

# Acceso a bases de datos

# Introducción

- Ni ASP ni el servidor IIS tienen por defecto asociado ningún **SGBD**.
- Permite asociar Bases de Datos hechas en
  - SQL Server
  - Microsoft Access
- Será necesario por tanto indicarle qué Base de Datos vamos a utilizar.

# Creando una Base de Datos

- En nuestro caso, por rapidez y sencillez vamos a hacer uso de una Base de Datos creada en **Microsoft Access**
- Por tanto, vamos a crear una Base de Datos que tenga las siguientes tablas:
  - Alumnos (ID, nombre, edad)
  - Asignaturas (ID, nombre, créditos)
  - Calificaciones (Alumno, Asignatura, año, nota)
- Introducir los siguientes datos:

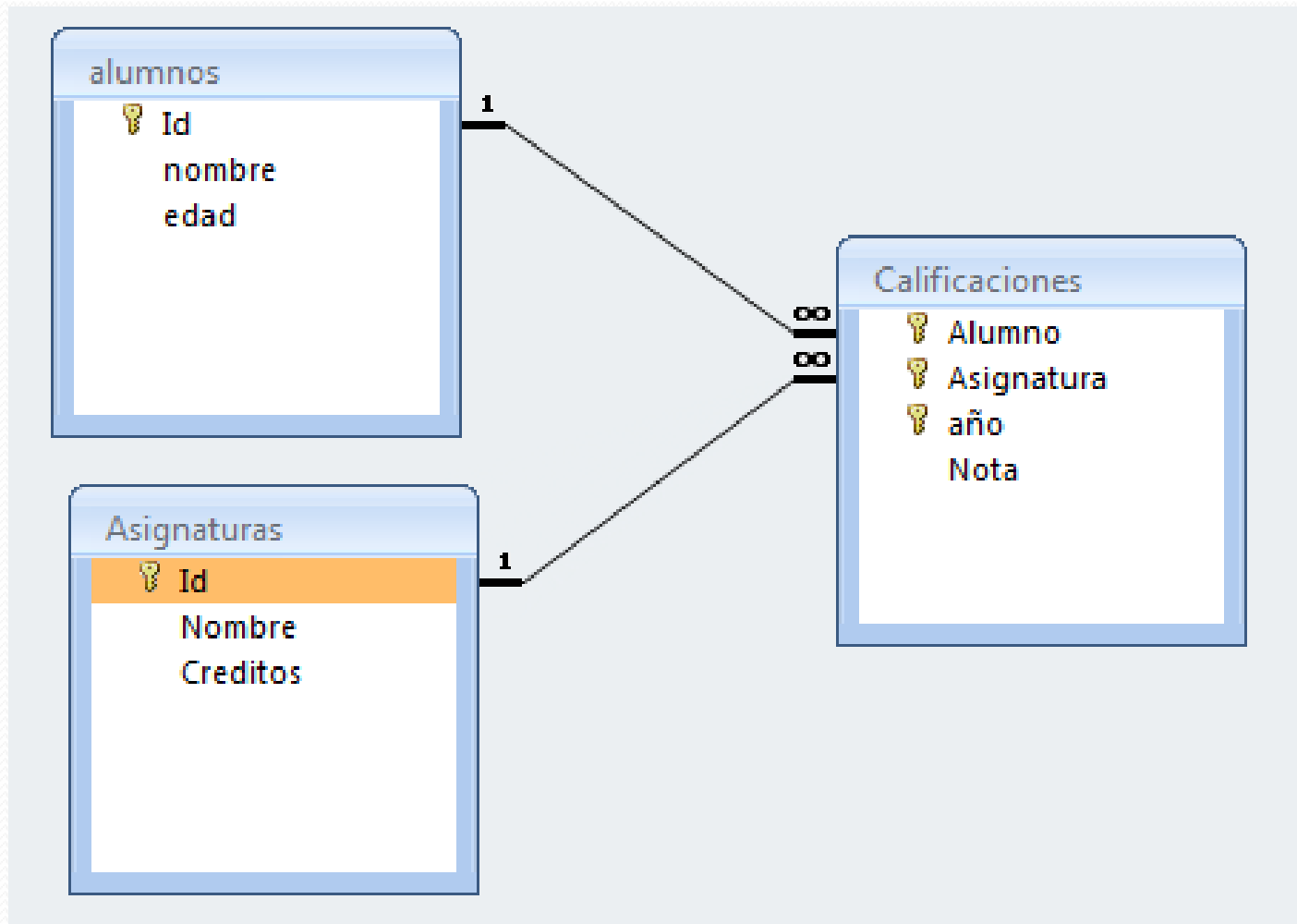
# Creando una Base de Datos

alumnos				
		Id	nombre	edad
	+	1	Juan	23
	+	2	Manolo	21
	+	3	Ana	21
*		(Nuevo)		

