Sequeira Leonardo Ariel
Legajo: 112776

11086 - Programación en Ambiente Web - UNLu

Primer Parcial 2020

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

1. ¿Por qué las sesiones pueden guardar mucha más información que las cookies? ¿Qué almacenaría para esta app en cookies y/o sesiones?

Las sesiones pueden guardar mucha más información que las cookies debido que se alojan del lado del servidor, lo cual está limitado por el espacio disponible del mismo (servidor web), en cambio las cookies son archivos muy chicos, ya que los mismos se envían en cada request.

Con respecto a esta app:

- En *cookies*, guardaría solo el ID de la sesión generada para el usuario logueado.
- Como las *sesiones* permiten "recordar" ciertos elementos cuando un usuario cambia de pagina dentro de un mismo sitio web, guardaría la configuración personalizada del estilo de Look & Feel (si lo tuviera) y cualquier otra información que requiera recordar.

2. ¿Qué ventajas ofrece el uso de Virtualhost en el contexto de servidores Web (en gral y en particular para esta app)?

Ventajas:

- Costo, ya que se puede alojar múltiples sitios web en un solo servidor, sin la necesidad de comprar grandes equipos.

3. Defina con sus palabras la diferencia principal entre contenido estático y dinámico.

El contenido *estático*, es aquel sitio que el servidor web devuelve al cliente con contenido más que nada informativo y fijo. El contenido suele ser solo HTML, CSS y JavaScript.

Mientras que el contenido *dinámico* es aquel que no es fijo, donde el HTML devuelto es formado en el servidor web antes de retornarlo al cliente. Estas páginas se forman con algún lenguaje de programación, como por ejemplo PHP, el cual genera un *client-side code* (código que corre del lado del cliente) y lo retorna al cliente.

4. ¿Cómo aplicaría el modelo MVC para el diseño de esta app? No necesita escribir código alguno, sino argumentar conceptualmente como separaría la lógica de la app en estos tres elementos.

En el modelo incluimos el acceso a los datos y la lógica del negocio. Como, por ejemplo, las consultas a la base de datos al momento de obtener información del usuario a loguearse, el tipo de servicio que el usuario tiene contratado y cuantas noticias tiene permitido visitar al día/mes.

En la vista se incluyen las representaciones visuales de los datos obtenidos del modelo, todo lo que tenga que ver con la interface gráfica y como se lo mostramos al usuario.

Y en el controlador vamos a incluir las peticiones que va realizando el usuario. Además, será el encargado de solicitarle los datos al modelo y comunicarla a través de la vista.

5. a) ¿Por qué es posible afirmar que PDO mejora la seguridad en la capa de base de datos de una app PHP?

Con PDO se puede mejorar la seguridad ya que se evita el *SQL inyenction*, básicamente lo que permite hacer es, validar los valores que provienen del lado del usuario y que luego irán en las queries a la base de datos.

b) ¿Qué otras cuestiones debemos tener en cuenta en la capa de base de datos en el sentido de la seguridad?

Cuestiones a tener en cuenta en la capa de base de datos con respecto a la seguridad:

- Gestionar bien los permisos, roles y usuarios.
- Nunca dar todos los permisos a un usuario de desarrollo, ni mucho menos dar la password del usuario root.
- En cuanto a la password de los usuarios deben ser lo más larga y compleja posible, ya que el acceso a la base de datos se debiera realizar por una única clase y no se debería reescribir la misma varias veces, ni en ningún otro lugar.

6. La app muestra signos de "envejecimiento" en cuanto al diseño, tanto usuarios finales como redactores del portal lo informan a diario. ¿Qué ideas se le ocurren al respecto?

Para una app con “síntomas” de envejecimiento, desde mi punto de vista, recomiendo un cambio en la interfaz del usuario, es decir, re diseñar el front end de la misma para mejorar la experiencia del usuario.

7. Se le informa al equipo de desarrollo que las nuevas funcionalidades están repercutiendo negativamente en la performance de esta app web en el ambiente productivo, no así en el ambiente de testing (QA). DevOps informa que existe últimamente mucha carga a nivel de bases de datos. ¿Qué se le ocurre hacer en su rol de Desarrollador Web?

8. Imagine ahora que el "portal de noticias" debe considerar tener un "paywall" (ciertos contenidos se vuelven pagos) y por ende almacenará tarjetas de débito / crédito de los clientes. a) ¿Cuáles son las implicancias de seguridad de esta nueva funcionalidad?

En principio, si es que no se hacía, pero el protocolo por el cual se debería implementar esta nueva funcionalidad debe ser HTTPS, debido a que la información a enviar de las tarjetas de crédito/débito junto con sus respectivos códigos de seguridad se considera información sensible.

Además, esa información sensible está obligada a viajar de manera encriptada.

b) ¿Cómo implementaría algún límite sobre la cantidad de noticias que puede ver un usuario que no paga, e.g. puede ver sólo 10 artículos por mes calendario?

