1. ¿Cuál es la diferencia entre un sistema cliente servidor y un sistema distribuido?

En un sistema Cliente-Servidor, el cliente hace un requerimiento al servidor para que este satisfaga un pedido. Se realiza un diálogo directo entre ambos sin ningún intermediario.

En un sistema Distribuido existe un intermediario entre el cliente y el servidor. Este intermediario gestiona el diálogo, los pedidos y las respuestas. Estos pueden ser orientados a objetos, a servicios, a mensajes, etc.

1. ¿Cuál es el problema de Estado? Definición completa. ¿Y cómo se soluciona?

ASP es asincrónica, por lo que cuando la información pasa de una página a otra se pierde la información. Esto se debe a que ASP está basada en el protocolo HTTP, que es sin estado (no persiste la información al pasar de página). Esto se resuelve mediante cookies o variables de sesión.

1. ¿Qué es una cookie?

Es un archivo de texto de como mucho 4KB. Estas se almacenan en el directorio del navegador de tu ordenador o en las subcarpetas de datos de programa.

Las cookies se crean cuando utilizas tu navegador para visitar un sitio web que las utiliza para hacer un seguimiento de tus movimientos por el sitio, ayudarte a indicarte dónde dejaste la navegación, recordar tu inicio de sesión y selección de tema, preferencias y otras funciones de personalización.

1. ¿Cómo la clasifica desde el punto de vista temporal?

Se pueden clasificar desde el punto de vista temporal por su permanencia como **cookies de sesión** o **cookies permanentes**. Las cookies permanentes poseen una propiedad llamada “Expires” que establece su permanencia.

Las cookies de sesión se crean de forma temporal en una subcarpeta de tu navegador mientras visitas una página web. Una vez que abandonas el sitio, la cookie de sesión es eliminada.

Las cookies permanentes siguen en la subcarpeta de tu navegador y se activan de nuevo en cuanto visitas otra vez la página específica que la creó.

1. ¿Cómo la clasifica desde el punto de vista de contenido?

Se pueden clasificar en **cookies tradicionales** y **diccionarios de cookies**. Las cookies poseen una propiedad llamada HasKeys que determinan de qué tipo se trata.

Las cookies tradicionales son las que poseen un valor único al ser consultadas. Es decir, son del tipo Cookies[“Nombre”] = “Valor”.

Los diccionarios de cookies son las cookies que poseen en su interior una colección de valores, cada uno con su clave. Es decir, son del tipo Cookies[“Nombre”][“Clave”] = “Valor de la clave”.

1. En el modelo de objetos. ¿Dónde están las cookies? ¿Qué son Propiedades/ eventos/ métodos/ colecciones?

Los objetos del modelo de ASP son:

Request-Response-session-application-server

**Request** Es una instancia de la clase System.Web.HttpRequest. Representa los valores y propiedades que permiten cargar la página en el navegador.

Propiedades:

TotalBytes: devuelve el número de bytes en el input actual.

ServerVariables: devuelve una colección de variables del servidor Web.

QueryString: devuelve una colección de variables HTTP Query String.

Métodos:

BinaryRead: realiza una lectura binaria en un número especificado de bytes del input actual.

**Response**: representa la respuesta del servidor a la Request del cliente. Es una instancia de la clase System.Web.HttpResponse.

Propiedades:

* Buffer: obtiene o modifica el valor que indica si se debe buffear el output y enviarlo cuando se termina de procesar la response completa.
* Charset: obtiene o modifica el set de caracteres HTTP del flujo de salida (output).
* Expires: obtiene o modifica los minutos restantes para que una página cacheada en un navegador expire.
* ExpiresAbsolute: obtiene o modifica la fecha y hora absoluta en que se removerá la información del caché.
* IsClientConnected: devuelve un valor que indica si el cliente está conectado al servidor.
* Status: modifica la línea de estado que se devuelve al cliente.

