Valeria Zúñiga Mendoza
A01705435

Preguntas laboratorios

Laboratorio 1: Introducción a las aplicaciones web, HTML5 y ciclo de vida de los sistemas de información

1. **¿Cuál es la diferencia entre Internet y la World Wide Web?**
   Internet es la infraestructura global de redes de computadoras interconectadas, mientras que la World Wide Web es un sistema de información que utiliza Internet para acceder a páginas web mediante navegadores.

2. **¿Cuáles son las partes de un URL?**
   Un URL consta de protocolo (https), dominio, subdominio, ruta y parámetros opcionales.

ejem:
https://docs.google.com/document/d/1babpFbpNgwah4h2Pl1yRGWhjTy9JY0b2H19x80_FvL8/edit

https: protocolo de cifrado
dominio: dirección única: (docs.google.com)
subdominio: (document) carpeta o nombre que podemos redirigir
nombre de archivo o ruta
(1babpFbpNgwah4h2Pl1yRGWhjTy9JY0b2H19x80_FvL8/edit)

agregar ?query
https://docs.google.com/document/d/1babpFbpNgwah4h2Pl1yRGWhjTy9JY0b2H19x80_FvL8/edit?variable=valor1&valor2

3. **¿Cuál es el propósito de los métodos HTTP: GET, HEAD, POST, PUT, PATCH, DELETE?**
   - GET para solicitar datos (no tiene cuerpo)
   - HEAD para obtener solo los encabezados
   - POST para enviar datos
   - PUT para actualizar recursos
   - PATCH para modificar parcialmente recursos
   - DELETE para eliminar recursos(enviar algo al servidor).

4. **¿Qué método HTTP se debe utilizar al enviar un formulario HTML, por ejemplo cuando ingresas tu usuario y contraseña en algún sitio? ¿Por qué?**
   Se debe utilizar el método POST porque envía los datos de forma segura al servidor sin mostrarlos en la URL, lo que es importante para información sensible como contraseñas.

5. **¿Qué método HTTP se utiliza cuando a través de un navegador web se accede a una página a través de un URL?**
   El método GET se utiliza para solicitar la página web correspondiente al URL

6. **Un servidor web devuelve una respuesta HTTP con código 200. ¿Qué significa esto? ¿Ocurrió algún error?**
   El código 200 indica que la solicitud fue exitosa, no hubo errores.

7. **¿Es responsabilidad del desarrollador corregir un sitio web si un usuario reporta que intentó acceder al sitio y se encontró con un error 404? ¿Por qué?**
   Sí, es responsabilidad del desarrollador corregir el error 404, que indica que la página solicitada no se encuentra, para mejorar la experiencia del usuario y mantener la integridad del sitio.
   *403 prohibido: usuario no tiene acceso. Mejor devuelve un 404 por seguridad

8. **¿Es responsabilidad del desarrollador corregir un sitio web si un usuario reporta que intentó acceder al sitio y se encontró con un error 500? ¿Por qué?**
   Sí, es responsabilidad del desarrollador corregir el error 500, que indica un error interno del servidor, para asegurar que el sitio funcione correctamente y evitar pérdida de datos o funcionalidades.

9. **¿Qué significa que un atributo HTML5 esté depreciado o desaprobado (deprecated)? Menciona algunos elementos de HTML 4 que en HTML5 estén desaprobados.**
   Un atributo depreciado en HTML5 significa que aún es compatible pero se considera obsoleto y puede ser eliminado en futuras versiones. Algunos elementos de HTML 4 que están desaprobados en HTML5 incluyen font, center, strike, entre otros. *Aún sirve

10. **¿Cuáles son las diferencias principales entre HTML 4 y HTML5?**
    HTML5 introduce nuevas etiquetas semánticas, soporte para audio y video, capacidades de almacenamiento local, mejor manejo de formularios y más, en comparación con HTML4.

11. **¿Qué componentes de estructura y estilo tiene una tabla?**
    Los componentes de estructura de una tabla incluyen filas (tr), columnas (td), encabezados de fila (th) y el cuerpo de la tabla (tbody). Para el estilo, se pueden aplicar estilos CSS para modificar el aspecto visual de la tabla, como el color de fondo, bordes y texto.

