

*Universidad Nacional Autónoma de Nicaragua
UNAN - León*

Facultad de Ciencias



*Tema: Diseño del sitio Web con acceso a datos sobre las especies
de la Reserva de Biósfera de Bosawás.*

Monografía para optar al título de Licenciado en Computación.

Integrantes:

- ☞ Br. Denis Rafael González Montenegro.*
- ☞ Bra. Benelip Eleonor Núñez Uriarte.*
- ☞ Bra. María Auxiliadora Prado Novoa.*

Tutores:

- ☞ Lic. Álvaro Rafael Altamirano Osorio.*
- ☞ MSc. Martín Eliseo Ibarra Padilla.*

Colaboración:

- ☞ Lic. Eduardo Santiago Molina Poveda.*

Agradecimiento

En el transcurso de esta aventura académica hemos encontrado diversos obstáculos, pero más que obstáculos hemos encontrado gente que ha apoyado y acogido este proyecto como propio. De manera sincera y franca agradecemos la colaboración de:

- *Lic. Álvaro Altamirano Osorio, por su paciencia y constancia.*
- *MsC. Martín Ibarra Padilla, por su inagotable fuente de consejos.*
- *Lic. Eduardo Santiago Molina, por su crítica constructiva y detallista con relación al mundo de ASP.*
- *Dr. Ricardo Rueda, por su colaboración abundante con respecto a la parte biológica en la construcción del sitio.*
- *Lic. Luis Contreras, por su exhaustiva revisión en lo gramatical y ortográfico en el documento.*

A todas las personas que de una u otra manera contribuyeron a la realización de este proyecto, ayudando a construir una obra – aporte por nuestro sufrido país.

Los autores.

Dedicatoria

Este trabajo ha sido producto de la dedicación y el esfuerzo que se han invertido para su finalización. Sin embargo, esa dedicación y ése esfuerzo son fruto de las enseñanzas y ejemplos de quienes nos han transmitido los valores que hoy ostentamos: nuestros padres.

Es por eso, que el presente trabajo está dedicado a nuestros padres:

María Auxiliadora Novoa

Leonardo José Prado

Lipcia Leonor Uriarte Oconor

Estela Oconor de Uriarte

y Dalia Montenegro Centeno.

Esto es el fruto de tantos años...

ÍNDICE

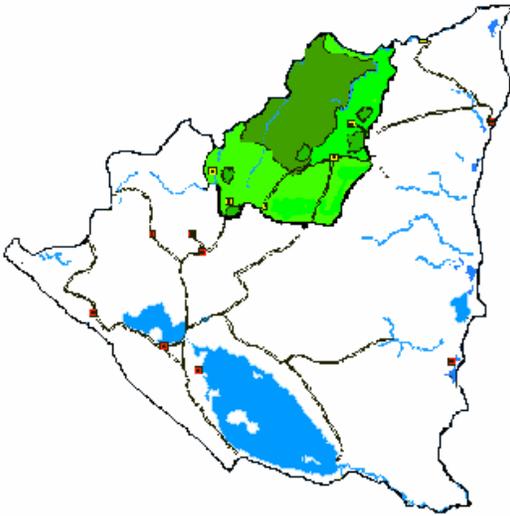
Agradecimiento	ii
Dedicatoria	iii
Índice	iv
I. Introducción	1
1. Generalidades de la Reserva de Biósfera de Bosawás	1
1.1 Bosawás y el entorno social	2
1.2 Problemática de la Reserva de Bosawás	3
II. Antecedentes	5
III. Justificación	7
IV. Objetivos	8
V. Marco Teórico	9
1. Fundamentos Biológicos	9
1.1 Relaciones biológicas	9
1.2 Clasificación de los seres vivos	10
1.3 Equilibrio ecológico	12
2. Fundamentos Informáticos	13
2.1 Bases de Datos: Conceptos Fundamentales	13
2.2 Bases de Datos Relacionales	14
2.3 Bases de Datos en el Web	15
3. Fundamentos del Internet	16
3.1 Direccionamiento IP	18
3.2 Nombres de dominio	19
3.3 Puertos	20
3.4 Protocolos	21
4. Servidores Web	23
VI. Materiales	25
VII. Metodología	26
VIII. Análisis del Sitio Web	27
A. Especificación de Requisitos Software	27
B. Diagrama de flujo de datos	50
IX. Diseño	55
A. Diagrama Entidad - Relación	55
B. Diseño de Datos	56
C. Diseño Arquitectónico	58
D. Diseño de Interfaz	59
X. Codificación	60
1. ASP: Conceptos Fundamentales	60

1.1	Introducción a ASP	60
1.2	Programación en el lado del servidor	62
1.3	Breve Historia	63
1.4	Ventajas frente a otras alternativas	64
1.4	Herramientas necesarias	65
1.4.1	Servidor Web	65
1.4.2	Navegador o Browser	66
1.4.3	Editor HTML	66
1.5	El Lenguaje Visual Basic Script	67
1.5.1	Variables	70
1.5.1.1	Funciones de Conversión	71
1.5.1.3	Restricciones de nombre de las variables	73
1.5.1.4	Alcance y vida de las variables	73
1.5.1.5	Variables escalares y variables de matrices	74
1.5.1.6	Declaración de constantes	74
1.5.2	Operadores	75
	Prioridad de Operadores	75
1.5.3	Funciones	77
1.5.3.1	Funciones de tratamiento de fechas	77
1.5.3.2	Funciones matemáticas	78
1.5.3.3	Funciones de tratamiento de strings	79
1.5.3.4	Funciones de tipos de datos	80
1.5.3.5	Otras	80
1.5.4	Sentencias de Control	80
1.5.4.1	Sentencia If	80
1.5.4.2	Sentencia SELECT CASE	81
1.5.5	Bucles	82
1.5.5.1	Uso de bucles Do	82
1.5.5.2	Uso de While ... Wend	83
1.5.5.3	Uso de For ... Next	83
1.5.5.4	Uso de For Each ... Next	84
1.5.6	Procedimientos y Funciones	84
1.5.6.1	Parámetros	85
1.6	Objetos implícitos ASP y componentes ASP	85
1.7	Objetos Application y Session	86
1.8	Global.asa	88
1.8.1	Application_OnStart	89
1.8.2	Application_OnEnd	89
1.8.3	Session_OnStart	89
1.8.4	Session_OnEnd	89
1.9	Response y Request	90
1.9.1	Caché y caducidad de la página	90
1.9.2	Recibir información de formularios: Métodos GET y POST	92
1.9.3	Trabajar con cookies	93
1.10	Objeto Server	95

1.10.1	HTMLEncode	95
1.10.2	Acceso a bases de datos	95
1.11	Seguridad desde el nivel de programación	102
2.	Ejemplo de codificación ASP utilizado en el sitio	104
2.1	Codificación de página de captación	104
2.2	Codificación de página de resultados	106
XI.	Prueba	110
1.	Resultados de la página de captación	110
2.	Prueba de página de resultados	111
XII.	Implementación	112
1.	Internet Information Server	112
1.1	Configuración	115
1.2	Nuevos Webs	129
1.3	Instalación de las extensiones de Front Page	132
1.4	Directorios Virtuales	135
1.5	Servidor Virtual	141
1.6	Seguridad	141
2.	Alojamiento del sitio Web	144
2.1	Plan de Alojamiento	144
2.2	Datos del dominio adquirido	146
XIII.	Conclusiones	148
XIV.	Recomendaciones	149
XV.	Glosario	150
XVI.	Anexos	153
A.	Pantalla de Presentación	154
B.	Pantalla de Consulta de Animales	155
C.	Enlace Etnias	156
XVII.	Bibliografía	157

I. INTRODUCCION

1. Generalidades de la Reserva de Biósfera de Bosawás



BOSAWÁS, la primera Reserva de Biósfera de Nicaragua, comprende el 14% del territorio nacional, aproximadamente 2,000,000 hectáreas (20,000 Km²). La legislación pertinente para la Reserva de la Biósfera BOSAWÁS diferencia un área núcleo y una zona de amortiguamiento. Su área núcleo consta de 800,000 hectáreas, espacio de vida de las etnias Mayangna y Miskitu, y su área de amortiguamiento de

1,200,000 hectáreas, habitada sobre todo por mestizos.

Esta Reserva de Biósfera cubre 2,181,500 hectáreas en el norte del país, contiguas a las fronteras de Honduras. Compone una constelación de áreas protegidas ya existentes. Es un área muy serrana, cubierta por bosques tropicales húmedos, pluvioselvas y sabanas de pino. Junto con la Reserva de Biósfera de Río Plátano en Honduras, conforma la mayor extensión de bosques protegidos en América Central.

Su biodiversidad, su importancia como moderador del clima, el hecho de constituir el hábitat para cientos de especies en peligro de extinción o bien especies endémicas, atribuyen a la Reserva de Biósfera de Bosawás una inigualable importancia en términos ambientales que trasciende hasta lo social y, en ocasiones, hasta lo económico.

La Reserva de BOSAWÁS se creó por Decreto Ejecutivo (44-91) en el año 1991 sin ser consultado con los pueblos indígenas que viven en ella, siendo la razón para ello la



amenaza que existía de convertir esta área en un polo de desarrollo agropecuario. Se quería pues defender uno de los bosques tropicales más intactos de Centroamérica de las políticas de Reforma Agraria y Reinserción de repatriados. La intención era en ese entonces asentar en esta zona a miles y miles de nicaragüenses, que durante la guerra vivieron en Honduras y eran miembros de la Resistencia Nicaragüense. Ante esta amenaza fue que el entonces Director del Instituto de Recursos Naturales manifestó a la Presidencia de la República su preocupación, encargándosele a él de elaborar un borrador para el decreto de Ley, que al término de escasamente una semana se acepta. Este nacimiento poco participativo, el cual con razón ha causado y causa críticas de varios sectores de la sociedad.

1.1 Bosawás y el entorno social

El hombre - en su afán de alcanzar el perfeccionamiento de la producción, de la industria y de la tecnología - ha olvidado las perspectivas de un futuro altamente incierto en términos de recursos naturales disponibles para que los habitantes de tiempos venideros puedan tener una esperanza de vida aceptable. Como consecuencias de las acciones realizadas en la búsqueda del desarrollo macroeconómico pueden citarse las problemáticas del deterioro del medio ambiente y la pobreza extrema que sufren los países tercermundistas entre los que se clasifica el nuestro: Nicaragua.

Lo anterior, ha influido de tal manera que algunas de las reservas de bosques más importantes de Nicaragua estén en peligro latente de destrucción debido al incumplimiento de las normativas impuestas (pero no ejecutadas) por el gobierno y por la falta de un consenso nacional que abogue por estos problemas de índole ambiental.

BOSAWÁS no es inmune a esta serie de problemas, por lo que se hace necesario crear conciencia en la población sobre su importancia, pero sobre todo, sobre la necesidad de su protección.



1.2 Problemáticas de la Reserva de Bosawás

Como una de las principales problemáticas debe destacarse indiscutiblemente el avance agropecuario que se ha venido dando a partir del año 1990 en la zona del norte del país. A tal punto que muchos pobladores inmigrantes de las regiones céntricas del país se han asentado en las zonas aledañas de la Reserva, poniendo en riesgo de esta manera la zona perimetral de la misma. Este avance descontrolado - en longitud y rapidez - pone en peligro inminente la integridad de la Reserva por las consecuencias catastróficas que se han manifestado hasta el momento y las futuras que aún no se perciben.

Es de suma importancia destacar algunos problemas específicos, como por ejemplo *la tala de los bosques*, que conlleva el avance de la frontera agrícola. Esto debido a la necesidad de los pobladores de expandir sus áreas de cultivo y de pastoreo para suplir sus elementos básicos de subsistencia. Este problema es perjudicial si tomamos en cuenta el hecho de que no hay una explotación racional, mucho menos organizada de los recursos forestales, lo que desembocaría en la posible disminución del número de especímenes de algunas especies forestales.

Esta extracción maderera no es legal en diversas ocasiones. Sin embargo, en la mayoría de los casos ésta no es penalizada de acuerdo a las reglamentaciones establecidas, e incluso es encubierta por las autoridades responsables de ejercer el cumplimiento de las leyes que amparan el uso racional de los recursos de la Reserva.

Otra problemática que es obligatorio mencionar, y a la vez es consecuencia del avance de la frontera agrícola, es el de los *incendios forestales*. Más aún en los últimos años cuando el índice de crecimiento de los mismos ha incrementado de manera alarmante, de tal manera que se han devastado inmensas extensiones de bosques y se ha alterado en gran magnitud el balance ecológico de la Reserva.



Agregado a los problemas mencionados anteriormente podemos mencionar el hecho de que se han concesionado extensiones de tierra para la *explotación minera*, la cual - a menudo - tiende a ser destructiva por el hecho de dejar al descubierto las zonas de explotación, las que antes eran bosques de abundante vida natural. Estos suelos desprovistos de vegetación son víctimas de una erosión inevitable causada por factores eólicos o hídricos que arrasan con los mismos.

Por último, es necesario mencionar otro tipo de problemática que se ha presentado en la Reserva, más que todo de tipo social, el *narcotráfico*. Esto debido a que la lejanía de la Reserva y el difícil acceso a ella proporcionan un elemento de comodidad relativa para los que se dedican a este tipo de actividad delictiva. Es por lo tanto, una problemática que debe combatirse con el cumplimiento de la ley para evitar así el crecimiento del narcotráfico en esta región.



II. ANTECEDENTES

Debido a la importancia de esta Reserva, considerada desde hace tiempo como el pulmón de Centroamérica, hemos decidido desarrollar una página web que sirva como un sitio de información sobre dicha reserva. Este proyecto es novedoso en el concepto de la implementación de acceso a datos por medio del Internet, pero además tratamos de poner la tecnología al servicio de una causa justa y necesaria: la protección de los recursos naturales.

Ya existen sitios que brindan información acerca de la Reserva de Biósfera de Bosawás, sin embargo, no existe un sitio que sea orientado a una documentación científica y actualizada sobre la Reserva misma. Además, este sitio pretende una accesibilidad total al estar alojado en un servidor para Internet, por lo que será accesible para todos los nicaragüenses y cualquier persona conectada a Internet en el mundo.

Publicar datos sobre las especies y generalidades de este patrimonio permitirá a muchos de los interesados en esta Reserva conocer los distintos ámbitos que presenta, su importancia, sus problemas actuales y a la vez proveerse de información sobre las especies que la habitan.

Desde el punto de vista informático, hemos encontrado a lo largo de nuestra investigación, previa al desarrollo del proyecto, algunos trabajos que abordan la integración de las bases de datos en el Internet. No obstante, hemos optado por realizar un proyecto tangible y verificable que verdaderamente se lleve a buen término y completo funcionamiento, que sirva al final como una referencia documental sólida y confiable con relación al desarrollo de sitio web con acceso a datos.

La fase investigativa ha sido larga. Nos hemos basado en abundante información del Internet con relación al aspecto del acceso a datos por medio del Internet, también en bibliografía que logramos obtener sobre el tema. Para la parte informativa de la Reserva,



hemos utilizado artículos, información que nos han brindado distintas organizaciones y trabajos realizados previamente.

Tratando, entonces, de tomar lo que nos parece más adecuado de los proyectos investigados, incluyendo nuestras ideas y basándonos en una metodología fiable es que tomamos el rumbo hacia el desarrollo del sitio web con acceso a datos para las especies de la Reserva de Biósfera de Bosawás.



III. JUSTIFICACIÓN

BOSAWÁS es la Primera Reserva de Biósfera de Nicaragua. La de mayor extensión a nivel centroamericano y alberga a cientos de especies en su interior; sin embargo es innegable el peligro que corre ante el despale indiscriminado, el avance de la frontera agrícola y los frentes de colonización en sus alrededores.

Todo esto traerá consecuencias tarde o temprano a nuestra vida, ya sea por variaciones de clima, una menor producción de oxígeno a nivel centroamericano y la extinción de especies animales y vegetales. Las causas de toda esta problemática son difíciles de prevenir y solucionar, sin embargo debe considerarse la falta de información sobre la importancia de la Reserva como una de ellas.

Por los aspectos considerados anteriormente, nos hemos dado a la tarea de la construcción de un sitio Web que brinde información sobre Bosawás, tanto de su importancia como de las especies animales y vegetales que existen dentro de esta Reserva.

Nosotros aceptamos esta misión, poniendo de esta manera nuestra pequeña contribución en pro de la conservación de este Patrimonio que como nicaragüenses tenemos el deber de conocer y proteger por el bien de las generaciones futuras.

El compromiso que adquirimos comprende esfuerzo y dedicación. Lo aceptamos porque es necesario para contribuir, tal vez en pequeña escala, a nuestra patria, a nuestra cultura y a nuestra gente.



IV. OBJETIVOS

General:

- Diseñar un sitio Web para publicar información sobre la Reserva de Biósfera de Bosawás y permitir el acceso restringido a datos sobre las especies de dicha Reserva.

Específicos:

- Construir un sitio Web atractivo para el alojamiento de información descriptiva de la Reserva de Biósfera de Bosawás.
- Diseñar una base de datos para el almacenamiento de información sobre las especies de la Reserva.
- Desarrollar consultas que permitan el acceso a los datos de la base de datos por medio del Internet.
- Proporcionar elementos de animación para mayor entretenimiento del usuario.
- Desarrollar un documento final que sirva como referencia de diseño de sitios Web y cómo implementar el acceso a datos por medio del Internet.



V. MARCO TEÓRICO

Al entrar en el área del medio ambiente, tenemos que abordar inexorablemente algunos conceptos fundamentales de los seres vivos que servirán más adelante para la comprensión del modelado de datos que se ha llevado a cabo con las especies de la Reserva. Por tanto, hemos dividido este marco teórico en dos partes que nos permitan entender de manera organizada el ámbito de desarrollo del proyecto.

El primer aspecto a abordar serán los *fundamentos biológicos* necesarios para la comprensión, desde el punto de vista de la Biología, de la clasificación de los seres vivos, la importancia del equilibrio ecológico y la fragilidad del mismo, entre otros conceptos cuya comprensión es obligatoria.

En la segunda parte abordaremos lo referente a los *fundamentos informáticos*. Se proveerá de los conceptos básicos sobre bases de datos que permitirán conocer de manera más específica el funcionamiento y desempeño del sitio web.

1. Fundamentos biológicos

1.1 Relaciones Biológicas

A primera vista, el mundo de las cosas vivas se ve formado por una gran variedad de plantas y animales, todos muy distintos, siguiendo sus caminos a su propio paso. Sin embargo, el estudio detallado revela los mismos problemas de alimentarse para obtener energía, ocupar espacio vital y producir nuevas generaciones. Además, cada uno debe actuar con las otras especies en una variedad de formas que aumentan o disminuyen la probabilidad de supervivencia.

En la solución de esos problemas, planta y animales se han diferenciado en formas infinitas, cada una adaptada a la vida de un medio en particular, la adaptación no sólo se ha relacionado con el ambiente físico (tolerancia hacia cierto grado de humedad, viento, sol, temperatura, gravedad y otras circunstancias variables), sino también con el medio biológico de la misma región. El estudio de las relaciones de los seres vivos con el medio en que se desenvuelven, físico y biológico, se llama ecología.

En años recientes el público ha comenzado a darse cuenta de los terribles problemas planteados por la ignorancia pasada y nuestro desinterés por los principios de la ecología. Nuestra preocupación por la creciente contaminación del aire y el agua es alentadora y sugiere que finalmente se darán los pasos apropiados para aliviar estos problemas y prevenir su recurrencia. No obstante, la contaminación es sólo una parte de nuestra situación ecológica actual, que tiene una base mucho más ancha. El hombre debe aprender finalmente a mantener su población dentro de ciertos límites y a vivir en este planeta sin alterar el medio ambiente físico y biótico de tal modo que ponga en peligro la existencia suya y de otras 1,700,000 clases de plantas y animales.

1.2 Clasificación de los seres vivos

Para tratar con estas múltiples formas de vida y describir sus características, los biólogos tuvieron primero que nombrarlas y clasificarlas. Por ejemplo, los animales fueron clasificados por San Agustín en el siglo IV en tres grupos: útiles, dañinos y superfluos, desde luego para el hombre. Los botánicos de la edad media clasificaban las plantas en función de si producían frutas, vegetales, fibra o maderas. La clasificación de las plantas por semejanzas estructurales fue establecida sobre bases sistemáticas firmes por el biólogo Carl von Linné o Linneo. Catalogó y describió las plantas en *Species Plantarum* (1753) y los animales en *Systema Naturae* (1758). Al parecer la teoría de la evolución, los taxonomistas han intentado presentar sistemas de clasificación basados en relaciones naturales, de manera que se reunieran en un mismo grupo los organismos que presentan cierta relación en su origen evolutivo. Puesto que muchas de las semejanzas estructurales



dependen de relaciones de evolución, la clasificación moderna de los organismos es en muchos puntos semejante a la de Linneo, basada en similitudes estructurales lógicas.

La unidad de clasificación para las plantas y animales es la *especie*. Es difícil dar una definición de esta palabra que puede aplicarse a los reinos animal y vegetal; pero la especie puede definirse como un grupo de individuos semejantes en cuanto a características estructurales y funcionales, que en la naturaleza solo se reproducen entre sí y tienen un antecesor común.

Las especies vecinas se agrupan en la siguiente unidad superior de clasificación, el *género*. Los nombre científicos de plantas y animales se escriben con dos palabras, el género y la especie, en latín. Este sistema para nombrar los organismos, llamado sistema binominal, fue usado por primera vez por Linneo. De acuerdo con él, el nombre científico del gato doméstico, *Felis domestica*, se aplica a todas las variedades de los gatos (de Persia, siameses, Manz, de Abisinia e incluso los corrientes), puesto que todos pertenecen a la misma especie y pueden cruzarse entre sí. Las especies afines del mismo género son *Felis leo*, el león; *Felis tigris*, el tigre y *Felis pardus*, el leopardo. El perro, que pertenece a un género diferente, se llama *Canis familiares*. Nótese que en todos estos nombres, el género se escribe primero, empieza con mayúscula, en tanto el nombre de la especie viene después y no lleva mayúscula (algunos nombres de especies de plantas sí tienen mayúscula). El empleo del latín en lugar de un idioma moderno para dar los nombres de las especies viene de la época en el que el latín era el idioma internacional de la ciencia.

Así como varias especies pueden agruparse para formar un género, varios géneros semejantes forman una *familia*, y a su vez, las familias pueden agruparse en *órdenes*, estos en *clases*, y las clases en *divisiones* (plantas) o *filos* (singular filo) (animales). Los filos, por tanto, son las principales divisiones grandes de los *reinos* animal y vegetal, así como las especies representaban unidades pequeñas fundamentales. Por ejemplo, la clasificación completa del roble blanco es división *Magnoliophyta*, clase *Magnoliopsida*, orden *Fagales*, familia *Fagaceae*, género *Quercus*, especie *alba*. El hombre es un miembro de filo



Chordata, subfilo *Vertebrata*, clase *Mammalia*, subclase *Eutheria*, orden *Primates*, familia *Hominidae*, género *Homo*, especie *sapiens*.

Muchas plantas y animales se encuentran en grupos naturales fáciles de reconocer y de clasificación fácil, pero otras formas, que parecen encontrarse sobre la divisoria entre dos grupos cuyas características comparten, resultan difíciles de colocar en cualquiera de ambos. La cantidad y extensión de los distintos grupos varía según la clasificación empleada y el criterio del investigador quien autoriza la clasificación. Algunos taxonomistas gustan de agrupar las cosas en unidades ya presentes; otros prefieren categorías separadas para las formas que no encajan fácilmente en una de las clasificaciones conocidas. En general, los distintos taxonomistas consideran que hay de 10 a 33 filos animales y de cuatro a 12 divisiones vegetales.

Asimismo, existen controversias por establecer el número de reinos en que se clasifican los seres vivos, que tradicionalmente hemos clasificado en animal y vegetal, pero que a partir investigaciones detalladas han causado polémica debido a que hay organismos de los que difícilmente puede distinguirse su pertenencia a uno de estos reinos.

1.3 Equilibrio Ecológico

Los seres que viven en un ecosistema están adaptados a las condiciones físicas del lugar donde habitan. Se relacionan unos con otros mediante las cadenas alimenticias, que a su vez, están unidas entre sí. Para que el ecosistema pueda ofrecer las condiciones adecuadas a quienes existen en él, todas estas relaciones deben mantenerse en equilibrio en su interior. Es lo que denominamos como *equilibrio ecológico*. Este se da cuando las poblaciones permanecen constantes, es decir, el número de individuos que nace y se desarrolla está en perfecta armonía con el número de individuos que muere.

Si por algún motivo, el equilibrio de una cadena alimenticia se rompe, ello repercute de inmediato en el equilibrio de todo el ecosistema. El equilibrio ecológico se puede romper o



perder cuando alguna especie animal o vegetal de la comunidad desaparece. Las causas de este hecho pueden ser muy variadas e ir desde cambios bruscos en el clima hasta la intervención directa del hombre en el ecosistema.

De todas las especies, es precisamente el ser humano quien más influye en el entorno y lo modifica con sus actividades, debido a su necesidad de sobrevivir. Debido a lo anterior, durante los últimos años el hombre se ha cuestionado su relación con el medio ambiente y ha buscado maneras para no continuar dañando la naturaleza.

Por eso, la ecología -aunque es una ciencia nueva- ha cobrado enorme importancia, y desde las naciones más desarrolladas se ha impuesto darle su debido lugar en todo el mundo.

2. Fundamentos Informáticos

2.1 Bases de Datos: Conceptos Fundamentales

Una base de datos es un conjunto de información guardada de forma persistente (los datos no se borran hasta que nosotros los borremos), estructurada (cada una de las entidades que deseamos almacenar se guarda en estructuras lógicas separadas, llamadas tablas) y flexible (podemos añadir un nuevo atributo - llamado campo en el lenguaje de las bases de datos - a una tabla de manera fácil).

Las bases de datos existen para dar servicio a las necesidades de las *aplicaciones*. Una aplicación, a su vez, es una unidad de software que existe para dar servicio a las necesidades de alguna *actividad*: un sistema de negocios, un juego, un servicio público, un anuncio o cualquier proceso del mundo real que pueda imaginar. En un mundo perfecto las necesidades de la actividad determinan los requisitos de la base de datos.

La misión de un diseñador de bases de datos es, en primer lugar, implementar todos los requisitos derivados de la aplicación y, en segundo, evitar imponer requisitos contrarios (es decir, restricciones) a la aplicación o la actividad.

La idea básica de una base de datos es la de guardar datos de una forma organizada. Esto tiene dos beneficios. En primer lugar, los datos están disponibles instantáneamente para diversos usos en su empresa. En segundo lugar, como la base de datos tiene una estructura conocida, el sistema que la almacena puede proporcionar herramientas potentes para extender su uso. Si nunca ha empleado bases de datos, puede que se sorprenda de lo rápido que estas sencillas ideas pueden complicarse.

2.2 Bases de datos Relacionales

Aunque las bases de datos físicamente son archivos de una computadora, las aplicaciones que utilizan bases de datos no manejan estos archivos directamente. En su lugar, las aplicaciones envían comandos y reciben respuestas de un *sistema de administración de base de datos* (SABD). SQL Server y Access son ejemplos de sistemas de administración de bases de datos. Un SABD libera al programador de muchas tareas tediosas y complicadas implicadas en el procesamiento de la base de datos y protege frente a la introducción de datos no válidos.

Casi todas las bases de datos modernas se ajustan al llamado *modelo relacional* (este modelo utiliza un grupo de tablas, para representar los datos y las relaciones entre ellos. Cada tabla esta compuesta por varias columnas, y cada columna tiene un nombre único). Esto es lo que les convierte en *bases de datos relacionales*.

Una base de datos relacional consiste en un conjunto de *tablas*, a cada una de las cuales se le asigna un nombre exclusivo. Cada tabla tiene una estructura. Cada fila de la tabla representa una relación entre un conjunto de valores.



2.3 Bases de Datos en el web

La utilización de la World Wide Web para presentar y acumular datos se ha desarrollado mucho más allá de la sencilla presentación de páginas. Están muy lejanos los días en que los diseñadores web que necesitaban presentar los elementos de una colección creaban una página independiente para cada uno de ellos. Dichas páginas eran difíciles de organizar inicialmente y era casi imposible mantenerlas actualizadas con el paso del tiempo.

Cada vez más el volumen y la estructura de los datos que se presentan en la web permite su almacenamiento y organización en bases de datos, y después la generación de páginas web basadas en esas bases de datos. La gama de sitios que pueden beneficiarse de este enfoque es asombrosa. Cualquier sitio web (o parte del mismo) que presente información sobre un conjunto de elementos similares es candidato para la utilización de una base de datos web. Estos elementos similares pueden ser productos para vender, fotografías personales, hipervínculos o casi cualquier cosa que se le ocurra.

La solución general consiste en definir una base de datos (quizá en Microsoft Access), añadir un registro para cada elemento y después consultar dicha base de datos para generar páginas web sobre la marcha. Una página de menú en HTML común se convierte en una consulta a una base de datos de varios registros. Una página subordinada se convierte en una presentación de detalle de un registro de la base de datos. Otras ventajas son:

- La base de datos es mucho más fácil de mantener que todas las páginas web individuales.
- El empleo de una base de datos facilita la búsqueda de los elementos deseados.
- Una base de datos facilita la presentación de los mismos datos de distintas formas: por categorías, por edad o por cualquier otro campo de la base de datos.

El atractivo de difundir o recoger información mediante páginas web es universal. No importa si su empresa es grande o pequeña, si es algo personal, gubernamental, educativo o



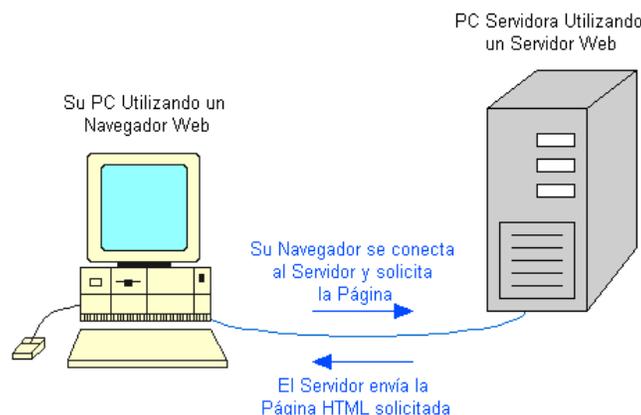
personal, o si tiene ánimo de lucro o no lo tiene. Se aplican las mismas técnicas tanto si los elementos son antigüedades, coches usados, sellos raros, obras de arte, casas en venta o niños perdidos. Si piensa que crear una página web estática para cada uno de un montón de elementos resulta tedioso, propenso a errores y difícil de organizar y mantener, las páginas web con bases de datos son para usted.

Por desgracia (a pesar de más de veinte años de intentos), nadie ha encontrado todavía la forma de construir aplicaciones de bases de datos utilizando la tecnología WYSIWYG o de apuntar y hacer clic. En el mejor de los casos, los sistemas de desarrollo automatizan sólo un subconjunto de los requisitos para un tipo de aplicación específico. Para desarrollar aplicaciones con base de datos de cualquier tipo (web, cliente/servidor o para gran computadora) el código es el rey supremo.

3. Funcionamiento del Internet

Digamos que usted se sienta frente a su computador para Navegar en el Web y recibe una llamada de un amigo(a) que le dice: "¡Leí un gran artículo, teclea esta URL y cerciórate!: <http://www.geocities.com/SunsetStrip/Amphitheatre/5064/INTERSERV.HTML>. Entonces tecleará la URL en su Navegador y presionará Entrar. Mágicamente, sin importar en qué lugar del mundo exista tal URL, la página aparecerá en su pantalla.