Métodos:

* AddHeader: añade un encabezado HTTP al flujo de salida.
* AppendToLog: añade un log de información personalizado al archivo log del IIS.
* BinaryWrite: escribe una cadena de caracteres binarios al flujo de salida HTTP.
* Close: cierra el socket de conexión al cliente.
* End: envía todo el flujo de salida buffeado al cliente, finaliza la ejecución de la página y activa el evento EndRequest.
* Flush: envía todo el flujo de salida buffeado al cliente.
* Write: escribe al flujo de salida del HTTP response.

**Session**: almacena información sobre la sesión del usuario y los cambios a la configuración. Contiene una colección (Contents) con todos los ítems de la sesión.

Propiedades:

* CodePage: especifica el set de caracteres que serán usados al mostrar el contenido dinámico.
* LCID: obtiene o modifica un entero que especifica el lugar o región.
* SessionID: obtiene el id único de cada usuario.
* Timeout: obtiene o devuelve el periodo de timeout en minutos para el objeto Session de la aplicación.

Métodos

* Abandon: destruye la sesión del usuario.
* Content.Removes: destruye un ítem de la colección Contents.f
* Content.RemoveAll(): destruye todos los ítems.

Eventos

* Session\_OnEnd: ocurre cuando acaba una sesión.
* Session\_OnStart: ocurre cuando comienza una sesión.

**Application**: mantiene información que será usada por las páginas de la aplicación. Posee una colección (Contents) con los ítems de la aplicación.

Métodos:

* Lock: prohíbe al usuario modificar las variables.
* Unlock: permite al usuario modificar las variables.

Eventos:

* Application\_OnEnd: ocurre cuando se cierran todas las sesiones y se finaliza la aplicación.
* Application\_OnStart: ocurre cuando se crea la primera sesión.

**Server** Es una instancia de la clase System.Web.HttpServerUtility.

Propiedades:

* ScriptTimeOut: lee y escribe el valor del time-out Request en segundos.

Métodos:

* CreateObject: crea una instancia de un objeto.
* Execute: ejecuta el handler de la ruta virtual especificada en el Request.
* GetLastError: devuelve la última excepción.
* HtmlEncode: cambia un String ordinario a uno con caracteres HTML permitidos.
* MapPath: devuelve la ruta física que corresponde a la ruta del archivo virtual especificado.
* Transfer: transfiere la ejecución a otra página web en la aplicación.

1. Si fuera un diccionario de cookies, ¿cómo me doy cuenta en el código? ¿Cómo la recorro?

En el código, me doy cuenta que se trata de un diccionario de cookies cuando existen dos corchetes luego del Request.Cookies. Es decir, Request.Cookies[“Nombre”][“Clave”]. Se recorre con un loop for o foreach.

1. ¿Qué es el global.asax? ¿Cuál es su estructura? ¿Para qué me sirve?

Es un archivo de configuración que contiene la declaración de los objetos, las variables y los métodos que pueden ser accedidos por una página de una aplicación ASP. Cada aplicación puede tener solo un archivo Global.asa y estará guardado en el directorio raíz de la aplicación.

Sirve para manejar eventos de aplicación (tales como el inicio y el final de una aplicación), así como también para declarar datos que están disponibles en diferentes solicitudes de la aplicación y a través de diferentes sesiones del navegador.

1. ¿Cuáles son los componentes del IIS? Internet information Server. ¿Cómo me doy cuenta de que lo tengo instalado?

Internet Information Server 5.1, o IIS, es el servidor web estándar del sistema operativo Windows/NT/2000/XP. Este ofrece varios servicios de Internet: Servidor web(HTTP), Servidor FTP y Servidor SMTP.

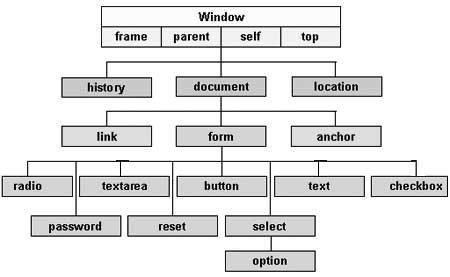
Para comprobar si está funcionando se abre una nueva ventana de su navegador preferido y se escribe la dirección http://localhost/, http://127.0.0.1/ o http://nombre/, donde ¨nombre¨ es el nombre del equipo.