12. **¿Cuáles son los principales controles de una forma HTML5?**
    Algunos de los principales controles de una forma HTML5 son input, textarea, select, button, entre otros.

13. **¿Qué tanto soporte HTML5 tiene el navegador que utilizas? Puedes utilizar la siguiente página para descubrirlo: http://html5test.com/ (Al responder la pregunta recuerda poner el navegador que utilizas)**
    Puntuación HTML5 de mi navegador: 581/594 puntos.
    Utilicé Chrome 121 en Windows 10.

14. **Sobre el ciclo de vida y desarrollo de los sistemas de información:**
    a. **¿Cuál es el ciclo de vida de los sistemas de información?**
       El ciclo de vida de los sistemas de información incluye la identificación de necesidades, diseño, desarrollo, implementación, operación y mantenimiento del sistema.
    b. **¿Cuál es el ciclo de desarrollo de sistemas de información?**
       El ciclo de desarrollo de sistemas de información sigue un proceso iterativo que incluye análisis, diseño, implementación y pruebas, con la posibilidad de

retroalimentación y ajustes en cada etapa.

Laboratorio 2: Control de versiones

Configuración de sistema de control de versiones con git.

Laboratorio 3: CSS

1. **Como ingeniero de software ¿cuál es tu recomendación sobre el uso de !important en un CSS?**
   El uso de !important en CSS debe evitarse en la medida de lo posible, ya que puede complicar el mantenimiento del código y dificultar la resolución de conflictos entre estilos. Es preferible utilizar especificidad adecuada y estructurar el CSS de manera que se minimicen los casos en los que sea necesario recurrir a !important.

2. **Si se pone una imagen de fondo en una página HTML, ¿por qué debe escogerse con cuidado?**
   La elección de una imagen de fondo en una página HTML debe hacerse con cuidado porque puede afectar la legibilidad del contenido, el rendimiento de la página y la accesibilidad para los usuarios. Es importante seleccionar una imagen que complemente el contenido de la página, sea lo suficientemente ligera para cargar rápidamente y no distraiga al usuario.

3. **Como ingeniero de software, ¿cuál es tu recomendación al elegir las unidades de un propiedad de estilo entre %, px y pt?**
   La elección de las unidades de estilo depende del contexto y del objetivo específico. En general, se recomienda utilizar unidades relativas como porcentaje (%) para dimensiones que deben ser flexibles y adaptarse al tamaño del contenedor, píxeles (px) para dimensiones fijas y puntos (pt) para tamaños de fuente cuando se requiere consistencia en dispositivos de pantalla.

4. **¿Por qué el uso de una versión minimizada del CSS mejora el rendimiento del sitio?**
   El uso de una versión minimizada del CSS mejora el rendimiento del sitio al reducir el tamaño del archivo CSS, lo que resulta en tiempos de carga más rápidos. Al eliminar espacios en blanco, comentarios y otros caracteres innecesarios, la versión minimizada reduce el tamaño del archivo CSS, lo que significa que se transfiere menos datos entre el servidor y el cliente, reduciendo así el tiempo de carga de la página.

Laboratorio 4: Fundamentos de JavaScript

1. **¿Qué diferencias y semejanzas hay entre Java y JavaScript?**
   Ambos lenguajes utilizan una sintaxis similar a C y comparten algunas estructuras de control de flujo y conceptos de programación como bucles, condicionales, funciones, etc. Sin embargo, Java es un lenguaje de programación de propósito general, mientras que JavaScript es un lenguaje de secuencias de comandos principalmente utilizado para desarrollo web.

2. **¿Qué métodos tiene el objeto Date? (Menciona al menos 5*)**
   - **a.** getDate(): Devuelve el día del mes (de 1 a 31).
   - **b.** getMonth(): Devuelve el mes (de 0 a 11).
   - **c.** getFullYear(): Devuelve el año con cuatro dígitos.
   - **d.** getHours(): Devuelve la hora (0-23).
   - **e.** getMinutes(): Devuelve los minutos (0-59).