Desde el nivel más básico posible, el siguiente diagrama muestra qué pasa:



Su Navegador genera una conexión a un Servidor Web, solicitará una página y la recibe. Si desea profundizar más, he aquí los pasos básicos que ocurren detrás de cámaras:

- El Navegador divide la URL en 3 partes:
 - a. El protocolo ("http").
 - b. El nombre del servidor (www.geocities.com).
 - c. La ruta (/SunsetStrip/Amphitheatre/5064/) hasta el archivo de destino (INTERSERV.HTML).
- El navegador se comunica con otro servidor que traduce el nombre (www.geocities.com) a números, llamados *dirección IP* para que así se pueda conectar a la máquina destino.
- Entonces el Navegador crea una conexión al servidor de la dirección IP en el puerto 80.
- Siguiendo en protocolo HTTP, el Servidor envía propuesta de obtención para encontrar el archivo "INTERSERV.HTML".
- El servidor envía el texto HTML de la página al Navegador. El Navegador lee los comandos HTML y muestra la página en la pantalla.

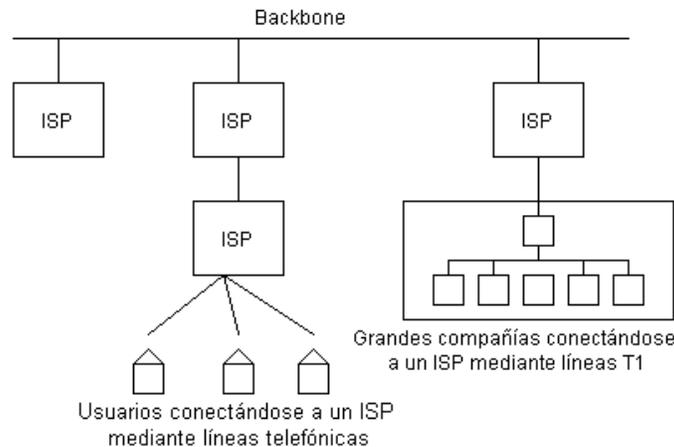
Por si lo ha notado, hay gran cantidad de vocabulario. Para entender todo el proceso en detalle, debe relacionarse con dirección IP, puertos, protocolos, etc. Además de esto nos veremos forzados a utilizar varios anglicismos, (ejemplo: ISP, Backbone) ya que ayudan a relacionarse mejor.

Pero, ¿qué es Internet? El Internet es una gigantesca colección de millones de computadoras que están unidas mediante una *Red Computacional*, también llamada *Network*. Esta red permite que todas las computadoras se comuniquen entre sí. Un computador casero es conectado usualmente a Internet utilizando una línea telefónica normal y un Módem que se comunica con un *ISP* (Internet Service Provider, o proveedor de servicios de Internet). Un computador de empresa o universidad posee un *NIC* (Network Interface Card, o tarjeta para interfase en red) que lo conecta directamente a una *LAN*



(Local Area Network, o red de área local) dentro de la empresa. Toda la entidad conecta su LAN a un ISP utilizando una línea telefónica de alta velocidad como por ejemplo una *línea T1* (una línea T1 puede manejar aproximadamente 1.5 millones de bits por segundo, mientras que una línea telefónica normal usando un módem debe ser capaz de manejar de 30,000 a 50,000 bits por segundo).

Los ISP se conectan a otros ISP más grandes, y éstos mantienen conexiones de fibra óptica llamados "backbones" (backbone significa columna vertebral) para una nación o región. Los backbones están conectados alrededor del mundo mediante cables submarinos o conexiones satelitales. De esta forma cada computador en Internet está conectado con los demás.



3.1 Direccionamiento IP

Para mantener a todas esas máquinas en orden, a cada máquina en Internet se le asigna una dirección única llamada dirección IP. Esta consta de Números de 32-bits expresados normalmente en 4 octetos en un número decimal con puntos. Una dirección IP típica podría ser algo como esto: *209.1.224.6*.

Los cuatro números en la dirección IP se llaman octetos, porque pueden tener valores entre el 0 y el 255 (2^8 posibilidades por octeto).



Cada máquina en Internet tiene una dirección IP única. Un servidor tiene una dirección IP que no cambia muy a menudo. Una máquina casera que se conecta a través de un módem a veces obtiene una dirección IP que es asignada por el ISP en el momento de la conexión. Esa dirección IP es única para su sesión pero podría ser diferente para la próxima vez que se conecte. De esta forma un ISP sólo necesita una dirección IP para cada módem de usuario.

Si usted está trabajando en una máquina Windows puede visualizar su actual dirección IP con el comando WINIPCFG.EXE. En una máquina UNIX teclee *ifconfig* para mostrar la dirección IP, y entonces escriba “*exit*” para salir del comando. Para que las máquinas en Internet funcionen, todo lo que se necesita es una dirección IP para poder "hablar" con el servidor. Por ejemplo, en su navegador o *browser* puede escribir la URL: *http://200.21.200.2* y llegará a uno de los servidores de mi ISP.

3.2 Nombres de Dominio

Debido a que los seres humanos a veces tienen problemas para recordar direcciones IP, y debido a que las direcciones IP a veces necesitan cambiar, todos los servidores en Internet también tienen nombres que son más entendibles para los humanos llamados *nombres de dominio*. Por ejemplo, *www.geocities.com* es un nombre permanente. Es más fácil para la mayoría de la gente recordar *www.geocities.com* que su equivalente en números.

El nombre *www.geocities.com* tiene 3 partes:

- a. El tipo de servicio (“*www*”).
- b. El nombre de dominio (“*geocities*”).
- c. El tipo de entidad (“*com*”).

Los nombres de dominio son manejados por una compañía llamada *InterNIC*. Su tarea primordial es crear nombres para tipos de entidades y garantizar que todos los nombres de



dominio son únicos. El nombre es creado por la compañía que maneja el servicio. “www” es una palabra muy común, pero en muchos lugares lo omiten o reemplazan con otro por ejemplo: *internet.telecom.com.co*.

Para transformar las direcciones IP a nombres, se utilizan un grupo de servidores llamados *Domain Name Servers* (DNS). Estos servidores tienen simples bases de datos que transforman las direcciones IP; estos están distribuidos por todo Internet. Si usted teclea la URL <http://www.geocities.com/SunsetStrip/Amphitheatre/5064/interserv.html> en su Navegador, este fracciona el nombre “www.geocities.com” y el Servidor DNS regresará la dirección IP correspondiente.

En una máquina UNIX usted puede obtener el mismo servicio utilizando el comando *nslookup*. Simplemente teclee el nombre www.geocities.com en el comando.

Ahora sabe que Internet está conformada por millones de máquinas, todas con una dirección IP única. Muchas de esas máquinas son máquinas servidoras que dan servicios a otras máquinas. Tal vez se ha relacionado con muchos de esos servidores: servidores de correo, servidores Web, servidores FTP, servidores Gopher, servidores Telnet, etc. Una máquina servidor no es más que una que suministra servicios a otras máquinas.

3.3 Puertos

Cualquier máquina servidor proporciona sus servicios a Internet utilizando *puertos numerados*, para cada servicio de que dispone el servidor. Por ejemplo, si una máquina servidor maneja un servidor Web y un servidor FTP, el servidor Web estará disponible típicamente en el puerto 80, y el servidor FTP estará disponible en el puerto 21. Los clientes se conectan a un servicio con una dirección IP específica y en un número de puerto específica.



Cada uno de los servicios es disponible a un “número de puerto conocido”. He aquí algunos de los números de puertos más conocidos:

- Daytime: 13
- FTP: 21
- Telnet: 23
- SMTP (Simple Mail Transfer, para correo): 25
- Gopher: 70
- Finger: 79
- WWW: 80

Si la Máquina servidor acepta conexiones en un puerto desde el mundo exterior y si un firewall no está protegiendo el puerto, usted se puede conectar a ese puerto y utilizar el respectivo servicio. Por ejemplo, un servidor Web debe estar en el puerto 80. Si usted configura su propia máquina e instala software para servidor Web en él, podría colocarse el servicio Web en el puerto 918 (o cualquier otro puerto libre) si lo desea. Si su máquina es conocida como: *www.yyy.com*, alguien podría conectarse a usted con la URL: *http://www.yyy.com:918*. EL “:918” especifica el número de puerto. Cuando el puerto no es especificado, el Navegador asume que el servidor utiliza el conocido puerto 80.

3.4 Protocolos

Una vez que un cliente se ha conectado a un servicio en un puerto específico, accede a él utilizando un protocolo específico. El *protocolo* es la forma pre-definida que se utiliza para “hablar” con el servicio. Los protocolos son generalmente de texto, y simplemente describen la forma en que un cliente y un servidor tendrán su conversación. Quizás el protocolo más simple es el protocolo *Daytime*. Si se conecta mediante el puerto 13 a una máquina que maneje un servidor Daytime, ésta enviaría datos de la fecha actual y la hora, entonces cerraría la conexión. El protocolo es: “si te conectas a mí, te enviaré la fecha y hora y me desconectaré”. La mayoría de máquinas UNIX manejan este tipo de servidor. Si



desea, puede probar esto conectándose a una máquina mediante el Telnet. Una sesión UNIX sería algo como esto:

```
%telnet www.geocities.com 13  
Trying 209.1.224.61...  
Connected to www.geocities.com.  
Escape character is '^]'.  
Sun Jan 12 08:34:06 1999  
Connection closed by foreign host.
```

En este ejemplo, www.geocities.com es una máquina UNIX (en el ejemplo, claro está) y 13 es el puerto del servicio Daytime. La aplicación Telnet se conecta al puerto 13 (el Telnet se conecta por defecto al puerto 23, pero esto puede cambiarse). El servidor envía la fecha y la hora y se desconecta. La mayoría de versiones de Telnet le permiten especificar un número de puerto, así que puede tratar por usted mismo utilizando cualquier versión de Telnet que tenga disponible en su máquina.

El comando básico que un servidor *HTTP* es *GET*, si se conecta a un servidor que entiende el protocolo HTTP y le dice “GET filename”, el servidor responderá enviando el contenido del archivo especificado y desconectándose. He aquí una sesión típica:

```
%telnet www.geocities.com 80  
Trying 209.1.224.61...  
Connected to geocities.com.  
Escape character is '^]'.  
GET http://www.geocities.com/  
...  
Connection closed by foreign host.
```



En el protocolo original HTTP todo lo que usted envió fue el nombre de archivo (con su respectiva ruta, si existe), en este caso: “/”, o:

“/SunsetStrip/Amphitheatre/5064/INTERSERV.HTML”.

El protocolo fue modificado después para permitir el envío de la URL completa. Esto permite que existan compañías que ofrecen servicios de *dominios virtuales* donde muchos dominios habitan en una misma máquina, para utilizar una dirección IP para todos.

4. Servidores Web

Usted puede ver de esta descripción que un servidor Web puede ser una pieza simple de software. Sólo toma el archivo especificado con el comando GET, y lo envía al servidor. Incluso usted puede crear su propio código para generar su propio servidor Web con alrededor de 500 líneas de código en un lenguaje de programación como el C. Obviamente, un servidor de nivel empresarial es muy diferente, pero los principios básicos son los mismos.

La mayoría de servidores añaden algún nivel de seguridad a sus tareas. Por ejemplo, si usted ha ido a alguna página y el navegador presenta una ventana de diálogo que pregunta su nombre de usuario y contraseña, ha encontrado una página protegida por contraseñas. El servidor deja que el dueño o el administrador del servidor mantenga una lista de nombres y contraseñas para las personas a las que se les permite ver la página, y el servidor deja que sólo esas personas quienes saben la contraseña tengan acceso. Los servidores más avanzados añaden seguridad para permitir una conexión encriptada entre el servidor y el navegador para que información de suma importancia como números de tarjetas de crédito puedan ser enviados por Internet.



Existen varias opciones para el montaje de un servidor web tanto para la plataforma Windows como *Unix*. Dos de los más reconocidos son *Apache Web Server* e *Internet Information Server* (al cual se le dedicará una sección completa más adelante).

Apache es un servidor web desarrollado para entornos Unix y sobre todo para *Linux* y es el más difundido en Internet debido a su buen comportamiento y funcionamiento.

Apache puede conseguirse en multitud de sitios, pero por citar alguno citaremos www.apache.org que es la página oficial de Apache. Allí no sólo se obtiene el propio apache, sino que además viene toda la documentación necesaria para instalarlo y ponerlo en funcionamiento. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan:

- Multiplataforma.
- Es un servidor de web conforme al protocolo HTTP/1.1.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

VI. MATERIALES

1. Hardware:

- Computadora con los siguientes requisitos mínimos:
 - Procesador Pentium II Intel.
 - 128 MB de RAM
 - Velocidad de 200 MHz.
 - Unidad de CD-ROM para lectura y escritura.
 - Disco duro 10GB.
 - Multimedia
 - Tarjeta de Red 10/100 Mbits/sec
- Impresora láser con capacidad de impresión de gráficos.
- Escáner

2. Software:

- Sistema Operativo: Windows 2000 Server.
- Internet Information Server 5.0: Servidor Web.
- Editor de Páginas Web (Front Page XP, Edit Plus versión 2.0).
- Procesador de texto Microsoft Word XP.
- Microsoft Access 97 como Sistema Gestor de Base de Datos.
- Macromedia Flash MX
- Macromedia Fireworks MX
- Swish V2.0

3. Otros:

- Conexión a Internet
- Espacio para el alojamiento de la página
- Unidades de Almacenamiento (Unidades de CD R, Unidades de CD RW, Unidades de Disquete, etc.).



VII. METODOLOGÍA

La construcción del sitio web para las especies de la Reserva de Biósfera de Bosawás nace de la ambición de tres jóvenes que decidieron aplicar sus conocimientos en algo útil, tangible y de trascendencia.

Amparados en ese empuje en desarrollo fuimos moldeando la idea, transformándola en proyecto, a tal punto que tuvimos la oportunidad de aplicarle al proceso una metodología científica en lo que al desarrollo del software concierne.

El método a utilizar más apropiado era, ineludiblemente, el de prototipo. Éste método permite la construcción de un modelo de la aplicación a desarrollar. Este modelo describe la interacción hombre-máquina, de forma que facilita proyectar la comprensión de cómo se llevará a cabo dicha interacción.

Para la construcción del prototipo se debe de establecer un conjunto de directrices a seguir, las cuales permitirán un desarrollo ordenado del modelo. En nuestro caso, hemos aplicado las fases de Análisis, Diseño, Codificación y Prueba para cada prototipo construido. Esto nos lleva, por deducción, que la metodología empleada es la de *Construcción de Prototipos en combinación con el ciclo de Vida Clásico*. Esto último por las fases aplicadas a la construcción de cada prototipo.

A lo largo del desarrollo del proyecto hemos construido un total de cinco prototipos para poder obtener un sistema final que satisface los requisitos especificados. En este documento, se muestran entonces los resultados de las fases aplicadas al prototipo final.



VIII. ANÁLISIS DEL SITIO WEB

En la fase de análisis, aplicadas a cada uno de los prototipos, se ha realizado una especificación de requisitos de software, la cual permite una visión detallada de los requerimientos con los que debe cumplir el modelo a construir.

Del análisis de los requisitos del software hemos obtenido el Diagrama de Flujo de Datos, el que nos da una perspectiva gráfica del desenvolvimiento del sitio web, mostrando los flujos de información, los procesos y subprocessos y almacenes de datos que requiere el sitio web en cuestión.

Presentamos, entonces, a continuación la Especificación de Requisitos de Software del prototipo Final, así como el Diagrama de Flujo de Datos.

A. Especificación de Requisitos de Software

1. Introducción

1.1 Propósito

Este documento define los requisitos con los que debe cumplir la Página Web con acceso a datos de la Reserva de Biósfera de Bosawás, la cual permitirá conocer información general de dicha reserva así como acceder a datos sobre las especies que en ella habitan.

1.2 Alcance

El proyecto será conocido como WEB BOSAWAS para el resto de la Especificación de Requisitos Software. WEB BOSAWÁS deberá cumplir (en el concepto de sistema) con las siguientes funciones:



- Brindar información sobre la Reserva de Biósfera de Bosawás.

- Permitir el acceso a una BASE DE DATOS sobre las especies de la Reserva por medio de consultas prediseñadas y no modificables por el USUARIO, que a continuación detallamos:
 - i. Consulta para listado de las especies en las que el USUARIO proporciona el Filo de las especies a listar.

 - ii. Consulta para listado de las especies en las que el USUARIO proporciona el Filo y la Clase de las especies a listar, donde Filo es un parámetro opcional.

 - iii. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase y Orden de las especies a listar, en donde Filo y Clase son parámetros opcionales.

 - iv. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden y Familia de las especies a listar, en donde Filo, Clase y Orden son parámetros opcionales.

 - v. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden, Familia y Género de las especies a listar, en donde Filo, Clase, Orden y Familia son parámetros opcionales.

 - vi. Consulta Libre para Listado de Especies.

Especificamos que WEB BOSAWAS no permite ingresar datos a través del sitio. Estos serán introducidos de manera manual o mediante una pequeña aplicación para dicho propósito.



1.3 Definiciones, Acrónimos y Abreviaturas

- **WEB BOSAWAS:** Aplicación Web para la Reserva de Biósfera de Bosawás que permitirá acceder a datos sobre las especies de la misma.
- **USUARIO:** Persona que accede desde un ordenador conectado a Internet a WEB BOSAWAS.
- **BASE DE DATOS:** Es la colección de datos ordenados en forma de tablas y columnas que almacenará la información correspondiente a las especies de la Reserva de Biósfera de Bosawás.
- **CONSULTAS:** Búsqueda de información que realiza el Web tomando en cuenta algunos parámetros ingresados por el USUARIO.
- **PARÁMETROS:** Conjunto de datos proporcionados por el USUARIO del Web que son ingresados de manera interactiva y son requeridos para la realización de un proceso.

1.4 Referencias

- Propuesta de Proyecto Monográfico: Página Web con acceso a datos para la Reserva de Biósfera de Bosawás. León (jueves, 4 de Abril del 2002).
- Protocolo de Proyecto de Monografía. Diseño de Página Web con acceso a datos para las especies de la Reserva de Biósfera de Bosawás. León (viernes, 6 de Septiembre del 2002).



1.5 Visión General

Antes de iniciar directamente con la especificación de los requisitos, se hará una descripción de las generalidades del producto, luego, se describirán los requisitos de una forma más detallada.

2. Descripción General

2.1 Relaciones del sitio Web

El sitio web estará alojado en un SERVIDOR Web privado. La BASE DE DATOS residirá en la misma máquina que servirá como SERVIDOR. El SERVIDOR debe estar conectado a Internet o bien adscrito como subdominio a un SERVIDOR que sí lo esté.

El funcionamiento del sitio web se basará obviamente en la arquitectura cliente – SERVIDOR. En donde el cliente enviará las peticiones al SERVIDOR quien se encargará de procesar la petición (entendiendo este proceso como el hecho de identificar la petición, ver si es posible satisfacerla o si dicha petición genera un error) y de enviar la respuesta al cliente (la respuesta consistirá en un documento HTML que contendrá el documento solicitado por el cliente o bien un documento indicando que hubo un error en el procesamiento de la solicitud).

Los requisitos mínimos de la computadora en que se alojará el sitio (o simplemente el SERVIDOR) Web serán los siguientes:

- Procesador Intel Pentium III de 1 GHZ.
- Memoria 1 GB RAM 133 MHZ SDRAM ECC.
- 36 GB Ultra 160 SCSI HDD.
- Red Dual Ethernet 100 MB/seg.



- Sistema Operativo Microsoft Windows 2000 Server.

2.2 Funciones del sitio

El sitio web está encaminado a dos ejes fundamentales desde el punto de vista funcional; el primero se centra en el hecho de proporcionar información sobre la Reserva de Biósfera de Bosawás, el segundo en permitir el acceso a una BASE DE DATOS mediante consultas prediseñadas que permitirán tener una amplia gama de resultados sobre los datos de las especies almacenados.

1. Para brindar información sobre la Reserva el sitio proporcionará un menú de navegación en la parte derecha de la pantalla en el que el USUARIO podrá escoger el apartado al cual se desea acceder simplemente dando clic sobre el hipervínculo, el cual estará relacionado con un documento que contendrá la información referente al tema en cuestión.
2. El acceso a la BASE DE DATOS, a como hemos especificado anteriormente, se realizará por medio de consultas que nosotros realizaremos y que no podrán ser modificadas por el USUARIO final. Para que el USUARIO pueda acceder a las consultas que más se ajusten a sus necesidades se han diseñado una serie de consultas que consideramos suficientes para el acceso a la información de la BASE DE DATOS.

En cada una de las consultas el USUARIO deberá proporcionar los parámetros de la consulta (en general podrán estar almacenados en una lista desplegable), luego el USUARIO deberá pulsar un botón para ejecutar la consulta.

Cabe destacar que en la mayoría de las consultas algunos de los parámetros a proporcionar serán opcionales y habrán otros que serán obligatorios. WEB BOSAWAS deberá orientar a



los USUARIOS sobre cuáles serán los parámetros opcionales y cuáles serán los parámetros obligatorios.

A continuación detallamos cada una de las consultas que el USUARIO podrá realizar:

- a. Consulta para listado de las especies en las que el USUARIO proporciona el Filo de las especies a listar.
- b. Consulta para listado de las especies en las que el USUARIO proporciona el Filo y la Clase de las especies a listar, donde Filo es un parámetro opcional.
- c. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase y Orden de las especies a listar, en donde Filo y Clase son parámetros opcionales.
- d. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden y Familia de las especies a listar, en donde Filo, Clase y Orden son parámetros opcionales.
- e. Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden, Familia y Género de las especies a listar, en donde Filo, Clase, Orden y Familia son parámetros opcionales.
- f. Consulta libre para listado de especies.

En la descripción de los requisitos funcionales describiremos cada consulta como una función independiente, es decir, como un requisito funcional independiente.



2.3 Características del USUARIO

Los USUARIOS del sitio pueden tener distintos perfiles, entre los que pueden destacarse personas que desean información sobre la Reserva por cultura general, estudiantes distintos niveles (Primaria, Secundaria, Universitarios) quienes podrán optar a la información del sitio, así como USUARIOS más avanzados cuyo objetivo será obtener datos de la BASE DE DATOS de WEB BOSAWAS para sus fines científicos particulares. Todas estas consideraciones en cuanto a los posibles perfiles de los USUARIOS se relacionan con la forma de utilización que el USUARIO hará de WEB BOSAWAS.

Con respecto a la experiencia de USUARIO en la utilización de las herramientas de navegación es posible que el sitio sea visitado por USUARIOS inexpertos, por lo que se proveerá de información explicativa en la medida de lo posible para permitir una orientación básica al USUARIO sobre la navegación eficaz del sitio.

2.4 Restricciones generales

Para el diseño del sitio utilizaremos diversos lenguajes según las necesidades de diseño. Entre estos lenguajes podemos asegurar la utilización de HTML para estructurar el sitio, Java para algunas animaciones y ASP (Visual Basic Script) para el acceso a datos.

2.5 Suposiciones y dependencias

Se ha concebido la construcción del sitio solamente en lenguaje Español, aunque no se descarta de ofrecer una traducción al Inglés para los visitantes de que únicamente dominen dicha lengua.



3. Requisitos específicos

3.1 Requisitos funcionales

3.1.1 Brindar información sobre la Reserva de Biósfera de Bosawás

3.1.1.1 Especificación

3.1.1.1.1 Introducción

Debemos considerar esta funcionalidad como un conjunto de subprocesos específicos que permitirán al USUARIO mediante la navegación acceder a la información descriptiva de la Reserva.

Para cumplir con este objetivo se le proporcionará al USUARIO un menú en la página principal para que este luego logre acceder a la información que el enlace describe.

3.1.1.1.2 Entradas

Las entradas para este proceso en dependencia de la información a obtener, serán las siguientes:

- Petición de la página generalidades.
- Petición de la página importancia.
- Petición de la página flora.
- Petición de la página fauna.
- Petición de la página etnias.
- Petición de la página problemática.
- Petición de la página galería.
- Petición de la página tour virtual.
- Petición de la página mapas.



Cada acceso a la información requiere de una petición distinta. Por ejemplo, para acceder a *Generalidades* debe hacerse la petición de la página generalidades. Para acceder a *Importancia* debe realizarse la petición de la página importancia.

3.1.1.1.3 Proceso

WEB BOSAWAS mostrará al USUARIO un menú en la parte izquierda de la pantalla del navegador con un conjunto de enlaces cuyos nombres harán referencia a cada una de las páginas que hemos mencionado anteriormente.

El USUARIO debe dar clic en el enlace referente a la información que desea explorar y posteriormente entrará a la información disponible para el apartado que ha elegido. En dicha información, en ocasiones, se encontrarán enlaces a información más específica sobre el tema en cuestión.

Cuando el USUARIO haga clic sobre el enlace se realiza la petición al SERVIDOR, luego éste procesa la petición y envía la respuesta al navegador del cliente.

3.1.1.1.4 Salidas

La salida de este proceso será la página enviada por el SERVIDOR como respuesta a la petición realizada por el USUARIO.

3.1.1.2 Interfaces externas

3.1.1.2.1 Interfaces de USUARIO

El USUARIO participará activamente por medio del envío de sus peticiones al SERVIDOR de WEB BOSAWAS.



3.1.1.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.1.2.3 Interfaces de software

En este proceso WEB BOSAWAS residirá en el SERVIDOR web. Para la realización de este proceso WEB BOSAWAS no deberá interactuar con la BASE DE DATOS. El navegador del USUARIO deberá - preferiblemente - tener capacidad para la interpretación de programas scripts.

3.1.1.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder explorar WEB BOSAWAS y obtener la información que el sitio tendrá. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.

3.1.2 Consulta para listado de las especies en las que el USUARIO proporciona el Filo de las especies a listar.

3.1.2.1 Especificación

3.1.2.1.1 Introducción

En este proceso el USUARIO tendrá como objetivo obtener una lista de especies restringidas a un solo Filo, parámetro que deberá introducir el USUARIO.



3.1.2.1.2 Entradas

El sistema debe proporcionar la lista de los Filos disponibles para que el USUARIO escoja uno de ellos. De esta manera se realizará la consulta con el Filo que el USUARIO ha escogido.

3.1.2.1.3 Proceso

Se le presentará al USUARIO la lista de los Filos disponibles de los cuales escogerá uno para realizar la consulta. Acto seguido debe pulsar en el botón denominado “Obtener lista de especies” para que se lleve a cabo la ejecución de la consulta. El SERVIDOR en el que estará alojado WEB BOSAWAS procesará la petición (que en este caso consistirá en la consulta a realizar con el parámetro Filo escogido por el USUARIO) y enviará la respuesta al cliente.

3.1.2.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).

3.1.2.2 Interfaces externas

3.1.2.2.1 Interfaces de USUARIO

El USUARIO tendrá a su disposición la lista de los Filos posiblemente en una lista desplegable para facilitar la búsqueda del mismo.



3.1.2.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.2.2.3 Interfaces de software

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.2.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder realizar la consulta sobre la BASE DE DATOS residente en WEB BOSAWAS y obtener los resultados de dicha consulta. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.

3.1.3 Consulta para listado de las especies en las que el USUARIO proporciona el Filo y la Clase de las especies a listar, donde Filo es un parámetro opcional.

3.1.3.1 Especificación

3.1.3.1.1 Introducción

El USUARIO tendrá como objetivo el obtener una lista de especies restringidas a una clase específica y opcionalmente podrán restringirse el conjunto de Clases a un Filo en particular.

3.1.3.1.2 Entradas

El sistema deberá proporcionar primeramente (al cargarse la página relacionada a esta consulta) en una lista desplegable todas las Clases disponibles en la BASE DE DATOS. En



otra lista desplegable estará disponible la lista de Filos que existen actualmente en la BASE DE DATOS más la opción “Todos los Filos” para que el USUARIO escoja si desea restringir las clases disponibles a un Filo en particular

3.1.3.1.3 Proceso

Se le presentará al USUARIO la lista de las Clases disponibles de las cuales escogerá una para realizar la consulta. Sin embargo, también se le presentará al USUARIO la lista de los Filos disponibles por si acaso éste desea restringir la lista de Clases a un Filo específico. En caso de que el USUARIO escoja uno de los Filos (si no escoge nada quedará la opción “Todos los Filos”) se volverá a llenar y/o actualizará la lista de Clases, listando únicamente aquellas que pertenecen al Filo escogido. Cuando el USUARIO haya escogido la Clase sobre la que desee obtener la lista de especies debe pulsar en el botón denominado “Obtener lista de especies” para que se lleve a cabo la ejecución de la consulta. El SERVIDOR en el que estará alojado WEB BOSAWAS procesará la petición (que en este caso consistirá en la consulta a realizar con el parámetro Clase escogido por el USUARIO) y enviará la respuesta al cliente.

3.1.3.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).

3.1.3.2 Interfaces externas

3.1.3.2.1 Interfaces de USUARIO

El USUARIO tendrá a su disposición la lista de Filos, Clases y Órdenes de acuerdo a los disponibles en la BASE DE DATOS y a los criterios de búsqueda que establezca en tiempo de ejecución.



3.1.3.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.3.2.3 Interfaces de software

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.3.2.4 Interfaces de comunicaciones

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.4 Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase y Orden de las especies a listar, en donde Filo y Clase son Parámetros opcionales.

3.1.4.1 Especificación

3.1.4.1.1 Introducción

En este proceso el USUARIO tiene como objetivo obtener una lista de especies pertenecientes a un determinado Orden, que puede estar restringido, a su vez, a un Filo y a una Clase.

3.1.4.1.2 Entradas

El sistema deberá proporcionar primeramente (al cargarse la página relacionada a esta consulta) en una lista desplegable todas los Órdenes disponibles en la BASE DE DATOS.



En otra lista desplegable estará disponible la lista de Filos que existen actualmente en la BASE DE DATOS más la opción “Todos los Filos”, además una lista desplegable en el que aparecerán las Clases disponibles en la BASE DE DATOS, todo esto para que el USUARIO escoja si desea restringir las clases disponibles a un Filo y a una Clase en particular.

3.1.4.1.3 Proceso

Se le presentará al USUARIO la lista de los Órdenes disponibles de las cuales escogerá una para realizar la consulta. Sin embargo, también se le presentará al USUARIO la lista de los Filos y la lista de las Clases disponibles, por si acaso el USUARIO desea restringir la lista de Órdenes a un Filo y a una clase en específico. En caso de que el USUARIO escoja uno de los Filos (si no escoge nada quedará la opción “Todos los Filos”) o bien una Clase (si no escoge ninguna de las clases se establecerá la opción “Todas las clases”) se actualizará la lista de Órdenes, listando únicamente aquellos que pertenecen al Filo y/o Clase escogida. Cuando el USUARIO haya escogido el Orden sobre el que desee obtener la lista de especies debe pulsar en el botón denominado “Obtener lista de especies” para que se lleve a cabo la ejecución de la consulta. El SERVIDOR en el que estará alojado WEB BOSAWAS procesará la petición (que en este caso consistirá en la consulta a realizar con el parámetro Orden escogido por el USUARIO) y enviará la respuesta al cliente.

3.1.4.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).



3.1.4.2 Interfaces externas

3.1.4.2.1 Interfaces de USUARIO

El USUARIO tendrá a su disposición la lista de Filos, Clases y Órdenes de acuerdo a los disponibles en la BASE DE DATOS y a los criterios de búsqueda que establezca en tiempo de ejecución.

3.1.4.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.4.2.3 Interfaces de software

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.4.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder realizar la consulta sobre la BASE DE DATOS residente en WEB BOSAWAS y obtener los resultados de dicha consulta. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.

3.1.5 Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden y Familia de las especies a listar, en donde Filo, Clase y Orden son parámetros opcionales.

3.1.5.1 Especificación

3.1.5.1.1 Introducción

En este proceso el USUARIO tendrá como objetivo obtener un listado de las especies pertenecientes a una determinada Familia.

