Nombre y Apellido: Delfina Morello

11086 - Programación en Ambiente Web - UNLU
Segundo Parcial 2020

Entrega: archivo PDF por mail a la dirección paw@unlu.edu.ar antes del viernes 26 de junio a las 12pm, es decir cuenta con 24hs para realizarlo. Además del envío por mail del PDF se solicita suban también dicho archivo PDF a un repositorio propio y envíen dicha dirección en mail aparte o por whatsapp/telegram para garantizar recepción a tiempo utilizando dos vías independientes y con timestamp.
Metodología: el examen es individual y si bien puede utilizar libros e Internet, el examen debe ser autocontenido y con respuestas «propias», no con URLs hacia material externo. Puede incluir esquemas o lo que considere necesario para ilustrar sus respuestas.
Nota Importante: En ninguno de los 10 puntos se solicita código. Responda cada punto considerando el escenario más real/auténtico que pueda imaginar y explicite cada una de sus asunciones.

1. ¿Qué diferencia existe entre una petición HTTP generada por un agente de usuario de forma asincrónica respecto a una sincrónica? ¿Cómo puede distinguir una aplicación web entre ambas?

En una petición HTTP asincrónica los datos que se le solicitan al servidor se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, por el contrario, en una petición HTTP sincrónica, la interactividad del usuario con la página se detiene/bloquea hasta recibir la respuesta por parte del servidor.

Podemos distinguir si se trata de una aplicación web sincrónica, cuando, por ejemplo, hacemos click en un botón y el sitio no nos deja seguir operando (se bloquea). En cambio, si al hacer click en dicho botón, podemos seguir utilizando el sitio, es decir, no está bloqueado, sabremos que se trata de una aplicación asincrónica.

2. Que diferencias existen entre el diseño responsivo y el universal? ¿En qué conceptos hay que hacer hincapié al momento de definir las media querys en cada caso?

Que un diseño sea responsive significa que se adapta a diferentes tipos de pantalla y dispositivos. Al momento de definir la media query, se hace hincapié en el tamaño (ancho y alto) de la pantalla. (ej de algunas propiedades min-width, max-width, max-heigth, min-heigth)

Un diseño universal consiste en que un sitio web pueda ser accedido por la mayor cantidad de personas posibles de manera satisfactoria y que hagan uso de sus contenidos, independientemente de cualquier limitación personal o derivada del entorno, es decir, en igualdad de condiciones y oportunidades con el conjunto de la sociedad. En este caso, al momento de definir las media query, hay que tener en cuenta si la persona tiene alguna discapacidad o no. En caso de poseer alguna discapacidad, saber cuál es y en base a ella implementar la media query (ej de algunos media type: braille, speech, tty, entre otras).

3. ¿Por qué decimos que no son directamente comparables REST y SOAP en el contexto de los Web Services?

REST es un estilo de arquitectura de software compuesta de pautas y buenas prácticas para el desarrollo de servicios web mantenibles y escalables.

SOAP se define como un protocolo estándar de comunicación que se utiliza para el intercambio de mensajes, basados en XML. Permite a los programas comunicarse a través de protocolos de transferencia (generalmente a través de HTTP), independientemente de la tecnología sobre la que estén implementados.

Por lo tanto, SOAP y REST no son directamente comparables ya que SOAP es un protocolo y REST es un estilo de arquitectura.

DIFERENCIAS ENTRE REST Y SOUP

-REST está orientado a recurso, la web es el universo de la información accesible.
-SOAP está orientado a servicios, a la actividad, a la comunicación.

-REST tiene muy poca sobrecarga.
-SOUP genera mayor sobrecarga.

-REST se centra en la escalabilidad y rendimiento de los sistemas hipermedia distribuidos a gran escala.
-SOUP se centra en el diseño de aplicaciones integradas (distribuidas).

-REST posee mecanismo de nomenclatura consistente para recursos.
-SOUP posee falta de nomenclatura estándar.

-REST realiza pocas operaciones en muchos recursos.
-SOUP realiza muchas operaciones en pocos recursos.

4. Explique brevemente tres principios de desarrollo seguro y de un ejemplo para cada uno.

Principios de desarrollo seguro

-Modularidad: dividir el programa en múltiples módulos y a su vez, esos módulos tienen funciones que hacen cosas específicas (funciones semi-independientes). Esto ayuda a la seguridad del sistema y es mucho más fácil de mantener. Por ejemplo, si hackean el módulo de autenticación y roban la tabla de usuarios, los demás datos siguen a salvo ya que se encuentran separados.

-Defensa en profundidad: consiste en construir múltiples capas de defensa que pueden salvar al sistema. Por ejemplo, para que un usuario pueda acceder a información sensible, debe pasar varios filtros, como por ej, su ip debe ser privada, luego autenticarse e ingresar una contraseña de seguridad.
En este principio asumimos que cada una de las capas va a ser vulnerada.