3. **¿Qué métodos tienen los arreglos? (Menciona al menos 5*)**
   - **a.** push(): Agrega uno o más elementos al final de un arreglo y devuelve la nueva longitud del arreglo.
   - **b.** pop(): Elimina el último elemento de un arreglo y lo devuelve.
   - **c.** splice(): Cambia el contenido de un arreglo eliminando elementos existentes y/o agregando nuevos elementos.
   - **d.** forEach(): Ejecuta una función dada una vez por cada elemento del arreglo.
   - **e.** indexOf(): Devuelve el primer índice en el que se encuentra un elemento dado en el arreglo, o -1 si no está presente.

4. **¿Cómo se declara una variable con alcance local dentro de una función?**
   ```
   function miFuncion() {
       let variableLocal = 10; // declara una variable de manera moderna
       const otraVariableLocal = "Hola"; // declarar constante
       // Resto del código...
   }
   ```
5. **¿Qué implicaciones tiene utilizar variables globales dentro de funciones?**
   - **a.** Las variables globales pueden ser accedidas y modificadas desde cualquier parte del código, lo que puede dificultar el seguimiento de los cambios y la depuración.
   - **b.** Las variables globales pueden ser sobrescritas por accidente dentro de funciones, lo que puede causar comportamientos inesperados y difíciles de diagnosticar.

c. Puede haber colisiones de nombres si se utilizan variables globales con nombres comunes en diferentes partes del código, lo que puede provocar errores difíciles de identificar.

Lab 5: Frameworks de estilo

Describe Material design:
Material Design es un sistema de diseño desarrollado por Google utilizado para crear interfaces de usuario coherentes, atractivas y funcionales en aplicaciones y sitios web.

Algunos aspectos clave de Material Design son:

1. **Principios de diseño claros y simples**: Materialidad, luz, movimiento y profundidad se combinan para crear una experiencia de usuario coherente y atractiva.

2. **Componentes y patrones de diseño:** Con una amplia gama de componentes y patrones de diseño predefinidos que los desarrolladores pueden utilizar para construir interfaces de usuario consistentes y funcionales. Esto incluye botones, tarjetas, barras de herramientas, menús, iconos y más.

3. **Materialidad y profundidad:** Utiliza sombras, bordes y capas, lo que ayuda a los usuarios a comprender la jerarquía visual y la interacción en la interfaz de usuario.

4. **Animaciones y transiciones suaves:** Material Design incluye un conjunto de pautas para animaciones y transiciones suaves que ayudan a mejorar la experiencia del usuario y a comunicar el estado y la interacción de manera efectiva. Esto incluye animaciones de entrada y salida, así como transiciones entre diferentes estados de la aplicación.

5. **Adaptabilidad y consistencia multiplataforma:** Material Design está diseñado para ser adaptable y consistente en una variedad de plataformas y dispositivos, incluyendo dispositivos móviles, tabletas, computadoras de escritorio y más. Esto ayuda a garantizar una experiencia de usuario coherente y familiar en todas las plataformas.

Lab 6: Programación orientada a eventos

1. **¿Por qué es una buena práctica usar JavaScript para checar que sean válidos los inputs de las formas antes de enviar los datos al servidor?**

- Experiencia del usuario: Permite brindar retroalimentación inmediata al usuario sobre los datos que está ingresando, lo que mejora la experiencia del usuario al evitar que envíe datos incorrectos y recibir mensajes de error después.
- Reducción de carga en el servidor: Al validar los datos en el cliente antes de enviarlos al servidor, se reduce la carga en el servidor al evitar procesar datos incorrectos o maliciosos.
- Mejora la eficiencia: La validación en el cliente puede ayudar a reducir la cantidad de solicitudes al servidor, ya que los datos incorrectos no se enviarán en primer lugar.