Me doy cuenta que lo tengo instalado al escribir “inetmgr” en la búsqueda de Windows.

1. ¿Qué objeto representa al browser? ¿Cuál representa a la página? ¿Cómo sería la jerarquía de objetos?

El objeto window representa al browser.

El objeto document representa la página.



1. ¿Cuál es la diferencia entre Get y Post?

La diferencia entre los métodos Get y Post radica en la forma de enviar los datos a la página cuando se hace un submit de un formulario. Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no podemos verlos (en un segundo plano u "ocultos" al usuario).

En Post los parámetros no son guardados en el historial del navegador (más seguro), en el Get si porque forman parte de la URL (menos seguro).

Get acepta marcadores y Post no. Get permite sólo caracteres ASCII y Post no tiene restricciones.

1. ¿Cuáles son las propiedades del formulario?

Id: es el identificador único del formulario

Name: es el nombre único del formulario.

Action: es el lugar al cual se envía el formulario para ser procesado. El action define la URL a la cual se envía dicho formulario.

Method: método de envío de los datos insertados en un formulario. El method puede ser:

* GET = envía los datos en una cadena "visible". Conveniente para enviar pocos datos.
* POST = envía los datos en forma "invisible". Conveniente para enviar una gran cantidad de datos.

Target: define la ventana o marco (frame) en la que se mostrarán o procesarán los resultados del formulario.

1. ¿Si mando con Post con qué recupero la información? ¿Si mando con Get con qué recupero la información?

Si el método utilizado es POST deberemos de recuperar la información mediante **Request.Form[“nombreCampo”]**, donde el “nombreCampo” es el valor que le hayamos asignado al campo name de los input del formulario.

En el caso de que el método utilizado sea GET deberemos de recuperar la información mediante **Request.QueryString[“nombreCampo”]**.

1. Grafique y explique el ciclo de vida Cliente-Servidor.
2. Se le muestra al usuario un formulario en el cual introducirá los datos.
3. El usuario presiona el botón para enviar la información.
4. Dicho botón llama a una función Javascript que validará los datos.
5. La función Javascript valida los datos, devolviendo al usuario un mensaje de error en caso de haber.
6. En caso de no haber error, la función Javascript hace un submit del formulario, enviando la info ingresada al servidor.
7. El servidor realiza la captura de los datos recibidos, poniéndolos en variables.
8. El servidor realiza las operaciones y cálculos necesarios con esa información.
9. El servidor le devuelve al cliente el resultado del procesamiento del paso anterior.
10. En el ámbito de XML, ¿qué es un analizador? ¿Qué es un Parser? ¿Cuáles conoce y en qué se diferencian?

Un analizador es un software que permite recorrer un documento y extraer información del mismo. Un analizador XML es una API (Application Programming Interface) que permite crear una estructura jerárquica que contiene los datos del documento XML inicial. Además, distinguimos entre analizadores validadores (que verifican la existencia y la validación de la DTD correspondiente) y analizadores no validadores (que verifican si el documento está bien formado).

XML Parser es una biblioteca de software o un paquete que proporciona la interfaz para aplicaciones cliente para trabajar con documentos XML. Se verifica la estructura del documento XML y también puede validar los documentos XML. Los navegadores modernos se han incorporado en los analizadores XML.

**DOM**

* Analizador de modelo de árbol (basado en objeto) (Árbol de nodos).
* DOM carga el archivo en la memoria y luego analiza el archivo.
* Tiene restricciones de memoria ya que carga todo el archivo XML antes de analizar.
* DOM es de lectura y escritura (puede insertar o eliminar nodos).
* Si el contenido XML es pequeño, entonces prefiera el analizador DOM.
* La búsqueda hacia atrás y hacia adelante es posible para buscar los tags y evaluar la información dentro de las tags. Así que esto da la facilidad de navegación.
* Más lento en el tiempo de ejecución.

