PAW Segundo parcial

Legajo 123105

Imagine una aplicación web "portal de noticias" y responda las siguientes consignas:

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 el caso de las peticiónes sincronas, se asume que los recursos requieren para ser procesados los resultados previos, por este motivo se pausa la carga/procesamiento del sitio hasta que los recursos esten disponibles. Para forzar que un recurso se descargue de forma asincronica, es decir que no espere a los datos previos para operar, se utiliza el atributo "async", el cual fuerza a que el recurso sea solicitado y procesado en segundo plano. Si hay varios recursos con este atributo, cada uno sera procesado en un hilo independiente y no se garantiza que el orden de ejecucion/procesamiento sea el mismo que el de solicitud, ya que un recurso solicitado al final podria estar disponible antes que el que se solicito primero, para solucionar este problema de sincronizacion existe el atributo "defer" el cual, combinado con "async" permiten lograr ventajas de perfomance, ya que los recursos se cargan en paralelo pero se ejecutan en secuencial.

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?

El diseño responsivo hace referencia a una forma de pensar y organizar el contenido de manera que este se adapte a los diferentes cambios de resoluciones, es un concepto ligado a la compatibilidad con la mayor cantidad de dispositivos posibles, el diseño universal en cambio esta relacionado con lograr la compatibilidad con la mayor cantidad de usuarios posibles, idependientemente de sus capacidades fisicas o cognitivas. En ambos casos el objetivo de fondo es la inclusion, uno se enfoca en la inclusion tecnica de la mayor cantidad de navegadores y resoluciones, el otro, de la inclusion humana de la mayor cantidad de usuarios. El debate de fondo detras de todo esto es la vieja pregunta filosofica: El emisor es responsable de la interpretacion que haga el receptor? Hay que ser inclusivo con lo exclusivo? Polemico. Discutible. Claro esta que las posturas son mixtas y que en la web ambos extremos se encuentran en un punto intermedio, por un lado no te pido que funcione en visores obsoletos pero si

te exijo que cumplas con estandares. La zona gris son los navegadores modernos usados masivamente que no siguen estandares (lease Safari), es un punto de conflicto que demuestra que pese a los enormes esfuerzos que se han hecho, estamos muy lejos de llegar a un concenso. Las madiaquerys permiten en este sentido, cambiar la presentacion del contenido en funcion de determinadas configuraciones, como ser el ancho/alto de los displays, pero tambien en funcion de la manera en que el usuario consume esa informacion, no es lo mismo si lo visualiza de forma interactiva en una pantalla, que si lo hace mediante una impresora por ser un persona anciana por ejemplo, en el primer caso se trata de una regla de diseño responsivo mientras que en el segundo tambien podria considerarse universal. Al momento de definir las mediaquerys deben considerarse los breakpoints, o puntos de ruptura, que son aquellos limites en los cuales, el contenido deja de ser accesible, por ejemplo, si al momento de imprimir el contenido deja de ser accesible se requiere una media query en ese aspecto, lo mismo al superar determinado umbral de resolucion o cualquier otra variable que haga que el contenido deje de ser accesible.

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

Es como querer comparar JQuery con Javascript, SOAP es una herramienta para gestionar un numero finito de recursos realizando multiples operaciones, validaciones, formatos estandarizados y un fuerte tipado, permite el autodescubrimiento de servicios y el montaje de una infraestructura distribuida de forma semiautomatica si se cuenta con las herramientas adecuadas. REST en cambio, no es mas que un nombre marketinero para las paticiones HTTP que no tienen como objetivo recuperar recursos para ser renderizados, sino que tienen como objetivo realizar algun tipo de transaccion, el envio de un formulario de contacto es un ejemplo super basico de una API REST. Comparar una peticion HTTP bien formada con un protocolo complejo como SOAP es como se mencionaba al inicio, comparar JQuery con Javascript, SOAP de fondo bien podria utilizar una API REST para funcionar en algunas implementaciones. Si bien algunas caracteristicas puntuales podrian ser comparadas, no es logico compararlas en su totalidad.

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

- Autenticación y autorización El desarrollo de un modulo de seguridad que valide que quien realiza cada accion sea un usuario que cuenta con suficientes privilegios para hacerlo, esto puede hacerse mediante usuario y clave, criptografia, entre otras cosas.
- Validacion de datos Validar todos los datos provenientes del usuario y desconfiar de cada uno de ellos, esto previene ataques XSS y SQL Injection entre otros.
- Gestionar eventos inesperados Todos los errores deben ser registrados y procesados de forma segura, ejemplo cuando un request da 404/500 se debe

registrar toda la info posible ya que quiza se trate de un atacante buscando vulnerabilidades.

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)?

