**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**DESARROLLO DE APLICACIONES WEB**

**SEGUNDA EVALUACIÓN - I TÉRMINO 2016**

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

**TEMA 1 (20 puntos)**

Seleccione la respuesta correcta:

1. 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
2. La Organización dedicada a determinar y combatir las causas que hacen que el software sea inseguro es:
   1. OWASP
   2. W3C
   3. ACM
3. 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
4. El significado de URI es:
   1. Uniform Resource Identifier
   2. Unique Resource Identifier
   3. Uniform Request Identifier
5. Cual es un ataque que consiste en insertar un código javascript malicioso en una página web a través de algún formulario?
   1. Cross Site Scripting (XSS)
   2. SQL Injection
   3. CSRF
6. En el manejo de los recursos que están en la caché, cuando el navegador posee el recurso más reciente, el servidor responderá con un status:
   1. 200
   2. 304
   3. 404
7. Si yo deseo indicarle al navegador que no guarde absolutamente nada en la caché, lo hago seteando en el Cache-Control el valor de:
   1. no-cache
   2. no-store
   3. no-all
8. Si nosotros necesitamos crear un nuevo registro o recurso a través de un API REST, lo más apropiado sería utilizar:
   1. POST
   2. GET
   3. DELETE

**TEMA 2 (30 PUNTOS)**

Conteste según corresponda:

1. En qué consiste el patrón de arquitectura MVC?
2. Qué es SOAP y para qué sirve el archivo WSDL?
3. Por qué es necesario expirar o renovar la sesión de un usuario?
4. Cómo prevenir un ataque por SQL Injection? Mencione y **explique** al menos **dos** formas
5. Para qué sirven los HTTP Headers Last-Modified y ETag en el manejo de la caché?
6. Explique el ataque por Falsificación de peticiones en sitios cruzados (CSRF)

**TEMA 3 (50 puntos)**

Un centro médico le ha solicitado una aplicación que permita la reserva de citas médicas. La aplicación debe permitir el manejo de pacientes que poseen un número de cédula, nombres, apellidos, fecha de nacimiento, y una lista síntomas, asimismo debe permitir el manejo de doctores donde cada doctor posee una cédula, nombres, apellidos, especialidad y un conjunto de horarios. Cada doctor define su horario de atención, este horario es por día de la semana y una hora de inicio y hora de fin. Finalmente, el sistema debe proveer de un módulo de citas, que básicamente permitirá establecer la cita entre un paciente y un doctor especificando la fecha y hora. Una cita puede ser cancelada y asimismo debe ser posible especificar que el paciente efectivamente llegó a la misma.

Se le solicita:

1. Definir todos los modelos necesarios para su aplicación.
2. Crear un diseño del API para el módulo de citas donde se especificará:
   1. La acción a realizar
   2. El método HTTP a utilizar
   3. La ruta o endpoint
   4. Una breve descripción del endpoint
3. Se le solicita implementar una ruta que permita obtener todas las citas médicas de un paciente de una fecha de inicio a una fecha de fin. Considere el siguiente código en node.js para implementar esta ruta donde usted deberá establecer: la ruta, el método HTTP que se va a realizar, el userId que está en la sesión del usuario, fechaIni y fechaFin son enviados como parámetros desde el cliente:

|  |
| --- |
| var ruta = \_\_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_\_\_;  router.\_\_(2)\_\_(ruta, function(req, res, next) {  var userId = \_\_\_\_\_\_\_\_(3)\_\_\_\_\_\_\_\_\_\_\_\_;  var fechaIni = \_\_\_\_\_\_(4)\_\_\_\_\_\_\_\_\_\_\_\_;  var fechaFin = \_\_\_\_\_\_(5)\_\_\_\_\_\_\_\_\_\_\_\_;  db.consultarCitas(userId, fechaIni, fechaFin, function(respuesta){  if (respuesta) {  res.send(respuesta);  }  });  }); |

|  |  |
| --- | --- |
| línea | Código |
| (1) |  |
| (2) |  |
| (3) |  |
| (4) |  |
| (5) |  |

1. A partir del código anterior, la línea “res.send(respuesta);“ envía un JSON de respuesta con el reporte de todas las citas de un paciente de una fecha a otra. De un ejemplo de un JSON de respuesta.