

Variables VIEWSTATE

El ViewState nos permite almacenar información de estado (la información se serializa en XML y luego se codifican en Base64) directamente en las páginas Web ASP.NET entre devoluciones de datos o Post Back. ElPostBack es un reenvío de los datos del formulario de vuelta al servidor, el cual puede ser provocado por la acción de un evento click de un botón o el cambio de selección de un combobox, etc. Cuando se procesa la página, se codifica el estado actual de la página y de los controles, y la cadena que se obtiene se guarda en la página en forma de campo oculto (**input type="hidden"**). Cuando la página se representa en el explorador, no se guardarán los cambios en el ViewState.

El ViewState proporciona información de estado para una página específica ASP.NET. Si se necesita utilizar esa información en más de una página o si se desea que la información permanezca para todas las visitas al sitio Web, se debe utilizar otro método para mantener el estado, como por ejemplo, estado de sesión, estado de aplicación o personalización.

TIPO DE INFORMACIÓN QUE ADMITEN LAS VARIABLES VIEWSTATE

Se puede almacenar objetos de tipo Cadenas, Enteros, Booleanos, Array, ArrayList, Tablas Hash, Convertidores de tipos personalizados y además puede almacenar otros tipos de datos en donde la clase debe compilarse con el atributo **Serializable** para que el ViewState pueda serializarlos en formato XML.

Como se ha dicho antes, la información del ViewState se serializa en XML y luego se codifica en Base64, lo cual en ocasiones nos puede generar grandes cantidades de información que puede afectar al rendimiento de la página y nos puede generar problemas con algunos proxy y firewall, por lo tanto, es recomendable probar el rendimiento de las páginas y si es necesario también deshabilitar el ViewState para los controles de datos como el GridView, DataList, Repeater, etc que pueden generar campos ocultos de gran tamaño.

Entonces, en resumen tenemos que ViewState nos brinda las siguientes ventajas:

- **No se requieren recursos del servidor:** Está incluido en una estructura dentro del código de la página.
- **Implementación sencilla:** No requiere que se utilice una programación personalizada. Se habilita de forma predeterminada para mantener los datos de estado en los controles.
- **Características de seguridad mejorada:** Los valores están fragmentados, comprimidos y codificados para implementaciones de Unicode, por lo que son más seguros que los campos ocultos.

Y las desventajas de utilizar el ViewState son:

- **Consideraciones sobre el rendimiento:** Dado que el ViewState se almacena en la propia página, el almacenamiento de valores de gran tamaño puede hacer que la página se muestre y se envíe de forma más lenta para los usuarios. Esto es especialmente relevante para los dispositivos móviles, donde el ancho de banda constituye a menudo una limitación.
- **Limitaciones de dispositivos:** Los dispositivos móviles pueden carecer de una capacidad de memoria suficiente para almacenar una gran cantidad de datos.
- **Posibles riesgos de seguridad:** El ViewState se almacena en uno o varios campos ocultos de la página. Aunque se almacenan los datos fragmentados, se pueden ver y manipular si se obtiene acceso al código fuente de la página.

CREACIÓN DE UNA VARIABLE VIEWSTATE

Las variables ViewState no se definen, se utilizan. Siempre se crean cuando se le asigna un valor y guardan la información durante toda la ejecución de nuestro sitio Web (a menos que demos el botón de atrás).

Asignación de un valor:

```
ViewState["nombreVariable"]= valor;
```

```
ViewState["ejemplo"]=txtejemplo.Text;
```

Devolver el valor de una variable ViewState a una variable:

(Debemos hacer un casting o convertir la variable a ese tipo de datos):

```
int i = (int)ViewState["ejemplo"];
```

O

```
int i = Convert.ToInt32(ViewState["ejemplo"]);
```

RECORDAD:

Hay que convertir el dato al tipo de datos que corresponde cuando leemos el valor del **ViewState** (en este caso al tipo integer, pero puede tratarse de cualquier tipo de datos)

Si queremos que no se guarde el estado de un control, no tenemos más que cambiar el valor de la propiedad EnableViewState a false.