**SAX**

* Analizador basado en eventos (Secuencia de eventos).
* SAX analiza el archivo tal como lo lee, es decir, analiza nodo por nodo.
* Sin restricciones de memoria, ya que no almacena el contenido XML en la memoria.
* SAX es de solo lectura, es decir, no puede insertar o eliminar el nodo.
* Use el analizador SAX cuando el contenido de la memoria sea grande.
* SAX lee el archivo XML de arriba a abajo y la navegación hacia atrás no es posible.
* Más rápido en tiempo de ejecución.

1. ¿Qué es un DTD y cuál es su estructura?

Una DTD es un documento que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc, que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc. El procesador XML utiliza la DTD para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

1. ~~Escriba 4 palabras reservadas y los 4 primeros renglones de una DTD.~~

~~ELEMENT, ATTLIST, NOTATION, ENTITY.~~

~~<?xml version="1.0"?>  
<!DOCTYPE peliculas [  
<!ELEMENT peliculas (pelicula\*)>  
<!ATTLIST pelicula tipo (drama | comedia | misterio | aventura) #IMPLIED>  
<!ATTLIST pelicula restricciones (13 | 21 | ATP) #IMPLIED>  
]>~~

1. XML Schema es una verdadera estructura XML. ¿En qué se diferencian una DTD de un XML Schema?

* Los esquemas XML se escriben en XML, mientras que DTD se deriva de SGML de sintaxis.
* Los esquemas XML define tipos de datos para los elementos y los atributos, mientras que DTD no admite los tipos de datos.
* Los esquemas XML permite el soporte para espacios de nombres mientras DTD no es compatible.
* Los esquemas XML manejan la cardinalidad con “minOccurs” y “maxOccurs”, mientras que DTD la maneja con “\*” (0 a muchos), “+” (1 a muchos) y “?” (opcional).
* Los esquemas XML se pueden manipular su propio con el DOM de XML, pero no es posible en el caso de la DTD.
* Con el esquema XML usuario no necesita aprender un nuevo idioma, pero trabajando con DTD es difícil para un usuario.
* Los esquemas XML son extensibles mientras DTD no es extensible.fa

1. Explique todo acerca de la palabra reservada #PCDATA

PCDATA - Parsed Character Data.

Es un texto que será analizado por un analizador. Las etiquetas dentro del texto serán tratadas como marcas y las entidades serán expandidas.

El modelo de contenido PCDATA dice que un elemento puede contener texto plano. La parte “analizada” significa que el marcado se analiza en lugar de mostrarse como texto sin procesar.

1. Explique todo acerca de la palabra reservada #CDATA

CDATA - (Unparsed) Character Data

Es un texto que *no* será analizado por un analizador. Las etiquetas dentro del texto *no* serán tratadas como marcas y las entidades *no* serán expandidas.

El modelo de contenido CDATA permite el contenido de texto plano, y las referencias de marca y entidad se ignoran en el contenido del elemento. En una sección CDATA, todos los caracteres son legales (incluidos <, >, &, ' y "), excepto la etiqueta de final "]]> ".

1. Explique DOM.

El DOM es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

DOM define un estándar de acceso y manipulación de documentos, tanto en formato XML como HTML. Este proporciona un conjunto estándar de objetos para representar documentos, ​ un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

Los objetos del DOM modelizan tanto la ventana del navegador como el historial, el documento o página web, y todos los elementos que pueda tener dentro la propia página, como párrafos, divisiones, tablas, formularios y sus campos, etc. A través del DOM se puede acceder, por medio de Javascript, a cualquiera de estos elementos, es decir a sus correspondientes objetos para alterar sus propiedades o invocar a sus métodos. Con todo, a través del DOM, queda disponible para los programadores de Javascript, cualquier elemento de la página, para modificarlos, suprimirlos, crear nuevos elementos y colocarlos en la página, etc.

