Programación Web 2 - 2do 2021

TP N°1: Repaso Programación Web 2

Ejercicio 1:

Realizar un análisis sobre la Plataforma MIeL, su contenido estético, su funcionalidad y la tecnología involucrada (donde la pueda determinar).

- Detalle el análisis realizado
- A partir de ello proponga al menos una mejora en cualquiera de los tres aspectos mencionados.
- La propuesta debe ser escrita (no es necesario maquetar ni programar) y debe establecer las características de la mejora, su objetivo, y una descripción de las deficiencias que resolvería.

Ejercicio 2:

Según lo observado por Usted en el Ejercicio 1 se le solicita construir una maqueta en HTML y CSS, poniendo en juego sus propuestas, de la Home Page de MIeL y de la primera página interna.

TP N°2: Introducción a PHP

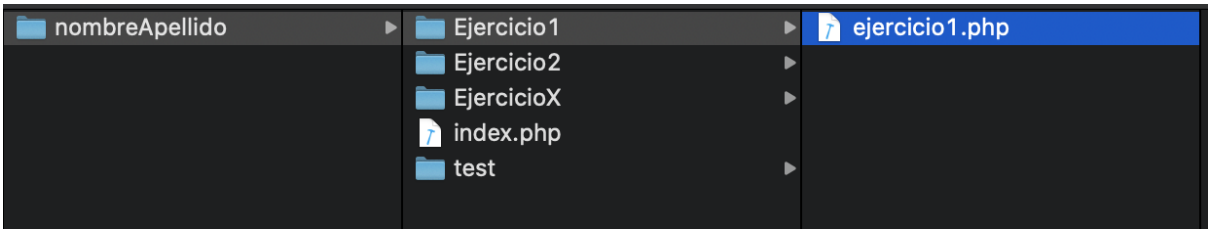
Pasos preliminares:

- Configure un servidor web que permita interpretar PHP
- Configure una IDE que permita desarrollar fácilmente (PHPStorm)

Pautas de resolución:

Todos los ejercicios constan de un enunciado, el cual se debe resolver del siguiente modo:

- La resolución del ejercicio debe estar en una función o clase, que se llamará desde el index.php . Dicha función o clase debe estar en otro archivo e incluirse en el index.php mediante un requiere_once()
- La entrega debe contener la siguiente estructura de carpetas (dejar de lado carpeta test) :



Ejercicio 1: Semáforo

Cree una función llamada Semaforo, que recibe por parametro un cólor cómo texto (“rojo” “amarillo”,”verde”). Dicha función devolverá el estado que corresponde: “frene”, “precaución”, “avance” o “estado desconocido” ante un caso no esperado.

a) función semaforo_a($color): Resuelva la solución utilizando if else
b) función semaforo_b($color): Resuelva la solución utilizando if inline (return ?:)
c) función semaforo_c($color): Resuelva la solución utilizando switch

Ejercicio 2: Binomio cuadrado perfecto

Cree una función llamada binomioCuadradoPerfecto que realice la ecuación de dicha problemática: recibe dos parámetros y devuelve el cuadrado de la suma de ambos $(a+b)^2$

a) función binomioCuadradoPerfecto_a($a, $b): Resuelva la solución utilizando la función de potencia
b) función binomioCuadradoPerfecto_b($a, $b): Resuelva la solución utilizando la fórmula desarrollada del binomio: $a^2 + 2*a*b + b^2$

Ejercicio 3: Concatenar textos

Cree una función concatenar($texto1, $texto2) que reciba dos textos como parámetro y devuelva ambos textos concatenados como uno solo.

Ejercicio 4: Incrementar

Cree una función llamada incrementar, que reciba una variable y sin devolver nada como retorno de la función, el valor del parámetro haya sido incrementado en 1
(Ver pasaje de parámetros por referencia)

Ejercicio 5: Sumatoria

Cree una función sumatoria que reciba un vector cómo parámetro, y devuelva la suma de todos sus valores.

a) función sumatoria_a($array): Resuelva la solución utilizando la estructura de control for
b) función sumatoria_b($array): Resuelva la solución utilizando la estructura for each
c) función sumatoria_c($array): Resuelva la solución utilizando la estructura de control while

Ejercicio 6: Nombre completo

Cree una clase llamada Saludar, la misma tendrá un constructor que reciba nombre, apellido de una persona.