Asignaturas				
		Id	Nombre	Creditos
	+	1	Matemáticas	12
	+	2	Lengua	10
	+	3	Historia	15

# Creando un Base de Datos

- Vamos a asegurarnos de que se mantiene la **integridad** en los datos añadiendo las claves externas:



# Creando una Base de Datos

- Insertar los datos:

Calificaciones				
	Alumno ▼	Asignatura ▼	año ▼	Nota ▼
	1	1	2010	5
	1	2	2011	6
	2	1	2010	7
	2	2	2011	7
	2	3	2011	8
	3	1	2010	3

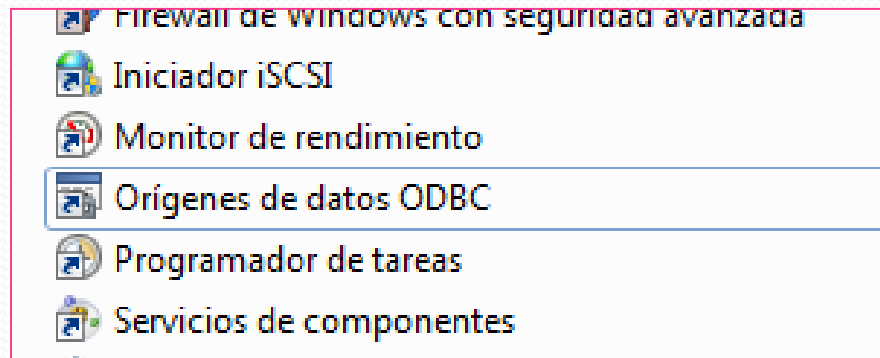
- Y guardar la Base de Datos donde queramos y con el nombre que queramos.

# Creando un conector ODBC

- Para indicar al sistema dónde debe enviar las solicitudes SQL que hagamos tenemos que crear un nuevo **Origen de Datos**.
- En este caso, nos encontramos con un pequeño problema:
  - Trabajamos con un Sistema de 64 bits
  - El software con el que se ha creado la Base de Datos es de 32 bits.
- Solución:

# Creando un conector ODBC

- En el grupo de Herramientas Administrativas buscaremos el Administrador de Orígenes de Datos ODBC



- Pulsamos con el botón derecho para cambiar la **configuración del acceso directo**



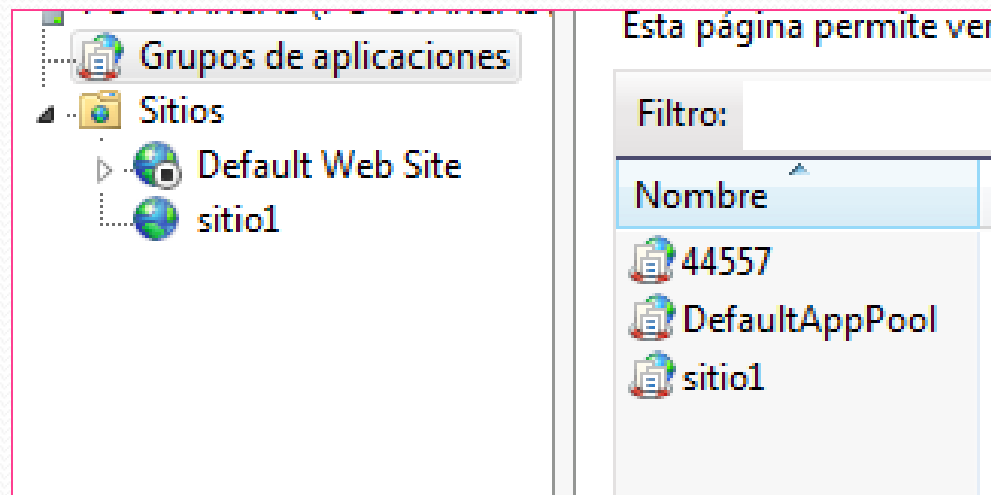
# Creando un conector ODBC

- Tanto en el destino del acceso directo como en Iniciar en, cambiaremos la carpeta **System32** por la carpeta **SysWOW64**
- En Windows 10 ya viene bien puesto

Tipo de destino:	Aplicación
Ubicación de destino:	SysWOW64
Destino:	<input type="text" value="%windir%\SysWOW64\odbcad32.exe"/>
<hr/>	
Iniciar en:	<input type="text" value="%windir%\SysWOW64"/>