3.1.5.1.2 Entradas

El sistema deberá proporcionar primeramente (al cargarse la página relacionada a esta consulta) en una lista desplegable todas las Familias disponibles en la BASE DE DATOS. En otras listas desplegables se mostrarán los Filos, las Clases y los Órdenes disponibles. El USUARIO podrá restringir la lista de Familias a un Filo, Clase u Orden en específico.

3.1.5.1.3 Proceso

Al cargarse la página se cargarán las listas con sus respectivos elementos. El USUARIO podrá elegir la Familia de la cual se desea obtener la lista de especies. Sin embargo, el USUARIO puede restringir la lista de Familias a un Filo, Clase u Orden específico en caso de que lo desee. Luego de escoger la Familia, el USUARIO deberá pulsar sobre el botón “Obtener Lista de Especies” para obtener el resultado de la consulta.

3.1.5.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).



3.1.5.2 Interfaces externas

3.1.5.2.1 Interfaces de USUARIO

El USUARIO tendrá a su disposición la lista de Filos, Clases, Órdenes y Familias de acuerdo a los disponibles en la BASE DE DATOS y a los criterios de búsqueda que establezca en tiempo de ejecución.

3.1.5.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.5.2.3 Interfaces de software

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.5.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder realizar la consulta sobre la BASE DE DATOS residente en WEB BOSAWAS y obtener los resultados de dicha consulta. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.



3.1.6 Consulta para listado de las especies en las que el USUARIO proporciona el Filo, Clase, Orden, Familia y Género de las especies a listar, en donde Filo, Clase, Orden y Familia son parámetros opcionales.

3.1.6.1 Especificación

3.1.6.1.1 Introducción

En este proceso el USUARIO tendrá como objetivo obtener un listado de las especies pertenecientes a un determinado Género.

3.1.6.1.2 Entradas

El sistema deberá proporcionar primeramente (al cargarse la página relacionada a esta consulta) en una lista desplegable todas las Familias disponibles en la BASE DE DATOS. En otras listas desplegables se mostrarán los Filos, las Clases, Órdenes y Familias disponibles. El USUARIO podrá restringir la lista de Familias a un Filo, Clase, Orden o Familia en específico.

3.1.6.1.3 Proceso

Al cargarse la página se cargarán las listas con sus respectivos elementos. El USUARIO podrá elegir el Género del cual se desea obtener la lista de especies. Sin embargo, el USUARIO puede restringir la lista de Géneros a un Filo, Clase, Orden o Familia en específico en caso de que lo desee. Luego de escoger el Género, el USUARIO deberá pulsar sobre el botón “Obtener Lista de Especies” para obtener el resultado de la consulta.

3.1.6.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).

3.1.6.2 Interfaces externas

3.1.6.2.1 Interfaces de USUARIO

El USUARIO tendrá a su disposición la lista de Filos, Clases, Órdenes, Familias y Géneros de acuerdo a los disponibles en la BASE DE DATOS y a los criterios de búsqueda que establezca en tiempo de ejecución.

3.1.6.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.6.2.3 Interfaces de software

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.6.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder realizar la consulta sobre la BASE DE DATOS residente en WEB BOSAWAS y obtener los resultados de dicha consulta. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.



3.1.7 Consulta Libre para Listado de Especies.

3.1.7.1 Especificación

3.1.7.1.1 Introducción

En este proceso el USUARIO tendrá como objetivo obtener un listado de las especies que coincidan con una expresión de búsqueda introducida.

3.1.7.1.2 Entradas

Únicamente la expresión a buscar dentro de los datos de las especies.

3.1.7.1.3 Proceso

El usuario podrá introducir en una caja de texto una cadena que será buscada entre los datos de las especies. Al pulsar el botón “Buscar” aquellas especies en cuyos datos se ha encontrado una coincidencia con respecto a la expresión buscada serán listadas.

3.1.7.1.4 Salidas

Las salidas de este proceso será el documento que el SERVIDOR enviará al navegador del cliente con los registros resultantes de la consulta ejecutada (con un formato de visualización de fácil entendimiento para el USUARIO).

3.1.7.2 Interfaces externas

3.1.7.2.1 Interfaces de USUARIO

Será necesaria la caja de texto para introducir la cadena a buscar y el botón para activar la búsqueda.



3.1.7.2.2 Interfaces de hardware

El USUARIO final deberá usar un ordenador conectado a Internet, preferiblemente deberá soportar una resolución de pantalla de 800 x 600 píxeles.

3.1.7.2.3 Interfaces de software

Las consultas se realizarán a la BASE DE DATOS alojada en el SERVIDOR de WEB BOSAWAS. El USUARIO debe tener un navegador capaz de la interpretación de scripts.

3.1.7.2.4 Interfaces de comunicaciones

El USUARIO debe tener un equipo de navegación con acceso a Internet para poder realizar la consulta sobre la BASE DE DATOS residente en WEB BOSAWAS y obtener los resultados de dicha consulta. Tanto el SERVIDOR como el USUARIO deberán tener, por lo tanto, sus debidas interfaces de redes (tarjetas de red) para la comunicación.

3.2 Requisitos de Funcionamiento

Requisitos estáticos: Aún no se ha propuesto un límite para el número de conexiones simultáneas activas a WEB BOSAWAS. Esto se determinará después de realizadas las pruebas correspondientes para ello.

Requisitos dinámicos: Se pretende que el tiempo de respuesta del SERVIDOR hacia el cliente no aumente con el número de USUARIOS conectados simultáneamente, tampoco es deseable que el tiempo de respuesta crezca en proporción a los registros devueltos como resultados en las consultas contra la BASE DE DATOS.



3.3 Restricciones de diseño

WEB BOSAWAS debe contener en su página de portada la información sobre los diseñadores del sitio, así como un enlace para el envío de correo electrónico a la persona encargada del mantenimiento del sitio.

3.4 Seguridad

3.4.1 Seguridad

La seguridad para el sitio debe de ser enfocada desde dos puntos de vista:

- Seguridad en la codificación del sitio (para evitar agujeros que sean explotados por hackers).
- Seguridad de administración en cuanto a las restricciones de directorios publicados, acceso a secuencias de comandos, exploración de directorios, etc. Elementos que serán proporcionados por el software de servidor.

3.4.2 Mantenimiento

Este documento estará sujeto a modificaciones y/o rectificaciones en las fases de análisis, diseño y codificación, las que deben hacerse constar en el mismo.

3.5 Otros Requisitos

3.5.1 Base de Datos

El almacenamiento de la información sobre las especies de la Reserva de Biósfera Bosawás se realizará en una base de datos en un gestor de bases de datos relacional.



B. Diagrama de Flujo de Datos

Es una técnica gráfica en red de un sistema. Se puede usar el DFD para representar el flujo de información y las transformaciones que se aplica a los datos al moverse desde las entradas hasta las salidas.

Notación básica de un DFD



Productor o consumidor de información que reside fuera de los límites del sistema.



Aquí es donde se transforma la información que reside dentro de los límites a ser modelado.



Un elemento de datos o una colección de elementos de datos; la cabeza de la flecha indica la dirección del flujo de datos. Todas las flechas deben estar etiquetadas.



Un depósito de datos donde se guardan para ser usados por uno o más procesos.

Diagrama de Flujo de Datos Nivel 0

En este nivel se muestra el proceso básico del sitio. En el se refleja a grosso modo la interacción del usuario con el mismo. El usuario tendrá la posibilidad de enviar una petición HTML (una dirección URL hacia algún punto de nuestro sitio, posiblemente mediante enlaces) o bien, podrá enviar una petición ASP (generalmente cuando se ejecute una de las consultas solicitadas por el usuario). En ambos casos el sitio web responderá al usuario con una página HTML que será el resultado de la petición previamente realizada (ya sea esta HTML o ASP).

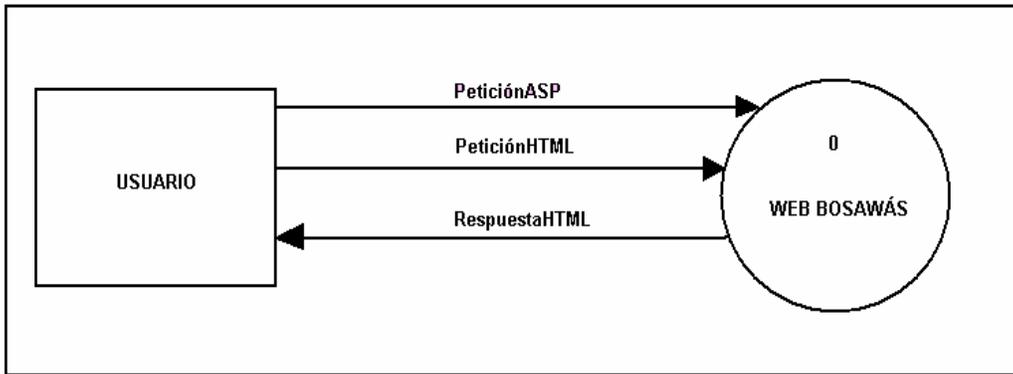


Diagrama de Flujo de Datos Nivel 1

En este nivel se detalla un poco más el anterior, mediante la refinación del proceso general en tres subprocesos:

Búsqueda de Página HTML: en este proceso únicamente se busca la página solicitada, en caso de que sea encontrada se devuelve, de lo contrario se retorna una página que indica el error.

Gestión de la petición (en la que se procesa la petición ASP, esto es ejecutar los comandos ASP y procesar la respuesta HTML que se le enviará al cliente en caso de que se encuentre

la página ASP, de lo contrario se envía una página HTML de error. También puede ocurrir un error en alguna sentencia de comando ASP, entonces también deberá generarse una página de error).

Envío de respuesta: cuya función radica en el envío del código HTML final que será interpretado por el navegador del cliente que realizó la petición al servidor web.

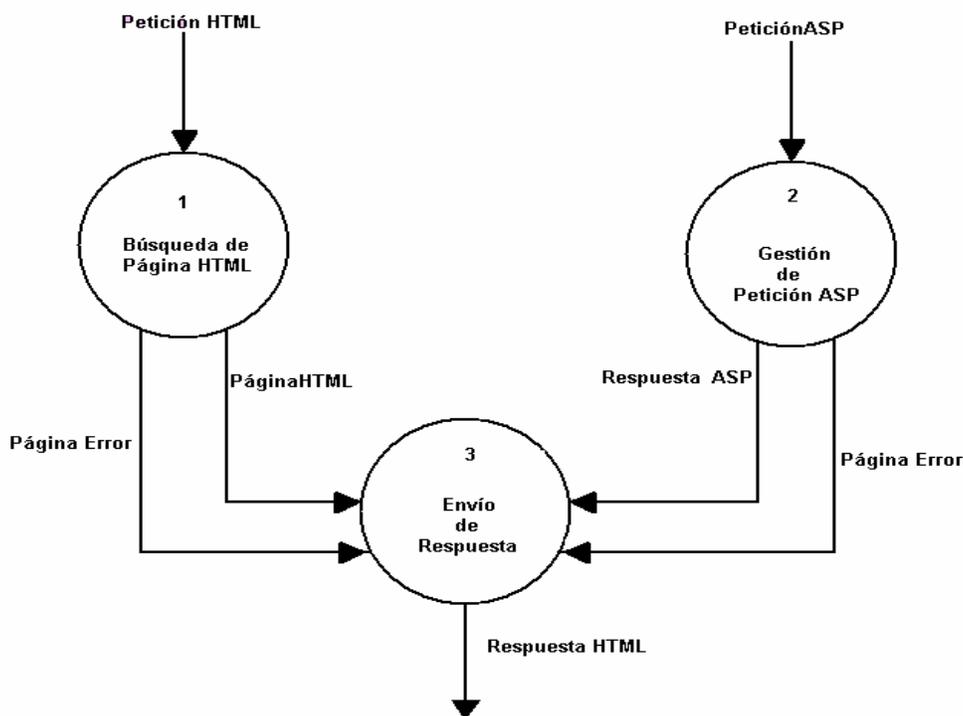


Diagrama de Flujo de Datos Nivel 2

En este proceso se explica el tratamiento que se realiza sobre la petición ASP en sí. Para tal efecto, se divide este proceso en tres subprocesos:

Análisis de la Petición ASP: Se analiza la petición ASP, se separan la parte de los parámetros de la parte de la dirección URL específica. El archivo a buscar (determinado por la dirección URL) es enviado al siguiente subproceso en caso de que se encuentre

(junto con los parámetros de la petición ASP inicial), caso contrario, se manda una página de error indicando que la página no fue encontrada.

Procesamiento ASP: En este proceso se analiza con detenimiento el código de la página ASP encontrada, se separa el código estático de los comandos ASP y se estructuran las consultas que serán ejecutadas contra la base de datos.

Ejecución de Consultas ASP: Por medio de los parámetros se construyen las consultas que se ejecutan contra la Base de Datos de Especies, de esta forma se obtienen los datos resultantes, los que consistirán en aquellos registros que cumplan con los requisitos establecidos por los parámetros introducidos por el usuario.

Estructuración de Respuesta ASP: Los datos resultantes, el código estático y los procedimientos ASP, permitirán estructurar la Respuesta ASP.

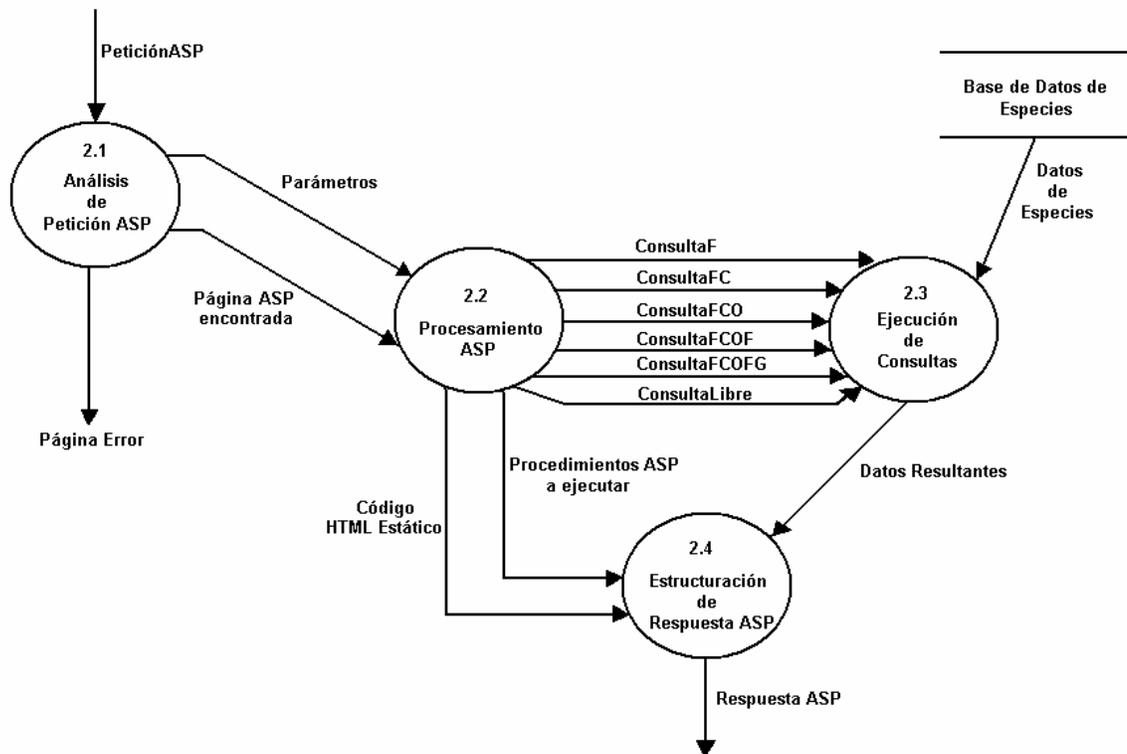
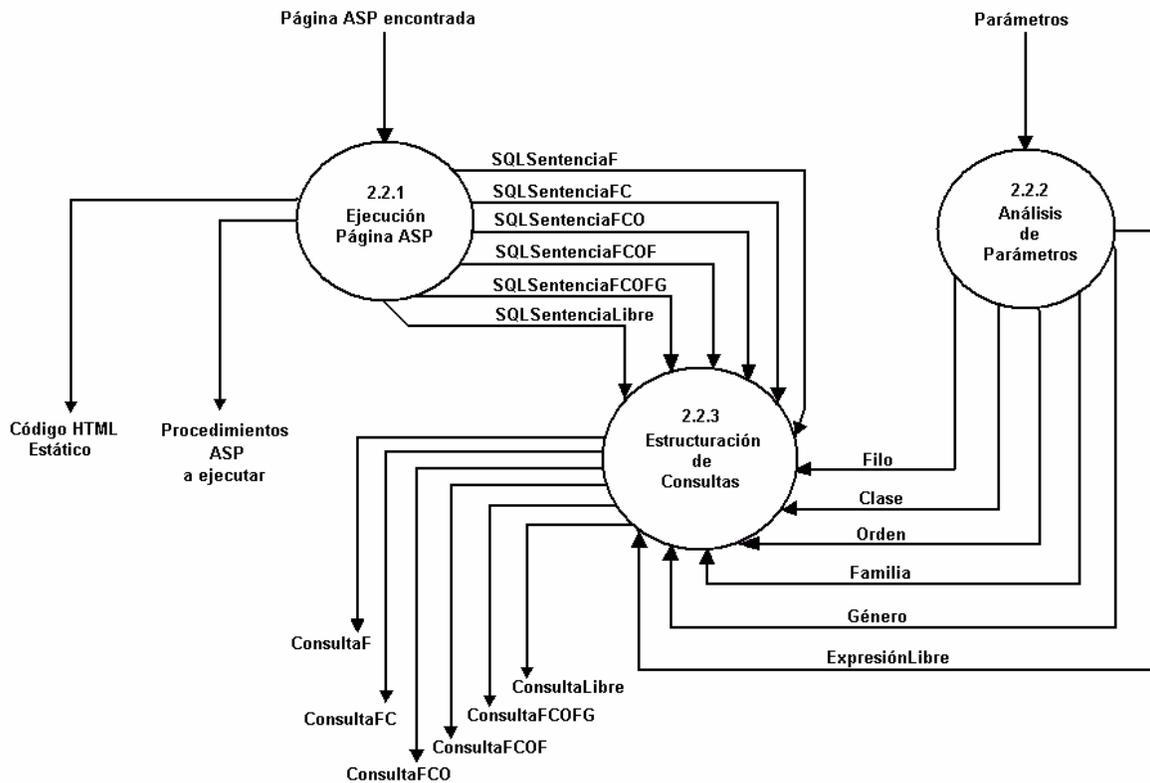


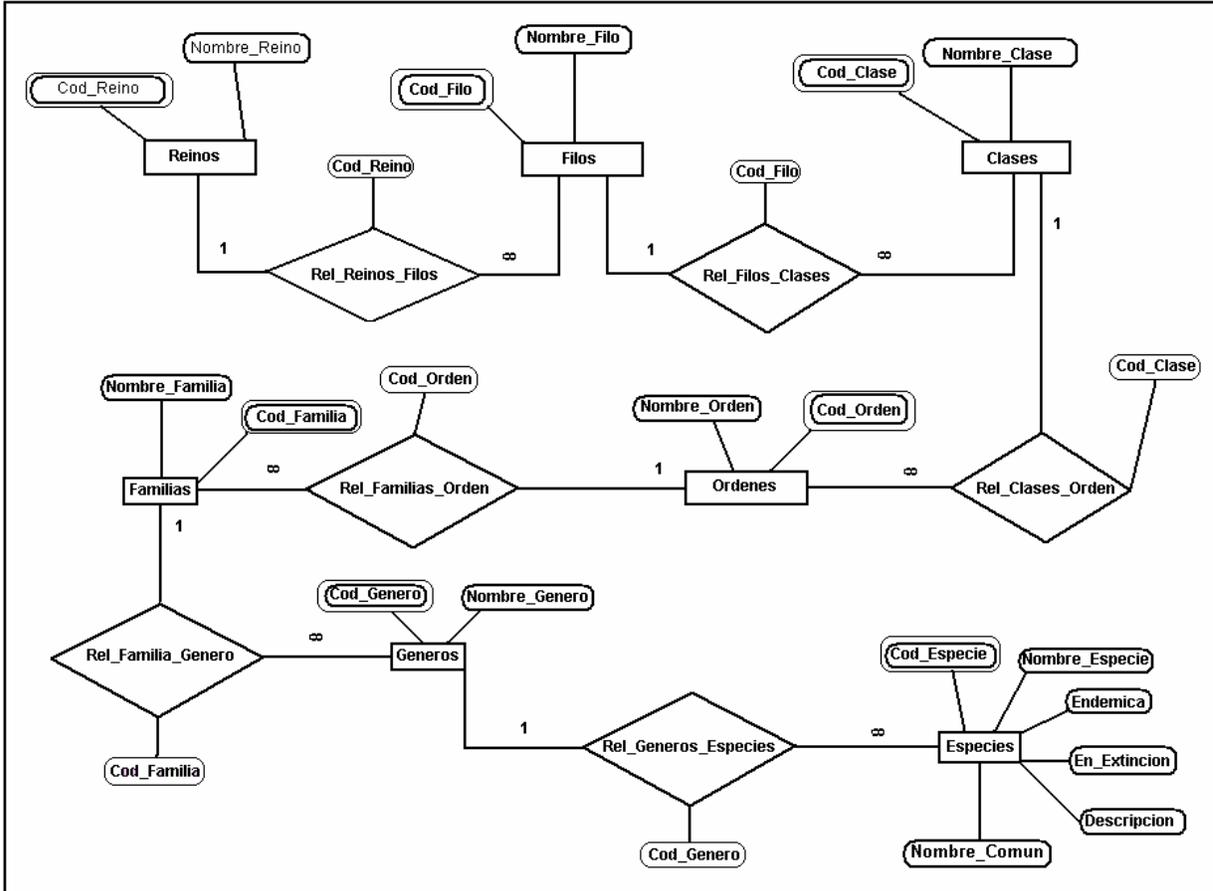
Diagrama de Flujo de Datos Nivel 3

En este Nivel se explica la estructuración de las consultas a partir de los parámetros que introduce el usuario. El *Procesamiento ASP* del nivel 2 es refinado y se detalla cómo se obtienen las consultas que se ejecutarán.



IX. DISEÑO

A. Diagrama Entidad - Relación



En este diagrama se muestra la estructura organizacional de la base de datos. Pueden apreciarse las relaciones o cardinalidades existentes entre las distintas entidades. Así pues, cada una de las subdivisiones de los seres vivos constituye una entidad con atributos propios.

En cuanto a las cardinalidades, nótese que están en estrecha similitud de la clasificación real de los seres vivos. Así, para un Filo puede haber muchos ordenes, para un Orden puede haber muchas clases, y así sucesivamente.



B. Diseño de Datos

Tabla Reinos

Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Reino	Número	Entero largo	Código de l reino
Nombre_Reino	Texto	50	Nombre del reino

Tabla Filos

Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Filo	Número	Entero largo	Código de filo o división
Nombre_Filo	Texto	50	Nombre del filo
Cod_Reino	Número	Entero largo	Código del reino al que pertenece el filo

Tabla Clases

Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Clase	Número	Entero largo	Código de la clase
Nombre_Filo	Texto	50	Nombre de la clase
Cod_Filo	Número	Entero largo	Código del filo al que pertenece la clase

Tabla Ordenes

Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Orden	Número	Entero largo	Código del orden
Nombre_Orden	Texto	50	Nombre del orden
Cod_Clase	Número	Entero largo	Código de la clase a la que pertenece el orden

Tabla Familias

Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Familia	Número	Entero largo	Código de la familia
Nombre_Familia	Texto	50	Nombre de la familia
Cod_Orden	Número	Entero largo	Código del orden al que pertenece la familia



Tabla Generos

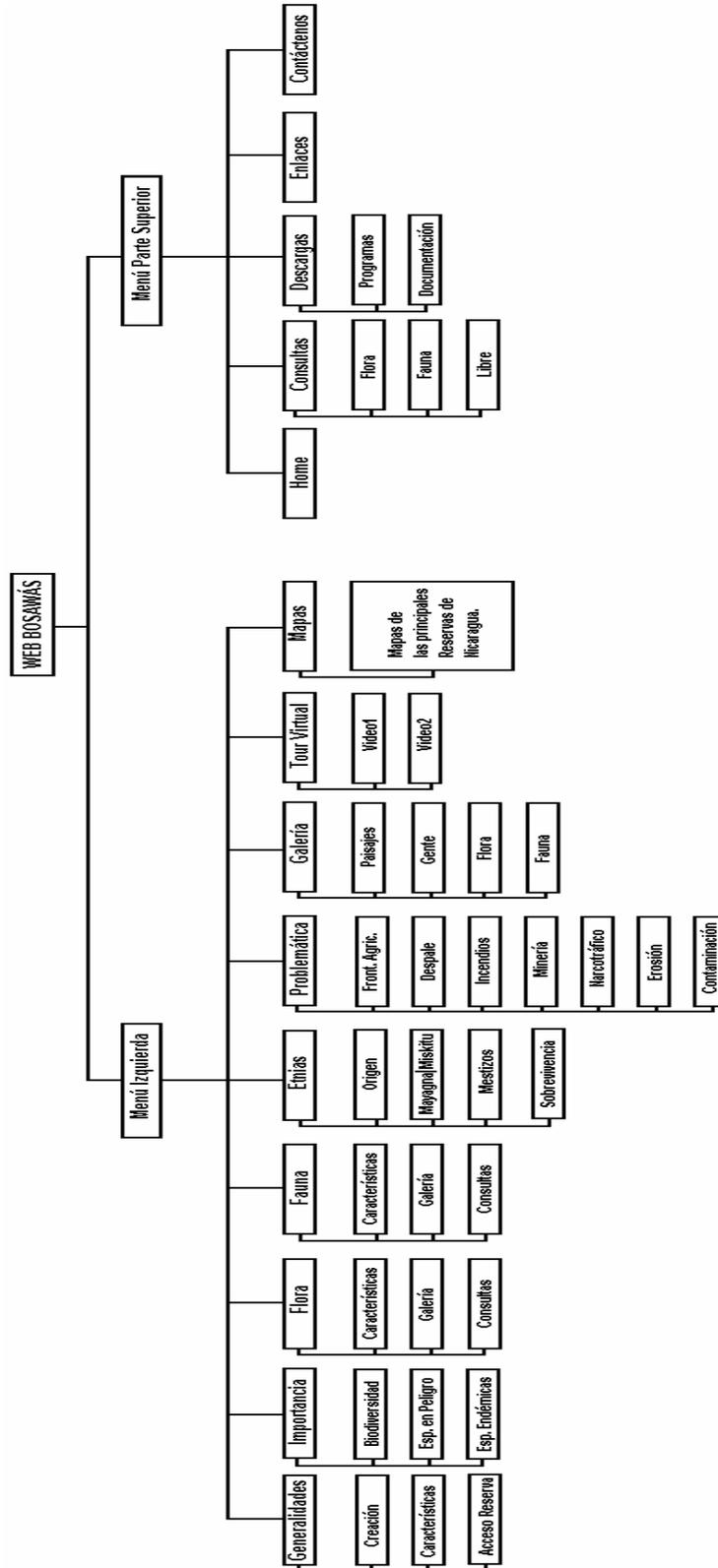
Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Genero	Número	Entero largo	Código del género
Nombre_Genero	Texto	50	Nombre del género
Cod_Familia	Número	Entero largo	Código de la familia a la que pertenece el orden

Tabla Especies

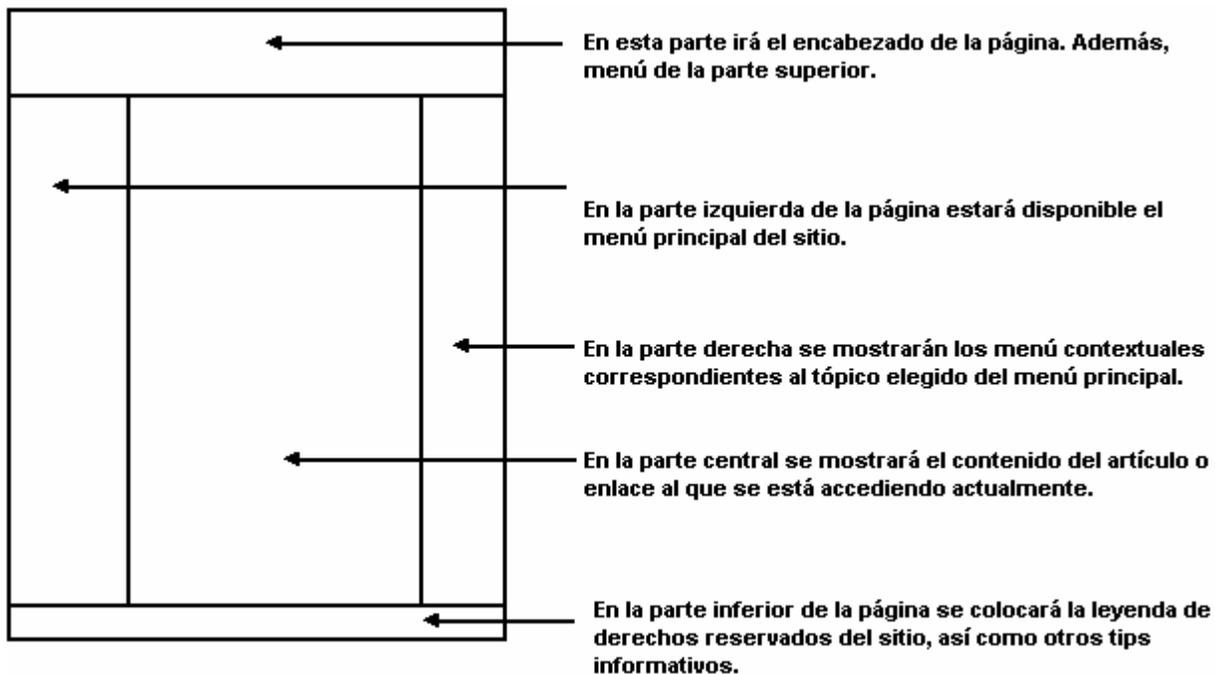
Nombre del Campo	Tipo de datos	Longitud	Descripción
Cod_Especie	Número	Entero largo	Código de la especie
Nombre_Especie	Texto	50	Nombre científico de la especie
Nombre_Comun	Texto	50	Nombre común de la especie
Cod_Genero	Número	Entero largo	Código del género al que pertenece la especie
Foto	Texto	50	Ruta de la foto de la especie
Endemica	Si/No	-	Determina si la especie es endémica
En_Extincion	Si/No	-	Determina si la especie está en peligro de extinción
Descripcion	Texto	50	Descripción adicional de la especie



C. Diseño Arquitectónico



D. Diseño de Interfaz



X. CODIFICACIÓN

1. ASP: Conceptos Fundamentales

1.1 Introducción a ASP

El lenguaje ASP surge como una extensión al código HTML para permitir un mayor dinamismo en la presentación de contenidos y obtener una mayor potencia del servidor web. Pero si la programación de estas páginas no se realiza de forma correcta puede producir graves problemas que pueden permitir a un atacante la posibilidad de acceder a datos, robar información e incluso llegar a tomar el control de la máquina.

Las páginas Activas de Servidor, representan el paradigma de la filosofía de Microsoft en su estrategia para Internet. A diferencia de otros productos de Microsoft, como los controles activex o los guiones en Visual Basic (VBScript), las aplicaciones ASP se procesan y ejecutan en el servidor y no en el cliente, siendo en este sentido parecidas a las aplicaciones en PERL o en PHP, pero integrando otros servicios y aplicaciones Microsoft, todo ello utilizando el lenguaje Visual Basic como aglutinante.