El Content-Security-Policy permite especificar de que fuentes se pueden ejecutar scripts, es decir, una lista blanca de protocolos y dominios aceptados, esto previene ataques de XSS ya que aunque el atacante logre inyectar condigo, el mismo sera inmunizado por esta politica. A nivel de servidor externo se puede lograr verificando el origen de la peticion, si la misma proviene de un dominio no autorizado se puede denegar el recurso, esto por supuesto siendo el server del que proviene el codigo malicioso un server responsable, si el server esta bajo control del atacante no va a ser posible. A nivel servidor propio, lo que se puede hacer es parsear la salida antes de enviarla al cliente en busca de enlaces maliciosos, esto agrega una complejidad adicional pero que puede ser util.

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?

Los recursos son finitos, la mano de obra escasa, las vulnerabilidades numerosas y el tiempo tirano. Ni se pueden atajar todos los penales que van al arco ni los que patean la tiran siempre afuera. Un buen analisis de riesgo permite destinar los recursos disponibles a prevenir las vuelnerabilidades mas urgentes, haciendo un trade-off entre riesgo/perdidas asociadas, de esta manera, los recursos se destinan de forma eficiente a arreglar y prevenir las fallas que mas inpacto tengan dentro de la organizacion, dependiendo el caso, tener el sitio caido por un ataque ddos no es nada comparado con un robo de informacion confidencial, si se manejan datos de tarjetas bancarias por ejemplo de las suscripciones de los lectores de un portal de noticias, es preferible dejarlos sin informacion por unas horas a que un atacante los deje sin dinero en sus cuentas, en cambio si se trata de un diario financiero como Investing.com el estar caido un par de horas durante una rueda agitada de wall street puede tener consecuencias catastroficas para la entidad que brinda informacion en tiempo real actualizada varias veces por segundo y no tanto si le roban la base de datos que tiene informacion publica, es decir, que con un analisis de riesgo adecuado se puede

destinar los recursos disponibles a las areas que mayor impacto tengan en el sitio en cuestion, que no siempre son las mismas incluso dentro del mismo rubro.

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

El SEO puede ser optimizado mediante el uso de microdatos y de la descripcion del contenido en un formato amigable para los crawlers. En general, los crawlers no son tan amigos del HTML como pareciera, ya que es un gran trabajo normalizar la informacion recuperada, mediante las herramientas semanticas, le ofrecemos a los buscadores nuestra informacion servida en bandeja, normalizada y organizada para que puedan hacer con ella lo que consideren oportuno. Esto es util de cara al SEO, pero tambien es una herramienta de doble filo, ya que si bien los buscadores podran accederla de una forma mas conveniente, tambien podrian hacerlos crawlers maliciosos y explotar estas herramientas en nuestro perjuicio. Por ejemplo, un negocio X podria tener un crawler que recorre el sitio del negocio Y y obtiene de los datos semanticos los precios de los productos en los que compiten, luego actualizar los precios propios a un valor mas conveniente y asi quitarle clientes. Existen servicios SaaS que ofrecen esta funcion actualmente y por lo tanto, el servidor considerando el user agent de google/bing/etc podria mostrar/ocultar los datos semanticos para prevenir esta vulnerabilidad comercial obteniendo un doble beneficio, por un lado proteje los datos de la competencia y por otro ahorra ancho de banda al enviar datos innecesarios al los visitantes que nunca los utilizarian.

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

Permiten una mayor escalabilidad y flexibilidad, facilitando la administracion, ademas se pueden subdividir un server con muchas funciones en varios mas chicos que tambien son mas faciles de mantener. Como contras esta la dependencia del proveedor y los costos. Un sistema pensado para correr en infra tradicional puede correr en serverless, uno pensado para ser usado en serverless no es tan facil migrarlo a una infra tradicional. En general las herramientas serverless se integran con otros servicios del ecosistema cloud por lo que terminan derivando en esquemas de vendor lock-in donde las empresas pueden cambiar de programadores pero no de proveedor de cloud.

9. Imagine 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.

Implementaria una API REST donde mediante un metodo post se envie en el body el PDF sin firmar y en los headers los datos de autenticacion necesarios para realizar la firma en cuestion, la respuesta se realizaria de la misma manera. Para permitir que el firmado de haga en el cliente, habilitaria politicas CORS para que pudiera ser consumido directamente desde el navegador. Las claves privadas para la firma las guardaria encriptadas utilizando como clave de encriptacion una llave mixta, formada por una parte con una clave secreta del proveedor y con una clave secreta del cliente, luego realizaria un hash de la concateniacion de ambos fragmentos mas un salt y ese resultado seria la clave de encriptacion para almacenar, de esta forma, aunque roben la info del servidor, no podran descifrarla ya que no se cuenta con toda la informacion. Tambien podria utilizarse un servidor de claves interno separado, dependiendo de los recursos con los que se cuente.

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?

El backend podria determinarlo en base a un parametro en la querystring o mediante un header o mediante una preferencia de la session del cliente, preferentemente con un parametro. En caso de no conocer el formato, podria realizar algunos test de formato mediante expresiones regulares, sin embargo preferiria que si no se incluye el parametro adecuado se devuelva un error indicando que el mismo es requerido.