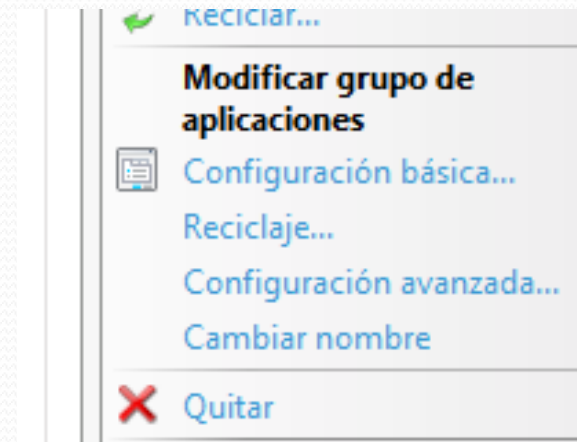
# Creando un conector ODBC

- Ahora queda decirle al sistema que permita utilizar aplicaciones de 32 bits:
- En el IIS, seleccionamos en el menú desplegable de la izquierda “Grupo de Aplicaciones”



# Creando un conector ODBC

- Seleccionamos el sitio con el que estamos trabajando y buscamos a la derecha “configuración avanzada”

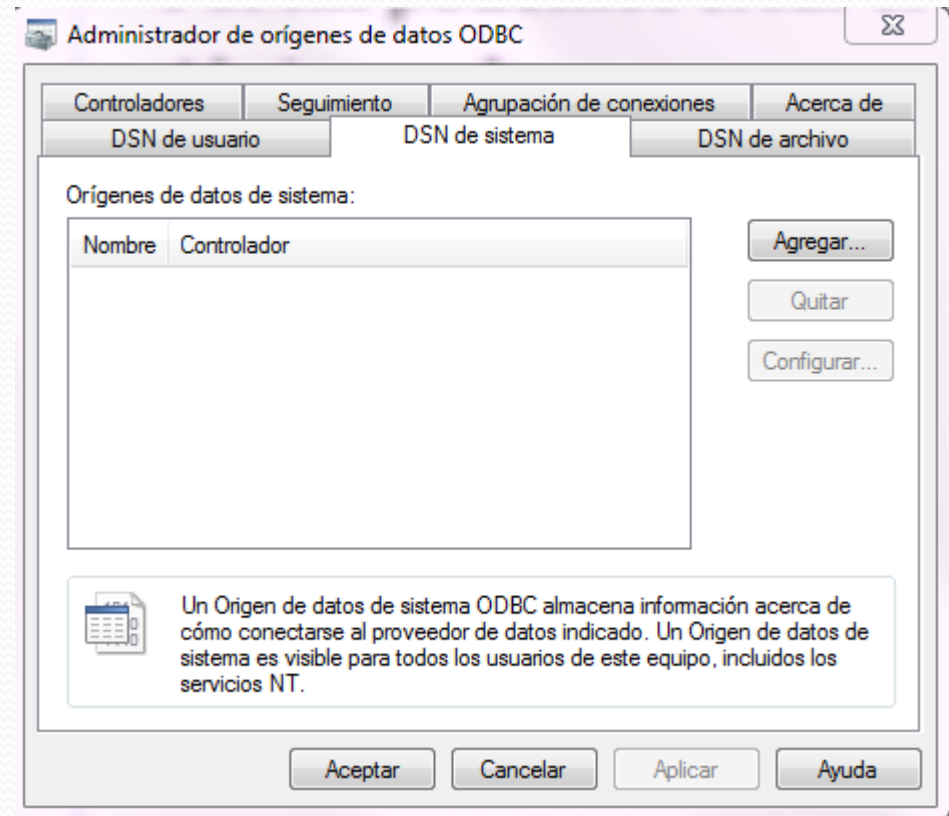


- Sólo nos queda poner “Habilitar aplicaciones de 32 bits” a **TRUE**

☐ (General)	
Habilitar aplicaciones de 32 bits	<b>True</b>
Inicio automático	True
Longitud de cola	1000

