## APLICACIONES WEB

Una aplicación web es cualquier [aplicación](http://www.alegsa.com.ar/Dic/aplicacion.php) que es accedida vía [web](http://www.alegsa.com.ar/Dic/web.php) por una [red](http://www.alegsa.com.ar/Dic/red.php) como [internet](http://www.alegsa.com.ar/Dic/internet.php) o una [intranet](http://www.alegsa.com.ar/Dic/intranet.php).

En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del [navegador](http://www.alegsa.com.ar/Dic/navegador.php) (por ejemplo, un [applet](http://www.alegsa.com.ar/Dic/applet.php) de Java) o codificado con algún lenguaje soportado por el navegador (como [JavaScript](http://www.alegsa.com.ar/Dic/javascript.php), combinado con [HTML](http://www.alegsa.com.ar/Dic/html.php)); confiándose en el navegador web para que reproduzca ([renderice](http://www.alegsa.com.ar/Dic/renderizacion.php)) la aplicación.

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples [plataformas](http://www.alegsa.com.ar/Dic/plataforma.php).

### Tecnología www (World Wide web)

La tecnología del World Wide Web es un sistema de información el cual está compuesto por agentes interconectados. Un agente es un programa que actúa a nombre de otra persona, entidad, o proceso con el fin de intercambiar información y presentar la información en un formato legible al usuario. Por ejemplo un navegador de páginas webs es un agente (Konqueror, Mozilla) utilizado por el usuario para accesar las páginas webs que se encuentran en los agentes servidores (Apache, Tomcat, etc). Para que los agentes puedan intercambiar información y presentar la información en un formato legible al usuario, los agentes deben satisfacer tres propiedades: representación, identificación e interacción.

**Representación**

La propiedad de representación es utilizada para estructurar la información contenida en un documento Web. Esta propiedad utiliza una combinación de grafos en forma de árbol, grafos directos y objetos para estructurar la información. En un documento Web, los siguientes tipos de información pueden ser estructurados: Texto, imágenes y objetos. La información contenida en un documento Web es estructurada en forma de árbol donde cada nodo es considerado un objeto. Cada nodo puede estar compuesto por atributos, nodos hijos y contenido. Cada nodo es considerado una entidad. Una entidad un recurso que goza de identidad. Por ejemplo un documento Web es un recurso, por lo tanto todos los nodos contenidos en el documento Web son recursos. Un recurso es nombrado e identificado por la propiedad de identificación. Una vez identificado un recurso, los agentes utilizan la propiedad de interacción para accesar, actualizar, eliminar o intercambiar recursos entre agentes.

El principal estándar internacional utilizado para representar la información de documentos electrónicos es el estándar ISO 8879. Este estándar es conocido con el nombre de Standard Generalized Mark Up Language (SGML). El SGML es un metalenguaje utilizado para definir, describir y normalizar documentos electrónicos basados en etiquetas. Una etiqueta es utilizada para dar: significado, estructura, nombre a un nodo, entidad y acción aplicada a la información etiquetada. La mayoría de las especificaciones de la tecnología World Wide Web son aplicaciones del SGML o derivan del SGML. Por ejemplo: El lenguaje extensible de etiquetas (Extensible Mark Up Language) (XML) es el principal estándar para estructurar la información en la tecnología World Wide Web. Este estándar deriva del SGML. Las especificaciones XML permiten a los usuarios definir las etiquetas de un documento modelo el cual es utilizado para dar estructura y significado a la información de un documento. Un documento modelo se define con las especificaciones XML Schema o Document Type Definition. Una vez definido un documento modelo se pueden crear múltiples documentos. De esta manera la información de un documento es estructurada y normalizada. Las especificaciones XML delegan la función de formato o presentación de la información a las especificaciones XSL y CSS.

**Identificación**

La función de la propiedad de identificación es identificar, localizar y nombrar los recursos definidos por la propiedad de representación los cuales son almacenados en los repositorios de información de los agentes. Las especificaciones del RFC 2396 (URI) satisface la propiedad de identificación. Un URI está compuesto por tres definiciones: uniformidad, recurso, identificador.

La definición de Uniformidad establece el conjunto de reglas que definen las secuencias correctas de los elementos que conforman a un URI. Este conjunto de reglas proporciona un mecanismo común para interpretar los diferentes tipos de identificadores de recursos.

La definición de recurso es el mapeo conceptual a un nodo. Este mapeo es visto como un grafo directo entre dos nodos.

La definición de identificador es un objeto que actúa como referencia a algo que tiene identidad. Ejemplo un recurso.

