11086 - Programación en Ambiente Web – UNLU

Primer Parcial 2020

Nombre y Apellido: Sofia Vazquez **Legajo:** 138224

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

Las cookies solo pueden almacenar una cantidad limitada de datos y cada una solo es un par nombre-valor dado que la almacena el navegador web en el ordenador del Usuario, en cambio las sesiones son almacenadas en el servidor por ende pueden almacenar grandes cantidades de datos fácilmente y de manera segura (El usuario o cliente no puede ver o editar)

En esta app de "Portal de noticias" almacenaría en las Cookies el usuario y la contraseña introducidas por el cliente u/o usuario, para que cuando vuelva a nuestro sitio web, se autentique automáticamente sin solicitar otra vez usuario y contraseña, ya que por ejemplo en un caso como el Portal de Clarín al leer una noticia me solicita el ingreso a mi cuenta al leer noticias (Aunque sea registrarme de manera gratuita). También almacenaría el idioma que eligió el usuario la última vez que accedió a la página (suponiendo que dicho portal brinde la opción de modificar el idioma) para mostrarle directamente la página en ese idioma y no solicitarle que elija el idioma cada vez que ingresa, lo mismo con la zona horaria por ejemplo para mostrarle datos como "Fecha y Hora". A su vez podría almacenar la última noticia que leyó para mostrarle noticias similares o relacionadas con la misma. En las Sesiones almacenaría datos respecto a si el usuario se autenticó o no, el ID del usuario que se autenticó; o por ejemplo si existiera la posibilidad en dicho portal de ir marcando Noticias de interés para leer luego que me las guarde.

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

Las ventajas que ofrece son:

- Que se puede disponer de N sitios a la vez en un mismo Web Server.
- Que se puede navegar por los sitios usando sus nombres de dominio y no localhost.
- Que no posee limitación de software siempre y cuando el Server pueda manejar la carga.
- Que puedo obtener backups por si pierdo servicio de un equipo.

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

El contenido estático es el que esta permanente en la página web (por ejemplo, para definir las pestañas o secciones en la barra de un menú) que considero que es necesario para el propósito de la página ya que son temas que no se van a modificar con frecuencia y considero que tienen que estar en un lugar de manera de que al navegar en la página se pueda encontrar fácilmente. En cambio, el contenido dinámico es el que se va modificando, como por ejemplo en una app como el portal de noticias, el contenido dinámico son las noticias, el cuerpo contiene los títulos de los sucesos más relevantes del día (Generalmente con poco contenido) y el enlace a otra página donde la va a mostrar con más detalle.

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.

Modelo	**Voy a colocar todo respecto a la conexión con la BD**
Vista	**Las distintas pantallas que le muestro al usuario**
Controlador	**La lógica del negocio del sistema, el depurado de la información. Una vez que el usuario se logea se realizarán las validaciones requeridas y la autenticación**

En nuestra aplicación lo aplicaría de la siguiente manera:

El Usuario accederá con su Usuario y Contraseña a través de un INPUT de datos a través de la Vista, esa información va a acceder al Controlador, va a ser validada, se van a determinar qué rol posee obviamente haciendo las consultas a la BD a través de las librerías que utilice (En este caso PDO). Luego se harán las validaciones correspondientes y en caso de que todo este OK va a volver al Frontend mostrándole al usuario otra Pantalla (Vista) distinta con ya su sesión completada.

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

Porque PDO me brinda objetos para la conexión a la base de datos de manera Abstracta y Uniforme, proporciona una capa de abstracción de acceso a los datos, lo que significa que, sea cual sea la BD que se esté usando, se emplean las mismas funciones para hacer consultas y obtener datos. PDO provee con respecto a seguridad evitar código malicioso que ponga en “Peligro” la BD (es decir cualquier tipo de combinación de caracteres que puedan hacer alguna inyección SQL). Uno de los ataques más comunes en PHP es por ejemplo: La inyección SQL, que con una consulta de búsqueda puede comprometer todo el sistema web donde se quieran alterar los datos a través de consultas SQL; esos datos pueden dañar la BD. Para esto PDO mejora la seguridad de una app PHP.