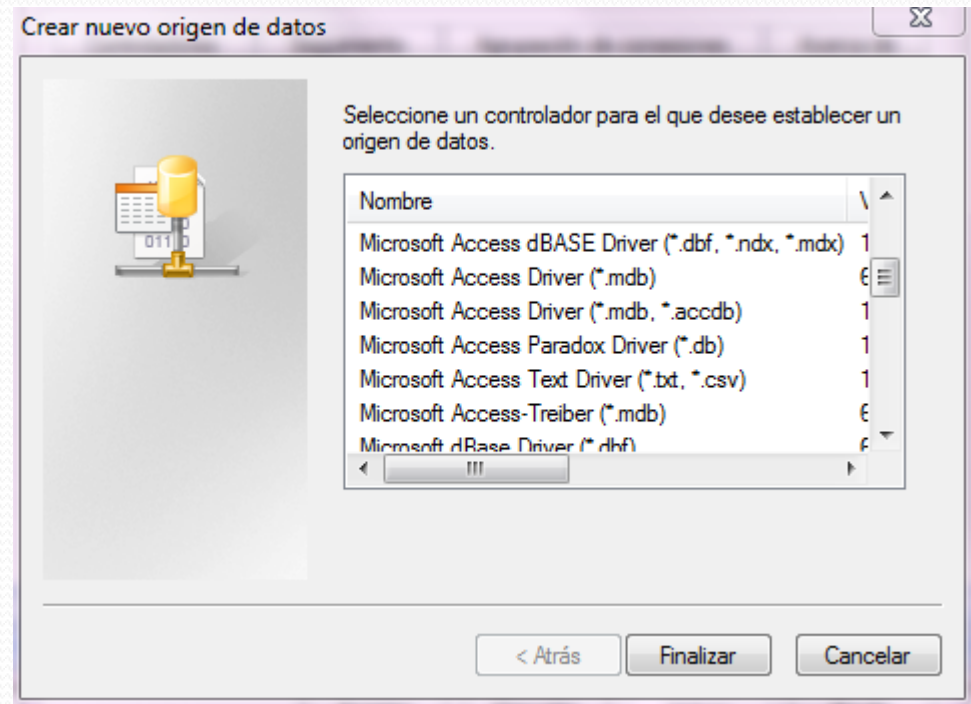
# Creando un conector ODBC

- Por fin podemos crear el conector sin problema.
- Abrimos el **Administrador de Orígenes de Datos**
- En la pestaña **DNS de Sistema** agregaremos un nuevo Origen de datos



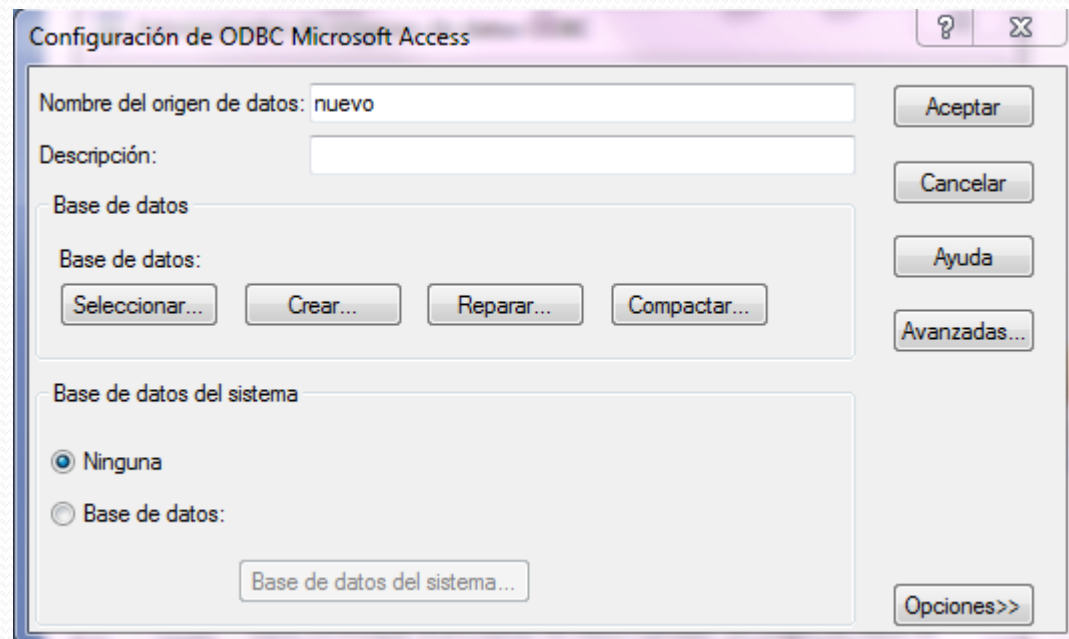
# Creando un conector ODBC

- Buscaremos de entre todas las aplicaciones permitidas la que ponga la opción **Microsoft Access Driver** con la extensión **accdb**



# Creando un conector ODBC

- Le damos un nombre al conector.
- Este **nombre** deberá ser recordado después puesto que debemos indicar que las peticiones se envíen por este conector.
- Seleccionamos la base de datos asociada al conector



# Creando un conector ODBC

- Ya está creado el conector, y ya podemos empezar a utilizar la base de datos creada.