En primer lugar, generar una nueva tabla (PLANES) en base de datos donde indicamos los planes de pagos vigentes y la cantidad de contenido mensual que debe tener acceso por plan, por ejemplo:

PLANES			
ID_PLAN	NOMBRE_PLAN	PRECIO	CANTIDAD_MES_HASTA
10001	PLAN_MIN	00.00	10
10002	PLAN_MEDIUM	100.00	20
10003	PLAN_MAX	200.00	50
10004	PLAN_PREMIUM	500.00	500

Inicialmente todos los usuarios contarán con un PLAN_MIN por defecto, el cual solo tendrán acceso a 10 artículos mensuales, el cuál se irá modificando a medida que el usuario lo requiera:

USUARIOS			
ID_USER	NOMBRE	APELLIDO	OTROS_DATOS
50001	XXX	XXX	…
50002	XXX	XXX	…
50003	XXX	XXX	…

A la tabla ARTICULOS, se indicará si un determinado articulo es pago o no:

ARTICULOS		
ID_ARTICULO	NOMBRE_ARTICULO	PAGO
90001	XXX	SI
90002	XXX	SI
90004	XXX	NO

En otra tabla de MOVIMIENTOS, se llevará el registro de todos los artículos pedidos por cada usuario:

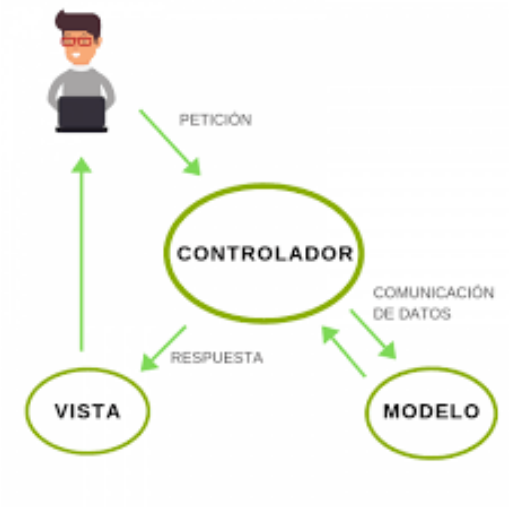
MOVIMIENTOS			
ID_USER	ID_ARTICULO	OTROS_DATOS	FECHA
15001	10001	…	12-04-2020
15002	10002	…	15-04-2020
15003	10004	…	13-04-2020

Al momento de que un usuario desee acceder a un artículo, desde backend se validará la cantidad de artículos pagos que el usuario ya tuvo acceso durante el mes calendario en curso. En caso de ser menor o igual a la cantidad permitida por plan contratado, se le permitirá el acceso al mismo, caso contrario se le notificará que no tiene permitido acceder al articulo y se lo invita a actualizar su plan a uno superior.

9. Se requiere implementar un buscador de noticias dentro de esta app. Explique qué responsabilidades tiene cada capa de la aplicación en la resolución de la búsqueda. ¿Qué método HTTP le parece el más adecuado para implementar esto? ¿Qué problemas observa?

Mediante una *VISTA* de búsqueda de noticias, la cual consiste en un input de texto y un botón de “Buscar”, el usuario ingresa el contenido a buscar y presiona el botón.

El *CONTROLADOR* recibe la petición de búsqueda del usuario y mediante el *MODELO* con PDO realiza la query a la base de datos, sane tizándolos los valores de entrada, para evitar SQL inyenction, cuando el *MODELO* obtiene los resultados de la búsqueda, le pasa al *CONTROLADOR* la respuesta, quien le informará al usuario mediante otra *VISTA*.



El método HTTP a utilizar es un GET, ya que solo tengo que enviar un solo parámetro en la query. El problema que se observa es que el parámetro viaja por la URL.

10. Se requiere que la experiencia del sitio sea uniforme en versiones de Chrome/Firefox/IE de hasta 3 años atrás. ¿Cómo puede cumplir con dicho requisito? ¿Qué estrategias adoptaría desde el punto de vista del diseño e implementación?

En principio adoptaría una nueva estrategia de diseño, pensando y teniendo en cuenta la tecnología a utilizar y la compatibilidad entre los diferentes browsers (Chrome, Firefox e Internet Explorer).

Algo muy importante también a tener en cuenta con esto ultimo es, que muchos colores se renderizan de forma diferente en los distintos navegadores, de los cuales hay una serie de colores estándares que se podrían llegar a utilizar en caso de que los colores actuales se comporten de manera diferente con el diseño actual.

En cuanto a la implementación hay ciertos factores que debemos tener en cuenta a la hora de implementar como, por ejemplo, si todos los browsers aceptan:

- Ejecución de ventanas modo pop up.
- Eventos en el movimiento del cursor.
- Que tecnologías son compatibles con cada una (CSS, JavaScript, etc).
- Formatos de imágenes soportadas.

- Que protocolos soportan cada una (SSL, etc).

Es decir, todas aquellas cuestiones que hacen diferentes a los distintos navegadores entre sí.

En cuanto a la etapa de testing, diseñaría una estrategia en donde el equipo se enfoque tanto en las pruebas funcionales, como en las pruebas de compatibilidad entre los diferentes browsers.

El objetivo es que el usuario final no note diferencias al utilizar la app con ningún browser en particular.