La sintaxis genérica que representa a un URI es la siguiente:

<esquema>:<parte especifica del esquema >

Dicha sintaxis es utilizada para definir las aplicaciones de un URI. Entre estas aplicaciones se encuentran los localizadores de recursos (URL) y nombre de recursos (URN).

Un URL define a un subconjunto de URI que identifican los siguientes parámetros: nombre del recurso, Localidad del recurso y protocolo de acceso del recurso. En general la parte específica del esquema de un URL es el siguiente:

<Esquema>://<usuario>:<password>@<host>:<puerto>/<ruta del recurso>

El parámetro <Esquema> identifica el protocolo de acceso del recurso. Los parámetros <usuario> y <password> son opcionales ya que la presencia de estos parámetros en un URL depende del protocolo de acceso del recurso. Por ejemplo el protocolo FTP permite el uso de estos dos parámetros. El parámetro <host> define el nombre de dominio completamente calificado del agente. El parámetro <puerto> identifica el número de puerto del agente. Y el parámetro <ruta del recurso> identifica el nombre del recurso

**Interacción**

Una vez que un recurso es identificado por la propiedad de identificación, los agentes utilizan la propiedad de interacción para accesar, actualizar, eliminar o intercambiar recursos entre agentes vía protocolos. El principal protocolo implementado en los agentes en la tecnología del World Wide Web es el protocolo (HTTP). El protocolo http funciona a partir de solicitudes. Las solicitudes más comunes del protocolo http son:

* GET. Es una solicitud para leer un recurso. Ejemplo una página Web.
* PUT. Es una petición para almacenar un recurso.
* DELETE. Indica una solicitud para remover un recurso.
* POST. Es una petición que añade información a un recurso nombrado.
* HEAD. Es una petición para leer la cabecera de una página Web.

Cada solicitud hecha por el navegador a través del protocolo http recibe una respuesta acompañada por un código de estado. El código de estado más común es el código 200 (OK), este código indica que el servidor respondió a la solicitud satisfactoriamente.

### Características

* El usuario puede acceder fácilmente a estas aplicaciones empleando un [navegador web](http://www.alegsa.com.ar/Dic/navegador.php) (cliente) o similar.
* Si es por internet, el usuario puede entrar desde cualquier lugar del mundo donde tenga un [acceso a internet](http://www.alegsa.com.ar/Dic/acceso%20a%20internet.php).
* Pueden existir miles de usuarios pero una única aplicación instalada en un [servidor](http://www.alegsa.com.ar/Dic/servidor.php), por lo tanto se puede actualizar y mantener una única aplicación y todos sus usuarios verán los resultados inmediatamente.
* Usan tecnologías como Java, JavaFX, JavaScript, DHTML, Flash, Ajax; que dan gran potencia a la [interfaz de usuario](http://www.alegsa.com.ar/Dic/gui.php).
* Emplean tecnologías que permiten una gran [portabilidad](http://www.alegsa.com.ar/Dic/portabilidad.php) entre diferentes [plataformas](http://www.alegsa.com.ar/Dic/plataforma.php). Por ejemplo, una aplicación web flash podría ejecutarse en un dispositivo móvil, en una computadora con Windows, Linux u otro sistema, en una consola de videojuegos, etc.

### Protocolos (http)

El protocolo  HTTP, siglas que en inglés significan Hypertext Transfer Protocol, cuyo equivalente en nuestro idioma sería el de Protocolo de [Transferencia](http://www.mastermagazine.info/termino/6939.php) de [Hipertexto](http://www.mastermagazine.info/termino/5254.php), es lo que nos permite navegar cómodamente por la red sin necesidad de memorizar grandes cifras o textos más que complicados.

Lo que permite este protocolo es justamente gestionar el Acceso a un punto remoto, brindando entonces una especie de atajo, para lo cual tendremos asignado una Vía de Comunicación determinada, que se otorga por el contenido de Hipertexto, es decir, la asignación de un texto específico para poder hallar rápidamente un destino en la Web.

Las conexiones que se realizan entre un equipo y otro se pierden constantemente, por lo que para poder conectarse en una futura ocasión se utilizan los conocidos Cookies, archivos que no ocupan un significante archivo en un medio de almacenamiento, gestionando simplemente la información del usuario que ha iniciado sesión para poder ser reconocido en caso de ingresar nuevamente al sitio.

De este modo, no solo se genera un atajo en lo que respecta la comunicación de un destino a otro, sino que también se obtiene una forma de reconocimiento con el destino elegido, mostrando el nombre de usuario y teniendo la posibilidad de acceder sin tener que reingresar la contraseña o accediendo nuevamente a los sitios restringidos sin tener que pasar nuevamente por el proceso de autenticación.