# Acceso a la Base de Datos



# Acceso a la Base de Datos

- Como todo lo visto hasta el momento, VBS accede a la Base de Datos por medio de **Objetos**.
- El estudio en profundidad de estos objetos lo veremos más adelante
- De esta forma, el proceso será el siguiente:
- Primero crear un objeto de tipo **conexión**

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

# Acceso a la Base de Datos

- Asignar al objeto conexión, qué **conector ODBC** debe utilizar para las peticiones

```
Conn.Open ("nuevo")
```

- Crear la **consulta SQL** y ejecutarla

```
sSQL="Select * From alumnos Order By nombre"  
set RS = Conn.Execute(sSQL)
```

# Acceso a la Base de Datos

- En este momento RS se convierte en un conjunto de datos (**recordset**) que incluye una fila por cada fila que haya devuelto el SELECT.
- Para recorrerlo utilizaremos un bucle que tendrá:
  - Como condición no llegar al final del fichero:
  - Como avance, coger una nueva línea:

```
Do While Not rs.EOF
```

```
rs.MoveNext
```

# Acceso a la Base de Datos

- El código quedará como lo siguiente

```
Do While Not rs.EOF

    response.write("<tr>")
    nom = RS("nombre")
    response.write("<td>"&nom"&"</td>")
    ed = RS("edad")
    response.write("<td>"&ed"&"</td>")
    response.write("</tr>")

    rs.MoveNext
Loop
```

# Acceso a la Base de datos

- No podemos olvidarnos nunca de cerrar la conexión con la Base de Datos

```
Conn.Close
```

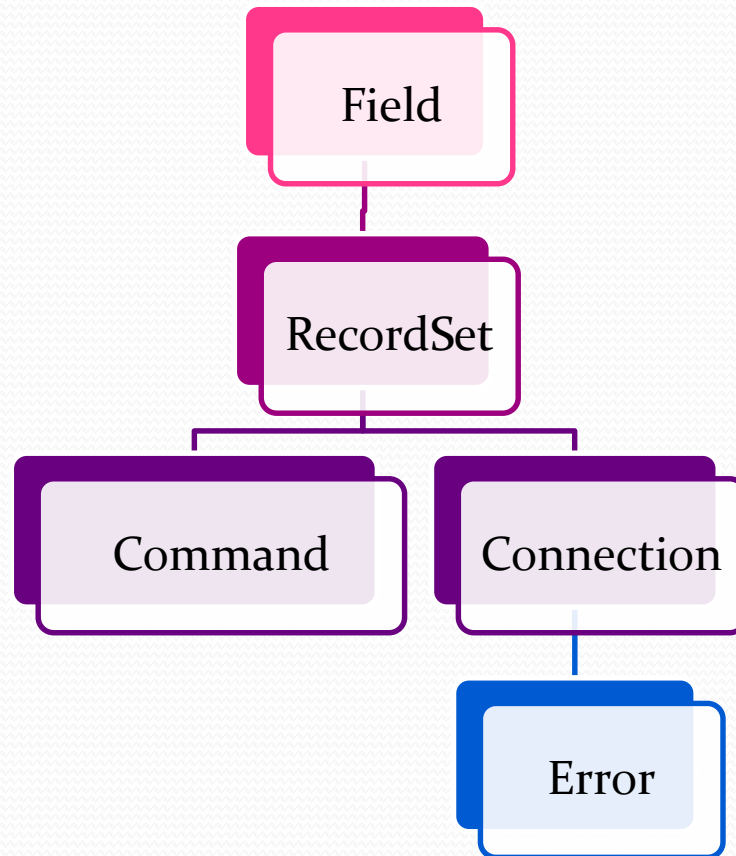
# Objetos ADO

# Objetos ADO

- Ahora que ya hemos visto el funcionamiento básico vamos a ver en profundidad cómo funciona.
- ASP utiliza objetos **ADO** (ActiveX Data Object) para manejar Bases de Datos.
- ADO, a su vez, utiliza **conectores ODBC** para acceder a las Bases de Datos
- Esto nos permite independizarnos del SGBD que utilicemos para las Bases de Datos y si mantenemos la misma estructura en las tablas, las aplicaciones ASP no tendrán que modificarse.