Sus principales características son:

- *Se ejecuta en el servidor:* no es el navegador quien ejecuta código ASP, sino el servidor Web, quien mandará el resultado de la ejecución al navegador.
- *Se programa en lenguaje script:* no es un lenguaje de programación propiamente dicho, ya que no sirve para crear programas independientes. En su lugar, los lenguajes script se entremezclan dentro de un documento (por ejemplo una página HTML) para dar funcionalidad en este documento e interactuar con él.



Esto implica una serie de ventajas respecto al HTML con JavaScript, las cuales se muestran a continuación:

- *Mayor seguridad:* al ejecutarse en el servidor, el código fuente nunca es enviado al navegador, con lo que al navegador le es imposible obtener el código fuente de nuestra página.
- *Mayor funcionalidad:* al ejecutarse en el servidor, podemos realizar cosas imposibles de realizar en el cliente, como por ejemplo: guardar datos en una base de datos, compartir datos entre diferentes usuarios, otro ejemplo sería realizar un contador de visitas, etcétera.
- *Mayor compatibilidad con los navegadores:* al poder realizar toda la programación en el servidor, es posible generar páginas que contengan solamente código HTML, con lo cual no estamos forzando a que el navegador deba soportar JavaScript. Existen, además, muchas incompatibilidades entre los modelos de objetos de JavaScript de distintos navegadores, por ejemplo: entre el navegador Netscape y el Microsoft Internet Explorer.
- *Lenguaje más fácil:* ASP se suele programar en Visual Basic Script (VBScript), el cual es un lenguaje idéntico casi al Visual Basic, lenguaje muy popular por su facilidad y velocidad de aprendizaje.
- *Multi-lenguaje:* ASP no es, en realidad, un lenguaje de programación, sino una plataforma de soporte para la programación de distintos lenguajes script en el servidor. Esto hace que si un programador conoce otro lenguaje script, pueda utilizarlo en lugar del VBScript, aprovechando sus conocimientos y experiencias ya adquiridos, dicho de otra forma el programador no se ve obligado a aprender un lenguaje nuevo.



1.2 Programación en el lado del servidor

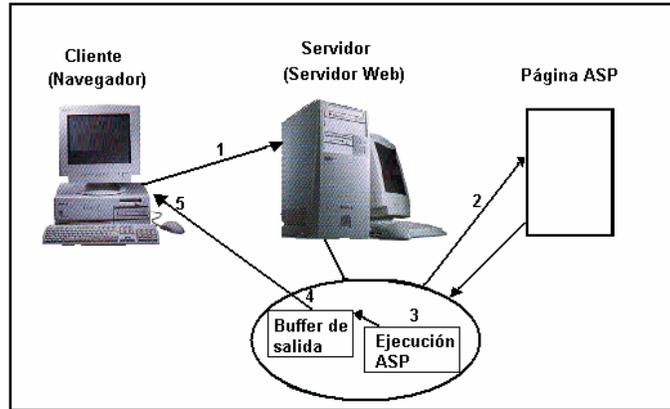
Cuando un navegador hace una petición de una página ASP se producen los siguientes acontecimientos:

1. El navegador localiza el servidor Web y le lanza una petición http donde se le solicita una página asp.
2. El servidor busca esta página asp en su disco o en su caché y si no la encuentra, genera un error. Si la encuentra la carga en el servidor.
3. El servidor interpreta la página asp, la cual contiene mezclados código HTML y código script del servidor. Cuando encuentra código HTML lo envía “tal cual“ a un buffer de salida. Cuando encuentra código script del servidor, lo ejecuta, y el resultado es enviado al mismo buffer de salida.
4. El servidor envía el buffer de salida al navegador, el cual contendrá únicamente código HTML (En realidad, el buffer de salida puede ser cualquier documento HTML válido, es decir, no tiene porqué contener sólo marcas HTML, pudiendo contener código JavaScript mezclado en el documento HTML, por ejemplo. Sin embargo, si tiene código script, será siempre código cliente, nunca código ASP.).
5. El navegador recibe el documento HTML (aunque la extensión que aparece en la URL sea .asp).

Como se puede apreciar, el paso más importante es el paso 3, donde el caso en que fuera una página HTML, simplemente se copiaría el fichero del disco al buffer de salida.



El siguiente esquema resume estos pasos:



1.3 Breve Historia

La programación en el lado del servidor existe desde los comienzos de Internet, aunque entonces se usaban tecnologías más rudimentarias. Concretamente, se usaba la tecnología CGI (Common Gateway Interface, pasarela de interficie común), que básicamente son programas independientes, escritos en cualquier lenguaje de programación (como C, Pascal, Fortran, etc.), los cuales ejecutan mandatos para generar salida HTML. Esta tecnología todavía existe, pero cada vez se encuentra más en desuso por sus inconvenientes.

Posteriormente, con la aparición de lenguajes script, apareció el concepto de script de servidor, es decir, lenguajes script (que se mezclan con documentos HTML) que se ejecutan en el servidor. Esto significó un salto importante porque ahora ya no haría falta construir todo un programa entero en un lenguaje “duro” de programación, sino que en su lugar, sólo se tendría que escribir las 3 ó 4 líneas de código en el sitio adecuado dentro de nuestro documento HTML para que el servidor ejecutara alguna acción específica.

La primera versión que apareció de ASP, en diciembre de 1996, fue la versión 1.0 de ASP, la cual se ejecutaba sobre II 3.0 (el servidor web profesional de Microsoft). Más tarde surgió una nueva versión de ASP, la versión 2.0, que se ejecuta sobre II 4.0, el cual se incluye en el option Pack para Windows NT 4.0. Este es probablemente el sistema más

usado actualmente, aunque existen dos nuevas versiones, que todavía no son muy usadas por ser bastante recientes.

Recientemente ha surgido la versión 3.0 de ASP, que se ejecuta sobre IIS 5.0, el cual se incluye junto con el sistema operativo Microsoft Windows 2000 Server.

Finalmente, existe una nueva versión llama ASP .NET, que forma parte de una nueva filosofía en las herramientas de desarrollo de Microsoft.

1.4 Ventajas frente a otras alternativas

CGI es una tecnología más antigua que ASP, la cual está cada vez en desuso debido a la facilidad y eficiencia de ASP.

Un programa CGI, a pesar de ser un programa independiente (esto es, un programa compilado escrito en un lenguaje de programación que puede ser mucho más optimizado que ASP) suele ofrecer un rendimiento muy inferior a las páginas ASP. El motivo es simple: al ser un programa independiente, debe comunicarse con el servidor web usando comunicación entre procesos. Esta comunicación entre el programa CGI y el servidor web es muy lenta, y además suele ocupar bastantes recursos. Sin embargo, esto no ocurre con las páginas ASP, las cuales utilizan otra tecnología llamada ISAPI (Internet Server Application Programming Interface, Interfaz de Programación de Aplicaciones de Servidores de Internet), que hace que el código ASP sea ejecutado por un módulo que se encuentra dentro del propio Servidor Web. Además en estas circunstancias, el servidor web puede compilar las páginas en lugar de interpretarlas, con lo que eficiencia aumenta todavía más.

Existen otras plataformas como JSP (Java Server Pages, de Sun Microsystems), ColdFusion (de Allaire) o PHP (de libre distribución bajo Open System).

Algunas de las ventajas que ofrece ASP respecto a estas otras alternativas son:



- Su bajo coste (frente a JSP o ColdFusion), pues el servidor web viene incluido con los sistemas operativos de Microsoft.
- La facilidad del lenguaje, pues ASP se suele programar en VBScript, lenguaje mucho más simple que Java o PHP (el cual es similar al C).
- Es, probablemente, la tecnología más usada (sobre todo si lo comparamos con JSP o ColdFusion). Esto hace que sea muy fácil de obtener información sobre el tema totalmente gratis, pues hay muchas páginas en Internet con manuales, tutoriales, o código fuente de páginas ASP.
- Existen servidores gratuitos para hospedar nuestras páginas ASP, y los que son de pagos (los cuales suelen ofrecer una mayor funcionalidad) no suelen ser caros.
- ASP utiliza componentes Active-X, los cuales amplían las funcionalidades de la plataforma ASP. Se puede encontrar en Internet una cantidad innumerable de componentes Active-X que pueden ser usados en nuestras páginas ASP.

1.4 Herramientas Necesarias

1.4.1 Servidor Web

Lo primero que necesitamos para ejecutar una página ASP, es tener instalado en el servidor un servidor web que soporte ASP. Dado que ASP es una tecnología propietaria de Microsoft, existen básicamente dos servidores web, los dos de Microsoft, los cuales soportan ASP: IIS y PWS.

IIS (Internet Information Server) es la solución profesional. Es decir, es un servidor web que soporta muchos usuarios conectados a la vez, y ofrece un rendimiento muy bueno. II



viene incluido en los servidores operativos basados en la tecnología NT Server, tal como Windows-NT Server 4.0 o Windows-2000 Server.

PWS (Personal Web Server) es la solución para el desarrollo. Es decir, es un servidor web con la misma funcionalidad que IIS, pero que soporta un limitado número de usuarios conectados a la vez. Concretamente, suele estar limitado a 10 usuarios a la vez, con lo que sirve a la perfección para desarrollar nuestro sitio, hacer pruebas, etc...., pero no sirve, en la mayoría de los casos, para hacer de servidor real de nuestra web, a no ser que ésta tenga pocas visitas.

La ventaja de PWS es que éste se incluye en los sistemas operativos personales de Microsoft, como Microsoft Windows-95 y Microsoft Windows-98. Concretamente, el PWS se encuentra en un directorio del CD de instalación llamado Add-ons. Por ejemplo, si nuestro sistema operativo es Windows-95 y nuestra unidad de CD es la D:, para instalar el PWS en nuestra máquina deberemos ejecutar D:\Add-ons\PWS\Instalar.exe.

1.4.2 Navegador o Browser

Con el fin de poder probar nuestras páginas ASP, necesitaremos también de un navegador, que bien puede ser Internet Explorer, Netscape Navigator, o cualquier otro que nos permita visualizar páginas HTML.

1.4.3 Editor HTML

Una página ASP, no es más que una página HTML, que contiene código ASP dentro. Es decir, una página ASP no es más que un fichero de texto llano. Por este motivo, podemos utilizar nuestro editor preferido, como por ejemplo el Block de notas de Windows, el editor del MS-DOS, o cualquier editor de HTML, como por ejemplo Dreamweaver.



Existen también herramientas específicas para la ayuda al desarrollo de las páginas ASP, tal como Microsoft Visual InterDev, incluido en la Suite Microsoft Visual Studio.

1.5 El lenguaje Visual Basic Script

En ASP, para añadir código script (script de servidor), puede emplearse también la marca HTML `<SCRIPT>`, tal y como se hace con JavaScript (script de cliente). Sin embargo, hay que añadirle un parámetro para indicarle al servidor web que este código script es de servidor, en vez de cliente. El parámetro a añadir se llama `RUNAT`, y los valores posibles son “Client” (para indicar que el script se ejecute en el cliente, este es el valor por defecto), y “Server” (para indicar que el script se ejecute en el servidor, este es el valor que usaremos nosotros).

La marca `SCRIPT`, acepta otro parámetro para indicar al lenguaje en el cual está escrito el script. Este parámetro es el parámetro `LANGUAGE`, y sus valores posibles son, entre otros, “VBScript”, “JavaScript”, “PerlScript”. Nosotros usaremos el lenguaje VBScript por ser el lenguaje más usado, más fácil y que mejor se integra con ASP.

Por lo tanto, para añadir nuestro código ASP en una página HTML, deberemos escribir:

```
<SCRIPT LANGUAGE = "VBSCRIPT" RUNAT = "Server">  
    aquí irá nuestro código servidor, escrito en VBScript  
</SCRIPT>
```

Estas marcas `<SCRIPT>` y `</SCRIPT>`, con todos sus parámetros, se tendrían que añadir cada vez que quisiéramos insertar un código script. Esto hace que el programador tenga que escribir bastante, añadiendo un overhead de código innecesario.



Por este motivo, los ingenieros de Microsoft crearon una alternativa para añadir código script en el servidor, usando las marcas `<% y %>` (equivalentes a `<SCRIPT LANGUAGE = "VBSCRIPT" RUNAT = "Server">`, y `</SCRIPT>`, respectivamente).

Por lo tanto, el anterior fragmento de código puede ser escrito de la siguiente manera:

```
<%  
    aquí irá nuestro código servidor, escrito en VBScript  
%>
```

siendo esta notación la más usada por su simplicidad.

Pero si nos detenemos a pensar un poco en los conocimientos que hemos adquirido hasta ahora, nos daremos cuenta de que algo no está bien. Por un lado, ASP se puede programar tanto en VBScript como en otros lenguajes script, y por otro lado con esta nueva notación parece que no estamos indicando el lenguaje que usamos. Entonces, cómo se sabe el intérprete de ASP que el código que está entre las marcas `<% y %>` es código escrito en VBScript? La respuesta es simple: VBScript es el lenguaje predeterminado por ASP. Sin embargo, si quisiéramos usar JavaScript en lugar de VBScript, tenemos que indicarle de algún modo al intérprete de ASP que el código script está escrito en JavaScript. Para esto se usa una directiva de compilador llamada LANGUAGE, cuyos valores posibles pueden ser "VBScript" o "JavaScript", entre otros.

Las directivas de compilación deben añadirse al comienzo de una página ASP, y se ponen entre las marcas `<%@ y %>`.

Por ejemplo, para indicar que el lenguaje script usado en nuestra página ASP va a ser VBScript, deberíamos escribir al comienzo de la página:

```
<%@ LANGUAGE = "VBScript" %>
```



Igualmente, si quisiéramos usar el lenguaje JavaScript en lugar de VBScript, escribiríamos, en lugar del código anterior, el siguiente código (también al comienzo de la página):

```
<%@ LANGUAGE = "JavaScript" %>
```

Otras marcas específicas de la plataforma ASP son las marcas `<% = y %>`. Estas marcas sirven para introducir expresiones (en lugar de instrucciones, como pasaba con `<% y %>`) el resultado de las cuales aparecerá en la página HTML resultante. Por ejemplo, si dentro de una página ASP escribo `<% = 2+2 %>`, al enviar el código HTML al navegador, este trozo será sustituido por un 4, tal y como ilustra el siguiente ejemplo:

Página ASP:

```
<HTML>
  <HEAD>
    <TITLE> Prueba de ASP </TITLE>
  </HEAD>
  <BODY>
    2 + 2 son <% = 2 + 2 %>
  </BODY>
</HTML>
```

Tras ejecutar esta página en el servidor, el código HTML que será enviado al navegador será:

```
<HTML>
  <HEAD>
    <TITLE> Prueba de ASP </TITLE>
  </HEAD>
  <BODY>
    2 + 2 son 4
  </BODY>
</HTML>
```



1.5.1 Variables

Al igual que ocurre con JavaScript, VBScript es un lenguaje débilmente tipificado. Esto significa que cuando declaramos una variable, no declaramos de qué tipo es, pues las variables pueden cambiar su tipo de función del valor que les sea asignado. Esto tiene como ventaja que podemos reciclar variables, pudiendo usar la misma variable para dos (o más) cosas distintas a lo largo de nuestro programa., incluso aunque cada una de estas cosas sea de un tipo diferente, pudiendo obtener, pues un ahorro importante de memoria.

Para declarar una variable en VBScript se usa la palabra clave Dim (igual que ocurre con Visual Basic). Así pues, si queremos declarar una variable llamada *nombre_especie*, escribiremos: *Dim nombre_especie*

En VBScript no es obligatorio, por defecto, declarar las variables. Sin embargo, es una buena práctica obligar a declarar las variables para evitar posibles errores a la hora de escribir su nombre. Si nos equivocásemos VBScript no nos avisará que se ha producido un error, sino que en su lugar creará una nueva variable, inicializada con un valor predeterminado (probablemente 0).

Para evitar esto, debemos añadir la instrucción Option Explicit al inicio de nuestro código VBScript. Esta instrucción nos obligará a declarar todas las variables que usemos y, por lo tanto, hará que el intérprete de ASP detecte el error.

Aunque en la declaración de una variable no declaremos su tipo, el valor que contiene una variable en un instante determinado es siempre de un cierto tipo. En realidad, se dice que todas las variables de VBScript son de tipo Variant (un tipo que puede ir cambiando durante la ejecución del script), y que el tipo Variant tiene distintos subtipos, equivalentes a cada uno de los tipos de valores que puede contener la variable.

La tabla siguiente muestra varios subtipos de datos que puede contener un tipo *Variant*:



Subtipo	Descripción
<i>Empty</i>	<i>Variant</i> está sin inicializar. El valor es 0 para variables numéricas o una cadena de longitud cero ("") para variables de cadena.
<i>Null</i>	<i>Variant</i> contiene intencionalmente datos no válidos.
<i>Boolean</i>	Contiene <i>True</i> o <i>False</i>
<i>Byte</i>	Contiene un entero entre 0 y 255.
<i>Integer</i>	Contiene un entero entre -32,768 y 32,767.
<i>Currency</i>	Número de -992,337,203,685,477,5808 a 992,337,203,685,477,5807
<i>Long</i>	Contiene un entero -2,147,483,648 y 2,147,483,647
<i>Single</i>	Contiene un número real de coma flotante de precisión simple entre -3,402823E38 y -1,401298E-45 para valores negativos, y entre 1,401298E-45 y 3,402823E38 para valores positivos.
<i>Double</i>	Contiene un número de coma flotante de precisión doble en el intervalo de -1,79769313486232E308 y -4,94065645841247E-324 para los valores negativos, y entre 4,94065645841247E-324 y 1,79769313486232E308 para los valores positivos.
<i>Date (Time)</i>	Contiene un número que representa una fecha entre el 1 de enero de 100 y el 31 de diciembre de 9999.
<i>String</i>	Contiene una cadena de longitud variable que puede contener hasta dos mil millones de caracteres de longitud.
<i>Object</i>	Contiene un objeto.
<i>Error</i>	Contiene un número de error.

Se puede usar las funciones de conversión para convertir datos de un subtipo a otro. Además la función *VarType* devuelve información acerca de cómo se almacena los datos en un tipo *Variant*.

1.5.1.1 Funciones de Conversión

- *Asc*: Devuelve el código ANSI del primer carácter del String pasado por parámetro.



- *Cbool*: Devuelve el valor booleano resultado de evaluar la expresión booleana pasado por parámetro. Si se pasa un entero, devuelve falso si el entero es 0.
- *Cbyte*: Convierte la expresión pasada a un entero pequeño (que cabe en un solo byte). Si hay decimales éstos se redondean. Si el valor no cabe en un byte ocurrirá un error.
- *CCur*: Convierte la expresión pasada por parámetro a tipo Currency (moneda).
- *CDate*: Convierte la expresión pasada por parámetro a tipo Date (Fecha).
- *CDbl*: Convierte la expresión pasada a un tipo Double (número real de doble precisión).
- *Char*: Convierte un código de carácter ANSI a su respectivo carácter, devolviendo un string con este carácter.
- *CInt*: Convierte la expresión pasada por parámetro a número entero (entero de 2 bytes). Es decir, convierte al subtipo Integer.
- *CLng*: Convierte la expresión pasada por parámetro a número entero largo (entero de 4 bytes). Es decir, convierte al subtipo Long.
- *CSng*: Convierte la expresión pasada por parámetro a un Single (número real de simple precisión).
- *CStr*: Convierte la expresión pasada por parámetros a String.
- *Hex*: Devuelve un string que representa el valor hexadecimal del número pasado por parámetro.



- *Oct*: Devuelve un string que representa el valor octal del número pasado por parámetro.

1.5.1.3 Restricciones de nombre de las variables

Los nombres de variables siguen las normas estándar de denominación de VBScript. Los nombres de las variables deben cumplir los siguientes requisitos:

- Debe comenzar con un carácter alfabético.
- No puede contener caracteres reservados de lenguaje (como puntos, operadores aritméticos, etc.....), siendo los caracteres válidos cualquier carácter alfabético o numérico, así como el carácter `_`.
- No debe superar los 255 caracteres de longitud.
- Debe ser único en el alcance donde se declara.

1.5.1.4 Alcance y vida de las variables

El alcance de una variable se determina cuando se declara. Cuando se declara una variable dentro de un procedimiento, sólo el código dentro de ese procedimiento puede tener acceso o cambiar el valor de esa variable. Tiene scope local y se llama variable de *nivel de procedimiento*. Si declara una variable fuera de un procedimiento, lo hace reconocible en todos los procedimientos de la secuencia de comandos. Este tipo de variable es de *nivel de secuencia de comandos* y tiene alcance de nivel de secuencia de comandos.

El tiempo que una variable existe es su vida. La vida de una variable de nivel de secuencia de comando se extiende desde el momento en que se declaró hasta el momento en que finaliza la ejecución de la secuencia de comandos. A nivel de procedimiento, una variable existe sólo cuando se encuentra en el procedimiento. Cuando sale del procedimiento, la variable se destruye. Las variables locales son adecuadas como espacio de almacenamiento temporal cuando se ejecuta un procedimiento. Puede tener variables locales del mismo



nombre en varios procedimientos diferentes, porque cada una sólo se reconoce en el procedimiento en que se declaró.

1.5.1.5 Variables escalares y variables de matrices

Muchas veces, sólo desea asignar un único valor a una variable que se ha declarado. Una variable que contiene un único valor es una variable escalar. Otras veces, es útil asignar más de un valor relacionado a una única variable. Entonces puede crear una variable que pueda contener una serie de valores. Esto se llama una variable de matriz, también conocido como un array.

Las variables “matriz” se declaran de la misma forma que las variables escalares. La diferencia es que la declaración de una variable de matriz utiliza paréntesis () a continuación del nombre de la variable.

Las matrices no están limitadas a una única dimensión. Puede tener hasta 60 dimensiones aunque la mayoría de las personas no pueden comprender más de tres o cuatro. Las dimensiones múltiples se declaran separando con comas los números de tamaño de la matriz dentro del paréntesis.

1.5.1.6 Declaración de constantes

Una constante es una variable cuyo valor se define en su declaración y no puede ser cambiado (su valor no es variable, sino constante).

Para crear una constante en VBScript se usa la palabra clave *Const* en lugar de *Dim*. Tras poner el nombre de la constante, se escribe el valor que contendrá dicha constante, separando el nombre de la constante y su valor con una asignación (que en VBScript es el símbolo =).



Los literales de tipo cadena de caracteres (string) se escriben siempre entre comillas dobles (“ ”), sin ser posible que estas comillas sean simples, tal y como ocurre con JavaScript, el cual permite tanto el empleo de comillas simples como dobles. Esto es debido a que VBScript la comilla simple se emplea para realizar comentarios de línea. Dicho de otro modo, una comilla simple (‘) en VBScript equivale a dos barras seguidas (//) en C++. Los literales de fecha, se escriben siempre entre sostenidos o almohadillas (#).

1.5.2 Operadores

Los operadores son símbolos que se insertan entre variables o constantes para formar expresiones. Existen operadores aritméticos (tal como la suma o la resta), operadores de comparación (tal como el igual o el distinto), operadores de concatenación (para juntar dos string en uno) y operadores lógicos (como la y-lógica y la o-lógica).

Prioridad de Operadores

Cuando se producen varias operaciones en una expresión, cada parte se evalúa y se resuelve en un orden predeterminado. Este orden se conoce como prioridad de los operadores. Puede utilizar paréntesis para invalidar el orden prioridad y forzar que se evalúen algunas partes de una expresión antes que otras. Las operaciones entre paréntesis siempre se ejecutan antes que las de afuera. Sin embargo, dentro de los paréntesis se mantiene la prioridad de los operadores.

Cuando las expresiones contienen operadores de más de una categoría, se evalúan primero los operadores aritméticos, a continuación los operadores de comparación y por último los lógicos. Todos los operadores de comparación tienen la misma prioridad; esto quiere decir que se evalúan en el orden en que aparecen, de izquierda a derecha. Los operadores aritméticos y lógicos se evalúan en el siguiente orden de prioridad:



Aritméticos		De comparación		Lógicos	
Descripción	Símbolo	Descripción	Símbolo	Descripción	Símbolo
Exponenciación	^	Igualdad	=	Negación	Not
Cambio de signo	-	Desigualdad	<>	Conjunción	And
Multiplicación	*	Menor que	<	Disyunción	Or
División	/	Mayor que	>	Exclusión	Xor
División entera	\	Menor o igual que	<=	Equivalencia	Eqv
Módulo Aritmético	Mod	Mayor o igual que	>=	Implicación	Imp
Suma	+	Equivalencia de objeto	Is		
Resta	-				
Concatenación de cadenas	&				

Cuando una multiplicación y una división se producen juntas en una expresión, cada operación se evalúa como ocurre de izquierda a derecha. Del mismo modo cuando una suma y una resta se producen juntas en una expresión, cada operación se evalúa según el orden de aparición, de izquierda a derecha.

El operador de concatenación de cadenas (&) no es un operador aritmético, pero en la prioridad se ejecuta después de todos los operadores aritméticos y antes de todos los operadores de comparación. El operador *Is* es un operador de comparación de referencia de objeto. No compara objetos o sus valores, solo se comprueban para determinar si dos referencias a objetos se refieren al mismo objeto.



1.5.3 Funciones

VBScript incorpora funciones propias para realizar tareas de distintos tipos. Cada una de estas funciones están explicadas detalladamente en las MSDN de Microsoft.

A continuación se nombran algunas de estas funciones clasificadas por categoría, junto con una breve descripción.

1.5.3.1 Funciones de tratamiento de fecha:

1.5.3.1.1 Obtención de la fecha y hora actual:

- Date: Obtiene la fecha actual
- Time: Obtiene la hora actual.
- Now: Obtiene el instante (Fecha + Hora) actual.

1.5.3.1.2 Funciones para operar con fechas:

- DateAdd: Suma un número de días a una fecha.
- DateDiff: Obtiene el número de días entre dos fechas.

1.5.3.1.3 Obtención de una parte de la Fecha:

- Year: Obtiene el año de una fecha.
- MonthName: Obtiene el nombre del mes de una Fecha.
- Month: Obtiene el número de mes de una fecha.
- WeekDayName: Obtiene el nombre del día de la semana.
- WeekDay: Obtiene el número de día de la semana.
- Day: Obtiene el día (dentro del mes) de una fecha.
- Hour: Obtiene la hora (Número entre 0 y 23) de una hora (obtenida con Time).
- Minute: Obtiene los minutos (entre 0 y 59) de una hora concreta.
- Second: Obtiene los segundos (entre 0 y 59) de una hora concreta.



1.5.3.2 Funciones Matemáticas

1.5.3.2.1 Funciones Trigonómicas

- Sin: Obtiene el seno de un ángulo en radianes.
- Cos: Obtiene el coseno de un ángulo en radianes.
- Tan: Obtiene la tangente de un ángulo en radianes.
- Atn: Obtiene el arco-tangente de un ángulo en radianes.

1.5.3.2.2 Funciones de Signo

- Abs: Obtiene el valor absoluto de un número.
- Sgn: Obtiene el signo de un número.

1.5.3.2.3 Funciones de Conversión y Redondeo

- Fix: Obtiene la parte entera de un número real.
- Int: Obtiene la parte entera de un número real.
- Hex: Devuelve un string con el valor hexadecimal de un número.
- Oct: Devuelve un string con el valor octal de un número.
- Round: Redondea un número real a un número concreto de decimales.

1.5.3.2.4 Funciones de Exponenciación y Logarítmicas

- Exp: Realiza la operación de exponenciación.
- Log: Obtiene el logaritmo natural.
- Sqr: Obtiene la raíz cuadrada de un número.

1.5.3.2.5 Otras

- Eval: Evalúa una expresión devolviendo su resultado.
- Rnd: Obtiene un número aleatorio.



1.5.3.3 Funciones de Tratamiento de Strings

1.5.3.3.1 Espacios

- LTrim: Elimina los espacios iniciales de un string.
- RTrim: Elimina los espacios finales de un string.
- Trim: Elimina espacios de un string.
- Space: Crea un string conteniendo un número determinado de espacios.

1.5.3.3.2 Obtención de una subcadena

- Left: Obtiene un prefijo de un string.
- Right: Obtiene un sufijo de un string.
- Mid: Obtiene una parte central de un string.

1.5.3.3.3 Funciones de Mayúsculas/Minúsculas

- Ucase: Pasa un string a mayúsculas.
- Lcase: Pasa un string a minúsculas.

1.5.3.3.4 Otras

- Len: Obtiene el número de caracteres de un string.
- InStr: Obtiene la posición de una subcadena dentro de otro string.
- InStrRev: Hace lo mismo que InStr pero empezando por el final.
- Replace: Sustituye dentro de un texto una cadena por otra.
- StrReverse: Invierte el orden de caracteres de un string.
- Split: Parte un string en dos.
- StrComp: Comparación de dos strings.



1.5.3.4 Funciones de Tipos de Datos

1.5.3.4.1 Averiguar si una variable es de un cierto tipo:

- **isArray:** Devuelve true si la variable es array.
- **isDate:** Devuelve true si la variable es de tipo Date.
- **isEmpty:** Devuelve true si la variable está vacía.
- **isNull:** Devuelve true si la variable es Null.
- **isNumeric:** Devuelve true si la variable es de tipo numérico.
- **isObject:** Devuelve true si la variable es un objeto.

1.5.3.4.2 Obtención del tipo de una variable.

- **VarType:** Devuelve una constante indicando el tipo de variable.
- **TypeName:** Devuelve un string con el nombre del tipo de variable.

1.5.3.5 Otras

- **RGB:** Devuelve un único valor numérico que representa un color RGB.

1.5.4 Sentencias de Control

Las sentencias de control se usan para controlar el flujo de ejecución de nuestro programa, permitiendo ejercer distintas alternativas en función de una condición, o añadir un bucle para ejecutar un fragmento de código repetidamente mientras se cumpla una condición.

1.5.4.1 Sentencia If

Una de las sentencias de control que tiene todo lenguaje de programación, es la sentencia if, que nos permite realizar alternativas dentro del código en función de una condición.



La sintaxis del if más simple es:

```
If condición_booleana Then  
    Instrucciones a ejecutar  
End If
```

Como se puede apreciar, la sintaxis es parecida a la instrucción de if de C o de JavaScript, pero poniendo la palabra clave Then en lugar de { y la palabra End If al finalizar el If (en lugar de }). Otra diferencia entre VBScript y JavaScript (o también C) es que VBScript no distingue entre mayúsculas y minúsculas, siendo equivalente, pues, escribir IF, if, If o incluso iF.

También se puede añadir la cláusula else para hacer que se ejecute un código alternativo en caso que no se cumpla la condición del if.

Finalmente, la palabra clave Elseif, nos permite realizar más de una comparación, permitiendo realizar más de 2 alternativas.

1.5.4.2 Sentencia SELECT CASE

Otra instrucción para realizar alternativas en la instrucción *Select case*. Una instrucción Select Case proporciona una funcionalidad similar a la instrucción *If...Then...Else*, pero con muchos lenguajes la incluyen debido a que hace el código más eficiente y legible. En C o en JavaScript existe una instrucción equivalente llamada *switch*.

Una estructura Select Case trabaja con una expresión de comprobación sencilla que se calcula una vez, al comienzo de la estructura. Después el resultado de la expresión se compara con los valores para cada Case. Si existe una coincidencia, se ejecuta el bloque de instrucciones asociado con ese Case.



Se puede añadir como último caso el Case Else, el cual ejecutará instrucciones si la expresión del Select Case no es igual a ningún valor de ninguno de los Case. Es decir, la alternativa Case Else se ejecutará sólo cuando no se ejecute ninguna otra alternativa. El Select Case equivale al default de C o JavaScript.

1.5.5 Bucles

La mayoría de lenguajes disponen también de mecanismos para hacer bucles p iteraciones, que consiste en la capacidad de poder repetir un bloque de instrucciones mientras se cumpla una cierta condición, o durante un número de veces concreto.