2. **¿Cómo puedes saltarte la seguridad de validaciones hechas con JavaScript?**

    - Desactivar JavaScript en el navegador: Si un usuario desactiva JavaScript en su navegador, todas las validaciones del lado del cliente dejarán de funcionar, lo que le permitirá enviar datos sin ser validados.
    - Modificar el código del cliente: Los usuarios malintencionados pueden inspeccionar y modificar el código JavaScript del lado del cliente en el navegador utilizando herramientas de desarrollo como el Inspector de elementos. Pueden eliminar o modificar las validaciones para permitir el envío de datos no válidos.
    - Enviar solicitudes directamente al servidor: Los usuarios pueden utilizar herramientas como Postman o cURL para enviar solicitudes HTTP directamente al servidor, sin pasar por la interfaz de usuario que incluye las validaciones del lado del cliente. Esto les permite enviar datos directamente al servidor sin ser validados por el cliente.
    - Interceptar y modificar las solicitudes: Los usuarios pueden utilizar herramientas de proxy como Burp Suite para interceptar las solicitudes HTTP entre el cliente y el servidor. Luego, pueden modificar los datos de la solicitud antes de que se envíen al servidor, eludiendo así las validaciones del lado del cliente.

3. **Si te puedes saltar la seguridad de las validaciones de JavaScript, entonces ¿por qué la primera pregunta dice que es una buena práctica?**

    Aunque la validación del lado del cliente no es una solución completa para garantizar la seguridad de los datos, sigue siendo una buena práctica porque proporciona beneficios significativos en términos de experiencia del usuario y eficiencia del servidor, aunque debe combinarse con una sólida validación del lado del servidor para una protección completa.

Lab 7: Manejo de ramas

Uso de ramas para controlar e integrar diferentes versiones de código.

Lab 8: Introducción al back-end

Introducción al desarrollo en el back-end con node.

Laboratorio 9: Bases de Datos de Escritorio (MS Access)

Usar un DBMS (Database Management System) de escritorio, Microsoft Access, para presentar algunas de las actividades necesarias para administrar bases de datos.

Lab 10: Rutas y formas

Cómo programar desde el servidor un módulo simple de ruteo y cómo mandarle datos al servidor.

Lab 11: Express

Express, un framework de Node.js para hacer desarrollo en el back-end.

1. **Describe el archivo package.json.**
   El archivo package.json es un archivo de metadatos utilizado en proyectos de Node.js para describir el proyecto, sus dependencias, configuraciones y scripts de ejecución. Este archivo es esencial en cualquier proyecto de Node.js y es utilizado por npm (Node Package Manager) para administrar las dependencias del proyecto y facilitar su gestión.

   Las principales secciones y propiedades que puedes encontrar en un archivo package.json son:

   - name: Especifica el nombre del proyecto. Debe ser único dentro del registro npm.
   - version: Indica la versión actual del proyecto. Se utiliza para gestionar las actualizaciones y las dependencias.
   - description: Descripción breve del proyecto.
   - keywords: Una lista de palabras clave que describen el proyecto, útiles para la búsqueda y la clasificación.
   - author: El autor o autores del proyecto.
   - license: La licencia bajo la cual se distribuye el proyecto.
   - main: El punto de entrada principal del proyecto. Por lo general, es el archivo JavaScript principal que se ejecutará cuando se requiera el módulo.

- scripts: Un objeto que contiene comandos de script personalizados que pueden ser ejecutados utilizando npm run <nombre_del_script>.
- dependencies: Un objeto que lista las dependencias del proyecto. Cada dependencia está compuesta por un nombre de paquete y su versión. Estas son las dependencias necesarias para ejecutar la aplicación en producción.
- devDependencies: Similar a dependencies, pero estas son las dependencias necesarias para el desarrollo del proyecto. Incluyen herramientas, bibliotecas de prueba y otros recursos que son necesarios solo durante el desarrollo.
- repository: Información sobre el repositorio de código fuente del proyecto, incluyendo el tipo de repositorio (por ejemplo, git) y la URL.
- engines: Especifica las versiones mínimas de Node.js y npm requeridas para ejecutar el proyecto.

Lab 12: HTML dinámico

¿Qué templating engines existen para node?