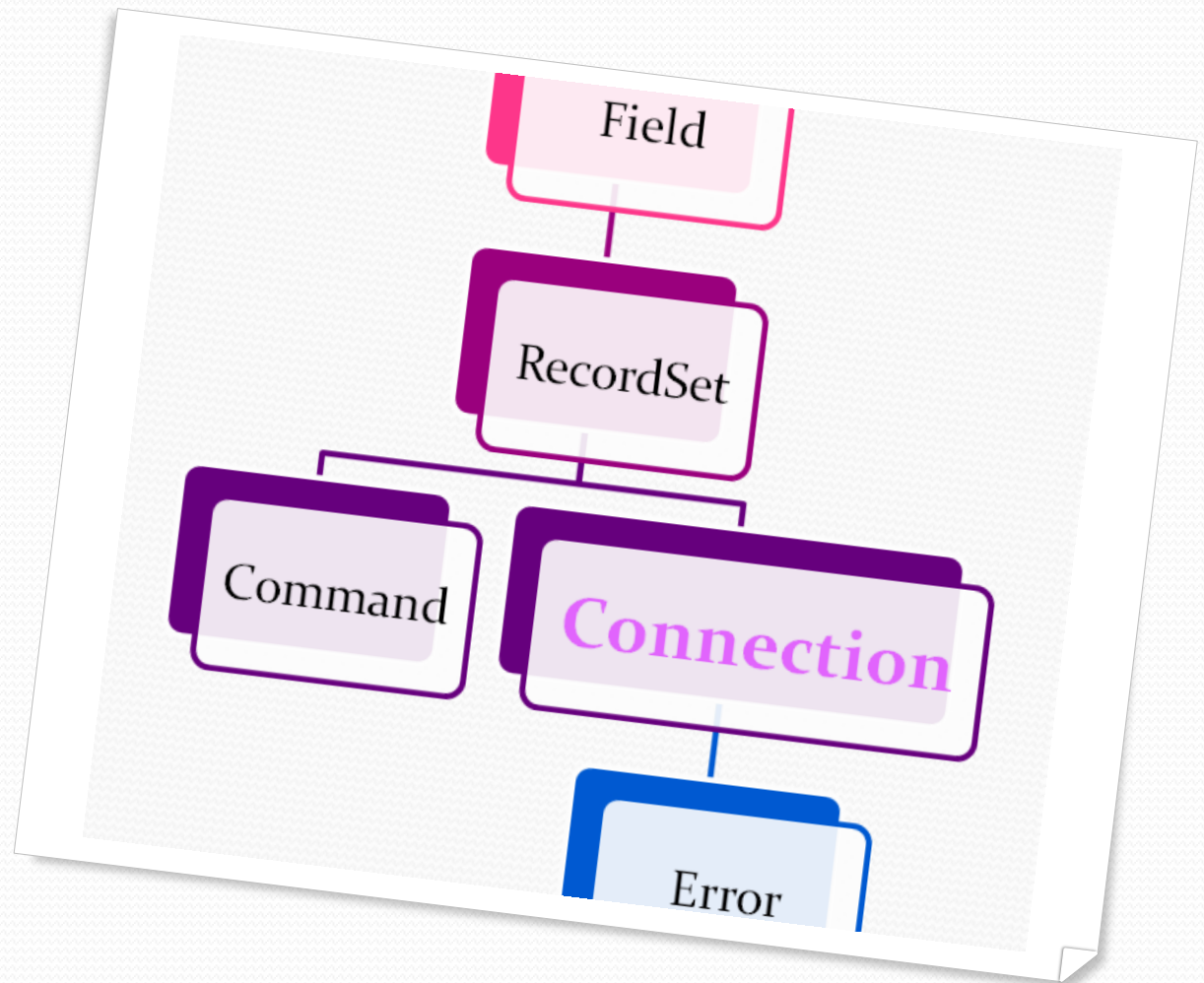
# Objetos ADO

- ADO está formado por varios objetos organizados de forma jerárquica cada uno de ellos con sus métodos y atributos específicos.





## El Objeto Connection



# El objeto Connection

- Proporciona **conexión** a la BD a través de un ODBC
- Es el objeto primario ADO, ninguno más podrá existir en nuestra página ASP si no hay una conexión.
- En caso de que no se declare de forma explícita, ADO lo declarará por sí mismo.
- La conexión se cerrará cuando nosotros lo hagamos de forma explícita

# El objeto Connection: atributos

- Sólo tiene un **atributo** => ConnectionString
  - Especifica los datos con los que queremos conectar:
    - DSN: Nombre del ODBC a utilizar => obligatorio
    - UID: Nombre de usuario => opcional
    - PWD: Password del usuario => opcional
  - ConnectionString contiene una única cadena de texto con la información necesaria para efectuar esa conexión mediante parejas de valores separadas por ";"

```
Conn.ConnectionString="DSN=nuevo;UID=user;PWD=pass"
```