En VBScript están disponibles las siguientes instrucciones de iteración:

- **While...Wend:** ejecuta el bucle mientras una condición sea cierta (**true**).
- **Do...Loop:** ejecuta el bucle mientras o hasta que una condición sea cierta (**true**).
- **For...Next:** utiliza un contador para ejecutar instrucciones un número de veces específico.
- **For Each...Next:** repite un grupo de instrucciones para cada elemento de una colección o para cada elemento de una matriz.

1.5.5.1 Uso de bucles Do

Se puede utilizar las instrucciones Do...Loop para ejecutar un bloque de instrucciones de número de veces indefinido. Las instrucciones se repiten mientras una condición es *True* o hasta que una condición pasa a ser *cierta*.



1.5.5.1.1 Repetición de instrucciones mientras una condición es cierta

Usaremos la palabra clave *While* para comprobar una condición en una instrucción *Do...Loop*. Se puede comprobar la condición antes de entrar en el bucle o puede comprobarla después que el bucle se haya ejecutado al menos una vez.

1.5.5.1.2 Repetición de una instrucción hasta que una condición pasa a ser True

Puede utilizar la palabra clave *Until* de dos formas para comprobar una condición en una instrucción *Do...Loop*. Puede comprobar la condición antes de entrar en el bucle o puede comprobarla después de que el bucle se haya ejecutado al menos una vez. Mientras la condición es *False* se ejecuta el bucle.

1.5.5.2 Uso de While...Wend

La instrucción *While...Wend* se proporciona en VBScript para las personas que están familiarizadas con su uso. Sin embargo, se recomienda que utilice *Do...Loop* debido a la escasa flexibilidad de la instrucción *While...Wend*. La instrucción *While...Wend* es equivalente a *Do...Loop*.

1.5.5.3 Uso de For...Next

Puede utilizar *For...Next* para ejecutar un bloque de instrucciones en número de veces específico. Utilice una variable de tipo contador cuyo valor aumente o disminuya con cada repetición de bucle. Utilizando la palabra clave *Step* puede aumentar o disminuir la variable tipo contador en el valor que especifique. Para disminuir la variable tipo contador, utilice un valor *Step* negativo. Cuando lo haga, debe de especificar un valor final que sea menor que el valor inicial.



Puede salir de cualquier instrucción For...Next antes que el contador alcance su valor final utilizando la instrucción *Exit For*. Como normalmente desea salir sólo en ciertas situaciones, como cuando se produce un error, debe utilizar la instrucción *Exit For* en el bloque de instrucciones *True* de una instrucción *If...Then...Else*. Si la condición es *False*, el bucle se ejecuta normalmente.

1.5.5.4 Uso de For Each...Next

Un bucle *For Each...Next* es parecido a un bucle *For...Next*. En lugar de repetir las instrucciones un número determinado de veces, un bucle *For Each...Next* repite un grupo de instrucciones para cada elemento de una colección de objetos para cada elemento de una matriz. Esto es especialmente útil si no desea conocer cuántos elementos hay en una colección.

1.5.6 Procedimientos y Funciones

Para crear procedimientos y funciones en VBScript, se usan las palabras clave *Sub* y *Function* al principio del procedimiento o función, y *End Sub* y *End Function* al final del procedimiento o función, respectivamente. La sintaxis completa para crear un procedimiento es:

```
Sub nombre_procedimiento (parámetros)  
    Cuerpo_del_procedimiento  
End Sub
```

Mientras que la sintaxis completa para crear una función es:

```
Function nombre_función (parámetros)  
    Cuerpo_de_la_función  
End Function
```



1.5.6.1 Parámetros

Dado que VBScript es un lenguaje débilmente tipificado, tal y como ocurre con JavaScript, la lista de parámetros es simplemente una lista con los nombres de cada parámetro separados por comas.

Para retornar el valor que debe retornar la función, simplemente se asigna al nombre de la función el valor que deseamos retornar, como si el nombre de la función fuera una variable.

Para llamar a un procedimiento se usa la palabra clave *call* seguido del nombre del procedimiento al que queremos llamar. Si éste tiene parámetros, éstos se ponen después del nombre del procedimiento, entre paréntesis y separados por comas.

Para llamar a una función, se escribe el nombre de la función, seguido de sus parámetros entre paréntesis y separados por comas. Hay que tener en cuenta que la llamada a una función devuelve siempre un valor, por lo que no se puede usar una función como instrucción, sino como expresión (por ejemplo se puede poner una llamada a una función a la derecha de una asignación o en la condición de un if).

1.6 Objetos implícitos ASP y componentes ASP

ASP es una plataforma basada en objetos. Esto significa que ofrece un conjunto de objetos para ampliar la funcionalidad del lenguaje que estemos usando (por ejemplo VBScript).

ASP ofrece 5 objetos, cada uno de los cuales contienen una serie de métodos, propiedades y eventos. Estos 5 objetos son: *Response*, *Request*, *Application*, *Session* y *Server*.

El objeto *Request* representa los datos que el navegador web envía al servidor web, es decir, representa la petición del navegador hacia el servidor web.



El objeto *Response* representa los datos que el servidor web envía al navegador, es decir, representa la respuesta que el servidor web da al navegador.

El objeto *Application* representa nuestra aplicación web, es decir, el conjunto de páginas web que forman nuestro sitio.

El objeto *Session* representa la sesión de usuario actual, es decir, el cliente que está usando nuestra aplicación.

Finalmente, el objeto *Server* representa al servidor web.

Estos 5 objetos son los llamados objetos implícitos de ASP, los cuales son fijos (siempre estos 5) y son la base de toda esta plataforma.

Existen, sin embargo, otro tipo de objetos llamados componentes ASP, destinados a ampliar aún más la funcionalidad de nuestras páginas ASP. Estos componentes son componentes realizados con la tecnología Active-X de Microsoft, lo que implica que no estamos hablando de una lista cerrada, sino de una lista que es ampliable, pudiendo incluso crear nuestros propios componentes Active-X desde otras herramientas de Microsoft como por ejemplo Visual Basic o Visual C++.

1.7 Objetos Application y Session

Ya hemos visto que con la palabra reservada Dim, podemos crear variables en VBScript. Sin embargo, estas variables son visibles tan sólo dentro de nuestra página (o solamente dentro de nuestro procedimiento o función, si la variable está definida dentro de este procedimiento o función).

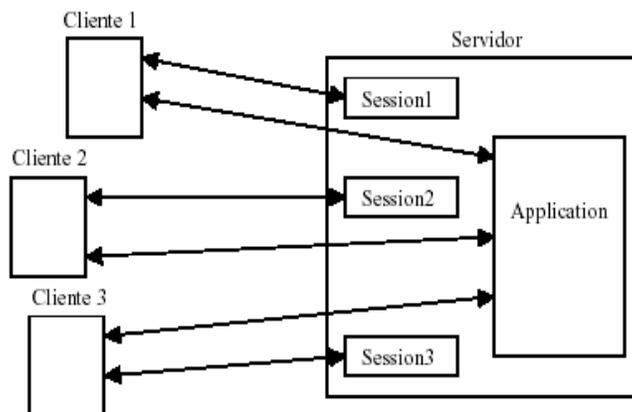
Sin embargo, en muchas ocasiones nos interesará crear variables globales que sean visibles por todo el mundo. Éstas son las llamadas *variables de aplicación*.



Para acceder a una variable de aplicación, simplemente ponemos entre comillas y paréntesis el nombre de la variable que queremos crear, siguiendo al objeto Application.

Existe un peligro en las variables de aplicación, y es el hecho del paralelismo de acceso a datos. Para evitar este problema, debemos asegurarnos que cuando estamos accediendo a la variable lo estamos haciendo exclusivamente, es decir, debemos bloquear la variable antes de ejecutar la instrucción, y liberarla justo después. Existen dos métodos del objeto Application para bloquear y liberar, respectivamente, son los métodos: *Lock* y *Unlock*.

El objeto *Session* sirve también para crear variables cuyo valor sea compartido por distintas páginas, sin embargo, el valor de las variables de sesión no se comparte entre las distintas sesiones (es decir, entre los distintos clientes o usuarios).



En la figura se puede observar esquemáticamente cómo cada cliente tiene acceso sólo al objeto Session que le corresponde. Sin embargo, todos pueden acceder al objeto Application, el cual es un único objeto compartido por todos los clientes (o compartidos por todas las sesiones).

Este tipo de variables se usa para almacenar valores personalizados como pueden ser datos personales del navegante, opciones que ha escogido, etc....

El objeto Session tiene un método llamado *abandon* el cual al ser llamado fuerza el cierre de la sesión, aunque todavía no haya expirado el timeout.

Cada sesión se identifica internamente por un identificador global (GUID). Aunque no es habitual, este identificador se puede obtener mediante la propiedad *SessionID* del objeto Session.

El objeto Session no requiere los métodos Lock y Unlock, ya que cada usuario tiene acceso a un objeto Session distinto (SU objeto Session), con lo que no hay conflictos de acceso.

Una característica importante de las variables Globales (tanto de aplicación como de sesión) es que éstas no se declaran, creándose la variable en el primer momento en que ésta sea utilizada.

1.8. Global.asa

Existe un fichero especial en todo sitio ASP, llamado Global.asa, el cual contiene 4 eventos que podemos programar. Son los eventos de inicio de aplicación (*Application_OnStart*), inicio de sesión (*Session_OnStart*), fin de aplicación (*Application_OnEnd*) y fin de sesión (*Session_OnEnd*). Este fichero debe estar en el directorio home de nuestro sitio.

Este fichero se programa también en VBScript (o en cualquier otro lenguaje script soportado por ASP), pero en lugar de usar las marcas `<% y %>`, estamos obligados a usar la sintaxis de las marcas SCRIPT. Es decir, al principio de este fichero deberemos escribir:

```
<SCRIPT LANGUAGE = "VBScript" RUNAT = "Server">
```

y al final del fichero deberemos escribir:

```
</SCRIPT>
```



escribiendo entre marcas aquellos procedimientos de evento que nos interese reprogramar (de entre los 4 procedimientos posibles).

1.8.1 Application_OnStart

El procedimiento de evento Application_OnStart se ejecuta en el momento de arrancar nuestro servidor web. Éste es el procedimiento que se utiliza para hacer las inicializaciones que necesitemos a nivel de aplicación.

1.8.2 Application_OnEnd

Éste procedimiento de evento se ejecuta en el momento en que nuestro servidor web se para. Se puede programar, por ejemplo para que guarde todas las variables. Application en una base de datos con el fin de poder recuperar posteriormente los valores actuales.

1.8.3 Session_OnStart

Este procedimiento se ejecuta en el momento en que se inicia una nueva sesión. Esto es, en el momento en que entre un nuevo usuario.

1.8.4 Session_OnEnd

Finalmente, existe un evento que se activa cuando un usuario abandona una sesión. Sin embargo, un navegador web no envía ninguna señal al servidor web cuando el usuario cierra el navegador. Por este motivo, la forma de detectar este evento es el siguiente:

Cada vez que un cliente hace una petición, se inicia un contador de tiempo, el cual es reiniciado cada vez que se detecta cualquier actividad de ese cliente (o sesión). Cuando transcurrido un timeout un cliente no ha tenido ninguna actividad, es decir, cuando el

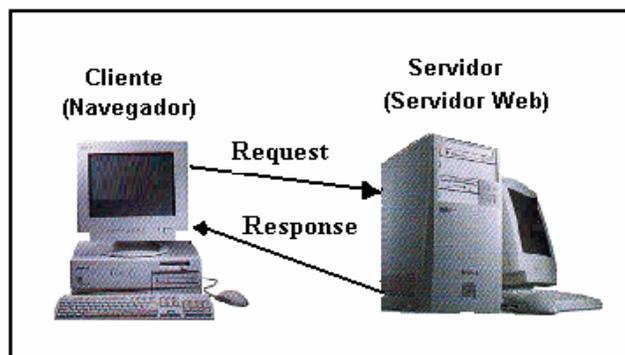


contador de tiempo alcanza este timeout, se considera que el usuario ha abandonado la sesión.

El timeout por defecto es de 20 minutos, sin embargo, este timeout es configurable por la propiedad *Timeout* del objeto Session.

1.9. Response y Request

Los objetos Response y Request representan, respectivamente, la respuesta que el servidor web envía al navegador, y los datos que el navegador envía al cliente.



Existe, por ejemplo, un método del objeto Response que se encarga de escribir en la salida HTML que se enviará al navegador. Es el método write, que tiene como único parámetro del código HTML que queremos escribir. En realidad, el método write no es muy usado, porque se suele usar la notación `<%=` en lugar de este método. Sin embargo, el método Response posee otros métodos y propiedades que sí se utilizan.

1.9.1 Caché y caducidad de la página

Una de las grandes funcionalidades que poseen prácticamente todos los navegadores es el poder guardar las páginas usadas más recientemente en una zona temporal llamada *caché*. Esto, aunque disminuye el tráfico de red y aumenta la velocidad de navegación, es una

fuente de problemas, ya que las páginas almacenadas en la caché no ejecutan código en el servidor.

Uno de los parámetros que el servidor web le da al navegador cuando éste recibe una página HTML es la *fecha de caducidad* de esta página. La fecha de caducidad indica el momento a partir del cual la página está caducada y por lo tanto el navegador está obligado a solicitar una nueva versión de la página al servidor web, aunque tenga esta página en caché. Es decir, un navegador guarda una página en la caché hasta que esta página “caduca”.

Esta fecha de caducidad está contenida en la cabecera http, y es uno de los parámetros que podemos controlar mediante el objeto Response. Concretamente, el objeto Response tiene una propiedad llamada “*expires*”, en la cual indicamos el número de minutos dentro de los cuales queremos que la página caduque.

Si queremos que nuestra página nunca sea guardada en la caché del navegador para que este pida siempre la página a nuestro servidor web, obligando así a que el código ASP sea ejecutado cada vez que la página sea visionada, deberemos escribir:

```
<%Response.Expires=0%>
```

con lo cual estamos indicando que la página caduque inmediatamente (dentro de 0 minutos).

Debido a que el parámetro expires viaja en la cabecera http, esto es, antes que el cuerpo del documento HTML sea enviado, deberemos escribir una instrucción antes de enviar ningún carácter HTML al navegador (esto es, antes de la marca <HTML>).

Existen otros métodos y propiedades del objeto Response. Quizás uno de los métodos más utilizados (aparte de la propiedad expires) es el método *Redirect*, que sirve para



redireccionar la página actual a una nueva página. El método *Redirect* tiene un solo parámetro que es la dirección URL donde queremos que nuestra página sea redireccionada. Esta URL puede ser tanto absoluta como relativa.

1.9.2 Recibir información de formularios. Métodos GET y POST

Hemos hablado bastante sobre cómo escribir scripts de servidor que generen la página HTML, incluso que modifiquen propiedades de esta como es el caso de su caducidad. Sin embargo, una de las cosas sobre la que no se ha hablado todavía y es de vital importancia, es de la obtención de datos introducidos por el usuario.

El principal mecanismo que tiene HTML para enviar información introducida por un usuario son los formularios. Los formularios son estructuras lógicas de un documento HTML que pueden contener distintos controles donde el usuario puede introducir información. Estos controles pueden ser cuadros de texto, listas, checkboxes, radiobuttons, etc..., que es donde el usuario introduce o selecciona de entre un conjunto de opciones de la información que se le solicita. Finalmente los formularios suelen tener un botón llamado *botón de submit*, el cual, al ser pulsado, toda la información del formulario es enviada al servidor.

Para hacer formulario en HTML se utiliza la marca <FORM>. Los parámetros importantes de esta marca son:

ACTION: indica a la URL hacia dónde hay que enviar estos datos. En nuestro caso, indica la página ASP que recibirá estos datos.

METHOD: indica el método que se desea usar para enviar la información del formulario. Existen dos métodos posibles: El método GET y el método POST.



Para obtener los parámetros de una URL, esto es, para obtener los datos de un formulario que ha sido enviado usando el método GET, se usa la propiedad `QueryString` del objeto `Request`. Esta propiedad todo lo que hay después del interrogante (que es la marca de inicio de parámetros de una URL).

Sin embargo, existe una forma mucho más cómoda de obtener cada uno de los parámetros. La propiedad `QueryString`, es, en realidad, lo que se llama *colección*, esto es, una lista de valores (algo parecido a un array, pero donde cada elemento tiene un nombre que lo identifica). Podemos acceder a un elemento concreto de una colección añadiendo detrás de la colección el nombre del elemento cuyo valor deseamos obtener entre comillas y paréntesis.

Existe un método mucho más fiable para enviar los datos de un formulario. Es el llamado método POST. Cuando un formulario se envía usando el método POST sus datos viajan dentro del cuerpo del mensaje del protocolo http, siendo posible codificar dicho contenido con certificados de seguridad como SSL. Además, no existe límite teórico en cuanto a bytes para un mensaje http. Por este motivo, este es el protocolo más usado para el envío de formularios.

Para recoger los datos de un formulario enviado mediante el método POST se usa la propiedad `Form` del objeto `Request`, siendo la sintaxis la misma que la de la propiedad `QueryString` (es decir, `Form` es también una colección, cuyos elementos son identificados por el nombre de cada uno de los controles del formulario y en cada uno de estos elementos se encuentra su valor).

1.9.3 Trabajar con Cookies

Otra de las posibilidades que nos ofrecen los objetos `Request` y `Response` es el de poder trabajar con cookies.



Las cookies son pequeñas informaciones que el navegador puede guardar en una zona especial del disco duro del cliente (por ejemplo, Netscape guarda las cookies en un fichero llamado cookies.txt dentro del directorio de aplicación de Netscape, mientras que Internet Explorer guarda las cookies en un directorio de Cookies dentro del directorio de Windows).

Cuando el navegador hace una petición http al servidor, este envía junto con la petición sus cookies. Es posible, pues, obtener estas cookies mediante el objeto Request.

El objeto Request ofrece una propiedad llamada Cookies, la cual contiene una colección con cada una de las cookies enviadas por el cliente. Toda cookie posee un nombre que la identifica, un valor, y una fecha de caducidad a partir de la cual la cookie desaparece.

Sin embargo, si lo que deseamos es asignar un valor a una cookie no podemos usar el objeto Request, sino que debemos usar el objeto Response, ya que lo queremos es que nuestro servidor web envíe al navegador de una cookie para que este la guarde en el disco duro del cliente. El objeto Response ofrece también una propiedad llamada cookies que es también una colección de cookies. La diferencia entre la propiedad cookie del objeto Response y la propiedad cookies del objeto Request es que la del objeto Response es de sólo escritura y la del objeto Request es de sólo lectura, ya que Request no permite consultar los valores de las cookies recibidas del cliente, mientras que Response nos permite enviar al navegador nuevos valores para las cookies (sean nuevas o no).

Por defecto, una cookie caduca al caducar la sesión. Por este motivo, cuando cerremos el navegador, la cookie desaparecerá, en realidad sin que esta haya llegado a ser guardada en disco.

Por eso deberíamos establecer una fecha de caducidad para la cookie. Para establecer una fecha de caducidad para la cookie deberemos utilizar la propiedad expires que tiene toda cookie. Es decir, escribiríamos la siguiente instrucción:



<%Response.cookies("Idioma predeterminado").expires=#2/23/2003#%>

1.10 Objeto Server

El objeto Server representa al servidor web, y aunque tiene otros métodos, los más importantes son HTML Encode y CreateObject.

1.10.1 HTML Encode

El método HTML Encode del objeto Server se utiliza para codificar un texto en HTML. Este método tiene un solo parámetro que es el texto a codificar.

El hecho de no codificar los textos a formatos HTML puede implicar no sólo que ciertos caracteres aparezcan incorrectamente, sino que nuestra página no sea interpretada correctamente por el navegador.

1.10.2 Acceso a Bases de Datos

1.10.2.1 Acceso a Bases de Datos desde ASP

Sin duda alguna, uno de los mayores atractivos de las páginas ASP y de los lenguajes script de servidor en general, es su capacidad de interactuar con bases de datos, pudiendo generar el contenido de la página HTML en función del contenido de una base de datos.

Sin embargo, Visual Basic Script no tiene instrucciones para el acceso a bases de datos, así como tampoco los tiene directamente ninguno de los cinco objetos implícitos de ASP. Es por este motivo que para acceder a bases de datos hay que utilizar componentes ActiveX.

Los componentes ActiveX que se instalan por defecto al instalar la plataforma ASP (ya sea instalando IIS o instalando PWS) son el componente ADODB.Connection para crear una



conexión a una base de datos, y el componente ADODB.Recordset el cual se usa para obtener un conjunto de registros de una conexión realizada a una base de datos.

El objeto Server nos ofrece el método CreateObject para crear una instancia de un componente ActiveX, esto es, para crear una conexión (una instancia de ADODB.Connection) o un recordser (una instancia del componente ADODB.Recordset). El método CreateObject tiene un único parámetro que es el nombre del tipo de componente ActiveX, del cual queremos crear una instancia. Por ejemplo, para crear una instancia de un componente ADODB.Connection escribiremos:

```
Server.CreateObject("ADODB.Connection")
```

Y para crear una instancia de un componente ADODB.Recordset escribiremos:

```
Server.CreateObject("ADODB.Recordset")
```

Las instancias de los objetos se declaran también con Dim, sin embargo, para asignar un objeto debemos usar la palabra clave Set.

Así pues, por ejemplo, para crear una instancia de un componente ADODB.Connection y asignarla a una variable (un objeto) llamada cn escribiremos:

```
Dim cn
```

```
Set cn=Server.CreateObject("ADODB.Connection")
```

1.10.1.2 ADODB.Connection

El objeto ADODB.Connection sirve para crear una conexión a una base de datos mediante los protocolos estándares de comunicación con las bases de datos ODBC (Open DataBase Connectivity) y OLEDB (Object Linked and Embedded – DataBase). Tanto ODBC como



OLEDB, son protocolos creados por Microsoft. La diferencia básica entre los dos es que OLEDB es de posterior creación siendo más eficiente, funcional y flexible.

1.10.1.2.1 El Método Open

Lo primero que hay que hacer siempre antes de acceder a una base de datos, es abrir una conexión a ella y esto es justamente lo que nos permite el método Open del componente OLEDB.Connection.

El método Open consta de tres parámetros: Cadena de Conexión, Usuario y Contraseña. Sin embargo, los dos últimos parámetros son opcionales, porque la cadena de conexión puede incluir ya, el usuario y contraseña con los que accederá a la base de datos.

La cadena de conexión es un string donde se especifica el proveedor OLEDB, la base de datos concreta que se quiere utilizar, el usuario y la contraseña.

El proveedor OLEDB se especifica mediante la palabra clave *Provider*, y sirve para escoger el controlador de la base de datos. A continuación se ofrece una pequeña lista de los proveedores OLEDB de bases de datos Microsoft más usados:

Nombre del Proveedor OLEDB	Descripción
Microsoft.Jet.OLEDB.3.51	Proveedor de Datos OLEDB para bases de datos Access 97
Microsoft.Jet.OLEDB.3.6	Proveedor de Datos OLEDB mejorado para bases de datos Access 97
Microsoft.Jet.OLEDB.4.0	Proveedor de Datos OLEDB para bases de datos Access 2000 y Access-XP
SQLOLEDB	Proveedor de Datos OLEDB para bases de datos SQLServer.

La base de datos, en el caso de Microsoft se refiere al archivo .mdb que deseamos abrir. Se especifica con la palabra clave DataSource.

El usuario con el que queremos acceder a la base de datos se puede especificar mediante el parámetro User ID, mientras que la contraseña se especificará con la palabra clave Password. Por ejemplo:

```
cn.open "Provider=Microsoft.Jet.OLEDB.4.0;DataSource=dat.mdb;User  
ID=Juan;Password=xx"
```

1.10.1.2.2 Método Execute

El método Execute del componente ADODB.Connection, se usa para enviar una sentencia SQL al proveedor de base de datos al que estamos conectados. SQL (Structured Query Language) es un lenguaje universal para el acceso a las bases de datos que nos permite, por ejemplo modificar o eliminar registros en una tabla de la base de datos. La sintaxis del método Execute es:

```
cn.Execute instrucción_SQL
```

donde *cn* es nuestra conexión a la base de datos (es decir, una instancia de ADODB.Connection) e *instrucción_SQL*, es un string que contiene la sentencia SQL que queremos ejecutar en la base de datos.

1.10.1.2.3 El método close

Este método cierra la conexión liberando todos los recursos que esta necesita. Es importante utilizar el método close cuando hemos terminado de usar la conexión. Igualmente, también es importante liberar la memoria usada por la instancia creada. Esto se consigue asignando a la variable que contiene la instancia el valor Nothing.



Resumiendo, cuando no usemos más la conexión, debemos escribir:

```
cn.close  
set cn=Nothing
```

1.10.1.2 ADODB.Recordset

Representa un conjunto de registros obtenidos de nuestra base de datos. Se usa principalmente para mostrar en nuestra página web datos obtenidos de nuestra base de datos. Para su utilización, primero se declara una variable y luego se le asigna una instancia de este componente, es decir:

```
Dim rs  
Set rs=Server.CreateObject("ADODB.Recordset")
```

Seguidamente, hay que usar el método open para abrir el recordset recientemente creado, sin embargo, es necesario abrirlo utilizando una conexión a la base de datos. Por eso, uno de los parámetros (concretamente el segundo) del método open del recordset es la conexión. El otro parámetro es una sentencia SQL para la obtención de los datos (Sentencia SELECT). Por ejemplo:

```
rs.open "Select * From Coches", cn
```

Un recordset es un conjunto de registros con un cursor que apunta al registro actual. Cuando se abre un recordset, el registro actual es el primero de los registros obtenidos (siempre que haya como mínimo un registro). Si no hay ningún registro, el registro actual apunta a un registro no válido.

El recordset posee dos propiedades para detectar si el registro actual es válido o no. Son las propiedades booleanas BOF y EOF. Si estas propiedades sin true, significa que el recordset



está vacío, es decir, no hay registros en la base de datos, y por lo tanto el registro actual no es válido.

1.10.1.2.1 Los métodos move

Existen cuatro métodos de desplazamiento en un recordset: Son los métodos MoveFirst, MoveNext, MovePrevious y MoveLast, los cuales desplazan el cursor actual al primer, siguiente, anterior y último registro del recordset respectivamente. Por ejemplo, podría usarse: *rs.MoveNext*

Dado que tras abrir el recordset el registro activo se posiciona al principio, podemos realizar un recorrido secuencial de todo el recordset simplemente utilizando el método MoveNext repetidamente hasta llegar al final del recordset. Pero necesitamos poder detectar si hemos llegado al final de recordset, lo cual es indicado por la propiedad EOF.

1.10.1.2.2 Obtener el valor de un campo en el Registro Actual

Un recordset es también una colección, siendo esta colección el conjunto de campos que posee el recordset. Esto supone que podemos acceder al valor de cualquier campo (del registro actual) escribiendo el nombre del campo entre comillas seguido de nuestra variable que contiene el recordset, por ejemplo: *rs("Matricula")*

1.10.1.3 Lenguaje SQL

El Lenguaje SQL (Structured Query Language) es el lenguaje universal de las bases de datos, soportado por todos los sistemas gestores de bases de datos importantes como Oracle, SQLServer e incluso Access.

El éxito de este lenguaje se debe a su compatibilidad con todos los gestores de bases de datos relacionales, a su clara sintaxis (siendo esta muy parecida al lenguaje natural inglés) y



a su alto nivel (en una instrucción SQL simplemente se indica lo que se desea hacer, sin tener que escribir la forma de proceder para conseguirlo, siendo esto último tarea del sistema gestor de la base de datos).

Las sentencias del lenguaje SQL se dividen en dos grandes categorías:

- **Sentencias DDL:** Las sentencias DDL (Data Definition Language) son las sentencias SQL que nos permiten definir las estructuras de los datos. Aquí están sentencias para crear tablas, índices, usuarios, etc. Este tipo de sentencias no son las que nos interesan a nosotros, pues supondremos que la base de datos ya está creada. Además algunos sistemas gestores de bases de datos, por ejemplo Microsoft Access, incorporan asistentes para crear las tablas de forma visual.
- **Sentencias DML:** Las sentencias DML (Data Manipulation Language) son las sentencias SQL que nos permiten insertar, modificar, borrar o consultar los datos almacenados en una tabla. Existen cuatro sentencias básicas que se corresponden a las cuatro operaciones anteriormente mencionadas: INSERT, UPDATE, DELETE y SELECT. Estas sentencias son usadas por todo programa que acceda a bases de datos, sin importar la plataforma de programación.

1.10.1.3.1 La sentencia SELECT

La sentencia SELECT se usa para obtener un conjunto de registros de la base de datos. La sintaxis es:

Select campos From tabla

Donde tabla es el nombre de la *tabla* que contiene los datos a consultar y *campos* es una lista de los nombres de los campos que deseamos obtener separados por coma, o bien un asterisco (*) indicando que se desean todos los campos.



1.10.1.3.2 Cláusula WHERE

La cláusula WHERE es una palabra clave que se puede añadir en una sentencia SELECT después de la cláusula FROM. El Where nos permite definir una condición que deben cumplir los registros para que sean incluidos como resultados, es decir, el Where nos permite obtener solo aquellos registros que cumplan una cierta condición. Ej:

Select NombreEmpleado From Empleados Where Ciudad= "León"

1.10.1.3.3 Otras Cláusulas

Existen más cláusulas posibles de la sentencia SELECT, como son las cláusulas GROUP BY, HAVING, ORDER BY, las cuales permiten obtener los registros de manera más apropiada a las necesidades del programador. Su uso no será explicado en este documento.

Así también, existen sentencias para la inserción de un nuevo registro (INSERT), la actualización (UPDATE) y borrado (DELETE), las cuales merecen detenimiento y por falta de espacio no explicaremos en este trabajo.

1.11 Seguridad desde el nivel de programación

Una de las características de las aplicaciones por Internet, es que éstas son accesibles a todo el mundo y muy a menudo de una forma anónima.

Es por esto que cuando desarrollemos este tipo de aplicaciones (ya sea con ASP o usando otras tecnologías) hay que prestar una atención muy especial al tema de la seguridad.

En general, la seguridad tiene distintos niveles. Por ejemplo, podemos hablar de que la plataforma ASP es segura, en cuanto a que como plataforma de Internet te ofrece unas garantías, esto es una base sobre la que partir, la cual parece segura (en realidad, ninguna



plataforma es segura al 100%, y conviene estar inscrito en grupos de noticias sobre la seguridad para saber en todo momento los agujeros de seguridad que se descubran en la plataforma que usemos, y podernos bajar la correspondiente actualización que solucione el problema.).

Sin embargo, es un error muy grave pensar que si la plataforma es segura, nuestra aplicación será segura. Nuestra aplicación será segura sólo si la plataforma es segura y si seguimos ciertas reglas, sobre todo al tratar elementos que el usuario introduce.

Una de las cosas que hay que evitar es que nuestra página puede producir errores. Al decir errores no nos referimos a errores de compilación, sino al hecho que, por ejemplo, el usuario pueda introducir valores que causen errores en nuestras páginas.

Por ejemplo, imaginemos que estamos haciendo una calculadora por Internet donde el usuario introduce dos números y tiene botones para sumar, restar, multiplicar y dividir. Las operaciones que ASP nos ofrece para recoger los valores del usuario (Request.Form), para realizar la operación que haya pulsado (operadores +, -, * o /), y para mostrar el resultado (Response.write) están (en principio) exentas de errores. Sin embargo, podría darse el caso que el usuario ponga como segundo número el 0 y pulse el botón dividir. Al realizar el cálculo de división la aplicación daría un error, haciendo que ASP genere como salida HTML una página indicando dicho error.

