**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**DESARROLLO DE APLICACIONES WEB**

**SEGUNDA EVALUACIÓN - II TÉRMINO 2016**

***Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matrícula: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**TEMA 1 (20 puntos)**

Seleccione la respuesta correcta:

1. El manejo de la cache con ETag, significa que se va a utilizar:
   1. Un tag para describir el recurso
   2. Un código hash basado en la fecha de modificación del recurso
   3. Un código hash basado en los contenidos del recurso
2. El navegador utilizará el recurso que tiene en la caché cuando:
   1. No esté disponible un nuevo recurso en el servidor
   2. En el Cache-Control se lo especifique
   3. El recurso no haya expirado
3. El ataque de Falsificación de peticiones en sitios cruzados (CSRF) consiste en:
   1. El atacante provoca que el usuario realice un requerimiento al servidor sin saberlo.
   2. El atacante roba las credenciales del usuario para hacer requerimientos.
   3. El atacante logra descifrar el token de la sesión del usuario.
4. Para evitar ataques por Inyección SQL, debo:
   1. Utilizar sesiones
   2. Utilizar AJAX para el envío de datos
   3. Validar los datos ingresados
5. Si nosotros necesitamos obtener un recurso a través de un API REST, lo más apropiado sería utilizar:
   1. POST
   2. GET
   3. DELETE
6. El archivo WSDL sirve para especificar:
   1. Los métodos que posee un Web Service.
   2. Todos los web services que posee una compañía.
   3. La forma de conectarse al Web Service.
7. La organización dedicada a determinar y combatir las causas que hacen que el software sea inseguro es:
   1. OWASP
   2. W3C
   3. ACM
8. El patrón MVC sirve para:
   1. Crear aplicaciones web que utilizan AJAX
   2. Separar las responsabilidades de cada componente en una aplicación web
   3. Modelar una solución computacional que utiliza los últimos estándares del web
9. Cuando se dice que el protocolo HTTP es stateless, quiere decir que:
   1. Tiene pocos estados web
   2. Cada requerimiento HTTP es independiente
   3. Permite definir métodos como GET, POST, PUT, DELETE
10. Un servidor puede responder a un requerimiento:
    1. Solo con JSON
    2. Con JSON, XML, o texto plano
    3. Con lo que se especifique en el Content-Type

**TEMA 2 (30 PUNTOS)**

Conteste según corresponda:

1. Cual es la diferencia entre no-store y no-cache?
2. En qué consiste el ataque de Cross-Site scripting (XSS)? Y explique cómo prevenirlo.
3. Cual es la diferencia entre asignar al Cache-Control el valor de public o private?
4. Se dice que Node.js es non-blocking (No bloqueante), que significa esto? Utilice un ejemplo.
5. Explique el uso de cada uno de los Métodos HTTP: POST, PUT, DELETE
6. Explique cómo se crea una sesión, donde se almacena una sesión (cliente o servidor), y cómo identifica el servidor a un cliente?

**TEMA 3 (50 puntos)**

El Teatro Sucre ha decidido implementar una aplicación web para la compra de tickets para los distintos eventos que se realizan. Actualmente, los eventos son llevados en un registro manual por parte del personal del teatro donde para la realización de un evento es necesario conocer el nombre del evento, el tipo de evento (musical, obra, etc), quién es el organizador, una descripción, y una lista de horarios en el que se va a repetir el evento. Los horarios son basados por fecha y por hora, por ejemplo, todos los fines de semana del mes de marzo de 19:00 a 20:00. El teatro desea llevar también un registro de los organizadores, de los cuales se necesita saber el nombre de la organización, dirección, teléfono, y correos electronico. Para la compra de un ticket es necesario que un cliente provea de su numero de cedula, nombres completos, celular, correo electrónico, y deberá seleccionar un asiento (se le presenta una distribución de asientos del teatro). Los asientos están divididos por 3 secciones: Black Box, VIP, y General.

Se le solicita:

1. Definir todos los modelos necesarios para su aplicación (Entidades)
2. Diseñar un API para el módulo de eventos (crear, obtener, borrar, actualizar) donde se especificará:
   1. La acción a realizar
   2. Método HTTP a utilizar
   3. Ruta o endpoint
   4. Parámetros a enviar
   5. Respuesta JSON
3. Se le solicita implementar la reserva de un ticket por parte de un cliente. Para la implementación, considere el siguiente código en node.js y complete según los espacios:
   1. La ruta o endpoint de reserva de tickets será similar a la siguiente: “/evento/123/reservar”, donde “123” representa al id del evento.
   2. El método HTTP a utilizar.
   3. El Id del evento que está como parámetro en la URL.
   4. El Id del cliente que está en sesión.
   5. El horario recibido como parámetro.
   6. El asiento recibido como parámetro.

|  |
| --- |
| var ruta = \_\_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_\_\_;  app.\_\_(2)\_\_(ruta, function(req, res, next) {  var eventoId = \_\_\_\_\_\_\_\_(3)\_\_\_\_\_\_\_\_\_\_\_\_;  var clienteId = \_\_\_\_\_\_\_\_(4)\_\_\_\_\_\_\_\_\_\_\_\_;  var horario = \_\_\_\_\_\_(5)\_\_\_\_\_\_\_\_\_\_\_\_;  var asiento = \_\_\_\_\_\_(6)\_\_\_\_\_\_\_\_\_\_\_\_;  eventoModel.reservar(eventoId, clienteId, horario, asiento, function(resp){  if (resp) {  res.send(respuesta);  }  });  }); |