-Simplicidad: lo complejo es inseguro. Hay que tener en cuenta en qué es lo mejor que puedo hacer con los recursos que tengo. Por ejemplo, un sistema complejo, es más probable que contenga bugs y pasen desapercibidos.

5. ¿Cómo se relaciona el header HTTP Content-Security-Policy con la seguridad de un sistema web y por qué es fundamental su uso hoy en día? ¿Se puede implementar esto mismo de otra forma que no sea vía header HTTP (a nivel del server web)?

La etiqueta Content-Security-Policy nos ayuda a prevenir algunos ataques como por ej la inyección de código y cross site scripting. Nos permite restringir las cosas que cargamos en el navegador. Aunque principalmente se usa en los encabezados de respuesta HTTP, también se lo puede implementar a través de una metaetiqueta de la siguiente manera: su política irá dentro del content, atributo de la metaetiqueta. El nombre del encabezado Content-Security-Policy debe ir dentro del http-equiv, atributo de la metaetiqueta.

6. ¿Por qué es útil un buen análisis de riesgos a la hora de priorizar las mejoras de seguridad que podamos aplicar a nuestro sistema web?

Es útil hacer un análisis de riesgos para saber cuáles son los principales recursos que se encuentran vulnerables y cuáles son las amenazas que podrían explotarlos. En base a ello se podrán establecer medidas preventivas y correctivas viables que garanticen una mayor seguridad de la información.

7. Describa cómo generar una buena estrategia de SEO a partir del uso de herramientas semánticas.

Lo que debemos hacer para generar una buena estrategia de SEO es:

-Investigar palabras claves y frases de búsqueda deseables.

-Identificar las frases de búsqueda para apuntar (debe ser relevante para el negocio, obtenible y rentable).

-“Limpiar” y optimizar el código HTML de un sitio web para obtener la densidad de palabras clave adecuada, la optimización de la etiqueta del título, la estructura de enlaces internos, los encabezados y subtítulos.

-Ayudar para escribir una copia para atraer tanto a los motores de búsqueda como a los visitantes reales del sitio.

-Estudiar competidores (sitios web competidores) y motores de búsqueda.

-Implementar una campaña de construcción de enlaces de calidad.

-Añadir contenido de calidad.

-Monitoreo constante de clasificaciones para términos de búsqueda específicos.

8. ¿Cuáles son las ventajas y desventajas del modelo serverless en el cloud respecto al modelo tradicional basado en infraestructura (servers físicos / VMs).

VENTAJAS

-No hacemos mantenimiento de los servidores en donde tenemos instalados todos los programas y aplicaciones. El código se ejecuta en un contenedor temporal, ya no vamos a necesitar instalar software, gestionar puertos de acceso o estar pendiente de las actualizaciones.

-El sistema lo podemos ir escalando de manera horizontal según lo que lo vayamos necesitando.

-Solamente vamos a pagar por el tiempo que estemos utilizando el servicio.

-Las funciones se pueden integrar fácilmente con otros servicios del mismo proveedor.

-Brinda facilidad a la hora de integrar componentes.

DESVENTAJAS

-Dependencia de un proveedor si no se desarrolla con cuidado el código.

-Probablemente tendremos costos adicionales si queremos cambiar de proveedor, ya que vamos a tener que cumplir con las especificaciones propuestas por este nuevo proveedor.

9. Imagine que tiene que implementar un sistema de firma digital: dado un pdf de entrada debe devolverlo firmado digitalmente. Para ello, y dado que debe integrarse a sistemas web existentes, debe diseñar una arquitectura que facilite dicha integración. Comente sobre los componentes de la misma y qué cuestiones contempla, dificultades, etc.

La aplicación digital funcionaría del siguiente modo:

Al momento de firmar, la aplicación calcula el hash del pdf. Luego utiliza la clave privada para cifrar ese hash (se solicita la contraseña con la que el usuario protegió su clave privada).
Por último, el hash cifrado se incorpora, junto con otros datos (fecha y hora de firma, datos del firmante, entre otros), como anexo del documento, obteniendo así un documento firmado digitalmente.

Pueden existir problemas como, por ejemplo: Firma Inválida (una o más firmas del pdf pueden generar este problema).

Alguno de los motivos de este error puede ser porque:
-El documento fue modificado luego de la firma.
-Ocurrió un error durante en proceso de firmado.

10. Suponga que está desarrollando una API que puede ser consumida utilizando diferentes formatos de intercambio de datos ¿De qué forma puede determinar el backend el formato a utilizar para atender un cliente determinado? ¿Cómo debería comportarse el mismo en caso de no conocer el formato solicitado?

En el header de la petición HTTP que le enviamos al servidor le especificamos en qué formato queremos recibir el recurso, indicándole varios en orden de preferencia, para lo cual utilizamos el método Accept.
El backend nos devolverá el recurso en el primer formato disponible. Si el servidor reconoce algunos de los formatos solicitados por el cliente nos devolverá el header Content-Type para que el usuario sepa en qué formato se devuelve dicho recurso, caso contrario, nos retornará el código HTTP 406 (Not Acceptable).