Igualmente, tampoco podemos estar seguros que el usuario teclee sólo números en las casillas. Podemos pensar el caso de controlar esto con un script en el cliente, sin embargo, algunos navegadores no ejecutan JavaScript, y otros permiten desactivarlo, por lo que esta forma de proceder no es una opción nada segura (en general, cualquier sistema de protección que implementemos usando JavaScript, no es nada seguro, siendo JavaScript útil únicamente para evitar tráfico excesivo por la red).



El hecho de controlar muy bien todos los errores que puedan suceder es muy importante no sólo por cuestiones de imagen (menuda será nuestra web si falla cada dos de tres), sino por cuestiones de seguridad, pues la mayoría de veces un error o un caso no tenido en cuenta por el programador, puede ser explotado por un hacker hábil.

2. Ejemplo de Codificación ASP utilizado en el sitio.

2.1 Codificación de página de captación

```
<%
'-----
' Este documento mostrará al cliente, a través de un combo box, una lista con
' los Filos disponibles al usuario que se encuentran en la base de datos
'-----

Dim oConn, rsFilo, SQLFilo 'Se declaran las variables para la conexión y recordsets.
set oConn=Server.CreateObject("ADODB.Connection") 'y se les inicializa
set rsFilo=Server.CreateObject("ADODB.Recordset")

' Cadena de Conexión por OLEDB para conectarse a la base de datos
oConn.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("Bosawas97.mdb") & ";"

'Se estructura a consulta para obtener los nombres de los fillos existentes en la base de datos.
SQLFilo="SELECT Filos.Nombre_Filo FROM Reinos INNER JOIN Filos ON Reinos.Cod_Reino = Filos.Cod_Reino
WHERE Reinos.Nombre_Reino='Animal'"

' Abrimos el recordset
rsFilo.Open SQLFilo, oConn, 1, 2

%>

<p align="center"><font face="Arial" color="#003C00"><b>Consulta de especies
para un determinado Filo</b></font></p>

'Se dibuja el formulario que contendrá el combobox, y demás controles que dispararán la consulta
<form method="POST" action="rs_cnsFilos.asp" style="text-align: center">
<input TYPE="hidden" NAME="pagina" VALUE="0">&nbsp;
```



```
<fieldset style="border: 1px solid #003C00; padding: 2; width: 250; font-size: 8pt; font-weight: bold">
  <legend align="center">&nbsp;</legend>
  <p align="center"><font face="Arial" color="#003C00">Lista de Filos</font><p align="center">
  <select size="1" name="optFilo" style="color: #003C00; font-weight: bold">
<%
    'Este código llena la lista desplegable con los Filos disponibles
    Do Until rsFilo.EOF = True
      Response.write "<OPTION>" & rsFilo("Nombre_Filo") & "<br>"
    rsFilo.MoveNext
  Loop
%>
</select></p>

<p align="center">
<input type="submit" value="Obtener Lista de Especies" name="B1" style="color: #003C00; font-weight: bold; height:
30; width: 200;"></p>
<p>&nbsp;</p>
</fieldset></form>

<p align="left"><b><font face="Arial" size="2" color="#003C00">Puede elegir uno
de los Filos, luego pulse en el botón 'Obtener Lista de Especies' para obtener
la lista de especies pertenecientes al filo escogido.</font></b></p>
<%
  'Se cierran los recordsets y conexiones y se libera la memoria que ocupan
  rsFilo.Close
  Set rsFilo=Nothing
  oConn.Close
  Set oConn=Nothing
%>
```



2.2 Codificación de Página de resultados

'En este ejemplo se codifica la página de resultados para el listado de especies x Filo

,

```
<% 'En esta parte se captura el filo del que se listarán las especies
```

```
Dim strFilo
```

```
strFilo=Request.Form("optFilo")
```

```
if Len(strfilo)<>0 then
```

```
Session("filo")=Request.Form("optFilo")
```

```
Else
```

```
strFilo=Session("filo")
```

```
End if
```

```
Session("reino")="animal"
```

```
%>
```

```
<html>
```

```
<head>
```

```
<title>Resultados de la consulta para el Filo <%= "" & Session("filo") & "" %></title>
```

```
</head>
```

```
<body>
```

```
<p align="center"><font face="Arial" color="#003C00"><b>Resultados de la  
consulta para el Filo <%= "" & Session("filo") & "" %>
```

```
<% 'Se declaran algunas variables necesarias para la paginación de resultados
```

```
Dim oConn, rs, SQL
```

```
Dim PaginaActual          ' en qué pagina estamos
```

```
Dim PaginasTotales        ' cuántas páginas tenemos
```

```
Dim TamPagina             ' cuantos registros por pagina
```

```
Dim CuantosRegistros      ' para imprimir solo el n° de registro por pagina que queramos
```

```
set oConn=Server.CreateObject("ADODB.Connection")
```

```
set rs=Server.CreateObject("ADODB.Recordset")
```

```
' Conexión por OLEDB
```

```
oConn.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("Bosawas97.mdb") & ";"
```

```
'oConn.Open "Driver={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath("Bosawas97.mdb")
```



'Aquí se determina el No de registros por página

TamPagina=10

'Leemos qué página mostrar. La primera vez será la inicial

if Request.QueryString("pagina")="" then

 PaginaActual=1

else

 PaginaActual=CInt(Request.QueryString("pagina"))

end if

'Estructuramos la consulta para listar las especies de un determinado Filo

```
SQL="SELECT      Especies.Nombre_Especie,      Generos.Nombre_Genero,      Familias.Nombre_Familia,
Ordenes.Nombre_Orden, Clases.Nombre_Clase FROM Reinos INNER JOIN (((Filos INNER JOIN Clases ON
Filos.Cod_Filo = Clases.Cod_Filo) INNER JOIN Ordenes ON Clases.Cod_Clase = Ordenes.Cod_Clase) INNER JOIN
((Familias INNER JOIN Generos ON Familias.Cod_Familia = Generos.Cod_Familia) INNER JOIN Especies ON
Generos.Cod_Genero = Especies.Cod_Genero) ON Ordenes.Cod_Orden = Familias.Cod_Orden) ON Reinos.Cod_Reino
= Filos.Cod_Reino WHERE (((Filos.Nombre_Filo)="" & strFilo & "")) AND Reinos.Nombre_Reino='Animal';"
```

' Defino el tamaño de las páginas

rs.PageSize=TamPagina

rs.CacheSize=TamPagina

' Abro el recordset

rs.Open SQL, oConn, 1, 2

'Cuento las páginas

PaginasTotales=rs.PageCount

'Compruebo que la pagina actual está en el rango

'de lo contrario: si se pasa tomamos la última página y si es negativo tomamos la primera

if PaginaActual < 1 then

 PaginaActual = 1

end if

if PaginaActual > PaginasTotales then

 PaginaActual = PaginasTotales

end if

'Por si la consulta no devuelve registros!

if PaginasTotales=0 then



```
        Response.Write("No se encontraron resultados.")
    else
        'Vamos a paginar
        rs.AbsolutePage=PaginaActual
        Response.Write(" (Pagina " & PaginaActual & " de " & PaginasTotales & ")<hr color=#003C00>")
        Response.write "</font>"
    %>

'Poneremos los títulos de las columnas de las tablas
<div align="center">
    <center>
    <table border="2" cellpadding="1" cellspacing="0" bordercolor="#003C00">
        <tr>
            <td align="center"><b><font size=2 color="#003C00">Especie</font></b></td>
            <td align="center"><b><font size=2 color="#003C00">Género</font></b></td>
            <td align="center"><b><font size=2 color="#003C00">Familia</font></b></td>
            <td align="center"><b><font size=2 color="#003C00">Orden</font></b></td>
            <td align="center"><b><font size=2 color="#003C00">Clase</font></b></td>
        </tr>
    </table>
</div>

'CuantosRegistros=0
'En este ciclo vamos ubicando cada uno de los registros, hasta que se llegue al último
'o bien hasta llegar al total de registros por página.
do while not rs.EOF and CuantosRegistros < TamPagina
    'Pintamos una línea de la tabla por cada registro
    Response.Write("<TR>"&VbCrLf)
    Response.Write("<TD align=center><font size=2 color=#003C00>"&rs.Fields(0)&"</font></TD>")
    Response.Write("<TD align=center><font size=2 color=#003C00>"&rs.Fields(1)&"</font></TD>")
    Response.Write("<TD align=center><font size=2 color=#003C00>"&rs.Fields(2)&"</font></TD>")
    Response.Write("<TD align=center><font size=2 color=#003C00>"&rs.Fields(3)&"</font></TD>")
    Response.Write("<TD align=center><font size=2 color=#003C00>"&rs.Fields(4)&"</font></TD>")
    Response.Write("</TR>")
    CuantosRegistros=CuantosRegistros+1
    rs.MoveNext
loop
Response.Write("</TABLE>")
end if
rs.Close                'Cerramos las conexiones y recordsets
oConn.Close            'Liberamos los recursos que ocupan
```



XI. PRUEBA

En este apartado se mostrarán los resultados obtenidos de las dos codificaciones ejemplo que hemos mostrado en el capítulo X correspondiente a *codificación*.

Cabe destacar que la prueba se realizó en Windows 2000 Server con Internet Information Server 5.0. Esto en la máquina de desarrollo del proyecto.

1. Resultados de la página de captación.

Reserva de Biósfera de Bosawás

Home Consultas Descargas Enlaces Contactos

Consulta de especies para un determinado Orden

Filos Clases Órdenes

Chordata Aves Galliformes

Obtener Lista de Especies

Puede restringir la lista de Clases a un Filo específico. A la vez puede restringir la lista de Órdenes a una Clase determinada. Para obtener la lista de especies correspondientes al orden seleccionado finalmente pulse en el botón "Obtener Lista de Especies"

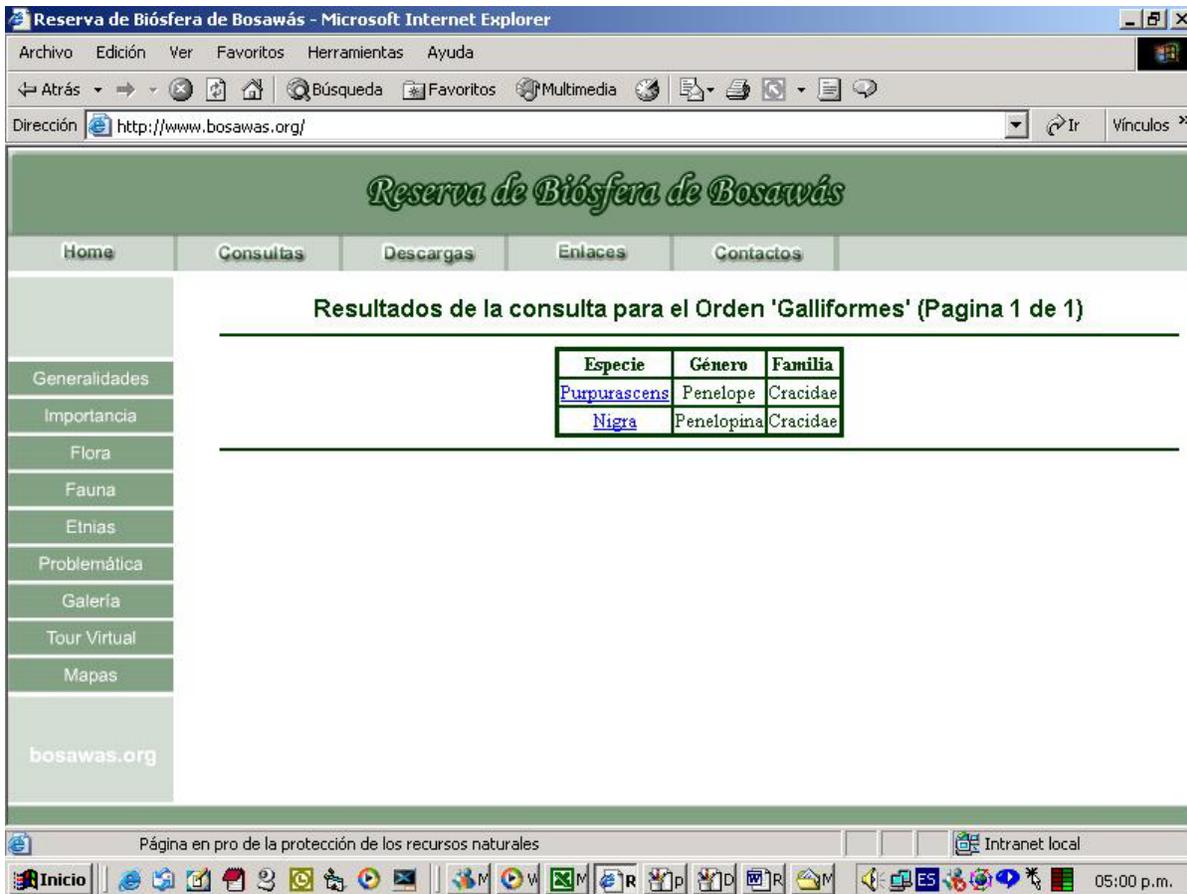
bosawas.org

Página en pro de la protección de los recursos naturales Sitio We Intranet local

04:51 p.m.



2. Prueba de Página de Resultados.



The screenshot shows a Microsoft Internet Explorer browser window displaying the website 'Reserva de Biósfera de Bosawás'. The address bar shows the URL 'http://www.bosawas.org/'. The website has a green header with the title 'Reserva de Biósfera de Bosawás' and a navigation menu with links for 'Home', 'Consultas', 'Descargas', 'Enlaces', and 'Contactos'. A sidebar on the left contains links for 'Generalidades', 'Importancia', 'Flora', 'Fauna', 'Etnias', 'Problemática', 'Galería', 'Tour Virtual', and 'Mapas'. The main content area displays the title 'Resultados de la consulta para el Orden 'Galliformes' (Pagina 1 de 1)' and a table with the following data:

Especie	Género	Familia
Purpurascens	Penelope	Cracidae
Nigra	Penelopina	Cracidae

At the bottom of the browser window, the taskbar shows the system tray with the date and time '05:00 p.m.' and the text 'Página en pro de la protección de los recursos naturales' and 'Intranet local'.



XII. IMPLEMENTACIÓN

En la fase de implementación deben considerarse dos elementos fundamentales: la plataforma de montaje del sitio (constituida por el servidor web Internet Information Server) y el Servicio de Alojamiento Web en donde residirá el sitio.

1. Internet Information Server

Internet Information Server es un servidor Web, que incluye los servicios de: HTTP, HTTPS, FTP, SMTP (correo saliente) y NNTP (grupos de noticias). Además es capaz de ejecutar varios motores de script como: ASP, PHP, Cold Fusion, etc.

A diferencia de su hermano menor IIS 4.0, la nueva versión de IIS 5.0 está incluida en el sistema operativo Windows 2000 Server y Advanced Server, si no indicamos lo contrario durante la instalación del Sistema Operativo, ésta aplicación se instalará por defecto con los servicios Web y SMTP, no instalándose el servidor NNTP para los grupos de noticias ni el FTP que debería especificarse si se quieren instalar.

Si por el contrario no se ha instalado IIS 5.0, es necesario seguir las siguientes instrucciones:

1. Ir a Mi PC
2. Panel de Control
3. Añadir o quitar programas
4. Seleccionar "añadir y quitar componentes de Windows" (figura 1).
5. Seleccionar Internet Information Server, nos pedirá el CD de Windows 2000 para copiar e instalar el software.



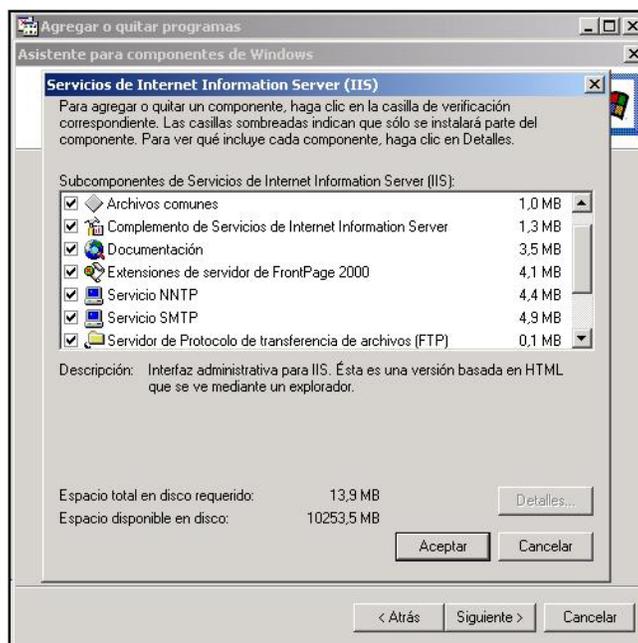


Figura 1

Una vez instalado para acceder al panel de control del IIS 5.0 tenemos:

1. Ir a Inicio
2. Programas
3. Herramientas de administración
4. Internet Information Server, obtendremos la Microsoft Management Console (figura 2), que es el panel de control para el IIS 5.0 desde donde administraremos, instalaremos y modificaremos nuestros Webs.

Por defecto cuando se instala el IIS 5.0 se crea:

- Sitio principal (Sitio Web Predeterminado)
- Sitio de administración, donde están las ayudas y las páginas de administración del IIS 5.0 en HTML, por seguridad este Sitio siempre se instala en otro puerto distinto al 80.
- Servidor SMTP



- Servidor NNTP para los grupos de noticias debe seleccionarse si se quiere instalar, al igual que el servidor FTP.

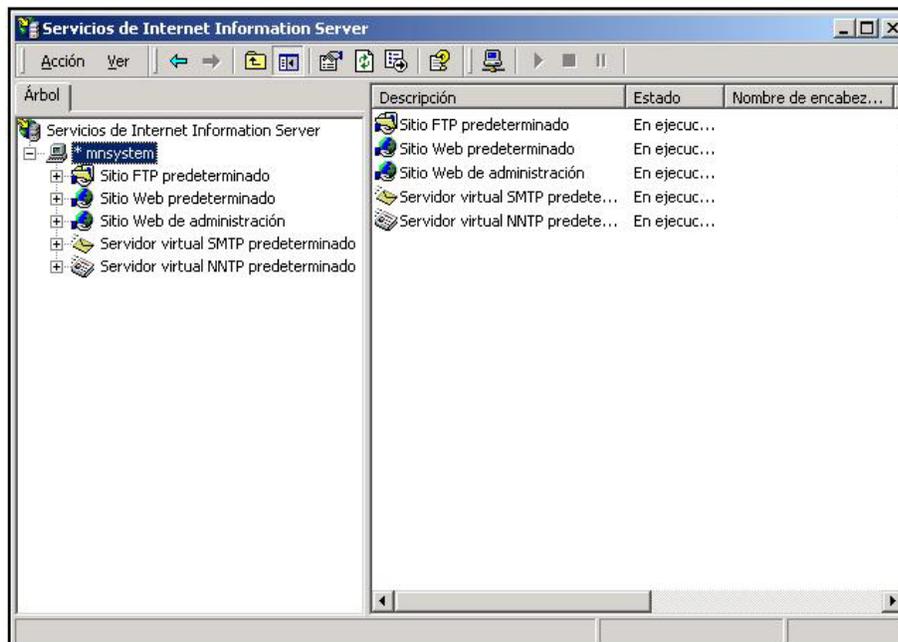


Figura 2

El IIS 5.0 por defecto se instala en los siguientes directorios:

- %Systemroot%\System32\InetSrv
- %Systemroot%\System32\InetSrv\IISAdmin
- C:\InetPub
- C:\Winnt\Help\IISHelp

El directorio del Web por defecto es c:\inetpub\wwwroot. También instala dos usuarios por defecto para el acceso anónimo a los Webs:

- IUSR_Nombre_de_Maquina
- IWAM_Nombre_de_Maquina



1.1 Configuración

En primer lugar nos debemos de asegurar de que el servicio Web está iniciado. Los servicios de Internet se pueden encontrar parados, en pausa o iniciados, solamente en este último estado los usuarios pueden obtener respuestas a sus peticiones de página.

La forma más simple de variar el estado del sitio Web es seleccionándolo en la consola y accediendo a *Iniciar*, *Detener*, *Pausa* accesibles haciendo clic con el botón derecho del ratón. También es posible utilizar los botones de acceso rápido situados en la barra de herramientas de la consola.

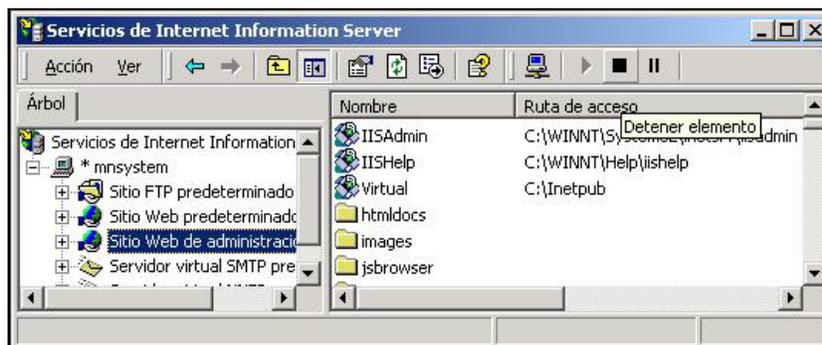


Figura 3

Cada sitio Web soportado por IIS tiene asignada una *hoja de propiedades*. Esta hoja de propiedades almacena la configuración del sitio Web en una estructura de datos llamada *metabase*.

Existen hojas de propiedades definidas en tres niveles:

- Sitio Web.
- Directorios de publicación.
- Archivo de información.

La configuración de cada nivel superior se hereda en los inferiores. Al cambiar una propiedad en un nivel, esta propiedad prevalece frente a la marcada en los niveles superiores.

Una vez instalado, como ya se ha comentado antes, se crea por defecto el "Sitio Web Predeterminado" y hay que configurarlo y optimizarlo, para eso con el botón derecho hacemos clic en sobre el "Sitio Web Predeterminado" y seleccionamos "Propiedades" (figura 4), aquí nos encontramos un panel con múltiples opciones: *Documentos*, *Seguridad de directorios*, *Encabezados HTTP*, *Errores personalizados*, *Sitio Web*, *Operadores*, *Rendimiento*, *Filtros ISAPI*, *Directorio particular*, éstas sirven para la configuración y optimización del Sitio. A continuación describiremos las opciones más importantes:

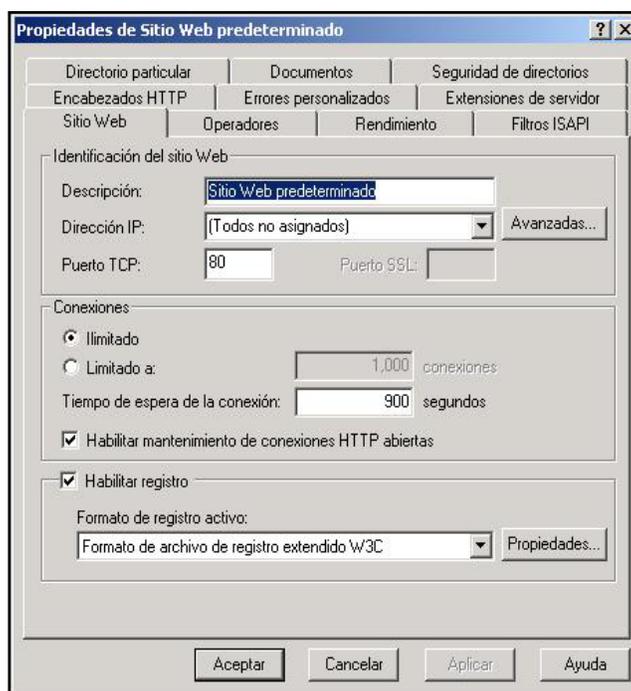


Figura 4

Sitio Web: esta opción nos va a servir para dar nombre a nuestro Sitio (en la consola del IIS) un nombre descriptivo para que nosotros identifiquemos el Web, para poner la IP del Web (si tenemos varias, recordemos que con el IIS 4.0 y 5.0 se pueden poner múltiples

Sitios con una misma IP). Se especifica el número de conexiones permitidas y se habilita o deshabilita el registro de actividades y datos generales de funcionamiento del sitio Web:

- *Identificación del sitio Web*
 - *Descripción:* en esta ventana se introduce el nombre que ha sido elegido para este sitio Web y que luego aparecerá como nodo de la consola.
 - *Dirección IP:* dirección IP del sitio Web. La opción “Todos no asignados” sirve para recoger las peticiones que se realicen a cualquiera de las posibles direcciones IP asignadas a esta máquina (en el caso de que tenga más de una). Mediante el botón de “Avanzadas”, nos permitirá colocar “el nombre del Web”, dicho en otras palabras poner la url: *www.reservabosawas.com* (sin el http://) (figura 4), o lo que es más preciso se establecerán Nombres de Encabezado de Hosts y de esta manera soportar varios nombres de dominios al mismo Web; otra característica de esta opción es que podemos cambiar el puerto que por defecto es 80 a otro que nos interese más; un ejemplo de esto sería: cambiamos el puerto a 81 y accederíamos al Web de la forma *http://www.reservabosawas.com:81* (figura 5).
 - *Puerto TCP:* dirección del puerto asignado al protocolo TCP (por defecto el 80).
- *Conexiones*
 - *Ilimitado:* el número de conexiones simultáneas al sitio Web no se limita.
 - *Limitado a:* el número de conexiones simultáneas al sitio Web se limita por razones de eficiencia al número establecido (demasiadas conexiones simultáneas puede hacer que el rendimiento del servidor sea muy pobre).

- *Tiempo de espera de la conexión (en segundos):* tiempo que se le concede a un usuario inactivo para que sea desconectado y de esta forma se liberen recursos del servidor.
- *Habilita registro*
 - *Formato de registro activo:* permite seleccionar el formato en el que se graban los datos registrados. Mediante el botón *Propiedades* se accesa a una ventana en la que se puede seleccionar; prioridad del registro de datos (diario, semanal, etc.), límite en el tamaño del archivo creado, ubicación del archivo de grabación, datos concretos que se desean registrar.

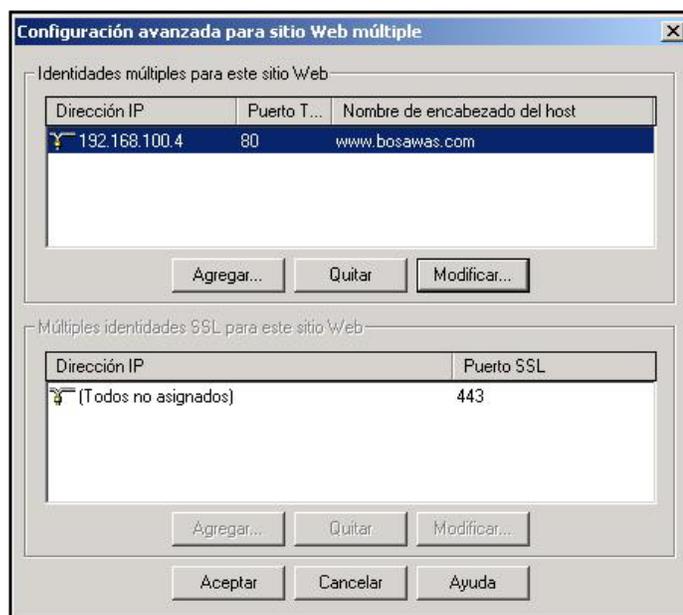


Figura 5

Operadores: por defecto los operadores del Web son los usuarios con permisos de administrador en Windows 2000 Server (figura 6). El IIS tanto la versión 4.0 como la 5.0 utilizan la seguridad del sistema operativo Windows NT para el IIS 4.0 y Windows 2000 Server o Advanced Server para IIS 5.0.



Figura 6

Rendimiento: también podemos optimizar el rendimiento del Sitio en función de los usuarios que esperamos que nos visiten (figura 7):

- *Ajustes de rendimiento:* este control de usuario nos permite indicar el número de referencias al sitio Web esperados por día, con el fin de que el servidor ajuste los recursos de una manera más apropiada.
- *Activar el límite de ancho de banda (Kb/s):* permite configurar un ancho de banda máximo para las comunicaciones relacionadas con el sitio Web. Esta opción nos permite racionalizar el ancho de banda y repartirlo entre diversos servicios de Internet.

Filtros ISAPI: (Internet Server Application Programming Interface) son programas (DLL por lo general y escritos en C++) que interactúan con el IIS, se utilizan para crear extensiones y filtros para el IIS.

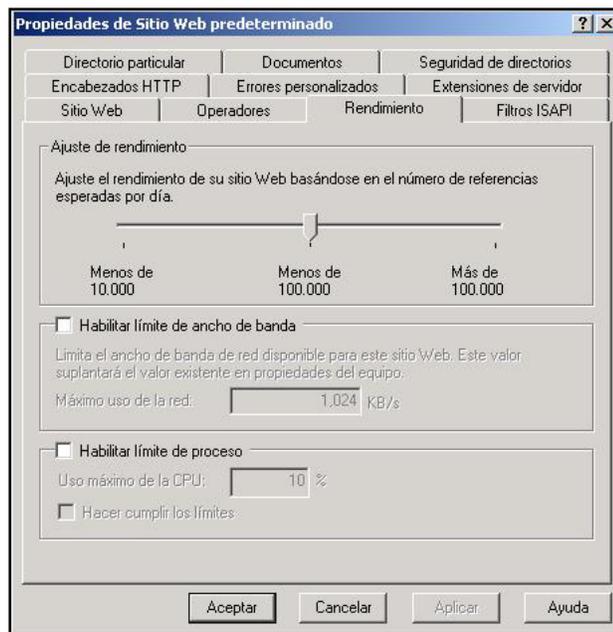


Figura 7

Directorio Particular: es la ubicación central del sitio Web publicado (figura 8). Los usuarios accesan a la página principal situada en el directorio principal, a través del nombre de dominio o del servidor (en el caso de Intranets).

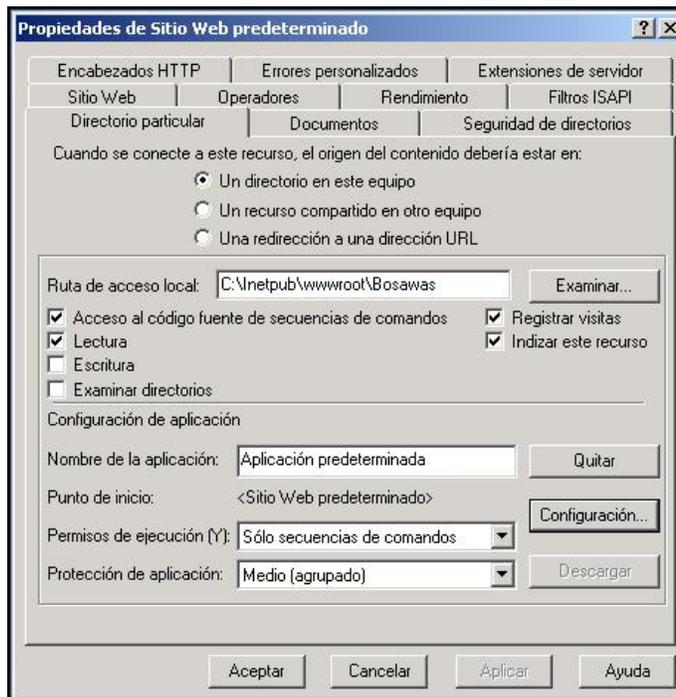


Figura 8



En el directorio principal se coloca una página que contiene enlaces al resto del sitio Web. Por defecto, al instalar IIS se crea un directorio particular situado en: *c:\inetpub\wwwroot*.

Si el usuario no solicita ninguna página en la URL que le proporciona a su explorador, IIS tomará por defecto la primera página que el administrador haya configurado con este fin. IIS coloca *default.htm* y *default.asp* en la instalación.

Desde este panel decidiremos donde se alojaran las páginas de nuestro Web, tenemos tres opciones:

- En mismo ordenador.
- En un ordenador de la red y accesamos al directorio como una unidad compartida.
- Redirigiremos a otra URL.