Dicha clase debe implementar el método saludoFormal($horario) el cual debe responder concatenado al nombre un prefijo dependiendo del horario:
05hs a 13hs “Buenos días”
13hs a 21hs “Buenas tardes”
21hs a 05hs “Buenas noches”

Ej. para clase instanciada para Ezequiel Perez, y parámetro 9hs: “Buenos días Ezequiel Perez”

Dicha clase debe implementar también el método saludoInformal($horario) el cual debe responder sin el apellido, iniciando con un “hola” por delante y al finalizar concatenar “que tengas un ...” saludo perteneciente al horario .

Ej. para clase instanciada para Ezequiel Perez, y parámetro 9hs:
“¡Hola Ezequiel! Que tengas un buen día”

Ejercicio 7: Integrando PHP, HTML y CSS

Cree una web a modo de demostrar las funciones resueltas en el trabajo práctico.
Cree un archivo index.php, que contenga cómo título su nombre, y apellido.
A continuación liste los ejercicios a resolver, y en cada caso comente con qué valores se llama a la función, la llame efectivamente y muestre por pantalla el resultado utilizando la función “echo” .
Cómo resultado se debería visualizar algo similar al ejemplo al ingresar a localhost:

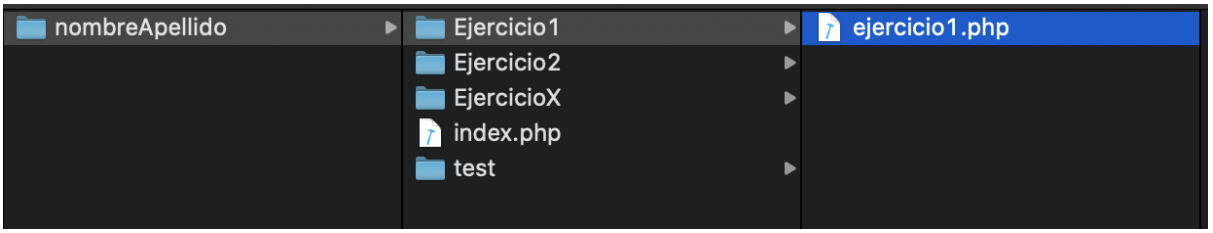
TP N°3: Algoritmos y funciones del lenguaje

Pasos preliminares:

- Configuración utilizada para resolver trabajo práctico N°2
- W3CSS para maquetar el estilo de la web (https://www.w3schools.com/w3css). Puede utilizar un template de la web de W3CSS, el ejemplo que se subió a MIeL en el "entorno docker" o crear un maquetado propio.

Pautas de resolución:

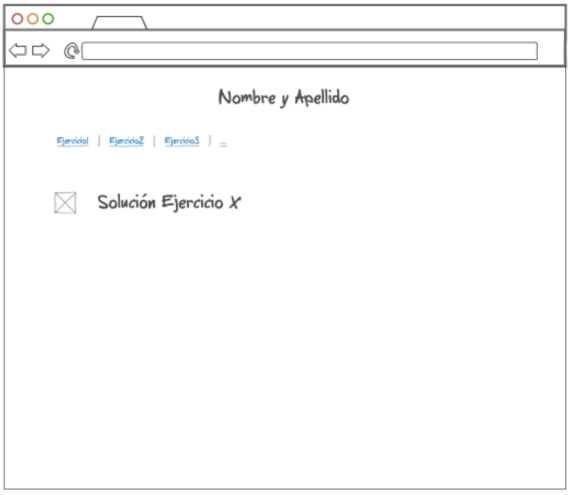
- Cada ejercicio muestra un bosquejo de cómo debe verse la pantalla. El ejercicio debe cumplir cómo mínimo con esa estructura
- La entrega debe contener la siguiente estructura de carpetas (obviar carpeta test):



Ejercicio 0: Punto de partida

Realizar una web home en el archivo index.php, que contenga:

- Su nombre y apellido como título
- Un menú horizontal para acceder a los ejercicios por su nombre
- Un iframe con el archivo PDF que contiene el enunciado
- La pantalla de cada ejercicio debe ser igual a esta, sólo que mostrar en el body la resolución. (Utilizar includes para el header y footer)