- Además de EJS (Embedded JavaScript) el cual permite incrustar código JavaScript dentro del HTML, también están:

- Pug (anteriormente conocido como Jade): Basado en la indentación y se centra en la simplicidad y la legibilidad del código. Utiliza una sintaxis concisa y permite generar HTML.
-
- Mustache: Mustache es un motor de plantillas que sigue el principio de separación del modelo y la vista. Es muy sencillo y se centra en la simplicidad y la portabilidad.

- Handlebars: Se basa en la sintaxis de Mustache y permite la creación de plantillas HTML reutilizables y modulares.

- Nunjucks: Ofrece características avanzadas como herencia de plantillas, macros y filtros, lo que lo hace adecuado para la creación de plantillas complejas y dinámicas.

- EJS-mate: Una extensión de EJS que agrega características adicionales y sintaxis más conveniente, como layouts, partials y helpers, para facilitar el desarrollo de aplicaciones web.

Lab 13: MVC + Laboratorio 14: Manejo de sesiones y cookies

En el laboratorio 13 exploraremos el estilo arquitectónico Modelo-Vista-Controlador y lo implementaremos con node+express.

En el laboratorio 14 exploraremos el manejo de sesiones y cookies con node+express.

1. **¿Qué beneficios encuentras en el estilo MVC?**
    - **Separación de responsabilidades entre los componentes del sistema.**
        - El modelo se encarga de la lógica de negocio y los datos
        - La vista se encarga de la presentación de la interfaz de usuario
        - El controlador se encarga de manejar las interacciones del usuario y coordinar las acciones.

    - **Reutilización de código:** La separación de responsabilidades facilita la reutilización de código. Los modelos y vistas pueden ser utilizados en diferentes partes de la aplicación sin necesidad de modificaciones.

    - **Facilidad de mantenimiento:** Debido a la separación clara de responsabilidades, las modificaciones en un componente del sistema tienen un impacto limitado en los otros componentes. Esto facilita la comprensión del código y el mantenimiento a largo plazo de la aplicación.

    - **Facilidad de pruebas:** La separación de responsabilidades también facilita la escritura de pruebas automatizadas. Los modelos y controladores pueden ser probados de forma unitaria.

    - **Flexibilidad y escalabilidad:** Los diferentes componentes del sistema pueden ser desarrollados y modificados de forma independiente, lo que facilita la adición de nuevas funcionalidades y la adaptación a los cambios en los requisitos del negocio.

2. **¿Encuentras alguna desventaja en el estilo arquitectónico MVC?**
    - **Complejidad inicial:** La implementación de una arquitectura MVC puede requerir un mayor esfuerzo inicial en comparación con enfoques más simples. Es necesario entender y aplicar correctamente los principios y patrones de diseño asociados con MVC.

    - **Posible sobrecarga en aplicaciones pequeñas:** En aplicaciones pequeñas o simples, la introducción de la arquitectura MVC puede resultar en una sobrecarga innecesaria. En estos casos, un enfoque más ligero puede ser más adecuado.

- **Posible exceso de acoplamiento:** Si no se implementa correctamente, MVC puede llevar a un exceso de acoplamiento entre los componentes del sistema, lo que puede dificultar la modificación y la reutilización del código.

Laboratorio 15: Conociendo el Ambiente de MariaDB

En esta práctica conoceremos el ambiente en el que utilizaremos un DBMS (Database Management System) con alcance empresarial.

Laboratorio 16: Creación de constraints para instrumentar integridad referencial en MariaDB

Laboratorio 17: Interacción con la base de datos

En esta actividad comenzaremos con la interacción con una base de datos desde node.

1. **¿Qué ventajas tiene escribir el código SQL únicamente en la capa del modelo?**
   - La lógica de acceso a la base de datos está encapsulada en los modelos, lo que facilita la comprensión y el mantenimiento del código.

   - Prevención de SQL injection: Al utilizar consultas parametrizadas o consultas preparadas, se reduce significativamente el riesgo de ataques de SQL injection. Estas técnicas permiten separar los datos de los comandos SQL, lo que evita que los datos de entrada manipulen la estructura de las consultas.

