema esta creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad; de esto se ocupa el sistema proveedor que se encarga de cifrar los datos.
* Disponibilidad de la información: No se hace necesario guardar los documentos editados por el usuario en su computadora o en medios físicos propios ya que la información radicara en Internet permitiendo su acceso desde cualquier dispositivo conectado a la red (con autorización requerida).

### Arquitectura de la nube

La computación en la nube basa su arquitectura haciendo una separación entre hardware, plataforma y aplicaciones quedando las siguientes capas:

**2.7.3.1 Software como servicio (SaaS)**

Se encuentra en la capa más alta y consiste en la entrega de aplicaciones completas como un servicio.

El proveedor de tecnologías de información y comunicación (TIC) ofrece el SaaS (Software as a Service).

Para ello dispone de una aplicación que se encarga de operar y mantener y que frecuentemente es desarrollada por ´el mismo. Con ella se encarga de dar servicio a multitud de clientes a través de la red, sin que estos tengan que instalar ningún software adicional.

Un ejemplo claro es la aplicación para el manejo del correo electrónico por medio de un navegador de Internet.

#### 2.7.3.2 Plataforma como Servicio (PaaS)

En orden descendente, PaaS (Platform as a Service) es la siguiente capa. Básicamente su objetivo se centra en un modelo en el que se proporciona un servicio de plataforma con todo lo necesario para dar soporte al ciclo de planteamiento, desarrollo y puesta en marcha de aplicaciones y servicios web a través de la misma. El proveedor es el encargado de escalar los recursos en caso de que la aplicación lo requiera, de que la plataforma tenga un rendimiento óptimo, de la seguridad de acceso, etc. Para desarrollar software se necesitan bases de datos, herramientas de desarrollo y en ocasiones servidores y redes.

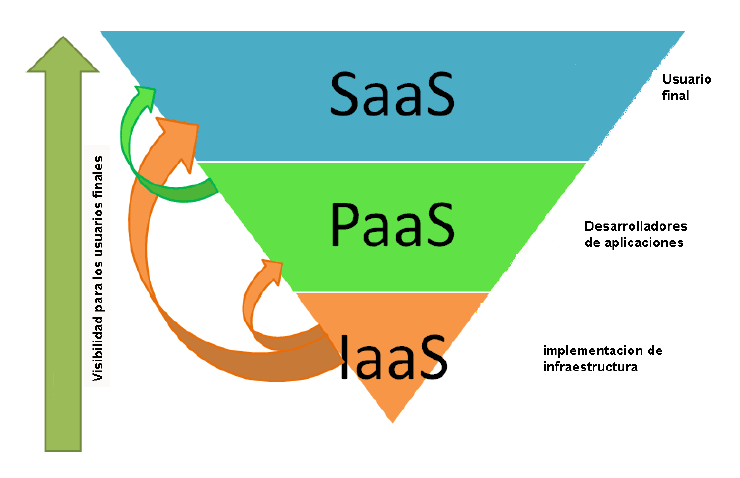
Con PaaS el cliente únicamente se enfoca en desarrollar, depurar y probar ya que la herramienta necesaria para el desarrollo de software es ofrecido a través de Internet, lo que teóricamente permite aumentar la productividad de los equipos de desarrollo.

Con el uso de PaaS, se abstrae del hardware físico al cliente, lo cual es interesante para muchos desarrolladores web, y es probable que llegue a reemplazar a las empresas de alojamiento tradicionales.

#### 2.7.3.3 Infraestructura como Servicio (IaaS)

IaaS (Infrastructure as a Service) corresponde a la capa más baja. La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, ruteadores, swtiches así como tiempo de computo evitando de esta manera tener un servidor local y toda la infraestructura necesaria para la conectividad y mantenimiento dentro de una organizaron. Con una IaaS lo que se tiene es una solución en la que se paga por consumo de recursos solamente usados: espacio en disco utilizado, tiempo de CPU, espacio para base de datos, transferencia de datos, etc.

Las IaaS permiten desplazar al proveedor la mayor parte de los factores relacionados con la gestión de las maquinas con el ahorro de costos al pagar solo por lo consumido y olvidarse de tratar con máquinas y su mantenimiento. Por otro lado, IaaS puede permitir una escalabilidad automática o semiautomática, de forma que se puedan contratar más recursos según los se requieran5.



*Figura 2.4: Capas de computación en la nube.*

*Fuente: (UAM, 2011)*

### 2.7.4 Ventajas

* **Prestación de servicios a nivel mundial.** Las infraestructuras de cloud computing proporcionan mayor capacidad de adaptación, recuperación completa de pérdida de datos (con copias de seguridad) y reducción al mínimo de los tiempos de inactividad.
* **Una infraestructura 100% de cloud computing** permite también al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de software, ya que éste es provisto por el proveedor de la infraestructura o la plataforma en la nube. Un gran beneficio del cloud computing es la simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.
* **Implementación más rápida y con menos riesgos**, ya que se comienza a trabajar más rápido y no es necesaria una gran inversión. Las aplicaciones del cloud computing suelen estar disponibles en cuestión de días u horas en lugar de semanas o meses, incluso con un nivel considerable de personalización o integración.
* **Actualizaciones automáticas** que no afectan negativamente a los recursos de TI. Al actualizar a la última versión de las aplicaciones, el usuario se ve obligado a dedicar tiempo y recursos para volver a personalizar e integrar la aplicación. Con el cloud computing no hay que decidir entre actualizar y conservar el trabajo, dado que esas personalizaciones e integraciones se conservan automáticamente durante la actualización.
* **Contribuye al uso eficiente de la energía.** En este caso, a la energía requerida para el funcionamiento de la infraestructura. En los centros de datos tradicionales, los servidores consumen mucha más energía de la requerida realmente. En cambio, en las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio.