Por defecto y si no se cambia de opción las páginas físicas del Web se sitúan en *C:\inetpub\wwwroot* en el mismo servidor donde está el IIS.

También tenemos varias opciones interesantes:

- **Lectura:** el sitio dejará que las páginas sean accesibles para leer.
- **Escritura:** permitirá hacer Uploads al directorio que se especifica arriba.
- **Examinar directorios:** permite hacer un listado del directorio.
- **Registrar visitas:** crea un registro de los visitantes del sitio.
- **Permisos de ejecución:** configura la seguridad a nivel de archivos y páginas que se pueden ejecutar:
 - *Ninguno:* no ejecuta nada y sólo procesa páginas HTML.



- *Solo secuencias de comando*: permite la ejecución de páginas ASP, HTML y código de scripting.
- *Secuencia de comandos y ejecutables*: permite ejecutar código scripting y programas.
- **Protección de aplicación**: configura la seguridad a nivel de aplicación. Existen tres formas de ejecutar una aplicación Web:
 - *Bajo (proceso IIS)*: donde las aplicaciones Web corren todas juntas en el proceso junto a los servicios de Web y si una aplicación se cae, cae todo el servidor.
 - *Medio (agrupado)*: donde las aplicaciones corren juntas separadas de los servicios Web.
 - *Alto (aislado)*: donde cada aplicación corre en espacios separados de memoria y separadas de los servicios Web.

Mediante el botón de “Configuración” (figura 9) se accesa a una ventana desde la que se puede configurar la ejecución de las aplicaciones de Active Server Page.

La ventana tiene tres solapas que agrupan tres funcionalidades diferentes:

- **Opciones de aplicación**: estas opciones nos sirven para controlar la ejecución de las secuencias de comando ASP:
 - *Habilitar estado de sesión*: permite mantener el estado de las sesiones de los usuarios. ASP utiliza el objeto Session con el fin de mantener información respecto a los distintos usuarios a lo largo de sus accesos a la aplicación Web.

- *Tiempo de espera de sesión:* tiempo máximo que se mantiene la información de una sesión sin que el usuario realice nuevas peticiones (o actualizaciones) de página de la aplicación. Por defecto IIS permite 20 minutos.
- *Habilitar el buffer:* permite recopilar toda la salida de una página ASP antes de enviarla al explorador. Esta acción aumenta la eficiencia de la transmisión. El almacenamiento en buffer del desarrollo permite configurar encabezados HTTP desde cualquier parte de una secuencia de comando ASP.

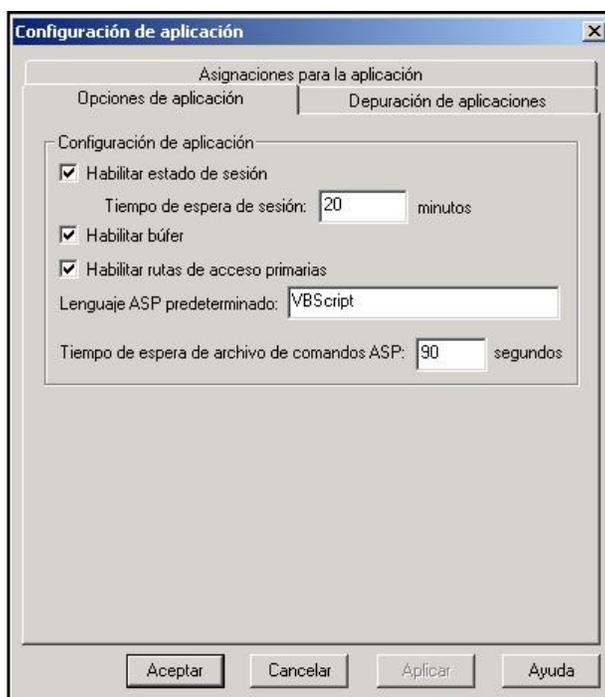


Figura 9

- *Habilitar rutas de acceso primarias:* con esta casilla activada, las instrucciones de ASP pueden hacer referencias a los directorios padre (..) de donde se encuentran los scripts.
- *Lenguaje ASP predeterminado:* establece el motor del lenguaje que se utilizará por defecto para interpretar los scripts de ASP. Las directivas `<% LENGUAJE`

= %> permiten introducir nuevos lenguajes en las páginas ASP. IIS establece Visual Basic Script como lenguaje predeterminado.

- *Tiempo del límite de archivo de comando ASP (segundos):* tiempo máximo que se permite a la ejecución de un archivo de comando ASP. El tiempo predefinido es 20 minutos, aunque este parámetro puede ser modificado por el programa mediante la instrucción: `Server.ScriptTimeot`. Establecer un tiempo límite de ejecución sirve para liberar al servidor de Web de scripts mal programados (por ejemplo con los bucles infinitos) o scripts que acaparan demasiado la CPU.
- **Depuración de aplicaciones:** permite configurar opciones de depuración (en servidor y cliente) de secuencias de comando ASP (figura 10):
 - *Habilitar depuración de secuencia de comando ASP en el servidor:* posibilita la actuación de Microsoft Script Debugger en las páginas ASP, y de esta forma poder hacer una depuración en el servidor.
 - *Habilitar depuración de secuencia de comando ASP en el cliente:* esta opción se reserva para las siguientes versiones de IIS.
 - *Enviar mensajes de error ASP detallados al cliente:* al habilitarse esta opción, se envían mensajes de error al equipo del cliente.
 - *Enviar este mensaje de error al cliente:* establece un mensaje por defecto, se le envía al cliente cuando se produce algún error en el procesamiento de las páginas ASP.



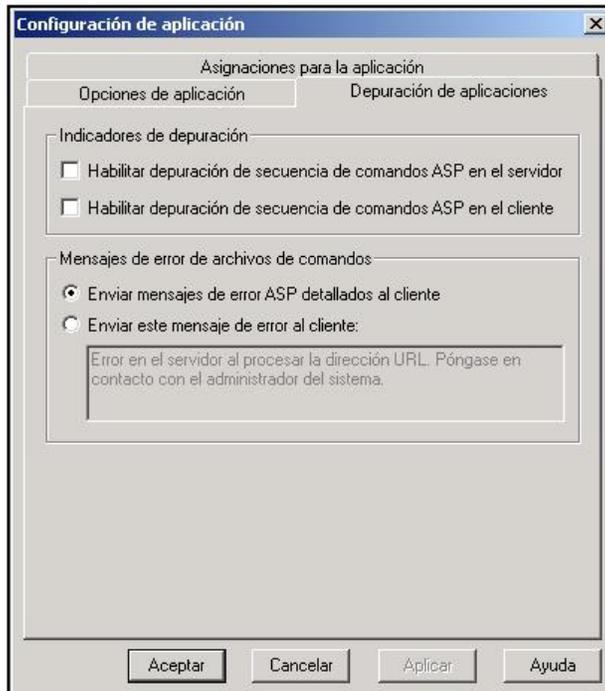


Figura 10

- **Asignaciones para la aplicación:** en esta ventana (figura 11) se pueden asociar extensiones de archivos a programas, de manera que se establezcan las relaciones (extensión de archivo ↔ aplicación que trata este tipo de archivo). Por ejemplo, podríamos asociar los archivos de extensión .dcp (documento personal) a una aplicación ASP creada por nosotros para tratar este tipo de archivos de “uso personal”:
 - *Almacenar aplicaciones ISAPI en caché:* por defecto, las aplicaciones ISAPI se almacenan en caché con el objetivo de evitar sucesivas cargas cada vez que son invocadas. Esta característica proporciona un alto grado de eficiencia a IIS. En algunas circunstancias excepcionales (como la depuración del código ISAPI) puede convenir desactivar esta opción.
 - *Asignaciones para la aplicación:* mediante los botones Agregar, Modificar y Quitar podemos mantener los pares (extensión ↔ aplicación).

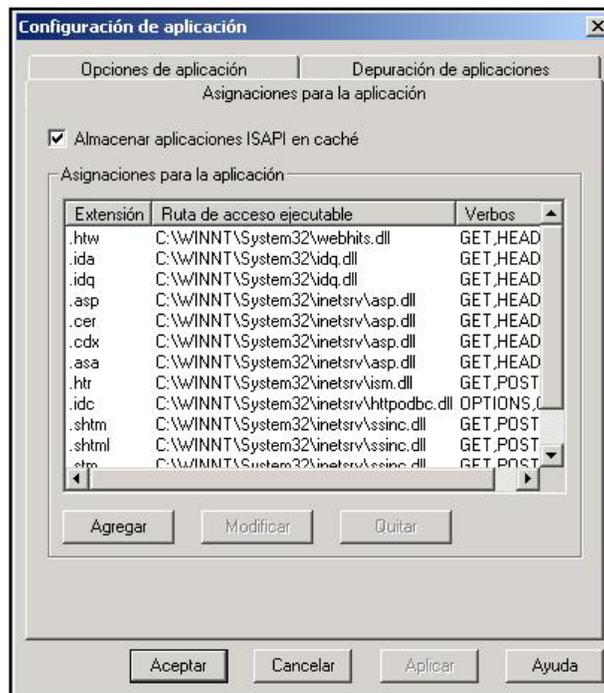


Figura 11

Documentos: en esta opción podemos poner un nombre a la página (figura 12) para que se ejecute automáticamente cuando se ponga la dirección del Web, dicho de otra forma nuestra página por defecto es *default.asp* cuando escribamos en el navegador: *www.minombre.com* el IIS mostrará por defecto la página *default.asp*.

- *Habilitar documento predeterminado:* si el usuario no establece un archivo en la URL que le proporciona al explorador, entonces IIS busca por orden cada uno de los documentos predeterminados que han sido incluidos en esta ventana, devolviendo al usuario el primero que encuentre.
- *Habilitar pie de página del documento:* habilita la inclusión de un pie de página de todas las páginas Web que envía el servidor. El formato del archivo que se inserta en al parte inferior de cada página debe ser instrucciones básicas HTML.

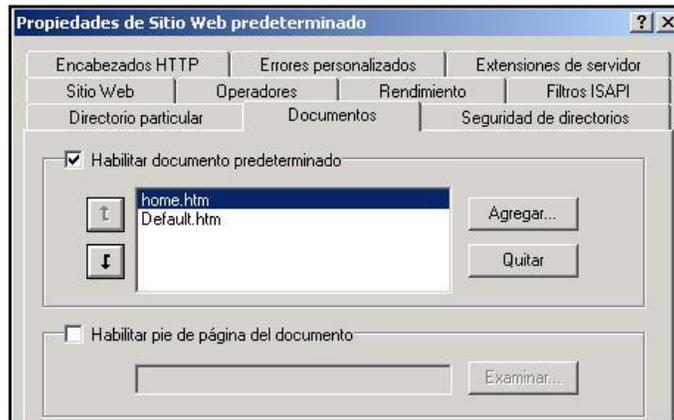


Figura 12

Seguridad de Directorios: la seguridad del Web viene integrada en la seguridad del Windows 2000 Server (figura 13), a través de esta opción daremos acceso al Web a todos los usuarios o restringiremos el acceso a todos los usuarios excepto al administrador y usuarios validados o podremos determinar que IP tienen acceso al Web y que IP no. Se pueden dar permisos para que determinados grupos y/o usuarios tengan acceso al Web (si el Web no es de dominio público), pero esto debemos hacerlo, a través del explorador de Windows con la seguridad de Windows 2000 Server.

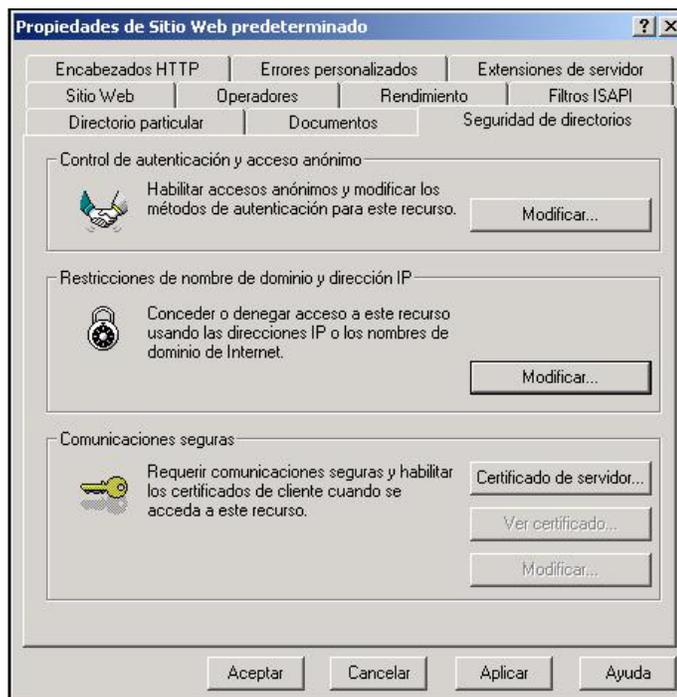


Figura 13

- Control de autenticación y acceso anónimo (figura 14)

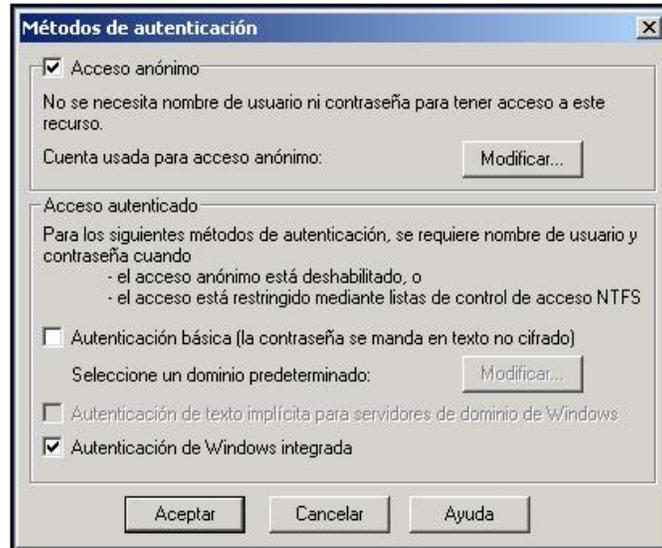


Figura 14

- *Permitir acceso anónimo:* la primera opción de seguridad es el acceso anónimo y la autenticación en el Web si pulsamos nos aparece lo que se muestra en la figura 6, a través de esta opción restringimos el acceso a todos los usuarios, excepto al administrador y usuarios permitidos, o se lo habilitaremos a todos dando permisos de acceso al usuario anónimo de Internet que se llama *IUSR_Nombre_de_Maquina* (la contraseña de este equipo la genera automáticamente el sistema por lo que no es conveniente tocarla). Debajo del acceso anónimo tenemos los métodos de autenticación, aquí es la opción más importante, es la segunda que si tenemos, un Web en una Intranet con acceso restringido a usuarios registrados, con esta opción chequeada los usuarios se validarán automáticamente con la seguridad de Windows 2000 Server.
- *Autenticación básica:* se pide el nombre de usuario y contraseña cuando la opción *Permitir acceso anónimo* está desactivada o bien cuando está activada y se deniega su acceso. El nombre de usuario y la contraseña se transmiten en

forma de texto, por lo que estos datos podrían ser interceptados con relativa facilidad.

- *Autenticación de Windows integrada:* es similar al de la autenticación básica, sin embargo, en este caso, el nombre del usuario y la contraseña se transmiten de forma cifrada.
- **Restricciones de nombre de dominio y dirección IP** (figura 15): esta opción nos permite aumentar la seguridad del sitio Web restringiendo el acceso en base a direcciones IP y dominios. Es posible permitir la entrada de todas las máquinas excepto las seleccionadas.



Figura 15

Estas opciones son las más importantes en el momento de configurar un servidor *Sitio Web*, existen algunas que permiten modificar y personalizar más el *Sitio Web*.

1.2 Nuevos Webs

La creación de nuevos Webs es bastante sencilla, sólo hay seguir los pasos:

1. Crear un nuevo Web. Con el botón derecho del ratón pulsamos encima del nombre de la máquina y seleccionamos nuevo Sitio Web (figura 16).

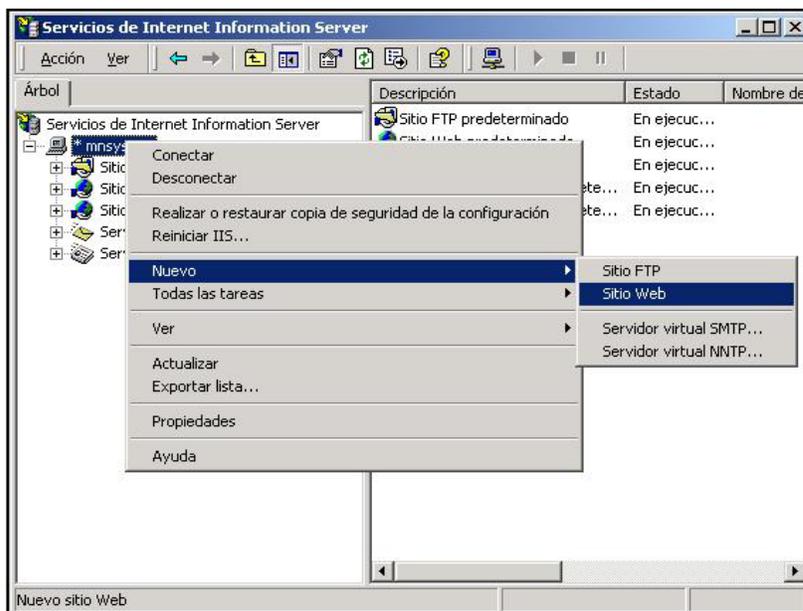


Figura 16

2. Poner un nombre para identificarlo en la MMC (figura 17).

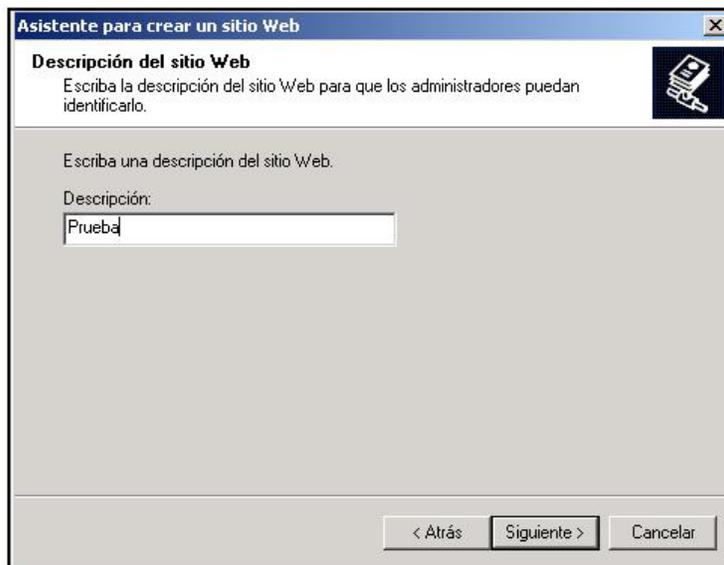


Figura 17

3. Asignarle una IP, puerto (por defecto el 80) y asignarle un Nombre de Dominio (figura 18), es decir, asignarle un nombre de dominio. Si trabajamos en un entorno local o de Intranet, el Nombre de Dominio no es necesario ponerlo.

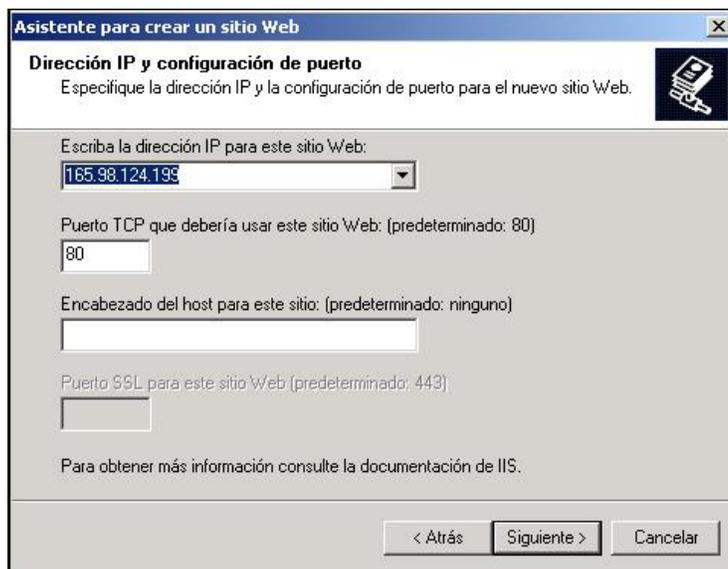


Figura 18

4. Asignarle una ruta física en disco (figura 19). Es decir, asignarle un directorio en disco. Recordemos que el “Web por defecto” del IIS se crea en C:\inetpub\wwwroot (ruta que puede cambiarse si se desea). Para no complicarnos y tener separados los Webs podemos crear un directorio root para contener todos los Webs que queramos crear, así nuestra ruta nos quedaría C:\inetpub\root\pruebas, siendo pruebas nuestro directorio para esta aplicación Web.

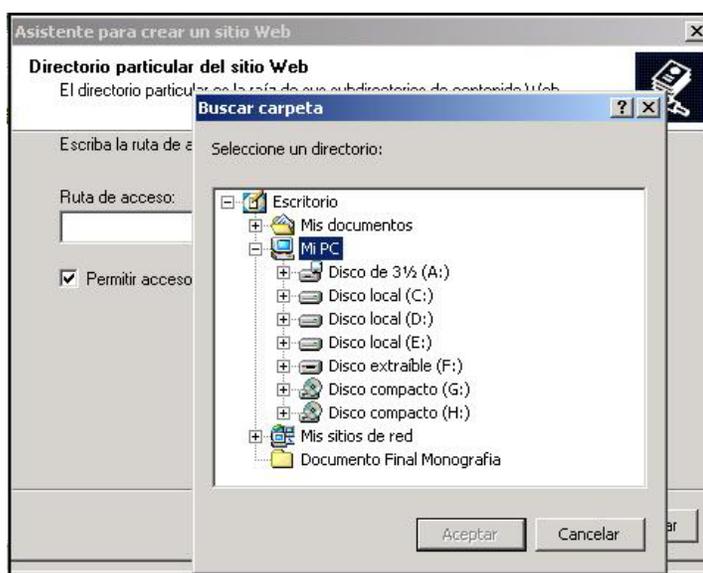


Figura 19

5. Seleccionar los permisos de ese Web (figura 20). Los permisos de ejecución de páginas, si ejecuta Scripts, dll o exe o sólo páginas asp.

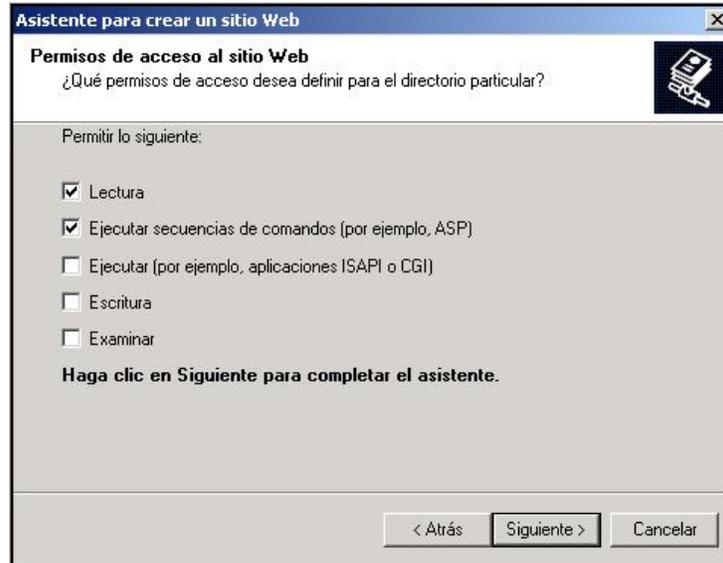


Figura 20

Una vez creado el nuevo Web, hay que instalar las extensiones de Front Page como se explica en el siguiente punto y configurar el Web como está explicado en el punto anterior.

1.3 Instalación de las extensiones de Front Page

El Web las trae instaladas por defecto, pero cada vez que creemos un nuevo Web hay que instalarlas aunque no es necesario hacerlo si ese Web no va a utilizar Front Page o Visual Interdev y sólo se conectarán a el clientes FTP. Como servidor Web funcionará igual con o sin las extensiones.

¿Por qué instalar las extensiones de Front Page?

Es probable que el administrador de sistemas o la persona responsable del servidor no las utilicen o no las necesite, pero si no se instalan las extensiones no se podrá tener acceso al Web mediante Front Page o mediante Visual Interdev para desarrollar.

Para instalar las extensiones de Front Page se realizan los siguientes pasos (figura 21):

1. Clic con el botón derecho del ratón sobre el nuevo Web que hemos creado.
2. Seleccionamos Todas las tareas
3. Configurar Extensiones de Servidor, a partir de aquí nos van a aparecer una serie de pantallas de ayuda para configurar las extensiones.

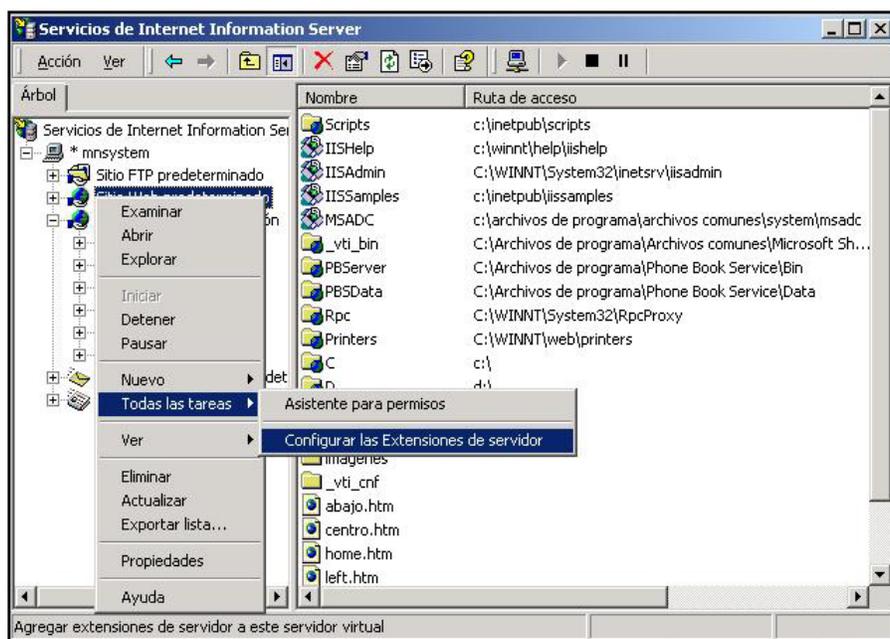


Figura 21

Creación de los grupos de Windows.

Por defecto cuando instalamos las extensiones de Front Page, el IIS nos va a sugerir crear un grupo específico para ese nuevo Web, pero esto no es aconsejable utilizarlo por lo que deschequearemos esta opción (figura 22).

No es aconsejable, porque esta opción lo que va hacer es crear un grupo de usuarios dentro de la administración de usuarios de Windows 2000 Server con lo que se nos creará un grupo por cada Web que creamos y es muy probable que algunas de estas Web no utilicen grupos, ni usuarios validados en el Web, es decir, que sea un Web para todo el público. Por

otro lado, es más aconsejable ir creando grupos desde el administrador de Windows 2000 Server y saber qué creamos y dónde vamos a utilizarlos.

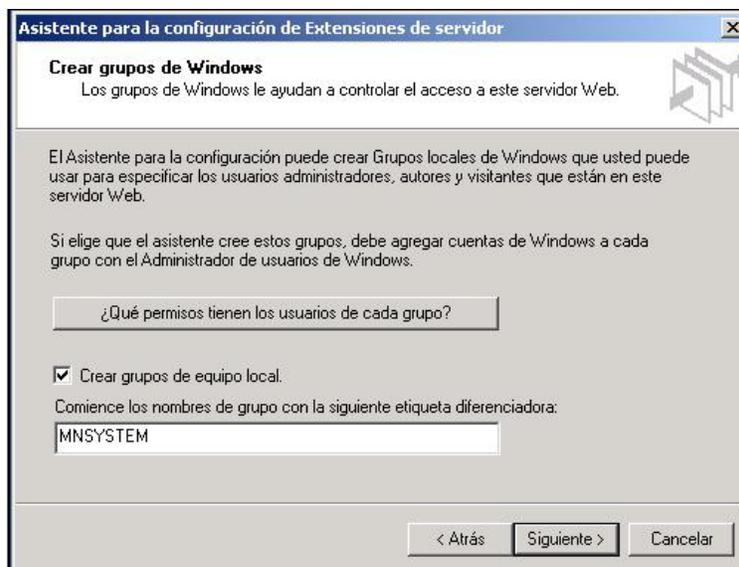


Figura 22

Control de Acceso.

Aquí daremos permiso a los grupos que administran el Web, por defecto es el grupo de administradores el que tiene permisos de administración (figura 23).



Figura 23

Mail Server.

Y con esto quedan instaladas las extensiones de Front Page (figura 24).



Figura 24

1.4 Directorios Virtuales

Los pasos siguientes explican como crear un directorio virtual ejecutable en Windows 2000 Server:

1. Desde el botón Inicio, seleccione Programas, Herramientas Administrativas y Administrador de Servicios de Internet (también servirá cualquier configuración de la MMC que incluya el complemento de Servicios de IIS).
2. Si la computadora que va a contener el directorio virtual ejecutable no aparece en Servicios de IIS, haga clic en el botón derecho sobre los Servicios de IIS, seleccione "Conectar", introduzca el nombre de la computadora y haga clic en aceptar.



Figura 26

6. Haga clic sobre el botón “Siguiente” y después, en el tercer panel del asistente, introduzca el Directorio de contenido del sitio Web. Puede ser cualquier directorio que exista físicamente en el servidor o en un lugar con partición de archivos especificando en el formato del Convenio Universal de Asignación de Nombres (UCN; Universal Naming Convention).
7. Haga clic sobre el botón “Siguiente” de nuevo para acceder a la pantalla (figura 27) que ofrece las siguiente opciones de seguridad para el nuevo directorio virtual:
 - **Lectura:** especifica que el servidor Web puede responder a solicitudes GET ordinarias de archivos del directorio virtual. Es decir, puede entregar archivos HTML, GIF, JPEG y similares de la manera habitual.
 - **Ejecutar secuencias de comandos:** especifica que el servidor web puede responder a solicitudes GET y POST que especifican una ubicación dentro del directorio virtual y que provocan que un intérprete de secuencias de comando

ejecute páginas web que contienen secuencias de comandos en el lado del servidor. Para que las páginas Active Server funcionen adecuadamente, *deberá* marcar esta opción.

- **Ejecutar:** especifica que el servidor web puede ejecutar cualquier archivo de programa (como los archivos .exe y .dll) que se encuentren dentro del directorio virtual y se especifiquen como parte de un URL.
- **Escritura:** especifica que el servidor web puede recibir archivos de los visitantes web y crear y sustituir los archivos correspondientes en el directorio virtual. Ésta es una opción que se emplea muy poco, excepto cuando los diseñadores web utilizan ciertos editores de HTML.
- **Examinar:** especifica que cuando un visitante web envía una URL sin nombre de archivo y el trayecto especificado no contiene una página web determinada, el servidor web debe responder con un listado de los archivos de ese directorio sobre los que se puede hacer clic.