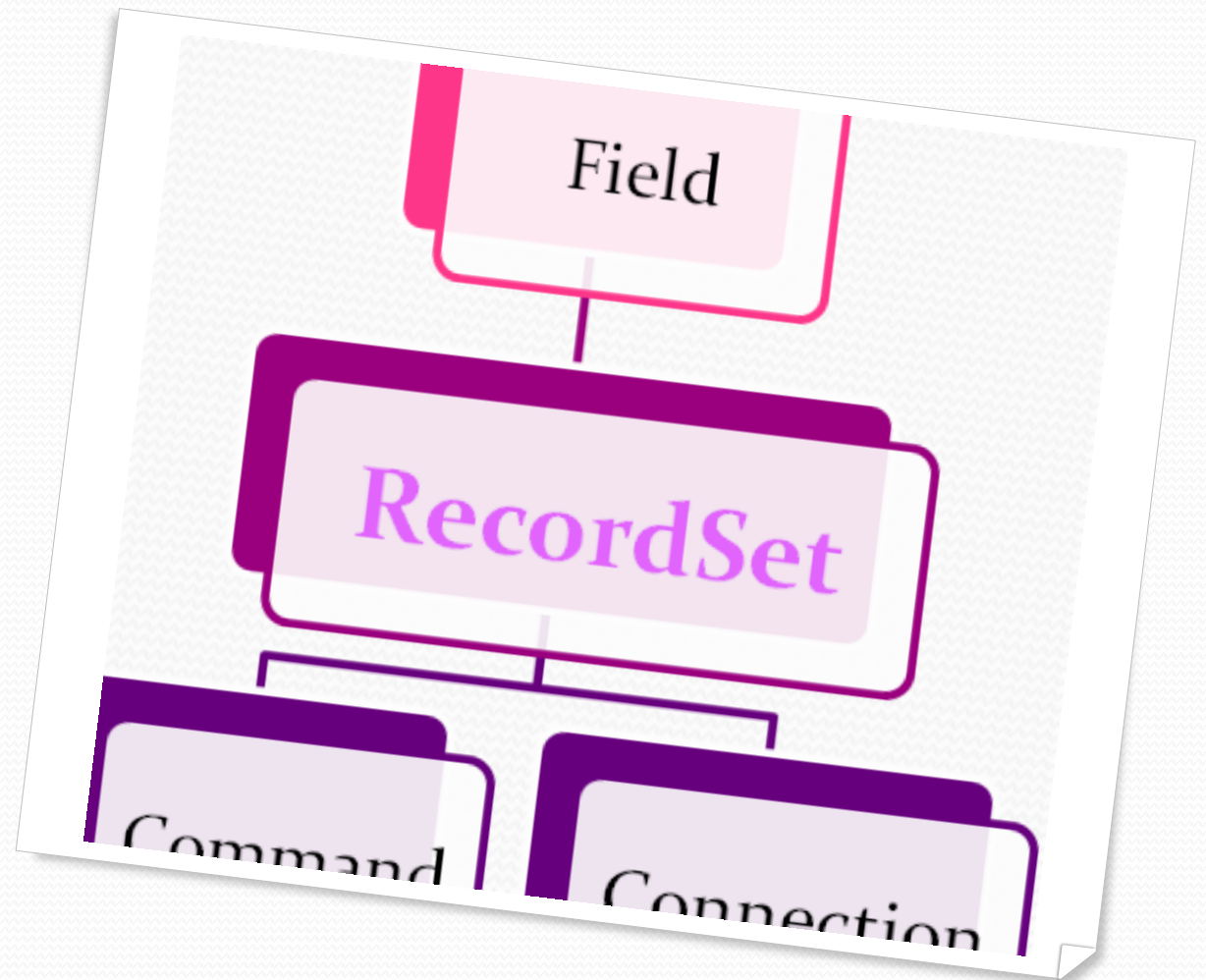
# El objeto Connection: Métodos

- BeginTrans
  - **Abre** una transacción, todas las operaciones que realicemos a partir de ese momento no serán efectivas hasta que no cerremos la transacción
- CommitTrans
  - **Cierra** una transacción y hace efectivos los cambios
- Close
  - **Cierra** el objeto

# El Objeto Connection: Métodos

- Execute
  - **Ejecuta** una sentencia SQL
  - El método Execute recibe dos parámetros: consulta SQL a ejecutar y un número.
  - En el número se almacenará la cantidad de filas afectadas por un **INSERT**, **UPDATE** o **DELETE**, o un **-1** si la sentencia era un **SELECT**
- Open
  - **Abre** una conexión con los parámetros establecidos en su `ConnectionString`
- RollBackTrans
  - **Deshace** todos los cambios efectuados desde el inicio de la transacción

## El Objeto RecordSet



# El objeto RecordSet

- Representa una tabla organizada en filas (registros) y columnas (campos).
- Los atributos y métodos de Recordset son muchos, veremos solo algunos

