**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
DESARROLLO DE APLICACIONES WEB –EXAMEN DE MEJORAMIENTO – TERMINO I 2011**

|  |  |  |
| --- | --- | --- |
| NOMBRE: |  | PARALELO: |

1. (5 puntos) Considere que necesita escribir la configuración web.xml para enviar los parámetros necesarios al servlet ConfigAdmin.class para la conexión a la base de datos: IP del servidor, usuario y clave, que son 200.12.45.12, admin, admin321, respectivamente. Complete la configuración del archivo web.xml para enviar dichos parámetros como parámetros de inicialización del servlet.

|  |
| --- |
| <servlet> <servlet-name>config\_servlet</servlet-name> <servlet-class> </servlet-class></servlet>  |

1. (5 puntos)Suponga que requiere escribir la configuración de un filtro para el servlet del tema 1 EN EL ARCHIVO WEB.XML, que permita controlar su utilización mediante autenticación, el nombre del filtro es verifyauth y la clase del filtro se encuentra dentro del paquete daw y se llama auth.java. El filtro debe aplicarse solo al servlet config\_servlet. Complete la configuración en el web.xml para este propósito.

|  |  |
| --- | --- |
| <filter> <filter-name> </filter-name> <filter-class> </filter-class></filter> | <filter-mapping> <filter-name> </filter-name> <url-pattern> </url-pattern></filter-mapping> |

1. (5 puntos) Cuál de los siguientes tipos de objetos pueden almacenar atributos? (Puede elegir más de uno)
2. ServletConfig
3. ServletResponse
4. RequestDispatcher
5. HttpServletRequest
6. HttpSessionContext
7. (5 puntos) Si una aplicación cliente no acepta una cookie, cual tipo de mecanismo para el manejo de sesiones pudiera utilizar (Elija solo uno):
	1. Cookies, y no reescritura del URL.
	2. Re-escritura de URL y no cookies.
	3. Se pueden usar cookies y re-escritura de URL
	4. No se puede usar ni cookies ni re-escritura de URL
8. (15 puntos) Dado los siguientes Beans

|  |
| --- |
| **public class TheBean {** **private int value;** **private TheInsideBean inside;** **public TheBean() { value = 42; inside = new TheInsideBean(value+1);}** **public int getValue() { return value; }** **public void setValue(int v) { value = v; }** **}****public class TheInsideBean {** **private int value;** **public TheInsideBean(){ this.value = -1; }** **public TheInsideBean(int value) { this.value = value; }** **public int getValue() { return value; }** **public void setValue(int v) { value = v; }** **}** |
| **Asuma que aún no se ha creado ninguna instancia de la clase**  |

1. Escriba el código JSP que permita crear una nueva instancia de éste bean y lo almacene en el request.
2. Escriba el código necesario sin usar EL que permita mostrar los valores del atributo value del bean TheBean y del bean interno inside.
3. Escriba el código necesario usando EL que permita mostrar los valores del atributo value del bean TheBean y del bean interno inside.
4. (5 puntos) Para evitar el problema de referenciar a los recursos dentro de las páginas JSP utilizando el nombre de la aplicación para formar rutas absolutas se utiliza la siguiente expresión EL:
	1. ${pageContext.eLContext}
	2. ${pageContext.servletContext.contextPath}
	3. ${pageContext.request.contextPath}
	4. ${pageContext.session.id}
5. (60 puntos) Usted es parte del equipo de desarrollo de una Aplicación Web para manejo de Compras en Línea, usted está encargado de programar el servlet para manejo de la funcionalidad de “Carrito de Compra” y una página de consulta del carrito de compra llamada “vercarrito.jsp”.
	1. **(40 puntos)Usted debe implementar UN SOLO servlet que dé soporte a los siguientes requerimientos de la interface:**

public class CarritoBean {

 private List<Carrito> carrito;

 public List<Producto> getCarrito() { … };

 public void AgregarProducto(int id) { … };

 public void RemoverProducto(int id) { … };

}

public class Carrito {

 private int id;

 private String nombre;

 private double precio;

 private int cantidad

 // METODOS GETTER Y SETTER

 // YA IMPLEMENTADOS

}

public class Producto {

 private int id;

 private String nombre;

 private double precio;

 // METODOS GETTER Y SETTER

 // YA IMPLEMENTADOS

}

1. El cliente ingresará a una página llamada “catalogo.jsp” (que ya se encuentra desarrollada) y podrá navegar con funciones AJAX por el catalogo de productos.
2. En caso de que el cliente desee comprar un producto debe hacer click en agregar producto con lo cual ser hará una llamada al servlet de la siguiente forma: servletCarrito?accion=agregar&id=IDPRODUCTO donde IDPRODUCTO es el identificador del producto que se ha seleccionado, en este caso usted debe implementar la llamada al servlet con los parámetros indicados y usted deberá generar un XML de respuesta con un mensaje OK indicando que el producto se agrego sin problemas, para **manejo de estado** se debe manejar una instancia de CarritoBean que será guardada en una variable de sesión la primera vez que se agregue un producto.
3. Debe soportar servletCarrito?accion=remover&id=IDPRODUCTO con lo cual se procederá a remover el producto del carrito.
4. Finalmente el servlet debe soportar el comando servletCarrito?accion=finalizar con lo cual se debe proceder a eliminar las variables de sesión existentes.
5. **(20 puntos)Usted debe implementar haciendo uso de JSTL y EL (no scriptlets) la página “vercarrito.jsp” donde se deberá mostrar la siguiente información dentro de una tabla, una instancia de CarritoBean está guardado en una variable de sesión llamada “carritoBean”:**
	1. Nombre
	2. Precio
	3. Cantidad
	4. Precio Total (Precio\*Cantidad)
	5. Enlace para Remover el producto (servletCarrito?accion=remover&id=IDPRODUCTO)