2. **¿Qué es SQL injection y cómo se puede prevenir?**
   SQL injection es un tipo de ataque informático en el que se inserta código SQL malicioso en las consultas SQL de una aplicación web. Esto puede permitir al atacante obtener acceso no autorizado a la base de datos, manipular datos, eliminar o modificar tablas, o incluso comprometer todo el sistema.

   Para prevenir SQL injection, algunas prácticas de seguridad son:

   - Usar consultas parametrizadas o consultas preparadas
   - Antes de utilizar datos proporcionados por el usuario en consultas SQL, es importante validar y sanitizar la entrada para asegurarse de que cumple con el formato esperado y no contiene caracteres maliciosos.

- Limitar los privilegios de la base de datos: Es importante minimizar los privilegios de acceso a la base de datos para los usuarios y aplicaciones, limitando así el impacto de posibles ataques de SQL injection.

Laboratorio 18: Autentificación

1. **¿Qué formas de autentificación existen?**
   Además de la autenticación basada en contraseñas también existe:
   - Autenticación de dos factores (2FA) o multifactor (MFA)
   - Autenticación OAuth: OAuth es un protocolo de autorización que permite que los usuarios autoricen aplicaciones de terceros para acceder a sus recursos en línea sin revelar sus credenciales de inicio de sesión.
   - Autenticación de clave pública: En este método, se utiliza una clave pública y una clave privada para autenticar a los usuarios. El usuario proporciona su clave pública al sistema, que luego verifica la identidad del usuario utilizando la clave privada correspondiente.
   - Autenticación biométrica: Este método utiliza características físicas únicas de los usuarios, como huellas dactilares, iris, voz o reconocimiento facial, para verificar su identidad.

Laboratorio 19: Role Based Acess Control (RBAC)

1. **¿En qué consiste el control de acceso basado en roles?**
   Es un modelo de control de acceso que define los permisos de los usuarios en función de los roles que desempeñan en una organización. En lugar de asignar permisos directamente a usuarios individuales, los permisos se asignan a roles y luego se asignan esos roles a usuarios.

2. **Investiguen y describan 2 sistemas, uno que aplique RBAC y uno que no. Realicen un análisis de las ventajas y desventajas de cada uno con respecto al control de acceso.**

	Sistema 1 con RBAC	**Sistema 2 sin RBAC**
Descripción	Sistema de Gestión de Relación con Clientes (CRM): En un CRM, los roles pueden incluir "Administrador de CRM", "Ejecutivo de Ventas", "Soporte al Cliente", etc., con diferentes niveles de acceso y permisos para gestionar la información de los clientes y las interacciones.	Sistema de Reservas de Hotel sin RBAC: En un sistema de reservas de hotel sin RBAC, los permisos pueden ser asignados directamente a usuarios individuales en función de sus funciones en el proceso de reservas. Por ejemplo, un recepcionista puede tener permisos para gestionar reservas y realizar check-ins sin necesidad de ser asignado a un rol específico.

Ventajas	- Mayor seguridad en cuanto a quién tiene permiso de modificar qué o quién tiene acceso y quién no - Es más escalable, lo que significa que es fácil de escalar a medida que el número de usuarios y la complejidad del CRM aumentan. - Facilita el seguimiento de las actividades de los usuarios en el CRM.	- Al asignar permisos directamente a usuarios individuales, se puede tener una mayor flexibilidad para adaptar los permisos a las necesidades específicas - La administración de permisos puede ser más sencilla ya que no se requiere definir roles y asignar permisos a esos roles.
Desventajas	- En algunos casos, la estructura de roles predefinida en RBAC puede ser demasiado rígida y no adaptarse completamente a las necesidades específicas de la organización.	- A medida que el sistema y el número de usuarios crecen, la administración de permisos sin RBAC puede volverse más compleja y propensa a errores. - Puede ser difícil mantener la coherencia en la asignación de permisos a lo largo del tiempo.

Laboratorio 20: Consultas en SQL

Reforzar el manejo del lenguaje SQL para la manipulación y consultas en tablas.