



Trabajo práctico N° 2



Tecnologías del lado del servidor

Fecha de entrega:

Chivilcoy: 15/04/20

Luján: 16/04/20

Objetivo: entender cómo un formulario HTML se traduce en una petición HTTP y cómo la misma es procesada por una aplicación.

1. Elabore una aplicación que presente al usuario un **formulario implementado por HTML** para la carga de los datos de una persona que solicita turno médico. El formulario deberá disponer de los siguientes **campos**:
 - a. Nombre del paciente (*obligatorio*)
 - b. Email (*obligatorio*)
 - c. Teléfono (*obligatorio*)
 - d. Edad
 - e. Talla de calzado (desde 20 a 45 enteros)
 - f. Altura (usando la herramienta de deslizador)
 - g. Fecha de nacimiento (*obligatorio*)
 - h. Color de pelo (Usando un elemento select con las opciones que usted considere adecuadas)
 - i. Fecha del turno (*obligatorio*)
 - j. Horario del turno (Entre las 8:00 hasta las 17:00 con turnos cada 15 minutos)
 - k. 2 botones: Enviar y Limpiar.

Todos los elementos del formulario deben **validarse del lado de cliente y servidor**, con el formato que mejor se ajuste y permitan HTML y PHP. Además, tomar en cuenta de validar que los datos ingresados se encuentren en los rangos especificados. ¿Por qué cree usted que se requiere validar los datos en ambos extremos de la comunicación?

2. Extienda el ejercicio anterior para que al enviar el formulario mediante el método **POST** se muestre al usuario un **resumen del turno**.
3. Realice las modificaciones necesarias para que el script del punto anterior reciba los datos mediante el método **GET**. ¿Qué diferencia nota? ¿Cuándo es conveniente usar cada método?



Consejo: Utilice las herramientas de desarrollador de su Navegador (Pestaña Red) para observar las diferencias entre las diferentes peticiones.

4. Agregue al formulario un campo que permita **adjuntar una imagen**, y que la etiqueta del campo sea Diagnóstico. El campo debe **validar** que sea un tipo de imagen valido (.jpg o .png) y será optativo. La imagen debe **almacenarse en un subdirectorío** del proyecto y también debe mostrarse al usuario al mostrar el resumen del turno del ejercicio 2. ¿Qué sucede si 2 usuarios cargan imágenes con el mismo nombre de imagen? ¿Qué mecanismo implementar para evitar que un usuario sobrescriba una imagen con el mismo nombre?
5. Utilice las **herramientas para desarrollador** del navegador y observe cómo fueron **codificados** por el navegador los datos enviados por el navegador en los dos ejercicios anteriores. ¿Qué diferencia nota?
6. **Agregar persistencia** al sistema de turnos. Todos los datos del formulario deben almacenarse mediante algún mecanismo para poder ser recuperados posteriormente. Crear una **nueva vista** que le permita a un empleado administrativo visualizar todos los turnos en una tabla. La tabla debe incluir los siguientes campos:
 - a. Fecha del turno
 - b. Hora del turno
 - c. Nombre del paciente
 - d. Teléfono
 - e. Email
 - f. Link a la ficha del turno (la ficha se implementa en el siguiente punto)

Esta página y la del formulario del punto 2 deben contar con una **barra de navegación** que permita ir a una u otra pantalla.

Además, al procesar el formulario en el lado servidor, **el sistema asigne un número de turno** (que no debe repetirse).

Para generar el sistema de persistencia, se aconseja estudiar algún mecanismo de **serialización** de datos.¹

¿Cómo **relaciona** la imagen del turno con los datos del turno? Comente alternativas que evaluó y opción elegida.

7. Construya la **vista de ficha de turno**. Dicha vista debe permitir acceder al turno y mostrar todos sus datos, recuperados del mecanismo de persistencia elaborado en el punto anterior. ¿Cómo se identifica y discrimina un turno de otro? Debe **funcionar el link** a la ficha que se encuentra en la tabla de turnos. Recuerde **agregar un enlace** para volver a la tabla de turnos.

¹ Opciones: <http://php.net/manual/es/function.serialize.php>; <http://php.net/manual/es/function.json-encode.php>



Bibliografía

- PHP Tutorial - <https://www.w3schools.com/php/>
- Manual de PHP - <https://www.php.net/manual/es/>
- Server-side website programming first steps - https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps
- Protocolo HTTP - <https://developer.mozilla.org/es/docs/Web/HTTP>
- Sebesta, R. (2010). Programming the World Wide Web. Pearson/Addison-Wesley. Capítulo 9
- Shklar, L., & Rosen, R. (2003). Web Application Architecture: Principles, Protocols and Practices. Wiley. Capítulos 3, 4 y 5.