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

Otras cuestiones que tendría en cuenta son:

- Las aplicaciones nunca deberían conectarse a la BD como su propietario o como un SU, porque puedo ejecutar cualquier consulta como borrar el contenido de una tabla por completo. Para que otros usuarios puedan utilizarla, se les deben dar privilegios (solo los necesarios), creando distintos usuarios con permisos limitados a los objetos de la BD. En el caso de que un atacante tenga acceso solamente va a poder alterar lo que la aplicación le permita.
- Con respecto a la conexión con la BD se puede establecer sobre SSL para cifrar la comunicación cliente/servidor y aumentar la seguridad. De esa manera va a ser difícil para un atacante monitorear el tráfico para obtener información de la BD.
- Una vez que un atacante tiene acceso directo a una BD (evitando un web server), la información debe estar protegida por la BD misma, cifrando los datos.

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?

Las ideas que me surgen al respecto son que su estilo de página quedó obsoleto, los componentes pueden estar ubicados en lugares que ya no se acostumbran a verse, por ende provocaría que no sea fácil para el usuario navegar en dicha página. A mi entender habría que modificar el diseño de la misma de manera que se pueda adaptar a páginas web de Noticias (Tomando en cuenta la aplicación brindada para el Parcial) que se encuentran actualizadas, intentar imitarlas en formatos para brindarle al usuario la posibilidad de pensar que conoce donde se ubican los componentes a pesar de que se encuentre navegando en un sitio diferente. Otro punto que se me ocurre al respecto es que si mi página web posee un tiempo de respuesta de más de 5 segundos el usuario va a optar por navegar en otro portal de noticias mejor A su vez podemos llevarlo a pensar que al ser una página en desuso u obsoleta, puede no ser apta para su uso en diferentes dispositivos como Tablets, celulares de distintas resoluciones o formatos. Se podría llevar la página a un estilo tanto de colores como de tipografía adecuada teniendo en cuenta lo utilizado actualmente. Por último lo adaptaría a los motores de búsqueda, que se respeten los diseños y formatos para que los Buscadores puedan reconocerlo más rápido y posicionarlo mejor en cuánto a búsqueda de la información.

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?

Me surge la idea de mantener información ya sea en el servidor o en el cliente sin ir a sobrecargar la BD realizándole consultas. Podría mantener en una cookie la información que se repite frecuentemente a lo largo de la navegación del usuario en el sitio web. Por ejemplo: Los datos del usuario no los voy a estar consultando la BD cada vez que el usuario navega de una a otra página. También podría realizar una reducción de la carga de usuarios dentro de mi sistema, que para llegar a cierta información o tener privilegios de acceso sobre cierta información y que no haya sobrecarga se pueden restringir los accesos a usuarios que solo estén registrados en mi página. (Este concepto lo asocio con lo visto en otras materias previamente)

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?

Al tener datos como en este caso de Tarjetas de débito o crédito, la clasifico como información importante a la que debo brindar seguridad de que ningún intruso/atacante pueda tener acceso a la misma. Una manera de brindar seguridad es utilizando distintas técnicas de cifrado de los datos (Transformando datos legibles en no legibles) a través de distintos algoritmos o funciones como Hash, puedo aplicar Hash sobre dichos datos, aplico salt y los guardo de esa manera en la BD; mismo las contraseñas no deben estar guardadas ni tampoco enviadas como texto plano. A su vez pienso en cuando yo realizo una compra

por un sitio web (Ya sea de suscripciones o de compra de productos) a veces se genera un doble control, ósea que una vez que registro los datos de mi tarjeta y usuario me envía por ejemplo a mi dispositivo un código de validación, un elemento que recibo para verificar mi identidad en cierto punto. No mostrar los datos de la tarjeta al cliente cuando tiene varias y puede elegir una de ellas, la edición de los datos personales van a ser mucho más críticos ya que no cualquiera puede acceder. (Autenticación)

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?