### Desventajas

* **La disponibilidad de las aplicaciones** está sujeta a la disponibilidad de acceso a Internet.
* **Los datos "sensibles" del negocio no residen en las instalaciones de las empresas**, lo que podría generar un contexto de alta vulnerabilidad para la sustracción o robo de información.
* **La confiabilidad de los servicios depende de la "salud" tecnológica y financiera** de los proveedores de servicios en nube. Empresas emergentes o alianzas entre empresas podrían crear un ambiente propicio para el monopolio y el crecimiento exagerado en los servicios.
* **La disponibilidad de servicios altamente especializados** podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
* **La madurez funcional de las aplicaciones** hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.
* **Seguridad.** La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que éstos requieren.

## Software como Servicio (SaaS)

Software como Servicio (del inglés: Software as a Service, SaaS) es un modelo de distribución de software, en donde la compañía de tecnologías de información (IT) provee el servicio de mantenimiento, operación diaria y soporte del software usado por el cliente. Regularmente el software puede ser consultado en cualquier computador, esté presente en la empresa o no. Se deduce que la información, el procesamiento, los insumos y los resultados de la lógica de negocio del software están hospedados en la compañía de IT.

Las actividades son gestionadas desde alguna ubicación central, en lugar de hacerlo desde la sede de cada cliente, permitiendo a los clientes el acceso remoto a las aplicaciones mediante la web.

### 2.8.1 Características

Las características del software como servicio incluyen:

* Acceso y administración a través de una red.
* Actividades gestionadas desde ubicaciones centrales, en lugar de la sede de cada cliente, permitiéndoles tener acceso remoto a las aplicaciones a través de la web.
* La distribución de la aplicación es más cercana al modelo uno a muchos (una instancia con múltiples usuarios) que al modelo uno a uno, incluyendo arquitectura, precios, colaboración, y administración.
* Actualizaciones centralizadas, lo cual elimina la necesidad de descargar parches por parte de los usuarios finales.
* Multiplataforma, en donde el cliente puede ejecutar la aplicación desde diferentes sistemas operativos, incluso, diferentes tipos de dispositivos, tales como celulares, tabletas o computadores.

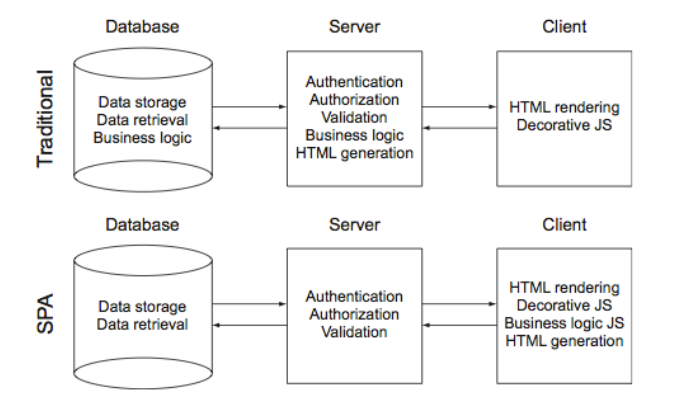
### 2.8.2 Aplicaciones de una sola página (SPA)

Una de las formas de implementar un servicio Saas (Software as a service), es desarrollando un SPA (Single page Application) en español, aplicación de una sola página, lo cual es una solución óptima para los servicios Saas.

Según (Hung, 2014) una aplicación de una sola página es: “Una aplicación de una sola página (SPA) es una aplicación web instalado en una sola página y hace que no vuelva a cargar la página durante su uso para proporcionar una mejor experiencia de usuario y aumentar el rendimiento”.

### 2.8.2.1 Estructura de un SPA

Cuando la aplicación web, se traslada al navegador, los requisitos para el rendimiento en el servidor son bastante importantes. La siguiente figura muestra cómo las SPA solucionan un problema de rendimiento con la lógica de negocio y el procesamiento de plantillas migran desde el servidor y el cliente.



