

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Parte II

1. En C#:

- = Es necesario inicializar las variables
- = Existe un recolector de basura
- = El bloque finally se ejecuta siempre
- = Con la conversión explícita perdemos datos

2. En el modelo de capas:

- = La Interfaz llama al EN

3. Donde se guarda la aplicación WEB:

- = En un Directorio Virtual

4. RAD son las siglas de:

- = Rápido Desarrollo de Aplicaciones

5. Es obligatorio establecer la etiqueta runat = "server" en ASP.NET:

- = Si

6. Al establecer la etiqueta runat = "server":

- = El formulario se debe procesar en el servidor

7. Un proyecto WEB:

- = Sirve para aplicaciones web avanzadas

8. Las paginas maestras:

- = Proporcionan coherencia a la pagina

9. En las master pages (paginas maestras):

= Definimos el contenido común y los contenedores de contenido

10. El enlace entre el mensaje del evento y el mensaje específico se lleva a cabo mediante:

= Delegado de eventos

11. Los delegados son:

= Puntero a función

12. Un Boton es de tipo:

= Evento de envío

13. Un Textbox es de tipo:

= Ambas son Correctas (Evento de cache y no Postback)

14. Se puede transformar No Postback a Postback:

= Asignando AutoPostBack = true

15. Se puede generar estilo de web sin usar archivo CSS:

= Si, con código inline

16. Con el comando Response.Redirect():

= Podemos enviar parámetros dentro de la web

17. Para leer parámetros en la Web usamos:

= Request

18. Un menú estático es aquel que:

= El control del menú esta expandido completamente

19. Un menú dinámico es aquel que:

= Son estáticas las porciones especificadas

20. Con RequiredFieldValidator Validamos:

= La cadena no sea vacía

21. Los controladores de validación pueden ser:

= Los dos (Estático y dinámico)

22. En los controladores de validación (marcar la falsa):

= No se puede controlar un rango de datos

23. La validación de un correo electrónico es:

= \S+@\S+\.\S+

24. En el objeto sesión:

= Ambas son correctas

(Al acabar timeout se cierra sesión y al cerrar el navegador se borran los datos)

25. Para cerrar sesión usamos:

= Session.Abandon

26. Cuales están implementados como colecciones o conjuntos de pares nombre-valor:

= Ambas (Session y Application)

27. Si queremos asignar variables para todos los usuarios usaremos:

= Application

28. Las variables application pueden ser bloqueadas:

= Si

29. El archivo Global.asax:

= No contiene etiquetas ASP.NET

30. Los cambios en Global.asax:

= Requieren reinicio de aplicación

31. En el entorno conectado usaremos:

= Datareader

32. En el entorno desconectado usaremos:

= Dataset

33. Los Objetos Connection y Command:

= Tienen prefijo

34. El archivo web.config:

= Basado en XML

35. Para realizar un SELECT:

= ExecuteReader

36. Con DataAdapter:

= La gestión por abrir y cerrar la BBDD es automática

37. Con DataAdapter:

= La gestión por insertar, borrar y actualizar la BBDD es automática utilizando
CommandBuilder

38. GridView:

= Los dos (Con asistente y con código)

48. MSIL es:

El lenguaje intermedio reconocido por el CLR

49. En la arquitectura de capas propuesta:

Las entidades de negocio no acceden directamente a las bb.dd nunca

50. En el acceso conectado, recuperamos una fila de una consulta:

Con el método Read de la clase DataReader

51. Las tareas CRUD son llevadas a cabo por:

El comportamiento de acceso a datos

52. C# es un lenguaje:

Fuertemente tipado

53. En ASP.net:

Una aplicación web sólo existe en una localización que ha sido publicada por IIS como un Directorio Virtual

54. En C# el enlace entre un evento y su controlador:

A y C son ciertas

(Implementa mediante un delegado y se implementa mediante un puntero a función)

55. En el acceso desconectado de la clase que está entre nuestra aplicación y la conexión de las bb.dd se llama:

DataAdapter

56. En el acceso conectado, la cadena de conexión se pasa como parámetro:

Al constructor de la clase SqlConnection

57. En el objeto Application:

Mantengo el contador de visitas del sitio web

58. En .NET cada lenguaje de programación define sus propios tipos de datos

Cierto

59. La propiedad que indica si un control de validación es correcto se llama:

IsValid

60. Los objetos de la clase DataColumn:

Contiene información sobre una columna

61. La información guardada en la sesión:

Perdura solo durante la sesión actual del usuario

62. La concurrencia en el método desconectado de ADO.net:

Se guarda una copia del dataset original y se compara con la BBDD para ver si ha sido modificada

63. En el objeto Session:

Mantengo el nombre de usuario

64. Un directorio virtual:

Se identifica por un alias que representa la localización física en el servidor

65. Las páginas maestras:

Me permiten crear una plantilla común para todas las páginas de la aplicación web

66. El término code behind:

Hace referencia a código C#

67. En C# el tipo ArrayList:

Todos los elementos almacenados en un ArrayList son tratados como objetos

68. En asp:

Podemos crear una cookie como instancia de la clase HttpCookie

69. En asp la clase que representa un Email se llama:

MailMessage

70. El objeto Response:

Lo utiliza el servidor para enviar información al cliente

71. El atributo runat ="server":

Se declara en los controles que requieren procesamiento en el servidor

72. El objeto Session:

Ninguna de las anteriores

(No mantiene las cookies, no mantiene las preferencias del navegador y no hace que el navegador recuerde los datos del usuario)