

48 HORAS DE EXAMEN

PARCIAL: LABORATORIO DE

INGENIERÍA DEL SOFTWARE 3

Competencias o Habilidades a evaluar:

- Creación de una página web con HTML
- Creación del estilo de la página web con CSS
- Creación de código en Javascript para modificar el DOM
- Creación de un archivo JSON
- Uso del AJAX
- Creación de una funcionalidad usando SSS con PHP

El presente examen parcial pretende evaluar sus capacidades de desarrollar una aplicación web sencilla. Para ello podrán organizarse en equipos de 2 o 3 personas.

REQUERIMIENTO

Muchas personas gustan de mantener una lista de tareas que deben culminar. Se requiere crear una aplicación que mantenga el conjunto de tareas pendientes de una persona y que se puedan ir marcado como hechas.

Al iniciar la aplicación una persona digita su código (su cédula). Una tarea tiene un nombre, una fecha de creación y un valor que marca si está terminada o no.

Requerimientos:

1. Se muestra una página inicial (index.html) con un formulario que solicita la cédula del usuario. Cuando se presiona “Enviar” se envía la petición a un SSS llamado tareas.php o tareas.html
2. El servidor almacena archivos JSON con las tareas de cada usuario. El nombre del archivo es la cédula del usuario y su extensión es json. Ejemplo: 34327390.json
3. Si el código del usuario no existe en los archivos del servidor (no existe el archivo) se crea el archivo con dos tareas incompletas por defecto “Planear Semana” y “Ser feliz”
4. El servidor retorna a una página (tareas.php o tareas.html) que muestra el código del usuario y una lista de sus tareas pendientes.
5. La lista de materias se muestra como una lista en letra extra-grandes y centrada. La página es responsive.
6. Para cada tarea se muestra un cuadro de chequeo.
7. Si la tarea está TERMINADA aparece en un color gris y el texto aparece tachado.

8. Cuando el estudiante da clic al checkbox de una tarea, esta cambia de estado. Si estaba TERMINADA cambia a SIN TERMINAR y viceversa.
9. Cuando el estudiante cambia el estado de una materia se actualiza su estilo para reflejar esto.
10. Cuando el estudiante cambia el estado de una materia se envían los datos necesarios por medio de AJAX y se almacenan en el servidor, actualizando el archivo JSON.
11. En la página hay un cuadro de texto para agregar tareas y un botón para enviar. Se puede escribir la tarea en una caja de texto y cuando se presiona el botón se envía por AJAX al servidor la nueva tarea y se agrega a la lista existente (se limpia la caja de texto cuando esto tiene éxito),

Estos criterios son lo mínimo que debe tener la aplicación. Por supuesto, pueden ser creativos y agregar funcionalidades, o elementos que estén de acuerdo con las necesidades dadas.

REGLAS

1. Tienen 48 horas para entregar la aplicación.
2. Entregas después de las 48 horas no serán evaluadas
3. No se permite código copiado, el requerimiento es suficientemente abierto como para que los códigos sean demasiado similares
4. No se permite la ayuda de compañeros de semestres superiores (confío en ustedes)
5. Pueden usar cualquier tipo de recurso encontrado en internet u otras fuentes
6. Pueden usar JQuery y/o bootstrap, esto es opcional.

CRITERIOS DE EVALUACIÓN

1. (valor 1.0) La aplicación muestra páginas escritas en HTML con la estructura adecuada para mostrar y recibir los datos.
 - a. La página de inicio muestra un formulario para pedir el código del usuario
 - b. La página de tareas muestra la lista
 - c. La página muestra una descripción del uso de la aplicación
2. (valor 1.0) La aplicación usa estilos CSS para mostrar los datos en la página HTML.
 - a. Los estilos son creados de manera acorde a las necesidades de la aplicación (colores, tamaños de letra)
 - b. Los estilos son usados por los elementos de la página HTML correctamente (las tareas se tachan y cambian de color)
3. (valor 1.0) La aplicación retorna al cliente y cambia, usando SSS, los datos de las tareas del usuario que solicita.
 - a. La aplicación crea el archivo con materias cuando no encuentra el código del estudiante.
 - b. La aplicación embebe código SSS en la página HTML para devolver los datos al cliente de manera dinámica (con los datos del archivo)
 - c. La aplicación actualiza el estado de las tareas cuando se solicita por AJAX
4. (valor 1.0) La aplicación usa AJAX para modificar el DOM y enviar peticiones al servidor.

- a. Al dar clic en la tarea, esta actualiza su estilo
 - b. Al dar clic en la tarea se envía por AJAX una petición al servidor con los datos necesarios.
5. (valor 1.0) Se usa JSON para almacenar los datos de las materias y retribuirlos
 - a. Los archivos JSON están bien formados.
 - b. La estructura del archivo JSON se lee correctamente para mostrarse en el HTML.
 - c. El archivo JSON se actualiza correctamente

ENTREGA

El equipo debe:

- Crear un repositorio GitHub con el código de la aplicación
- Se debe crear una “etiqueta” ([tag](#)) de git llamada v1.0 la cual será clonada por el profesor para su evaluación. *Un tag es un punto particular en la historia de un repositorio que marca un “hito” en las versiones del sistema. Sirve para marcar una versión especial del código (ejemplo, el lanzamiento de una nueva versión). No olvide subir la tag al repositorio remoto usando **git push --tags***
- El tag no debe tener fecha y hora posterior a la finalización de las 48 horas de examen parcial.
- Entregue por medio del Moodle, la dirección del repositorio y el nombre del tag.
- Se sustentará con el profesor el examen parcial clonando el repositorio