*Figura 2.5: Responsabilidades de la base de datos, servidor y cliente en una aplicación web tradicional y una aplicación de una sola página (SPA)*

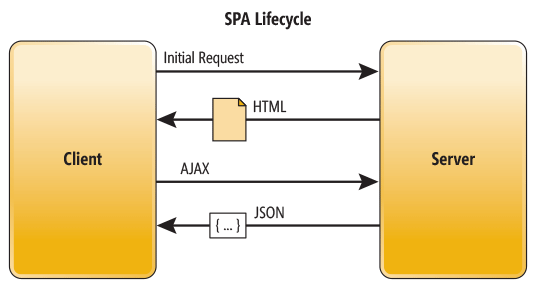
*Fuente: (*HUNG, 2014*)*

Como se puede ver en el gráfico, con una SPA, toda la lógica de negocio de la base de datos y el servidor (Tradicional) se desplazan al lado del cliente (SPA) con el fin de mejorar el rendimiento de la Base de datos y servidor. Esto porque una de las principales características de un servicio SaaS es que puede tener múltiples usuarios.

Una aplicación web tradicional, como se puede ver en la figura, realiza trabajos que hacen que las solicitudes tomen mucho mas tiempo para ser resueltas, lo cual se traduce en una experiencia baja de usuario, sobretodo tratándose de un servicio SaaS.

La forma en que una SPA funciona es básicamente de la siguiente manera: cuando el usuario interactúa con la página realizando alguna acción (pulsar un botón, enlaces u otros eventos) se realiza una petición asíncrona (AJAX) al servidor, cuya respuesta (generalmente XML, JSON o similar) es interpretada por la aplicación que actualizará el DOM (Document Object Model) **“sin recargar el sitio”**, permitiendo visualizar los cambios con alta velocidad y sin desperdicio de ancho de banda. Las SPA delegan muchas operaciones que anteriormente se realizaban en el servidor al lado del cliente, reduciendo así la complejidad del sistema.

La figura siguiente muestra de una forma clara el proceso de funcionamiento de una SPA, realizando peticiones asíncronas mediante tecnología AJAX.



*Figura 2.6: Funcionamiento de un SPA, realizando peticiones asíncronas*

*Fuente: (*HUNG, 2014)

#### 2.8.2.2 Comunicación con el Servidor

Las peticiones que realiza una SPA, normalmente resultan en datos crudos (XML o JSON), o son plantillas HTML nuevos que están siendo regresados. En los casos donde el HTML regresa como resultado, JavaScript en el cliente actualiza el área parcial de DOM (Document Object Model). Cuando el dato crudo es regresado, el JavaScript de lado del cliente procesa el XML o el JSON para hacer la traducción de los datos crudos a un HTML la cual es usada para actualizar la información del área parcial del DOM.

Con todo lo descrito anteriormente sobre aplicaciones de una sola página (SPA), se concluye que la forma de comunicación entre el cliente (SPA) y el servidor, es mediante peticiones asíncronas, las cuales requieren únicamente de la URL del recurso solicitado. Razon por la cual en el lado del servidor se hace uso de los llamados “Servicios REST”.

**2.8.2.3 Transferencia de Estado Representacional (REST) API**

Según (Hung, 2014), un servicio REST es: “Transferencia de estado representacional (REST) es la arquitectura sin estado, basado en el protocolo HTTP para representar el modelo de cómo debería funcionar los sistemas web modernos. Los servicios REST utilizan la tecnología y protocolos de web existente para acceder y manipular los recursos que utilizan los verbos (métodos de solicitud HTTP).”

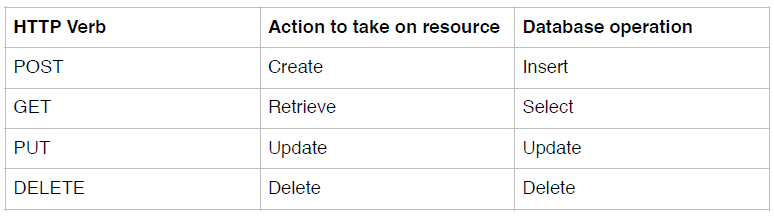


Tabla 2.7: Cómo las acciones de HTTP son relevantes para el funcionamiento de bases de datos.

Fuente: (HUNG, 2014)

Hay una serie de verbos para el enfoque REST como GET, POST, PUT y DELETE para manipular los datos en la base de datos. Estos verbos corresponden a un CRUD tradicional (Crear, Leer, actualizar, eliminar) las operaciones en una base de datos. Además, REST también tiene 5 categorías de respuestas: información general, una petición correcta, redirigir, error en el cliente y el error en el lado del servidor. La respuesta depende del tipo de petición HTTP, como se indica en la siguiente tabla:

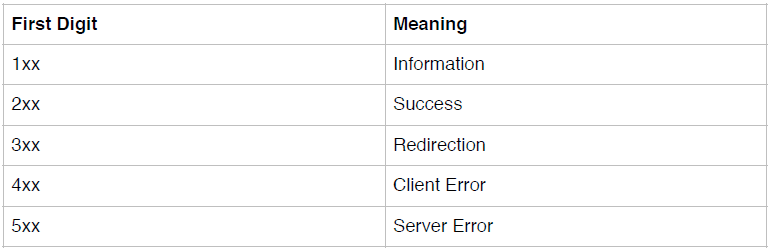


Tabla 2.8: Respuestas a peticiones comunes en un servicio REST.

Fuente: (HUNG, 2014)

### 2.8.3 Diseño web Adaptable