UNIVERSIDAD ICESI

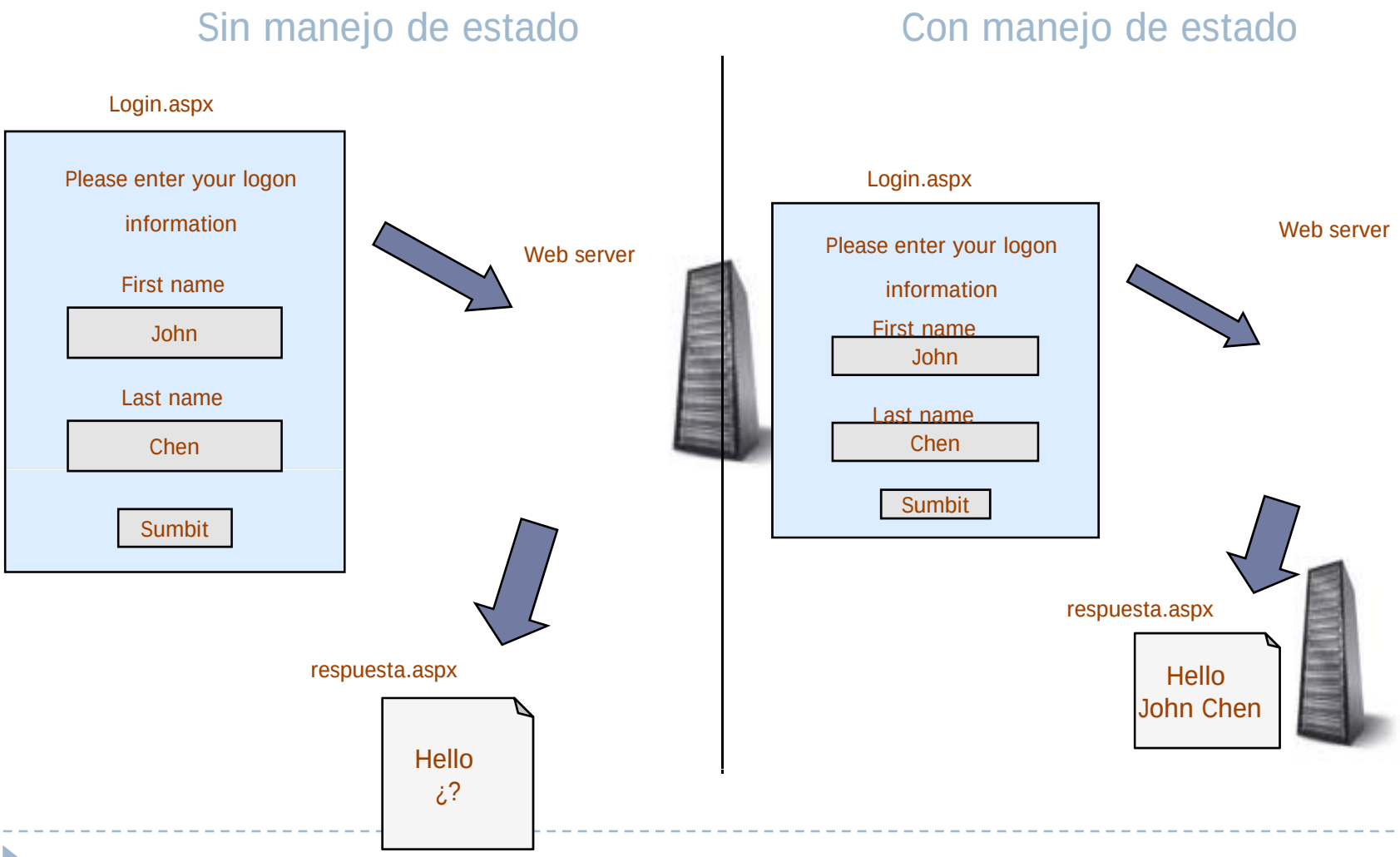
Administración del estado ASP .NET

Laboratorio de Programación
Lorena Castañeda Bueno

Introducción

- Http es un protocolo sin estados, cada solicitud se atiende a medida que se recibe. Una vez procesada la solicitud se descartan sus datos.
- Es útil mantener el estado entre solicitudes.
- Asp.Net permite mantener el estado mediante el uso de variables de aplicación y de sesión.
- Sesión → La conexión que es establecida entre un cliente y un servidor web.
- La administración del estado es el proceso que permite conservar la información ingresada por el usuario a través de diferentes request sobre la misma o diferentes páginas.

¿Qué es la administración del estado?



Tipos de administración de estado

Manejo de estado del lado del servidor	Manejo de estado del lado del cliente
Estado de la aplicación ■ Información disponible para todos los usuarios de la aplicación web.	Cookies ■ Archivos de texto que almacenan la información en el cliente.
Estado de sesión ■ Información disponible solo para un usuario en una sesión específica.	La propiedad ViewState ■ Retiene valores de múltiples request para la misma página.
Base de datos ■ Estado a través de la base de datos.	Query Strings ■ Información que se adiciona al final de la URL.