Ejercicio 1: Insta-gramo

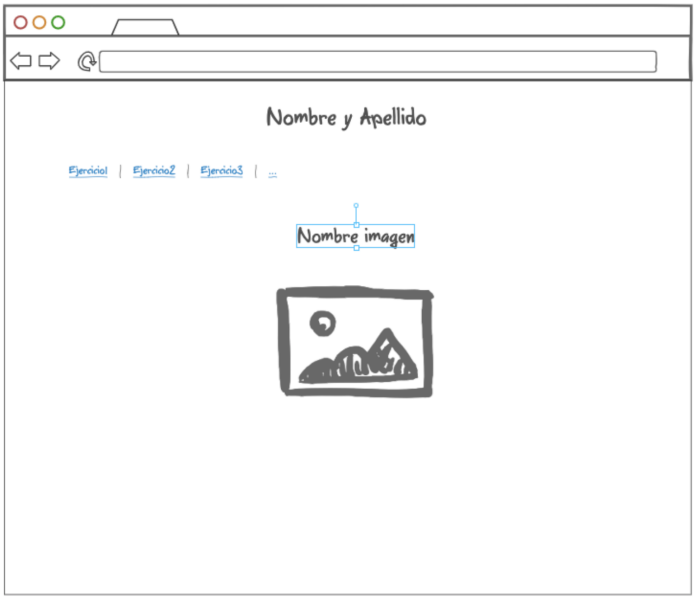
Realizar una web que muestre todas las imágenes que contiene en la carpeta “/imagenes” con su respectivo nombre de archivo como pié de imagen. Al final de dicha web debe haber un formulario que permita subir una imagen con un nombre a designar.

Tip: Luego de cargar la imagen, debe volver a la misma página, sin tener que tocar links “volver” o cosas similares (Como en todo insta-gramo!)
Tip: Entregar con 3 imágenes cargadas

Ejercicio 2: Insta-reciclado (para cuidar el medio ambiente):

Reutilizando el ejercicio anterior, realizar una web que liste todos los nombres de imagenes que contiene en la carpeta /imagenes cómo link, que al hacer clic, lleve a una segunda pantalla donde efectivamente se muestre dicha imagen.

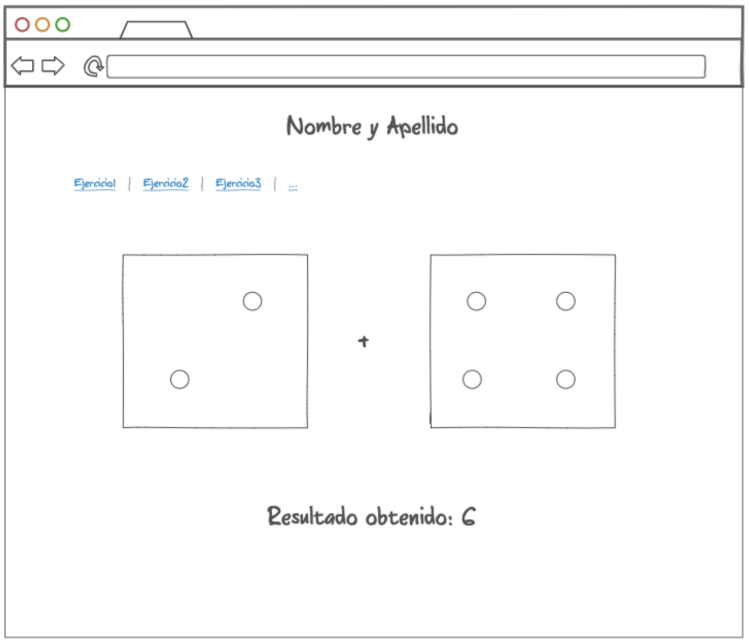
Tip: Entregar con 3 imágenes cargadas



Ejercicio 3: Lanzar dados

Realizar una web que permita indicar la cantidad de dados a lanzar (mediante input de tipo option) y al tocar un botón “lanzar dados” pase a una segunda pantalla donde muestre los dados lanzados como imagen y la suma de sus valores como puntaje obtenido.

Tip: Buscar una función en php que devuelve un número random para resolver el ejercicio.



Ejercicio 4: Contador de visitas… extraterrestres

Crear una web que permita ingresar en un formulario, los datos de ingreso a nuestro planeta: nombre del visitante y planeta perteneciente (mediante un combo).

Una vez enviado el formulario, muestra una segunda página que liste todos los visitantes que cruzaron este control interplanetario, y un contador del total de visitas, que no pertenecen al planeta Tierra.