# El objeto RecordSet

- **Métodos**

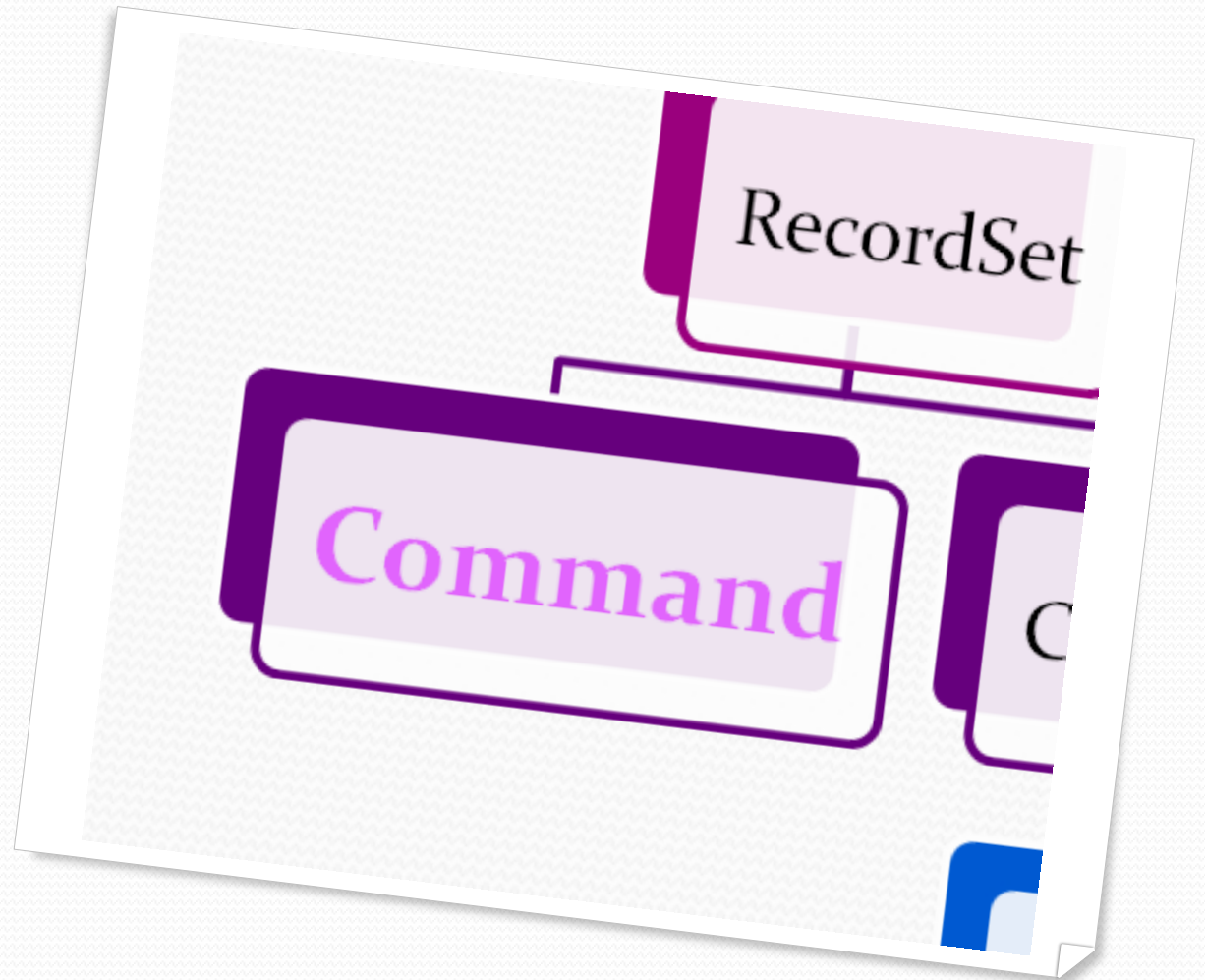
- Move(numero): Se mueve numero registros
- MoveFirst
- MoveLast
- MoveNext
- MovePrevious

- **Atributos**

- RecordCount: Contiene el número de registros del RecordSet
- EOF: True cuando estamos en el último registro del RecordSet
- BOF: True cuando estamos en el primer registro del RecordSet



## El Objeto Command



# El objeto Command

- Permite **ejecutar** un comando SQL contra una Base de Datos y almacenar su resultado (si lo hay) en un ResultSet
  - **Atributos**
    - CommandText: Cadena de texto con el comando a ejecutar
  - **Métodos**
    - Execute: Ejecuta el comando almacenado



# Notas

- Para comparar con **fechas**, ASP y ACCESS sólo admiten el operador **like**, no el = o el <>
- Los **caracteres de sustitución** para utilizar con LIKE en ACCESS son:
  - ? → Un solo carácter
  - \* → Varios caracteres
- Sin embargo, en **ASP** estos caracteres no funcionan, por lo que en nuestras consultas deberemos utilizar:
  - \_ → Un solo carácter
  - % → Varios caracteres

# Notas

- En caso de que el nombre de un campo incluya espacios, en las consultas encerraremos el campo entre corchetes:

```
sql = "Select * from personas where [nombre amigo] = 'Juan'";
```

- Para obligar al IIS a que nos envíe los datos de la base de datos con la codificación correcta (en nuestro caso UTF-8) deberemos incluir como primera línea de todas las páginas:

```
<% @ CODEPAGE = 65001 %>
```

# Acceso a bases de datos