Administración de estados del lado del servidor

- Estado de la aplicación
  - Una instancia de HttpApplicationState por cada aplicación web.
  - Mecanismo de almacenamiento global disponible desde todas las páginas de la aplicación web
  - Variables de aplicación
    - Almacenar variables en la aplicación compartidas por múltiples sesiones y con poca frecuencia de cambio.
- Estado de la sesión
  - Limitado a la sesión actual.
  - Variables de sesión
    - Puede almacenar valores que tengan que mantenerse durante la sesión de un usuario en las variables de sesión.
    - Únicas de cada sesión de usuario y se puede tener acceso a ellas en cualquier página ASP.NET de una aplicación.

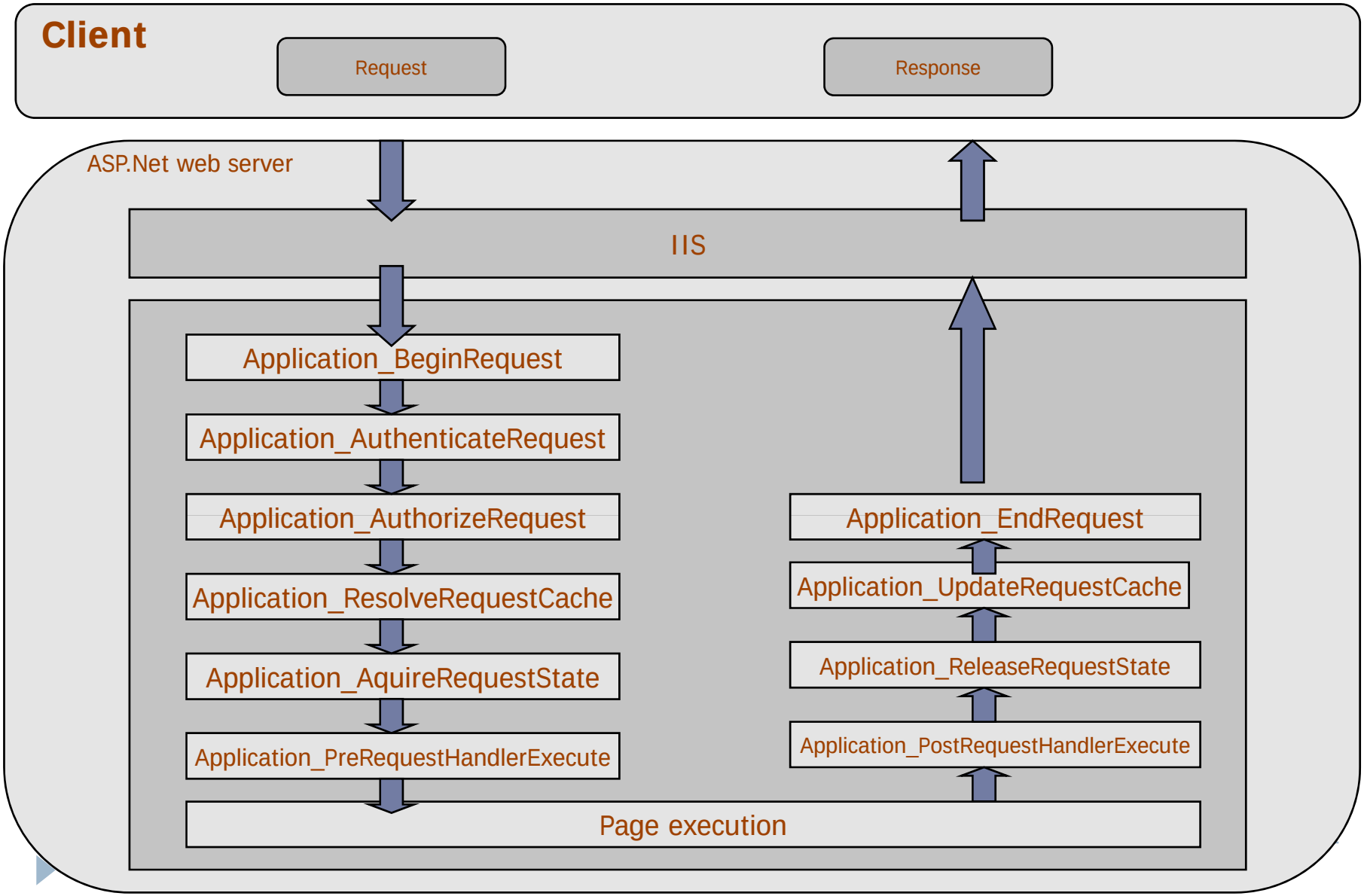
Administración de estado del lado del cliente

- Cookies
  - Pequeña cantidad de datos almacenado en un archivo texto en el cliente o en la memoria en la sesión.
  - Contiene información específica de la página que el servidor envia al cliente.
  - El servidor está autorizado para leer la cookie y extraer su información.
- Tipos de cookies
  - Temporales → de sesión o no persistentes. Existen solo en la memoria del browser. Mueren cuando se cierra el browser.
  - Persistentes → Similares a las temporales pero tienen un período de expiración. Es guardada en el disco duro.
- Límites en el tamaño de la información: no más de 4 KB.
- Poco seguras pues el usuario puede manipularlas

El archivo Global.asax

- Usado para manejar eventos mientras que la aplicación web se está ejecutando.
- Cada aplicación web de ASP.NET soporta un archivo global.asax.
- Almacenado en el directorio virtual de la aplicación.
- Puede manejar los eventos de aplicación y sesión que son usados para inicializar las variables de aplicación y de sesión.
- Es un archivo opcional, si no se define, se asume que no se manejará eventos de aplicación y de sesión.