Tip: Almacenar en un archivo (en el servidor) cada una de las visitas, para dicho almacenado, pueden utilizar formato json, archivo separado por comas, cada visita en un archivo o algún otro tipo de su preferencia. No utilizar base de datos

Tip: Es recomendable, realizar el cálculo de visitas, cuando leen el archivo, justo antes de mostrarlo y no persistir esta sumatoria

Ejercicio 5: ConociendINIs con el Menú no saludable

Mostrar en pantalla un formulario con checkbox que permita elegir más de un paso en el menú de hoy: Entrada, plato principal, acompañamiento y postre.
Una vez enviado el formulario, lea el archivo menu.ini, y muestre en pantalla sólo los pasos del menú que solicitó el comensal:

menu.ini
entrada=Snacks
plato_principal=Milanesa
acompañamiento=Papas fritas
postre=helado

Tip: Utilizar función parse_ini_file que permite leer archivos .ini de modo sencillo
Tip: Si la letra ñ es un problema, no la utilice
Tip: Puede cambiar los menú por lo que prefiera

Ejercicio 6: La Matrix… digo, Matriz

Solicite mediante un formulario, la dimensión de una matriz cuadrada.
En la siguiente página, cree esa matriz de NxN y realice las siguientes acciones:

a) Recorrer la matriz con un sólo FOR y mostrar en pantalla los valores que componen la diagonal principal (1,12,23,etc)
b) Recorrer la matriz con un sólo FOR y mostrar en pantalla los valores que componen la diagonal secundaria (10,19,28,etc)
c) Recorra la matriz (Ahora si con 2 for) y sume todos los valores que contiene. Muestre el resultado

Tip: En los pasos a y b, no utilice un while, ni un flag, sólo con el contador del for debe alcanzar. Analice las posiciones de la matriz que componen las diagonales ;)

TP N°3: Pokédex (ABML con login por sesión y persistencia a BD)

Realizar una web que represente una Pokédex (listado de pokémon) y permita realizar búsqueda sobre la misma. Además, el sistema debe permitir que el administrador se loguee al mismo y pueda realizar altas, bajas y modificaciones de los elementos. Ejemplo de donde pueden tomar datos: http://es.pokemon.wikia.com/wiki/Lista_de_Pok%C3%A9mon

Tip: Si no les agrada o no conocen pokémon, pueden reemplazar el contenido por el de otro tema que los motive, siempre que se mantengan todas las funcionalidades (Ej: Música, Jugadores de Fútbol, auto fantástico, Meteoro, etc.)

Casos de uso:

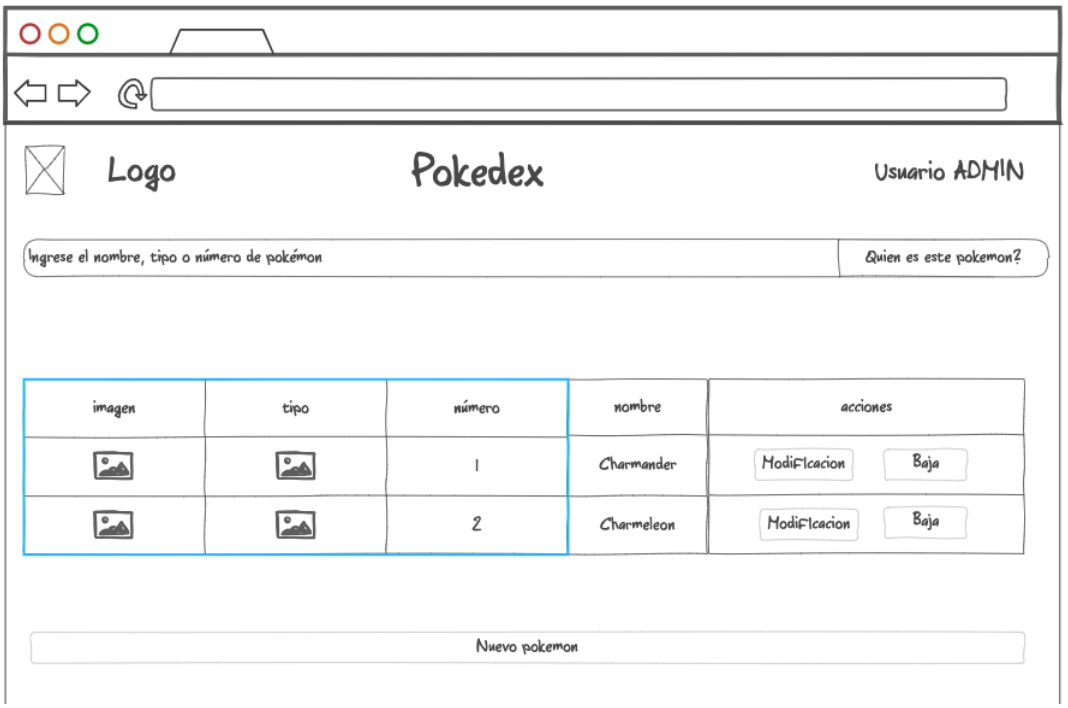
- Si se busca un pokemon existente, mostrar sólo ese.
- Si se busca un pokemon inexistente, informar "pokemon no encontrado" y - mostrar la lista de todos ellos.
- Si no se busca nada, mostrar la lista de todos ellos
- Al ingresar al pokemon buscado, mostrar todos sus datos en una página completa
- El sistema debe permitir loguear usuarios a modo de administradores
- El tipo de pokemon (fuego, agua, hierba) debe mostrarse con una imágen. No como texto