1. XML. ¿Con qué palabra reservada se representa un atributo?

ATTLIST. Si tiene #IMPLIED significa que es obligatorio.

1. ~~Dado el siguiente código XML, encontrar el error.~~

~~No aplica.~~

1. El objeto Server del modelo ASP. Es importantísimo. ¿Cuál es la única propiedad que tiene?

La propiedad es ScriptTimeOut, que determina el tiempo límite en segundos que un llamado al servidor puede estar ejecutándose.

1. ¿Cuál es el método de este objeto que permite incorporar un componente en mi página? ¿Cómo lo hace?

CreateObject

var Conn = new Server.CreateObject[“ADODB.Connection”];

1. ¿En qué se diferencian URL Encode de HTML Encode?

**URL Encode** reemplaza los caracteres especiales con caracteres que pueden ser entendidos por los navegadores web/servidores web con el propósito de direccionar, por ejemplo, pasando una URL como parámetro. Por ejemplo, los espacios se reemplazan con %20, ' = %27, etc.

**HTML Encode** reemplaza los caracteres especiales con cadenas de caracteres que son reconocidas por el propio motor HTML para representar el contenido de la página: cosas como & se convierte en &amp; or < = &lt;, > = &lt;, esto evita que el motor HTML interprete estos caracteres como partes del formato HTML y, por lo tanto, los imprima tal cual son.

1. ¿En qué se diferencian Transfer de Execute?

Cuando se usa Server.Execute, la URL es pasada como parámetro y el control pasa a la nueva página. La ejecución del Código ocurre en la nueva página. Una vez que se ejecuta, el control vuelve a la página inicial.

Cuando se usa Server.Transfer, la diferencia es que la ejecución frena en la nueva página y el control no vuelve a la página inicial

1. El componente del servidor ADRotator. ¿Para qué me sirve? ¿Qué estructura tiene en ASP? ¿Qué estructura tiene en ASP.NET? ¿Cómo lo implementaría en mi página?

Es un componente del IIS que se utiliza para añadir mensajes animados a una página web, como pueden ser noticias o publicidad, siendo capaz de presentarlos de forma aleatoria ya que se actualiza cada vez con la carga de página, lo que permite mostrar diferentes mensajes para cada usuario. Es posible asignar prioridades a los anuncios para que unos aparezcan con mayor frecuencia que otros.

El control AdRotator dispone de tres propiedades adicionales a las heredadas de la clase System.Web.UI.Control, estas son:

* AdvertisementFile
* Keyword Filter
* Target

ASP.NET:

<body>

<form id="form1" runat="server">

<h3>AdRotator Example</h3>

<asp:AdRotator id="AdRotator1" runat="server"

Target="\_self"

AdvertisementFile="~/App\_Data/Ads.xml"/>

</form>

</body>

1. Los eventos OnStart y OnEnd ¿Dónde están dentro del modelo de objetos ASP? ¿Con qué objeto del modelo de objetos están vinculado? ¿Y con qué archivo de configuración están vinculados?

Están vinculados a los objetos Application y Session. Se encuentran en el archivo Global.asa.

1. Nombre 3 eventos del objeto cliente.

onclick, onload, onscroll

1. ¿Qué es la jerarquía de objetos dentro de una página dinámica de ASP?

Es la forma en la que están anidados los objetos dentro de una página ASP. Dichos objetos pueden ser manipulados usando javascript, ya sea para cambiar su contenido, eliminarlos o moverlos al interior de otro elemento.

1. Nombre dos componentes del servidor.

AdRotator

ContentLinking

PageCounter

1. Nombre 3 archivos de configuración que forman el mundo de ASP.NET

Web.Config: maneja varias opciones que definen al sitio web. Se almacenan en archivos XML separados del código de la aplicación. Posee conexiones a la base de datos, opciones del caché, estados de sesión, seguridad y manejo de errores.

Machine.Config: es el archivo de configuración del servidor. Contiene todas las configuraciones de todos los sitios corriendo en la máquina.