El archivo Global.asax



El archivo Global.asax

- Soporta tres categorías de eventos
  - Los que se activan cuando se hace request de una página.
  - Los que se activan cuando el request es enviado.
  - Condicionales

Eventos que se activan cuando se hace un request

Nombre del evento	Descripción
Application_BeginRequest	Se activa cuando un nuevo request es recibido.
Application_AuthenticateRequest	Indica que el request está listo para ser autenticado.
Application_AuthorizeRequest	El request está listo para ser autorizado.
Application_ResolveRequestCache	Usado por el caché de salida para parar el procesamiento de los request que estaban en caché.

Eventos que se activan cuando el request es enviado

Nombre del evento	Descripción
Application_PostRequestHandlerExecute	Después de que la página o servicio web ha sido procesado.
Application_UpdateRequestCache	Se procesó todo el código y está listo para adicionarse al caché.
Application_EndRequest	Cuando se termina el envío del request.

Eventos condicionales

Nombre del evento	Descripción
Application_Start	Cuando se inicia la aplicación web.
Application_End	Cuando termina la aplicación.
Session_Start	Cuando comienza una sesión en la aplicación web
Session_End	Cuando teermina una sesión
Application_Error	Cuando se presenta un error no manejado.

Variables de aplicación y de sesión

- Inicialización.
- Asignación y lectura de valores.
- Modificación de la duración.
- Proceso para almacenar el estado de la sesión.

Inicializando variables de aplicación y de sesión

- Las variables son

Inicializando variables de sesión

```
protected void Session_Start(Object sender, EventArgs e)
{
    Session["ColorFondo"] = "blue";
    Session["ColorFuente"] = "gray";

}
```

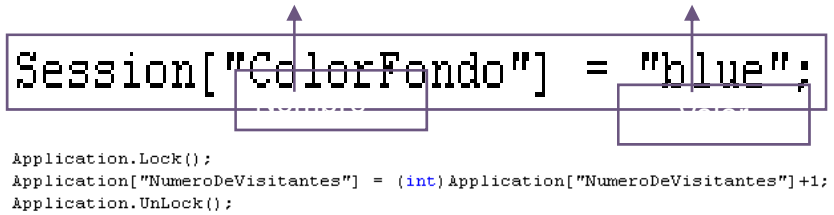
Inicializando variables de aplicación

```
protected void Application_Start(Object sender, EventArgs e)
{

    Application["NumeroDeVisitantes"] = 0;
}
```

Usando variables de aplicación y de sesión: asignando valores

- Parejas, id y valor. El id es el nombre que identificará a la variable.



- Es necesario controlar concurrencias en las variables de aplicación pues son usadas por todas las sesiones.

Usando variables de aplicación y de sesión: leyendo valores

- Simplemente leer el valor desde el objeto Session o del objeto Application.

```
StrColorFondo = (String)Session["ColorFondo"];
lblNumVisitantes.Text = Application["NumeroDeVisitantes"].ToString();
```

Cookies

- Usando cookies para almacenar datos de la sesión.
- Recuperando información de una cookie.

Usando cookies para almacenar datos de la sesión

```
//Creando una cookie llamada myCookie
HttpCookie objCookie = new HttpCookie("MyCookie");

//Asignando valores a la cookie
objCookie.Values.Add("Tiempo", DateTime.Now.ToString());
objCookie.Values.Add("ColorDeFondo", "Blue");
objCookie.Values.Add("ColorDeLetras", "Gray");

//El siguiente código pone el tiempo de expiración en una hora
objCookie.Expires = DateTime.Now.AddHours(1);

//Agregando la Cookie a la colección de Cookies del objeto Response
Response.Cookies.Add(objCookie);
```

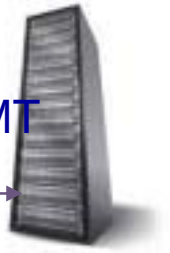
Si no se asigna tiempo de expiración la crea como temporal

¿Cómo funcionan las cookies?

Encabezado http

Set_Cookie: Username=John+Chen;

Path=/: domain=microsoft.com;
Expires=Tuesday, 01-feb-05 00.00.01 GMT



Cliente

Servidor

Siguientes requests

Recuperando información de una cookie

```
//leer la Cookie

HttpCookie objCookie = Request.Cookies["MyCookie"];

//Recuperando los valores almacenados en la Cookie

lblTime.Text = objCookie.Values["Tiempo"];
```