- El usuario logueado: puede dar de alta un pokemon
- El usuario logueado: puede dar de baja un pokemon
- El usuario logueado: puede editar un pokemon

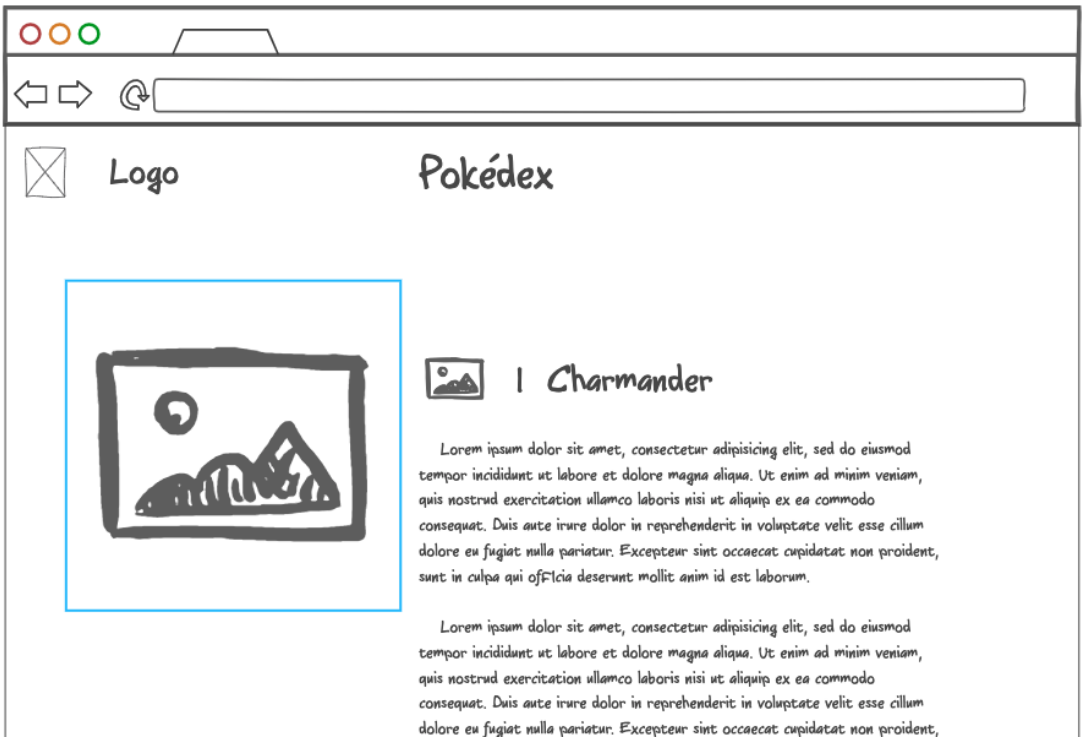
Guía de maquetado a modo de ejemplo:

Página principal de Búsqueda:

Página principal de Búsqueda para administrador logueado:



Página de visualización del pokemon (interna):



Requisitos técnicos:

- La web debe ser visualizada correctamente desde un celular o computadora (ser responsive). Utilice w3css, bootstraps, materialize o similar
- Los datos mínimos de cada pokémon serán:
    - id de base de datos autoincremental (no puede modificarse)
    - Número identificador único (no autoincremental, se debe cargar a mano y se puede editar por el administrador)
    - Imagen (subir imagen al servidor y guardar en la base de datos la ruta a la carpeta y nombre del archivo)
    - Nombre
    - Tipo (seleccione al menos 4 valores posibles)
    - Descripción
    - agregar todos los datos extras que deseen
- Debe trabajarse sobre una base de datos MySql (MariaDb o postgre si prefieren).
- El backend debe realizarse en PHP
- Toda la lógica debe residir del lado del servidor en PHP
- No utilice Ajax ni funciones complejas de js
- Entregar un archivo comprimido .zip que contenga:
    - código del sitio
    - sql para crear la base del sitio y sus datos
    - debe poder corregirse descomprimiendo el zip sobre el raíz del apache y accediendo a localhost<puerto>:// chequee en la computadora de algún compañero que esto funcione