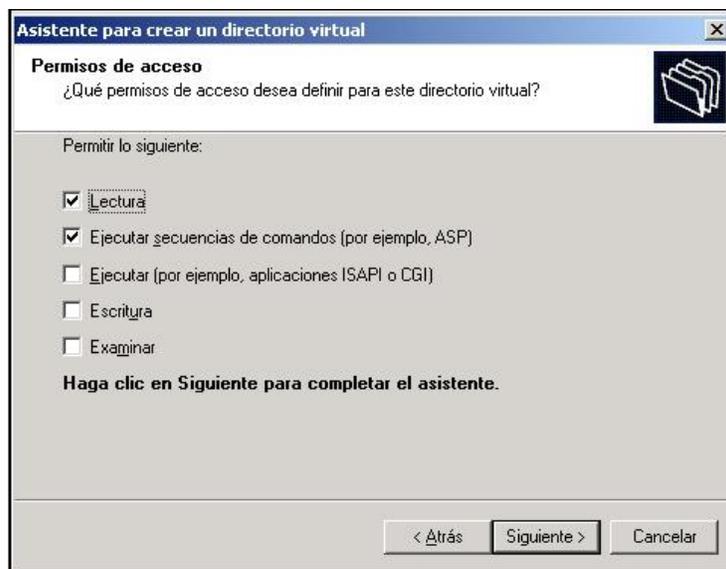


Figura 27

8. Haga clic sobre “Siguiete” una vez más para acceder a la página final del asistente, y después haga clic sobre “Finalizar”. Al hacerlo el nuevo directorio virtual debe aparecer bajo la entrada del servidor web de la MMC.

9. Para revisar o modificar sus resultados haga clic con el botón derecho sobre el nuevo directorio virtual en la MMC y después seleccione “Propiedades” en el menú contextual. Esta acción presentará el cuadro de diálogo (figura 28). Al hacer clic sobre el botón de “Configuración”, en la cual la primera opción de la ficha “Depuración” de aplicaciones activa o desactiva la depuración interactiva de las secuencias de comandos.

10. Como es lógico, el cuadro de diálogo “Propiedades de directorio Virtual” (figura 28) contiene una gran cantidad de opciones que el Asistente para crear un directorio virtual, no configura. Por suerte, la configuración predeterminada es perfectamente aceptable para la mayor parte de los casos. Si no lo es, puede obtener una ayuda adicional haciendo clic sobre el botón “Ayuda” o consultando un libro de IIS. Sin embargo, una opción merece una mención especial. Al hacer clic sobre el botón “Configuración” (parcialmente oculto en la figura 28) y seleccionar después la ficha “Depuración de aplicaciones” se accesa a la siguiente opción:
 - **Habilitar depuración de secuencias de comando ASP en el servidor:** debe activarla si desea depurar de manera interactiva sus aplicaciones ASP.

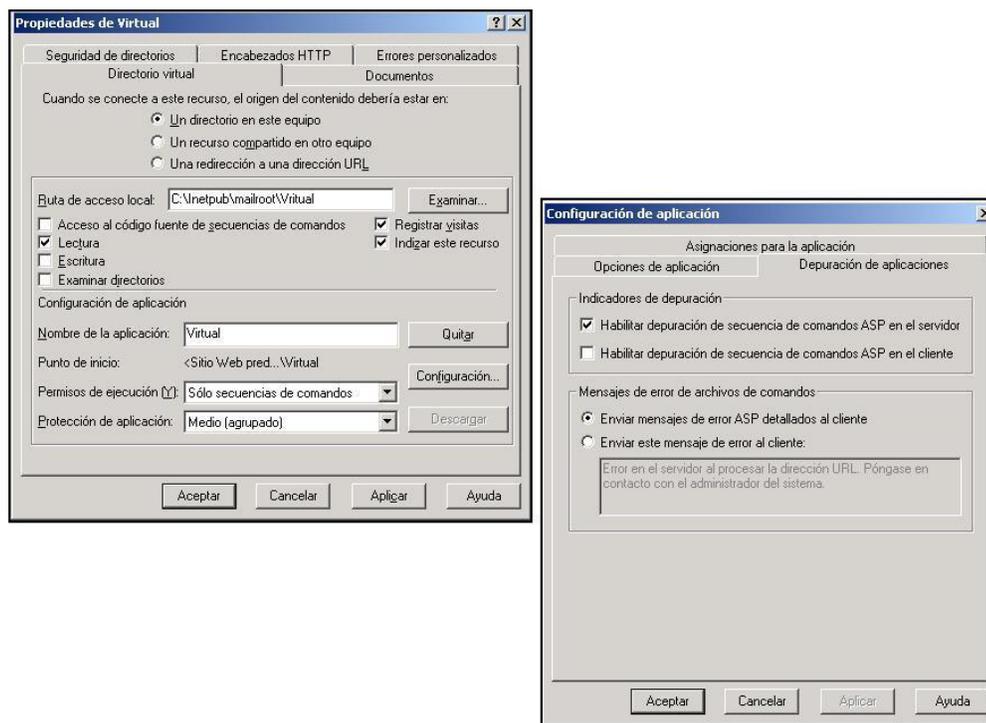


Figura 28

¿Cómo convertimos un directorio existente en directorio virtual?

Nos posicionamos con el ratón en el directorio que queremos cambiar a directorio virtual, un clic con el botón derecho del ratón y seleccionamos propiedades, aquí tenemos que pulsar el botón “Crear” y convertiremos nuestro directorio normal en virtual.

Para eliminarlo, es decir, dejar de ser directorio virtual y pasar a ser directorio normal, pulsamos el botón llamado eliminar, esto no quiere decir que borraremos el contenido del directorio, pulsando este botón el directorio y su contenido seguirán existiendo, pero como un directorio normal y corriente sin acceso a él desde un cliente Front Page o Visual Interdev.



1.5 Servidor Virtual

Con IIS se pueden establecer varios dominios sobre un mismo servidor. Existen dos formas diferentes de hacerlo:

1. Asignar una dirección IP diferente para cada dominio.
2. Usar Cabeceras de Host (nombres lógicos) sobre una misma dirección IP (sólo en el servicio WWW).

Los servidores virtuales nos permiten alojar varios sitios Web en una misma máquina y con una sola dirección IP, consiguiéndose de esta manera una administración más sencilla y una utilización de medios físicos más adecuada.

Hay que tener en cuenta que el uso de las cabeceras de servidor requiere como soporte el protocolo HTTP, y por lo tanto (desde el punto de vista del usuario) un explorador que lo reconozca, por ejemplo Microsoft Internet Explorer 4.0 o superior.

1.6 Seguridad

La seguridad de un Web, a través de la MMC, viene restringida a poder decirle al Web “este Web lo pueden ver todos los usuarios, o este Web sólo lo utilizan los administradores (figura 1), o sólo lo ven determinadas IP”, esto es así porque la seguridad del IIS va asociada a la seguridad de Windows 2000 Server, por lo tanto para permitir y denegar accesos a determinadas partes del Web como poder acceder a directorios virtuales o poder enviar ficheros al Web, se ha de utilizar la seguridad de Windows, a través de los grupos y usuarios.



Supongamos el siguiente escenario de trabajo

Tenemos un Web con un directorio virtual llamado “musica” que sólo queremos que ciertos usuarios puedan acceder a él. Lo primero que tendremos que hacer es ir a las propiedades del directorio virtual desde la MMC, clic con el botón derecho del ratón y seleccionamos propiedades y la opción Seguridad del Directorio y lo dejamos tal como muestra la figura 24 con lo que conseguiremos que los usuarios anónimos, que no estén identificados en el servidor, no puedan acceder al directorio, pero podrán acceder al resto del Web, y a este directorio sólo accesan los usuarios validados en el Windows 2000.

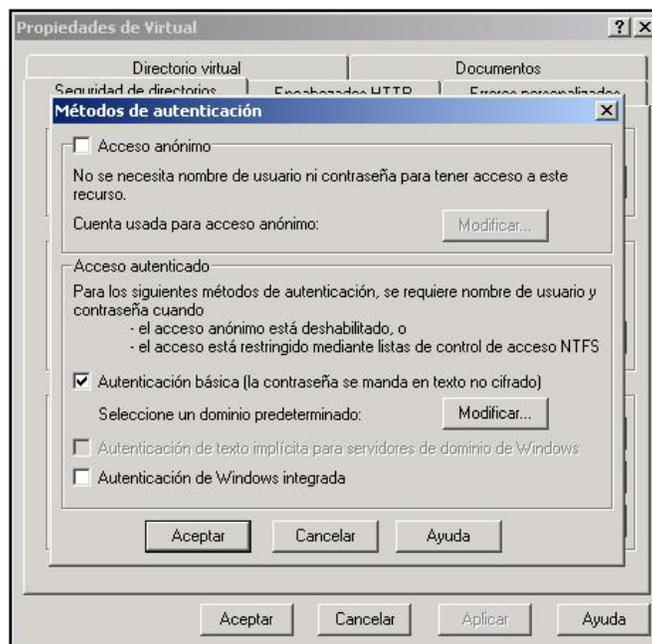


Figura 29

Con esta opción chequeada sólo conseguimos que los usuarios con permisos de administrador puedan acceder al Web, para dar permisos de acceso a otros usuarios deberemos ir al explorador de Windows y situarnos en el directorio físico, clic con el botón derecho del ratón y seleccionamos la opción de propiedades y ahí la opción seguridad (figura 30). Aquí deberemos añadir o eliminar los usuarios o grupos de usuarios que querramos que tengan acceso al Web.

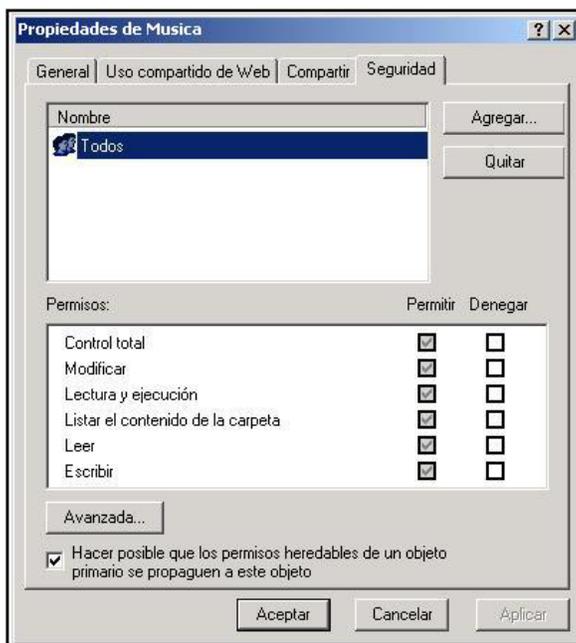


Figura 30

Los usuarios y/o grupos en Windows 2000 Server (figura 31) se crean, a través de la opción Administrador de Equipos, desde:

1. Inicio.
2. Programas.
3. Administrador de Equipos y seleccionando Usuarios Locales y Grupos.

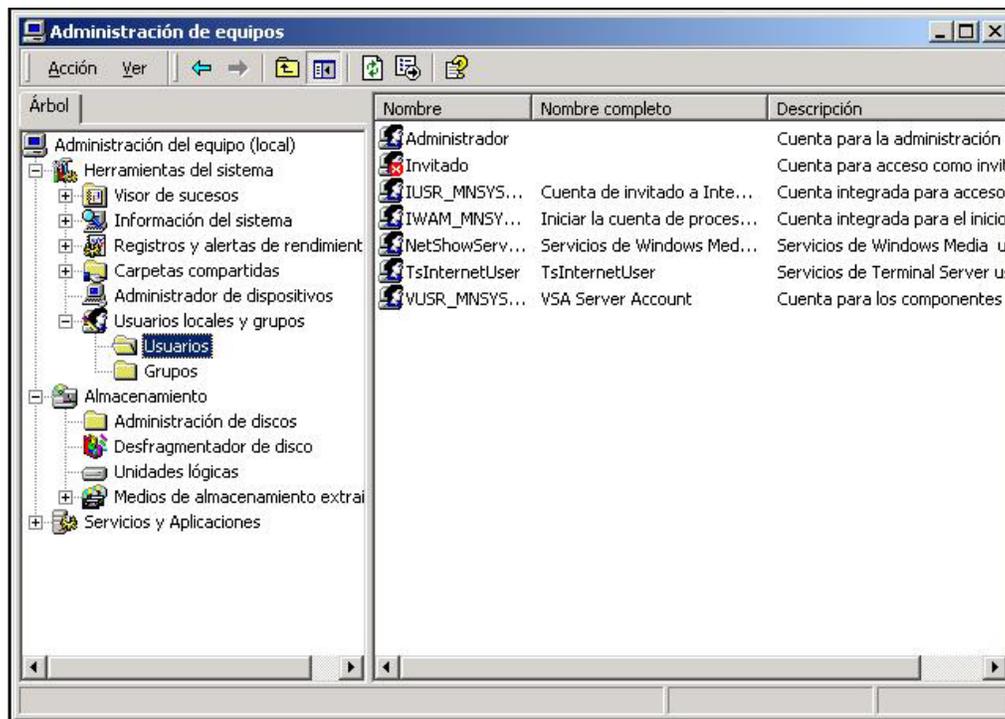


Figura 31

2. Alojamiento del Sitio Web en Internet

En este proceso debe de elegirse el dominio en el que se alojará el sitio. Esto debe hacerse contratando un proveedor de nombres de dominio y un alojamiento web que permita la ejecución de sentencias de comando ASP (Visual Basic Script).

2.1 Plan de Alojamiento

El primer paso fue contratar un plan de alojamiento que nos permitiera disponer del espacio necesario para nuestra aplicación. Después de realizar las cotizaciones de varios proveedores hemos elegido uno que nos ofrece un plan de alojamiento con las características siguientes:

Características:	
Espacio Web	30 MB
Transferencia Mensual (Bandwith)	500 MB
Cuentas POP3	5
Webmail	SI
Panel de Control	SI
Cuentas FTP	1
Acceso FTP	Ilimitado
Ext. FrontPage 2002	SI
Estadísticas Diarias Online	SI
Directorios Seguros	SI
Soporte SSI	SI
Dirección IP exclusiva	SI
Sub-Dominios	Opcional
Errores del Servidor Personalizables	SI
Conectores ODBC	Ilimitados
Soporte PHP3 / PHP4	SI
Soporte ASP VBScript 5 .NET	SI
Base de datos MS Access	SI
Base de datos Pyton	-
Base de datos MySQL	-
Soporte CGI / PERL 5.5.6.1	SI
ASPEMAIL de Persits	SI
Soporte Midi / Real Audio	Opcional
Soporte Macromedia Flash	SI
Soporte SSL (128 bits)	Opcional
Garantía de Devolución	SI

Las características de los servidores que nos ofrece el proveedor son las siguientes:



Hosting en Servidores IBM xSeries:		
Procesador	Intel Pentium III de 1 GHZ.	
Memoria:	1 GB RAM 133 MHZ SDRAM ECC	
Disco Duro:	36 GB Ultra 160 SCSI HDD	
Red:	Dual Ethernet 100 MB/seg.	
Sistema Operativo:	Microsoft Windows 2000 Server.	

Con este tipo de servidores se satisfacen los requisitos mínimos de hardware establecidos al inicio del desarrollo del documento.

2.2 Datos del dominio adquirido.

Finalmente, nuestro proveedor nos concede el dominio y estamos listos para el montaje del sitio, ya sea via FTP o por el panel de control que ofrece a través del web.

Datos de Acceso al Panel de Control	
URL del Panel de Control:	http://www.bosawas.org/admin/
Nombre de Usuario:	
Password o Contraseña:	
Datos del Dominio y Acceso FTP	
Nombre de Dominio:	bosawas.org
Dirección IP:	69.0.236.83
Dirección FTP:	ftp.bosawas.org
Directorio Remoto FTP:	bosawas.org
Nombre de Usuario FTP:	
Password o Contraseña FTP:	
Datos de Acceso al Sistema de Mail	
Servidor POP3 y SMTP:	mail.bosawas.org
URL de Panel de Control de Mail:	http://www.zonahosting.net/meadmin/
URL del WEBMAIL:	http://www.bosawas.org/mewebmail/
Nombre de Usuario para el Administrador de Mail:	
Password o Contraseña:	
Datos de Acceso a Estadísticas On-Line	
URL de las Estadísticas:	http://www.bosawas.org/stats/
Nombre de Servidores:	
Nombre de Servidor Primario:	ns.zonahosting.net
Nombre de Servidor Secundario:	ns2.zonahosting.net

El acceso al sitio en cuestión será, entonces, mediante la dirección <http://www.bosawas.org>



XIII. CONCLUSIONES

En base a los objetivos planteados al inicio del proyecto, podemos asegurar que hemos logrado cumplir con las metas propuestas. Se ha diseñado el sitio con acceso a datos el cual resuelve de manera consistente las peticiones de listados de especies.

Se ha logrado diseñar la base de datos con respecto a la clasificación de los seres vivos, se han proporcionado elementos de animación al sitio, se ha cumplido con las consultas establecidas en la Especificación de Requisitos de Software y, finalmente, este documento podrá servir como guía para el aprendizaje de desarrollo de sitios web con acceso a datos.

En resumen, se han cumplido los objetivos propuestos.



XIV. RECOMENDACIONES

Para la introducción de los datos de nuevas especies no se cuenta una aplicación específica. Por lo tanto recomendamos el desarrollo de una aplicación que permita la introducción de datos de nuevas especies y - en general - la administración de la base de datos de las especies.

También será necesario una actualización y mantenimiento de manera periódica del sitio para no desentonar con las nuevas tecnologías que asomen en esta ciencia de empuje vertiginoso y proporcionar siempre información actualizada con respecto al eje central del sitio: *La Reserva de Biósfera de Bosawás*.



XV. Glosario

- *Servidor World Wide Web*: proporciona a los visitantes Web acceso a páginas Web normales, páginas ASP y otros programas en el lado del servidor por medio de HTTP.
- *ASP*: es una sencilla facilidad de creación de secuencias de comando en el lado del servidor que Microsoft incluye en todos sus servidores Web. Cuando un visitante Web solicita una página que contiene secuencias de comando en el lado del servidor, la secuencia de comandos personaliza la página antes que abandone el servidor.
- *ADO (Objetos de datos ActiveX) de Microsoft*: proporciona una interfaz de programación entre una página Web con bases de datos y diversos orígenes de datos. Igual que el origen de datos subyacente, ADO también proporciona una amplia variedad de objetos, métodos y propiedades.
- *IIS (Internet Information Server)*: es un servidor Web que es capaz de ejecutar varios motores de script como: ASP, PHP, entre otros. Convierte la máquina en la que se ejecuta en un servidor FTP, servidor de grupo de noticias o en un servidor de envío de correo electrónico. Para la utilización de páginas Web dinámicas IIS soporta la comunicación con programas externos mediante CGI o ISAPI, y por supuesto, da soporte a la tecnología de Microsoft ASP.
- *Microsoft Visual Basic Scripting Edition (VBScript)*: es el lenguaje de programación preferente para las páginas Active Server.
- *CGI (Pasarela de Interfaz Común)*: son programas independientes, escritos en cualquier lenguaje de programación los cuales ejecutan mandatos para generar una salida HTML.



- *ISAPI (Interfaz de Programación de Aplicaciones de Servidores de Internet)*: que hace que el código ASP sea ejecutado por un módulo que se encuentra dentro del propio Servidor Web.
- *DNS*: es una abreviatura para Sistema de nombres de dominio (*Domain Name System*), un sistema para asignar nombres a equipos y servicios de red que se organiza en una jerarquía de dominios. La asignación de nombres DNS se utiliza en las redes TCP/IP, como Internet, para localizar equipos y servicios con nombres sencillos. Cuando un usuario escriba un nombre DNS en una aplicación, los servicios DNS podrán traducir el nombre a otra información asociada con el mismo, como una dirección IP.
- *ODBC*: (Open Database Connectivity) es un programa de interface de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos, tanto relacional como no relacional, utilizando para ello SQL (lenguaje de consulta estructurado).
- *OLEDB*: es una tecnología que tiene como objetivo resolver ciertas restricciones de ODBC. Autoriza el acceso a todo tipo de datos, permitiendo administrar el aspecto de tener distribuidas las fuentes de datos, y tiene en cuenta las restricciones de la web.
- *LAN*: Son redes de propiedad privada, de hasta unos cuantos kilómetros de extensión. Se usan para conectar computadoras personales o estaciones de trabajo, con objeto de compartir recursos e intercambiar información. Están restringidas en tamaño, lo cual significa que el tiempo de transmisión, en el peor de los casos, se conoce, lo que permite cierto tipo de diseños (deterministas) que de otro modo podrían resultar ineficientes. Además, simplifica la administración de la red.
- *WAN*: Son redes que se extienden sobre un área geográfica extensa. Contiene una colección de máquinas dedicadas a ejecutar los programas de usuarios (hosts). Estos están conectados por la red que lleva los mensajes de un host a otro. Estas LAN de host acceden a la subred de la WAN por un router. Suelen ser por tanto redes punto a punto.



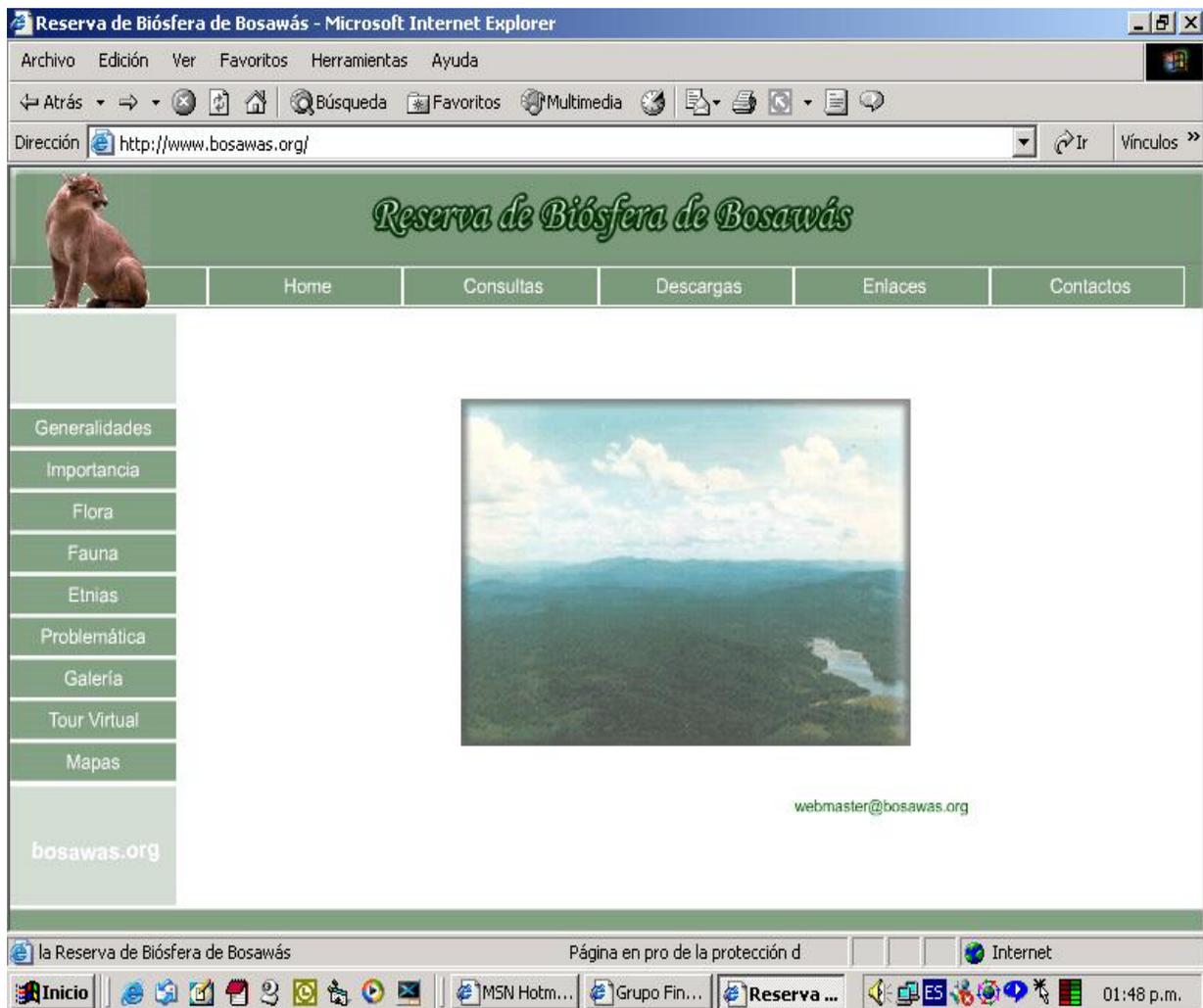
- *ISP*: Internet Service Provider (Proveedor de Servicios Internet). Los ISP han aparecido por doquier, para actuar como su interfaz (idealmente) amigable para todo lo que Internet ofrece. La mayoría de los ISPs tienen una red de servidores (e-mail, Usenet, Web, etc.), routers y modems conectados a una línea de conexión permanente y de gran velocidad a algún "backbone" de Internet. Suscriptores llaman la red local para obtener acceso a Internet sin tener que mantener servidores, archivos de nombres de dominios, y sin tener que aprender UNIX.
- *Endémica*: Especie vegetal o animal confinada en su distribución a un área natural restringida. Propio del lugar, como autóctono pero muy restringido en su dispersión. Constituye un ejemplo el cóndor de los Andes. Es el más majestuoso de la especie.
- *Biosfera*: La parte de la Tierra en la que habitan seres vivos. También se aplica a la porción de la Tierra y de la atmósfera donde puede existir vida. Está constituida por la capa más superficial de la corteza terrestre (litosfera), por los océanos (hidrosfera) y por las capas más bajas de la atmósfera. A la biosfera se la considera como la única zona donde es posible que se desarrolle la vida en forma normal o por especies adaptadas a la misma. En ella, se encuentran todos los elementos indispensables como el agua, aire, presión, suelo, alimentos, etc.
- *Biodiversidad*: Este es el término utilizado para describir la riqueza de vida animal y vegetal que existe en el planeta. Los científicos de vanguardia y los defensores del ambiente consideran que el mantenimiento de la mayor variedad posible de formas de vida no es solamente una cuestión moral relacionada con la protección de especies en peligro sino que también es de vital importancia en términos de supervivencia misma del planeta y de la calidad de vida de sus habitantes.



Anexos



A. Pantalla de Presentación



B. Pantalla de Consulta Animales

Reserva de Biósfera de Bosawás - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://www.bosawas.org>

Reserva de Biósfera de Bosawás

Home Consultas Descargas Enlaces Contactos

Consultas disponibles para especies del Reino Animal

- [Búsqueda de especies restringidas por Filo o División](#)
- [Búsqueda de especies restringidas por Filo \(o División\) y Clase](#)
- [Búsqueda de especies restringidas por Filo \(o División\), Clase y Orden](#)
- [Búsqueda de especies restringidas por Filo \(o División\), Clase, Orden y Familia](#)
- [Búsqueda de especies restringidas por Filo \(o División\), Clase, Orden, Familia y Género](#)
- [Búsqueda Libre \(por frase de búsqueda\)](#)

Generalidades

Importancia

Flora

Fauna

Etnias

Problemática

Galería

Tour Virtual

Mapas

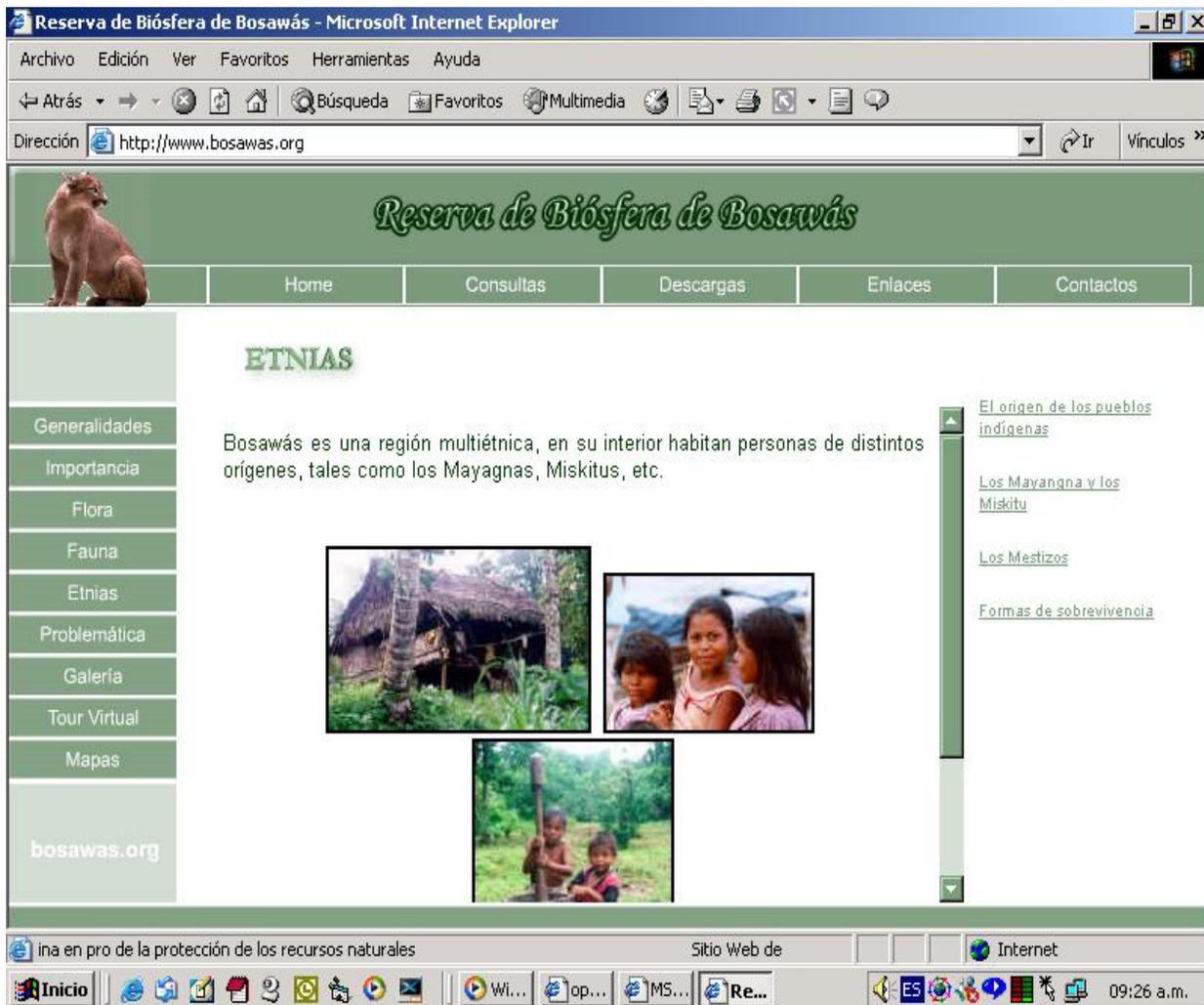
bosawas.org

Página en pro de la protección de los recursos naturales

Internet

Inicio Wi... op... MS... Re... 09:27 a.m.

C. Enlace Etnias



XVI. Bibliografía

Libros

Buyens, Jim

Aprenda Desarrollo de Bases de Datos Web Ya

Editorial McGraw Hill, Microsoft Press

ISBN 84-481-2903-2

2001

Scot Hillier, Mezick Daniel

Programación de Active Server Pages

Editorial McGraw Hill, Microsoft Press

ISBN 84-481-1466-3

Ville, Claude A.

Biología. Séptima Edición

Editorial McGraw Hill.

ISBN 0-7216-9023-8

1991

Información en la Web

La información correspondiente a ASP se ha encontrado dispersa en la web en varios sitios, entre los que destacamos:

<http://www.asptutor.com>

<http://www.desarrolloweb.com>

<http://www.aspfacil.com>

<http://portalvb.com>

Con respecto a Internet Information Server debemos destacar la URL siguiente:

<http://desarrollo.ethek.com/desarrollo/contenido.asp?IDContenido=86>