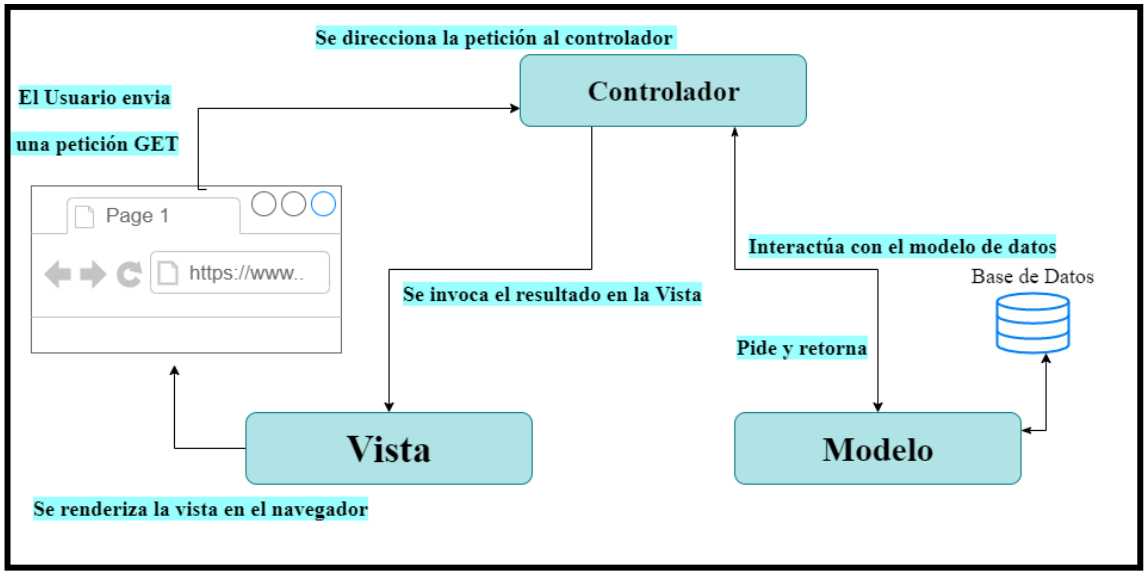
Lo implementaría con una cookie que vaya indicando la cantidad de artículos por mes de ese usuario (Como una especie de counter) de hasta 10 artículos como en el ejemplo que expira automáticamente cuando se cumplen los 30 días del mes. Al expirar dicha cookie la vuelvo a crear inicializando el Counter.

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?

Si hablamos de Front end y Back end, el front se encargaría de enviar la petición al back end en base a la información ingresada en el buscador. Por otro lado el back end se encargaría de hacer toda la depuración de lo que ingresó el usuario para evitar que a través de un query me puedan robar información o romperme la BD como en el caso de Inyección. Uno se encarga de realizar la petición y el otro del control, la depuración de lo que se ingresó para que no se produzca un robo de información hacia el exterior.

Si hablamos del modelo MVC, cuando el Usuario ingresa algo a buscar la Vista (la que presenta la información al Usuario y da mecanismos de interacción con el mismo) le envía los datos al Controlador (Dispatch entre el Modelo y la Vista), este envía consultas SQL al Modelo (Que accede a la capa de almacenamiento de datos).

El método HTTP más adecuado para implementar esto es **GET** porque es una petición de consulta, estoy buscando una noticia en este caso. Los problemas que observo es la inyección de dependencias a través del buscador de noticias, que puede surgir si no hay una buena implementación en cuanto el Back end para el manejo de la información que se ingresa antes de asesorar la consulta.



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?

Para cumplir con dicho requisito se podría usar Cross Browser, el diseño de las páginas webs que se comportan exactamente igual sin importar que navegador las muestra. Los navegadores deberían basarse en las reglas que organismos como el W3C brindan para una única interpretación del lenguaje HTML, CSS o JavaScript pero se pueden ver problemas en la interpretación del código y etiquetas. Por ejemplo, si tengo una hoja de estilos CSS compuesta de ciertos atributos con ciertos valores, puedo ver diferencias al ejecutar el mismo documento en otro navegador. Como esas reglas no son respetadas por todos se pueden optar por distintas estrategias:

- **Hojas de Estilos Específicas**
- **Validación:** la idea es poder seguir los estándares recomendados por el W3C, para posteriormente ser validado cada uno de ellos a través de los validadores oficiales.
- **Reset CSS:** son hojas de estilos CSS que se definen al principio de un documento HTML con el objetivo de minimizar las diferencias, la idea es establecer predeterminadamente las propiedades a un valor como 0, para que se tomen como referencia el navegador que se utilice.

Una solución mucho más eficiente sería neutralizar todos los estilos que afectan a la manera de renderizarse en el navegador.