En los últimos tiempos se ha establecido además la evolución conocida como HTTPS, que es justamente la variante "Segura" del protocolo HTTP que permite a los Navegadores Web (Es decir, a las aplicaciones que hacen uso de este protocolo para acceder a los sitios de destino) mostrar una conexión libre de amenazas.

## HERRAMIENTAS TECNOLÓGICAS

### Java

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

Java permite jugar, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como, por ejemplo, cursos en línea, servicios bancarios en línea y mapas interactivos. Si no dispone de Java, muchas aplicaciones y sitios web no funcionarán.

### Java standar edition (1.7)

Java Standar edition es un kit de desarrollo de software que se utiliza para escribir applets y aplicaciones con el lenguaje de programación Java. El mismo que va dirigido a Los programadores de software que programen applets y aplicaciones que utilicen la tecnología Java.

### Java EE (Java Enterprise edition 7)

Java Enterprise Edition o mejor conocido como Java EE define el estándar en cuanto al desarrollo de aplicaciones empresariales distribuidas basadas en componentes, utilizando el modelo de múltiples capas. El mismo está orientado a la construcción de sistemas empresariales para lo cual proporciona: un modelo de componentes, un modelo estandarizado y un conjunto de servicios estandarizados.

Su finalidad es facilitar y proporcionar un mayor soporte y factibilidad en cuanto a la construcción de una aplicación web empresarial.

Su diseño y plataforma están basados en Java Standar Edition (SE), sin embargo no significa un reemplazo de la misma, sino un subproducto especializado del mismo, para un fin específico. Es por ello, que Java SE proporciona el soporte de lenguaje básico sobre el que Java EE se ejecuta.

### JSF

Java Server Faces (JSF) es un marco de trabajo para crear aplicaciones java J2EE basadas en el patrón MVC de tipo 1. Las principales características de JSF:

* Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios HTML.
* Asocia a cada vista con formularios un conjunto de objetos java manejados por el controlador (managed beans) que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.
* Introduce una serie de etapas en el procesamiento de la petición, como por ejemplo la de validación, reconstrucción de la vista, recuperación de los valores de los elementos, etc.
* Utiliza un sencillo fichero de configuración para el controlador en formato xml.
* Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.
* Y lo que es más importante: forma parte del estándar J2EE. En efecto, hay muchas alternativas para crear la capa de presentación y control de una aplicación web java, como Struts y otros frameworks, pero solo JSP forma parte del estándar.

### Ventajas y desventajas

**CUADRO N.**

**VENTAJAS Y DESVENTAJAS DE JSF**

|  |  |
| --- | --- |
| **Ventajas** | **Desventajas** |
| El código JSF con el que se crea las vistas (etiquetas jsp) es muy parecido al HTML estándar. Se lo puede  utilizar fácilmente por desarrolladores y diseñadores web. | El desarrollo de la página se complicaría si se usa una metodología de desarrollo habitual, mas no se adapta al sistema de desarrollo de JSF. |
| JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página |
| Resuelve validaciones, conversiones, mensajes de error e internacionalización (i18n) | La maquetación compleja también complica el desarrollo ya que obliga a utilizar muchas etiquetas y atributos, especialmente en los datatables. |
| Permite introducir javascript en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario). |
| Es extensible, por lo que se pueden desarrollar nuevos componentes a medida, También se puede modificar el comportamiento del framework mediante APIs que controlan su funcionamiento. | Abuso del javascript. JSF permite utilizar javascipt para hacer más rápida una página html, evitando peticiones al servidor. Sin embargo la introducción de javascript en la página complica y alarga los desarrollos con JSF, y en general con JSP. La capa javascript añade etapas adicionales a la aplicación, que hace más difícil su depurado. |
| Forma parte del estándar J2EE, mientras que otras tecnologías para creación de vistas de las aplicaciones no lo forman, como por ejemplo Struts. |

### Primefaces

PrimeFaces es una librería de componentes visuales open source desarrollada y mantenida por [Prime Technology](http://www.prime.com.tr/), una compañía Turca de IT especializada en consultoría ágil, JSF, Java EE y Outsourcing. El proyecto es liderado por Çağatay Çivici, un miembro del "JSF Expert Group" (y forofo del Barça).

Las principales características de Primefaces son:

* Soporte nativo de Ajax, incluyendo Push/Comet.
* Kit para crear aplicaciones web para móviles.
* Es compatible con otras librerías de componentes, como [JBoss RichFaces](http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsfIntro).
* Uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).
* Es un proyecto open source, activo y bastante estable entre versiones.