Global.asax: Es el archivo de configuración que contiene la declaración de los objetos, las variables y los métodos que pueden ser accedidos por una página de una aplicación ASP.

1. ¿Por qué se usa Javascript?

Es útil para añadir características interactivas a un Sitio Web y puede ser aplicado a un documento HTML.

1. ¿Cómo se diferencia el sistema cliente servidor del sistema distribuido?

Ídem pregunta 1.

1. ¿Cuáles son las diferencias entre XML y HTML?

HTML es el HyperText Markup Language, un lenguaje de marcado donde los tags vienen predefinidos y no es case sensitive.

XML es el Extensible Markup Language, un lenguaje de marcado donde los tags pueden ser creados por el usuario (extensible), es case sensitive y programable.

1. ¿Cuáles son las diferencias entre un lenguaje compilado y un lenguaje interpretado?

Los **lenguajes compilados** son aquellos lenguajes de alto nivel que requieren de un compilador para traducirlo y crear la parte ejecutable. Ejemplos: C y C++.

Los **lenguajes interpretados** son aquellos lenguajes de programación también de alto nivel en donde el código fuente debe ser traducido a un lenguaje que la máquina entienda y pueda realizar, no crean un archivo externo. Estas instrucciones son leídas y ejecutadas en tiempo real. Ejemplos: Python y Ruby

1. ¿Cuáles son las ventajas de tener datos estructurados de forma arbórea?

Son fáciles de recorrer, permitiendo acceder rápidamente a los datos. Posee métodos para facilitarlo: FirstChild, etc.

1. ¿Cuáles son las diferencias entre un DTD y XML Schema?

Ídem pregunta 18.

1. ¿Cómo puedo vincular o capturar una página o imagen para que esté dentro de otra página?

Utilizando el objeto iframe, que es un control que representa otra ventana de navegador.

1. ¿Qué es AdRotator? ¿Cómo lo instancio?

Ídem pregunta 28. Se puede instanciar agregando el control a la página ASP.NET o con el metodo Server.CreateObject.

1. ¿Qué es CreateObject? ¿Que necesito para crear un banner dentro de un HTML o XML?

Es un método del objeto Server que me permite agregar un componente a mi página, instanciando un objeto.

Para crear un banner, necesito las imágenes que contendrá el banner y un archivo de texto que contendrá todas las propiedades del banner. Luego, se crea un objeto AdRotator utilizando el método Server.CreateObject.

1. **¿Qué es GetAdvertisement?**

Es un método del objeto AdRotator que devuelve un HTML que muestra un anuncio en una página.

1. ¿Qué es HasKeys?

Es una propiedad de Request. Cookies que permite saber si una cookie es tradicional o se trata de un diccionario de cookies (si es 1).

1. ¿Cómo se forma un diccionario de cookies? ¿Cómo se diferencia una cookie común de un diccionario de cookies?

Un diccionario de cookies contiene una llave y un valor, y puede ser recorrido mediante una estructura for each, in, next.

1. ¿Qué es el ScriptTimeOut?

La propiedad es ScriptTimeOut representa la cantidad máxima de segundos en que una página asp puede ser procesada. Pasado ese valor se corta la ejecución de la página.

1. ¿Cómo es la estructura de HTML del cliente?

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

</body>

</html>

1. ¿Cómo es la estructura básica del servidor?

???? Ni idea a qué se refiere.

1. Si envío información con post, ¿con que lo recibo?

string razonSocial = Request.Form[“TxtRazonSocial”];

1. Si envío información con get. ¿Con qué lo recibo?

string razonSocial = Request.QueryString[“TxtRazonSocial”];

1. ¿Qué es el evento Load?

Es el método de una página que se ejecuta cuando se carga.

1. ¿Qué es el default.aspx?

Es la página índice de un sitio web. Por defecto es la primera página que se carga en el sitio web.

1. ¿Qué es una variable de sesión?

Una variable de sesión permite guardar un valor y mantenerlo entre varias páginas. Ayuda a pasar datos entre páginas. Y colabora con la limitación de estado de las páginas asp.