### Primefaces Extension

### Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) es una arquitectura de componentes de servidor que simplifica el proceso de construcción de aplicaciones de componentes empresariales distribuidos en Java.

Con su utilización es posible escribir aplicaciones escalables, fiables y seguras sin escribir código de infraestructura. La existencia de infraestructura permite un desarrollo más rápido de la parte servidora. Dado que son componentes, permiten desarrollar aplicaciones portables entre distintas plataformas (son Java) y servidores de aplicaciones (especificación estándar).

### JPA

El Java Persistence API (JPA) es una especificación de Sun Microsystems para la persistencia de objetos Java a cualquier base de datos relacional. Esta API fue desarrollada para la plataforma JEE e incluida en el estándar de EJB 3.0, formando parte de la Java Specification Request JSR 220, aunque su uso no se limita a los componentes software EJB. Se puede utilizar en aplicaciones web y aplicaciones clientes.

Para ello, combina ideas y conceptos de los principales frameworks de persistencia, como Hibernate, Toplink y JDO. El mapeo objeto-relacional (es decir, la relación entre entidades Java y tablas de la base de datos, queries con nombre, etc) se realiza mediante anotaciones en las propias clases de entidad.

La persistencia o el almacenamiento permanente, es una de las necesidades básicas de cualquier sistema de información de cualquier tipo. En primer lugar, se propuso que el programa tratara los datos haciendo consultas directas a la base de datos. Después, se propuso trabajar con objetos, pero las bases de datos tradicionales no admiten esta opción.

Debido a esta situación, aparecieron los motores de persistencia, cuya función es traducir entre los dos formatos de datos: de registros a objetos y de objetos a registros. Persistir objetos Java en una base de datos relacional implica serializar un árbol de objetos Java en una base de datos de estructura tabular y viceversa. Esencial es la necesidad de mapear objetos Java para optimizar velocidad y eficiencia de la base de datos.

### Interfaces JPA

Los tipos de interfaces de las que se compone JPA son:

**javax.persistence.Persistence**: Contiene métodos estáticos de ayuda para obtener una instancia de Entity Manager Factory de una forma independiente al vendedor de la implementación de JPA. Una clase de inicialización que va proporcionar un método estático para la creación de una Entity Manager Factory.

**javax.persistence.EntityManagerFactory**: La clase javax.persistence.Entity.Manager.Factory nos ayuda a crear objetos de EntityManager utilizando el patrón de diseño del Factory. Este objeto en tiempo de ejecución representa una unidad de persistencia particular. Generalmente va a ser manejado como un singleton y proporciona métodos para la creación de instancias EntityManager.

**javax.persistence.EntityManagerFactory**: La clase javax.persistence.Entity es una anotación Java que se coloca a nivel de clases Java serializables y que cada objeto de una de estas clases anotadas representa un registro de una base de datos.

**javax.persistence.EntityManager**: Es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada Entity Manager puede realizar operaciones CRUD (Create, Read, Update, Delete) sobre un conjunto de objetos persistentes. Es un objeto único, no compartido que representa una unidad de trabajo particular para el acceso a datos. Proporciona métodos para gestionar el ciclo de vida de las instancias entidad y para crear instancias Query.

**javax.persistence.Query**: La interface javax.persistence.Query está implementada por cada vendedor de JPA para encontrar objetos persistentes manejando cierto criterio de búsqueda. JPA estandariza el soporte para consultas utilizando Java Persistence Query Language (JPQL) y Structured Query Language (SQL). Podemos obtener una instancia de Query desde una instancia de un Entity Manager.

**javax.persistence.EntityTransaction**: Cada instancia de Entity Manager tiene una relación de uno a uno con una instancia de javax.persistence.EntityTransaction, permite operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.

### Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias sql. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada puede generar BBDD en cualquiera de los entornos soportados : Oracle, DB2, MySql, etc.. Y lo más importante de todo, es open source, es decir su adquisición es de manera gratuita.

Uno de los posibles procesos de desarrollo consiste en, una vez realizado el diseño de datos realizado, se debe mapear este a ficheros XML siguiendo la DTD de mapeo de Hibernate. Desde estos se podrá generar el código de objetos persistentes en clases Java y también crear BBDD independientemente del entorno escogido.

Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Una posible configuración básica de hibernate es la siguiente:

**GRAFICO N.**

**CONFIGURACIÓN BÁSICA DE HIBERNATE**

